

Youtube Playlist:

Java

<https://youtube.com/playlist?list=PLcaKRNhUYNxIKxQGzfJPymWFCBrc8KusX>

Sql/sqlserver

<https://youtube.com/playlist?list=PLcaKRNhUYNxLwmQeqsP2wBiXZTkdpArc>

<https://youtube.com/playlist?list=PLcaKRNhUYNxLrMr4gTyiT79iQ5sBpEPMp>

Csharp

<https://youtube.com/playlist?list=PLcaKRNhUYNxJCIUb5DTChHjuOMcO4IRMW>

https://youtube.com/playlist?list=PLcaKRNhUYNxJiUSvHqlro_5GMEAf4XIPc

Selenium

<https://youtube.com/playlist?list=PLcaKRNhUYNxJcigZYw9N7PZqE7EfCEUHT>

Html css javascript

https://youtube.com/playlist?list=PLcaKRNhUYNxLZCn1WP_Tu02TpW38PL2vF

Design principle

<https://youtube.com/playlist?list=PLcaKRNhUYNxINlwGY6GszcJH6eAI7dZvY>

Hacker rank

https://youtube.com/playlist?list=PLcaKRNhUYNxLQcsDqRMOH4_vruqljzMdm

Java unit testing code quality

https://youtube.com/playlist?list=PLcaKRNhUYNxK93DH4HTpoR_rrFUVBcdjH

RDP KBA AND SBA

https://youtube.com/playlist?list=PLcaKRNhUYNxITj7f2iTPXSrNkVOr_K3_W

Assimilation

<https://youtube.com/playlist?list=PLcaKRNhUYNxluV-2ZNViFhKJ9KkqrGAHU>

Micro certification

https://youtube.com/playlist?list=PLcaKRNhUYNxJ5kqi8qHzy4lblx9p_JH8

Java spring core maven

<https://youtube.com/playlist?list=PLcaKRNhUYNxJ184Zt-t4GSbzGL1PRRtEL>

Azure cloud

https://youtube.com/playlist?list=PLcaKRNhUYNxKIFbCLq872WDV-ZW_3WffS

Stepplus english

https://youtube.com/playlist?list=PLcaKRNhUYNxJRV4RIDhGh-sgd_IgAxOoz

ICT

1. [Metalapps sales automation](#)
2. [Expense management system](#)
3. [Electricity bill automation](#)
4. [Insurance project](#)
5. [Restaurant management system](#)
6. [Pages for bank](#)
7. [Topseller](#)
8. [Employee management](#)
9. [Bank account management](#)
10. [Design the display](#)

Code challenge

1. [Ebill post/html helper](#)
2. Expense management system
<https://youtu.be/EHitachOneA>
3. Food delivery app
<https://youtu.be/QWIVvA6LNHfg>

Entity framework

1. [Student details using codefirst](#)
2. [Bookentry fluent api](#)
3. [Course repository](#)

HANDSON

1. [Vehicle repository](#)
2. [Mobile repository](#)
3. [Shipment fluent code](#)
4. [Seminar ticket booking](#)
5. [Workshop enrolment app](#)
6. [Quiz competition report jagged array](#)
7. [Account details](#)
8. [Extract Book code](#)
9. [Calculator program](#)
10. [Balls bowled](#)
11. [Generate bill details](#)
12. [Game inheritance](#)
13. [Find age of person](#)
14. [Load validator](#)
15. [Copy file contents](#)
16. [User details](#)

17. [Find square and cube](#)
18. [String concatenate](#)
19. [Append text to file](#)
20. [Multiplication tables chart](#)
21. [Vehicle released year](#)
22. [Reverse a sentence](#)
23. [Display long number](#)
24. [Openable interface](#)
25. [Boolean result](#)
26. [Change a word](#)
27. [Max value of single byte](#)
28. [Double discount](#)
29. [Grade calculation](#)
30. [Pay calculation](#)
31. [Salary greater than average xling](#)
32. [Capture score](#)
33. [Sales generator - top product](#)
34. [Club member](#)
35. [Filter employee list](#)

Design principles

[Click here](#)

Step Plus english assessment

<https://youtu.be/RoWQO2-zoHg>

Also see video description for

Hacker rank solutions

https://youtube.com/playlist?list=PLcaKRNhUYNxLQcsDqRMOH4_vruqljzMdm

1. [Outlook fan market](#)
2. [Number of days](#)
3. [Ac maintenance](#)
4. [Good harvest rice](#)
5. [York admission](#)
6. [Agape food delivery](#)
7. [Health care center](#)
8. [Royal wedding planners](#)
9. [Platinum properties](#)
10. [Bank transactions](#)
11. [Truyum UI](#)
12. [Discount price](#)
13. [Ebook grand sale](#)
14. [Piggy bank savings calculator](#)
15. [Book form](#)
16. [Football league performance](#)
17. [Dream theme park](#)
18. [Broadband and payment portal](#)
19. [DAC bank loan automation](#)
20. [Book a show](#)
21. [Palindrome check mahant](#)
22. [Boat riding rate card](#)
23. [Simple calculator](#)
24. [Learning material styling](#)
25. [Reduced interest estimation](#)
26. [Bill calculator](#)
27. [ACTB CONNECTION PORTAL](#)
28. [Trainer feedback chart](#)
29. [Feedback details](#)
30. [EMI calculator](#)
31. [Consignement booking -Air cargo](#)
32. [Frequency calculation](#)
33. [Highest enrolment of policy](#)
34. [Asset summary](#)
35. [Address book](#)
36. [User detail](#)
37. [Total expense](#)
38. [Email validation](#)
39. [KASA IT FINISHING SCHOOL](#)
40. [Customer-address-user details](#)

Please do subscribe share and support

<https://youtube.com/channel/UC2EQRc2Zic5PDIKaaxHwR7A>

Today's update:

1.bootstrap typography

<https://youtu.be/9scwrSsn4FE>

2.table

<https://youtu.be/45jTKlfdFFI>

3.panel

<https://youtu.be/bXOrm9Y0e-I>

4.tringet

<https://youtu.be/5NEzYeRvAOE>

5.page layout

<https://youtu.be/W4afhb-bEtY>

6.inline form

<https://youtu.be/AkSkV9m7YeE>

7.responsive wbe page

https://youtu.be/WHOh6_JYsDM

8.navigation bar

<https://youtu.be/aokBRjldl9o>

9.well and glyphicons

<https://youtu.be/lmZqchthQb4>

Java cc

1. [Java sba genc](#)
2. [Search for school](#) project
3. [Annual salary](#)
4. [Hotel booking](#)
5. [Shape area volume](#) calculator
6. [Bank app](#)
7. [Insurance premium generator v2](#)
8. [Health insurance](#)
9. [Trainer attendence](#) marking system
10. [Stationary bill](#)
11. [Creditcard admin](#)
12. [CD gallery](#) system
13. [Credit Card](#)
14. [Singapore tourism](#)
15. [Exam scheduler generate assessment](#)
16. [Insurance premium generator](#)
17. [Travel request](#)
18. [Ecommerce](#)
19. [Credit Card validator](#)
20. [Irctc set1](#)
21. [Car store](#)
22. [Event management](#)
23. [Construction cost time estimation](#)
24. [Next recharge date](#)
25. [Construction cost estimation alternative](#)

1. Parcel service spring

<https://youtu.be/927N-SR-ICQ>

2. Eshopping

<https://youtu.be/di98Yk6RNcc>

3. exam scheduler

<https://youtu.be/71nry6i6jqk>

4. hotel booking

<https://youtu.be/LysD-nQs6pQ>

5. traffic chellan

<https://youtu.be/lECKUiUQGYM>

6. Furnturn

<https://youtu.be/T5jri6R8px4>

7. [Cheque payment process](#)

ICT:

1. Tax claim - TMS
<https://youtu.be/gaZnvBKWM0>
2. [A new spa](#)
3. [Bon bon publishing](#)
4. [Annual rainfall](#)
5. [Electricity bill automation](#)
6. [Creditcard payment](#)

Code challenge:

1. [Passenger amenity](#)
2. [Perform calculation](#)
3. [Oil store](#)
4. [Alliteration change the case](#)
5. [Power progress](#)
6. [Substitution cipher technique](#)
7. [Bank account interface](#)
8. [Check number type](#)
9. [Silver health plan insurance](#)
10. [Hunger eats](#)
11. [Payment inheritance](#)
12. [Grade calculation](#)
13. [Employee eligible for promotion](#)
14. [Batting average](#)
15. [Numerology number](#)
16. [Zee zee bank](#)
17. [Club member](#)
18. [Movie ticket](#)
19. [Singapore tourism](#)
20. [Employee audit](#)
21. [Travel agency](#)
22. [Book manipulation](#)
23. [Pair of two digits](#)
24. [Employee loan](#)
25. [Air voice registration](#)
26. [Query data set](#)
27. [Call details](#)
28. [Annual salary](#)
29. [Change the case](#)
30. [Alternative number difference](#)
- 31.

Handson:

1. [Extract book](#)
2. [Least offer](#)

3. [String concatenation](#)
4. [Vehicle loan insurance](#)
5. [Insurance bazar](#)
6. [Fuel consumption](#)
7. [Account Manipulation](#)
8. [Search for trains jdbc](#)
9. [Bank account details](#)
10. [Highest placement](#)
11. [Ticket price calculation](#)
12. [Array Manipulation using multiple catch](#)
13. [Contact details of hosteller](#)
14. [Register a candidate user defined exception](#)
15. [Phonebook Manipulation](#)
16. [Student detail constructor](#)
17. [Add flight using jdbc](#)
18. [Employee loan eligibility polymorphism](#)
19. [Player selection using jdbc](#)
20. [Number of new words](#)
21. [Travel agency](#)
22. [Retrieve data from file](#)
23. [Validate name](#)
24. [Employee audit](#)
25. [Mall parking system](#)
26. [Placement enrollment count](#)
27. [Employee promotion](#)
28. [Fruit basket estimation](#)
29. [Employee loan eligibility alternative](#)
30. [Least offer](#)

Junit:

1. [Movie ticket booking](#)
2. [Test using assertion](#)

1.[jquery window resize](#)

2.[drag an object jquery](#)

3.[three division jquery](#)

4.[checkbox using jquery](#)

5.[new singapore tourism](#)

6.[springjpa movire cruiser](#)

[7.springjpa truyum](#)

[8.new creditcard code](#)

1. [Selenium week1 all handson](#)
2. [Selenium form registration](#)
3. [Selenium work with multiple windows](#)
4. [Selenium working with javascript executor](#)
5. [Selenium cargo poi with pom](#)
6. [Online basket xpath](#)
7. [Pepperfry automation miniproject](#)
8. [Dropdown menu miniproject](#)
9. [Upload image to search miniproject](#)
10. [User registration basic web element](#)
11. [Student registration basic web element](#)
12. [Registration interview eligibility](#)
13. [Customer registration](#)
14. [Agent registration basic web element](#)
15. [Address book basic web element](#)
16. [Shipping details basic webelement](#)
17. [Software license dropdown handling](#)
18. [Css locator](#)
19. [Alert](#)
20. [Drag and drop](#)
21. [Work with javascript executor](#)
22. [Work with multiple windows part2](#)
23. [Irctc flight search automation](#)
24. [Irctc trian search automation](#)
- 25.

Selenium handson

1. [Applying poi](#)
2. [Applying POM](#)
3. [Applying poi with pom 2](#)
4. [Applying poi with pom 1](#)

Dynamic Xpath

1. [Online basket](#)
2. Xpath [work with javascript executor](#)
3. [Users submit and locate](#)
4. [Work with javascript executor](#)
5. [Work with alert box](#)
- 6.

1. [Carpool update booking table](#)
2. [Carpool vehicle details](#)
3. [Password generation](#)
4. [Movie master](#)
5. [Report on department salary](#)
6. [Extract the employee and department salary](#)
7. [Display employee having salary greater than average salary of their department](#)
8. [Cricket number of players in city](#)
9. [Hospital maximum fees paid](#)
10. [Hospital alter sql](#)
11. [Car maximum time driven](#)
12. [Retrieve department salary](#)
13. [Cricket all sql](#)
14. [Employee count](#)
15. Dream home flat details based on year
<https://youtu.be/cZolj5JoK-8>

16. Dream home customer name details based on total cost
<https://youtu.be/rMkZTEUbEJ4>
17. Car pool create booking table
<https://youtu.be/NVjkFgTMyUE>

18. Carpool sql all
<https://youtu.be/LLxAxVbpqtg>

19. Hospital total fees received based on shift
<https://youtu.be/h8VeL44MUIU>

20. Supplier 1.2
<https://youtu.be/T9Zuv1WvmNE>

21. Insurance policy related all sql
https://youtu.be/5XfCZ7XT_FI

22. Customer details based on coupon
<https://youtu.be/cZxN4tLjk3I>

Code challenge:

1. [Hospital change the datatype](#)
2. [Hospital update student record](#)
3. [Hostel insert student record](#)

4. [Movie modify the datatype](#)
5. [Concatenating details](#)
6. [Hospital add a new column](#)
7. [Minimum and maximum discount](#)
8. [Patient details based on month](#)
9. [Student detail in capital case](#)
10. [Patient appointment detail based on reason](#)
11. [Number of appointments](#)
12. [Student room details](#)
13. [Event hall customer having average cost](#)
14. [Dream home max cost of flat](#)
- 15.

RDBMS ALL:

[CLICKHERE](#)

Event hall SQL

<https://youtu.be/BoqeBXT7ILw>

https://youtu.be/exqu3_AJQv8

<https://youtu.be/zhZMStXnQAo>

<https://youtu.be/KifmUVhE6rl>

<https://youtu.be/5alcxCDZXUI>

<https://youtu.be/GMKgYWbcZrE>

PIZZA SQL

https://youtu.be/nyP48K_ytB8

<https://youtu.be/YCOA6n1q-B0>

https://youtu.be/J_Uad0PoB9U

Insurance sql

<https://youtu.be/W8BbKdW48tc>

<https://youtu.be/F-MAItsG0rA>

https://youtu.be/5XfcZ7XT_FI

Cricket sql

<https://youtu.be/X6Sn7rXSO0I>

<https://youtu.be/sH9CDpfmqcs>

<https://youtu.be/vec1CGSqlp0>

<https://youtu.be/94b2a-NzN9A>

<https://youtu.be/MAxEbGJmpes>

Rdbms

<https://youtu.be/PbrTZvePDbo>

<https://youtu.be/P2NhAtpywnl>

<https://youtu.be/l gj67XxVUNE>

https://youtu.be/vs_MAgbyi90

Ddl drl:

<https://youtu.be/mP2X9RilRg0>

<https://youtu.be/XL9IBPgB2jA>

<https://youtu.be/tk9OZ-Psr0Y>

<https://youtu.be/n9bDVnr9x0g>

<https://youtu.be/8QnIVzl01UE>

<https://youtu.be/2XXxzUf8pww>

1. [Sqlserver cc all](#)
 2. [Procedure to display all the department](#)
 3. [Create table with foreign key constraints](#)
 4. [Employee working in new york](#)
 5. [Alter supplier table with check constraints](#)
 6. [Employee with reporting manager](#)
 7. [Display employee of the specific department](#)
 8. [Employee with reporting manager 2](#)
 9. [Worker and admin departments](#)
10. TOTAL SALARY CC

<https://youtu.be/wnaZ-4sDTp8>

BANK

```
create database bank;
use bank;
```

```
CREATE TABLE customer_master(  
CUSTOMER_NUMBER VARCHAR(6),  
FIRSTNAME VARCHAR(30),  
middlename VARCHAR(30),  
lastname VARCHAR(30),  
CUSTOMER_CITY VARCHAR(15),  
CUSTOMER_CONTACT_NO VARCHAR(10),  
occupation VARCHAR(10),  
CUSTOMER_DATE_OF_BIRTH DATE,  
CONSTRAINT customer_custid_pk PRIMARY KEY (CUSTOMER_NUMBER));
```

```
CREATE TABLE branch_master(  
branch_id VARCHAR(6),  
branch_name VARCHAR(30),  
branch_city VARCHAR(30),  
CONSTRAINT branch_bid_pk PRIMARY KEY (branch_id));
```

```
CREATE TABLE account_master  
(account_number VARCHAR(255),  
customer_number VARCHAR(255),  
branch_id VARCHAR(255),  
opening_balance INT(20),  
account_opening_date DATE,  
account_type VARCHAR(10),  
account_status VARCHAR(10),  
PRIMARY KEY (account_number),  
FOREIGN KEY (customer_number) references customer_master(customer_number),  
FOREIGN KEY (branch_id) references branch_master(branch_id));
```

```
CREATE TABLE transaction_details  
transaction_number VARCHAR(6),  
account_number VARCHAR(6),
```

```

date_of_transaction DATE,
medium_of_transaction VARCHAR(20),
transaction_type VARCHAR(20),
transaction_amount INT(7),
CONSTRAINT transaction_details_tnumber_pk PRIMARY KEY (transaction_number),
CONSTRAINT transaction_details_acnumber_fk FOREIGN KEY (account_number)
REFERENCES account_master (account_number));

```

CREATE TABLE loan_details

```

(customer_number varchar(255),
branch_id varchar(255),
loan_amount bigint(20),
foreign key(customer_number) references customer_master(customer_number));

```

```

insert into customer_master values('C00001',      'RAMESH',    'CHANDRA',  'SHARMA',
'DELHI',        '9543198345', 'SERVICE'   , '1976-12-06');
insert into customer_master values('C00002',      'AVINASH',    'SUNDER',   'MINHA',
'DELHI',        '9876532109', 'SERVICE'   , '1974-10-16');
insert into customer_master values('C00003',      'RAHUL',      'NULL',     'RASTOGI',
'DELHI',        '9765178901', 'STUDENT'   , '1981-09-26');
insert into customer_master values('C00004',      'PARUL',      'NULL',     'GANDHI',
'DELHI',        '9876532109', 'HOUSEWIFE', '1976-11-03');
insert into customer_master values('C00005',      'NAVEEN',     'CHANDRA',  'AEDEKAR',
'MUMBAI',       '8976523190', 'SERVICE'   , '1976-09-19');
insert into customer_master values('C00006',      'CHITRESH',   'NULL',     'BARWE',
'MUMBAI',       '7651298321', 'STUDENT'   , '1992-11-06');
insert into customer_master values('C00007',      'AMIT',       'KUMAR',    'BORKAR',
'MUMBAI',       '9875189761', 'STUDENT'   , '1981-09-06');
insert into customer_master values('C00008',      'NISHA',      'NULL',     'DAMLE',
'MUMBAI',       '7954198761', 'SERVICE'   , '1975-12-03');
insert into customer_master values('C00009',      'ABHISHEK',   'NULL',     'DUTTA',
'KOLKATA',     '9856198761', 'SERVICE'   , '1973-05-22');
insert into customer_master values('C00010', 'SHANKAR',   'NULL',     'NAIR',    'CHENNAI',
'8765489076', 'SERVICE',   '1976-07-12');

```

```

insert into branch_master values('B00001', 'ASAF ALI ROAD','DELHI');
insert into branch_master values('B00002','NEW DELHI MAIN BRANCH','DELHI');
insert into branch_master values('B00003' , 'DELHI CANTT',     'DELHI');
insert into branch_master values('B00004' , 'JASOLA',        'DELHI');
insert into branch_master values('B00005' , 'MAHIM',         'MUMBAI');
insert into branch_master values('B00006' , 'VILE PARLE',    'MUMBAI');
insert into branch_master values('B00007' , 'MANDVI',        'MUMBAI');
insert into branch_master values('B00008' , 'JADAVPUR',      'KOLKATA');
insert into branch_master values('B00009' , 'KODAMBAKKAM',   'CHENNAI');

```

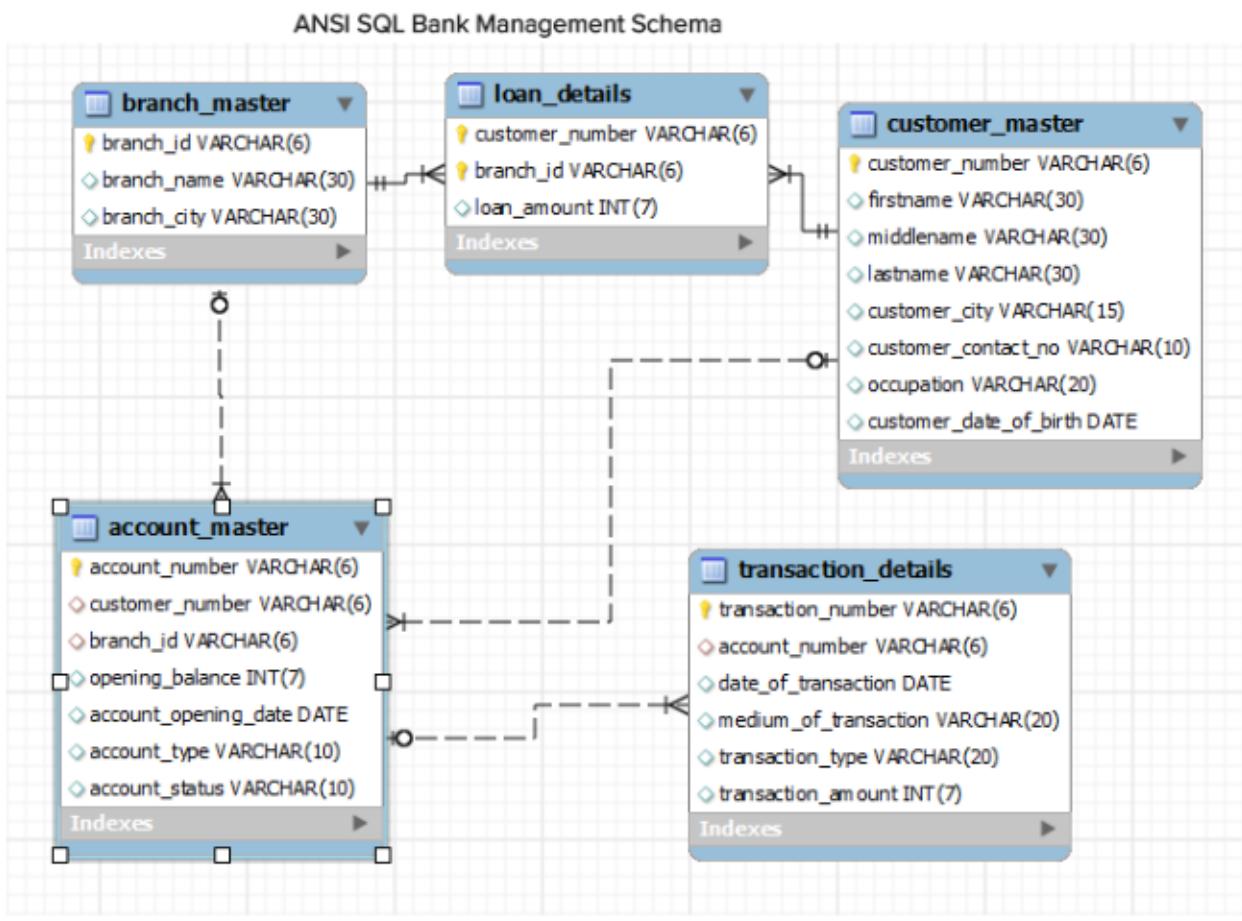
```
insert into account_master values('A00001' , 'C00001','B00001',1000      , '2012-12-15',  
      'SAVING',      'ACTIVE');  
insert into account_master values('A00002' , 'C00002','B00001',1000      , '2012-06-12'  
      , 'SAVING',      'ACTIVE');  
insert into account_master values('A00003' , 'C00003',      'B00002', 1000 , '2012-05-17'  
      , 'SAVING',      'ACTIVE');  
insert into account_master values('A00004' , 'C00002',      'B00005', 1000 , '2013-01-27'  
      , 'SAVING      , 'ACTIVE');  
insert into account_master values('A00005' , 'C00006',      'B00006', 1000 , '2012-12-17'  
      , 'SAVING',      'ACTIVE');  
insert into account_master values('A00006' , 'C00007',      'B00007', 1000 , '2010-08-12'  
      , 'SAVING      , 'SUSPENDED');  
insert into account_master values('A00007' , 'C00007',      'B00001', 1000 , '2012-10-02'  
      , 'SAVING      , 'ACTIVE');  
insert into account_master values('A00008' , 'C00001','B00003', 1000 , '2009-11-09'  
      , 'SAVING      , 'TERMINATED');  
insert into account_master values('A00009' , 'C00003',      'B00007', 1000 , '2008-11-30'  
      , 'SAVING',      'TERMINATED');  
insert into account_master values('A00010' , 'C00004',      'B00002', 1000 , '2013-03-01'  
      , 'SAVING',      'ACTIVE');  
  
insert into transaction_details values('T00001',      'A00001',      '2013-01-01', 'CHEQUE',  
      'DEPOSIT',    2000);  
insert into transaction_details values('T00002' , 'A00001' , '2013-02-01' , 'CASH'  
      , 'WITHDRAWAL', 1000);  
insert into transaction_details values('T00003',      'A00002',      '2013-01-01', 'CASH'  
      , 'DEPOSIT',   2000);  
insert into transaction_details values('T00004',      'A00002',      '2013-02-01', 'CASH'  
      , 'DEPOSIT',   3000);  
insert into transaction_details values('T00005',      'A00007',      '2013-01-11', 'CASH'  
      , 'DEPOSIT',   7000);  
insert into transaction_details values('T00006',      'A00007',      '2013-01-13', 'CASH'  
      , 'DEPOSIT',   9000);  
insert into transaction_details values('T00007',      'A00001',      '2013-03-13', 'CASH'  
      , 'DEPOSIT' , 4000);  
insert into transaction_details values('T00008',      'A00001',      '2013-03-14', 'CHEQUE'  
      , 'DEPOSIT' , 3000);  
insert into transaction_details values('T00009',      'A00001',      '2013-03-21', 'CASH'  
      , 'WITHDRAWAL' , 9000);  
insert into transaction_details values('T00010',      'A00001',      '2013-03-22', 'CASH'  
      , 'WITHDRAWAL' , 2000);
```

```

insert into transaction_details values('T00011',      'A00002',      '2013-03-25',  'CASH'
, 'WITHDRAWAL' ,7000);
insert into transaction_details values('T00012',      'A00007',      '2013-03-26',  'CASH'
, 'WITHDRAWAL' ,2000);

insert into Loan_details values('C00001',      'B00001',      100000);
insert into Loan_details values('C00002',      'B00002',      200000);
insert into Loan_details values('C00009',      'B00008',      400000);
insert into Loan_details values('C00010',      'B00009',      500000);
insert into Loan_details values('C00001',      'B00003',      600000);
insert into Loan_details values('C00002',      'B00001',      600000);

```



CUSTOMER MASTER

CUSTOMER_NUMBER	FIRSTNAME	middlename	lastname	CUSTOMER_CITY	CUSTOMER_CONTACT_NO	occupation	CUSTOMER_DATE_OF_BIRTH
C00001	RAMESH	CHANDRA	SHARMA	DELHI	9543198345	SERVICE	1976-12-06
C00002	AVINASH	SUNDER	MINHA	DELHI	9876532109	SERVICE	1974-10-16
C00003	RAHUL	NULL	RASTOGI	DELHI	9765178901	STUDENT	1981-09-26
C00004	PARUL	NULL	GANDHI	DELHI	9876532109	HOUSEWIFE	1976-11-03
C00005	NAVEEN	CHANDRA	AEDEKAR	MUMBAI	8976523190	SERVICE	1976-09-19
C00006	CHITRESH	NULL	BARWE	MUMBAI	7651298321	STUDENT	1992-11-06
C00007	AMIT	KUMAR	BORKAR	MUMBAI	9875189761	STUDENT	1981-09-06
C00008	NISHA	NULL	DAMLE	MUMBAI	7954198761	SERVICE	1975-12-03
C00009	ABHISHEK	NULL	DUTTA	KOLKATA	9856198761	SERVICE	1973-05-22
C00010	SHANKAR	NULL	NAIR	CHENNAI	8765489076	SERVICE	1976-07-12
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

ACCOUNT MASTER

account_number	customer_number	branch_id	opening_balance	account_opening_date	account_type	account_status
A00001	C00001	B00001	1000	2012-12-15	SAVING	ACTIVE
A00002	C00002	B00001	1000	2012-06-12	SAVING	ACTIVE
A00003	C00003	B00002	1000	2012-05-17	SAVING	ACTIVE
A00004	C00002	B00005	1000	2013-01-27	SAVING	ACTIVE
A00005	C00006	B00006	1000	2012-12-17	SAVING	ACTIVE
A00006	C00007	B00007	1000	2010-08-12	SAVING	SUSPENDED
A00007	C00007	B00001	1000	2012-10-02	SAVING	ACTIVE
A00008	C00001	B00003	1000	2009-11-09	SAVING	TERMINATED
A00009	C00003	B00007	1000	2008-11-30	SAVING	TERMINATED
A00010	C00004	B00002	1000	2013-03-01	SAVING	ACTIVE
NULL	NULL	NULL	NULL	NULL	NULL	NULL

BRANCH MASTER

branch_id	branch_name	branch_city
B00001	ASAF ALI ROAD	DELHI
B00002	NEW DELHI MAIN BRANCH	DELHI
B00003	DELHI CANTT	DELHI
B00004	JASOLA	DELHI
B00005	MAHIM	MUMBAI
B00006	VILE PARLE	MUMBAI
B00007	MANDVI	MUMBAI
B00008	JADAVPUR	KOLKATA
B00009	KODAMBAKKAM	CHENNAI
NULL	NULL	NULL

LOAN DETAILS

customer_number	branch_id	loan_amount
C00001	B00001	100000
C00002	B00002	200000
C00009	B00008	400000
C00010	B00009	500000
C00001	B00003	600000
C00002	B00001	600000

TRANSACTION DETAILS

transaction_number	account_number	date_of_transaction	medium_of_transaction	transaction_type	transaction_amount
T00001	A00001	2013-01-01	CHEQUE	DEPOSIT	2000
T00002	A00001	2013-02-01	CASH	WITHDRAWAL	1000
T00003	A00002	2013-01-01	CASH	DEPOSIT	2000
T00004	A00002	2013-02-01	CASH	DEPOSIT	3000
T00005	A00007	2013-01-11	CASH	DEPOSIT	7000
T00006	A00007	2013-01-13	CASH	DEPOSIT	9000
T00007	A00001	2013-03-13	CASH	DEPOSIT	4000
T00008	A00001	2013-03-14	CHEQUE	DEPOSIT	3000
T00009	A00001	2013-03-21	CASH	WITHDRAWAL	9000
T00010	A00001	2013-03-22	CASH	WITHDRAWAL	2000
T00011	A00002	2013-03-25	CASH	WITHDRAWAL	7000
T00012	A00007	2013-03-26	CASH	WITHDRAWAL	2000
NULL	NULL	NULL	NULL	NULL	NULL

QUERIES

1. Write a query to display account number, customer's number, customer's firstname, lastname, account opening date. Display the records sorted in ascending order based on account number.

```
SELECT a.account_number,c.customer_number,c.firstname,c.lastname,a.account_number
```

```

FROM      customer_master      c      JOIN      account_master      a      ON
c.customer_number=a.customer_number
ORDER BY a.account_number;

```

account_number	customer_number	firstname	lastname	account_opening_date
A00001	C00001	RAMESH	SHARMA	2012-12-15
A00002	C00002	AVINASH	MINHA	2012-06-12
A00003	C00003	RAHUL	RASTOGI	2012-05-17
A00004	C00002	AVINASH	MINHA	2013-01-27
A00005	C00006	CHITRESH	BARWE	2012-12-17
A00006	C00007	AMIT	BORKAR	2010-08-12
A00007	C00007	AMIT	BORKAR	2012-10-02
A00008	C00001	RAMESH	SHARMA	2009-11-09
A00009	C00003	RAHUL	RASTOGI	2008-11-30
A00010	C00004	PARUL	GANDHI	2013-03-01

2. Write a query to display the number of customer's from Delhi. Give the count an alias name of Cust_Count.

```

SELECT  count(customer_number)  Cust_Count  FROM  customer_master  WHERE
customer_city='Delhi';

```

cust_count
4

3. Write a query to display the customer number, customer firstname, account number for the customer's whose accounts were created after 15th of any month. Display the records sorted in ascending order based on customer number and then by account number.

```

SELECT  c.customer_number,c.firstname,a.account_number  FROM  account_master a  join
customer_master      c      ON      c.customer_number=a.customer_number      WHERE
day(a.account_opening_date)>'15' ORDER BY c.customer_number,a.account_number;

```

customer_number	firstname	account_number
C00002	AVINASH	A00004
C00003	RAHUL	A00003
C00003	RAHUL	A00009
C00006	CHITRESH	A00005

4. Write a query to display customer number, customer's first name, account number where the account status is terminated. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT c.customer_number,c.firstname,a.account_number
FROM account_master a JOIN customer_master c
ON c.customer_number=a.customer_number
WHERE a.account_status='Terminated'
ORDER BY c.customer_number,a.account_number;
```

customer_number	firstname	account_number
C00001	RAMESH	A00008
C00003	RAHUL	A00009

5. Write a query to display the total number of withdrawals and total number of deposits being done by customer whose customer number ends with 001. The query should display transaction type and the number of transactions. Give an alias name as Trans_Count for number of transactions. Display the records sorted in ascending order based on transaction type.

```
SELECT transaction_type,count(transaction_number) Trans_Count
FROM account_master am JOIN transaction_details td
ON am.account_number=td.account_number
WHERE customer_number like '%001'
GROUP BY transaction_type
ORDER BY transaction_type;
```

transaction_type	Trans_count
DEPOSIT	3
WITHDRAWAL	3

6. Write a query to display the number of customers who have registration but no account in the bank. Give the alias name as Count_Customer for number of customers.

```
SELECT count(customer_number) Count_Customer FROM customer_master
WHERE customer_number NOT IN (SELECT customer_number FROM account_master);
```

Count_customer
4

7. Write a query to display account number and total amount deposited by each account holder (Including the opening balance). Give the total amount deposited an alias name of Deposit_Amount. Display the records in sorted order based on account number.

```
SELECT a.account_number,a.opening_balance+sum(t.transaction_amount)
FROM account_master a JOIN transaction_details t ON a.account_number=t.account_number
WHERE t.transaction_type='Deposit' GROUP BY t.account_number;
```

account_number	Deposit_Amount
A00001	10000
A00002	6000
A00007	17000

8. Write a query to display the number of accounts opened in each city .The Query should display Branch City and number of accounts as No_of_Accounts.For the branch city where we don't have any accounts opened display 0. Display the records in sorted order based on branch city.

```
SELECT branch.branch_city, count(account.account_number) No_of_Accounts
FROM branch_master LEFT JOIN account_master
ON account.branch_id=branch.branch_id
GROUP BY branch.branch_city ORDER BY branch_city;
```

branch_city	No_of_accounts
CHENNAI	0
DELHI	6
KOLKATA	0
MUMBAI	4

9. Write a query to display the firstname of the customers who have more than 1 account. Display the records in sorted order based on firstname.

```
SELECT c.firstname FROM
customer_master c JOIN account_master a ON a.customer_number=c.customer_number
GROUP BY a.customer_number HAVING count(a.account_number)>1;
```

firstname
AMIT
AVINASH
RAHUL
RAMESH

10. Write a query to display the customer number, customer firstname, customer lastname who has taken loan from more than 1 branch. Display the records sorted in order based on customer number.

```
SELECT c.customer_number,c.firstname,c.lastname FROM
customer_master c JOIN loan_details l ON c.customer_number=l.customer_number
GROUP BY l.customer_number HAVING count(l.branch_id)>1
ORDER BY c.customer_number;
```

customer_number	firstname	lastname
C00001	RAMESH	SHARMA
C00002	AVINASH	MINHA

11. Write a query to display the customer's number, customer's firstname, customer's city and branch city where the city of the customer and city of the branch is different. Display the records sorted in ascending order based on customer number.

```
SELECT c.customer_number,c.firstname,c.customer_city,b.branch_city FROM
Customer_master c JOIN Account_master a ON c.customer_number=a.customer_number
JOIN Branch_master b ON b.branch_id=a.branch_id
WHERE b.branch_city<>c.customer_city
ORDER BY c.customer_number;
```

customer_number	firstname	customer_city	branch_city
C00002	AVINASH	DELHI	MUMBAI
C00003	RAHUL	DELHI	MUMBAI
C00007	AMIT	MUMBAI	DELHI

12. Write a query to display the number of clients who have asked for loans but they don't have any account in the bank though they are registered customers. Give the count an alias name of Count.

```
SELECT count(c.customer_number)Count FROM customer_master c JOIN loan_details l
ON c.customer_number=l.customer_number
WHERE c.customer_number NOT IN (SELECT customer_number FROM account_master);
```

Count

2

13. Write a query to display the account number who has done the highest transaction. For example the account A00023 has done 5 transactions i.e. suppose 3 withdrawal and 2 deposits. Whereas the account A00024 has done 3 transactions i.e. suppose 2 withdrawals and 1 deposit. So account number of A00023 should be displayed. In case of multiple records, display the records sorted in ascending order based on account number.

```
SELECT account_number FROM transaction_details
GROUP BY account_number
HAVING count(transaction_number)>=ALL
(SELECT count(transaction_number) FROM transaction_details
GROUP BY account_number) ORDER BY account_number;
```

account_number

A00001

14. Write a query to show the branch name, branch city where we have the maximum customers. For example the branch B00019 has 3 customers, B00020 has 7 and B00021 has 10. So branch id B00021 is having maximum customers. If B00021 is Koramangla branch Bangalore, Koramangla branch should be displayed along with city name

Bangalore. In case of multiple records, display the records sorted in ascending order based on branch name.

```
SELECT b.branch_name,b.branch_city FROM  
Branch_master b JOIN account a ON a.branch_id=b.branch_id  
GROUP BY a.branch_id HAVING count(a.customer_number)>=ALL  
(SELECT count(customer_number) FROM  
Account_master GROUP BY branch_id)  
ORDER BY b.branch_name;
```

branch_name	branch_city
ASAF ALI ROAD	DELHI

15. Write a query to display all those account number, deposit, withdrawal where withdrawal is more than deposit amount. Hint: Deposit should include opening balance as well. For example A00011 account opened with Opening Balance 1000 and A00011 deposited 2000 rupees on 2012-12-01 and 3000 rupees on 2012-12-02. The same account i.e A00011 withdrawn 3000 rupees on 2013-01-01 and 7000 rupees on 2013-01-03. So the total deposited amount is 6000 and total withdrawal amount is 10000. So withdrawal amount is more than deposited amount for account number A00011. Display the records sorted in ascending order based on account number.

```
SELECT td.account_number,  
sum(CASE WHEN transaction_type='Deposit' THEN transaction_amount END)  
+(SELECT opening_balance  
FROM account_master where account_number=td.account_number) Deposit,  
sum(CASE WHEN transaction_type='Withdrawal' THEN transaction_amount END) Withdrawal  
FROM transaction_details td  
GROUP BY td.account_number  
HAVING Withdrawal > Deposit  
ORDER BY td.account_number;
```

(or)

```
SELECT ifnull(t1.account_number,t2.account_number) account_number,  
t2.d Deposit,ifnull(t1.w,0) Withdrawal FROM  
(SELECT account_number,transaction_type,sum(transaction_amount) w from  
transaction_details  
WHERE transaction_type='Withdrawal' GROUP BY account_number) t1  
RIGHT JOIN  
(SELECT a.account_number,a.opening_balance+sum(t.transaction_amount) d  
FROM account_master a JOIN transaction_details t ON a.account_number=t.account_number  
WHERE t.transaction_type='Deposit' GROUP BY t.account_number) t2  
ON t1.account_number=t2.account_number  
WHERE ifnull(t1.w,0)>t2.d  
ORDER BY account_number;
```

account_number	Deposit	Withdrawal
A00001	10000	12000
A00002	6000	7000

16. Write a query to show the balance amount for account number that ends with 001.
Note: Balance amount includes account opening balance also. Give alias name as Balance_Amount. For example A00015 is having an opening balance of 1000. A00015 has deposited 2000 on 2012-06-12 and deposited 3000 on 2012-07-13. The same account has drawn money of 500 on 2012-08-12 , 500 on 2012-09-15, 1000 on 2012-12-17. So balance amount is 4000 i.e $(1000 \text{ (opening balance)} + 2000 + 3000) - (500 + 500 + 1000)$.

```
SELECT ifnull((SUM(CASE WHEN transaction_type='Deposit'
THEN transaction_amount END)) -
(SUM(CASE WHEN transaction_type='Withdrawal'
THEN transaction_amount END))+(select opening_balance
from account_master where account_number like '%001'),(SUM(CASE WHEN
transaction_type='Deposit'
THEN transaction_amount END))+(select opening_balance
from account_master where account_number like '%001')) AS Balance_Amount
FROM transaction_details where account_number like '%001';
```

(or)

```
SELECT ifnull(t1.account_number,t2.account_number) account_number,
t2.d-ifnull(t1.w,0) Balance_Amount FROM
(SELECT account_number,transaction_type,sum(transaction_amount) w from
transaction_details
WHERE transaction_type='Withdrawal' GROUP BY account_number) t1
RIGHT JOIN
(SELECT a.account_number,a.opening_balance+sum(t.transaction_amount) d
FROM account a JOIN transaction_details t ON a.account_number=t.account_number
WHERE t.transaction_type='Deposit' GROUP BY t.account_number) t2
ON t1.account_number=t2.account_number
WHERE ifnull(t1.account_number,t2.account_number) LIKE '%001'
ORDER BY account_number;
```

account_number	Balance_Amount
A00001	-2000

17. Display the customer number, customer's first name, account number and number of transactions being made by the customers from each account. Give the alias name for number of transactions as Count_Trans. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT c.customer_number,c.firstname,t.account_number, count(t.account_number)
Count_Trans
FROM transaction_details t JOIN account_master a ON a.account_number=t.account_number
JOIN customer c ON c.customer_number=a.customer_number
GROUP BY t.account_number ORDER BY c.customer_number, a.account_number;
```

customer_number	firstname	account_number	Count_Trans
C00001	RAMESH	A00001	6
C00002	AVINASH	A00002	3
C00007	AMIT	A00007	3

18. Write a query to display the customer's firstname who have multiple accounts (atleast 2 accounts). Display the records sorted in ascending order based on customer's firstname.

```
SELECT c.firstname FROM
Customer_master c JOIN account_master a ON c.customer_number=a.customer_number
GROUP BY a.customer_number HAVING count(a.account_number)>1
ORDER BY c.firstname;
```

firstname
AMIT
AVINASH
RAHUL
RAMESH

19. Write a query to display the customer number, firstname, lastname for those client where total loan amount taken is maximum and at least taken from 2 branches. For example the customer C00012 took a loan of 100000 from bank branch with id B00009 and C00012 Took a loan of 500000 from bank branch with id B00010. So total loan amount for customer C00012 is 600000. C00013 took a loan of 100000 from bank branch B00009 and 200000 from bank branch B00011. So total loan taken is 300000. So loan taken by C00012 is more then C00013.

```
SELECT Id.customer_number, firstname, lastname
FROM customer_master cm JOIN loan_details ld
ON cm.customer_number=ld.customer_number
GROUP BY customer_number
HAVING count(branch_id)>=2 AND sum(loan_amount)>=
ALL(SELECT sum(loan_amount) FROM loan GROUP BY customer_number);
```

customer_number	firstname	lastname
C00002	AVINASH	MINHA

20. Write a query to display the customer's number, customer's firstname, branch id and loan amount for people who have taken loans. Display the records sorted in ascending order based on customer number and then by branch id and then by loan amount.

```
SELECT c.customer_number,c.firstname,l.branch_id,l.loan_amount FROM
Customer_master c JOIN loan_details l ON c.customer_number=l.customer_number
ORDER BY c.customer_number,l.branch_id,l.loan_amount;
```

customer_number	firstname	branch_id	loan_amount
C00001	RAMESH	B00001	100000
C00001	RAMESH	B00003	600000
C00002	AVINASH	B00001	600000
C00002	AVINASH	B00002	200000
C00009	ABHISHEK	B00008	400000
C00010	SHANKAR	B00009	500000

21. Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count_Branch. Display the records sorted in ascending order based on city name.

```
SELECT branch_city, count(branch_id) Count_Branch FROM
Branch_master GROUP BY branch_city
ORDER BY branch_city;
```

branch_city	Count_Branch
CHENNAI	1
DELHI	4
KOLKATA	1
MUMBAI	3

22. Write a query to display account id, customer's firstname, customer's lastname for the customer's whose account is Active. Display the records sorted in ascending order based on account id /account number.

```
SELECT a.account_number, c.firstname, c.lastname FROM
Customer_master c JOIN account_master a ON c.customer_number=a.customer_number and
a.account_status='Active'
ORDER BY a.account_number;
```

account_number	firstname	lastname
A00001	RAMESH	SHARMA
A00002	AVINASH	MINHA
A00003	RAHUL	RASTOGI
A00004	AVINASH	MINHA
A00005	CHITRESH	BARWE
A00007	AMIT	BORKAR
A00010	PARUL	GANDHI

23. Write a query to display customer's number, first name and middle name. For the customers who don't have middle name, display their last name as middle name. Give the alias name as Middle_Name. Display the records sorted in ascending order based on customer number.

```
SELECT customer_number,firstname,ifnull(middlename,lastname) Middle_name FROM
Customer_master ORDER BY customer_number;
```

customer_number	firstname	Middle_name
C00001	RAMESH	CHANDRA
C00002	AVINASH	SUNDER
C00003	RAHUL	NULL
C00004	PARUL	NULL
C00005	NAVEEN	CHANDRA
C00006	CHITRESH	NULL
C00007	AMIT	KUMAR
C00008	NISHA	DAMLE
C00009	ABHISHEK	DUTTA
C00010	SHANKAR	NAIR

24. Write a query to display the customer number , firstname, customer's date of birth . Display the records sorted in ascending order of date of birth year and within that sort by firstname in ascending order.

```
SELECT customer_number,firstname,customer_date_of_birth FROM
Customer_master ORDER BY year(customer_date_of_birth),customer_number;
```

customer_number	firstname	customer_date_of_birth
C00009	ABHISHEK	1973-05-22
C00002	AVINASH	1974-10-16
C00008	NISHA	1975-12-03
C00001	RAMESH	1976-12-06
C00004	PARUL	1976-11-03
C00005	NAVEEN	1976-09-19
C00010	SHANKAR	1976-07-12
C00003	RAHUL	1981-09-26
C00007	AMIT	1981-09-06
C00006	CHITRESH	1992-11-06

25. Write a query to display the customers firstname, city and account number whose occupation are not into Business, Service or Student. Display the records sorted in ascending order based on customer first name and then by account number.

```
SELECT c.firstname,c.customer_city,a.account_number FROM
Customer_master c JOIN account_master a ON a.customer_number=c.customer_number
WHERE c.occupation NOT IN ('Service','Student','Business')
ORDER BY c.firstname,a.account_number;
```

firstname	customer_city	account_number
PARUL	DELHI	A00010

AIRLINES

```
create database flight;
use flight;
```

```
CREATE TABLE air_credit_card_details
(
profile_id VARCHAR(10)      NOT NULL,
card_number  BIGINT,
card_type VARCHAR(45),
expiration_month INT,
expiration_year INT
);
```

```
CREATE TABLE air_passenger_profile
(
profile_id VARCHAR(10) NOT NULL ,
password VARCHAR(45) NULL ,
first_name VARCHAR(45) NULL ,
last_name VARCHAR(45) NULL ,
address VARCHAR(45) NULL ,
mobile_number BIGINT NULL ,
email_id VARCHAR(45) NULL
);
```

```
CREATE TABLE air_ticket_info
(
ticket_id VARCHAR(45) NOT NULL ,
profile_id VARCHAR(10) NULL ,
flight_id VARCHAR(45) NULL ,
flight_departure_date DATE NULL ,
status VARCHAR(45) NULL
);
```

```
CREATE TABLE air_flight_details
(
flight_id VARCHAR(45) NOT NULL ,
flight_departure_date DATE NULL ,
price DECIMAL(10,2) NULL ,
available_seats INT NULL
);
```

```
CREATE TABLE air_flight
```

```

(
flight_id VARCHAR(45) NOT NULL ,
airline_id VARCHAR(45) NULL ,
airline_name VARCHAR(45) NULL ,
from_location VARCHAR(45) NULL ,
to_location VARCHAR(45) NULL ,
departure_time TIME NULL ,
arrival_time TIME NULL ,
duration TIME NULL ,
total_seats INT NULL
);

```

```
INSERT INTO air_credit_card_details VALUES
```

```
(1, 622098761234, 'debit', 5, 2013),
(2, 652362563625, 'credit', 1, 2013),
(1, 765432345678, 'credit', 2, 2013),
(3, 654378561234, 'debit', 6, 2013),
(4, 625417895623, 'debit', 2, 2013),
(5, 865478956325, 'debit', 3, 2013),
(6, 789563521457, 'credit', 4, 2013),
(2, 543267895432, 'credit', 8, 2013),
(1, 256369856321, 'debit', 1, 2013);
```

```
INSERT INTO air_flight VALUES
```

```
(3173, 'MH370', 'abc', 'hyderabad', 'chennai', '06:30:00', '07:15:00',
'0:45:00', 100),
(3178, 'MH17', 'def', 'chennai', 'hyderabad', '08:00:00', '09:00:00',
'1:00:00', 200),
(3172, 'AR342', 'fgh', 'kolkata', 'chennai', '11:30:00', '13:00:00',
'1:30:00', 100),
(3071, 'JT564', 'jkl', 'chennai', 'delhi', '08:00:00', '10:00:00', '2:00:00',
100),
(3170, 'DT345', 'xyz', 'delhi', 'kolkata', '21:00:00', '22:30:00', '1:30:00',
100),
(3175, 'MJ654', 'abc', 'chennai', 'hyderabad', '15:00:00', '16:00:00',
'1:00:00', 200),
(3176, 'MH370', 'def', 'kochi', 'chennai', '18:00:00', '19:05:00', '1:05:00',
100),
(3177, 'MH45', 'fgh', 'delhi', 'kochi', '19:00:00', '21:00:00', '2:00:00',
200),
(3174, 'MH321', 'xyz', 'kolkata', 'delhi', '0:00:00', '2:00:00', '2:00:00',
100),
```

```
(3179, 'JT435',      'abc',   'chennai',     'kolkata',      '14:00:00',      '15:00:00',
      '1:00:00',        100),
(3180, 'JT456',      'ijk',    'kolkata',     'kochi',       '5:00:00',      '5:45:00',
      200);
```

```
INSERT INTO air_flight_details VALUES
```

```
(3170, '2013-02-14', 1000, 10),
(3171, '2013-03-15', 5000, 0),
(3172, '2013-02-05', 3000, 32),
(3173, '2013-04-07', 2000, 12),
(3174, '2013-04-05', 3800, 3),
(3175, '2013-05-25', 3500, 10),
(3176, '2013-03-14', 8000, 2),
(3177, '2013-06-15', 1500, 0),
(3178, '2013-05-06', 3000, 5),
(3179, '2013-04-03', 4000, 15),
(3180, '2013-04-02', 3000, 14);
```

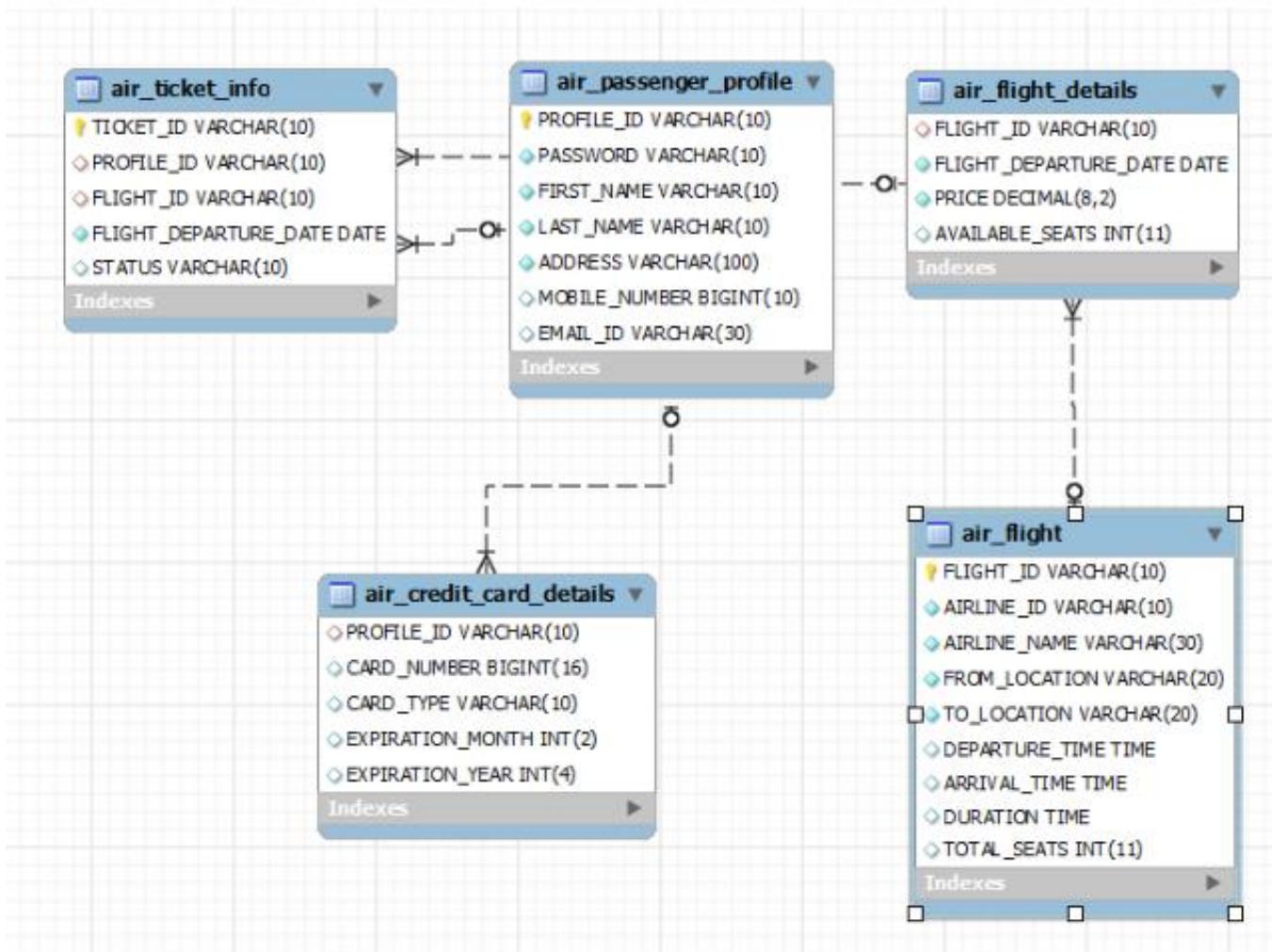
```
INSERT INTO air_ticket_info VALUES
```

```
(1, 1, 3178, '2013-05-06', 'delayed'),
(2, 5, 3179, '2013-04-03', 'on time'),
(2, 4, 3180, '2013-04-02', 'on time'),
(1, 2, 3177, '2013-06-15', 'on time'),
(1, 3, 3176, '2013-03-14', 'on time'),
(3, 1, 3171, '2013-03-15', 'on time'),
(4, 4, 3172, '2013-02-06', 'delayed'),
(5, 2, 3178, '2013-06-05', 'on time'),
(4, 3, 3171, '2013-03-15', 'on time'),
(5, 1, 3175, '2013-05-25', 'on time'),
(6, 3, 3177, '2013-06-15', 'on time');
```

```
INSERT INTO air_passenger_profile VALUES
```

```
(1, 'godbless', 'John', 'Stuart', 'Street 21, Near Bus Stop-Hyderabad-432126', 9865263251, 'john@gmail.com'),
(2, 'heyyaa', 'Robert', 'Clive', 'Sector 3, Technopolis-Kolkata-700102', 9733015875, 'robert@yahoo.com'),
(3, 'hello123', 'Raj', 'Sharma', 'House No. 3, Anna Nagar-Kochi-452314', 9775470232, 'raj3452@hotmail.com'),
(4, 'yesboss', 'Sanjay', 'Mittal', '21 Cauunaught Place-Delhi-144985', 9856856321, 'sanjay@yahoo.com'),
```

(5, 'imhere', 'Tony', 'Stark', '51A, Greams Lane-Chennai-144587', 9832015785, 'tony@gmail.com');



AIR TICKET INFO

ticket_id	profile_id	flight_id	flight_departure_date	status
1	1	3178	2013-05-06	delayed
2	5	3179	2013-04-03	on time
2	4	3180	2013-04-02	on time
1	2	3177	2013-06-15	on time
1	3	3176	2013-03-14	on time
3	1	3171	2013-03-15	on time
4	4	3172	2013-02-06	delayed
5	2	3178	2013-06-05	on time
4	3	3171	2013-03-15	on time
5	1	3175	2013-05-25	on time
6	3	3177	2013-06-15	on time

AIR PASSENGER DETAILS

profile_id	password	first_name	last_name	address	mobile_number	email_id
1	godbless	John	Stuart	Street 21, Near Bus Stop-Hyderabad-432126	9865263251	john@gmail.com
2	heyyaa	Robert	Clive	Sector 3, Technopolis-Kolkata-700102	9733015875	robert@yahoo.com
3	hello123	Raj	Sharma	House No. 3, Anna Nagar-Kochi-452314	9775470232	raj3452@hotmail...
4	yesboss	Sanjay	Mittal	21 Cauunaught Place-Delhi-144985	9856856321	sanjay@yahoo.c...
5	imhere	Tony	Stark	51A, Greams Lane-Chennai-144587	9832015785	tony@gmail.com

AIR FLIGHT DETAILS

flight_id	flight_departure_date	price	available_seats
3170	2013-02-14	1000.00	10
3171	2013-03-15	5000.00	0
3172	2013-02-05	3000.00	32
3173	2013-04-07	2000.00	12
3174	2013-04-05	3800.00	3
3175	2013-05-25	3500.00	10
3176	2013-03-14	8000.00	2
3177	2013-06-15	1500.00	0
3178	2013-05-06	3000.00	5
3179	2013-04-03	4000.00	15
3180	2013-04-02	3000.00	14

AIR CREDIT CARD DETAILS

profile_id	card_number	card_type	expiration_month	expiration_year
1	622098761234	debit	5	2013
2	652362563625	credit	1	2013
1	765432345678	credit	2	2013
3	654378561234	debit	6	2013
4	625417895623	debit	2	2013
5	865478956325	debit	3	2013
6	789563521457	credit	4	2013
2	543267895432	credit	8	2013
1	256369856321	debit	1	2013

AIR FLIGHT

flight_id	airline_id	airline_name	from_location	to_location	departure_time	arrival_time	duration	total_seats
3170	DT345	xyz	delhi	kolkata	21:00:00	22:30:00	01:30:00	100
3171	JT564	JKL	chennai	delhi	08:00:00	10:00:00	02:00:00	100
3172	AR342	fgh	kolkata	chennai	11:30:00	13:00:00	01:30:00	100
3173	MH370	abc	hyderabad	chennai	06:30:00	07:15:00	00:45:00	100
3174	MH321	xyz	kolkata	delhi	00:00:00	02:00:00	02:00:00	100
3175	MJ654	abc	chennai	hyderabad	15:00:00	16:00:00	01:00:00	200
3176	MH370	def	kochi	chennai	18:00:00	19:05:00	01:05:00	100
3177	MH45	fgh	delhi	kochi	19:00:00	21:00:00	02:00:00	200
3178	MH17	def	chennai	hyderabad	08:00:00	09:00:00	01:00:00	200
3179	JT435	abc	chennai	kolkata	14:00:00	15:00:00	01:00:00	100
3180	JT456	ijk	kolkata	kochi	05:00:00	05:45:00	00:45:00	200

QUERIES

1. Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight_Id,From_Location,To_Location,Month Name as “Month_Name” and average price as “Average_Price”. Display the records sorted in ascending order based on flight id and then by Month Name.

```
SELECT f.flight_id,f.from_location,f.to_location,
monthname(af.flight_departure_date) Month_Name,
AVG(price) Average_Price FROM air_flight f JOIN air_flight_details af
ON f.flight_id = af.flight_id WHERE f.airline_name = 'abc'
GROUP BY f.flight_id,f.from_location,f.to_location,Month_Name
ORDER BY f.flight_id, Month_Name;
```

flight_id	from_location	to_location	Month_Name	Average_Price
3173	hyderabad	chennai	April	2000.000000
3175	chennai	hyderabad	May	3500.000000
3179	chennai	kolkata	April	4000.000000

2. Write a query to display the number of flight services between locations in a month. The Query should display From_Location, To_Location, Month as “Month_Name” and number of flight services as “No_of_Services”. Hint: The Number of Services can be calculated from the number of scheduled departure dates of a flight. The records should be displayed in ascending order based on From_Location and then by To_Location and then by month name.

```
SELECT f.from_location,f.to_location,
monthname(af.flight_departure_date) Month_Name,
count(af.flight_departure_date) No_of_Services
FROM air_flight f JOIN air_flight_details af
ON f.flight_id = af.flight_id
GROUP BY f.from_location,f.to_location,Month_Name
ORDER BY f.from_location,f.to_Location,Month_Name;
```

from_location	to_location	Month_Name	No_of_Services
chennai	delhi	March	1
chennai	hyderabad	May	2
chennai	kolkata	April	1
delhi	kochi	June	1
delhi	kolkata	February	1
hyderabad	chennai	April	1
kochi	chennai	March	1
kolkata	chennai	February	1
kolkata	delhi	April	1
kolkata	kochi	April	1

3. Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile_id, customer's first_name, Address and Number of tickets booked as “No_of_Tickets” Display the records sorted in ascending order based on customer's first name.

```
SELECT ap.profile_id,ap.first_name,ap.address,count(ati.ticket_id) No_of_Tickets
FROM air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id and af.airline_name='abc'
GROUP BY ap.profile_id,ap.first_name,ap.address HAVING count(ati.ticket_id)<=ALL
(SELECT count(ticket_id)
FROM air_ticket_info GROUP BY profile_id)
```

ORDER BY ap.first_name;

profile_id	first_name	address	No_of_Tickets
1	John	Street 21, Near Bus Stop-Hyderabad-432126	1
5	Tony	51A, Greams Lane-Chennai-144587	1

4. Write a query to display the number of tickets booked from Chennai to Hyderabad. The Query should display passenger profile_id,first_name,last_name, Flight_Id , Departure_Date and number of tickets booked as “No_of_Tickets”.Display the records sorted in ascending order based on profile id and then by flight id and then by departure date.

```
SELECT ap.profile_id,ap.first_name,ap.last_name,af.flight_id,ati.flight_departure_date,  
count(ati.profile_id) No_of_Tickets FROM  
air_ticket_info ati JOIN air_passenger_profile ap ON ap.profile_id=ati.profile_id  
JOIN air_flight af ON af.flight_id=ati.flight_id  
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'  
GROUP BY ati.flight_id,ati.profile_id  
ORDER BY ap.profile_id,af.flight_id,ati.flight_departure_date;
```

profile_id	first_name	last_name	flight_id	flight_departure_date	No_of_Tickets
1	John	Stuart	3175	2013-05-25	1
1	John	Stuart	3178	2013-05-06	1
2	Robert	Clive	3178	2013-06-05	1

5. Write a query to display flight id,from location, to location and ticket price of flights whose departure is in the month of april.Display the records sorted in ascending order based on flight id and then by from location.

```
SELECT af.flight_id,af.from_location,af.to_location,afd.price FROM  
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id  
and month(afd.flight_departure_date)='04'  
ORDER BY af.flight_id,af.from_location;
```

flight_id	from_location	to_location	price
3173	hyderabad	chennai	2000.00
3174	kolkata	delhi	3800.00
3179	chennai	kolkata	4000.00
3180	kolkata	kochi	3000.00

6. Write a query to display the average cost of the tickets in each flight on all scheduled dates. The query should display flight_id, from_location, to_location and Average price as “Price”. Display the records sorted in ascending order based on flight id and then by from_location and then by to_location.

```
SELECT af.flight_id,af.from_location,af.to_location,avg(afd.price) Average_Price FROM  
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
```

```

GROUP BY af.flight_id
ORDER BY af.flight_id,af.from_location,af.to_location;

```

flight_id	from_location	to_location	Average_Price
3170	delhi	kolkata	1000.000000
3171	chennai	delhi	5000.000000
3172	kolkata	chennai	3000.000000
3173	hyderabad	chennai	2000.000000
3174	kolkata	delhi	3800.000000
3175	chennai	hyderabad	3500.000000
3176	kochi	chennai	8000.000000
3177	delhi	kochi	1500.000000
3178	chennai	hyderabad	3000.000000
3179	chennai	kolkata	4000.000000
3180	kolkata	kochi	3000.000000

7. Write a query to display the customers who have booked tickets from Chennai to Hyderabad. The query should display profile_id, customer_name (combine first_name & last_name with comma in b/w), address of the customer. Give an alias to the name as customer_name. Hint: Query should fetch unique customers irrespective of multiple tickets booked. Display the records sorted in ascending order based on profile id.

```

SELECT ap.profile_id,concat(ap.first_name,',',ap.last_name) customer_name,ap.address FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
GROUP BY ati.profile_id
ORDER BY ap.profile_id;

```

profile_id	Customer_name	address
1	John,Stuart	Street 21, Near Bus Stop-Hyderabad-432126
2	Robert,Clive	Sector 3, Technopolis-Kolkata-700102

8. Write a query to display profile id of the passenger(s) who has/have booked maximum number of tickets. In case of multiple records, display the records sorted in ascending order based on profile id.

```

SELECT profile_id FROM air_ticket_info
group by profile_id
having count(ticket_id)>=all(select count(ticket_id)
from air_ticket_info
group by profile_id) order by profile_id;

```

profile_id
1
3

9. Write a query to display the total number of tickets as “No_of_Tickets” booked in each flight in ABC Airlines. The Query should display the flight_id, from_location, to_location and the number of tickets. Display only the flights in which atleast 1 ticket is booked. Display the records sorted in ascending order based on flight id.

```
SELECT f.flight_id,f.from_location,f.to_location,COUNT(t.ticket_id) AS No_of_Tickets
FROM air_ticket_info t JOIN air_flight f
ON f.flight_id = t.flight_id where AIRLINE_NAME = 'abc' GROUP by
f.flight_id,f.from_location,f.to_location
having count(t.ticket_id)>=1
ORDER by f.flight_id;
```

flight_id	from_location	to_location	No_of_Tickets
3175	chennai	hyderabad	1
3179	chennai	kolkata	1

10. Write a query to display the no of services offered by each flight and the total price of the services. The Query should display flight_id, number of services as “No_of_Services” and the cost as “Total_Price” in the same order. Order the result by Total Price in descending order and then by flight_id in descending order. Hint: The number of services can be calculated from the number of scheduled departure dates of the flight

```
SELECT flight_id,count(flight_departure_date) No_of_services,sum(price) Total_Price FROM
air_flight_details GROUP BY flight_id
ORDER BY Total_price DESC,flight_id DESC;
```

flight_id	No_of_services	Total_Price
3176	1	8000.00
3171	1	5000.00
3179	1	4000.00
3174	1	3800.00
3175	1	3500.00
3180	1	3000.00
3178	1	3000.00
3172	1	3000.00
3173	1	2000.00
3177	1	1500.00
3170	1	1000.00

11. Write a query to display the number of passengers who have travelled in each flight in each scheduled date. The Query should display flight_id, flight_departure_date and the number of passengers as “No_of_Passengers” in the same order. Display the records sorted in ascending order based on flight id and then by flight departure date.

```
SELECT flight_id,flight_departure_date,count(ticket_id) No_of_passengers FROM air_ticket_info GROUP BY flight_id,flight_departure_date ORDER BY flight_id,flight_departure_date;
```

flight_id	flight_departure_date	No_of_passengers
3171	2013-03-15	2
3172	2013-02-06	1
3175	2013-05-25	1
3176	2013-03-14	1
3177	2013-06-15	2
3178	2013-05-06	1
3178	2013-06-05	1
3179	2013-04-03	1
3180	2013-04-02	1

12. Write a query to display profile id of passenger(s) who booked minimum number of tickets. In case of multiple records, display the records sorted in ascending order based on profile id.

```
SELECT profile_id FROM air_ticket_info
GROUP BY profile_id HAVING count(ticket_id)<=ALL
(SELECT count(ticket_id) FROM air_ticket_info GROUP BY profile_id)
ORDER BY profile_id;
```

profile_id
5

13. Write a query to display unique passenger profile id, first name, mobile number and email address of passengers who booked ticket to travel from HYDERABAD to CHENNAI. Display the records sorted in ascending order based on profile id.

```
SELECT DISTINCT ap.profile_id,ap.first_name,ap.mobile_number,ap.email_id FROM air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON ati.flight_id=af.flight_id
WHERE af.from_location='Hyderabad' and af.to_location='Chennai'
ORDER BY profile_id;
```

profile_id	first_name	mobile_number	email_id

14. Write a query to intimate the passengers who are boarding Chennai to Hyderabad Flight on 6th May 2013 stating the delay of 1hr in the departure time. The Query should display the passenger's profile_id, first_name, last_name, flight_id, flight_departure_date, actual departure time , actual arrival time , delayed departure time as "Delayed_Departure_Time", delayed arrival time as "Delayed_Arrival_Time" Hint: Distinct Profile ID should be

displayed irrespective of multiple tickets booked by the same profile. Display the records sorted in ascending order based on passenger's profile id.

```

SELECT DISTINCT
ap.profile_id,ap.first_name,ap.last_name,ati.flight_id,ati.flight_departure_date,
af.departure_time,af.arrival_time,
addtime(af.departure_time,'01:00:00') Delayed_Deparuture_Time,
addtime(af.arrival_time,'01:00:00') Delayed_Arrival_Time FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
and ati.flight_departure_date='2013-05-06'
ORDER BY profile_id;

```

profile_id	first_name	last_name	flight_id	flight_departure_date	departure_time	arrival_time	Delayed_Deparuture_Time	Delayed_Arrival_Time
1	John	Stuart	3178	2013-05-06	08:00:00	09:00:00	09:00:00	10:00:00

15. Write a query to display the number of tickets as “No_of_Tickets” booked by Kochi Customers. The Query should display the Profile_Id, First_Name, Base_Location and number of tickets booked. Hint: Use String functions to get the base location of customer from their Address and give alias name as “Base_Location” Display the records sorted in ascending order based on customer first name.

```

SELECT ap.profile_id,ap.first_name,
substring_index(substring_index(ap.address,'-',2),'-',1) Base_Location,
count(atি.ticket_id) No_of_Tickets FROM
air_passenger_profile ap JOIN air_ticket_info atি ON atি.profile_id=ap.profile_id
WHERE ap.address LIKE '%Kochi%'
ORDER BY ap.first_name;

```

profile_id	first_name	Base_Location	No_of_Tickets
3	Raj	Kochi	3

16. Write a query to display the flight_id, from_location, to_location, number of Services as “No_of_Services” offered in the month of May.

```

SELECT af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date)
No_of_services FROM
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
WHERE month(flight_departure_date)=05'
GROUP BY af.flight_id,af.from_location,af.to_location
ORDER BY af.flight_id;

```

flight_id	from_location	to_location	No_of_services
3175	chennai	hyderabad	1
3178	chennai	hyderabad	1

17. Write a query to display profile id, last name, mobile number and email id of passengers whose base location is chennai. Display the records sorted in ascending order based on profile id.

```
SELECT profile_id, last_name, mobile_number, email_id
FROM air_passenger_profile
WHERE address LIKE '%Chennai%'
ORDER BY profile_id;
```

profile_id	last_name	mobile_number	email_id
5	Stark	9832015785	tony@gmail.com

18. Write a query to display number of flights between 6.00 AM and 6.00 PM from chennai. Hint Use FLIGHT_COUNT as alias name.

```
SELECT count(flight_id) FLIGHT_COUNT FROM air_flight
WHERE from_location='CHENNAI'
AND departure_time BETWEEN '06:00:00' AND '18:00:00';
```

FLIGHT_COUNT
4

19. Write a query to display unique profile id, first name , email id and contact number of passenger(s) who travelled on flight with id 3178. Display the records sorted in ascending order based on first name.

```
SELECT DISTINCT ap.profile_id,ap.first_name,ap.email_id,ap.mobile_number FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
WHERE ati.flight_id='3178'
ORDER BY ap.first_name;
```

profile_id	first_name	email_id	mobile_number
1	John	john@gmail.com	9865263251
2	Robert	robert@yahoo.com	9733015875

20. Write a query to display flight id, departure date, flight type of all flights. Flight type can be identified based on the following rules : if ticket price is less than 3000 then 'AIR PASSENGER', ticket price between 3000 and less than 4000 'AIR BUS' and ticket price between 4000 and greater than 4000 then 'EXECUTIVE PASSENGER'. Hint use FLIGHT_TYPE as alias name. Display the records sorted in ascending order based on flight_id and then by departure date.

```
SELECT flight_id, flight_departure_date,
CASE WHEN price < 3000 THEN 'AIR PASSENGER'
WHEN price >= 3000 AND price < 4000 THEN 'AIR BUS'
WHEN price >= 4000 THEN 'EXECUTIVE PASSENGER'
```

end FLIGHT_TYPE FROM air_flight_details
 ORDER BY flight_id,flight_departure_date;

flight_id	flight_departure_date	FLIGHT_TYPE
3170	2013-02-14	AIR PASSENGER
3171	2013-03-15	EXECUTIVE PASSENGER
3172	2013-02-05	AIR BUS
3173	2013-04-07	AIR PASSENGER
3174	2013-04-05	AIR BUS
3175	2013-05-25	AIR BUS
3176	2013-03-14	EXECUTIVE PASSENGER
3177	2013-06-15	AIR PASSENGER
3178	2013-05-06	AIR BUS
3179	2013-04-03	EXECUTIVE PASSENGER
3180	2013-04-02	AIR BUS

21. Write a query to display the credit card type and no of credit cards used on the same type. Display the records sorted in ascending order based on credit card type. Hint: Use CARD_COUNT AS Alias name for no of cards.

SELECT card_type, count(card_type) Card_Count FROM air_credit_card_details
 GROUP BY card_type ORDER BY card_type;

card_type	Card_Count
credit	4
debit	5

22. Write a Query to display serial no, first name, mobile number, email id of all the passengers who holds email address from gmail.com. The Serial No will be the last three digits of profile ID. Hint: Use SERIAL_NO as Alias name for serial number. Display the records sorted in ascending order based on name.

SELECT substring(profile_id,-3) SERIAL_NO,first_name,mobile_number,email_id FROM air_passenger_profile
 WHERE email_id LIKE '%@gmail.com'
 ORDER BY first_name;

SERIAL_NO	first_name	mobile_number	email_id
	John	9865263251	john@gmail.com
	Tony	9832015785	tony@gmail.com

23. Write a query to display the flight(s) which has least number of services in the month of May. The Query should fetch flight_id, from_location, to_location, least number of Services as "No_of_Services" Hint: Number of services offered can be calculated from the number of scheduled departure dates of a flight if there are multiple flights, display them sorted in ascending order based on flight id.

```

SELECT afd.flight_id,af.from_location,af.to_location,count(afd.flight_id) No_of_Services
FROM air_flight_details afd JOIN air_flight af ON af.flight_id=afd.flight_id
WHERE monthname(afd.flight_departure_date)='May'
GROUP BY afd.flight_departure_date HAVING count(afd.flight_id) <=
ALL(SELECT count(flight_id) FROM air_flight_details
WHERE monthname(flight_departure_date)='May'
GROUP BY flight_departure_date)
ORDER BY flight_id;

```

flight_id	from_location	to_location	No_of_Services
3175	chennai	hyderabad	1
3178	chennai	hyderabad	1

24. Write a query to display the flights available in Morning, AfterNoon, Evening& Night. The Query should display the Flight_Id, From_Location, To_Location , Departure_Time, time of service as "Time_of_Service". Time of Service should be calculated as: From 05:00:01 Hrs to 12:00:00 Hrs - Morning, 12:00:01 to 18:00:00 Hrs -AfterNoon, 18:00:01 to 24:00:00 - Evening and 00:00:01 to 05:00:00 - NightDisplay the records sorted in ascending order based on flight id.

```

SELECT flight_id,from_location,to_location,Departure_Time,
CASE
WHEN departure_time BETWEEN ('05:00:01') AND ('12:00:00')
THEN 'Morning'
WHEN departure_time BETWEEN ('12:00:01') AND ('18:00:00')
THEN 'AfterNoon'
WHEN departure_time BETWEEN ('18:00:01') AND ('24:00:00')
THEN 'Evening'
WHEN departure_time='00:00:00'
THEN 'Evening'
WHEN departure_time BETWEEN ('00:00:01') AND ('05:00:00')
THEN 'Night'
END Time_of_Service
FROM air_flight
order by flight_id;

```

flight_id	from_location	to_location	Departure_Time	Time_of_Service
3170	delhi	kolkata	21:00:00	Evening
3171	chennai	delhi	08:00:00	Morning
3172	kolkata	chennai	11:30:00	Morning
3173	hyderabad	chennai	06:30:00	Morning
3174	kolkata	delhi	00:00:00	Evening
3175	chennai	hyderabad	15:00:00	AfterNoon
3176	kochi	chennai	18:00:00	AfterNoon
3177	delhi	kochi	19:00:00	Evening
3178	chennai	hyderabad	08:00:00	Morning
3179	chennai	kolkata	14:00:00	AfterNoon
3180	kolkata	kochi	05:00:00	Night

25. Write a query to display the number of flights flying from each location. The Query should display the from location and the number of flights to other locations as “No_of_Flights”. Hint: Get the distinct from location and to location. Display the records sorted in ascending order based on from location.

```
SELECT from_location, count(flight_id) No_of_Flights FROM
air_flight GROUP BY from_location
ORDER BY from_location;
```

from_location	No_of_Flights
chennai	4
delhi	2
hyderabad	1
kochi	1
kolkata	3

26. Write a query to display the number of passengers traveled in each flight in each scheduled date. The Query should display flight_id, from_location, To_location, flight_departure_date and the number of passengers as “No_of_Passengers”. Hint: The Number of passengers inclusive of all the tickets booked with single profile id. Display the records sorted in ascending order based on flight id and then by flight departure date.

```
SELECT af.flight_id, af.from_location, af.to_location, ati.flight_departure_date,
count(ati.ticket_id) No_of_Passengers FROM
air_flight af JOIN air_ticket_info ati ON af.flight_id=ati.flight_id
GROUP BY af.flight_id, af.from_location, af.to_location, ati.flight_departure_date
ORDER BY af.flight_id, ati.flight_departure_date;
```

flight_id	from_location	to_location	flight_departure_date	No_of_Passengers
3171	chennai	delhi	2013-03-15	2
3172	kolkata	chennai	2013-02-06	1
3175	chennai	hyderabad	2013-05-25	1
3176	kochi	chennai	2013-03-14	1
3177	delhi	kochi	2013-06-15	2
3178	chennai	hyderabad	2013-05-06	1
3178	chennai	hyderabad	2013-06-05	1
3179	chennai	kolkata	2013-04-03	1
3180	kolkata	kochi	2013-04-02	1

27. Write a query to display the flight details in which more than 10% of seats have been booked. The query should display Flight_Id, From_Location, To_Location, Total_Seats, seats booked as “No_of_Seats_Booked” .Display the records sorted in ascending order based on flight id and then by No_of_Seats_Booked.

```
SELECT af.flight_id,af.from_location,af.to_location,af.total_seats,
(af.total_seats-afd.available_seats) No_of_Seats_Booked FROM
air_flight_details afd JOIN air_flight af ON afd.flight_id=af.flight_id
WHERE (af.total_seats-afd.available_seats)>(af.total_seats*0.1)
ORDER BY flight_id,No_of_Seats_Booked;
```

flight_id	from_location	to_location	total_seats	No_of_Seats_Booked
3170	delhi	kolkata	100	90
3171	chennai	delhi	100	100
3172	kolkata	chennai	100	68
3173	hyderabad	chennai	100	88
3174	kolkata	delhi	100	97
3175	chennai	hyderabad	200	190
3176	kochi	chennai	100	98
3177	delhi	kochi	200	200
3178	chennai	hyderabad	200	195
3179	chennai	kolkata	100	85
3180	kolkata	kochi	200	186

28. Write a query to display the Flight_Id, Flight_Departure_Date, From_Location, To_Location and Duration of all flights which has duration of travel less than 1 Hour, 10 Minutes.

```
SELECT af.flight_id,afd.flight_Departure_Date,af.From_Location,af.To_Location,af.duration
FROM air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
WHERE af.duration<'01:10:00';
```

flight_Id	flight_Departure_Date	From_Location	To_Location	duration
3173	2013-04-07	hyderabad	chennai	00:45:00
3175	2013-05-25	chennai	hyderabad	01:00:00
3176	2013-03-14	kochi	chennai	01:05:00
3178	2013-05-06	chennai	hyderabad	01:00:00
3179	2013-04-03	chennai	kolkata	01:00:00
3180	2013-04-02	kolkata	kochi	00:45:00

29. Write a query to display the flight_id, from_location,to_location,number of services as “No_of_Services” , average ticket price as “Average_Price” whose average ticket price is greater than the total average ticket cost of all flights. Order the result by lowest average price.

```
SELECT afd.flight_id,af.from_location,af.to_location,
count(afd.flight_departure_date) No_of_Service, avg(price) Average_Price
FROM air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
GROUP BY af.flight_id,af.from_location,af.to_location
HAVING avg(price)>(SELECT avg(price) FROM air_flight_details)
ORDER BY average_price;
```

flight_id	from_location	to_location	No_of_Service	Average_Price
3175	chennai	hyderabad	1	3500.000000
3174	kolkata	delhi	1	3800.000000
3179	chennai	kolkata	1	4000.000000
3171	chennai	delhi	1	5000.000000
3176	kochi	chennai	1	8000.000000

MOVIE

```
CREATE DATABASE video;USE video;
Create table CUSTOMER_MASTER
(CUSTOMER_ID Varchar(10),CUSTOMER_NAME Varchar(30) NOT
NULL,CONTACT_NO BIGINT(10),CONTACT_ADD
```

Varchar(20),DATE_OF_REGISTRATION Date NOT NULL,AGE Varchar(15)NOT
 NULL,Constraint MT_cts1 PRIMARY KEY(CUSTOMER_ID));
 Create table LIBRARY_CARD_MASTER
 (CARD_ID Varchar(10),DESCRIPTION Varchar(30) NOT NULL,AMOUNT
 BIGINT(50),NUMBER_OF_YEARS bigint(10) NOT NULL,Constraint MT_cts2
 PRIMARY KEY(CARD_ID));
 Create table MOVIES_MASTER
 (MOVIE_ID Varchar(10), MOVIE_NAME Varchar(50) NOT NULL,RELEASE_DATE
 Varchar(30) NOT NULL,LANGUAGE Varchar(30),RATING int(2),DURATION
 VARCHAR(10) NOT NULL, MOVIE_TYPE Varchar(3),MOVIE_CATEGORY
 VARCHAR(20) NOT NULL,DIRECTOR VARCHAR(20) NOT NULL,
 LEAD_ROLE_1 Varchar(3) NOT NULL,LEAD_ROLE_2 VARCHAR(4) NOT
 NULL,RENT_COST BIGINT(10),Constraint MT_cts4 PRIMARY KEY(MOVIE_ID));
 Create table CUSTOMER_CARD_DETAILS
 (CUSTOMER_ID Varchar(10),CARD_ID VARCHAR(10),ISSUE_DATE DATE NOT
 NULL,Constraint MT_cts3 PRIMARY KEY(CUSTOMER_ID),Constraint MT_CTS41
 FOREIGN KEY(CUSTOMER_ID) References
 CUSTOMER_MASTER(CUSTOMER_ID),Constraint MT_CTS42 FOREIGN
 KEY(CARD_ID) References LIBRARY_CARD_MASTER(CARD_ID));
 Create table CUSTOMER_ISSUE_DETAILS
 (ISSUE_ID Varchar(10) NOT NULL,CUSTOMER_ID Varchar(10) NOT
 NULL,MOVIE_ID VARCHAR(10),ISSUE_DATE Date NOT NULL,RETURN_DATE Date
 NOT NULL,
 ACTUAL_DATE_RETURN Date NOT NULL,Constraint MT_cts5 PRIMARY
 KEY(ISSUE_ID),Constraint MT_Mem FOREIGN KEY(CUSTOMER_ID) References
 CUSTOMER_MASTER(CUSTOMER_ID), Constraint MT_Mem1 FOREIGN
 KEY(MOVIE_ID) References MOVIES_MASTER(MOVIE_ID));

Insert into CUSTOMER_MASTER Values('CUS001', 'AMIT', 9876543210,'ADD1',
 '2012-02-12', '21');
 Insert into CUSTOMER_MASTER Values('CUS002', 'ABDHUL', 8765432109,'ADD2',
 '2012-02-12', '21');
 Insert into CUSTOMER_MASTER Values('CUS003', 'GAYAN', 7654321098,'ADD3',
 '2012-02-12', '21');
 Insert into CUSTOMER_MASTER Values('CUS004', 'RADHA', 6543210987,'ADD4',
 '2012-02-12', '21');
 Insert into CUSTOMER_MASTER Values('CUS005', 'GURU', NULL,'ADD5', '2012-02-
 12', '21');
 Insert into CUSTOMER_MASTER Values('CUS006', 'MOHAN', 4321098765,'ADD6',
 '2012-02-12', '21');

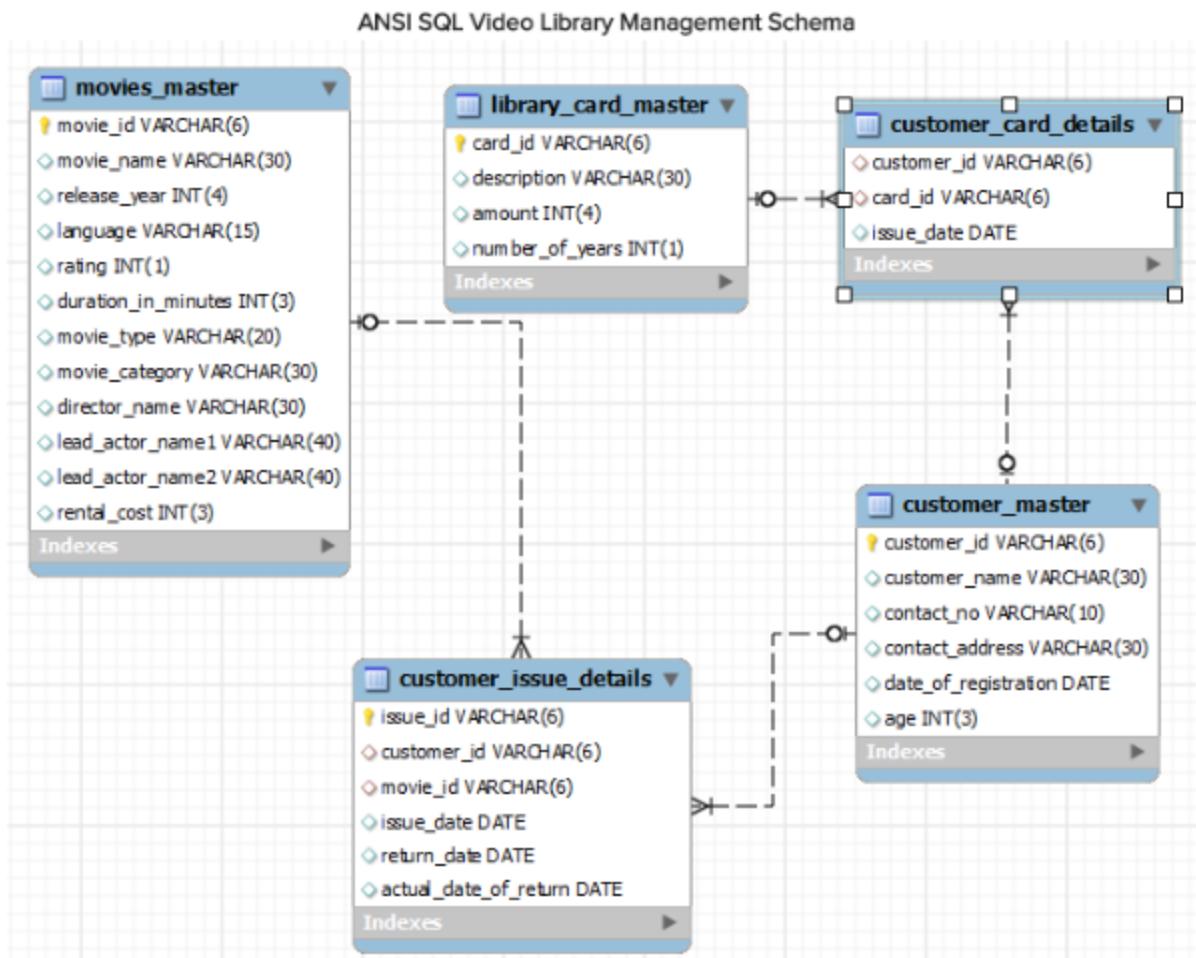
```
Insert into CUSTOMER_MASTER Values('CUS007', 'NAME7', 3210987654,'ADD7',  
'2012-02-12', '21');  
Insert into CUSTOMER_MASTER Values('CUS008', 'NAME8', 2109876543,'ADD8',  
'2013-02-12', '21');  
Insert into CUSTOMER_MASTER Values('CUS009', 'NAME9', NULL,'ADD9', '2013-02-  
12', '21');  
Insert into CUSTOMER_MASTER Values('CUS010', 'NAM10', 9934567890,'ADD10',  
'2013-02-12', '21');  
Insert into CUSTOMER_MASTER Values('CUS011', 'NAM11', 9875678910,'ADD11',  
'2013-02-12', '21');  
  
Insert into LIBRARY_CARD_MASTER Values('CR001', 'Silver', 200, 5);  
Insert into LIBRARY_CARD_MASTER Values('CR002', 'Gold', 400, 9);  
Insert into LIBRARY_CARD_MASTER Values('CR003', 'Platinum', 600, 8);  
Insert into LIBRARY_CARD_MASTER Values('CR004', 'VISA', 800, 7);  
Insert into LIBRARY_CARD_MASTER Values('CR005', 'CREDIT', 1200, 6);  
  
Insert into MOVIES_MASTER Values('MV001', 'DIEHARD', '2012-05-13','ENGLISH', 4 ,  
'2HRS', 'U/A','ACTION','DIR1','L1','L2',100);  
Insert into MOVIES_MASTER Values('MV002', 'THE MATRIX', '2012-05-13','ENGLISH',  
4 , '2HRS', 'A','ACTION','DIR2','L1','L2',100);  
Insert into MOVIES_MASTER Values('MV003', 'INCEPTION', '2012-05-13','ENGLISH',  
4 , '2HRS', 'U/A','ACTION','DIR3','L15','L2',100);  
Insert into MOVIES_MASTER Values('MV004', 'DARK KNIGHT', '2012-05-  
13','ENGLISH', 4 , '2HRS', 'A','ACTION','DIR4','L15','L2',100);  
Insert into MOVIES_MASTER Values('MV005', 'OFFICE S', '2012-05-13','ENGLISH', 4 ,  
'2HRS', 'U/A','COMEDY','DIR5','L12','L24',100);  
Insert into MOVIES_MASTER Values('MV006', 'SHAWN OF DEAD', '2012-05-  
13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR6','L1','L25',100);  
Insert into MOVIES_MASTER Values('MV007', 'YOUNG FRANKEN', '2012-05-  
13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR7','L1','L2',100);  
Insert into MOVIES_MASTER Values('MV008', 'CAS', '2012-05-13','ENGLISH', 4 ,  
'2HRS', 'A','ROMANCE','DIR8','L1','L2',100);  
Insert into MOVIES_MASTER Values('MV009', 'GWW', '2012-05-13','ENGLISH', 4 ,  
'2HRS', 'A','ROMANCE','DIR9','L1','L2',100);  
Insert into MOVIES_MASTER Values('MV010', 'TITANIC', '2012-05-13','ENGLISH', 4 ,  
'2HRS', 'A','ROMANCE','DIR10','L1','L2',100);  
Insert into MOVIES_MASTER Values('MV011', 'THE NOTE BOOK', '2012-05-  
13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR11','L1','L2',100);
```

```
Insert into CUSTOMER_CARD_DETAILS Values('CUS001', 'CR001', '2012-05-13');
Insert into CUSTOMER_CARD_DETAILS Values('CUS002', 'CR002', '2012-05-13');
Insert into CUSTOMER_CARD_DETAILS Values('CUS003', 'CR002', '2013-05-13');
Insert into CUSTOMER_CARD_DETAILS Values('CUS004', 'CR003', '2013-05-13');
Insert into CUSTOMER_CARD_DETAILS Values('CUS005', 'CR003', '2012-05-13');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS001', 'CUS001', 'MV001', '2012-05-13', '2012-05-13','2012-05-13');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS002', 'CUS001', 'MV001', '2012-05-01', '2012-05-16','2012-05-16');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS003', 'CUS002', 'MV004', '2012-05-02', '2012-05-06','2012-05-16');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS004', 'CUS002', 'MV004', '2012-04-03', '2012-04-16','2012-04-20');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS005', 'CUS002', 'MV009', '2012-04-04', '2012-04-16','2012-04-20');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS006', 'CUS003', 'MV002', '2012-03-30', '2012-04-15','2012-04-20');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS007', 'CUS003', 'MV003', '2012-04-20', '2012-05-05','2012-05-05');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS008', 'CUS003', 'MV005', '2012-04-21', '2012-05-07','2012-05-25');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS009', 'CUS003', 'MV001', '2012-04-22', '2012-05-07','2012-05-25');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS010', 'CUS003', 'MV009', '2012-04-22', '2012-05-07','2012-05-25');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS011', 'CUS003', 'MV010', '2012-04-23', '2012-05-07','2012-05-25');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS012', 'CUS003', 'MV010', '2012-04-24', '2012-05-07','2012-05-25');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS013', 'CUS003', 'MV008', '2012-04-25', '2012-05-07','2012-05-25');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS014', 'CUS004', 'MV007', '2012-04-26', '2012-05-07','2012-05-25');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS015', 'CUS004', 'MV006', '2012-04-27', '2012-05-07','2012-05-25');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS016', 'CUS004', 'MV006', '2012-04-28', '2012-05-07','2012-05-25');
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS017', 'CUS004', 'MV001', '2012-04-29', '2012-05-07','2012-05-25');
```

```
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS018', 'CUS010', 'MV008', '2012-04-24', '2012-05-07','2012-05-25');
```

```
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS019', 'CUS011', 'MV009', '2012-04-27', '2012-05-07','2012-05-25');
```



MOVIE MASTER

LEAD_ROLE_2	RENT_COST
L2	100
L24	100
L25	100
L2	100
NULL	NULL

CUSTOMER MASTER

CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO	CONTACT_ADD	DATE_OF_REGISTRATION	AGE
CUS001	AMIT	9876543210	ADD1	2012-02-12	21
CUS002	ABDHUL	8765432109	ADD2	2012-02-12	21
CUS003	GAYAN	7654321098	ADD3	2012-02-12	21
CUS004	RADHA	6543210987	ADD4	2012-02-12	21
CUS005	GURU	NULL	ADD5	2012-02-12	21
CUS006	MOHAN	4321098765	ADD6	2012-02-12	21
CUS007	NAME7	3210987654	ADD7	2012-02-12	21
CUS008	NAME8	2109876543	ADD8	2013-02-12	21
CUS009	NAME9	NULL	ADD9	2013-02-12	21
CUS010	NAM10	9934567890	ADD10	2013-02-12	21
CUS011	NAM11	9875678910	ADD11	2013-02-12	21
NULL	NULL	NULL	NULL	NULL	NULL

LIBRARY CARD MASTER

CARD_ID	DESCRIPTION	AMOUNT	NUMBER_OF_YEARS
CR001	Silver	200	5
CR002	Gold	400	9
CR003	Platinum	600	8
CR004	VISA	800	7
CR005	CREDIT	1200	6
NULL	NULL	NULL	NULL

CUSTOMER CARD DETAILS

CUSTOMER_ID	CARD_ID	ISSUE_DATE
CUS001	CR001	2012-05-13
CUS002	CR002	2012-05-13
CUS003	CR002	2013-05-13
CUS004	CR003	2013-05-13
CUS005	CR003	2012-05-13
NULL	NULL	NULL

CUSTOMER ISSUE DETAILS

ISSUE_ID	CUSTOMER_ID	MOVIE_ID	ISSUE_DATE	RETURN_DATE	ACTUAL_DATE_RETURN
IS001	CUS001	MV001	2012-05-13	2012-05-13	2012-05-13
IS002	CUS001	MV001	2012-05-01	2012-05-16	2012-05-16
IS003	CUS002	MV004	2012-05-02	2012-05-06	2012-05-16
IS004	CUS002	MV004	2012-04-03	2012-04-16	2012-04-20
IS005	CUS002	MV009	2012-04-04	2012-04-16	2012-04-20
IS006	CUS003	MV002	2012-03-30	2012-04-15	2012-04-20
IS007	CUS003	MV003	2012-04-20	2012-05-05	2012-05-05
IS008	CUS003	MV005	2012-04-21	2012-05-07	2012-05-25
IS009	CUS003	MV001	2012-04-22	2012-05-07	2012-05-25
IS010	CUS003	MV009	2012-04-22	2012-05-07	2012-05-25
IS011	CUS003	MV010	2012-04-23	2012-05-07	2012-05-25
IS012	CUS003	MV010	2012-04-24	2012-05-07	2012-05-25
IS013	CUS003	MV008	2012-04-25	2012-05-07	2012-05-25
IS014	CUS004	MV007	2012-04-26	2012-05-07	2012-05-25
IS015	CUS004	MV006	2012-04-27	2012-05-07	2012-05-25
IS016	CUS004	MV006	2012-04-28	2012-05-07	2012-05-25
IS017	CUS004	MV001	2012-04-29	2012-05-07	2012-05-25
IS018	CUS010	MV008	2012-04-24	2012-05-07	2012-05-25
IS019	CUS011	MV009	2012-04-27	2012-05-07	2012-05-25
NULL	NULL	NULL	NULL	NULL	NULL

1. Write a query to display movie names and number of times that movie is issued to customers. Incase movies are never issued to customers display number of times as 0. Display the details in sorted order based on number of times (in descending order) and then by movie name (in ascending order). The Alias name for the number of movies issued is ISSUE_COUNT.

```
SELECT m.MOVIE_NAME, count(ISSUE_ID) ISSUE_COUNT FROM
movies_master m LEFT JOIN customer_issue_details c ON m.MOVIE_ID=c.MOVIE_ID
GROUP BY m.movie_name
ORDER BY ISSUE_COUNT DESC, MOVIE_NAME;
```

MOVIE_NAME	ISSUE_COUNT
DIEHARD	4
GWW	3
CAS	2
DARK KNIGHT	2
SHAWN OF DEAD	2
TITANIC	2
INCEPTION	1
OFFICE S	1
THE MATRIX	1
YOUNG FRANKEN	1
THE NOTE BOOK	0

2. Write a query to display id, name, age, contact no of customers whose age is greater than 25 and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

```
SELECT CUSTOMER_ID,CUSTOMER_NAME,AGE,ifnull(
concat('+91-',substring(contact_no,1,3),'-',
substring(contact_no,4,3),'-',substring(contact_no,7)),'N/A') CONTACT_ISD
FROM customer_master WHERE age>25 and year(date_of_registration)='2012'
ORDER BY age,CUSTOMER_NAME;
```

CUSTOMER_ID	CUSTOMER_NAME	AGE	CONTACT_ISD
-------------	---------------	-----	-------------

3. Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order. Hint: Use NO_OF_MOVIES as alias name for number of movies.

```
SELECT MOVIE_CATEGORY,count(MOVIE_ID) NO_OF_MOVIES FROM
movies_master GROUP BY MOVIE_CATEGORY
ORDER BY NO_OF_MOVIES DESC,MOVIE_CATEGORY;
```

MOVIE_CATEGORY	NO_OF_MOVIES
ACTION	4
ROMANCE	4
COMEDY	3

4. Write a query to display the number of customers having card with description "Gold card".
 Hint: Use CUSTOMER_COUNT as alias name for number of customers

```
SELECT count(c.customer_id) CUSTOMER_COUNT FROM
library_card_master l JOIN customer_card_details c ON l.CARD_ID=c.CARD_ID
WHERE description='Gold';
```

CUSTOMER_COUNT
2

5. Write a query to display the customer id, customer name, year of registration, library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order. Use REGISTERED_YEAR as alias name for year of registration.

```
SELECT c.customer_id,c.customer_name,
year(c.DATE_OF_REGISTRATION) REGISTERED_YEAR,cd.card_id,cd.issue_date
FROM
customer_master c JOIN customer_card_details cd ON c.customer_id=cd.customer_id
ORDER BY CUSTOMER_NAME DESC;
```

customer_id	customer_name	REGISTERED_YEAR	card_id	issue_date
CUS004	RADHA	2012	CR003	2013-05-13
CUS005	GURU	2012	CR003	2012-05-13
CUS003	GAYAN	2012	CR002	2013-05-13
CUS001	AMIT	2012	CR001	2012-05-13
CUS002	ABDHUL	2012	CR002	2012-05-13

6. Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer order by customer name.

```
SELECT ci.issue_id,ci.CUSTOMER_ID,c.CUSTOMER_NAME FROM
customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
WHERE customer_name LIKE 'R%' and ci.actual_date_return>ci.return_date
ORDER BY customer_name;
```

issue_id	CUSTOMER_ID	CUSTOMER_NAME
IS014	CUS004	RADHA
IS015	CUS004	RADHA
IS016	CUS004	RADHA
IS017	CUS004	RADHA

7. Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered. For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details. AMOUNT_DOLLAR = amount/52.42 and round it to zero decimal places and display as \$Amount. Example Assume 500 is the amount then dollar value will be

\$10. Hint: Use AMOUNT_DOLLAR as alias name for amount in dollar. Display the records in ascending order based on customer name.

```
SELECT c.CUSTOMER_ID,c.CUSTOMER_NAME,l.card_id,l.DESCRIPTION,  
concat('$',round(amount/52.42)) AMOUNT_DOLLAR FROM  
customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id  
JOIN customer_card_details cc ON cc.customer_id=c.customer_id  
JOIN library_card_master l ON cc.card_id=l.card_id  
WHERE c.DATE_OF_REGISTRATION=ci.issue_date  
ORDER BY customer_name;
```

CUSTOMER_ID	CUSTOMER_NAME	card_id	DESCRIPTION	AMOUNT_DOLLAR
-------------	---------------	---------	-------------	---------------

8. Write a query to display the customer id, customer name, contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'. Display customer name in upper case.

Hint: Use CUSTOMER_NAME as alias name for customer name. Display the details sorted in ascending order based on customer name.

```
SELECT CUSTOMER_ID,upper(CUSTOMER_NAME)  
CUSTOMER_NAME,contact_no,contact_add FROM  
customer_master WHERE contact_add LIKE '%Nagar' and  
customer_id NOT IN (SELECT customer_id FROM customer_card_details)  
and customer_id IN (SELECT customer_id FROM customer_issue_details)  
ORDER BY CUSTOMER_NAME;
```

CUSTOMER_ID	CUSTOMER_NAME	contact_no	contact_add
-------------	---------------	------------	-------------

9. Write a query to display the movie id, movie name, release year, director name of movies acted by the leadactor1 who acted maximum number of movies .Display the records sorted in ascending order based on movie name.

```
SELECT movie_id,movie_name,release_date,director FROM movies_master  
WHERE lead_role_1 IN(SELECT lead_role_1 FROM  
(SELECT lead_role_1,count(movie_id)ct FROM movies_master  
GROUP BY lead_role_1)t WHERE t.ct>=ALL(SELECT count(movie_id)  
FROM movies_master GROUP BY lead_role_1)) ORDER BY movie_name;
```

movie_id	movie_name	release_date	director
MV008	CAS	2012-05-13	DIR8
MV001	DIEHARD	2012-05-13	DIR1
MV009	GWW	2012-05-13	DIR9
MV006	SHAWN OF DEAD	2012-05-13	DIR6
MV002	THE MATRIX	2012-05-13	DIR2
MV011	THE NOTE BOOK	2012-05-13	DIR11
MV010	TITANIC	2012-05-13	DIR10
MV007	YOUNG FRANK...	2012-05-13	DIR7

10.Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0.
Hint: Use MOVIE_COUNT as alias name for number of movies issued.

```
SELECT c.customer_name, count(ci.movie_id) MOVIE_COUNT FROM
customer_master c LEFT JOIN customer_issue_details ci ON
c.customer_id=ci.customer_id
GROUP BY c.customer_id ORDER BY c.customer_name;
```

customer_name	MOVIE_COUNT
ABDHUL	3
AMIT	2
GAYAN	8
GURU	0
MOHAN	0
NAM10	1
NAM11	1
NAME7	0
NAME8	0
NAME9	0
RADHA	4

11.Write a query to display serial number,issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number.Serial number can be generated from the issue id , that is last two characters of issue id is the serial number. For Example Assume the issue id is I00005 then the serial number is 05 Hint: Alias name for serial number is 'SERIAL_NO'

```
SELECT substring(ci.issue_id,-2)
SERIAL_NO,ci.issue_id,c.customer_id,c.customer_name,
```

m.movie_id,m.movie_name FROM customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id JOIN movies_master m ON m.movie_id=ci.movie_id ORDER BY SERIAL_NO;

SERIAL_NO	issue_id	customer_id	customer_name	movie_id	movie_name
01	IS001	CUS001	AMIT	MV001	DIEHARD
02	IS002	CUS001	AMIT	MV001	DIEHARD
03	IS003	CUS002	ABDHUL	MV004	DARK KNIGHT
04	IS004	CUS002	ABDHUL	MV004	DARK KNIGHT
05	IS005	CUS002	ABDHUL	MV009	GWW
06	IS006	CUS003	GAYAN	MV002	THE MATRIX
07	IS007	CUS003	GAYAN	MV003	INCEPTION
08	IS008	CUS003	GAYAN	MV005	OFFICE S
09	IS009	CUS003	GAYAN	MV001	DIEHARD
10	IS010	CUS003	GAYAN	MV009	GWW
11	IS011	CUS003	GAYAN	MV010	TITANIC
12	IS012	CUS003	GAYAN	MV010	TITANIC
13	IS013	CUS003	GAYAN	MV008	CAS
14	IS014	CUS004	RADHA	MV007	YOUNG FRAN...
15	IS015	CUS004	RADHA	MV006	SHAWN OF D...
16	IS016	CUS004	RADHA	MV006	SHAWN OF D...
17	IS017	CUS004	RADHA	MV001	DIEHARD
18	IS018	CUS010	NAM10	MV008	CAS
19	IS019	CUS011	NAM11	MV009	GWW

12.Write a query to display the issue id,issue date, customer id, customer name and contact number for videos that are issued in the year 2013.Display the records in decending order based on issue date of the video.

SELECT ci.issue_id,ci.issue_date,c.customer_id,c.customer_name,c.contact_no FROM customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id and year(ci.issue_date)='2013' ORDER BY ci.issue_date DESC;

issue_id	issue_date	customer_id	customer_name	contact_no
----------	------------	-------------	---------------	------------

13.Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers.
 Actors Name to be displayed in the below format.LEAD_ACTOR_ONE space ambersant space

LEAD_ACTOR_TWO.Example: Assume lead actor one's name is "Jack Tomson" and Lead actor two's name is "Maria" then Actors name will be "Jack Tomsom &

Maria"
Hint:Use ACTORS as alias name for actors name.
 Display the records in ascending order based on movie name.

```
SELECT movie_id,movie_name,concat(lead_role_1,' & ',lead_role_2) ACTOR FROM movies_master  
WHERE movie_id NOT IN (SELECT movie_id FROM customer_issue_details) ORDER BY movie_name;
```

movie_id	movie_name	ACTOR
MV011	THE NOTE BOOK	L1 & L2

14.Write a query to display the director's name, movie name and lead_actor_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR_NAME as alias name for director name column Display the records sorted in ascending order based on director_name and then by movie_name in descending order.

```
SELECT upper(director) DIRECTOR_NAME,movie_name,lead_role_1 FROM movies_master  
GROUP BY director HAVING count(movie_id)>1 ORDER BY director,movie_name DESC;
```

DIRECTOR_NAME	movie_name	lead_role_1

15.Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number.
Hint:Use NO_OF_CUSTOMERS as alias name for number of customers.

```
SELECT count(customer_id) NO_OF_CUSTOMER FROM customer_master  
WHERE contact_no is not null and year(date_of_registration)='2012';
```

NO_OF_CUSTOMER
6

16.Write a query to display the customer's name, contact number,library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. **Hint:** Use CONTACT_DETAILS as alias name for customer contact.

```
SELECT c.customer_name,ifnull(c.contact_no,c.contact_add)  
CONTACT_DETAILS,l.card_id,l.description FROM  
customer_master c LEFT JOIN customer_card_details cc ON  
c.customer_id=cc.customer_id  
LEFT JOIN library_card_master l ON l.card_id=cc.card_id  
ORDER BY customer_name;
```

customer_name	CONTACT_DETAILS	card_id	description
ABDHUL	8765432109	CR002	Gold
AMIT	9876543210	CR001	Silver
GAYAN	7654321098	CR002	Gold
GURU	ADD5	CR003	Platinum
MOHAN	4321098765	NULL	NULL
NAM10	9934567890	NULL	NULL
NAM11	9875678910	NULL	NULL
NAME7	3210987654	NULL	NULL
NAME8	2109876543	NULL	NULL
NAME9	ADD9	NULL	NULL
RADHA	6543210987	CR003	Platinum

17. Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in descending order For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john.

Hint: Use NO_OF_TIMES as alias name for number of times

```
SELECT ci.customer_id,c.customer_name,count(ci.movie_id) NO_OF_TIMES FROM
customer_issue_details ci JOIN customer_master c ON c.customer_id=ci.customer_id
GROUP BY ci.customer_id,ci.movie_id HAVING count(movie_id)>1
ORDER BY customer_name DESC;
```

customer_id	customer_name	NO_OF_TIMES
CUS004	RADHA	2
CUS003	GAYAN	2
CUS001	AMIT	2
CUS002	ABDHUL	2

18. Write a query to display customer id, customer name, contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category.

Example: Display contact number as "+91-876-456-2345" format. Hint: Use NO_OF_MOVIES as alias name for number of movies column. Hint: Use CONTACT_ISD as alias name for contact number. Display the records sorted in ascending order based on customer name and then by movie category.

```
SELECT c.customer_id,c.customer_name,concat('+91-',substring(c.contact_no,1,3),'-',substring(c.contact_no,4,3),'-',substring(c.contact_no,7)) CONTACT_ISD
,m.movie_category,count(cc.movie_id) NO_OF_MOVIES FROM customer_master c
JOIN customer_issue_details cc
```

```

ON c.customer_id=cc.customer_id JOIN movies_master m ON
m.movie_id=cc.movie_id
GROUP BY c.customer_id,m.movie_category HAVING count(cc.movie_id)>1
ORDER BY customer_name,movie_category;

```

customer_id	customer_name	CONTACT_ISD	movie_category	NO_OF_MOVIES
CUS002	ABDHUL	+91-876-543-2109	ACTION	2
CUS001	AMIT	+91-987-654-3210	ACTION	2
CUS003	GAYAN	+91-765-432-1098	ACTION	3
CUS003	GAYAN	+91-765-432-1098	ROMANCE	4
CUS004	RADHA	+91-654-321-0987	COMEDY	3

19.Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies. For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.

```

SELECT cid.customer_id , customer_name FROM customer_master cm JOIN
customer_issue_details cid ON cm.customer_id=cid.customer_id
GROUP BY customer_id , customer_name
HAVING count(movie_id)>=ALL(SELECT count(movie_id)
FROM customer_issue_details
GROUP BY customer_id)
UNION
SELECT cid.customer_id , customer_name FROM
customer_master cm JOIN customer_issue_details cid
ON cm.customer_id=cid.customer_id
GROUP BY customer_id , customer_name
HAVING count(movie_id)<=ALL(SELECT count(movie_id)
FROM customer_issue_details
GROUP BY customer_id) ORDER BY customer_name;

```

customer_id	customer_name
CUS003	GAYAN
CUS010	NAM10
CUS011	NAM11

20. Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once. Hint: Use NO_OF_TIMES as alias name Display the records in ascending order based on customer name.

```
SELECT c.customer_id,c.customer_name,count(m.movie_id) NO_OF_TIMES FROM
customer_master c JOIN customer_issue_details cc ON c.customer_id=cc.customer_id
JOIN movies_master m ON m.movie_id=cc.movie_id
WHERE m.movie_category='Comedy'
GROUP BY c.customer_id HAVING count(m.movie_id)>1
ORDER BY customer_name;
```

customer_id	customer_name	NO_OF_TIMES
CUS004	RADHA	3

21. Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL_COST.

Display the records sorted in ascending order based on customer id

```
SELECT cid.customer_id, sum(m.rent_cost) TOTAL_COST FROM
customer_issue_details cid JOIN movies_master mm ON cid.movie_id=mm.movie_id
GROUP BY cid.customer_id order by customer_id;
```

customer_id	TOTAL_COST
CUS001	200
CUS002	300
CUS003	800
CUS004	400
CUS010	100
CUS011	100

LOAN

```
create database loan;
use loan;
```

```
CREATE TABLE loan_card_master
(
    loan_id      varchar(6)  PRIMARY KEY,
    loan_type    varchar(15),
    duration_in_years int(2)
);
```

```
CREATE TABLE employee_master
(
    employee_id      varchar(6)  PRIMARY KEY,
    employee_name    varchar(20),
    designation      varchar(25),
    department       varchar(25),
    gender           char(1),
    date_of_birth    date,
    date_of_joining  date
);
```

```
CREATE TABLE item_master
(
    item_id      varchar(6)  PRIMARY KEY,
    item_description    varchar(25),
```

```

        issue_status      char(1),
        item_make         varchar(25),
        item_category     varchar(20),
        item_valuation    int(6)
);

CREATE TABLE employee_card_details
(
    employee_id      varchar(6)   REFERENCES employee_master,
    loan_id          varchar(6)   REFERENCES loan_card_master,
    card_issue_date  date
);

CREATE TABLE employee_issue_details
(
    issue_id         varchar(6)   PRIMARY KEY,
    employee_id      varchar(6)   REFERENCES employee_master,
    item_id          varchar(6)   REFERENCES item_master,
    issue_date       date,
    return_date      date
);

insert into loan_card_master values('L00001','Furniture',5);
insert into loan_card_master values('L00002','Stationary',0);
insert into loan_card_master values('L00003','Crockery',1);

insert into employee_issue_details values('ISS001','E00001','I00001','2012-02-03','2014-02-03');
insert into employee_issue_details values('ISS002','E00001','I00004','2012-02-03','2020-02-03');
insert into employee_issue_details values('ISS003','E00002','I00005','2013-01-03','2015-01-03');
insert into employee_issue_details values('ISS004','E00003','I00007','2010-07-04','2012-07-04');
insert into employee_issue_details values('ISS005','E00003','I00008','2010-07-04','2012-08-05');
insert into employee_issue_details values('ISS006','E00003','I00010','2012-03-14','2012-06-15');

```

```

insert into employee_issue_details values('ISS007','E00004','I00012','2013-04-14','2016-04-14');
insert into employee_issue_details values('ISS008','E00006','I00018','2012-08-18','2019-04-17');
insert into employee_issue_details values('ISS009','E00004','I00018','2013-04-18','2013-05-18');

insert into employee_master values('E00001','Ram','Manager','Finance','M','1973-12-01','2000-01-01');
insert into employee_master values('E00002','Abhay','Assistant Manager','Finance','M','1976-01-01','2006-12-01');
insert into employee_master values('E00003','Anita','Senior Executive','Marketing','F','1977-05-12','2007-03-21');
insert into employee_master values('E00004','Zuben','Manager','Marketing','M','1974-10-12','2003-07-23');
insert into employee_master values('E00005','Radhica','Manager','HR','F','1976-07-22','2004-01-23');
insert into employee_master values('E00006','John','Executive','HR','M','1983-11-08','2010-05-17');

insert into employee_card_details values('E00001','L00001','2000-01-01');
insert into employee_card_details values('E00001','L00002','2000-01-01');
insert into employee_card_details values('E00001','L00003','2002-12-14');
insert into employee_card_details values('E00002','L00001','2007-02-01');
insert into employee_card_details values('E00002','L00002','2007-03-11');
insert into employee_card_details values('E00003','L00001','2007-04-15');
insert into employee_card_details values('E00003','L00002','2007-04-15');
insert into employee_card_details values('E00003','L00003','2007-04-15');

INSERT INTO item_master VALUES ('I00001','Tea Table','Y','Wooden','Furniture',5000);
INSERT INTO item_master VALUES ('I00002','Dinning Table','N','Wooden','Furniture',15000);
INSERT INTO item_master VALUES ('I00003','Tea Table','N','Steel','Furniture',6000);
INSERT INTO item_master VALUES ('I00004','Side Table','Y','Wooden','Furniture',2000);

```

```

INSERT INTO item_master VALUES ('I00005','Side
Table','Y','Steel','Furniture',1500);
INSERT INTO item_master VALUES ('I00006','Tea
Table','N','Steel','Furniture',7000);
INSERT INTO item_master VALUES ('I00007','Dinning
Chair','Y','Wooden','Furniture',1500);
INSERT INTO item_master VALUES ('I00008','Tea
Table','Y','Wooden','Furniture',4000);
INSERT INTO item_master VALUES ('I00009','Sofa','N','Wooden','Furniture',18000);
INSERT INTO item_master VALUES
('I00010','Cupboard','Y','Steel','Furniture',10000);
INSERT INTO item_master VALUES
('I00011','Cupboard','N','Steel','Furniture',14000);
INSERT INTO item_master VALUES ('I00012','Double
Bed','Y','Wooden','Furniture',21000);
INSERT INTO item_master VALUES ('I00013','Double
Bed','Y','Wooden','Furniture',20000);
INSERT INTO item_master VALUES ('I00014','Single
Bed','Y','Steel','Furniture',10000);
INSERT INTO item_master VALUES ('I00015','Single
Bed','N','Steel','Furniture',10000);
INSERT INTO item_master VALUES ('I00016','Tea Set','Y','Glass','Crockery',3000);
INSERT INTO item_master VALUES ('I00017','Tea
Set','Y','Bonechina','Crockery',4000);
INSERT INTO item_master VALUES ('I00018','Dinning
Set','Y','Glass','Crockery',4500);
INSERT INTO item_master VALUES ('I00019','Dinning
Set','N','Bonechina','Crockery',5000);
INSERT INTO item_master VALUES ('I00020','Pencil','Y','Wooden','Stationary',5);
INSERT INTO item_master VALUES ('I00021','Pen','Y','Plastic','Stationary',100);
INSERT INTO item_master VALUES ('I00022','Pen','N','Plastic','Stationary',200);

```

LOAN CARD MASTER

loan_id	loan_type	duration_in_years
L00001	Furniture	5
L00002	Stationary	0
L00003	Crockery	1
NULL	NULL	NULL

EMPLOYEE CARD DETAILS

employee_id	loan_id	card_issue_date
E00001	L00001	2000-01-01
E00001	L00002	2000-01-01
E00001	L00003	2002-12-14
E00002	L00001	2007-02-01
E00002	L00002	2007-03-11
E00003	L00001	2007-04-15
E00003	L00002	2007-04-15
E00003	L00003	2007-04-15

EMPLOYEE ISSUE DETAILS

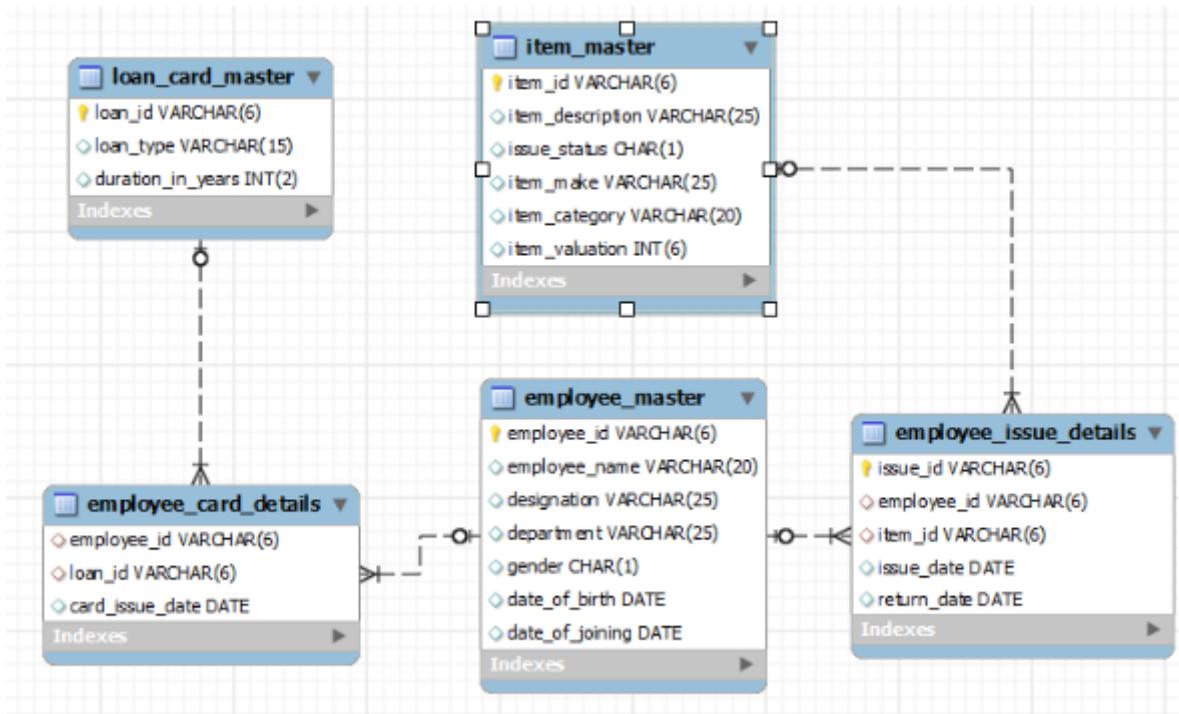
issue_id	employee_id	item_id	issue_date	return_date
ISS001	E00001	I00001	2012-02-03	2014-02-03
ISS002	E00001	I00004	2012-02-03	2020-02-03
ISS003	E00002	I00005	2013-01-03	2015-01-03
ISS004	E00003	I00007	2010-07-04	2012-07-04
ISS005	E00003	I00008	2010-07-04	2012-08-05
ISS006	E00003	I00010	2012-03-14	2012-06-15
ISS007	E00004	I00012	2013-04-14	2016-04-14
ISS008	E00006	I00018	2012-08-18	2019-04-17
ISS009	E00004	I00018	2013-04-18	2013-05-18
NULL	NULL	NULL	NULL	NULL

EMPLOYEE MASTER

employee_id	employee_name	designation	department	gender	date_of_birth	date_of_joining
E00001	Ram	Manager	Finance	M	1973-12-01	2000-01-01
E00002	Abhay	Assistant Manager	Finance	M	1976-01-01	2006-12-01
E00003	Anita	Senior Executive	Marketing	F	1977-05-12	2007-03-21
E00004	Zuben	Manager	Marketing	M	1974-10-12	2003-07-23
E00005	Radhica	Manager	HR	F	1976-07-22	2004-01-23
E00006	John	Executive	HR	M	1983-11-08	2010-05-17
NULL	NULL	NULL	NULL	NULL	NULL	NULL

ITEM MASTER

item_id	item_description	issue_status	item_make	item_category	item_valuation
I00001	Tea Table	Y	Wooden	Furniture	5000
I00002	Dinning Table	N	Wooden	Furniture	15000
I00003	Tea Table	N	Steel	Furniture	6000
I00004	Side Table	Y	Wooden	Furniture	2000
I00005	Side Table	Y	Steel	Furniture	1500
I00006	Tea Table	N	Steel	Furniture	7000
I00007	Dinning Chair	Y	Wooden	Furniture	1500
I00008	Tea Table	Y	Wooden	Furniture	4000
I00009	Sofa	N	Wooden	Furniture	18000
I00010	Cupboard	Y	Steel	Furniture	10000
I00011	Cupboard	N	Steel	Furniture	14000
I00012	Double Bed	Y	Wooden	Furniture	21000
I00013	Double Bed	Y	Wooden	Furniture	20000
I00014	Single Bed	Y	Steel	Furniture	10000
I00015	Single Bed	N	Steel	Furniture	10000
I00016	Tea Set	Y	Glass	Crockery	3000
I00017	Tea Set	Y	Bonechina	Crockery	4000
I00018	Dinning Set	Y	Glass	Crockery	4500
I00019	Dinning Set	N	Bonechina	Crockery	5000
I00020	Pencil	Y	Wooden	Stationary	5
I00021	Pen	Y	Plastic	Stationary	100
I00022	Pen	N	Plastic	Stationary	200
NULL	NULL	NULL	NULL	NULL	NULL



1. Write a query to display category and number of items in that category. Give the count an alias name of Count_category. Display the details on the sorted order of count in descending order.

```
SELECT item_category, count(item_id) Count_category FROM item_master GROUP BY item_category ORDER BY Count_category DESC;
```

item_category	Count_category
Furniture	15
Crockery	4
Stationary	3

2. Write a query to display the number of employees in HR department. Give the alias name as No_of_Employees.

```
SELECT count(employee_id) No_of_Employees FROM employee_master WHERE department='HR';
```

No_of_Employees
2

3. Write a query to display employee id, employee name, designation and department for employees who have never been issued an item as a loan from the company. Display the records sorted in ascending order based on employee id.

```
SELECT employee_id, employee_name, designation, department FROM employee_master WHERE employee_id NOT IN (SELECT employee_id FROM employee_issue_details) ORDER BY employee_id;
```

employee_id	employee_name	designation	department
E00005	Radhica	Manager	HR
NULL	NULL	NULL	NULL

4. Write a query to display the employee id, employee name who was issued an item of highest valuation. In case of multiple records, display the records sorted in ascending order based on employee id.[Hint Suppose an item called dinning table is of 22000 and that is the highest price of the item that has been issued. So display the employee id and employee name who issued dinning table whose price is 22000.]

```
SELECT employee_id,employee_name FROM employee_master
WHERE employee_id IN(SELECT employee_id FROM employee_issue_details
WHERE item_id IN (SELECT item_id FROM item_master
WHERE item_valuation=(SELECT max(item_valuation) FROM
item_master i JOIN employee_issue_details e ON i.item_id=e.item_id)));
```

employee_id	employee_name
E00004	Zuben
NULL	NULL

5. Write a query to display issue_id, employee_id, employee_name. Display the records sorted in ascending order based on issue id.

```
SELECT eid.issue_id, eid.employee_id, em.employee_name
FROM employee_master em JOIN employee_issue_details eid
ON em.employee_id=eid.employee_id ORDER BY eid.issue_id;
```

issue_id	employee_id	employee_name
ISS001	E00001	Ram
ISS002	E00001	Ram
ISS003	E00002	Abhay
ISS004	E00003	Anita
ISS005	E00003	Anita
ISS006	E00003	Anita
ISS007	E00004	Zuben
ISS008	E00006	John
ISS009	E00004	Zuben

6. Write a query to display employee id, employee name who don't have loan cards. Display the records sorted in ascending order based on employee id.

```
SELECT employee_id,employee_name FROM employee_master
WHERE employee_id NOT IN(SELECT employee_id FROM employee_card_details);
```

employee_id	employee_name
E00004	Zuben
E00005	Radhica
E00006	John
NULL	NULL

7. Write a query to count the number of cards issued to an employee “Ram”. Give the count an alias name as No_of_Cards.

```
SELECT count(loan_id) No_of_Cards FROM  
employee_card_details WHERE employee_id IN  
(SELECT employee_id FROM employee_master WHERE employee_name='Ram');  
(or)
```

```
SELECT count(loan_id) No_of_Cards FROM  
employee_card_details c JOIN employee_master e  
ON c.employee_id = e.employee_id  
WHERE e.employee_name= 'Ram';
```

No_of_Cards

3

8. Write a query to display the count of customers who have gone for loan type stationary. Give the count an alias name as Count_stationary.

```
SELECT count(e.employee_id) Count_Stationary  
FROM employee_card_details e JOIN loan_card_master l  
ON e.loan_id=l.loan_id WHERE l.loan_type='Stationary';
```

Count_Stationary

3

9. Write a query to display the employee id, employee name and number of items issued to them. Give the number of items an alias name as Count. Display the details in descending order of count and then

```
SELECT e.employee_id,employee_name,count(e.item_id) Count FROM  
employee_issue_details e JOIN employee_master em ON e.employee_id=em.employee_id  
GROUP BY e.employee_id ORDER BY count DESC,e.employee_id;
```

employee_id	employee_name	Count
E00003	Anita	3
E00001	Ram	2
E00004	Zuben	2
E00002	Abhay	1
E00006	John	1

10. Write a query to display the employee id, employee name who was issued an item of minimum valuation.In case of multiple records, display them sorted in ascending order based on employee id.[Hint Suppose an item called pen is of rupees 20 and that is the lowest price. So display the employee id and employee name who issued pen where the valuation is 20.]

```
SELECT employee_id,employee_name FROM employee_master  
WHERE employee_id IN(SELECT employee_id FROM employee_issue_details  
WHERE item_id IN (SELECT item_id FROM item_master  
WHERE item_valuation=(SELECT min(item_valuation) FROM  
item_master i JOIN employee_issue_details e ON i.item_id=e.item_id)))  
ORDER BY employee_id;
```

employee_id	employee_name
E00002	Abhay
E00003	Anita
NULL	NULL

11. Write a query to display the employee id, employee name and total valuation of the product issued to each employee. Give the alias name as TOTAL_VALUATION. Display the records sorted in ascending order based on employee id. Consider only employees who have been issued atleast 1 item.

```
SELECT e.employee_id,em.employee_name,sum(i.item_valuation) TOTAL_VALUATION FROM item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id ORDER BY employee_id;
```

employee_id	employee_name	TOTAL_VALUATION
E00001	Ram	7000
E00002	Abhay	1500
E00003	Anita	15500
E00004	Zuben	25500
E00006	John	4500

12. Write a query to display distinct employee id, employee name who kept the item issued for more than a year. Hint: Use Date time function to calculate the difference between item issue and return date. Display the records only if it is more than 365 Days. Display the records sorted in ascending order based on employee id.

```
SELECT DISTINCT e.employee_id,e.employee_name FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
WHERE datediff(ei.return_date,ei.issue_date)>365
ORDER BY employee_id;
```

employee_id	employee_name
E00001	Ram
E00002	Abhay
E00003	Anita
E00004	Zuben
E00006	John

13. Write a query to display employee id, employee name and count of items of those who asked for more than 1 furniture. Give the alias name for count of items as COUNT_ITEMS. Display the records sorted in ascending order on employee id.

```
SELECT e.employee_id,e.employee_name,count(ei.item_id) COUNT_ITEMS FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON ei.item_id=i.item_id
WHERE i.item_category='Furniture'
GROUP BY ei.employee_id HAVING count(ei.item_id)>1;
```

employee_id	employee_name	COUNT_ITEMS
E00001	Ram	2
E00003	Anita	3

14. Write a query to display the number of men & women Employees. The query should display the gender and number of Employees as No_of_Employees. Display the records sorted in ascending order based on gender.

```
SELECT gender,count(employee_id) FROM employee_master
GROUP BY gender ORDER BY gender;
```

gender	count(employee_id)
F	2
M	4

15. Write a query to display employee id, employee name who joined the company after 2005. Display the records sorted in ascending order based on employee id.

```
SELECT employee_id,employee_name FROM employee_master
WHERE year(date_of_joining)>'2005'
ORDER BY employee_id;
```

employee_id	employee_name
E00002	Abhay
E00003	Anita
E00006	John
NULL	NULL

16. Write a query to get the number of items of the furniture category issued and not issued. The query should display issue status and the number of furniture as No_of_Furnitures. Display the records sorted in ascending order based on issue_status.

```
SELECT issue_status,count(item_id) No_of_Furniture
FROM item_master WHERE item_category='Furniture'
GROUP BY issue_status ORDER BY issue_status;
```

issue_status	No_of_Furniture
N	6
Y	9

17. Write a query to find the number of items in each category, make and description. The Query should display Item Category, Make, description and the number of items as No_of_Items. Display the records in ascending order based on Item Category, then by item make and then by item description.

```
SELECT item_category,item_make,item_description,count(item_id) No_of_items
FROM item_master GROUP BY item_category,item_make,item_description
ORDER BY item_category,item_make,item_description;
```

item_category	item_make	item_description	No_of_items
Crockery	Bonechina	Dinning Set	1
Crockery	Bonechina	Tea Set	1
Crockery	Glass	Dinning Set	1
Crockery	Glass	Tea Set	1
Furniture	Steel	Cupboard	2
Furniture	Steel	Side Table	1
Furniture	Steel	Single Bed	2
Furniture	Steel	Tea Table	2
Furniture	Wooden	Dinning Chair	1
Furniture	Wooden	Dinning Table	1
Furniture	Wooden	Double Bed	2
Furniture	Wooden	Side Table	1
Furniture	Wooden	Sofa	1
Furniture	Wooden	Tea Table	2
Stationary	Plastic	Pen	2
Stationary	Wooden	Pencil	1

18. Write a query to display employee id, employee name, item id and item description of employees who were issued item(s) in the month of January 2013. Display the records sorted in order based on employee id and then by item id in ascending order.

```
SELECT e.employee_id,employee_name,i.item_id,i.item_description FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON i.item_id=ei.item_id
WHERE month(ei.issue_date)='01' and year(ei.issue_date)='2013'
ORDER BY employee_id,item_id;
```

employee_id	employee_name	item_id	item_description
E00002	Abhay	I00005	Side Table

19. Write a query to display the employee id, employee name and count of item category of the employees who have been issued items in at least 2 different categories. Give the alias name for category count as COUNT_CATEGORY. Display the records sorted in ascending order based on employee id.

```
SELECT ei.employee_id,e.employee_name,count(DISTINCT i.item_category)
COUNT_CATEGORY FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON i.item_id=ei.item_id
GROUP BY ei.employee_id
HAVING COUNT_CATEGORY>=2
ORDER BY employee_id;
```

employee_id	employee_name	COUNT_CATEGORY
E00004	Zuben	2

20. Write a query to display the item id , item description which was never issued to any employee. Display the records sorted in ascending order based on item id.

```
SELECT item_id, item_description FROM item_master  
WHERE item_id NOT IN (SELECT item_id from employee_issue_details)  
ORDER BY item_id;
```

item_id	item_description
I00002	Dinning Table
I00003	Tea Table
I00006	Tea Table
I00009	Sofa
I00011	Cupboard
I00013	Double Bed
I00014	Single Bed
I00015	Single Bed
I00016	Tea Set
I00017	Tea Set
I00019	Dinning Set
I00020	Pencil
I00021	Pen
I00022	Pen
NULL	NULL

21. Write a query to display the employee id, employee name and total valuation for the employees who has issued minimum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION.[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed.]

```
SELECT e.employee_id,em.employee_name,sum(i.item_valuation) TOTAL_VALUATION FROM  
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id  
JOIN employee_master em ON em.employee_id=e.employee_id  
GROUP BY e.employee_id HAVING sum(i.item_valuation)<=ALL(  
SELECT sum(i.item_valuation) TOTAL_VALUATION FROM  
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id  
JOIN employee_master em ON em.employee_id=e.employee_id  
GROUP BY e.employee_id);
```

employee_id	employee_name	TOTAL_VALUATION
E00002	Abhay	1500

22. Write a query to display the employee id, employee name, card issue date and card valid date. Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD_VALID_DATE.[Hint: Validity in years for the loan card is given in loan_card_master table. Validity date is calculated by adding number of years

in the loan card issue date. If the duration of year is zero then display AS 'No Validity Date'.

]

```
SELECT e.employee_id,e.employee_name,card_issue_date,
case
when l.duration_in_years>0 then date_add(ec.card_issue_date,interval l.duration_in_years year)
when l.duration_in_years=0 then 'No Validity Date' end CARD_VALID_DATE
FROM
employee_master e JOIN employee_card_details ec ON e.employee_id=ec.employee_id
JOIN loan_card_master l ON l.loan_id=ec.loan_id
ORDER BY employee_name,CARD_VALID_DATE;
```

employee_id	employee_name	card_issue_date	CARD_VALID_DATE
E00002	Abhay	2007-02-01	2012-02-01
E00002	Abhay	2007-03-11	No Validity Date
E00003	Anita	2007-04-15	2008-04-15
E00003	Anita	2007-04-15	2012-04-15
E00003	Anita	2007-04-15	No Validity Date
E00001	Ram	2002-12-14	2003-12-14
E00001	Ram	2000-01-01	2005-01-01
E00001	Ram	2000-01-01	No Validity Date

23. Write a query to display the employee id, employee name who have not issued with any item in the year 2013. Hint: Exclude those employees who was never issued with any of the items in all the years. Display the records sorted in ascending order based on employee id.

```
SELECT DISTINCT e.employee_id,e.employee_name FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
WHERE e.employee_id NOT IN (SELECT employee_id FROM employee_issue_details
WHERE year(issue_date)='2013')
ORDER BY employee_id;
```

employee_id	employee_name
E00001	Ram
E00003	Anita
E00006	John

24. Write a query to display issue id, employee id, employee name, item id, item description and issue date. Display the data in descending order of date and then by issue id in ascending order.

```
SELECT issue_id, eid.employee_id, employee_name, im.item_id, item_description,issue_date
FROM employee_issue_details eid JOIN employee_master em ON
eid.employee_id=em.employee_id
JOIN item_master im ON eid.item_id=im.item_id
ORDER BY issue_date DESC, issue_id;
```

issue_id	employee_id	employee_name	item_id	item_description	issue_date
ISS009	E00004	Zuben	I00018	Dinning Set	2013-04-18
ISS007	E00004	Zuben	I00012	Double Bed	2013-04-14
ISS003	E00002	Abhay	I00005	Side Table	2013-01-03
ISS008	E00006	John	I00018	Dinning Set	2012-08-18
ISS006	E00003	Anita	I00010	Cupboard	2012-03-14
ISS001	E00001	Ram	I00001	Tea Table	2012-02-03
ISS002	E00001	Ram	I00004	Side Table	2012-02-03
ISS004	E00003	Anita	I00007	Dinning Chair	2010-07-04
ISS005	E00003	Anita	I00008	Tea Table	2010-07-04

25. Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION.[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name and total valuation of E00019 should display.]

```

SELECT e.employee_id, em.employee_name, sum(i.item_valuation) TOTAL_VALUATION
FROM item_master i JOIN employee_issue_details e ON e.item_id = i.item_id
JOIN employee_master em ON em.employee_id = e.employee_id
GROUP BY e.employee_id HAVING sum(i.item_valuation) >= ALL(
SELECT sum(i.item_valuation) TOTAL_VALUATION
FROM item_master i JOIN employee_issue_details e ON e.item_id = i.item_id
JOIN employee_master em ON em.employee_id = e.employee_id
GROUP BY e.employee_id);

```

employee_id	employee_name	TOTAL_VALUATION
E00004	Zuben	25500

http://nammatkadai.blogspot.com/2016/03/dump-flight.html#:~:text=location%20is%20chennai.-,Display%20the%20records%20sorted%20in%20ascending%20order%20based%20on%20profile,Use%20FLIGHT_COUNT%20as%20alias%20name

```
SELECT cus.profile_id, cus.first_name, cus.address, mintkt.No_of_Tickets
FROM air_passenger_profile cus, (SELECT MIN(s.Tot_No_of_Tickets) AS No_of_Tickets
FROM (SELECT profile_id, COUNT(ticket_id) AS Tot_No_of_Tickets FROM air_ticket_info
GROUP BY profile_id) s) mintkt,
(SELECT profile_id, COUNT(ticket_id) AS Tot_No_of_Tickets FROM air_ticket_info GROUP BY
profile_id) totkt
WHERE mintkt.No_of_Tickets = totkt.Tot_No_of_Tickets
AND cus.profile_id = totkt.profile_id
order by cus.first_name;
```

Video Management database queries:

1. Please follow instructions given below.

Write a query to display movie names and number of times that movie is issued to customers.

Incase movies are never issued to customers display number of times as 0.

Display the details in sorted order based on number of times (in descending order) and then by movie name (in ascending order).

The Alias name for the number of movies issued is ISSUE_COUNT.

11 rows

```
select mm.movie_name, count(cid.issue_id) as ISSUE_COUNT  
from movies_master mm left outer join customer_issue_details  
cid on mm.movie_id=cid.movie_id group by mm.movie_name  
order by ISSUE_COUNT desc,mm.movie_name asc;
```

MOVIE_NAME	ISSUE_COUNT
DIE HARD	4
GONE WITH THE WIND	3
CASABLANCA	2
SHAUN OF THE DEAD	2
THE DARK KNIGHT	2
TITANIC	2
INCEPTION	1
OFFICE SPACE	1
THE MATRIX	1

YOUNG FRANKENSTEIN	1
THE NOTEBOOK	0

2. Please follow instructions given below.

Write a query to display id, name, age, contact no of customers whose age is greater than 25 and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

4 rows

```
select customer_id,customer_name,age,
ifnull(concat('+91-',substring(contact_no,1,3),'-',substring(contact_no,4,3),'-',
'substring(contact_no,7,4)),'N/A')
as CONTACT_ISD from customer_master where age>25 and
year(date_of_registration)=2012
order by age,customer_name;
```

CUSTOMER_ID	CUSTOMER_NAME	AGE	CONTACT_ISD
C00007	GEETHA REDDY	30	+91-897-616-7890
C00005	SHIV PRASAD	30	N/A
C00002	AGNESH	35	+91-892-315-6781
C00004	RAJIB MITRA	45	+91-983-035-6781

3. Please follow instructions given below.

Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order.

Hint: Use NO_OF_MOVIES as alias name for number of movies.

3 rows

Ans:

```

select movie_category,count(movie_id) as NO_OF_MOVIES from movies_master group by movie_category
order by NO_OF_MOVIES desc,movie_category asc;

```

MOVIE_CATEGORY	NO_OF_MOVIES
ACTION	4
ROMANCE	4
COMEDY	3

4. Please follow instructions given below.

Write a query to display the number of customers having card with description "Gold card".

 Hint: Use CUSTOMER_COUNT as alias name for number of customers

1 row

```

select count(ccd.customer_id) as CUSTOMER_COUNT from customer_card_details ccd
join
library_card_master lcd on ccd.card_id= lcd.card_id where lcd.description='Gold Card';

```

CUSTOMER_COUNT
2

4. Please follow instructions given below.

Write a query to display the customer id, customer name, year of registration, library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order.

Use REGISTERED_YEAR as alias name for year of registration.

5 rows

```

select cm.customer_id,cm.customer_name,year(cm.date_of_registration) as
REGISTERED_YEAR,ccd.card_id,ccd.issue_date
from customer_master cm join customer_card_details ccd on cm.customer_id=ccd.customer_id
order by cm.customer_name desc;

```

CUSTOMER_ID	CUSTOMER_NAME	REGISTERED_YEAR	CARD_ID	ISSUE_DATE

C00003	T RAMACHANDRAN	2012	CRD002	2012-11-02
C00005	SHIV PRASAD	2012	CRD003	2012-12-26
C00004	RAJIB MITRA	2012	CRD003	2012-11-21
C00001	NITIN	2012	CRD001	2012-10-15
C00002	AGNESH	2012	CRD002	2012-12-01

5. Please follow instructions given below.

Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer.

Display the records sorted in ascending order based on customer name.

2 rows

```
select cid.issue_id,cid.customer_id,cm.customer_name from customer_issue_details cid
join
customer_master cm on cid.customer_id=cm.customer_id where cm.customer_name like
'R%'
and cid.actual_date_return>cid.return_date order by cm.customer_name;
```

ISSUE_ID	CUSTOMER_ID	CUSTOMER_NAME
I00008	C00010	RAGHAV SINGH
I00007	C00004	RAJIB MITRA

6. Please follow instructions given below.

Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered.

For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details.

AMOUNT_DOLLAR = amount/52.42 and round it to zero decimal places and display as \$Amount. Example Assume 500 is the amount then dollar value will be \$10.

Hint: Use **AMOUNT_DOLLAR** as alias name for amount in dollar.

Display the records in ascending order based on customer name.

```
SELECT ccd.customer_id, customer_name, ccd.card_id,
description,concat('$',round(amount/52.42,0)) AMOUNT_DOLLAR FROM
customer_master cm INNER JOIN customer_card_details ccd ON
cm.customer_id=ccd.customer_id INNER JOIN library_card_master lcm ON
ccd.card_id=lcm.card_id INNER JOIN customer_issue_details cid ON cid.customer_id =
cm.customer_id WHERE cm.date_of_registration=cid.issue_date order by
customer_name;
```

CUSTOMER_ID	CUSTOMER_NAME	CARD_ID	DESCRIPTION	AMOUNT_DOLLAR
C00001	NITIN	CRD001	SILVER CARD	\$19
C00004	RAJIB MITRA	CRD003	PLATINUM CARD	\$57
C00003	T RAMACHANDRAN	CRD002	GOLD CARD	\$38

7. Please follow instructions given below.

Write a query to display the customer id, customer name, contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'.

Display customer name in upper case. Hint: Use **CUSTOMER_NAME** as alias name for customer name. Display the details sorted in ascending order based on customer name.

```
SELECT customer_id , upper(customer_name)
CUSTOMER_NAME,contact_no,contact_address FROM customer_master WHERE
customer_id NOT IN ( select customer_id from customer_card_details ) AND customer_id
IN ( SELECT customer_id from customer_issue_details ) and contact_address like
'%Nagar' order by customer_name ;
```

CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO	CONTACT_ADDRESS

C00010	RAGHAV SINGH	9675167890	A/6 NEHRU JAWAHAR NAGAR
--------	--------------	------------	-------------------------

8. Please follow instructions given below.

Write a query to display the movie id, movie name, release year, director name of movies acted by the lead actor who acted maximum number of movies. Display the records sorted in ascending order based on movie name.

```
select movie_id, movie_name , release_date , director from movies_master where
lead_actor_name1 in(select lead_actor_name1 from(select
lead_actor_name1, count(movie_id) ct from movies_master group by lead_actor_name1)t where
t.ct>=all(select count(movie_id) from movies_master
group by lead_actor_name1))order by movie_name;
```

MOVIE_ID	MOVIE_NAME	RELEASE_YEAR	DIRECTOR_NAME
M00004	INCEPTION	2010	CHRISTOPHER NOLAN
M00011	TITANIC	1997	JAMES CAMERON

9. Please follow instructions given below.

Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0.
 Hint: Use MOVIE_COUNT as alias name for number of movies issued.

11 rows

```
select cm.customer_name, count(cid.movie_id) as MOVIE_COUNT from customer_master cm
left join
customer_issue_details cid on cm.customer_id=cid.customer_id group by cm.customer_name
order by cm.customer_name;
```

CUSTOMER_NAME	MOVIE_COUNT
AGNESH	3

AJAY GHOSH	0
GEETHA REDDY	0
NITIN	2
RAGHAV SINGH	1
RAJ SEKHANRAN	1
RAJAN PILLAI	0
RAJIB MITRA	4
RIA NATRAJAN	0
SHIV PRASAD	0
T RAMACHANDRAN	8

10. Please follow instructions given below.

Write a query to display serial number, issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number.

Serial number can be generated from the issue id , that is last two characters of issue id is the serial number.

For Example Assume the issue id is I00005 then the serial number is 05

Hint: Alias name for serial number is 'SERIAL_NO'

19 rows

```
select substring(cid.issue_id,5,2) as
SERIAL_NO,cid.issue_id,cid.customer_id,cm.customer_name,mm.movie_id,mm.movie_n
ame
from customer_issue_details cid join customer_master cm on
cm.customer_id=cid.customer_id
```

```

join movies_master mm on cid.movie_id=mm.movie_id group by
SERIAL_NO,cid.customer_id,mm.movie_id
order by SERIAL_NO;

```

SERIAL_NO	ISSUE_ID	CUSTOMER_ID	CUSTOMER_NAME	MOVIE_ID	MOVIE_NAME
01	I00001	C00001	NITIN	M00001	DIE HARD
02	I00002	C00002	AGNESH	M00002	THE DARK KNIGHT
03	I00003	C00002	AGNESH	M00002	THE DARK KNIGHT
04	I00004	C00003	T RAMACHANDRAN	M00003	THE MATRIX
05	I00005	C00003	T RAMACHANDRAN	M00004	INCEPTION
06	I00006	C00003	T RAMACHANDRAN	M00005	OFFICE SPACE
07	I00007	C00004	RAJIB MITRA	M00006	YOUNG FRANKENSTEIN
08	I00008	C00010	RAGHAV SINGH	M00008	CASABLANCA
09	I00009	C00011	RAJ SEKHANRAN	M00010	GONE WITH THE WIND
10	I00010	C00004	RAJIB MITRA	M00007	SHAUN OF THE DEAD

11	I00011	C00004	RAJIB MITRA	M00007	SHAUN OF THE DEAD
12	I00012	C00001	NITIN	M00001	DIE HARD
13	I00013	C00003	T RAMACHANDRAN	M00001	DIE HARD
14	I00014	C00003	T RAMACHANDRAN	M00010	GONE WITH THE WIND
15	I00015	C00003	T RAMACHANDRAN	M00011	TITANIC
16	I00016	C00003	T RAMACHANDRAN	M00011	TITANIC
17	I00017	C00003	T RAMACHANDRAN	M00008	CASABLANCA
18	I00018	C00002	AGNESH	M00010	GONE WITH THE WIND
19	I00019	C00004	RAJIB MITRA	M00001	DIE HARD

11. Please follow instructions given below.

Write a query to display the issue id, issue date, customer id, customer name and contact number for videos that are issued in the year 2013. Display the records in descending order based on issue date of the video.

7 rows

```
select cid.issue_id,cid.issue_date,cid.customer_id,cm.customer_name,cm.contact_no
from customer_issue_details cid join customer_master cm on
cid.customer_id=cm.customer_id
where year(issue_date)=2013 group by issue_id,issue_date,customer_id order by
issue_date desc;
```

ISSUE_ID	ISSUE_DATE	CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO
I00012	2013-11-28	C00001	NITIN	9830354218
I00017	2013-04-15	C00003	T RAMACHANDRAN	9831289761
I00009	2013-03-16	C00011	RAJ SEKHANRAN	8423178906
I00016	2013-03-05	C00003	T RAMACHANDRAN	9831289761
I00008	2013-03-02	C00010	RAGHAV SINGH	9675167890
I00015	2013-02-03	C00003	T RAMACHANDRAN	9831289761
I00014	2013-01-02	C00003	T RAMACHANDRAN	9831289761

12. Please follow instructions given below.

Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers.
 Actors Name to be displayed in the below format LEAD_ACTOR_ONE space ambersant space LEAD_ACTOR_TWO.

Example: Assume lead actor one's name is "Jack Tomson" and Lead actor two's name is "Maria" then Actors name will be "Jack Tomsom & Maria" Hint: Use ACTORS as alias name for actors name.
 Display the records in ascending order based on movie name.

1 row

```
select movie_id,movie_name,concat(lead_actor_name1,' & ',lead_actor_name2) as
ACTORS
from movies_master where movie_id
not in (select movie_id from customer_issue_details) order by
movie_name;
```

MOVIE_ID	MOVIE_NAME	ACTORS
M00009	THE NOTEBOOK	RYAN GOSLING &

		RACHEL MCADAMS
--	--	-------------------

13. Please follow instructions given below.

Write a query to display the director's name, movie name and lead_actor_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR_NAME as alias name for director name column Display the records sorted in ascending order based on director_name and then by movie_name in descending order.

2 rows

```
SELECT upper(director_name) DIRECTOR_NAME,movie_name,lead_actor_name1 FROM
movies_master WHERE director_name in (SELECT director_name FROM movies_master
GROUP BY director_name HAVING count(movie_id)>1) order by director_name,
movie_name desc;
```

DIRECTOR_NAME	MOVIE_NAME	LEAD_ACTOR_NAME1
CHRISTOPHER NOLAN	THE DARK KNIGHT	CHRISTIAN BALE
CHRISTOPHER NOLAN	INCEPTION	LEONARDO DICAPRIO

14. Please follow instructions given below.

Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number.
 Hint: Use NO_OF_CUSTOMERS as alias name for number of customers.

1 row

```
select count(customer_id) as NO_OF_CUSTOMERS from customer_master where
year(date_of_registration)
=2012 and contact_no != 'NULL'
```

NO_OF_CUSTOMERS
6

15. Please follow instructions given below.

Write a query to display the customer's name, contact number, library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT_DETAILS as alias name for customer contact.

11 rows

```
select cm.customer_name,ifnull(cm.contact_no,cm.contact_add) as
CONTACT_DETAILS,lcd.card_id,lcd.description from customer_master cm
left join customer_card_details ccd on cm.customer_id=ccd.customer_id
left join library_card_master lcd on ccd.card_id=lcd.card_id group by
customer_name,description,CONTACT_DETAILS
order by customer_name;
```

CUSTOMER_NAME	CONTACT_DETAILS	CARD_ID	DESCRIPTION
AGNESH	8923156781	CRD002	GOLD CARD
AJAY GHOSH	8763478901	NULL	NULL
GEETHA REDDY	8976167890	NULL	NULL
NITIN	9830354218	CRD001	SILVER CARD
RAGHAV SINGH	9675167890	NULL	NULL
RAJ SEKHANRAN	8423178906	NULL	NULL
RAJAN PILLAI	A 1/66 KODAMBAKKAM	NULL	NULL
RAJIB MITRA	9830356781	CRD003	PLATINUM CARD
RIA NATRAJAN	9856723190	NULL	NULL

SHIV PRASAD	2/2 PHASE II, JAWAHAR NAGAR	CRD003	PLATINUM CARD
T RAMACHANDRAN	9831289761	CRD002	GOLD CARD

16. Please follow instructions given below.

Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in descending order. For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john. Hint: Use NO_OF_TIMES as alias name for number of times

4 rows

```
select cm.customer_id,cm.customer_name,count(cid.movie_id) as NO_OF_TIMES from
customer_master
cm join customer_issue_details cid on cm.customer_id=cid.customer_id group by
customer_id,movie_id having
count(movie_id)>1 order by customer_name desc;
```

CUSTOMER_ID	CUSTOMER_NAME	NO_OF_TIMES
C00003	T RAMACHANDRAN	2
C00004	RAJIB MITRA	2
C00001	NITIN	2
C00002	AGNESH	2

17. Please follow instructions given below.

Write a query to display customer id, customer name, contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-456-2345" format.

Hint: Use NO_OF_MOVIES as alias name for number of movies column.

Hint: Use CONTACT_ISD as alias name for contact number.

Display the records sorted in ascending order based on customer name and then by movie category.

5 rows

```
select cid.customer_id,cm.customer_name,  
concat('+91-',substring(cm.contact_no,1,3),'-',substring(cm.contact_no,4,3),'-',  
substring(cm.contact_no,7,4)) as CONTACT_ISD,  
mm.movie_category,count(mm.movie_category) as NO_OF_MOVIES from  
customer_master  
cm join customer_issue_details cid  
on cm.customer_id=cid.customer_id join movies_master mm on  
cid.movie_id=mm.movie_id  
group by mm.movie_category,cm.customer_name having count(movie_category)>1  
order by cm.customer_name,mm.movie_category;
```

CUSTOMER_ID	CUSTOMER_NAME	CONTACT_ISD	MOVIE_CATEGORY	NO_OF_MOVIES
C00002	AGNESH	+91-892-315-6781	ACTION	2
C00001	NITIN	+91-983-035-4218	ACTION	2
C00004	RAJIB MITRA	+91-983-035-6781	COMEDY	3
C00003	T RAMACHANDRAN	+91-983-128-9761	ACTION	3
C00003	T RAMACHANDRAN	+91-983-128-9761	ROMANCE	4

18. Please follow instructions given below.

Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies.

For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The name and id of Ram should be displayed for

issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.

3 rows

```
(select cm.customer_id,cm.customer_name from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id
having count(cid.issue_id) >= all (select count(cid.issue_id) from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id) order by
cm.customer_name)
union all
(select cm.customer_id,cm.customer_name from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id
having count(cid.issue_id) <= all (select count(cid.issue_id) from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id) order by
cm.customer_name);
```

CUSTOMER_ID	CUSTOMER_NAME
C00003	T RAMACHANDRAN
C00010	RAGHAV SINGH
C00011	RAJ SEKHANRAN

19. Please follow instructions given below.

Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once.

Hint: Use NO_OF_TIMES as alias name

Display the records in ascending order based on customer name.

1 row

```
select cm.customer_id,cm.customer_name,count(mm.movie_id) as NO_OF_TIMES from
customer_master cm
join customer_issue_details cid on cm.customer_id=cid.customer_id join
```

```

movies_master mm on cid.movie_id=mm.movie_id where mm.movie_category='comedy' group
by customer_id
order by customer_name>1;

```

CUSTOMER_ID	CUSTOMER_NAME	NO_OF_TIMES
C00004	RAJIB MITRA	3

20. Please follow instructions given below.

Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL_COST. Display the records sorted in ascending order based on customer id

6 rows

```

select cid.customer_id,sum(mm.rental_cost) as TOTAL_COST from customer_issue_details cid
join movies_master mm
on cid.movie_id=mm.movie_id group by customer_id order by customer_id;

```

	customer_id	TOTAL_COST
▶	C00001	200
	C00002	300
	C00003	1700
	C00004	400
	C00010	1000
	C00011	100

```
create table CUSTOMER_MASTER
(
    CUSTOMER_ID Varchar(10),
    CUSTOMER_NAME Varchar(30) NOT NULL,
    CONTACT_NO BIGINT(20),
    CONTACT_ADD Varchar(100),
    DATE_OF_REGISTRATION Date NOT NULL,
    AGE Varchar(15)NOT NULL,
    Constraint MT_cts1 PRIMARY KEY(CUSTOMER_ID)
);
```

```
Create table MOVIES_MASTER
(
    MOVIE_ID Varchar(10),
    MOVIE_NAME Varchar(80) NOT NULL,
    RELEASE_DATE Varchar(30) NOT NULL,
    LANGUAGE Varchar(30),
    RATING int(2),
    DURATION_In_Minutes VARCHAR(10) NOT NULL,
    MOVIE_TYPE Varchar(100),
    MOVIE_CATEGORY VARCHAR(40) NOT NULL,
    DIRECTOR VARCHAR(60) NOT NULL,
    LEAD_Actor_name1 Varchar(50) NOT NULL,
    LEAD_Actor_name2 VARCHAR(60) NOT NULL,
    RENTAL_COST BIGINT(10),
    Constraint MT_cts4 PRIMARY KEY(MOVIE_ID)
);
```

```
Create table CUSTOMER_ISSUE_DETAILS
(
    ISSUE_ID Varchar(10) NOT NULL,
    CUSTOMER_ID Varchar(10) NOT NULL,
    MOVIE_ID VARCHAR(10),
    ISSUE_DATE Date NOT NULL,
    RETURN_DATE Date NOT NULL,
    ACTUAL_DATE_RETURN Date NOT NULL,
    Constraint MT_cts5 PRIMARY KEY(ISSUE_ID),
    Constraint MT_Mem FOREIGN KEY(CUSTOMER_ID) References
    CUSTOMER_MASTER(CUSTOMER_ID),
    Constraint MT_Mem1 FOREIGN KEY(MOVIE_ID) References
    MOVIES_MASTER(MOVIE_ID)
);
```

```
Create table LIBRARY_CARD_MASTER
(
    CARD_ID Varchar(10),
    DESCRIPTION Varchar(30) NOT NULL,
    AMOUNT      BIGINT(50),
    NUMBER_OF_YEARS bigint(10) NOT NULL,
    Constraint MT_cts2 PRIMARY KEY(CARD_ID)
);
```

```
Create table CUSTOMER_CARD_DETAILS
(
    CUSTOMER_ID Varchar(10),
    CARD_ID VARCHAR(10),
    ISSUE_DATE DATE NOT NULL,
    Constraint MT_cts3 PRIMARY KEY(CUSTOMER_ID),
    Constraint MT_CTS41 FOREIGN KEY(CUSTOMER_ID) References
    CUSTOMER_MASTER(CUSTOMER_ID),
    Constraint MT_CTS42 FOREIGN KEY(CARD_ID) References
    LIBRARY_CARD_MASTER(CARD_ID)
);
```

```
insert into customer_master values
('C00001',' NITIN ','9830354218',' A/122, KALKAJI',' 2012-10-15', 22),

('C00002',' AGNESH ','8923156781',' '9/1,ANDHERI EAST',' 2012-11-01',
35),
```

('C00003','T RAMACHANDRAN','9831289761','9/1,NANDABAKKAM','2012-11-02', 25),

('C00004','RAJIB MITRA','9830356781','H/56, BLOCK1,JADAVPUR','2012-11-21', 45),

('C00005','SHIV PRASAD','NULL,'2/2 PHASE II JAWAHAR NAGAR','2012-12-25', 30),

('C00006','AJAY GHOSH','8763478901','N/2,GANDHI COLONY DUM DUM','2012-12-30', 20),

('C00007','GEETHA REDDY','8976167890','AH 1/1 T NAGAR','2012-12-31', 30),

('C00008','RIA NATRAJAN','9856723190','A/B GANDHI COLONY','2013-01-01', 45),

('C00009','RAJAN PILLAI','NULL,'A 1/66 KODAMBAKKAM','2013-01-02', 40),

('C00010','RAGHAV SINGH','9675167890','A/6 NEHRU JAWAHAR NAGAR','2013-03-02', 50),

('C00011','RAJ SEKHANRAN','8423178906','A/1 MAYUR KUNJ','2013-03-15', 25);

insert into movies_master values
(‘M00001’, ‘DIE HARD’, 1998, ‘ENGLISH’, 4, 120, ‘UNIVERSAL’,
‘ACTION’,
‘JOHN MCTIERNAN’, ‘BRUCE WILLIS’, ‘BONNIE BEDELIA’, 100),

(‘M00002’, ‘THE DARK KNIGHT’, 2008, ‘ENGLISH’, 5, 90, ‘PARENTAL
GUIDENCE’, ‘ACTION’,
‘CHRISTOPHER NOLAN’,
‘CHRISTIAN BALE’, ‘HEALTH LEDGER’, 100),

(‘M00003’, ‘THE MATRIX’, 1999, ‘ENGLISH’, 4, 120, ‘UNIVERSAL’, ‘ACTION’,
‘ANDY LARRY’, ‘KEANU REEVES’,
‘CARRIE-ANEE MOSS’, 100),

(‘M00004’, ‘INCEPTION’, 2010, ‘ENGLISH’, 5, 120, ‘PARENTAL GUIDENCE
’, ‘ACTION’,
‘CHRISTOPHER NOLAN’, ‘LEONARDO DICAPRIO’, ‘JOSEPH GORDAN’, 100),

(‘M00005’, ‘OFFICE SPACE’, 1999, ‘ENGLISH’, 4, 95, ‘UNIVERSAL’, ‘COMEDY
’,
‘MIKE JUDGE’, ‘RON LIVINGSTON’, ‘JENNIFER ANISTON’, 100),

(‘M00006’, ‘YOUNG FRANKENSTEIN’, 1974, ‘ENGLISH’, 4, 130, ‘UNIVERSAL’,
‘COMEDY’,
‘MEL BROOKS’, ‘GENE WILDER’, ‘TERI GARR’, 100),

(‘M00007’, ‘SHAUN OF THE DEAD’, 2004, ‘ENGLISH’, 4, 95, ‘UNIVERSAL’,
‘COMEDY’,
‘EDGAR WRIGHT’, ‘SIMON PEGG’, ‘KATE ASHFIELD’, 100),

(‘M00008’, ‘CASABLANCA’, 1942, ‘ENGLISH’, 3, 120, ‘UNIVERSAL’,
‘ROMANCE’, ‘MICHAEL
CURTIZ’, ‘HUMPREY BOGART’, ‘INGRID BERGMAN’, 1000),

('M00009','THE NOTEBOOK', 2004,'ENGLISH', 3 ,120 , 'PARENTAL
GUIDENCE', ' ROMANCE ',
'NICK CASSAVETES',' RYAN GOSLING',' RACHEL MCADAMS', 100),

('M00010','GONE WITH THE WIND ',1939 , 'ENGLISH', 3 ,120, ' PARENTAL
GUIDENCE ','ROMANCE',
'VICTOR FLEMMING', 'CLARK GABLE', ' VIVIEN LEIGH ',100),

('M00011','TITANIC', 1997 , 'ENGLISH ',3 ,120 , 'PARENTAL GUIDENCE', '
ROMANCE'
, 'JAMES CAMERON', LEONARDO DICAPRIO , 'KATE WINSLET', 100);

insert into customer_issue_details values
(I00001,'C00001','M00001','2012-10-15','2012-10-17','2012-10-17');

insert into customer_issue_details values(I00002, 'C00002','M00002','2012-11-02' , '2012-
11-04','2012-11-05');

insert into customer_issue_details values
(I00003,'C00002','M00002','2012-12-02','2012-12-04','2012-12-03');

insert into customer_issue_details values(I00004,'C00003','M00003','2012-11-02',2012-11-
04,'2012-11-10');

insert into customer_issue_details values(I00005,'C00003','M00004','2012-11-10',
'2012-11-12','2012-11-12');

insert into customer_issue_details values(I00006,'C00003','M00005',
'2012-11-12','2012-11-14','2012-11-14');

insert into customer_issue_details values(I00007,'C00004','M00006' , '2012-11-21',
'2012-11-23','2012-11-24');

insert into customer_issue_details values(I00008,'C00010','M00008','2013-03-02',
'2013-03-04','2013-03-05');

insert into customer_issue_details values(I00009,'C00011','M00010' , '2013-03-16','2013-
03-18','2013-03-18');

insert into customer_issue_details values(I00010,'C00004','M00007',
'2012-11-25','2012-11-27','2012-11-27');

insert into customer_issue_details values(I00011,'C00004','M00007','2012-11-28',
'2012-11-30','2012-11-30');

```
insert into customer_issue_details values('I00012','C00001','M00001','2013-11-28','2013-11-30','2013-11-30');
```

```
insert into customer_issue_details values('I00013','C00003','M00001','2012-12-03','2012-12-05','2012-12-05');
```

```
insert into customer_issue_details values('I00014','C00003','M00010      ','2013-01-02','2013-01-04','2013-01-05');
```

```
insert into customer_issue_details values('I00015','C00003','M00011      ','2013-02-03','2013-02-05','2013-02-06');
```

```
insert into customer_issue_details values('I00016','C00003','M00011      ','2013-03-05','2013-03-07','2013-03-07');
```

```
insert into customer_issue_details values('I00017','C00003','M00008','2013-04-15','2013-04-17','2013-04-17');
```

```
insert into customer_issue_details values('I00018','C00002','M00010      ','2015-01-15','2015-01-17','2015-01-17');
```

```
insert into customer_issue_details values('I00019','C00004','M00001','2012-11-15      ','2012-11-17','2012-11-17');
```

```
insert into library_card_master values
```

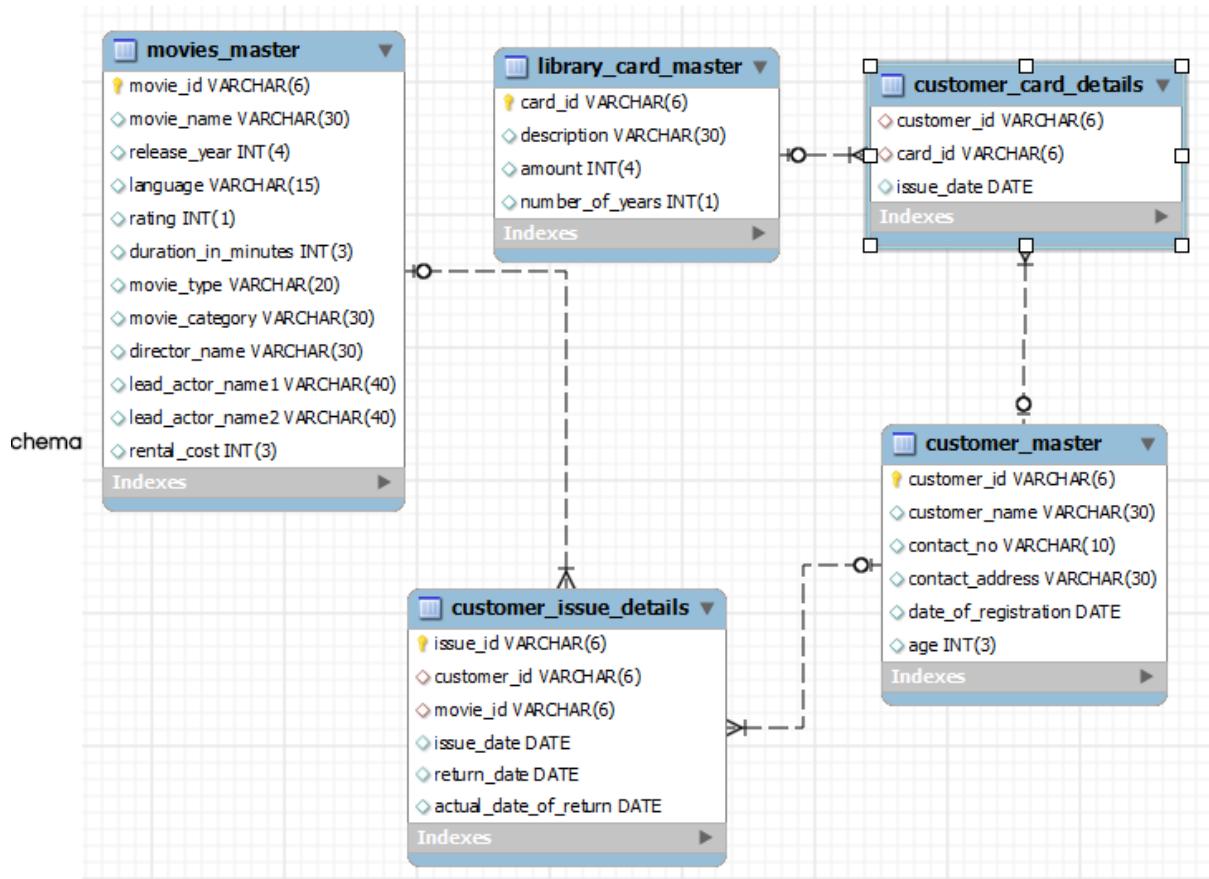
```
('CRD001',    'SILVER CARD'      ,1000, 1),
```

```
('CRD002',    'GOLD CARD ' ,2000 ,2),
```

```
('CRD003',    'PLATINUM CARD'   ,3000, 3),
```

```
('CRD004',    'DIAMOND CARD',  4000 ,5);
```

```
Insert into CUSTOMER_CARD_DETAILS Values('C00001','CRD001','2012-05-13');  
Insert into CUSTOMER_CARD_DETAILS Values('C00002','CRD002','2012-05-13');  
Insert into CUSTOMER_CARD_DETAILS Values('C00003','CRD002','2013-05-13');  
Insert into CUSTOMER_CARD_DETAILS Values('C00004','CRD003','2013-05-13');  
Insert into CUSTOMER_CARD_DETAILS Values('C00005','CRD003','2012-05-13');
```



///////////Event Hall-Average cost of booked halls
select CITY,round(avg(cost_perday),0) as AVERAGE_COST from t_hall_details
where hall_id in(Select hall_id from t_hall_booking) and capacity>150
group by CITY
order by average_cost;

///////////Event Hall-Average cost of not booked halls
SELECT
 City, ROUND(AVG(cost_perday)) 'Average_Cost'
FROM
 T_hall_details
WHERE
 Hall_id NOT IN (SELECT
 Hall_id
 FROM
 T_hall_booking)
 AND capacity > 100
GROUP BY city;

///////////Dream Home-Customer name details based on total cost
SELECT c.customer_name,SUM(f.cost_inlakh) FROM t_flat_booking b
JOIN t_flat_details f ON b.flat_no=f.flat_no
JOIN t_customer_details C ON c.customer_id=b.customer_id
WHERE LENGTH(c.customer_name)>'10'
GROUP BY c.customer_name
ORDER BY customer_name

///////////Hospital-Total fees received based on gender and shift
SELECT t_patient.gender, sum(t_doctor.fees) FEES_RECEIVED
FROM t_patient
JOIN t_doctor on t_doctor.doctor_id=t_patient.doctor_id
JOIN t_hospital on t_hospital.available_doctor=t_doctor.doctor_id
WHERE upper(t_hospital.shift_time)="MORNING"
GROUP BY t_patient.gender
ORDER BY t_patient.gender DESC;

///////////Insurance-List of Agents
select a.agent_id,p.policy_name,sum(p.policy_sum)as policy_sum from t_agent a
join t_member m on m.agent_id=a.agent_id
join t_policy p on p.policy_id=m.policy_id
group by a.agent_id,p.policy_name
having count(m.member_id) >=1
order by a.agent_id,p.policy_name,policy_sum;
///////////Minimum & Maximum Discount Amount
SELECT MIN(DISCOUNT_AMOUNT) AS MIN_DISCOUNT,
MAX(DISCOUNT_AMOUNT) AS MAX_DISCOUNT

```

FROM DISCOUNT_MASTER;
///////////////////////////////Number of Appointments..

SELECT doctor_id, COUNT(app_number) as APPOINTMENT_COUNT
FROM appointment
GROUP BY doctor_id
ORDER BY doctor_id;
Student Details In Capital Case..
select student_id,upper(student_name) as NAME,department,phone_no
from student_details
where address='BANGALORE'
order by student_id;

/////////////////////////////Pizza-Delivery Partner Details With Rating..
select partner_id,partner_name,concat(partner_id,substr(partner_name,1,4))
Name,if(rating>=9,'Excellent',if(rating>=7 and rating<9,'Good','Worst')) as FEEDBACK from
delivery_partner order by partner_id;
/////////////////////////////Pizza-Framing Customer password ..
Select concat (cust_name,cust_id)
As USERNAME
Concat (substring(cust_name,1,3),
Substring(cust_phone,-4,4) as PASSWORD
FROM customer
ORDER BY USERNAME;

/////////////////////////////Pizza-Low cost and High cost pizza..
Select pizza_type, pizza_name from pizza
Where amount IN(select max(amount) from pizza) or
Amount IN(SELECT MIN(amount) from pizza)
LIMIT 2;
/////////////////////////////Cricket-Average runs of players based on name..

select m.player_id, round(avg(m.player_runs)) as average_runs from t_match_score_card
m
join t_player p
on p.player_id=m.player_id where player_name like 'S%'
group by m.player_id
order by average_runs desc;
/////////////////////////////Car Pooling-Vehicle details .
select c.vehicle_model,c.vehicle_type,sum(ci.distance)from car c
join booking b on b.vehicle_no=c.vehicle_no
join city_locations ci on (ci.city1=b.pickup_from and
ci.city2=b.drop_at)or(ci.city2=b.pickup_from and ci.city1=b.drop_at)
group by c.vehicle_type,c.vehicle_model
order by sum(ci.distance);

```

///////////Event Hall-Customer details with booking done..

```
SELECT customer_id
,customer_name
,mobile_no
FROM t_customer_details
WHERE length(customer_name) > 10
AND customer_id IN (
    SELECT customer_id
    FROM (
        SELECT customer_id
        ,count(hall_id)
        FROM t_hall_booking
        GROUP BY customer_id
        HAVING count(hall_id) > (
            SELECT count(h.hall_id)
            FROM t_hall_booking h
            INNER JOIN t_customer_details c ON c.customer_id = h.customer_id
            WHERE c.customer_name = 'Suman Singh'
            GROUP BY h.customer_id
        )
    ) AS T1
)
```

ORDER BY customer_name;

///////////Dream Home -Flat details based on year..

```
select a.flat_no FLAT_NO, b.size SIZE, b.area area
from t_flat_booking a
join t_flat_details b
on a.flat_no = b.flat_no
where year(a.registration_date)
in (select year(b.registration_date)
from t_customer_details a
join t_flat_booking b
on a.customer_id=b.customer_id
where upper(a.customer_name='Niraj Kumar'))
order by area asc,a.flat_no desc;
```

///////////Cricket-Player details..

```
SELECT DISTINCT p.PLAYER_ID,p.PLAYER_NAME,p.PLAYER_CITY from t_player p
join t_match_score_card s on p.player_id=s.player_id
join t_match_record r on r.match_id=s.match_id
WHERE s.wayof_dismissal='STUMPED' AND r.played_city='BANGALORE'
order by player_name desc;
```

///////////Room Details Based On Location..

```
select
ROOM_DETAILS.ROOM_ID,ROOM_DETAILS.ROOM_TYPE,ROOM_DETAILS.MEMBER_C
APACITY,ROOM_DETAILS.ROOM_RENT
from ROOM_DETAILS
inner join HOSTEL_DETAILS
on ROOM_DETAILS.HOSTEL_ID=HOSTEL_DETAILS.HOSTEL_ID
where HOSTEL_DETAILS.LOCATION = 'PHASE-A'
order by ROOM_DETAILS.ROOM_ID;
```

/////////////////////////////Patient Appointment details Based On Month..

```
SELECT DISTINCT(PATIENT_ID),P_FIRST_NAME,P_AGE,ADDRESS,CONTACT_NUMBER
FROM PATIENT
WHERE PATIENT_ID IN(SELECT PATIENT_ID FROM APPOINTMENT WHERE APP_DATE
BETWEEN '2019-06-01' AND '2019-06-31')
ORDER BY PATIENT_ID;
```

//////////////////cricket number of players in each city

```
select player_city as PLAYER_CITY, count(player_id) as NUMBER_OF_PLAYERS from
t_player where player_city not in
(select distinct played_city from t_match_record)
group by player_city
order by NUMBER_OF_PLAYERS,PLAYER_CITY;
```

//////////////////Hospital-Maximum fees paid patient details..

```
select p.patient_name , d.doctor_name , d.fees as 'fees_paid' , h.shift_time as
'checkup_done'
from t_patient p
join t_doctor d on p.doctor_id = d.doctor_id
join t_hospital h on h.available_doctor = d.doctor_id
where d.fees > ( select max(fees) from t_doctor where specialization = 'DERMA')
order by d.doctor_name , p.patient_name;
```

//////////////////Insurance-Agent details..

```
select count(b.member_id) as NUMBER_OF_MEMBERS,a.agent_name as AGENT_NAME
from t_agent a join t_member b
on a.agent_id=b.agent_id
where a.agent_name like 'S%' or a.agent_name like 's%'
group by a.agent_name
order by AGENT_NAME,NUMBER_OF_MEMBERS asc;
Concatenating Details ..
SELECT CONCAT(MOVIE_NAME," is a ",LANGUAGE," Movie") AS MOVIE_DETAILS
FROM MOVIE_MASTER
ORDER BY MOVIE_DETAILS DESC;
```

```

///////////////////Pizza-Highest Business Customer Details..
select pizza.cust_id, customer.cust_name, sum(pizza.amount) as Max_Amount from customer
join pizza on customer.cust_id=pizza.cust_id group by pizza.cust_id order by Max_Amount
desc limit 1;
///////////////////Pizza-Total Cost of Pizza Ordered..
select cust_id, pizza_name, count(cust_id) as 'Times taken', sum(amount) as 'Total cost'
from pizza
where amount > 1200 group by pizza_name , cust_id order by cust_id asc;
/////////////////// Pizza-Extra Large Pizza ..
///////////////////Event Hall-Customer having Average payment..

select cd.customer_name, round(avg(hd.cost_perday)) as A from t_customer_details cd join
t_hall_booking hb
on cd.customer_id=hb.customer_id
join t_hall_details hd on hd.hall_id=hb.hall_id
group by cd.customer_name
having A > (select max(hdd.cost_perday) from t_hall_details hdd join
t_hall_booking hbb on hbb.hall_id=hdd.hall_id
join t_customer_details cdd on cdd.customer_id=hbb.customer_id
where cdd.customer_name='Suraj Kumar')
order by A desc;
///////////////////Car Pooling-Maximum time driven driver details ..

select b.driver_id, d.driver_name, count(driver_id) as MaxTimesDriven
from driver d
inner join booking b on d.id=b.driver_id
group by b.driver_id
having count(driver_id)>2
order by b.driver_id;

///////////////////Hostel-Insert Student Records
insert into Student_details values
('S1001','Varsha','ECE','1999-06-12','CHENNAI',9845712345,'varsha123@gmail.com'),
('S1002','William','ECE','1999-02-04','CALCUTTA',6845712345,'william123@gmail.com'),
('S1003','Basha','EEE','1999-06-14','DELHI',9945712345,'basha222@gmail.com'),
('S1004','Catherine','CSE','1998-08-16','DELHI',6785712345,'cathu123@gmail.com'),
('S1005','Kate','ECE','1999-06-30','BANGALORE',7685712345,'katedd@gmail.com'),
('S1006','Michel','ECE','1998-06-04','COIMBATORE',6645712345,'michel000@gmail.com');

///////////////////Movie - Modify the datatype..
ALTER TABLE CUSTOMER_MASTER MODIFY COLUMN PHONE_NO INT(10);

///////////////////Create Movie_Master table set1..

```

|||||||||||||||||||||Hospital- Change the datatype/column ..
alter table patient modify
contact_number int(10);
alter table patient change p_age patient_age int;

|||||||||||||||||Hospital- Add a new column set1..
alter table doctor add column dr_contact_number int(10);

|||||||||||||||||Pizza Store- Update PIZZA table discount1.2..
UPDATE pizza
set amount = (amount * 0.95)/100
Where pizza_type = " Extra Large" ;

|||||||||||||||||Pizza Store- Alter table-Foreign key 1.1..
ALTER TABLE pizza ADD CONSTRAINT FK1 FOREIGN KEY (cust_id) REFERENCES
customer(cust_id);
ALTER TABLE pizza ADD CONSTRAINT FK2 FOREIGN KEY (partner_id) REFERENCES
delivery_partner(partner_id);
|||||||||||||Pizza Store - Update PIZZA table 1.2..

|||||||||||||Pizza Store- Alter table Pizza1.1..

|||||||||Event Hall- Update the event date1.2..

|||||||||Event Hall -Update T_HALL_DETAILS table1.2..

|||||||||Event Hall-Alter T_HALL_BOOKING table1|||||||||1 ..
alter table t_hall_booking
modify hall_id varchar(10) not null;
alter table t_hall_booking
add foreign key(hall_id) references t_hall_details(hall_id);
alter table t_hall_booking
modify customer_id varchar(10) not null;
alter table t_hall_booking
add foreign key(customer_id) references t_customer_details(customer_id);

|||||||||Dream Home- Update t_flat_details table1.2..

|||||||||Dream Home- Alter table t_flat_booking1.1..

```

//////////////////Cricket-Update T_PLAYER table(1.2)..
update t_player
set total_wickets=case
when(player_city='BANGALORE' and player_name like 'A%')
THEN total_wickets+5
when(player_city='DELHI' and player_name like 'A%')
THEN total_wickets+7
ELSE total_wickets
END;
//////////////////Cricket -Alter T_MATCH_SCORE_CARD table(1.1)..  

alter table t_match_score_card add foreign key (match_id) references t_match_record  

(match_id);
alter table t_match_score_card add foreign key (player_id) references t_player(player_id);
//////////////////Car Pooling - Update booking table1.2..
update booking
set fare=(select min(distance)*11 from city_locations ct
join booking b On b.pickup_from=ct.city1 AND b.drop_at=ct.city2);

//////////////////Car Pooling- Create BOOKING table 1.1
create table booking (
booking_no varchar(50),
pickup_from varchar(50),
drop_at varchar(50),
customer_id varchar(50),
vehicle_no varchar(50),
driver_id varchar(50),
fare decimal(7,2),
primary key (booking_no),
foreign key (customer_id) references customer(id),
foreign key (vehicle_no) references car(vehicle_no),
foreign key (driver_id) references driver(id)
);
//////////////////Hospital-Update T_DOCTOR table 1.2..
update t_doctor set fees=350
where specialization="ENT" and doctor_name like "J%";
update t_doctor set fees=600
where specialization="DERMA" and doctor_name like "J%";
update t_doctor set fees=null
where specialization="SURGEON" and doctor_name like "J%";
update t_doctor set fees=null
where specialization="ORTHO" and doctor_name like "J%";

//////////////////Hospital- Alter T_HOSPITAL table 1.1..
alter table t_hospital
add foreign key (available_doctor) references t_doctor(doctor_id);

```

||||||||||||||||||||||||Insurance-Update Agent details(1.2)..
Update t_agent
set target_policy_sum=
case
when upper(agent_city)='PUNE' and upper(agent_id) like 'M%'
then 400000
when upper(agent_city)='CHENNAI' and upper(agent_id) like 'M%'
then 250000
else target_policy_sum
end;

||||||||||||||||||||||||Insurance- Alter table-add constraint(1.1)..
alter table T_MEMBER
ADD foreign key(AGENT_ID) references T_AGENT(agent_id),
ADD foreign key(POLICY_ID) references T_POLICY(policy_id);

||||||||||||||||||||Event Hall- Alter table Hall Booking 1.1..
alter table t_hall_booking
modify hall_id varchar(10) not null;
alter table t_hall_booking
add foreign key(hall_id) references t_hall_details(hall_id);

|||||||||||||||||||||Pizza Store-Update PIZZA table 1.2..

|||||||||||||||||||||Pizza Store- Alter table Pizza 1.1..

|||||||||||||||||||||Patient Appointment Details based on reason..

|||||||||||||||||||||2.Pizza-Highest Business Date..

|||||||||||||||||||||2. Event Hall-Number of booking customer wise ..
select distinct c.customer_id,c.customer_name,count(h.hall_id) as NO_OF_BOOKING
from t_customer_details c
right join t_hall_booking h
on c.customer_id = h.customer_id
where h.event_date like '2020%'
group by c.customer_id
having c.customer_name like 'S%'
order by 2;

|||||||||||||||||||||2. Dream Home-Maximum cost of flat
SELECT floor_no AS FLOOR_NO,MAX(cost_inlakh) AS MAX_PRICE FROM t_flat_details
GROUP BY floor_no

```
ORDER BY floor_no DESC
```

|||||||||||||||||||||||2.Cricket-Number of players in each city..

```
select player_city as PLAYER_CITY, count(player_id) as NUMBER_OF_PLAYERS from
t_player where player_city not in
(select distinct played_city from t_match_record)
group by player_city
order by NUMBER_OF_PLAYERS,PLAYER_CITY;
|||||||||||||||||||||||2.Car pooling-Driver booking details based on name..
select a.booking_no,b.user_name,c.driver_name,a.pickup_from,a.drop_at,d.distance
from booking a join customer b on a.customer_id=b.id
join driver c on a.driver_id=c.id
join city_locations d on ((a.pickup_from=d.city1 and
a.drop_at=d.city2)or(a.pickup_from=d.city2 and a.drop_at=d.city1))
where upper(c.driver_name)='JOE AMAL'
order by d.distance
```

|||||||||||||||||||2.Hospital-Number of doctors based on shift..

```
select h.shift_time as SHIFT_TIME,count(h.available_doctor) AS NUMBER_OF_DOCTORS
from t_hospital h
join t_doctor d on d.doctor_id=h.available_doctor
where specialization = 'SURGEON'
group by shift_time
having count(available_doctor)>=1
order by shift_time desc;
```

|||||||||||||||||||Movie details based on Certification and Duration..

```
select movie_id,movie_name,director_name,language from movie_master where
certification='U'
and duration>130
order by movie_id;
```

|||||||||||||||||||Student-Room Details..

```
select s.student_id,student_name,department,DOJ,r.room_id,
room_type from student_details s join admission_details a
on s.student_id=a.student_id join room_details r
on r.room_id=a.room_id order by 1;
```

|||||||||||||||||||2.Pizza-Delivery partner details..

|||||||||||||||||||2.Pizza-Highest Selling Pizza..
SELECT order_date, sum(amount) as Highest_Business

```
FROM pizza
GROUP BY order_date
ORDER BY Highest_Business DESC
LIMIT 1;
/////////////////////////////////////////////////////////////////2. Event Hall-Halls booked more than once ..
select a.hall_name, count(b.hall_id) as no_of_times_booked
from t_hall_details a join t_hall_booking b on a.hall_id = b.hall_id
group by a.hall_name
having length(a.hall_name)>5 and count(b.hall_id)>1
order by a.hall_name desc
;

/////////////////////////////////////////////////////////////////2.Insurance-List of Policies..
select distinct p.policy_name, p.policy_type
from t_policy p, t_member m
where p.policy_id = m.policy_id
and m.member_id >= '1'
order by policy_name, policy_type asc;
```

Airline Flight Management :

1. Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight_Id, From_location, To_Location, Month Name as "Month_Name" and average price as "Average_Price".

Display the records sorted in ascending order based on flight id and then by Month Name.

15 rows

```
select f.flight_id,f.from_location,f.to_location,monthname(fd.flight_departure_date) as Month_name,avg(fd.price) as Average_price from air_flight f join air_flight_details fd on f.flight_id=fd.flight_id group by f.flight_id,Month_name order by f.flight_id,Month_name;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	MONTH_NAME	AVERAGE_PRICE
1011	HYDERABAD	CHENNAI	APRIL	4614.000000
1011	HYDERABAD	CHENNAI	MAY	3855.500000
1262	HYDERABAD	CHENNAI	MAY	3444.500000
1265	CHENNAI	HYDERABAD	APRIL	4086.000000
1265	CHENNAI	HYDERABAD	MAY	3303.666667
289	CHENNAI	KOCHI	MAY	3257.750000
3004	BENGALURU	CHENNAI	MAY	3319.666667
3013	CHENNAI	BENGALURU	MAY	3257.750000
3148	CHENNAI	BENGALURU	JUNE	2773.000000
3148	CHENNAI	BENGALURU	MAY	3052.000000

3241	CHENNAI	KOCHI	MAY	3303.666667
3244	KOCHI	CHENNAI	MAY	3371.500000
3307	BENGALURU	CHENNAI	MAY	3309.000000
916	CHENNAI	HYDERABAD	APRIL	4086.000000
916	CHENNAI	HYDERABAD	MAY	3570.666667

2. Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile_id, customer's first_name, Address and Number of tickets booked as "No_of_Tickets"

Display the records sorted in ascending order based on customer's first name.

1 row

```
select apf.profile_id,apf.first_name,apf.address,count(ati.ticket_id) as No_of_Tickets
from air_passenger_profile apf
join air_ticket_info ati on apf.profile_id=ati.profile_id group by apf.profile_id having
count(ati.ticket_id) <=all
(select count(ati.ticket_id) from air_passenger_profile apf
join air_ticket_info ati on apf.profile_id=ati.profile_id group by apf.profile_id) order by
first_name;
```

PROFILE_ID	FIRST_NAME	ADDRESS	NO_OF_TICKETS
PFL008	GANESH	45 3RD ST, HYDERABAD- 24	1

3. Write a query to display the number of flight services between locations in a month. The Query should display From_Location, To_Location, Month as "Month_Name" and number of flight services as "No_of_Services".

Hint: The Number of Services can be calculated from the number of scheduled departure dates of a flight.

The records should be displayed in ascending order based on From_Location and then by To_Location and then by month name

9 rows

```
select af.from_location,af.to_location,monthname(afd.flight_departure_date)
as Month_Name,
count(afd.flight_departure_date) as No_of_Services from air_flight af join
air_flight_details afd
on af.flight_id=afd.flight_id group by
af.from_location,af.to_location,month_name order by
from_location,to_location,month_name;
```

FROM_LOCATION	TO_LOCATION	MONTH_NAME	NO_OF_SERVICES
BENGALURU	CHENNAI	MAY	7
CHENNAI	BENGALURU	JUNE	1
CHENNAI	BENGALURU	MAY	6
CHENNAI	HYDERABAD	APRIL	2
CHENNAI	HYDERABAD	MAY	6
CHENNAI	KOCHI	MAY	7
HYDERABAD	CHENNAI	APRIL	1
HYDERABAD	CHENNAI	MAY	4
KOCHI	CHENNAI	MAY	2

4. Write a query to display the customer(s) who has/have booked maximum number of tickets in ABC Airlines. The Query should display profile_id, customer's first_name, Address and Number of tickets booked as "No_of_Tickets"

Display the records in ascending order based on customer's first name.

1 row

```
select app.profile_id,app.first_name,app.address,count(ati.ticket_id) as No_of_Tickets  
from air_passenger_profile app  
join air_ticket_info ati on app.profile_id=ati.profile_id join air_flight af on ati.flight_id=af.flight_id  
where af.airline_name= 'ABC Airlines' group by app.profile_id  
having count(ati.ticket_id) >= all (select count(ati.ticket_id) from air_passenger_profile app  
join air_ticket_info ati on app.profile_id=ati.profile_id join air_flight af on ati.flight_id=af.flight_id  
where af.airline_name= 'ABC Airlines' group by app.profile_id) order by app.first_name;
```

PROFILE_ID	FIRST_NAME	ADDRESS	NO_OF_TICKETS
PFL009	RAM	119 2ND CROSS ST,ERNAKULAM-12	8

5. Write a query to display the number of tickets booked from Chennai to Hyderabad. The Query should display passenger profile_id,first_name,last_name, Flight_Id , Departure_Date and number of tickets booked as "No_of_Tickets".

Display the records sorted in ascending order based on profile id and then by flight id and then by departure date.

3 rows

```
select  
ati.profile_id,app.first_name,app.last_name,ati.flight_id,ati.flight_departure_date,count(ati.ticket_id)  
as No_of_Tickets from air_ticket_info ati join air_passenger_profile app on ati.profile_id=app.profile_id join air_flight af on ati.flight_id=af.flight_id  
where af.from_location='chennai' and af.to_location='hyderabad' group by ati.profile_id,  
ati.flight_id,ati.flight_departure_date order by  
ati.profile_id,  
ati.flight_id,ati.flight_departure_date;
```

PROFILE_ID	FIRST_NAME	LAST_NAME	FLIGHT_ID	FLIGHT_DEPARTURE_DATE	NO_OF_TICKETS
PFL001	LATHA	SANKAR	1265	2013-04-29	1

PFL004	AARTHI	RAMESH	1265	2013-05-29	1
PFL005	SIVA	KUMAR	916	2013-05-06	2

6. Write a query to display flight id, from location, to location and ticket price of flights whose departure is in the month of april.

3 rows

Display the records sorted in ascending order based on flight id and then by from location.

```
select af.flight_id,af.from_location,af.to_location,afd.price from
air_flight af
join air_flight_details afd on af.flight_id=afd.flight_id
where monthname(afd.flight_departure_date)='april' order by
flight_id,from_location;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	PRICE
1011	HYDERABAD	CHENNAI	4614.00
1265	CHENNAI	HYDERABAD	4086.00
916	CHENNAI	HYDERABAD	4086.00

7. Write a query to display the average cost of the tickets in each flight on all scheduled dates.

The query should display flight_id, from_location, to_location and Average price as "Price".

Display the records sorted in ascending order based on flight id and then by from_location and then by to_location.

11 rows

```
select af.flight_id,af.from_location,af.to_location,avg(afd.price)
from air_flight af join air_flight_details afd
on af.flight_id=afd.flight_id group by af.flight_id,af.from_location,af.to_location
order by af.flight_id,af.from_location,af.to_location;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	PRICE
1011	HYDERABAD	CHENNAI	4108.333333

1262	HYDERABAD	CHENNAI	3444.500000
1265	CHENNAI	HYDERABAD	3499.250000
289	CHENNAI	KOCHI	3257.750000
3004	BENGALURU	CHENNAI	3319.666667
3013	CHENNAI	BENGALURU	3257.750000
3148	CHENNAI	BENGALURU	2959.000000
3241	CHENNAI	KOCHI	3303.666667
3244	KOCHI	CHENNAI	3371.500000
3307	BENGALURU	CHENNAI	3309.000000
916	CHENNAI	HYDERABAD	3699.500000

8. Write a query to display the customers who have booked tickets from Chennai to Hyderabad. The query should display profile_id, customer_name (combine first_name & last_name with comma in b/w), address of the customer.

Give an alias to the name as customer_name.

Hint: Query should fetch unique customers irrespective of multiple tickets booked.

Display the records sorted in ascending order based on profile id.

3 rows

```
select app.profile_id, concat(app.first_name,',',app.last_name) as customer_name,app.address
from air_passenger_profile app join air_ticket_info ati on app.profile_id=ati.profile_id
join air_flight af on ati.flight_id=af.flight_id where af.from_location='chennai'
and af.to_location='hyderabad' group by app.profile_id order by app.profile_id;
```

PROFILE_ID	CUSTOMER_NAME	ADDRESS

PFL001	LATHA,SANKAR	123 BROAD CROSS ST,CHENNAI-48
PFL004	AARTHI,RAMESH	343 6TH STREET,HYDERABAD- 76
PFL005	SIVA,KUMAR	125 8TH STREET,CHENNAI-46

9. Write a query to display profile id of the passenger(s) who has/have booked maximum number of tickets.

In case of multiple records, display the records sorted in ascending order based on profile id.

2 rows

```
select profile_id from air_ticket_info group by profile_id having
count(ticket_id) >= all (select count(ticket_id)
from air_ticket_info group by profile_id) order by profile_id;
```

PROFILE_ID
PFL002
PFL007

10. Write a query to display the total number of tickets as “No_of_Tickets” booked in each flight in ABC Airlines. The Query should display the flight_id, from_location, to_location and the number of tickets.

Display only the flights in which atleast 1 ticket is booked.

Display the records sorted in ascending order based on flight id.

7 rows

```
select af.flight_id,af.from_location,af.to_location,count(ati.ticket_id) as
No_of_Tickets
from air_flight af join air_ticket_info ati on af.flight_id=ati.flight_id
group by af.flight_id having count(ati.ticket_id) >= 1;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_TICKETS
1011	HYDERABAD	CHENNAI	4
1262	HYDERABAD	CHENNAI	1
1265	CHENNAI	HYDERABAD	2
3004	BENGALURU	CHENNAI	3
3148	CHENNAI	BENGALURU	7
3244	KOCHI	CHENNAI	7
916	CHENNAI	HYDERABAD	2

11. Write a query to display the no of services offered by each flight and the total price of the services. The Query should display flight_id, number of services as "No_of_Services" and the cost as "Total_Price" in the same order.

Order the result by Total Price in descending order and then by flight_id in descending order.

Hint: The number of services can be calculated from the number of scheduled departure dates of the flight

11 rows

```
select af.flight_id, count(afd.flight_departure_date) as No_of_Services, sum(afd.price) as Total_Price from air_flight af join air_flight_details afd on af.flight_id=afd.flight_id group by flight_id order by total_price desc, flight_id desc;
```

FLIGHT_ID	NO_OF_SERVICES	TOTAL_PRICE
916	4	14798.00

1265	4	13997.00
3307	4	13236.00
3013	4	13031.00
289	4	13031.00
1011	3	12325.00
3004	3	9959.00
3241	3	9911.00
3148	3	8877.00
1262	2	6889.00
3244	2	6743.00

12. Write a query to display the number of passengers who have travelled in each flight in each scheduled date. The Query should display flight_id, flight_departure_date and the number of passengers as "No_of_Passengers" in the same order.

Display the records sorted in ascending order based on flight id and then by flight departure date.

9 rows

```
SELECT flight_id,
       flight_departure_date,
       COUNT(ticket_id) AS No_of_Passengers
FROM air_ticket_info
GROUP BY flight_id,
         flight_departure_date
ORDER BY flight_id, flight_departure_date;t
```

FLIGHT_ID	FLIGHT_DEPARTURE_DATE	NO_OF_PASSENGERS
1011	2013-05-09	4
1262	2013-05-20	1
1265	2013-04-29	1
1265	2013-05-29	1
3004	2013-05-02	3
3148	2013-05-21	2
3148	2013-06-01	5
3244	2013-05-03	7
916	2013-05-06	2

13. Write a query to display profile id of passenger(s) who booked minimum number of tickets.

In case of multiple records, display the records sorted in ascending order based on profile id.

1 row

```
select profile_id from air_ticket_info group by profile_id having count(profile_id) <= all
(select count(profile_id) from air_ticket_info group by profile_id) order by profile_id;
```

PROFILE_ID
PFL008

14. Write a query to display unique passenger profile id, first name, mobile number and email

address of passengers who booked ticket to travel from HYDERABAD to CHENNAI.

Display the records sorted in ascending order based on profile id.

4 rows

```
select distinct ati.profile_id,app.first_name,app.mobile_number,app.email_id  
from air_ticket_info  
ati join air_passenger_profile app on ati.profile_id=app.profile_id join air_flight af  
on ati.flight_id=af.flight_id  
where af.from_location='hyderabad' and af.to_location='chennai' order by profile_id;
```

PROFILE_ID	FIRST_NAME	MOBILE_NUMBER	EMAIL_ID
PFL001	LATHA	9876543210	LATHA@GMAIL.COM
PFL004	AARTHI	9595652530	AARTHI@GMAIL.COM
PFL005	SIVA	9884416986	SIVA@GMAIL.COM
PFL008	GANESH	9375237890	GANESH@GMAIL.COM

15. Write a query to intimate the passengers who are boarding Chennai to Hyderabad Flight on 6th May 2013 stating the delay of 1hr in the departure time. The Query should display the passenger's profile_id, first_name, last_name, flight_id, flight_departure_date, actual departure time , actual arrival time , delayed departure time as "Delayed_Departure_Time", delayed arrival time as "Delayed_Arrival_Time" Hint: Distinct Profile ID should be displayed irrespective of multiple tickets booked by the same profile.

Display the records sorted in ascending order based on passenger's profile id.

1 row

```
select distinct app.profile_id,app.first_name,app.last_name,ati.flight_id,ati.flight_departure_date,  
af.departure_time,af.arrival_time, af.departure_time ,ADDTIME(af.departure_time,'1:00:00') as  
Delayed_Departure_Time,  
ADDTIME(af.arrival_time,'1:00:00') as Delayed_Arrival_Time from air_passenger_profile app  
join air_ticket_info ati on app.profile_id=ati.profile_id join air_flight af on  
ati.flight_id=af.flight_id where ati.flight_departure_date='2013-05-06' order by app.profile_id;
```

PROFILE_ID	FIRST_NAME	LAST_NAME	FLIGHT_ID	FLIGHT_DEPARTURE	DEPARTURE_TIME	ARRIVAL_TIME

				_DATE		
PFL005	SIVA	KUMAR	916	2013-05-06	19:55:00	21:00:00

DELAYED_DEPARTURE_TIME	DELAYED_ARRIVAL_TIME
20:55:00	22:00:00

16. Write a query to display the number of tickets as "No_of_Tickets" booked by Kochi Customers. The Query should display the Profile_Id, First_Name, Base_Location and number of tickets booked.

Hint: Use String functions to get the base location of customer from their Address and give alias name as "Base_Location"

Display the records sorted in ascending order based on customer first name.

2 rows

```
select
ap.profile_id, ap.first_name, substring_index(substring_index(ap.address,',',-1),'-',1)
as base_location, count(at.ticket_id) as No_of_Tickets from
air_passenger_profile ap join air_ticket_info at
on at.profile_id=ap.profile_id
where substring_index(substring_index(ap.address,',',-1),'-',1) ='kochi'
group by ap.profile_id order by first_name
```

PROFILE_ID	FIRST_NAME	BASE_LOCATION	NO_OF_TICKETS
PFL003	AMIT	KOCHI	3
PFL006	RAMESH	KOCHI	4

17. Write a query to display the flight_id, from_location, to_location, number of Services as "No_of_Services" offered in the month of May.

Hint:The number of services can be calculated from the number of scheduled departure dates of the flight

Display the records sorted in ascending order based on flight id.

11 rows

```
select af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date)  
as No_of_Services from air_flight af join air_flight_details afd  
on af.flight_id=afd.flight_id where month(afd.flight_departure_date)='05'  
group by flight_id order by flight_id;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_SERVICES
1011	HYDERABAD	CHENNAI	2
1262	HYDERABAD	CHENNAI	2
1265	CHENNAI	HYDERABAD	3
289	CHENNAI	KOCHI	4
3004	BENGALURU	CHENNAI	3
3013	CHENNAI	BENGALURU	4
3148	CHENNAI	BENGALURU	2
3241	CHENNAI	KOCHI	3
3244	KOCHI	CHENNAI	2
3307	BENGALURU	CHENNAI	4
916	CHENNAI	HYDERABAD	3

18. Write a query to display profile id, last name, mobile number and email id of passengers whose base location is chennai.

Display the records sorted in ascending order based on profile id.

2 rows

```
select profile_id, last_name, mobile_number, email_id from air_passenger_profile where
substring_index(substring_index(address, ',', -1), '-', 1)='chennai'
order by profile_id;
```

PROFILE_ID	LAST_NAME	MOBILE_NUMBER	EMAIL_ID
PFL001	SANKAR	9876543210	LATHA@GMAIL.COM
PFL005	KUMAR	9884416986	SIVA@GMAIL.COM

18. Write a query to display number of flights between 6.00 AM and 6.00 PM from chennai. Hint
Use FLIGHT_COUNT as alias name.

1 row

```
select count(flight_id) as FLIGHT_COUNT from air_flight where departure_time between
'6:00:00' and '18:00:00' and from_location='chennai';
```

FLIGHT_COUNT
3

19. Write a query to display unique profile id, first name , email id and contact number of passenger(s) who travelled on flight with id 3148. Display the records sorted in ascending order based on first name.

2 rows

```
select distinct app.profile_id, app.first_name, app.email_id, app.mobile_number from
air_passenger_profile app
join air_ticket_info ati on app.profile_id=ati.profile_id
where ati.flight_id= 3148 group by app.first_name order by app.first_name;
```

PROFILE_ID	FIRST_NAME	EMAIL_ID	MOBILE_NUMBER
PFL002	ARUN	ARUN@AOL.COM	8094564243

PFL007	GAYATHRI	GAYATHRI@GMAIL.COM	8073245678
--------	----------	--------------------	------------

20. Write a query to display the flights available in Morning, AfterNoon, Evening & Night. The Query should display the Flight_Id, From_Location, To_Location , Departure_Time, time of service as "Time_of_Service".

Time of Service should be calculated as: From 05:00:01 Hrs to 12:00:00 Hrs - Morning,
12:00:01 to 18:00:00 Hrs -AfterNoon, 18:00:01 to 24:00:00 - Evening and 00:00:01 to 05:00:00 - Night

Display the records sorted in ascending order based on flight id.

11 rows

```
select flight_id,from_location,to_location,departure_time,
case when departure_time between '05:00:01' and '12:00:00' then 'Morning'
when departure_time between '12:00:01' and '18:00:00' then 'Afternoon'
when departure_time between '18:00:01' and '24:00:00' then 'Evening'
when departure_time between '00:00:01' and '05:00:00' then 'Night'
end as Time_of_Service
from air_flight order by flight_id;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	DEPARTURE_TIME	TIME_OF_SERVICE
1011	HYDERABAD	CHENNAI	12:30:00	AFTERNOON
1262	HYDERABAD	CHENNAI	06:00:00	MORNING
1265	CHENNAI	HYDERABAD	21:25:00	EVENING
289	CHENNAI	KOCHI	08:40:00	MORNING
3004	BENGALURU	CHENNAI	09:05:00	MORNING
3013	CHENNAI	BENGALURU	07:40:00	MORNING
3148	CHENNAI	BENGALURU	20:15:00	EVENING

3241	CHENNAI	KOCHI	10:40:00	MORNING
3244	KOCHI	CHENNAI	21:10:00	EVENING
3307	BENGALURU	CHENNAI	18:45:00	EVENING
916	CHENNAI	HYDERABAD	19:55:00	EVENING

21. Please follow instructions given below.

Write a query to display flight id, departure date, flight type of all flights. Flight type can be identified based on the following rules : if ticket price is less than 3000 then 'AIR PASSENGER', ticket price between 3000 and less than 4000 'AIR BUS' and ticket price between 4000 and greater than 4000 then 'EXECUTIVE PASSENGER'. Hint use FLIGHT_TYPE as alias name.

Display the records sorted in ascending order based on flight_id and then by departure date.

36 rows

```
select flight_id,flight_departure_date,
case when price<3000 then 'AIR PASSENGER'
when price>=3000 and price<=4000 then 'AIR BUS'
when price>4000 then 'EXECUTIVE PASSENGER'
end as FLIGHT_TYPE from air_flight_details order by flight_id,flight_departure_date;
```

FLIGHT_ID	FLIGHT_DEPARTURE_DATE	FLIGHT_TYPE
1011	2013-04-30	EXECUTIVE PASSENGER
1011	2013-05-09	EXECUTIVE PASSENGER
1011	2013-05-21	AIR BUS
1262	2013-05-20	AIR BUS

1262	2013-05-29	AIR BUS
1265	2013-04-29	EXECUTIVE PASSENGER
1265	2013-05-14	AIR BUS
1265	2013-05-18	EXECUTIVE PASSENGER
1265	2013-05-29	AIR PASSENGER
289	2013-05-06	AIR BUS
289	2013-05-08	AIR BUS
289	2013-05-20	AIR BUS
289	2013-05-31	AIR PASSENGER
3004	2013-05-02	AIR BUS
3004	2013-05-19	AIR BUS
3004	2013-05-24	AIR BUS
3013	2013-05-04	AIR BUS
3013	2013-05-06	AIR BUS

3013	2013-05-22	AIR BUS
3013	2013-05-30	AIR PASSENGER
3148	2013-05-16	AIR BUS
3148	2013-05-21	AIR BUS
3148	2013-06-01	AIR PASSENGER
3241	2013-05-01	EXECUTIVE PASSENGER
3241	2013-05-13	AIR BUS
3241	2013-05-27	AIR PASSENGER
3244	2013-05-03	AIR BUS
3244	2013-05-15	AIR BUS
3307	2013-05-03	AIR BUS
3307	2013-05-03	AIR BUS
3307	2013-05-23	AIR BUS
3307	2013-05-29	AIR BUS

916	2013-04-28	EXECUTIVE PASSENGER
916	2013-05-01	EXECUTIVE PASSENGER
916	2013-05-06	AIR BUS
916	2013-05-12	AIR BUS

22. Please follow instructions given below.

Write a query to display the credit card type and no of credit cards used on the same type.

Display the records sorted in ascending order based on credit card type.

Hint: Use CARD_COUNT AS Alias name for no of cards.

3 rows

SELECT CARD_TYPE, count(card_type) CARD_COUNT FROM air_credit_card_details group by CARD_TYPE order by CARD_TYPE;

CARD_TYPE	CARD_COUNT
GOLD	3
INSTANT	2
PLATINIUM	3

23. Please follow instructions given below.

Write a Query to display serial no, first name, mobile number, email id of all the passengers who holds email address from gmail.com.

The Serial No will be the last three digits of profile ID.

Hint: Use SERIAL_NO as Alias name for serial number.

Display the records sorted in ascending order based on name.

6 rows

```
select substring(profile_id,4) as SERIAL_NO,first_name,mobile_number,email_id
from air_passenger_profile where email_id like '%gmail.com' order by first_name;
```

SERIAL_NO	FIRST_NAME	MOBILE_NUMBER	EMAIL_ID
004	AARTHI	9595652530	AARTHI@GMAIL.COM
008	GANESH	9375237890	GANESH@GMAIL.COM
007	GAYATHRI	8073245678	GAYATHRI@GMAIL.COM
001	LATHA	9876543210	LATHA@GMAIL.COM
006	RAMESH	9432198760	RAMESH@GMAIL.COM
005	SIVA	9884416986	SIVA@GMAIL.COM

24. Please follow instructions given below.

Write a query to display the flight(s) which has least number of services in the month of May.
The Query should fetch flight_id, from_location, to_location, least number of Services as "No_of_Services" Hint: Number of services offered can be calculated from the number of scheduled departure dates of a flight

If there are multiple flights, display them sorted in ascending order based on flight id.

4 rows

```
select af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date) as
No_of_Services from air_flight af join air_flight_details afd on
af.flight_id=afd.flight_id where month(afd.flight_departure_date)='05' group by af.flight_id
having count(afd.flight_departure_date)
<= all (select count(afd.flight_departure_date) from air_flight af join air_flight_details afd on
af.flight_id=afd.flight_id where month(afd.flight_departure_date)='05' group by af.flight_id)
order by af.flight_id;
```

LIGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_SERVICES
1011	HYDERABAD	CHENNAI	2

1262	HYDERABAD	CHENNAI	2
3148	CHENNAI	BENGALURU	2
3244	KOCHI	CHENNAI	2

25. Please follow instructions given below.

Write a query to display the number of flights flying from each location. The Query should display the from location and the number of flights to other locations as "No_of_Flights".

Hint: Get the distinct from location and to location.

Display the records sorted in ascending order based on from location.

4 rows

```
select distinct from_location, count(to_location) as No_of_Flights from
air_flight
group by from_location order by from_location;
```

FROM_LOCATION	NO_OF_FLIGHTS
BENGALURU	2
CHENNAI	6
HYDERABAD	2
KOCHI	1

26. Please follow instructions given below.

Write a query to display the number of passengers traveled in each flight in each scheduled date. The Query should display flight_id, from_location, To_location, flight_departure_date and the number of passengers as "No_of_Passengers".

Hint: The Number of passengers inclusive of all the tickets booked with single profile id.

Display the records sorted in ascending order based on flight id and then by flight departure date.

9 rows

```
select af.flight_id,af.from_location,af.to_location,ati.flight_departure_date,count(ati.ticket_id)
as No_of_Passengers from air_flight af join air_ticket_info ati on af.flight_id=ati.flight_id
group by af.flight_id,ati.flight_departure_date order by af.flight_id,ati.flight_departure_date;
```

flight_id	from_location	to_location	flight_departure_date	No_of_Passengers
1011	HYDERABAD	CHENNAI	2013-05-09	3
1262	HYDERABAD	CHENNAI	2013-05-20	1
1265	CHENNAI	HYDERABAD	2013-04-29	1
1265	CHENNAI	HYDERABAD	2013-05-29	1
3004	BENGALURU	CHENNAI	2013-05-02	3
3148	CHENNAI	BENGALURU	2013-05-21	1
3148	CHENNAI	BENGALURU	2013-06-01	3
3244	KOCHI	CHENNAI	2013-05-03	7
916	CHENNAI	HYDERABAD	2013-05-06	2

27. Please follow instructions given below.

Write a query to display the flight details in which more than 10% of seats have been booked.

The query should display Flight_Id, From_Location, To_Location, Total_Seats, seats booked as "No_of_Seats_Booked".

Display the records sorted in ascending order based on flight id and then by

No_of_Seats_Booked.

1 row

```
select af.flight_id,af.from_location,af.to_location,af.total_seats,(af.total_seats-
afd.available_seats)
as No_of_Seats_Booked from air_flight af join air_flight_details afd on af.flight_id=
afd.flight_id where (af.total_seats-afd.available_seats)>(af.total_seats*0.1) group by flight_id
order by
flight_id,No_of_Seats_Booked;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	TOTAL_SEATS	NO_OF_SEATS_BOOKED
3244	KOCHI	CHENNAI	50	7

28. Please follow instructions given below.

Write a query to display the Flight_Id, Flight_Departure_Date, From_Location, To_Location and Duration of all flights which has duration of travel less than 1 Hour, 10 Minutes.

Display the records sorted in ascending order based on flight id and then by flight departure date.

14 rows

```
select af.flight_id,afd.flight_departure_date,af.from_location,af.to_location,af.duration  
from air_flight af join air_flight_details afd on af.flight_id=afd.flight_id  
where duration<'1:10:00' group by af.flight_id,afd.flight_departure_date  
order by af.flight_id,afd.flight_departure_date;
```

FLIGHT_ID	FLIGHT_DEPARTURE_DATE	FROM_LOCATION	TO_LOCATION	DURATION
3013	2013-05-04	CHENNAI	BENGALURU	01:05:00
3013	2013-05-06	CHENNAI	BENGALURU	01:05:00
3013	2013-05-22	CHENNAI	BENGALURU	01:05:00
3013	2013-05-30	CHENNAI	BENGALURU	01:05:00
3148	2013-05-16	CHENNAI	BENGALURU	01:05:00
3148	2013-05-21	CHENNAI	BENGALURU	01:05:00
3148	2013-06-01	CHENNAI	BENGALURU	01:05:00
3307	2013-05-03	BENGALURU	CHENNAI	01:00:00
3307	2013-05-23	BENGALURU	CHENNAI	01:00:00
3307	2013-05-29	BENGALURU	CHENNAI	01:00:00
916	2013-04-28	CHENNAI	HYDERABAD	01:05:00
916	2013-05-01	CHENNAI	HYDERABAD	01:05:00

916	2013-05-06	CHENNAI	HYDERABAD	01:05:00
916	2013-05-12	CHENNAI	HYDERABAD	01:05:00

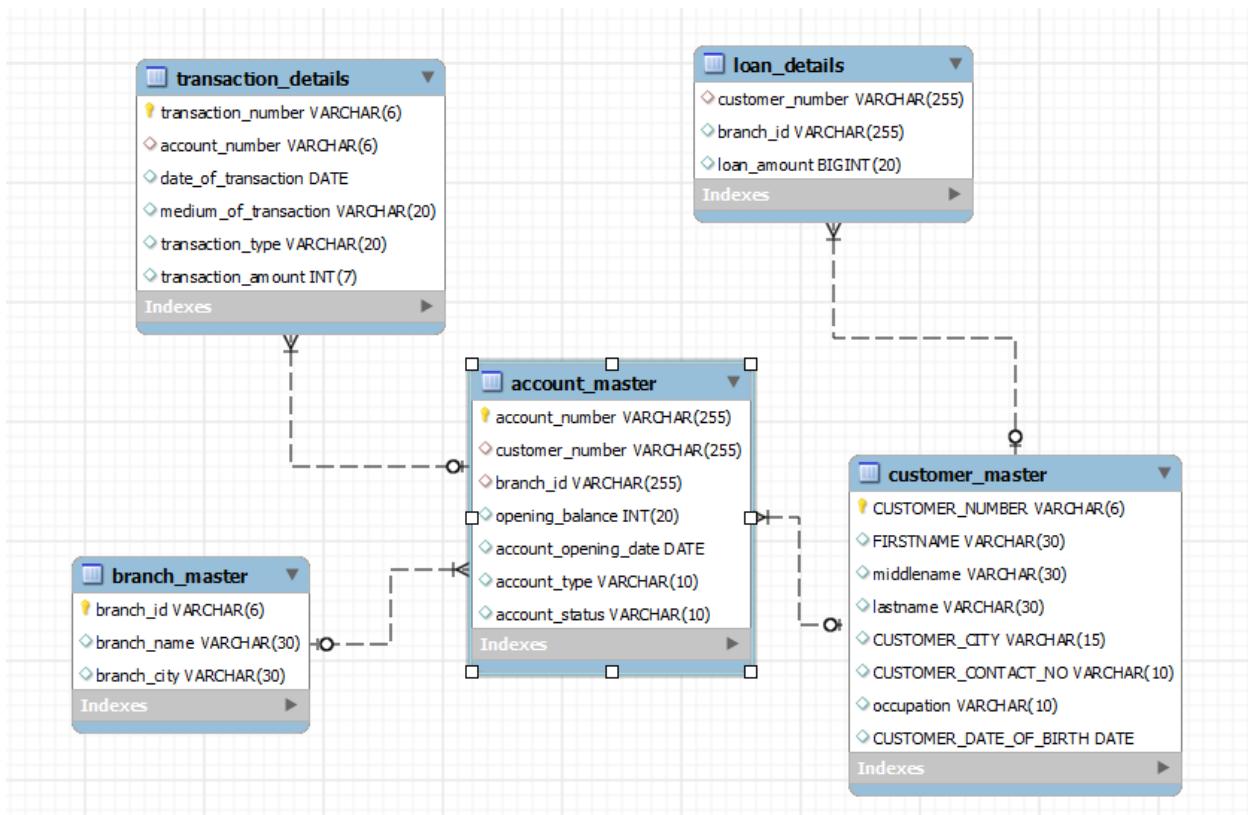
29. Please follow instructions given below.

Write a query to display the flight_id, from_location,to_location,number of services as "No_of_Services", average ticket price as "Average_Price" whose average ticket price is greater than the total average ticket cost of all flights. Order the result by lowest average price.

4 rows

```
select af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date) as No_of_Services,
avg(afd.price) as Average_Price from air_flight af join air_flight_details afd
on af.flight_id=afd.flight_id group by af.flight_id having avg(afd.price)>
(select avg(afd.price) from air_flight_details afd) order by afd.price;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_SERVICES	AVERAGE
1262	HYDERABAD	CHENNAI	2	3444.500
1265	CHENNAI	HYDERABAD	4	3499.250
916	CHENNAI	HYDERABAD	4	3699.500
1011	HYDERABAD	CHENNAI	3	4108.333



Bank Management System Queries:

1. Please follow instructions given below.

Write a query to display account number, customer's number, customer's firstname, lastname, account opening date.

Display the records sorted in ascending order based on account number.

SELECT

```
account_number, am.customer_number, firstname, lastname, account_opening_date  
FROM customer_master cm INNER JOIN account_master am  
ON cm.customer_number=am.customer_number  
ORDER BY account_number;
```

2. Please follow instructions given below.

Write a query to display the number of customer's from Delhi. Give the count an alias name of Cust_Count.

```
SELECT count(customer_number) Cust_Count  
FROM customer_master  
WHERE customer_city='Delhi'
```

3. Please follow instructions given below.

Write a query to display the customer number, customer firstname, account number for the customer's whose accounts were created after 15th of any month.

Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT am.customer_number, firstname, account_number  
FROM customer_master cm INNER JOIN account_master am  
ON cm.customer_number=am.customer_number  
WHERE extract(day from account_opening_date)>15  
ORDER BY am.customer_number, account_number
```

4. Please follow instructions given below.

Write a query to display customer number, customer's first name, account number where the account status is terminated.

Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT am.customer_number, firstname, account_number  
FROM customer_master cm INNER JOIN account_master am  
ON cm.customer_number=am.customer_number  
WHERE account_status='Terminated'  
ORDER BY am.customer_number, account_number
```

5.Pleas follow instructions given below.

Write a query to display the total number of withdrawals and total number of deposits being done by customer whose customer number ends with 001. The query should display transaction type and the number of transactions. Give an alias name as Trans_Count for number of transactions.

Display the records sorted in ascending order based on transaction type.

```
SELECT transaction_type, count(transaction_number) Trans_Count  
FROM account_master am INNER JOIN transaction_details td  
ON am.account_number=td.account_number  
WHERE customer_number like '%001'  
GROUP BY transaction_type  
ORDER BY transaction_type
```

6.Pleas follow instructions given below.

Write a query to display the number of customers who have registration but no account in the bank.

Give the alias name as Count_Customer for number of customers.

```
SELECT count(customer_number) Count_Customer  
FROM customer_master  
WHERE customer_number NOT IN (SELECT customer_number FROM account_master)
```

7.Pleas follow instructions given below.

Write a query to display account number and total amount deposited by each account holder (Including the opening balance). Give the total amount deposited an alias name of Deposit_Amount. Display the records in sorted order based on account number.

```
SELECT td.account_number, opening_balance+sum(transaction_amount)  
Deposit_Amount  
FROM account_master am INNER JOIN transaction_details td  
ON am.account_number=td.account_number  
WHERE transaction_type='deposit'  
GROUP BY account_number  
ORDER BY account_number
```

8.Pleas follow instructions given below.

Write a query to display the number of accounts opened in each city .The Query should display Branch City and number of accounts as No_of_Accounts.For the branch city where we don't have any accounts opened display 0. Display the records in sorted order based on branch city.

```
select branch_master.branch_city, count(account_master.account_number) as  
No_of_Accounts from branch_master left join account_master on  
account_master.branch_id=branch_master.branch_id  
group by branch_master.branch_city order by branch_city;
```

9. Please follow instructions given below.

Write a query to display the firstname of the customers who have more than 1 account.
Display the records in sorted order based on firstname.

```
select firstname  
FROM customer_master cm INNER JOIN account_master am  
ON cm.customer_number=am.customer_number  
group by firstname  
having count(account_number)>1  
order by firstname;
```

10. Please follow instructions given below.

Write a query to display the customer number, customer firstname, customer lastname
who has taken loan from more than 1 branch.
Display the records sorted in order based on customer number.

```
SELECT Id.customer_number, firstname, lastname  
FROM customer_master cm INNER JOIN loan_details Id  
ON cm.customer_number=Id.customer_number  
GROUP BY customer_number  
HAVING count(branch_id)>1  
ORDER BY customer_number
```

11. Please follow instructions given below.

Write a query to display the customer's number, customer's firstname, customer's city
and branch city where the city of the customer and city of the branch is different.
Display the records sorted in ascending order based on customer number.

```
select customer_master.customer_number, firstname, customer_city, branch_city  
from account_master inner join customer_master on account_master.customer_number  
= customer_master.customer_number  
inner join branch_master on account_master.branch_id = branch_master.branch_id  
where customer_city != branch_city order by customer_master.customer_number;
```

12. Please follow instructions given below.

Write a query to display the number of clients who have asked for loans but they don't
have any account in the bank though they are registered customers. Give the count an
alias name of Count.

```
SELECT count(Id.customer_number) Count
FROM customer_master cm INNER JOIN loan_details Id
ON cm.customer_number=Id.customer_number
WHERE cm.customer_number NOT IN ( SELECT customer_number FROM
account_master)
```

13. Please follow instructions given below.

Write a query to display the account number who has done the highest transaction. For example the account A00023 has done 5 transactions i.e. suppose 3 withdrawal and 2 deposits. Whereas the account A00024 has done 3 transactions i.e. suppose 2 withdrawals and 1 deposit. So account number of A00023 should be displayed. In case of multiple records, display the records sorted in ascending order based on account number.

```
SELECT td.account_number
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by td.account_number
having count(td.transaction_number)>=ALL
(SELECT count(td.transaction_number)
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by td.account_number) order by am.account_number;
```

14. Please follow instructions given below.

Write a query to show the branch name, branch city where we have the maximum customers.

For example the branch B00019 has 3 customers, B00020 has 7 and B00021 has 10. So branch id B00021 is having maximum customers. If B00021 is Koramangla branch Bangalore, Koramangla branch should be displayed along with city name Bangalore. In case of multiple records, display the records sorted in ascending order based on branch name.

```
select branch_name,branch_city
FROM branch_master INNER JOIN account_master
ON branch_master.branch_id=account_master.branch_id
group by branch_name
having count(customer_number)>=ALL
(select count(customer_number)
FROM branch_master INNER JOIN account_master
ON branch_master.branch_id=account_master.branch_id
group by branch_name) order by branch_name;
```

15.Please follow instructions given below.

Write a query to display all those account number, deposit, withdrawal where withdrawal is more than deposit amount. Hint: Deposit should include opening balance as well.

For example A00011 account opened with Opening Balance 1000 and A00011 deposited 2000 rupees on 2012-12-01 and 3000 rupees on 2012-12-02. The same account i.e A00011 withdrawn 3000 rupees on 2013-01-01 and 7000 rupees on 2013-01-03. So the total deposited amount is 6000 and total withdrawal amount is 10000. So withdrawal amount is more than deposited amount for account number A00011.

Display the records sorted in ascending order based on account number.

```
SELECT td.account_number,sum(CASE WHEN transaction_type='Deposit' THEN  
transaction_amount END)  
+(SELECT opening_balance FROM account_master am2 where  
am2.account_number=am.account_number) Deposit,  
sum(CASE WHEN transaction_type='Withdrawal' THEN transaction_amount END)  
Withdrawal  
FROM account_master am INNER JOIN transaction_details td  
ON am.account_number=td.account_number  
GROUP BY td.account_number  
HAVING Withdrawal > Deposit  
ORDER BY am.account_number
```

16.Please follow instructions given below.

Write a query to show the balance amount for account number that ends with 001.

Note: Balance amount includes account opening balance also. Give alias name as Balance_Amount.

For example A00015 is having an opening balance of 1000. A00015 has deposited 2000 on 2012-06-12 and deposited 3000 on 2012-07-13. The same account has drawn money of 500 on 2012-08-12 , 500 on 2012-09-15, 1000 on 2012-12-17. So balance amount is 4000 i.e $(1000 \text{ (opening balance)} + 2000 + 3000) - (500 + 500 + 1000)$.

```
SELECT (SUM(CASE WHEN transaction_type='Deposit'  
THEN transaction_amount END)) -  
(SUM(CASE WHEN transaction_type='Withdrawal'  
THEN transaction_amount END))+(select opening_balance  
from account_master where account_number like '%001') AS Balance_Amount  
FROM transaction_details where account_number like '%001'
```

17.Please follow instructions given below.

Display the customer number, customer's first name, account number and number of transactions being made by the customers from each account. Give the alias name for number of transactions as Count_Trans. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT cm.customer_number,firstname,  
am.account_number,count(transaction_number) Count_Trans  
FROM customer_master cm inner JOIN account_master am  
ON cm.customer_number=am.customer_number  
INNER JOIN transaction_details td  
ON am.account_number=td.account_number  
group by am.account_number order by cm.customer_number, am.account_number
```

18.Please follow instructions given below.

Write a query to display the customer's firstname who have multiple accounts (atleast 2 accounts). Display the records sorted in ascending order based on customer's firstname.

```
SELECT firstname  
FROM customer_master INNER JOIN account_master  
ON customer_master.customer_number=account_master.customer_number  
GROUP BY firstname  
having count(firstname)>=2 order by firstname;
```

19.Please follow instructions given below.

Write a query to display the customer number, firstname, lastname for those client where total loan amount taken is maximum and at least taken from 2 branches.

For example the customer C00012 took a loan of 100000 from bank branch with id B00009 and C00012

Took a loan of 500000 from bank branch with id B00010. So total loan amount for customer C00012 is

600000. C00013 took a loan of 100000 from bank branch B00009 and 200000 from bank branch B00011.

So total loan taken is 300000. So loan taken by C00012 is more then C00013.

```
SELECT Id.customer_number,firstname, lastname  
FROM customer_master cm INNER JOIN loan_details Id  
ON cm.customer_number=Id.customer_number  
group by customer_number  
having count(branch_id)>=2 and sum(loan_amount)>=All(select sum(loan_amount) from  
loan_details group by customer_number)
```

20.Please follow instructions given below.

Write a query to display the customer's number, customer's firstname, branch id and loan amount for people who have taken loans..

Display the records sorted in ascending order based on customer number and then by branch id and then by loan amount.

```
SELECT Id.customer_number,firstname,branch_id,loan_amount  
FROM customer_master cm INNER JOIN loan_details Id  
ON cm.customer_number=Id.customer_number order by cm.customer_number,  
branch_id,loan_amount
```

21.Please follow instructions given below.

Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count_Branch.

Display the records sorted in ascending order based on city name.

```
SELECT branch_city, count(branch_id) Count_Branch  
FROM branch_master  
GROUP BY branch_city  
ORDER BY branch_city
```

22.Please follow instructions given below.

Write a query to display account id, customer's firstname, customer's lastname for the customer's whose account is Active.

Display the records sorted in ascending order based on account id /account number.

```
SELECT account_number,firstname,lastname  
FROM customer_master cm INNER JOIN account_master am  
ON cm.customer_number=am.customer_number  
WHERE account_status='Active'  
ORDER BY account_number
```

23.Please follow instructions given below.

Write a query to display customer's number, first name and middle name. For the customers who don't have middle name, display their last name as middle name. Give the alias name as Middle_Name.

Display the records sorted in ascending order based on customer number.

```
SELECT customer_number,firstname,coalesce(middlename,lastname) Middle_Name  
FROM customer_master order by customer_number
```

24.Please follow instructions given below.

Write a query to display the customer number , firstname, customer's date of birth . Display the records sorted in ascending order of date of birth year and within that sort by firstname in ascending order.

```
SELECT customer_number,firstname,customer_date_of_birth  
FROM customer_master order by year(customer_date_of_birth),firstname;
```

25. Please follow instructions given below.

Write a query to display the customers firstname, city and account number whose occupation are not into Business, Service or Student.

Display the records sorted in ascending order based on customer first name and then by account number.

```
SELECT firstname, customer_city, account_number  
FROM customer_master cm INNER JOIN account_master am  
ON cm.customer_number=am.customer_number  
WHERE occupation !='Service' and occupation != 'Student' and occupation != 'Business'  
order by firstname, account_number
```

Bank Management System Queries:

1. Please follow instructions given below.

Write a query to display account number, customer's number, customer's firstname, lastname, account opening date.

Display the records sorted in ascending order based on account number.

SELECT

```
account_number, am.customer_number, firstname, lastname, account_opening_date  
FROM customer_master cm JOIN account_master am  
ON cm.customer_number=am.customer_number  
ORDER BY account_number;
```

ACCOUNT_NUMBER	CUSTOMER_NUMBER	FIRSTNAME	LASTNAME	ACCOUNT_OPENING_DATE
A00001	C00001	RAMESH	SHARMA	2012-12-15
A00002	C00002	AVINASH	MINHA	2012-06-12
A00003	C00003	RAHUL	RASTOGI	2012-05-17
A00004	C00002	AVINASH	MINHA	2013-01-27
A00005	C00006	CHITRESH	BARWE	2012-12-17
A00006	C00007	AMIT	BORKAR	2010-08-12
A00007	C00007	AMIT	BORKAR	2012-10-02
A00008	C00001	RAMESH	SHARMA	2009-11-09
A00009	C00003	RAHUL	RASTOGI	2008-11-30
A00010	C00004	PARUL	GANDHI	2013-03-01

2. Please follow instructions given below.

Write a query to display the number of customer's from Delhi. Give the count an alias name of Cust_Count.

```
SELECT count(customer_number) Cust_Count
```

```
FROM customer_master
```

```
WHERE customer_city='Delhi'
```

CUST_COUNT
4

3. Please follow instructions given below.

Write a query to display the customer number, customer firstname, account number for the customer's whose accounts were created after 15th of any month.

Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT am.customer_number, firstname, account_number  
FROM customer_master cm JOIN account_master am  
ON cm.customer_number=am.customer_number  
WHERE day(account_opening_date)>15  
ORDER BY am.customer_number, account_number
```

CUSTOMER_NUMBER	FIRSTNAMEMiddle_Name	ACCOUNT_NUMBER
C00002	AVINASH	A00004
C00003	RAHUL	A00003
C00003	RAHUL	A00009
C00006	CHITRESH	A00005

4. Please follow instructions given below.

Write a query to display customer number, customer's first name, account number where the account status is terminated.

Display the records sorted in ascending order based on customer number and then by account number.

```

SELECT am.customer_number,firstname, account_number
FROM customer_master cm JOIN account_master am
ON cm.customer_number=am.customer_number
WHERE account_status='Terminated'
ORDER BY am.customer_number, account_number

```

CUSTOMER_NUMBER	FIRSTNAME	ACCOUNT_NUMBER
C00001	RAMESH	A00008
C00003	RAHUL	A00009

5. Please follow instructions given below.

Write a query to display the total number of withdrawals and total number of deposits being done by customer whose customer number ends with 001. The query should display transaction type and the number of transactions. Give an alias name as Trans_Count for number of transactions.

Display the records sorted in ascending order based on transaction type.

```

SELECT transaction_type, count(transaction_number) Trans_Count
FROM account_master am JOIN transaction_details td
ON am.account_number=td.account_number
WHERE customer_number like '%001'
GROUP BY transaction_type
ORDER BY transaction_type

```

TRANSACTION_TYPE	TRANS_COUNT
DEPOSIT	3
WITHDRAWAL	3

6. Please follow instructions given below.

Write a query to display the number of customers who have registration but no account in the bank.

Give the alias name as Count_Customer for number of customers.

```

SELECT count(customer_number) Count_Customer
FROM customer_master
WHERE customer_number NOT IN (SELECT customer_number FROM account_master)

```

COUNT_CUSTOMER
4

7. Please follow instructions given below.

Write a query to display account number and total amount deposited by each account holder (Including the opening balance). Give the total amount deposited an alias name of Deposit_Amount. Display the records in sorted order based on account number.

```

SELECT td.account_number, opening_balance+sum(transaction_amount)
Deposit_Amount
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
WHERE transaction_type='deposit'
GROUP BY account_number
ORDER BY account_number

```

ACCOUNT_NUMBER	DEPOSIT_AMOUNT
A00001	10000
A00002	6000
A00007	17000

8. Please follow instructions given below.

Write a query to display the number of accounts opened in each city .The Query should display Branch City and number of accounts as No_of_Accounts.For the branch city where we don't have any accounts opened display 0. Display the records in sorted order based on branch city.

```

select branch_master.branch_city, count(account_master.account_number) as
No_of_Accounts from branch_master left join account_master on
account_master.branch_id=branch_master.branch_id
group by branch_master.branch_city order by branch_city;

```

BRANCH_CITY	NO_OF_ACCOUNTS
CHENNAI	0
DELHI	6
KOLKATA	0
MUMBAI	4

9. Please follow instructions given below.

Write a query to display the firstname of the customers who have more than 1 account.
Display the records in sorted order based on firstname.

```

select firstname
FROM customer_master cm INNER JOIN account_master am
ON cm.customer_number=am.customer_number
group by firstname
having count(account_number)>1
order by firstname;

```

FIRSTNAME
AMIT
AVINASH
RAHUL
RAMESH

10. Please follow instructions given below.

Write a query to display the customer number, customer firstname, customer lastname who has taken loan from more than 1 branch.

Display the records sorted in order based on customer number.

```
SELECT Id.customer_number, firstname, lastname  
FROM customer_master cm INNER JOIN loan_details Id  
ON cm.customer_number=Id.customer_number  
GROUP BY customer_number  
HAVING count(branch_id)>1  
ORDER BY customer_number
```

CUSTOMER_NUMBER	FIRSTNAME	LASTNAME
C00001	RAMESH	SHARMA
C00002	AVINASH	MINHA

11. Please follow instructions given below.

Write a query to display the customer's number, customer's firstname, customer's city and branch city where the city of the customer and city of the branch is different.

Display the records sorted in ascending order based on customer number.

```
select customer_master.customer_number, firstname, customer_city, branch_city  
from account_master inner join customer_master on account_master.customer_number  
= customer_master.customer_number  
inner join branch_master on account_master.branch_id = branch_master.branch_id  
where customer_city != branch_city order by customer_master.customer_number;
```

CUSTOMER_NUMBER	FIRSTNAME	CUSTOMER_CITY	BRANCH_CITY
C00002	AVINASH	DELHI	MUMBAI
C00003	RAHUL	DELHI	MUMBAI
C00007	AMIT	MUMBAI	DELHI

12. Please follow instructions given below.

Write a query to display the number of clients who have asked for loans but they don't have any account in the bank though they are registered customers. Give the count an alias name of Count.

```
SELECT count(Id.customer_number) Count
FROM customer_master cm INNER JOIN loan_details Id
ON cm.customer_number=Id.customer_number
WHERE cm.customer_number NOT IN ( SELECT customer_number FROM
account_master)
(Or)
select count(customer_number) as Count from customer_master where
customer_number not in
(select customer_number from account_master) and customer_number in
(select customer_number from loan_details);
```

COUNT
2

13. Please follow instructions given below.

**Write a query to display the account number who has done the highest transaction.
For example the account A00023 has done 5 transactions i.e. suppose 3 withdrawal and 2 deposits. Whereas the account A00024 has done 3 transactions i.e. suppose 2 withdrawals and 1 deposit. So account number of A00023 should be displayed.
In case of multiple records, display the records sorted in ascending order based on account number.**

```
SELECT td.account_number
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by td.account_number
having count(td.transaction_number)>=ALL
(SELECT count(td.transaction_number)
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by td.account_number) order by am.account_number;
```

ACCOUNT_NUMBER
A00001

14. Please follow instructions given below.

Write a query to show the branch name, branch city where we have the maximum customers.

For example the branch B00019 has 3 customers, B00020 has 7 and B00021 has 10. So branch id B00021 is having maximum customers. If B00021 is Koramangla branch Bangalore, Koramangla branch should be displayed along with city name Bangalore. In case of multiple records, display the records sorted in ascending order based on branch name.

```
select branch_name,branch_city
FROM branch_master INNER JOIN account_master
ON branch_master.branch_id=account_master.branch_id
group by branch_name
having count(customer_number)>=ALL
(select count(customer_number)
FROM branch_master INNER JOIN account_master
ON branch_master.branch_id=account_master.branch_id
group by branch_name) order by branch_name;
```

BRANCH_NAME	BRANCH_CITY
ASAF ALI ROAD	DELHI

15. Please follow instructions given below.

Write a query to display all those account number, deposit, withdrawal where withdrawal is more than deposit amount. Hint: Deposit should include opening balance as well.

For example A00011 account opened with Opening Balance 1000 and A00011 deposited 2000 rupees on 2012-12-01 and 3000 rupees on 2012-12-02. The same account i.e A00011 withdrawn 3000 rupees on 2013-01-01 and 7000 rupees on 2013-01-03. So the total deposited amount is 6000 and total withdrawal amount is 10000. So withdrawal amount is more than deposited amount for account number A00011.

Display the records sorted in ascending order based on account number.

```
select am.account_number,opening_balance+sum(case when transaction_type='Deposit'
then
transaction_amount end) as Deposit,sum(case when transaction_type='withdrawal' then
transaction_amount end) as Withdrawal from account_master am join
transaction_details td
on am.account_number=td.account_number group by am.account_number having
Withdrawal>Deposit;
```

ACCOUNT_NUMBER	DEPOSIT	WITHDRAWAL
A00001	10000	12000
A00002	6000	7000

16. Please follow instructions given below.

Write a query to show the balance amount for account number that ends with 001.

Note: Balance amount includes account opening balance also. Give alias name as Balance_Amount.

For example A00015 is having an opening balance of 1000. A00015 has deposited 2000 on 2012-06-12 and deposited 3000 on 2012-07-13. The same account has drawn money of 500 on 2012-08-12 , 500 on 2012-09-15, 1000 on 2012-12-17. So balance amount is 4000 i.e (1000 (opening balance)+2000+3000) – (500+500+1000).

```
SELECT (SUM(CASE WHEN transaction_type='Deposit'
THEN transaction_amount END)) -
(SUM(CASE WHEN transaction_type='Withdrawal'
THEN transaction_amount END))+(select opening_balance
from account_master where account_number like '%001') AS Balance_Amount
FROM transaction_details where account_number like '%001'
```

BALANCE_AMOUNT
-2000

17. Please follow instructions given below.

Display the customer number, customer's first name, account number and number of transactions being made by the customers from each account. Give the alias name for number of transactions as Count_Trans. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT cm.customer_number,firstname,
am.account_number,count(transaction_number) Count_Trans
FROM customer_master cm inner JOIN account_master am
ON cm.customer_number=am.customer_number
INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by am.account_number order by cm.customer_number, am.account_number
```

CUSTOMER_NUMBER	FIRSTNAME	ACCOUNT_NUMBER	COUNT_TRANS
C00001	RAMESH	A00001	6
C00002	AVINASH	A00002	3
C00007	AMIT	A00007	3

18. Please follow instructions given below.

Write a query to display the customer's firstname who have multiple accounts (atleast 2 accounts). Display the records sorted in ascending order based on customer's firstname.

```
SELECT firstname
FROM customer_master INNER JOIN account_master
ON customer_master.customer_number=account_master.customer_number
GROUP BY firstname
having count(firstname)>=2 order by firstname;
```

FIRSTNAME
AMIT
AVINASH
RAHUL
RAMESH

19. Please follow instructions given below.

Write a query to display the customer number, firstname, lastname for those client where total loan amount taken is maximum and at least taken from 2 branches.

For example the customer C00012 took a loan of 100000 from bank branch with id B00009 and C00012

Took a loan of 500000 from bank branch with id B00010. So total loan amount for customer C00012 is 600000. C00013 took a loan of 100000 from bank branch B00009 and 200000 from bank branch B00011.

So total loan taken is 300000. So loan taken by C00012 is more than C00013.

```
SELECT Id.customer_number, firstname, lastname
FROM customer_master cm INNER JOIN loan_details ld
ON cm.customer_number=ld.customer_number
group by customer_number
having count(branch_id)>=2 and sum(loan_amount)>=(select sum(loan_amount) from
loan_details group by customer_number)
```

CUSTOMER_NUMBER	FIRSTNAME	LASTNAME
C00002	AVINASH	MINHA

20. Please follow instructions given below.

Write a query to display the customer's number, customer's firstname, branch id and loan amount for people who have taken loans..

Display the records sorted in ascending order based on customer number and then by branch id and then by loan amount.

```
SELECT Id.customer_number, firstname, branch_id, loan_amount
FROM customer_master cm INNER JOIN loan_details ld
ON cm.customer_number=ld.customer_number order by cm.customer_number,
branch_id, loan_amount
```

CUSTOMER_NUMBER	FIRSTNAME	BRANCH_ID	LOAN_AMOUNT
C00001	RAMESH	B00001	100000
C00001	RAMESH	B00003	600000
C00002	AVINASH	B00001	600000
C00002	AVINASH	B00002	200000

C00009	ABHISHEK	B00008	400000
C00010	SHANKAR	B00009	500000

21.Please follow instructions given below.

Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count_Branch.

Display the records sorted in ascending order based on city name.

```
SELECT branch_city, count(branch_id) Count_Branch
FROM branch_master
GROUP BY branch_city
ORDER BY branch_city
```

BRANCH_CITY	COUNT_BRANCH
CHENNAI	1
DELHI	4
KOLKATA	1
MUMBAI	3

22.Please follow instructions given below.

Write a query to display account id, customer's firstname, customer's lastname for the customer's whose account is Active.

Display the records sorted in ascending order based on account id /account number.

```
SELECT account_number, firstname, lastname
FROM customer_master cm INNER JOIN account_master am
ON cm.customer_number=am.customer_number
WHERE account_status='Active'
ORDER BY account_number
```

ACCOUNT_NUMBER	FIRSTNAME	LASTNAME

A00001	RAMESH	SHARMA
A00002	AVINASH	MINHA
A00003	RAHUL	RASTOGI
A00004	AVINASH	MINHA
A00005	CHITRESH	BARWE
A00007	AMIT	BORKAR
A00010	PARUL	GANDHI

23. Please follow instructions given below.

Write a query to display customer's number, first name and middle name. For the customers who don't have middle name, display their last name as middle name. Give the alias name as Middle_Name.

Display the records sorted in ascending order based on customer number.

```
SELECT customer_number,firstname,coalesce(middlename,lastname) Middle_Name
FROM customer_master order by customer_number
```

CUSTOMER_NUMBER	FIRSTNAME	MIDDLE_NAME
C00001	RAMESH	CHANDRA
C00002	AVINASH	SUNDER
C00003	RAHUL	RASTOGI
C00004	PARUL	GANDHI

C00005	NAVEEN	CHANDRA
C00006	CHITRESH	BARWE
C00007	AMIT	KUMAR
C00008	NISHA	DAMLE
C00009	ABHISHEK	DUTTA
C00010	SHANKAR	NAIR

24. Please follow instructions given below.

Write a query to display the customer number , firstname, customer's date of birth .

Display the records sorted in ascending order of date of birth year and within that sort by firstname in ascending order.

```
SELECT customer_number,firstname,customer_date_of_birth
FROM customer_master order by year(customer_date_of_birth),firstname;
```

CUSTOMER_NUMBER	FIRSTNAME	CUSTOMER_DATE_OF_BIRTH
C00009	ABHISHEK	1973-05-22
C00002	AVINASH	1974-10-16
C00008	NISHA	1975-12-03
C00005	NAVEEN	1976-09-19
C00004	PARUL	1976-11-03
C00001	RAMESH	1976-12-06

C00010	SHANKAR	1976-07-12
C00007	AMIT	1981-09-06
C00003	RAHUL	1981-09-26
C00006	CHITRESH	1992-11-06

25. Please follow instructions given below.

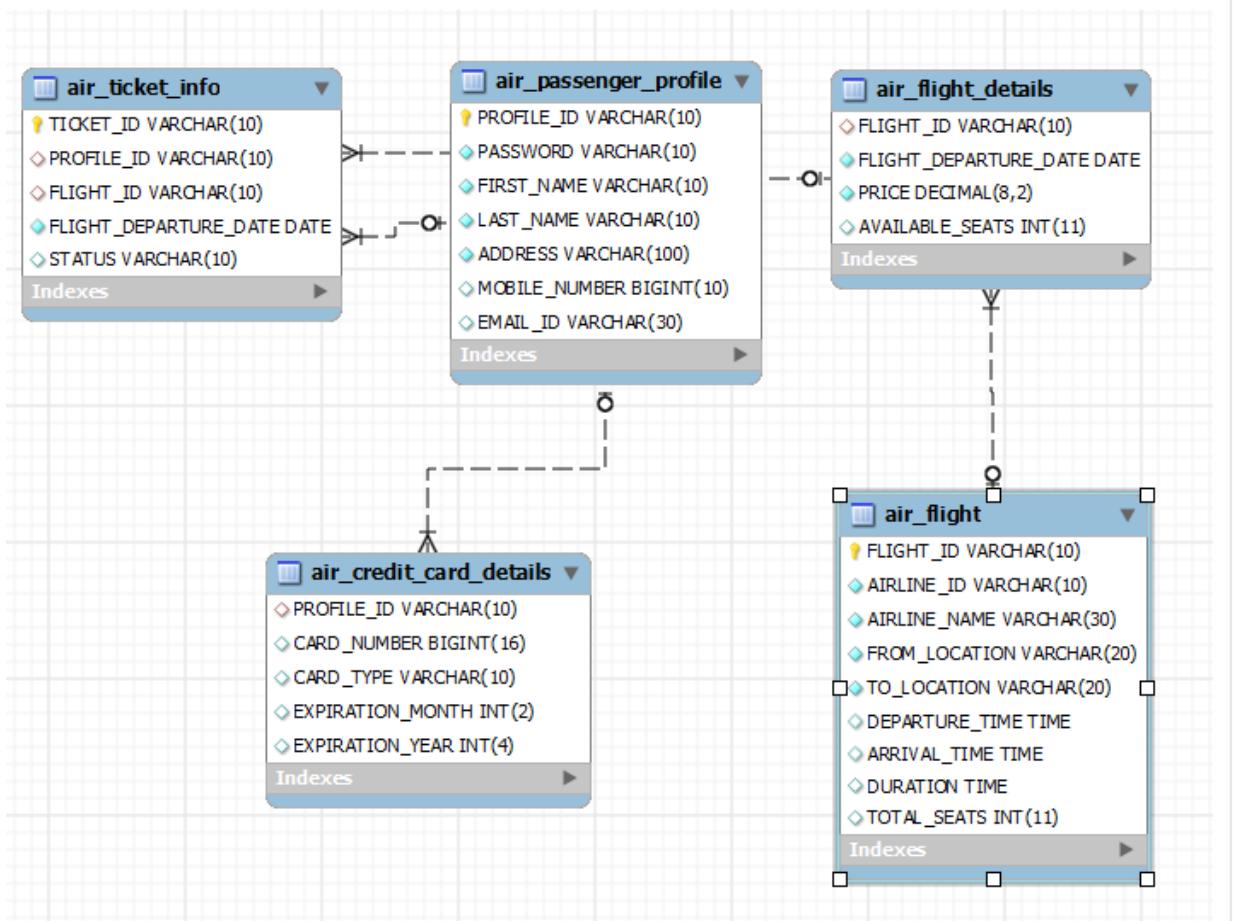
Write a query to display the customers firstname, city and account number whose occupation are not into Business, Service or Student.

Display the records sorted in ascending order based on customer first name and then by account number.

```
SELECT firstname, customer_city,account_number
FROM customer_master cm INNER JOIN account_master am
ON cm.customer_number=am.customer_number
WHERE occupation !='Service' and occupation != 'Student' and occupation != 'Business'
order by firstname, account_number
```

FIRSTNAME	CUSTOMER_CITY	ACCOUNT_NUMBER
PARUL	DELHI	A00010

ANSI SQL Airlines Management Schema



```
CREATE TABLE air_credit_card_details
(
    profile_id      VARCHAR(10)      NOT NULL,
    card_number     BIGINT(16),
    card_type       VARCHAR(10),
    expiration_month INT(2),
    expiration_year INT(4)
);
```

```
CREATE TABLE air_passenger_profile
(
    profile_id VARCHAR(10) NOT NULL ,
    password VARCHAR(10) NULL ,
    first_name VARCHAR(10) NULL ,
    last_name VARCHAR(10) NULL ,
    address VARCHAR(100) NULL ,
    mobile_number BIGINT(10) ,
    email_id VARCHAR(30) NUll
);
```

```
CREATE TABLE air_ticket_info
(
    ticket_id VARCHAR(10) NOT NULL ,
    profile_id VARCHAR(10) NULL ,
    flight_id VARCHAR(10) NULL ,
    flight_departure_date DATE NULL ,
    status VARCHAR(10) NULL
);
```

```
CREATE TABLE air_flight_details
(
    flight_id VARCHAR(10) NOT NULL ,
    flight_departure_date DATE NULL ,
    price DECIMAL(8,2) NULL ,
    available_seats INT (11)
);
```

```
CREATE TABLE air_flight
(
```

```

flight_id VARCHAR(10) NOT NULL ,
airline_id VARCHAR(10) NULL ,
airline_name VARCHAR(30) NULL ,
from_location VARCHAR(20) NULL ,
to_location VARCHAR(20) NULL ,
departure_time TIME NULL ,
arrival_time TIME NULL ,
duration TIME NULL ,
total_seats int(11)
);

```

```

Insert into air_flight values('916' ,      '100000'      , 'ABC AIRLINES' ,      'CHENNAI ' ,
                            'HYDERABAD'      , '19:55:00' ,    '21:00:00',
                            '01:05:00 ', 40),
('289', '100000'      , 'ABC AIRLINES' ,      'CHENNAI'     , 'KOCHI'      , '08:40:00'
                            , '09:55:00' ,    '01:15:00' ,    80),
('1011','100000',      'ABC AIRLINES' ,      'HYDERABAD'     , 'CHENNAI'     , '12:30:00',
                            '13:55:00' ,    '01:25:00' ,    50),
('3004','100000',      'ABC AIRLINES' ,      'BENGALURU'     , 'CHENNAI'     , '09:05:00',
                            '10:25:00' ,    '01:20:00' ,    100),
('3307','100000',      'ABC AIRLINES' ,      'BENGALURU'     , 'CHENNAI'     , '18:45:00'
                            , '19:45:00' ,    '01:00:00' ,    40),
('3013','100000'      , 'ABC AIRLINES' ,      'CHENNAI' ,    'BENGALURU' ,    '07:40:00'
                            , '08:45:00' ,    '01:05:00' ,    50),
('3148','100000'      , 'ABC AIRLINES','CHENNAI','BENGALURU','20:15:00' , '21:20:00',
                            '01:05:00' ,    100),
('1265','100000'      , 'ABC AIRLINES', 'CHENNAI', 'HYDERABAD','21:25:00' , '22:55:00',
                            '01:30:00' ,    100),
('3241','100000'      , 'ABC AIRLINES', 'CHENNAI' , 'KOCHI'      , '10:40:00' , '12:05:00',
                            '01:25:00' ,    80),
('3244','100000'      , 'ABC AIRLINES' , 'KOCHI'      , 'CHENNAI' ,    '21:10:00',
                            '22:40:00' ,    '01:30:00' ,    50),
('1262','100000'      , 'ABC AIRLINES' , 'HYDERABAD'     , 'CHENNAI' ,    '06:00:00'
                            , '07:20:00' ,    '01:20:00' ,    100);

```

Insert into air_ticket_info values

```
(TKT001 ' ,          'PFL007' ,        '3004' ,        '2013-05-02' , 'BOOKED'),
```

```

('TKT002',           'PFL007'           , '3004',           '2013-05-02 ',' BOOKED '),
('TKT003',' PFL007', '3004','2013-05-02','BOOKED'),
('TKT004','PFL001','1265','2013-04-29',' BOOKED '),
('TKT005', ' PFL001 ','1011' , '2013-05-09','BOOKED' ),
('TKT006','PFL007',' 3148','2013-05-21','BOOKED '),
('TKT007',' PFL007','3148',' 2013-05-21','BOOKED '),
('TKT008','PFL003','3244','2013-05-03',' BOOKED' ),
('TKT009',' PFL003','3244',' 2013-05-03','BOOKED' ),
('TKT010','PFL003','3244','2013-05-03','BOOKED' ),
('TKT011','PFL002',' 3148',' 2013-06-01',' BOOKED '),
('TKT012',' PFL002','3148',' 2013-06-01','BOOKED' ),
('TKT013',' PFL002','3148',' 2013-06-01',' BOOKED '),
('TKT014',' PFL002','3148','2013-06-01','BOOKED '),
('TKT015','PFL002','3148','2013-06-01','BOOKED '),
('TKT016',' PFL004','1262',' 2013-05-20',' BOOKED '),
('TKT017','PFL004','1265',' 2013-05-29','BOOKED '),
('TKT018',' PFL006','3244 ','2013-05-03',' BOOKED '),
('TKT019','PFL006','3244','2013-05-03','BOOKED '),
('TKT020','PFL006','3244','2013-05-03',' BOOKED '),
('TKT021',' PFL006','3244','2013-05-03','BOOKED '),
('TKT022',' PFL005','916',' 2013-05-06',' BOOKED '),
('TKT023','PFL005','916',' 2013-05-06','BOOKED '),
('TKT024',' PFL005',' 1011','2013-05-09','BOOKED '),
('TKT025','PFL005','1011','2013-05-09','BOOKED '),
('TKT026',' PFL008','1011',' 2013-05-09','BOOKED');

```

```

Insert into air_flight_details values ( '916',      '2013-04-28','4086.00',40),
('916','2013-05-12',    '3023.00',        40),
('916', '2013-05-01','4086.00',        40),
('916','2013-05-06','3603.00', 38),
('289', '2013-05-06','3603.00', 80),
('289','2013-05-08','3603.00', 80),
('289','2013-05-20','3052.00', 80),
('289','2013-05-31','2773.00', 80),
('1011','2013-04-30','4614.00', 50),
('1011','2013-05-09','4131.00', 46),
('1011','2013-05-21','3580.00', 50),
('3004','2013-05-02','3603.00', 97),
('3004','2013-05-24','3052.00', 100),
('3004','2013-05-19','3304.00', 100),
('3307','2013-05-03','3603.00', 40),
('3307','2013-05-03','3304.00', 40),

```

```

('3307','2013-05-23','3304.00',        40),
('3307','2013-05-29','3305.00',        40),
('3013','2013-05-04','3603.00',        50),
('3013','2013-05-06','3603.00',        50),
('3013','2013-05-22','3052.00',        50),
('3013','2013-05-30','2773.00',        50),
('3148','2013-05-16','3052.00',       100),
('3148','2013-05-21','3052.00',        98),
('3148' ,2013-06-01', '2773.00',        95),
('1265' ,2013-04-29', '4086.00',        99),
('1265','2013-05-14', '3052.00',       100),
('1265','2013-05-18','4086.00',       100),
('1265' ,2013-05-29', '2773.00',        99),
('3241','2013-05-01', '4086.00',        80),
('3241','2013-05-13', '3052.00',        80),
('3241','2013-05-27', '2773.00',        80),
('3244' ,2013-05-03' , '3647.00',        43),
('3244','2013-05-15', '3096.00',        50),
('1262' , '2013-05-20', '3580.00' , 99),
('1262' , '2013-05-29', '3309.00' ,100);

```

```

insert into air_passenger_profile values('PFL001', 'PFL001', 'LATHA', 'SANKAR', '123
BROAD CROSS ST,CHENNAI-48'
,'9876543210','LATHA@GMAIL.COM' ),
('PFL002',' PFL002',' ARUN',' PRAKASH',' 768 2ND STREET,BENGALURU-20',
'8094564243 ','ARUN@AOL.COM' ),
('PFL003',' PFL003 ','AMIT',' VIKARAM',' 43 5TH STREET,KOCHI-84',   '9497996990
','AMIT@AOL.COM'),
('PFL004',' PFL004',' AARTHI',' RAMESH',' 343 6TH STREET,HYDERABAD-76','
9595652530',' AARTHI@GMAIL.COM' ),
('PFL005',' PFL005',' SIVA',' KUMAR ','125 8TH STREET,CHENNAI-46 ',' 9884416986
','SIVA@GMAIL.COM' ),
('PFL006', 'PFL006 ','RAMESH',' BABU',' 109 2ND CROSS ST,KOCHI-12',   '9432198760','
RAMESH@GMAIL.COM '),
('PFL007', 'PFL007' , 'GAYATHRI',' RAGHU' , '23 2ND CROSS ST,BENGALURU-12'
,'8073245678','GAYATHRI@GMAIL.COM '),
('PFL008', 'PFL008',' GANESH ','KANNAN',' 45 3RD ST,HYDERABAD-21',   '9375237890
','GANESH@GMAIL.COM ');

```

```

insert into air_credit_card_details values('PFL001', '4312467849804008','GOLD',      2,
2020),
('PFL002',    '4763421245304900',' PLATINIUM', 10      ,2015),
('PFL003'     , '4112432643784789',' INSTANT',   3      ,2014),
('PFL004',    '4230412745604112' , 'GOLD',      7      ,2016),

```

('PFL005', '4122432147834003', 'PLATINUM', 12, 2016),
('PFL006', '4227438945674009', 'GOLD', 12, 2018),
('PFL007', '4124459047784138', 'INSTANT', 3, 2015),
('PFL008', '4312411745884010', 'PLATINUM', 9, 2017);

Item Loan Database Queries

1. Please follow instructions given below.

Write a query to display category and number of items in that category. Give the count an alias name of Count_category. Display the details on the sorted order of count in descending order.

3 rows

```
SELECT item_category , count(item_id) Count_category  
FROM item_master  
GROUP BY item_category order by count_category DESC;
```

Filter:		
	item_category	Count_category
▶	furniture	15
	Crockery	4
	Stationary	3

2. Please follow instructions given below.

Write a query to display the number of employees in HR department. Give the alias name as No_of_Employees.

1 row

```
SELECT count(employee_id) AS No_of_Employees  
FROM employee_master  
WHERE department= 'HR'
```

Filter:	
	No_of_Employees
▶	2

3. Please follow instructions given below.

Write a query to display employee id, employee name, designation and department for employees who have never been issued an item as a loan from the company. Display the records sorted in ascending order based on employee id.

1 row

```
select employee_id,employee_name,designation,department from employee_master  
where employee_id  
not in (select employee_id from employee_issue_details) order by employee_id;
```

	employee_id	employee_name	designation	department
▶	E00005	Radica	Manager	HR
*	NULL	NULL	NULL	NULL

4. Please follow instructions given below.

Write a query to display the employee id, employee name who was issued an item of highest valuation.

In case of multiple records, display the records sorted in ascending order based on employee id.

[Hint Suppose an item called dinning table is of 22000 and that is the highest price of the item that has been issued. So display the employee id and employee name who issued dinning table whose price is 22000.]

1 row

```
select em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id
and im.item_valuation>=all(select im.item_valuation from employee_master em
join employee_issue_details eid
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id)
order by employee_id;
```

	employee_id	employee_name
▶	E00004	Zuben

5. Please follow instructions given below.

Write a query to display issue_id, employee_id, employee_name.

Display the records sorted in ascending order based on issue id.

9 rows

```
select eid.issue_id,eid.employee_id,em.employee_name from employee_issue_details eid join
employee_master em on eid.employee_id=em.employee_id group by
eid.issue_id,eid.employee_id
```

	issue_id	employee_id	employee_name
▶	ISS001	E00001	Ram
	ISS002	E00001	Ram
	ISS003	E00002	Abhay
	ISS004	E00003	Anita
	ISS005	E00003	Anita
	ISS006	E00003	Anita
	ISS007	E00004	Zuben
	ISS008	E00006	John
	ISS009	E00004	Zuben

order by eid.issue_id;

6. Please follow instructions given below.

Write a query to display employee id, employee name who don't have loan cards.

Display the records sorted in ascending order based on employee id.

3 rows

```
SELECT employee_id, employee_name  
FROM employee_master  
WHERE employee_id NOT IN ( SELECT employee_id FROM employee_card_details )  
order by employee_id;
```

	employee_id	employee_name
▶	E00004	Zuben
	E00005	Radica
▶	E00006	John
*	NULL	NULL

7. Please follow instructions given below.

Write a query to count the number of cards issued to an employee "Ram". Give the count an alias name as No_of_Cards.

1 row

```
select count(eid.loan_id) as No_of_Cards from employee_card_details eid join  
employee_master em  
on eid.employee_id=em.employee_id where em.employee_name='Ram'
```

	No_of_Cards
▶	3

8. Please follow instructions given below.

Write a query to display the count of customers who have gone for loan type stationary. Give the count an alias name as Count_stationary.

1 row

```
select count(ecd.employee_id) as Count_Stationary from employee_card_details ecd  
join loan_card_master lcm on ecd.loan_id=lcm.loan_id where lcm.loan_type='Stationary'
```

	Count_stationary
▶	3

9. Please follow instructions given below.

Write a query to display the employee id, employee name and number of items issued to them. Give the number of items an alias name as Count. Display the details in descending order of count and then by employee id in ascending order. Consider only employees who have been issued atleast 1 item.

5 rows

```
select em.employee_id,em.employee_name,count(eid.item_id) as Count from
employee_master em join
employee_issue_details eid on em.employee_id=eid.employee_id group by em.employee_id
having
count(eid.item_id)>=1 order by Count desc,employee_id asc;
```

	employee_id	employee_name	Count
▶	E00003	Anita	3
	E00001	Ram	2
	E00004	Zuben	2
	E00002	Abhay	1
	E00006	John	1

10.Please follow instructions given below.

Write a query to display the employee id, employee name who was issued an item of minimum valuation.

In case of multiple records, display them sorted in ascending order based on employee id.

[Hint Suppose an item called pen is of rupees 20 and that is the lowest price. So display the employee id and employee name who issued pen where the valuation is 20.]

2 rows

```
select em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id
and im.item_valuation<=all (select im.item_valuation from employee_master em join
employee_issue_details eid
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id) order by
employee_id;
```

	employee_id	employee_name
▶	E00002	Abhay
	E00003	Anita

11.Please follow instructions given below.

Write a query to display the employee id, employee name and total valuation of the product issued to each employee. Give the alias name as TOTAL_VALUATION.

Display the records sorted in ascending order based on employee id.

Consider only employees who have been issued atleast 1 item.

5 rows

```
select em.employee_id,em.employee_name,sum(im.item_valuation) as TOTAL_VALUATION
from employee_master em
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im
on eid.item_id=im.item_id group by em.employee_id having count(im.item_valuation)>=1
```

```
order by em.employee_id;
```

	employee_id	employee_name	TOTAL_VALUATION
▶	E00001	Ram	7000.00
	E00002	Abhay	1500.00
	E00003	Anita	15500.00
	E00004	Zuben	25500.00
	E00006	John	4500.00

12. Please follow instructions given below.

Write a query to display distinct employee id, employee name who kept the item issued for more than a year. Hint: Use Date time function to calculate the difference between item issue and return date. Display the records only if it is more than 365 Days.

Display the records sorted in ascending order based on employee id.

5 rows

```
select distinct em.employee_id,em.employee_name from employee_master em join  
employee_issue_details eid  
on em.employee_id=eid.employee_id where datediff(return_date,issue_date)>365 order by  
employee_id;
```

	employee_id	employee_name
▶	E00001	Ram
	E00002	Abhay
	E00003	Anita
	E00004	Zuben
	E00006	John

13. Please follow instructions given below.

Write a query to display employee id, employee name and count of items of those who asked for more than 1 furniture. Give the alias name for count of items as COUNT_ITEMS.

Display the records sorted in ascending order on employee id.

2 rows

```
select em.employee_id,em.employee_name,count(im.item_id) as COUNT_ITEMS from  
employee_master em  
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im  
on eid.item_id=im.item_id where item_category='furniture' group by employee_id having  
count(COUNT_ITEMS)>1 order by employee_id;
```

Filter: | Export: | Autos

	employee_id	employee_name	COUNT_ITEMS
▶	E00001	Ram	2
	E00003	Anita	3

14. Please follow instructions given below.

Write a query to display the number of men & women Employees. The query should display the gender and number of Employees as No_of_Employees. Display the records sorted in ascending order based on gender.

2 rows

select gender,count(employee_id) as No_of_Employees from employee_master group by

Filter: |

	gender	No_of_Employees
▶	F	2
	M	4

gender order by gender;

15. Please follow instructions given below.

Write a query to display employee id, employee name who joined the company after 2005. Display the records sorted in ascending order based on employee id.

3 rows

select employee_id,employee_name from employee_master where year(date_of_joining)>2005

Filter: |

	employee_id	employee_name
	E00002	Abhay
	E00003	Anita
	E00006	John

order by employee_id;

16. Please follow instructions given below.

Write a query to get the number of items of the furniture category issued and not issued. The query should display issue status and the number of furniture as No_of_Furnitures. Display the records sorted in ascending order based on issue_status.

2 rows

select issue_status,count(item_id) as No_of_Furnitures from item_master where item_category='furniture' group by issue_status order by issue_status;

	issue_status	No_of_Furnitures
▶	N	6
	Y	9

17. Please follow instructions given below.

Write a query to find the number of items in each category, make and description. The Query should display Item Category, Make, description and the number of items as No_of_Items.

Display the records in ascending order based on Item Category, then by item make and then by item description.

16 rows

```
select item_category,item_make,item_description,count(item_id) as No_of_Items from
item_master im group by item_category,item_make,item_description order by
item_category,item_make,item_description;
```

	item_category	item_make	item_description	No_of_Items
▶	Crockery	Bonechina	Dining Set	1
	Crockery	Bonechina	Tea Set	1
	Crockery	Glass	Dining Set	1
	Crockery	Glass	Tea Set	1
	furniture	Steel	Cupboard	2
	furniture	Steel	Side Table	1
	furniture	Steel	Single Bed	2
	furniture	Steel	Tea Table	2
	furniture	Wooden	Dining Chair	1
	furniture	Wooden	Dining Table	1
	furniture	Wooden	Double Bed	2
	furniture	Wooden	Side Table	1
	furniture	Wooden	Sofa	1
	furniture	Wooden	Tea Table	2
	Stationary	Plastic	Pen	2
	Stationary	Wooden	Pencil	1

18. Please follow instructions given below.

Write a query to display employee id, employee name, item id and item description of employees who were issued item(s) in the month of January 2013. Display the records sorted in order based on employee id and then by item id in ascending order.

1 row

```

select em.employee_id,em.employee_name,im.item_id,im.item_description from
employee_master em join
employee_issue_details eid on em.employee_id=eid.employee_id join item_master im on
eid.item_id=im.item_id where year(eid.issue_date)=2013 and month(eid.issue_date)=01 order
by

```

	employee_id	employee_name	item_id	item_description
▶	E00002	Abhay	I00005	Side Table

em.employee_id,im.item_id;

19. Please follow instructions given below.

Write a query to display the employee id, employee name and count of item category of the employees who have been issued items in at least 2 different categories.

Give the alias name for category count as COUNT_CATEGORY.

Display the records sorted in ascending order based on employee id.

1 row

```

select em.employee_id,em.employee_name,count(distinct im.item_category) as
COUNT_CATEGORY from employee_master em
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im
on eid.item_id=im.item_id group by em.employee_id having COUNT_CATEGORY>=2
order by em.employee_id;

```

	employee_id	employee_name	COUNT_CATEGORY
▶	E00004	Zuben	2

20. Please follow instructions given below.

Write a query to display the item id , item description which was never issued to any employee.

Display the records sorted in ascending order based on item id.

14 rows

```

select item_id,item_description from item_master where item_id not in (select item_id
from employee_issue_details) order by item_id;

```

	item_id	item_description
▶	I00002	Dining Table
	I00003	Tea Table
	I00006	Tea Table
	I00009	Sofa
	I00011	Cupboard
	I00013	Double Bed
	I00014	Single Bed
	I00015	Single Bed
	I00016	Tea Set
	I00017	Tea Set
	I00019	Dining Set
	I00020	Pencil
	I00021	Pen
	I00022	Pen
	NUL	NUL

21. Please follow instructions given below.

Write a query to display the employee id, employee name and total valuation for the employees who has issued minimum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION.

[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issued item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed.]

1 row

```
select em.employee_id,em.employee_name,sum(im.item_valuation) as TOTAL_VALUATION
from employee_master em
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im on
eid.item_id=im.item_id group by em.employee_id having sum(im.item_valuation) <= all
(select sum(im.item_valuation) from employee_master em
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im on
eid.item_id=im.item_id group by em.employee_id) order by employee_id;
```

	employee_id	employee_name	TOTAL_VALUATION
▶	E00002	Abhay	1500.00

22. Please follow instructions given below.

Write a query to display the employee id, employee name, card issue date and card valid date.

Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD_VALID_DATE.

[Hint: Validity in years for the loan card is given in loan_card_master table. Validity date is calculated by adding number of years in the loan card issue date. If the duration of year is zero then display AS 'No Validity Date'.]

```
SELECT ecd.employee_id,employee_name,
card_issue_date, if(lcd.duration_in_years=0, 'NO-VALIDITY DATE',
date_add(ec.card_issue_date, interval duration_in_years year)) as CARD_VALIDITY_DATE
FROM employee_master em INNER JOIN
employee_card_details ecd
ON em.employee_id=ecd.employee_id
INNER JOIN loan_card_master lcd
ON ecd.loan_id=lcd.loan_id
order by employee_name, CARD_VALID_DATE;
```

	employee_id	employee_name	card_issue_date	CARD_VALID_DATE
▶	E00002	Abhay	2007-02-01	2012-02-01
	E00002	Abhay	2007-03-11	No Validity Date
	E00003	Anita	2007-04-15	2008-04-15
	E00003	Anita	2007-04-15	2012-04-15
	E00003	Anita	2007-04-15	No Validity Date
	E00001	Ram	2002-12-14	2003-12-14
	E00001	Ram	2000-01-01	2005-01-01
	E00001	Ram	2000-01-01	No Validity Date

23. Please follow instructions given below.

Write a query to display the employee id, employee name who have not issued with any item in the year 2013. Hint: Exclude those employees who was never issued with any of the items in all the years. Display the records sorted in ascending order based on employee id.

3 rows

```
select distinct em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid on
em.employee_id=eid.employee_id where em.employee_id not in
(select employee_id from employee_issue_details where year(issue_date)=2013)
order by employee_id;
```

Filter:		
	employee_id	employee_name
▶	E00001	Ram
	E00003	Anita
	E00006	John

24. Please follow instructions given below.

Write a query to display issue id, employee id, employee name, item id, item description and issue date. Display the data in descending order of date and then by issue id in ascending order.

9 rows

```
select
eid.issue_id,em.employee_id,em.employee_name,im.item_id,im.item_description,eid.issue_date
from employee_issue_details eid join employee_master em on
eid.employee_id=em.employee_id
join item_master im on eid.item_id=im.item_id order by eid.issue_date desc,eid.issue_id;
```

Filter:		Export:	Autosize:			
	issue_id	employee_id	employee_name	item_id	item_description	issue_date
	ISS009	E00004	Zuben	I00018	Dining Set	2013-04-18
	ISS007	E00004	Zuben	I00012	Double Bed	2013-04-14
	ISS003	E00002	Abhay	I00005	Side Table	2013-01-03
	ISS008	E00006	John	I00018	Dining Set	2012-08-18
	ISS006	E00003	Anita	I00010	Cupboard	2012-03-14
	ISS001	E00001	Ram	I00001	Tea Table	2012-02-03
	ISS002	E00001	Ram	I00004	Side Table	2012-02-03
	ISS004	E00003	Anita	I00007	Dining Chair	2010-07-04
	ISS005	E00003	Anita	I00008	Tea Table	2010-07-04

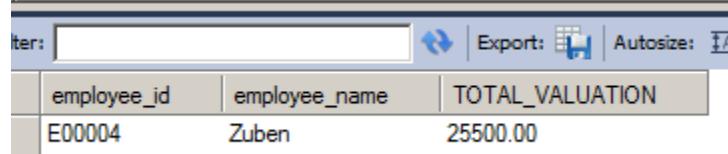
25. Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION;

[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issued item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name and total valuation of E00019 should display.]

1 row

```
select em.employee_id,em.employee_name,sum(im.item_valuation) as TOTAL_VALUATION
```

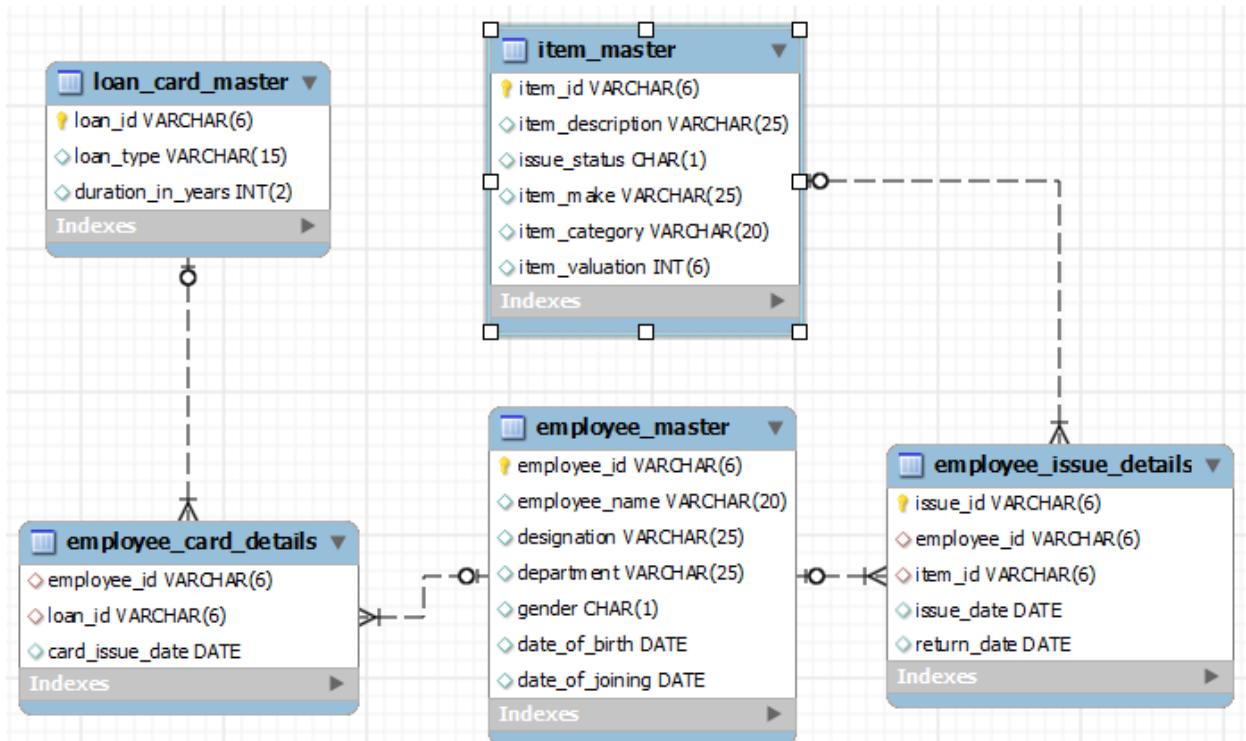
```
from employee_master em join employee_issue_details eid on  
em.employee_id=eid.employee_id  
join item_master im on eid.item_id=im.item_id group by em.employee_id having  
sum(im.item_valuation)  
>= all (select sum(im.item_valuation) from employee_master em join employee_issue_details  
eid on em.employee_id=eid.employee_id  
join item_master im on eid.item_id=im.item_id group by em.employee_id);;
```



A screenshot of a database query results window. The window has a title bar with a search field, an 'Export' button, and an 'Autosize' button. Below the title bar is a table with three columns: 'employee_id', 'employee_name', and 'TOTAL_VALUATION'. A single row of data is displayed: E00004, Zuben, and 25500.00.

	employee_id	employee_name	TOTAL_VALUATION
	E00004	Zuben	25500.00

ANSI SQL Item Loan Management System Schema



```

CREATE TABLE loan_card_master
(
    loan_id                  int(6)          PRIMARY KEY,
    loan_type                varchar(20),
    duration_in_years        int(2)
);

CREATE TABLE employee_master
(
    employee_id              varchar(10)      PRIMARY KEY,
    employee_name             varchar(30),
    designation               varchar(255),
    department                varchar(255),
    gender                    varchar(6),
    date_of_birth              date,
    date_of_joining            date
);

CREATE TABLE item_master
(
    item_id                  varchar(10)      PRIMARY KEY,
    item_description          varchar(50),
    issue_status              varchar(10),
    item_make                 varchar(20),
    item_category              varchar(20),
    item_valuation             double(7,2)
);

CREATE TABLE employee_card_details
(
    employee_id              varchar(10)      REFERENCES employee_master,
    loan_id                   varchar(10)      REFERENCES loan_card_master,
    card_issue_date           date
);

CREATE TABLE employee_issue_details
(
    issue_id                 varchar(10)      PRIMARY KEY,
    employee_id               varchar(10)      REFERENCES employee_master,
    item_id                   varchar(10)      REFERENCES item_master,
    issue_date                date,
    return_date                date
);

```

```
insert into loan_card_master
values('00001','stationary',5);
insert into loan_card_master
values('00002','recurring',0);
insert into loan_card_master
values('00003','Crockery',1);

insert into employee_master
values('E00001','Ram','Manager','Finance','M','1973-12-01','2001-01-01');

insert into employee_master
values('E00002','Abhay','Assistant Manager',
'Finance','M','1976-01-01','2006-12-01');

insert into employee_master
values('E00003','Anita','Senior Executive','Marketing','F','1977-05-12','2007-03-21');

insert into employee_master
values('E00004','Zuben','Manager','Marketing','M','1974-10-12','2003-07-23');

insert into employee_master
values('E00005','Radica','Manager','HR','F','1976-07-22','2004-01-23');

insert into employee_master
values('E00006','John','Executive','HR','M','1983-11-08','2010-05-17');

insert into item_master
values ('I00001','Tea Table','Y','Wooden','furniture',5000);

insert into item_master
values ('I00002','Dining Table','N','Wooden','furniture',15000);

insert into item_master
values ('I00003','Tea Table','N','Steel','furniture',6000);

insert into item_master
values ('I00004','Side Table','Y','Wooden','furniture',2000);

insert into item_master
values ('I00005','Side Table','Y','Steel','furniture',1500);
```

```
insert into item_master
values ('I00006','Tea Table','N','Steel','furniture',7000);

insert into item_master
values ('I00007','Dining Chair','Y','Wooden','furniture',1500);

insert into item_master
values ('I00008','Tea Table','Y','Wooden','furniture',4000);

insert into item_master
values ('I00009','Sofa','N','Wooden','furniture',18000);

insert into item_master
values ('I00010','Cupboard','Y','Steel','furniture',10000);

insert into item_master
values ('I00011','Cupboard','N','Steel','furniture',14000);

insert into item_master
values ('I00012','Double Bed','Y','Wooden','furniture',21000);

insert into item_master
values ('I00013','Double Bed','Y','Wooden','furniture',20000);

insert into item_master
values ('I00014','Single Bed','Y','Steel','furniture',10000);

insert into item_master
values ('I00015','Single Bed','N','Steel','furniture',10000);

insert into item_master
values ('I00016','Tea Set','Y','Glass','Crockery',3000);

insert into item_master
values ('I00017','Tea Set','Y','Bonechina','Crockery',4000);

insert into item_master
values ('I00018','Dining Set','Y','Glass','Crockery',4500);

insert into item_master
values ('I00019','Dining Set','N','Bonechina','Crockery',5000);

insert into item_master
values ('I00020','Pencil','Y','Wooden','Stationary',5);
```

```
insert into item_master
values ('I00021','Pen','Y','Plastic','Stationary',100);

insert into item_master
values ('I00022','Pen','N','Plastic','Stationary',200);

insert into employee_card_details
values('E00001','00001','2000-01-01');

insert into employee_card_details
values('E00001','00002','2000-01-01');

insert into employee_card_details
values('E00001','00003','2002-12-14');

insert into employee_card_details
values('E00002','00001','2007-02-01');

insert into employee_card_details
values('E00002','00002','2007-03-11');

insert into employee_card_details
values('E00003','00001','2007-04-15');

insert into employee_card_details
values('E00003','00002','2007-04-15');

insert into employee_card_details
values('E00003','00003','2007-04-15');

insert into employee_issue_details
values('ISS001','E00001','I00001','2012-02-03','2014-02-03');
insert into employee_issue_details
values('ISS002','E00001','I00004','2012-02-03','2020-02-03');

insert into employee_issue_details
values('ISS003','E00002','I00005','2013-01-03','2015-01-03');
insert into employee_issue_details
values('ISS004','E00003','I00007','2010-07-04','2012-07-04');
```

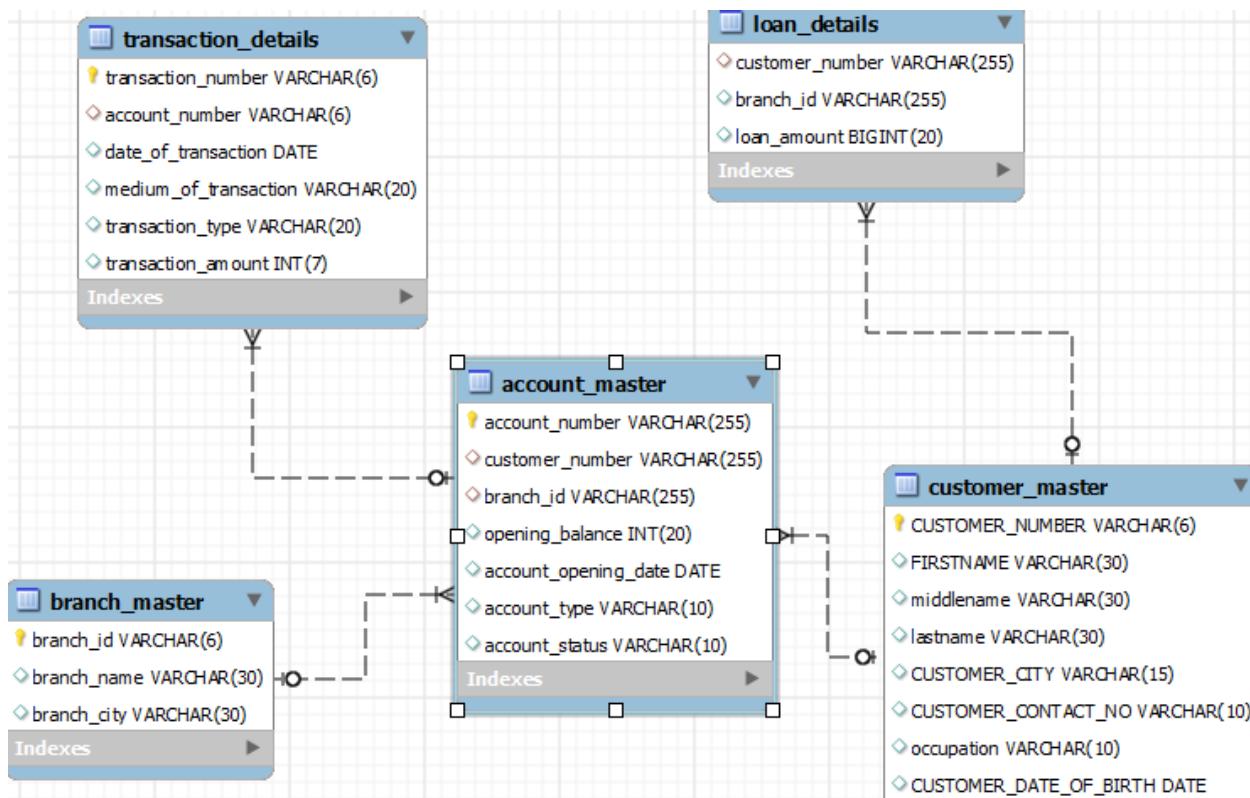
```
insert into employee_issue_details
values('ISS005','E00003','I00008','2010-07-04','2012-08-05');

insert into employee_issue_details
values('ISS006','E00003','I00010','2012-03-14','2012-06-15');

insert into employee_issue_details
values('ISS007','E00004','I00012','2013-04-14','2016-04-14');

insert into employee_issue_details
values('ISS008','E00006','I00018','2012-08-18','2019-04-17');

insert into employee_issue_details
values('ISS009','E00004','I00018','2013-04-18','2013-05-18');
```



Bank Management System Queries:

Bank Management System Queries:

1. Please follow instructions given below.

Write a query to display account number, customer's number, customer's firstname, lastname, account opening date.

Display the records sorted in ascending order based on account number.

SELECT

```

account_number, am.customer_number, firstname, lastname, account_opening_date
FROM customer_master cm JOIN account_master am
ON cm.customer_number=am.customer_number
ORDER BY account_number;
    
```

ACCOUNT_NUMBER	CUSTOMER_NUMBER	FIRSTNAME	LASTNAME	ACCOUNT_OPENING_DATE
A00001	C00001	RAMESH	SHARMA	2012-12-15
A00002	C00002	AVINASH	MINHA	2012-06-12
A00003	C00003	RAHUL	RASTOGI	2012-05-17
A00004	C00002	AVINASH	MINHA	2013-01-27

A00005	C00006	CHITRESH	BARWE	2012-12-17
A00006	C00007	AMIT	BORKAR	2010-08-12
A00007	C00007	AMIT	BORKAR	2012-10-02
A00008	C00001	RAMESH	SHARMA	2009-11-09
A00009	C00003	RAHUL	RASTOGI	2008-11-30
A00010	C00004	PARUL	GANDHI	2013-03-01

2. Please follow instructions given below.

Write a query to display the number of customer's from Delhi. Give the count an alias name of Cust_Count.

```
SELECT count(customer_number) Cust_Count
FROM customer_master
WHERE customer_city='Delhi'
```

CUST_COUNT
4

3. Please follow instructions given below.

Write a query to display the customer number, customer firstname, account number for the customer's whose accounts were created after 15th of any month.

Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT am.customer_number, firstname, account_number
FROM customer_master cm JOIN account_master am
ON cm.customer_number=am.customer_number
WHERE day(account_opening_date)>15
ORDER BY am.customer_number, account_number
```

CUSTOMER_NUMBER	FIRSTNAMEMiddle_Name	ACCOUNT_NUMBER
C00002	AVINASH	A00004
C00003	RAHUL	A00003
C00003	RAHUL	A00009

C00006	CHITRESH	A00005
--------	----------	--------

4. Please follow instructions given below.

Write a query to display customer number, customer's first name, account number where the account status is terminated.

Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT am.customer_number,firstname,account_number
FROM customer_master cm JOIN account_master am
ON cm.customer_number=am.customer_number
WHERE account_status='Terminated'
ORDER BY am.customer_number,account_number
```

CUSTOMER_NUMBER	FIRSTNAME	ACCOUNT_NUMBER
C00001	RAMESH	A00008
C00003	RAHUL	A00009

5. Please follow instructions given below.

Write a query to display the total number of withdrawals and total number of deposits being done by customer whose customer number ends with 001. The query should display transaction type and the number of transactions. Give an alias name as Trans_Count for number of transactions.

Display the records sorted in ascending order based on transaction type.

```
SELECT transaction_type,count(transaction_number) Trans_Count
FROM account_master am JOIN transaction_details td
ON am.account_number=td.account_number
WHERE customer_number like '%001'
GROUP BY transaction_type
ORDER BY transaction_type
```

TRANSACTION_TYPE	TRANS_COUNT
DEPOSIT	3
WITHDRAWAL	3

6. Please follow instructions given below.

Write a query to display the number of customers who have registration but no account in the bank.

Give the alias name as Count_Customer for number of customers.

```
SELECT count(customer_number) Count_Customer  
FROM customer_master  
WHERE customer_number NOT IN (SELECT customer_number FROM account_master)
```

COUNT_CUSTOMER
4

7. Please follow instructions given below.

Write a query to display account number and total amount deposited by each account holder (Including the opening balance). Give the total amount deposited an alias name of Deposit_Amount. Display the records in sorted order based on account number.

```
SELECT td.account_number, opening_balance+sum(transaction_amount)  
Deposit_Amount  
FROM account_master am INNER JOIN transaction_details td  
ON am.account_number=td.account_number  
WHERE transaction_type='deposit'  
GROUP BY account_number  
ORDER BY account_number
```

ACCOUNT_NUMBER	DEPOSIT_AMOUNT
A00001	10000
A00002	6000
A00007	17000

8. Please follow instructions given below.

Write a query to display the number of accounts opened in each city .The Query should display Branch City and number of accounts as No_of_Accounts.For the branch city where we don't have any accounts opened display 0. Display the records in sorted order based on branch city.

```

select branch_master.branch_city, count(account_master.account_number) as
No_of_Accounts from branch_master left join account_master on
account_master.branch_id=branch_master.branch_id
group by branch_master.branch_city order by branch_city;

```

BRANCH_CITY	NO_OF_ACCOUNTS
CHENNAI	0
DELHI	6
KOLKATA	0
MUMBAI	4

9. Please follow instructions given below.

Write a query to display the firstname of the customers who have more than 1 account.
Display the records in sorted order based on firstname.

```

select firstname
FROM customer_master cm INNER JOIN account_master am
ON cm.customer_number=am.customer_number
group by firstname
having count(account_number)>1
order by firstname;

```

FIRSTNAME
AMIT
AVINASH
RAHUL
RAMESH

10. Please follow instructions given below.

Write a query to display the customer number, customer firstname, customer lastname
who has taken loan from more than 1 branch.
Display the records sorted in order based on customer number.

```

SELECT Id.customer_number, firstname, lastname
FROM customer_master cm INNER JOIN loan_details Id
ON cm.customer_number=Id.customer_number

```

**GROUP BY customer_number
HAVING count(branch_id)>1
ORDER BY customer_number**

CUSTOMER_NUMBER	FIRSTNAME	LASTNAME
C00001	RAMESH	SHARMA
C00002	AVINASH	MINHA

11. Please follow instructions given below.

**Write a query to display the customer's number, customer's firstname, customer's city and branch city where the city of the customer and city of the branch is different.
Display the records sorted in ascending order based on customer number.**

```
select customer_master.customer_number, firstname, customer_city, branch_city
from account_master inner join customer_master on account_master.customer_number
= customer_master.customer_number
inner join branch_master on account_master.branch_id = branch_master.branch_id
where customer_city != branch_city order by customer_master.customer_number;
```

CUSTOMER_NUMBER	FIRSTNAME	CUSTOMER_CITY	BRANCH_CITY
C00002	AVINASH	DELHI	MUMBAI
C00003	RAHUL	DELHI	MUMBAI
C00007	AMIT	MUMBAI	DELHI

12. Please follow instructions given below.

Write a query to display the number of clients who have asked for loans but they don't have any account in the bank though they are registered customers. Give the count an alias name of Count.

```
SELECT count(Id.customer_number) Count
FROM customer_master cm INNER JOIN loan_details Id
ON cm.customer_number=Id.customer_number
WHERE cm.customer_number NOT IN ( SELECT customer_number FROM
account_master)
(Or)
select count(customer_number) as Count from customer_master where
customer_number not in
(select customer_number from account_master) and customer_number in
(select customer_number from loan_details);
```

COUNT
2

13. Please follow instructions given below.

Write a query to display the account number who has done the highest transaction. For example the account A00023 has done 5 transactions i.e. suppose 3 withdrawal and 2 deposits. Whereas the account A00024 has done 3 transactions i.e. suppose 2 withdrawals and 1 deposit. So account number of A00023 should be displayed. In case of multiple records, display the records sorted in ascending order based on account number.

```
SELECT td.account_number
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by td.account_number
having count(td.transaction_number)>=ALL
(SELECT count(td.transaction_number)
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by td.account_number) order by am.account_number;
```

ACCOUNT_NUMBER
A00001

14. Please follow instructions given below.

Write a query to show the branch name, branch city where we have the maximum customers.

For example the branch B00019 has 3 customers, B00020 has 7 and B00021 has 10. So branch id B00021 is having maximum customers. If B00021 is Koramangla branch Bangalore, Koramangla branch should be displayed along with city name Bangalore. In case of multiple records, display the records sorted in ascending order based on branch name.

```
select branch_name,branch_city
FROM branch_master INNER JOIN account_master
ON branch_master.branch_id=account_master.branch_id
group by branch_name
having count(customer_number)>=ALL
(select count(customer_number)
FROM branch_master INNER JOIN account_master
```

**ON branch_master.branch_id=account_master.branch_id
group by branch_name) order by branch_name;**

BRANCH_NAME	BRANCH_CITY
ASAF ALI ROAD	DELHI

15. Please follow instructions given below.

Write a query to display all those account number, deposit, withdrawal where withdrawal is more than deposit amount. Hint: Deposit should include opening balance as well.

For example A00011 account opened with Opening Balance 1000 and A00011 deposited 2000 rupees on 2012-12-01 and 3000 rupees on 2012-12-02. The same account i.e A00011 withdrawn 3000 rupees on 2013-01-01 and 7000 rupees on 2013-01-03. So the total deposited amount is 6000 and total withdrawal amount is 10000. So withdrawal amount is more than deposited amount for account number A00011.

Display the records sorted in ascending order based on account number.

```
select am.account_number,opening_balance+sum(case when transaction_type='Deposit'  
then  
transaction_amount end) as Deposit,sum(case when transaction_type='withdrawal' then  
transaction_amount end) as Withdrawal from account_master am join  
transaction_details td  
on am.account_number=td.account_number group by am.account_number having  
Withdrawal>Deposit;
```

ACCOUNT_NUMBER	DEPOSIT	WITHDRAWAL
A00001	10000	12000
A00002	6000	7000

16. Please follow instructions given below.

Write a query to show the balance amount for account number that ends with 001.

Note: Balance amount includes account opening balance also. Give alias name as Balance_Amount.

For example A00015 is having an opening balance of 1000. A00015 has deposited 2000 on 2012-06-12 and deposited 3000 on 2012-07-13. The same account has drawn money of 500 on 2012-08-12 , 500 on 2012-09-15, 1000 on 2012-12-17. So balance amount is 4000 i.e $(1000 \text{ (opening balance)} + 2000 + 3000) - (500 + 500 + 1000)$.

```
SELECT (SUM(CASE WHEN transaction_type='Deposit'  
THEN transaction_amount END)) -  
(SUM(CASE WHEN transaction_type='Withdrawal'
```

```

THEN transaction_amount END)+(select opening_balance
from account_master where account_number like '%001') AS Balance_Amount
FROM transaction_details where account_number like '%001'

```

BALANCE_AMOUNT
-2000

17. Please follow instructions given below.

Display the customer number, customer's first name, account number and number of transactions being made by the customers from each account. Give the alias name for number of transactions as Count_Trans. Display the records sorted in ascending order based on customer number and then by account number.

```

SELECT cm.customer_number,firstname,
am.account_number,count(transaction_number) Count_Trans
FROM customer_master cm inner JOIN account_master am
ON cm.customer_number=am.customer_number
INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by am.account_number order by cm.customer_number, am.account_number

```

CUSTOMER_NUMBER	FIRSTNAME	ACCOUNT_NUMBER	COUNT_TRANS
C00001	RAMESH	A00001	6
C00002	AVINASH	A00002	3
C00007	AMIT	A00007	3

18. Please follow instructions given below.

Write a query to display the customer's firstname who have multiple accounts (atleast 2 accounts). Display the records sorted in ascending order based on customer's firstname.

```

SELECT firstname
FROM customer_master INNER JOIN account_master
ON customer_master.customer_number=account_master.customer_number
GROUP BY firstname
having count(firstname)>=2 order by firstname;

```

FIRSTNAME

AMIT
AVINASH
RAHUL
RAMESH

19. Please follow instructions given below.

Write a query to display the customer number, firstname, lastname for those client where total loan amount taken is maximum and at least taken from 2 branches.

For example the customer C00012 took a loan of 100000 from bank branch with id B00009 and C00012

Took a loan of 500000 from bank branch with id B00010. So total loan amount for customer C00012 is

600000. C00013 took a loan of 100000 from bank branch B00009 and 200000 from bank branch B00011.

So total loan taken is 300000. So loan taken by C00012 is more than C00013.

```
SELECT Id.customer_number, firstname, lastname
FROM customer_master cm INNER JOIN loan_details ld
ON cm.customer_number=ld.customer_number
group by customer_number
having count(branch_id)>=2 and sum(loan_amount)>=All(select sum(loan_amount) from
loan_details group by customer_number)
```

CUSTOMER_NUMBER	FIRSTNAME	LASTNAME
C00002	AVINASH	MINHA

20. Please follow instructions given below.

Write a query to display the customer's number, customer's firstname, branch id and loan amount for people who have taken loans..

Display the records sorted in ascending order based on customer number and then by branch id and then by loan amount.

```
SELECT Id.customer_number, firstname,branch_id, loan_amount
FROM customer_master cm INNER JOIN loan_details ld
```

ON cm.customer_number=id.customer_number order by cm.customer_number, branch_id, loan_amount

CUSTOMER_NUMBER	FIRSTNAME	BRANCH_ID	LOAN_AMOUNT
C00001	RAMESH	B00001	100000
C00001	RAMESH	B00003	600000
C00002	AVINASH	B00001	600000
C00002	AVINASH	B00002	200000
C00009	ABHISHEK	B00008	400000
C00010	SHANKAR	B00009	500000

21. Please follow instructions given below.

Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count_Branch.

Display the records sorted in ascending order based on city name.

```
SELECT branch_city, count(branch_id) Count_Branch
FROM branch_master
GROUP BY branch_city
ORDER BY branch_city
```

BRANCH_CITY	COUNT_BRANCH
CHENNAI	1
DELHI	4
KOLKATA	1
MUMBAI	3

22. Please follow instructions given below.

Write a query to display account id, customer's firstname, customer's lastname for the customer's whose account is Active.

Display the records sorted in ascending order based on account id /account number.

```
SELECT account_number, firstname, lastname
FROM customer_master cm INNER JOIN account_master am
ON cm.customer_number=am.customer_number
```

WHERE account_status='Active'

ORDER BY account_number

ACCOUNT_NUMBER	FIRSTNAME	LASTNAME
A00001	RAMESH	SHARMA
A00002	AVINASH	MINHA
A00003	RAHUL	RASTOGI
A00004	AVINASH	MINHA
A00005	CHITRESH	BARWE
A00007	AMIT	BORKAR
A00010	PARUL	GANDHI

23. Please follow instructions given below.

Write a query to display customer's number, first name and middle name. For the customers who don't have middle name, display their last name as middle name. Give the alias name as Middle_Name.

Display the records sorted in ascending order based on customer number.

**SELECT customer_number,firstname,coalesce(middlename,lastname) Middle_Name
FROM customer_master order by customer_number**

CUSTOMER_NUMBER	FIRSTNAME	MIDDLE_NAME
C00001	RAMESH	CHANDRA
C00002	AVINASH	SUNDER
C00003	RAHUL	RASTOGI
C00004	PARUL	GANDHI
C00005	NAVEEN	CHANDRA
C00006	CHITRESH	BARWE

C00007	AMIT	KUMAR
C00008	NISHA	DAMLE
C00009	ABHISHEK	DUTTA
C00010	SHANKAR	NAIR

24. Please follow instructions given below.

Write a query to display the customer number , firstname, customer's date of birth .

Display the records sorted in ascending order of date of birth year and within that sort by firstname in ascending order.

SELECT customer_number,firstname,customer_date_of_birth

FROM customer_master order by year(customer_date_of_birth), firstname;

CUSTOMER_NUMBER	FIRSTNAME	CUSTOMER_DATE_OF_BIRTH
C00009	ABHISHEK	1973-05-22
C00002	AVINASH	1974-10-16
C00008	NISHA	1975-12-03
C00005	NAVEEN	1976-09-19
C00004	PARUL	1976-11-03
C00001	RAMESH	1976-12-06
C00010	SHANKAR	1976-07-12
C00007	AMIT	1981-09-06
C00003	RAHUL	1981-09-26
C00006	CHITRESH	1992-11-06

25. Please follow instructions given below.

Write a query to display the customers firstname, city and account number whose occupation are not into Business, Service or Student.

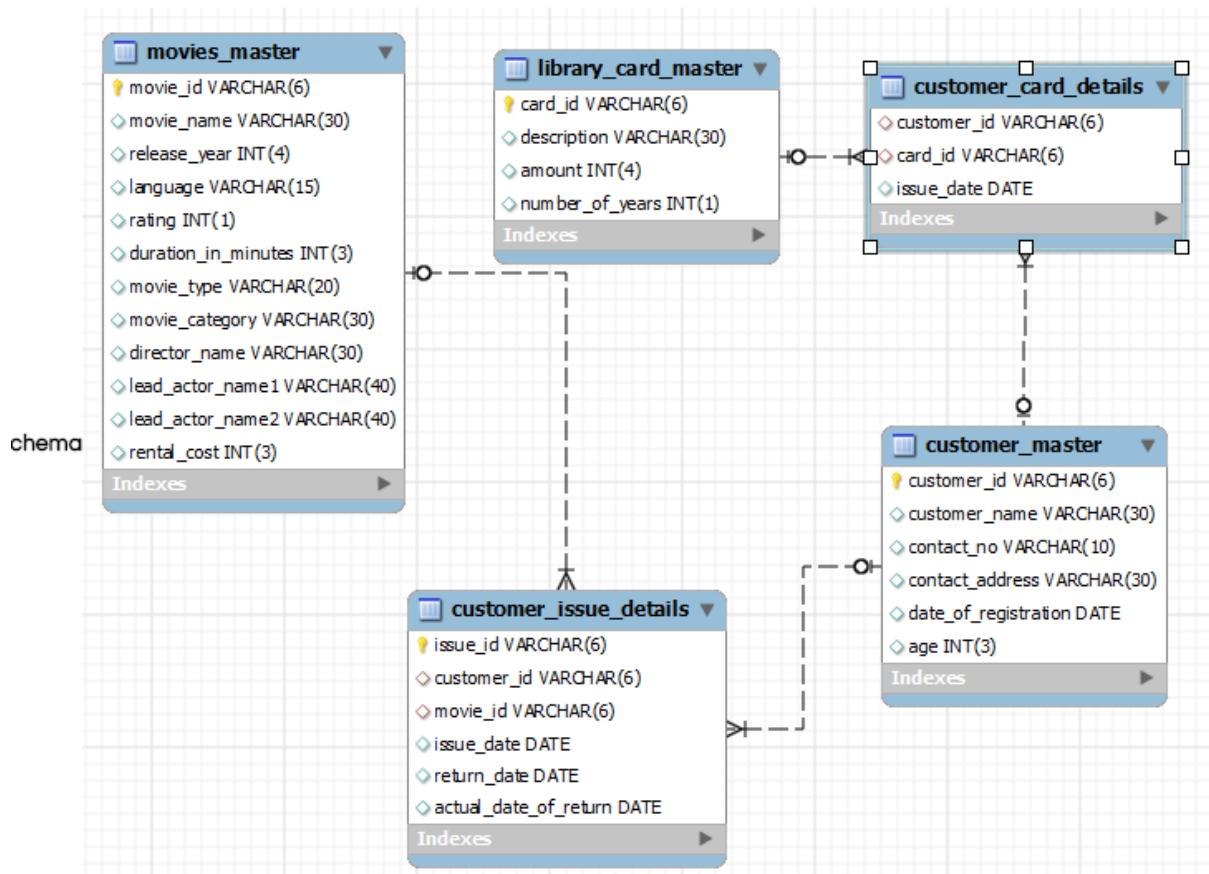
Display the records sorted in ascending order based on customer first name and then by account number.

```

SELECT firstname, customer_city, account_number
FROM customer_master cm INNER JOIN account_master am
ON cm.customer_number=am.customer_number
WHERE occupation !='Service' and occupation != 'Student' and occupation != 'Business'
order by firstname, account_number

```

FIRSTNAME	CUSTOMER_CITY	ACCOUNT_NUMBER
PARUL	DELHI	A00010



Video Management database queries:

1. Please follow instructions given below.

Write a query to display movie names and number of times that movie is issued to customers. Incase movies are never issued to customers display number of times as 0.

Display the details in sorted order based on number of times (in descending order) and then by movie name (in ascending order).

The Alias name for the number of movies issued is ISSUE_COUNT.

11 rows

```
select mm.movie_name, count(cid.issue_id) as ISSUE_COUNT  
from movies_master mm left outer join customer_issue_details  
cid on mm.movie_id=cid.movie_id group by mm.movie_name  
order by ISSUE_COUNT desc,mm.movie_name asc;
```

MOVIE_NAME	ISSUE_COUNT
DIE HARD	4
GONE WITH THE WIND	3
CASABLANCA	2
SHAUN OF THE DEAD	2
THE DARK KNIGHT	2
TITANIC	2
INCEPTION	1
OFFICE SPACE	1
THE MATRIX	1
YOUNG FRANKENSTEIN	1
THE NOTEBOOK	0

2. Please follow instructions given below.

Write a query to display id, name, age, contact no of customers whose age is greater than 25 and and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

4 rows

```

select customer_id,customer_name,age,
ifnull(concat('+91-',substring(contact_no,1,3),'-',substring(contact_no,4,3),'-',
',substring(contact_no,7,4)),'N/A')
as CONTACT_ISD from customer_master where age>25 and
year(date_of_registration)=2012
order by age,customer_name;

```

CUSTOMER_ID	CUSTOMER_NAME	AGE	CONTACT_ISD
C00007	GEETHA REDDY	30	+91-897-616-7890
C00005	SHIV PRASAD	30	N/A
C00002	AGNESH	35	+91-892-315-6781
C00004	RAJIB MITRA	45	+91-983-035-6781

3. Please follow instructions given below.

Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order.

Hint: Use NO_OF_MOVIES as alias name for number of movies.

3 rows

Ans:

```

select movie_category,count(movie_id) as NO_OF_MOVIES from movies_master group
by movie_category
order by NO_OF_MOVIES desc,movie_category asc;

```

MOVIE_CATEGORY	NO_OF_MOVIES
ACTION	4
ROMANCE	4
COMEDY	3

4. Please follow instructions given below.

Write a query to display the number of customers having card with description "Gold card".

 Hint: Use CUSTOMER_COUNT as alias name for number of customers

1 row

```

select count(ccd.customer_id) as CUSTOMER_COUNT from customer_card_details ccd
join
library_card_master lcd on ccd.card_id= lcd.card_id where lcd.description='Gold Card';

```

CUSTOMER_COUNT
2

4. Please follow instructions given below.

Write a query to display the customer id, customer name, year of registration, library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order.

Use REGISTERED_YEAR as alias name for year of registration.

5 rows

```
select cm.customer_id,cm.customer_name,year(cm.date_of_registration) as
REGISTERED_YEAR,ccd.card_id,ccd.issue_date
from customer_master cm join customer_card_details ccd on cm.customer_id=ccd.customer_id
order by cm.customer_name desc;
```

CUSTOMER_ID	CUSTOMER_NAME	REGISTERED_YEAR	CARD_ID	ISSUE_DATE
C00003	T RAMACHANDRAN	2012	CRD002	2012-11-02
C00005	SHIV PRASAD	2012	CRD003	2012-12-26
C00004	RAJIB MITRA	2012	CRD003	2012-11-21
C00001	NITIN	2012	CRD001	2012-10-15
C00002	AGNESH	2012	CRD002	2012-12-01

5. Please follow instructions given below.

Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer.

Display the records sorted in ascending order based on customer name.

2 rows

```
select cid.issue_id,cid.customer_id,cm.customer_name from customer_issue_details cid
join
customer_master cm on cid.customer_id=cm.customer_id where cm.customer_name like
'R%'
and cid.actual_date_return>cid.return_date order by cm.customer_name;
```

ISSUE_ID	CUSTOMER_ID	CUSTOMER_NAME

I00008	C00010	RAGHAV SINGH
I00007	C00004	RAJIB MITRA

6. Please follow instructions given below.

Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered.

For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details.

AMOUNT_DOLLAR = amount/52.42 and round it to zero decimal places and display as \$Amount. Example Assume 500 is the amount then dollar value will be \$10.

Hint: Use AMOUNT_DOLLAR as alias name for amount in dollar.

Display the records in ascending order based on customer name.

```
SELECT ccd.customer_id, customer_name, ccd.card_id,
description,concat('$',round(amount/52.42,0)) AMOUNT_DOLLAR FROM
customer_master cm INNER JOIN customer_card_details ccd ON
cm.customer_id=ccd.customer_id INNER JOIN library_card_master lcm ON
ccd.card_id=lcm.card_id INNER JOIN customer_issue_details cid ON cid.customer_id =
cm.customer_id WHERE cm.date_of_registration=cid.issue_date order by
customer_name;
```

CUSTOMER_ID	CUSTOMER_NAME	CARD_ID	DESCRIPTION	AMOUNT_DOLLAR
C00001	NITIN	CRD001	SILVER CARD	\$19
C00004	RAJIB MITRA	CRD003	PLATINUM CARD	\$57
C00003	T RAMACHANDRAN	CRD002	GOLD CARD	\$38

7. Please follow instructions given below.

Write a query to display the customer id, customer name, contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'.

Display customer name in upper case. Hint: Use CUSTOMER_NAME as alias name for customer name. Display the details sorted in ascending order based on customer name.

```
SELECT customer_id , upper(customer_name)
CUSTOMER_NAME,contact_no,contact_address FROM customer_master WHERE
customer_id NOT IN ( select customer_id from customer_card_details ) AND customer_id
```

IN (SELECT customer_id from customer_issue_details) and contact_address like '%Nagar' order by customer_name ;

CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO	CONTACT_ADDRESS
C00010	RAGHAV SINGH	9675167890	A/6 NEHRU JAWAHAR NAGAR

8. Please follow instructions given below.

Write a query to display the movie id, movie name, release year, director name of movies acted by the leadactor1 who acted maximum number of movies .Display the records sorted in ascending order based on movie name.

- **select movie_id,movie_name , release_year ,director_name from movies gmaster where lead_actor_name1 in(select lead_actor_name1 from(select lead_actor_name1,count(movie_id) ct from movies_master group by lead_actor_name1)t where t.rct>=all(select count(movie_id) from movies_master group by lead_actor_name1))order by movie_name;**

MOVIE_ID	MOVIE_NAME	RELEASE_YEAR	DIRECTOR_NAME
M00004	INCEPTION	2010	CHRISTOPHER NOLAN
M00011	TITANIC	1997	JAMES CAMERON

9. Please follow instructions given below.

Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0.
 Hint: Use MOVIE_COUNT as alias name for number of movies issued.

11 rows

select cm.customer_name,count(cid.movie_id) as MOVIE_COUNT from customer_master cm left join customer_issue_details cid on cm.customer_id=cid.customer_id group by cm.customer_name order by cm.customer_name;

CUSTOMER_NAME	MOVIE_COUNT
AGNESH	3

AJAY GHOSH	0
GEETHA REDDY	0
NITIN	2
RAGHAV SINGH	1
RAJ SEKHANRAN	1
RAJAN PILLAI	0
RAJIB MITRA	4
RIA NATRAJAN	0
SHIV PRASAD	0
T RAMACHANDRAN	8

10. Please follow instructions given below.

Write a query to display serial number, issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number.

Serial number can be generated from the issue id , that is last two characters of issue id is the serial number.

For Example Assume the issue id is I00005 then the serial number is 05

Hint: Alias name for serial number is 'SERIAL_NO'

19 rows

```
select substring(cid.issue_id,5,2) as
SERIAL_NO,cid.issue_id,cid.customer_id,cm.customer_name,mm.movie_id,mm.movie_n
ame
from customer_issue_details cid join customer_master cm on
cm.customer_id=cid.customer_id
join movies_master mm on cid.movie_id=mm.movie_id group by
SERIAL_NO,cid.customer_id,mm.movie_id
order by SERIAL_NO;
```

SERIAL_NO	ISSUE_ID	CUSTOMER_ID	CUSTOMER_NAME	MOVIE_ID	MOVIE_NAME
01	I00001	C00001	NITIN	M00001	DIE HARD

02	I00002	C00002	AGNESH	M00002	THE DARK KNIGHT
03	I00003	C00002	AGNESH	M00002	THE DARK KNIGHT
04	I00004	C00003	T RAMACHANDRAN	M00003	THE MATRIX
05	I00005	C00003	T RAMACHANDRAN	M00004	INCEPTION
06	I00006	C00003	T RAMACHANDRAN	M00005	OFFICE SPACE
07	I00007	C00004	RAJIB MITRA	M00006	YOUNG FRANKENSTEIN
08	I00008	C00010	RAGHAV SINGH	M00008	CASABLANCA
09	I00009	C00011	RAJ SEKHANRAN	M00010	GONE WITH THE WIND
10	I00010	C00004	RAJIB MITRA	M00007	SHAUN OF THE DEAD
11	I00011	C00004	RAJIB MITRA	M00007	SHAUN OF THE DEAD
12	I00012	C00001	NITIN	M00001	DIE HARD
13	I00013	C00003	T RAMACHANDRAN	M00001	DIE HARD
14	I00014	C00003	T RAMACHANDRAN	M00010	GONE WITH THE WIND
15	I00015	C00003	T RAMACHANDRAN	M00011	TITANIC
16	I00016	C00003	T RAMACHANDRAN	M00011	TITANIC
17	I00017	C00003	T RAMACHANDRAN	M00008	CASABLANCA
18	I00018	C00002	AGNESH	M00010	GONE WITH THE WIND
19	I00019	C00004	RAJIB MITRA	M00001	DIE HARD



11. Please follow instructions given below.

Write a query to display the issue id, issue date, customer id, customer name and contact number for videos that are issued in the year 2013. Display the records in descending order based on issue date of the video.

7 rows

```
select cid.issue_id,cid.issue_date,cid.customer_id,cm.customer_name,cm.contact_no  
from customer_issue_details cid join customer_master cm on  
cid.customer_id=cm.customer_id  
where year(issue_date)=2013 group by issue_id,issue_date,customer_id order by  
issue_date desc;
```

ISSUE_ID	ISSUE_DATE	CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO
I00012	2013-11-28	C00001	NITIN	9830354218
I00017	2013-04-15	C00003	T RAMACHANDRAN	9831289761
I00009	2013-03-16	C00011	RAJ SEKHANRAN	8423178906
I00016	2013-03-05	C00003	T RAMACHANDRAN	9831289761
I00008	2013-03-02	C00010	RAGHAV SINGH	9675167890
I00015	2013-02-03	C00003	T RAMACHANDRAN	9831289761
I00014	2013-01-02	C00003	T RAMACHANDRAN	9831289761

12. Please follow instructions given below.

Write a query to display movie id, movie name and actor names of movies which are not issued to any customers.
 Actors Name to be displayed in the below format. LEAD_ACTOR_ONE space ampersant space LEAD_ACTOR_TWO.

Example: Assume lead actor one's name is "Jack Tomson" and Lead actor two's name is "Maria" then Actors name will be "Jack Tomsom & Maria" Hint: Use ACTORS as alias name for actors name.
 Display the records in ascending order based on movie name.

1 row

```
select movie_id,movie_name,concat(lead_actor_name1,' & ',lead_actor_name2) as  
ACTORS  
from movies_master where movie_id  
not in (select movie_id from customer_issue_details) order by  
movie_name;
```

MOVIE_ID	MOVIE_NAME	ACTORS

M00009	THE NOTEBOOK	RYAN GOSLING & RACHEL MCADAMS
--------	--------------	-------------------------------

13. Please follow instructions given below.

Write a query to display the director's name, movie name and lead_actor_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR_NAME as alias name for director name column Display the records sorted in ascending order based on director_name and then by movie_name in descending order.

2 rows

```
SELECT upper(director_name) DIRECTOR_NAME,movie_name,lead_actor_name1 FROM movies_master WHERE director_name in (SELECT director_name FROM movies_master GROUP BY director_name HAVING count(movie_id)>1) order by director_name, movie_name desc;
```

DIRECTOR_NAME	MOVIE_NAME	LEAD_ACTOR_NAME1
CHRISTOPHER NOLAN	THE DARK KNIGHT	CHRISTIAN BALE
CHRISTOPHER NOLAN	INCEPTION	LEONARDO DICAPRIO

14. Please follow instructions given below.

Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number.
 Hint: Use NO_OF_CUSTOMERS as alias name for number of customers.

1 row

```
select count(customer_id) as NO_OF_CUSTOMERS from customer_master where year(date_of_registration)=2012 and contact_no != 'NULL'
```

NO_OF_CUSTOMERS
6

15. Please follow instructions given below.

Write a query to display the customer's name, contact number, library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in

ascending order based on customer name. Hint: Use CONTACT_DETAILS as alias name for customer contact.

11 rows

```
select cm.customer_name,ifnull(cm.contact_no,cm.contact_add) as  
CONTACT_DETAILS,lcd.card_id,lcd.description from customer_master cm  
left join customer_card_details ccd on cm.customer_id=ccd.customer_id  
left join library_card_master lcd on ccd.card_id=lcd.card_id group by  
customer_name,description,CONTACT_DETAILS  
order by customer_name;
```

CUSTOMER_NAME	CONTACT_DETAILS	CARD_ID	DESCRIPTION
AGNESH	8923156781	CRD002	GOLD CARD
AJAY GHOSH	8763478901	NULL	NULL
GEETHA REDDY	8976167890	NULL	NULL
NITIN	9830354218	CRD001	SILVER CARD
RAGHAV SINGH	9675167890	NULL	NULL
RAJ SEKHANRAN	8423178906	NULL	NULL
RAJAN PILLAI	A 1/66 KODAMBAKKAM	NULL	NULL
RAJIB MITRA	9830356781	CRD003	PLATINUM CARD
RIA NATRAJAN	9856723190	NULL	NULL
SHIV PRASAD	2/2 PHASE II, JAWAHAR NAGAR	CRD003	PLATINUM CARD
T RAMACHANDRAN	9831289761	CRD002	GOLD CARD

16. Please follow instructions given below.

Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in descending order. For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john. Hint: Use NO_OF_TIMES as alias name for number of times

4 rows

```
select cm.customer_id,cm.customer_name,count(cid.movie_id) as NO_OF_TIMES from
customer_master
cm join customer_issue_details cid on cm.customer_id=cid.customer_id group by
customer_id,movie_id having
count(movie_id)>1 order by customer_name desc;
```

CUSTOMER_ID	CUSTOMER_NAME	NO_OF_TIMES
C00003	T RAMACHANDRAN	2
C00004	RAJIB MITRA	2
C00001	NITIN	2
C00002	AGNESH	2

17. Please follow instructions given below.

Write a query to display customer id, customer name, contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-456-2345" format.

Hint: Use NO_OF_MOVIES as alias name for number of movies column.

Hint: Use CONTACT_ISD as alias name for contact number.

Display the records sorted in ascending order based on customer name and then by movie category.

5 rows

```
select cid.customer_id,cm.customer_name,
concat('+91-',substring(cm.contact_no,1,3),'-',substring(cm.contact_no,4,3),'-',
substring(cm.contact_no,7,4)) as CONTACT_ISD,
mm.movie_category,count(mm.movie_category) as NO_OF_MOVIES from
customer_master
cm join customer_issue_details cid
on cm.customer_id=cid.customer_id join movies_master mm on
cid.movie_id=mm.movie_id
group by mm.movie_category,cm.customer_name having count(movie_category)>1
order by cm.customer_name,mm.movie_category;
```

CUSTOMER_ID	CUSTOMER_NAME	CONTACT_ISD	MOVIE_CATEGORY	NO_OF_MOVIES
C00002	AGNESH	+91-892-315-6781	ACTION	2

C00001	NITIN	+91-983-035-4218	ACTION	2
C00004	RAJIB MITRA	+91-983-035-6781	COMEDY	3
C00003	T RAMACHANDRAN	+91-983-128-9761	ACTION	3
C00003	T RAMACHANDRAN	+91-983-128-9761	ROMANCE	4

18. Please follow instructions given below.

Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies.

For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies.

Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.

3 rows

```
(select cm.customer_id,cm.customer_name from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id
having count(cid.issue_id) >= all (select count(cid.issue_id) from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id) order by
cm.customer_name)
union all
(select cm.customer_id,cm.customer_name from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id
having count(cid.issue_id) <= all (select count(cid.issue_id) from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id) order by
cm.customer_name)
```

CUSTOMER_ID	CUSTOMER_NAME
C00003	T RAMACHANDRAN

C00010	RAGHAV SINGH
C00011	RAJ SEKHANRAN

19. Please follow instructions given below.

Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once.

Hint: Use NO_OF_TIMES as alias name

Display the records in ascending order based on customer name.

1 row

```
select cm.customer_id,cm.customer_name,count(mm.movie_id) as NO_OF_TIMES from
customer_master cm
join customer_issue_details cid on cm.customer_id=cid.customer_id join
movies_master mm on cid.movie_id=mm.movie_id where mm.movie_category='comedy' group
by customer_id
order by customer_name>1;
```

CUSTOMER_ID	CUSTOMER_NAME	NO_OF_TIMES
C00004	RAJIB MITRA	3

20. Please follow instructions given below.

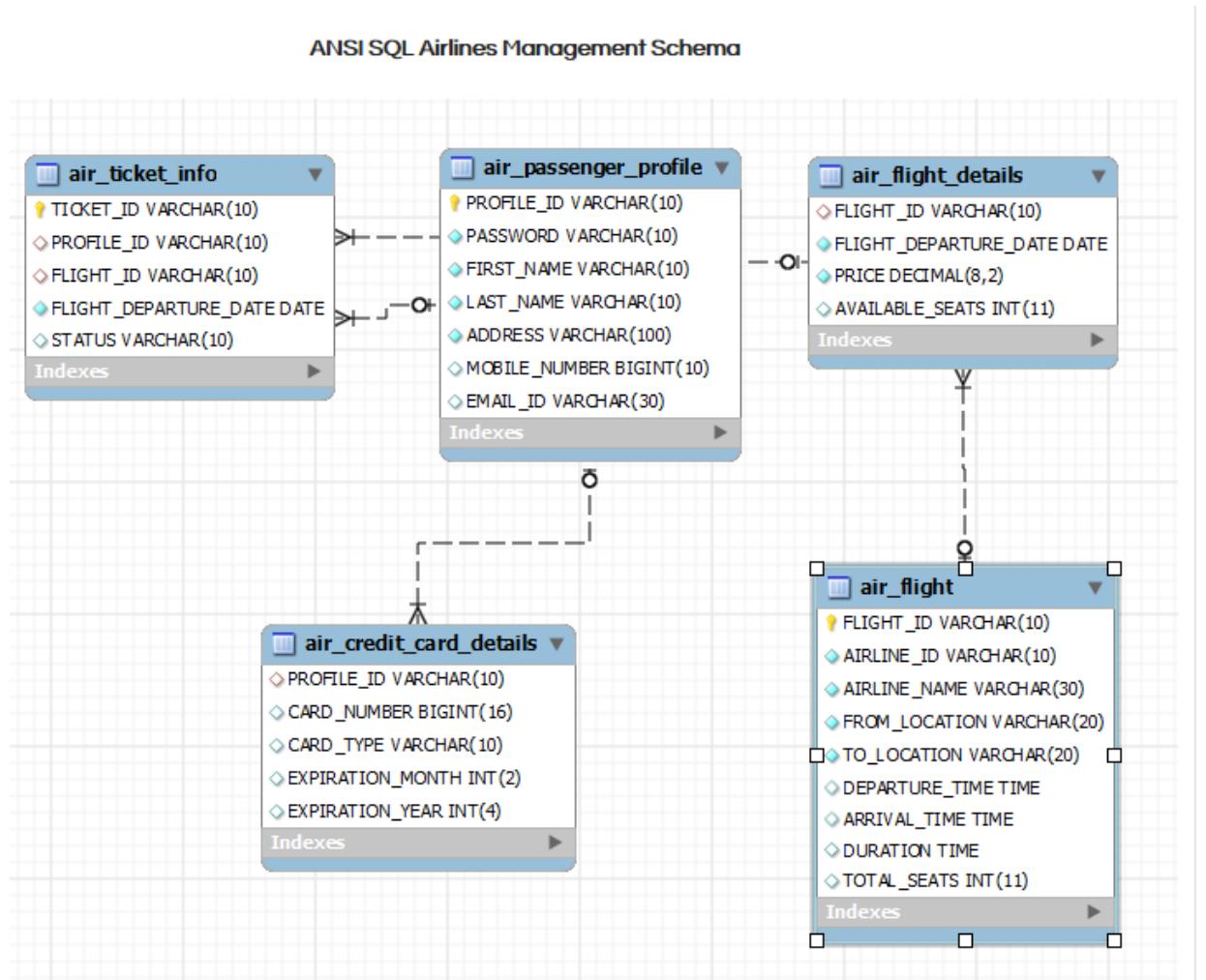
Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL_COST. Display the records sorted in ascending order based on customer id

6 rows

```
select cid.customer_id,sum(mm.rental_cost) as TOTAL_COST from customer_issue_details cid
join movies_master mm
on cid.movie_id=mm.movie_id group by customer_id order by customer_id;
```

	customer_id	TOTAL_COST
▶	C00001	200
	C00002	300
	C00003	1700
	C00004	400
	C00010	1000
	C00011	100

Airline Flight Management :



1. Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight_Id, From_Location, To_Location, Month Name as "Month_Name" and average price as "Average_Price".

Display the records sorted in ascending order based on flight id and then by Month Name.

15 rows

```
select f.flight_id,f.from_location,f.to_location,monthname(fd.flight_departure_date) as Month_name,avg(fd.price) as Average_price from air_flight f join air_flight_details fd on f.flight_id=fd.flight_id group by f.flight_id,Month_name order by f.flight_id,Month_name;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	MONTH_NAME	AVERAGE_PRICE
1011	HYDERABAD	CHENNAI	APRIL	4614.000000
1011	HYDERABAD	CHENNAI	MAY	3855.500000

1262	HYDERABAD	CHENNAI	MAY	3444.500000
1265	CHENNAI	HYDERABAD	APRIL	4086.000000
1265	CHENNAI	HYDERABAD	MAY	3303.666667
289	CHENNAI	KOCHI	MAY	3257.750000
3004	BENGALURU	CHENNAI	MAY	3319.666667
3013	CHENNAI	BENGALURU	MAY	3257.750000
3148	CHENNAI	BENGALURU	JUNE	2773.000000
3148	CHENNAI	BENGALURU	MAY	3052.000000
3241	CHENNAI	KOCHI	MAY	3303.666667
3244	KOCHI	CHENNAI	MAY	3371.500000
3307	BENGALURU	CHENNAI	MAY	3309.000000
916	CHENNAI	HYDERABAD	APRIL	4086.000000
916	CHENNAI	HYDERABAD	MAY	3570.666667

2. Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile_id, customer's first_name, Address and Number of tickets booked as "No_of_Tickets"

Display the records sorted in ascending order based on customer's first name.

1 row

```
select apf.profile_id,apf.first_name,apf.address,count(ati.ticket_id) as No_of_Tickets
from air_passenger_profile apf
join air_ticket_info ati on apf.profile_id=ati.profile_id group by apf.profile_id having
count(ati.ticket_id) <=all
(select count(ati.ticket_id) from air_passenger_profile apf
join air_ticket_info ati on apf.profile_id=ati.profile_id group by apf.profile_id) order by
```

first_name;

PROFILE_ID	FIRST_NAME	ADDRESS	NO_OF_TICKETS
PFL008	GANESH	45 3RD ST, HYDERABAD-24	1

3. Write a query to display the number of flight services between locations in a month. The Query should display From_Location, To_Location, Month as "Month_Name" and number of flight services as "No_of_Services".

Hint: The Number of Services can be calculated from the number of scheduled departure dates of a flight.

The records should be displayed in ascending order based on From_Location and then by To_Location and then by month name

9 rows

```
select af.from_location, af.to_location, monthname(afd.flight_departure_date)
as Month_Name,
count(afd.flight_departure_date) as No_of_Services from air_flight af join
air_flight_details afd
on af.flight_id=afd.flight_id group by
af.from_location, af.to_location, month_name order by
from_location, to_location, month_name;
```

FROM_LOCATION	TO_LOCATION	MONTH_NAME	NO_OF_SERVICES
BENGALURU	CHENNAI	MAY	7
CHENNAI	BENGALURU	JUNE	1
CHENNAI	BENGALURU	MAY	6
CHENNAI	HYDERABAD	APRIL	2
CHENNAI	HYDERABAD	MAY	6
CHENNAI	KOCHI	MAY	7
HYDERABAD	CHENNAI	APRIL	1
HYDERABAD	CHENNAI	MAY	4

KOCHI	CHENNAI	MAY	2
-------	---------	-----	---

4. Write a query to display the customer(s) who has/have booked maximum number of tickets in ABC Airlines. The Query should display profile_id, customer's first_name, Address and Number of tickets booked as "No_of_Tickets"

Display the records in ascending order based on customer's first name.

1 row

```
select app.profile_id,app.first_name,app.address,count(ati.ticket_id) as No_of_Tickets
from air_passenger_profile app
join air_ticket_info ati on app.profile_id=ati.profile_id join air_flight af on ati.flight_id=af.flight_id
where af.airline_name= 'ABC Airlines' group by app.profile_id
having count(ati.ticket_id) >= all (select count(ati.ticket_id) from air_passenger_profile app
join air_ticket_info ati on app.profile_id=ati.profile_id join air_flight af on ati.flight_id=af.flight_id
where af.airline_name= 'ABC Airlines' group by app.profile_id) order by app.first_name;
```

PROFILE_ID	FIRST_NAME	ADDRESS	NO_OF_TICKETS
PFL009	RAM	119 2ND CROSS ST,ERNAKULAM-12	8

5. Write a query to display the number of tickets booked from Chennai to Hyderabad. The Query should display passenger profile_id,first_name,last_name, Flight_Id , Departure_Date and number of tickets booked as "No_of_Tickets".

Display the records sorted in ascending order based on profile id and then by flight id and then by departure date.

3 rows

```
select
ati.profile_id,app.first_name,app.last_name,ati.flight_id,ati.flight_departure_date,count(ati.ticket_id)
as No_of_Tickets from air_ticket_info ati join air_passenger_profile app on ati.profile_id=
app.profile_id join air_flight af on ati.flight_id=af.flight_id
where af.from_location='chennai' and af.to_location='hyderabad' group by ati.profile_id,
ati.flight_id,ati.flight_departure_date order by
ati.profile_id,
ati.flight_id,ati.flight_departure_date;
```

PROFILE_ID	FIRST_NAME	LAST_NAME	FLIGHT_ID	FLIGHT_DEPARTURE_DATE	NO_OF_TICKETS
------------	------------	-----------	-----------	-----------------------	---------------

PFL001	LATHA	SANKAR	1265	2013-04-29	1
PFL004	AARTHI	RAMESH	1265	2013-05-29	1
PFL005	SIVA	KUMAR	916	2013-05-06	2

6. Write a query to display flight id, from location, to location and ticket price of flights whose departure is in the month of april.

3 rows

Display the records sorted in ascending order based on flight id and then by from location.

```
select af.flight_id, af.from_location, af.to_location, afd.price from
air_flight af
join air_flight_details afd on af.flight_id=afd.flight_id
where monthname(afd.flight_departure_date)='april' order by
flight_id, from_location;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	PRICE
1011	HYDERABAD	CHENNAI	4614.00
1265	CHENNAI	HYDERABAD	4086.00
916	CHENNAI	HYDERABAD	4086.00

7. Write a query to display the average cost of the tickets in each flight on all scheduled dates.

The query should display flight_id, from_location, to_location and Average price as "Price".

Display the records sorted in ascending order based on flight id and then by from_location and then by to_location.

11 rows

```
select af.flight_id, af.from_location, af.to_location, avg(afd.price)
from air_flight af join air_flight_details afd
on af.flight_id=afd.flight_id group by af.flight_id, af.from_location, af.to_location
order by af.flight_id, af.from_location, af.to_location;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	PRICE
1011	HYDERABAD	CHENNAI	4108.333333
1262	HYDERABAD	CHENNAI	3444.500000

1265	CHENNAI	HYDERABAD	3499.250000
289	CHENNAI	KOCHI	3257.750000
3004	BENGALURU	CHENNAI	3319.666667
3013	CHENNAI	BENGALURU	3257.750000
3148	CHENNAI	BENGALURU	2959.000000
3241	CHENNAI	KOCHI	3303.666667
3244	KOCHI	CHENNAI	3371.500000
3307	BENGALURU	CHENNAI	3309.000000
916	CHENNAI	HYDERABAD	3699.500000

8. Write a query to display the customers who have booked tickets from Chennai to Hyderabad. The query should display profile_id, customer_name (combine first_name & last_name with comma in b/w), address of the customer.

Give an alias to the name as customer_name.

Hint: Query should fetch unique customers irrespective of multiple tickets booked.

Display the records sorted in ascending order based on profile id.

3 rows

```
select app.profile_id, concat(app.first_name,',',app.last_name) as customer_name,app.address
from air_passenger_profile app join air_ticket_info ati on app.profile_id=ati.profile_id
join air_flight af on ati.flight_id=af.flight_id where af.from_location='chennai'
and af.to_location='hyderabad' group by app.profile_id order by app.profile_id;
```

ROFILE_ID	CUSTOMER_NAME	ADDRESS
PFL001	LATHA,SANKAR	123 BROAD CROSS ST,CHENNAI-48
PFL004	AARTHI,RAMESH	343 6TH STREET,HYDERABAD-76
PFL005	SIVA,KUMAR	125 8TH STREET,CHENNAI-46

9. Write a query to display profile id of the passenger(s) who has/have booked maximum number of tickets.

In case of multiple records, display the records sorted in ascending order based on profile id.

2 rows

```
select profile_id from air_ticket_info group by profile_id having count(ticket_id) >= all (select count(ticket_id) from air_ticket_info group by profile_id) order by profile_id;
```

PROFILE_ID
PFL002
PFL007

10. Write a query to display the total number of tickets as "No_of_Tickets" booked in each flight in ABC Airlines. The Query should display the flight_id, from_location, to_location and the number of tickets.

Display only the flights in which atleast 1 ticket is booked.

Display the records sorted in ascending order based on flight id.

7 rows

```
select af.flight_id, af.from_location, af.to_location, count(ati.ticket_id) as No_of_Tickets  
from air_flight af join air_ticket_info ati on af.flight_id=ati.flight_id  
group by af.flight_id having count(ati.ticket_id) >= 1;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_TICKETS
1011	HYDERABAD	CHENNAI	4
1262	HYDERABAD	CHENNAI	1
1265	CHENNAI	HYDERABAD	2
3004	BENGALURU	CHENNAI	3
3148	CHENNAI	BENGALURU	7
3244	KOCHI	CHENNAI	7
916	CHENNAI	HYDERABAD	2

11. Write a query to display the no of services offered by each flight and the total price of the services. The Query should display flight_id, number of services as "No_of_Services" and the cost as "Total_Price" in the same order.

Order the result by Total Price in descending order and then by flight_id in descending order.

Hint: The number of services can be calculated from the number of scheduled departure dates of the flight

11 rows

```
select af.flight_id, count(afd.flight_departure_date) as No_of_Services, sum(afd.price) as Total_Price from air_flight af join air_flight_details afd on af.flight_id=afd.flight_id group by flight_id order by total_price desc,flight_id desc;
```

FLIGHT_ID	NO_OF_SERVICES	TOTAL_PRICE
916	4	14798.00
1265	4	13997.00
3307	4	13236.00
3013	4	13031.00
289	4	13031.00
1011	3	12325.00
3004	3	9959.00
3241	3	9911.00
3148	3	8877.00
1262	2	6889.00
3244	2	6743.00

12. Write a query to display the number of passengers who have travelled in each flight in each scheduled date. The Query should display flight_id, flight_departure_date and the number of passengers as "No_of_Passengers" in the same order.

Display the records sorted in ascending order based on flight id and then by flight departure date.

9 rows

```
SELECT flight_id,  
       flight_departure_date,  
       COUNT(ticket_id) AS No_of_Passengers  
FROM air_ticket_info  
GROUP BY flight_id,  
        flight_departure_date  
ORDER BY flight_id, flight_departure_date;
```

FLIGHT_ID	FLIGHT_DEPARTURE_DATE	NO_OF_PASSENGERS
1011	2013-05-09	4
1262	2013-05-20	1
1265	2013-04-29	1
1265	2013-05-29	1
3004	2013-05-02	3
3148	2013-05-21	2
3148	2013-06-01	5
3244	2013-05-03	7
916	2013-05-06	2

13. Write a query to display profile id of passenger(s) who booked minimum number of tickets.

In case of multiple records, display the records sorted in ascending order based on profile id.

1 row

```
select profile_id from air_ticket_info group by profile_id having count(profile_id) <= all  
(select count(profile_id) from air_ticket_info group by profile_id) order by profile_id;
```

PROFILE_ID

PFL008

14. Write a query to display unique passenger profile id, first name, mobile number and email address of passengers who booked ticket to travel from HYDERABAD to CHENNAI.

Display the records sorted in ascending order based on profile id.

4 rows

```
select distinct ati.profile_id,app.first_name,app.mobile_number,app.email_id  
from air_ticket_info  
ati join air_passenger_profile app on ati.profile_id=app.profile_id join air_flight af  
on ati.flight_id=af.flight_id  
where af.from_location='hyderabad' and af.to_location='chennai' order by profile_id;
```

PROFILE_ID	FIRST_NAME	MOBILE_NUMBER	EMAIL_ID
PFL001	LATHA	9876543210	LATHA@GMAIL.COM
PFL004	AARTHI	9595652530	AARTHI@GMAIL.COM
PFL005	SIVA	9884416986	SIVA@GMAIL.COM
PFL008	GANESH	9375237890	GANESH@GMAIL.COM

15. Write a query to intimate the passengers who are boarding Chennai to Hyderabad Flight on 6th May 2013 stating the delay of 1hr in the departure time. The Query should display the passenger's profile_id, first_name, last_name, flight_id, flight_departure_date, actual departure time, actual arrival time, delayed departure time as "Delayed_Departure_Time", delayed arrival time as "Delayed_Arrival_Time". Hint: Distinct Profile ID should be displayed irrespective of multiple tickets booked by the same profile.

Display the records sorted in ascending order based on passenger's profile id.

1 row

```
select distinct app.profile_id,app.first_name,app.last_name,ati.flight_id,ati.flight_departure_date,  
af.departure_time,af.arrival_time, af.departure_time ,ADDTIME(af.departure_time,'1:00:00') as  
Delayed_Departure_Time,  
ADDTIME(af.arrival_time,'1:00:00') as Delayed_Arrival_Time from air_passenger_profile app  
join air_ticket_info ati on app.profile_id=ati.profile_id join air_flight af on  
ati.flight_id=af.flight_id where ati.flight_departure_date='2013-05-06' order by app.profile_id;
```

PROFILE_ID	FIRST_NAME	LAST_NAME	FLIGHT_ID	FLIGHT_DATE	DEPARTURE_TIME	ARRIVAL_TIME
PFL008	GANESH		AF101	2013-05-06	10:00:00	11:00:00

ID	NAME		ID	DEPARTURE DATE		TIME
PFL005	SIVA	KUMAR	916	2013-05-06	19:55:00	21:00:00

DELAYED_DEPARTURE_TIME	DELAYED_ARRIVAL_TIME
20:55:00	22:00:00

16. Write a query to display the number of tickets as "No_of_Tickets" booked by Kochi Customers. The Query should display the Profile_Id, First_Name, Base_Location and number of tickets booked.

Hint: Use String functions to get the base location of customer from their Address and give alias name as "Base_Location"

Display the records sorted in ascending order based on customer first name.

2 rows

```
select
ap.profile_id, ap.first_name, substring_index(substring_index(ap.address,',',-1),'-',1)
as base_location, count(at.ticket_id) as No_of_Tickets from
air_passenger_profile ap join air_ticket_info at
on at.profile_id=ap.profile_id
where substring_index(substring_index(ap.address,',',-1),'-',1) ='kochi'
group by ap.profile_id order by first_name
```

PROFILE_ID	FIRST_NAME	BASE_LOCATION	NO_OF_TICKETS
PFL003	AMIT	KOCHI	3
PFL006	RAMESH	KOCHI	4

17. Write a query to display the flight_id, from_location, to_location, number of Services as "No_of_Services" offered in the month of May.

Hint: The number of services can be calculated from the number of scheduled departure dates of the flight

Display the records sorted in ascending order based on flight id.

11 rows

```
select af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date)
as No_of_Services from air_flight af join air_flight_details afd
on af.flight_id=afd.flight_id where month(afd.flight_departure_date)='05'
group by flight_id order by flight_id;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_SERVICES
1011	HYDERABAD	CHENNAI	2
1262	HYDERABAD	CHENNAI	2
1265	CHENNAI	HYDERABAD	3
289	CHENNAI	KOCHI	4
3004	BENGALURU	CHENNAI	3
3013	CHENNAI	BENGALURU	4
3148	CHENNAI	BENGALURU	2
3241	CHENNAI	KOCHI	3
3244	KOCHI	CHENNAI	2
3307	BENGALURU	CHENNAI	4
916	CHENNAI	HYDERABAD	3

18. Write a query to display profile id, last name, mobile number and email id of passengers whose base location is chennai.

Display the records sorted in ascending order based on profile id.

2 rows

```
select profile_id,last_name,mobile_number,email_id from air_passenger_profile where
substring_index(substring_index(address,',',-1),'-',1)='chennai'
order by profile_id;
```

PROFILE_ID	LAST_NAME	MOBILE_NUMBER	EMAIL_ID
PFL001	SANKAR	9876543210	LATHA@GMAIL.COM
PFL005	KUMAR	9884416986	SIVA@GMAIL.COM

18. Write a query to display number of flights between 6.00 AM and 6.00 PM from chennai. Hint
Use FLIGHT_COUNT as alias name.

1 row

```
select count(flight_id) as FLIGHT_COUNT from air_flight where departure_time between  
'6:00:00' and '18:00:00' and from_location='chennai';;
```

FLIGHT_COUNT
3

19. Write a query to display unique profile id, first name , email id and contact number of passenger(s) who travelled on flight with id 3148. Display the records sorted in ascending order based on first name.

2 rows

```
select distinct app.profile_id,app.first_name,app.email_id,app.mobile_number from  
air_passenger_profile app  
join air_ticket_info ati on app.profile_id=ati.profile_id  
where ati.flight_id= 3148 group by app.first_name order by app.first_name;
```

PROFILE_ID	FIRST_NAME	EMAIL_ID	MOBILE_NUMBER
PFL002	ARUN	ARUN@AOL.COM	8094564243
PFL007	GAYATHRI	GAYATHRI@GMAIL.COM	8073245678

20. Write a query to display the flights available in Morning, AfterNoon, Evening & Night. The Query should display the Flight_Id, From_Location, To_Location , Departure_Time, time of service as "Time_of_Service".

Time of Service should be calculated as: From 05:00:01 Hrs to 12:00:00 Hrs - Morning,
12:00:01 to 18:00:00 Hrs -AfterNoon, 18:00:01 to 24:00:00 - Evening and 00:00:01 to 05:00:00 - Night

Display the records sorted in ascending order based on flight id.

11 rows

```
select flight_id,from_location,to_location,departure_time,  
case when departure_time between '05:00:01' and '12:00:00' then 'Morning'  
when departure_time between '12:00:01' and '18:00:00' then 'Afternoon'  
when departure_time between '18:00:01' and '24:00:00' then 'Evening'  
when departure_time between '00:00:01' and '05:00:00' then 'Night'  
end as Time_of_Service  
from air_flight order by flight_id;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	DEPARTURE_TIME	TIME_OF_SERVICE
1011	HYDERABAD	CHENNAI	12:30:00	AFTERNOON
1262	HYDERABAD	CHENNAI	06:00:00	MORNING
1265	CHENNAI	HYDERABAD	21:25:00	EVENING
289	CHENNAI	KOCHI	08:40:00	MORNING
3004	BENGALURU	CHENNAI	09:05:00	MORNING
3013	CHENNAI	BENGALURU	07:40:00	MORNING
3148	CHENNAI	BENGALURU	20:15:00	EVENING
3241	CHENNAI	KOCHI	10:40:00	MORNING
3244	KOCHI	CHENNAI	21:10:00	EVENING
3307	BENGALURU	CHENNAI	18:45:00	EVENING
916	CHENNAI	HYDERABAD	19:55:00	EVENING

21. Please follow instructions given below.

Write a query to display flight id,departure date,flight type of all flights. Flight type can be identified based on the following rules : if ticket price is less than 3000 then 'AIR PASSENGER', ticket price between 3000 and less than 4000 'AIR BUS' and ticket price between

4000 and greater than 4000 then 'EXECUTIVE PASSENGER'. Hint use FLIGHT_TYPE as alias name.

Display the records sorted in ascending order based on flight_id and then by departure date.

36 rows

```
select flight_id,flight_departure_date,  
case when price<3000 then 'AIR PASSENGER'  
when price>=3000 and price<=4000 then 'AIR BUS'  
when price>4000 then 'EXECUTIVE PASSENGER'  
end as FLIGHT_TYPE from air_flight_details order by flight_id,flight_departure_date;
```

FLIGHT_ID	FLIGHT_DEPARTURE_DATE	FLIGHT_TYPE
1011	2013-04-30	EXECUTIVE PASSENGER
1011	2013-05-09	EXECUTIVE PASSENGER
1011	2013-05-21	AIR BUS
1262	2013-05-20	AIR BUS
1262	2013-05-29	AIR BUS
1265	2013-04-29	EXECUTIVE PASSENGER
1265	2013-05-14	AIR BUS
1265	2013-05-18	EXECUTIVE PASSENGER
1265	2013-05-29	AIR PASSENGER
289	2013-05-06	AIR BUS
289	2013-05-08	AIR BUS
289	2013-05-20	AIR BUS
289	2013-05-31	AIR PASSENGER

3004	2013-05-02	AIR BUS
3004	2013-05-19	AIR BUS
3004	2013-05-24	AIR BUS
3013	2013-05-04	AIR BUS
3013	2013-05-06	AIR BUS
3013	2013-05-22	AIR BUS
3013	2013-05-30	AIR PASSENGER
3148	2013-05-16	AIR BUS
3148	2013-05-21	AIR BUS
3148	2013-06-01	AIR PASSENGER
3241	2013-05-01	EXECUTIVE PASSENGER
3241	2013-05-13	AIR BUS
3241	2013-05-27	AIR PASSENGER
3244	2013-05-03	AIR BUS
3244	2013-05-15	AIR BUS
3307	2013-05-03	AIR BUS
3307	2013-05-03	AIR BUS
3307	2013-05-23	AIR BUS
3307	2013-05-29	AIR BUS

916	2013-04-28	EXECUTIVE PASSENGER
916	2013-05-01	EXECUTIVE PASSENGER
916	2013-05-06	AIR BUS
916	2013-05-12	AIR BUS

22. Please follow instructions given below.

Write a query to display the credit card type and no of credit cards used on the same type.

Display the records sorted in ascending order based on credit card type.

Hint: Use CARD_COUNT AS Alias name for no of cards.

3 rows

```
SELECT CARD_TYPE, count(card_type) CARD_COUNT FROM air_credit_card_details group by CARD_TYPE order by CARD_TYPE;
```

CARD_TYPE	CARD_COUNT
GOLD	3
INSTANT	2
PLATINUM	3

23. Please follow instructions given below.

Write a Query to display serial no, first name, mobile number, email id of all the passengers who holds email address from gmail.com.

The Serial No will be the last three digits of profile ID.

Hint: Use SERIAL_NO as Alias name for serial number.

Display the records sorted in ascending order based on name.

6 rows

```
select substring(profile_id,4) as SERIAL_NO, first_name, mobile_number, email_id
from air_passenger_profile where email_id like '%gmail.com' order by first_name;
```

SERIAL_NO	FIRST_NAME	MOBILE_NUMBER	EMAIL_ID
004	AARTHI	9595652530	AARTHI@GMAIL.COM

008	GANESH	9375237890	GANESH@GMAIL.COM
007	GAYATHRI	8073245678	GAYATHRI@GMAIL.COM
001	LATHA	9876543210	LATHA@GMAIL.COM
006	RAMESH	9432198760	RAMESH@GMAIL.COM
005	SIVA	9884416986	SIVA@GMAIL.COM

24. Please follow instructions given below.

Write a query to display the flight(s) which has least number of services in the month of May. The Query should fetch flight_id, from_location, to_location, least number of Services as "No_of_Services" Hint: Number of services offered can be calculated from the number of scheduled departure dates of a flight

If there are multiple flights, display them sorted in ascending order based on flight id.

4 rows

```
select af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date) as No_of_Services from air_flight af join air_flight_details afd on af.flight_id=afd.flight_id where month(afd.flight_departure_date)='05' group by af.flight_id having count(afd.flight_departure_date)
<= all (select count(afd.flight_departure_date) from air_flight af join air_flight_details afd on af.flight_id=afd.flight_id where month(afd.flight_departure_date)='05' group by af.flight_id)
order by af.flight_id;
```

LIGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_SERVICES
1011	HYDERABAD	CHENNAI	2
1262	HYDERABAD	CHENNAI	2
3148	CHENNAI	BENGALURU	2
3244	KOCHI	CHENNAI	2

25. Please follow instructions given below.

Write a query to display the number of flights flying from each location. The Query should display the from location and the number of flights to other locations as "No_of_Flights".

Hint: Get the distinct from location and to location.

Display the records sorted in ascending order based on from location.

4 rows

```

select distinct from_location, count(to_location) as No_of_Flights from
air_flight
group by from_location order by from_location;

```

FROM_LOCATION	NO_OF_FLIGHTS
BENGALURU	2
CHENNAI	6
HYDERABAD	2
KOCHI	1

26. Please follow instructions given below.

Write a query to display the number of passengers traveled in each flight in each scheduled date. The Query should display flight_id,from_location,To_location, flight_departure_date and the number of passengers as "No_of_Passengers".

Hint: The Number of passengers inclusive of all the tickets booked with single profile id.
Display the records sorted in ascending order based on flight id and then by flight departure date.

9 rows

```

select af.flight_id,af.from_location,af.to_location,ati.flight_departure_date,count(ati.ticket_id)
as No_of_Passengers from air_flight af join air_ticket_info ati on af.flight_id=ati.flight_id
group by af.flight_id,ati.flight_departure_date order by af.flight_id,ati.flight_departure_date;

```

flight_id	from_location	to_location	flight_departure_date	No_of_Passengers
1011	HYDERABAD	CHENNAI	2013-05-09	3
1262	HYDERABAD	CHENNAI	2013-05-20	1
1265	CHENNAI	HYDERABAD	2013-04-29	1
1265	CHENNAI	HYDERABAD	2013-05-29	1
3004	BENGALURU	CHENNAI	2013-05-02	3
3148	CHENNAI	BENGALURU	2013-05-21	1
3148	CHENNAI	BENGALURU	2013-06-01	3
3244	KOCHI	CHENNAI	2013-05-03	7
916	CHENNAI	HYDERABAD	2013-05-06	2

27. Please follow instructions given below.

Write a query to display the flight details in which more than 10% of seats have been booked.
The query should display Flight_Id, From_Location, To_Location, Total_Seats, seats booked as "No_of_Seats_Booked".

Display the records sorted in ascending order based on flight id and then by No_of_Seats_Booked.

1 row

```
select af.flight_id,af.from_location,af.to_location,af.total_seats,(af.total_seats-  
afd.available_seats)  
as No_of_Seats_Booked from air_flight af join air_flight_details afd on af.flight_id=  
afd.flight_id where (af.total_seats-afd.available_seats)>(af.total_seats*0.1) group by flight_id  
order by  
flight_id, No_of_Seats_Booked;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	TOTAL_SEATS	NO_OF_SEATS_BOOKED
3244	KOCHI	CHENNAI	50	7

28. Please follow instructions given below.

Write a query to display the Flight_Id, Flight_Departure_Date, From_Location, To_Location and Duration of all flights which has duration of travel less than 1 Hour, 10 Minutes.

Display the records sorted in ascending order based on flight id and then by flight departure date.

14 rows

```
select af.flight_id, afd.flight_departure_date, af.from_location, af.to_location, af.duration  
from air_flight af join air_flight_details afd on af.flight_id=afd.flight_id  
where duration<'1:10:00' group by af.flight_id, afd.flight_departure_date  
order by af.flight_id, afd.flight_departure_date;
```

FLIGHT_ID	FLIGHT_DEPARTURE_DATE	FROM_LOCATION	TO_LOCATION	DURATION
3013	2013-05-04	CHENNAI	BENGALURU	01:05:00
3013	2013-05-06	CHENNAI	BENGALURU	01:05:00
3013	2013-05-22	CHENNAI	BENGALURU	01:05:00
3013	2013-05-30	CHENNAI	BENGALURU	01:05:00
3148	2013-05-16	CHENNAI	BENGALURU	01:05:00

3148	2013-05-21	CHENNAI	BENGALURU	01:05:00
3148	2013-06-01	CHENNAI	BENGALURU	01:05:00
3307	2013-05-03	BENGALURU	CHENNAI	01:00:00
3307	2013-05-23	BENGALURU	CHENNAI	01:00:00
3307	2013-05-29	BENGALURU	CHENNAI	01:00:00
916	2013-04-28	CHENNAI	HYDERABAD	01:05:00
916	2013-05-01	CHENNAI	HYDERABAD	01:05:00
916	2013-05-06	CHENNAI	HYDERABAD	01:05:00
916	2013-05-12	CHENNAI	HYDERABAD	01:05:00

29. Please follow instructions given below.

Write a query to display the flight_id, from_location,to_location,number of services as "No_of_Services" , average ticket price as "Average_Price" whose average ticket price is greater than the total average ticket cost of all flights. Order the result by lowest average price.

4 rows

```
select af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date) as
```

No_of_Services,

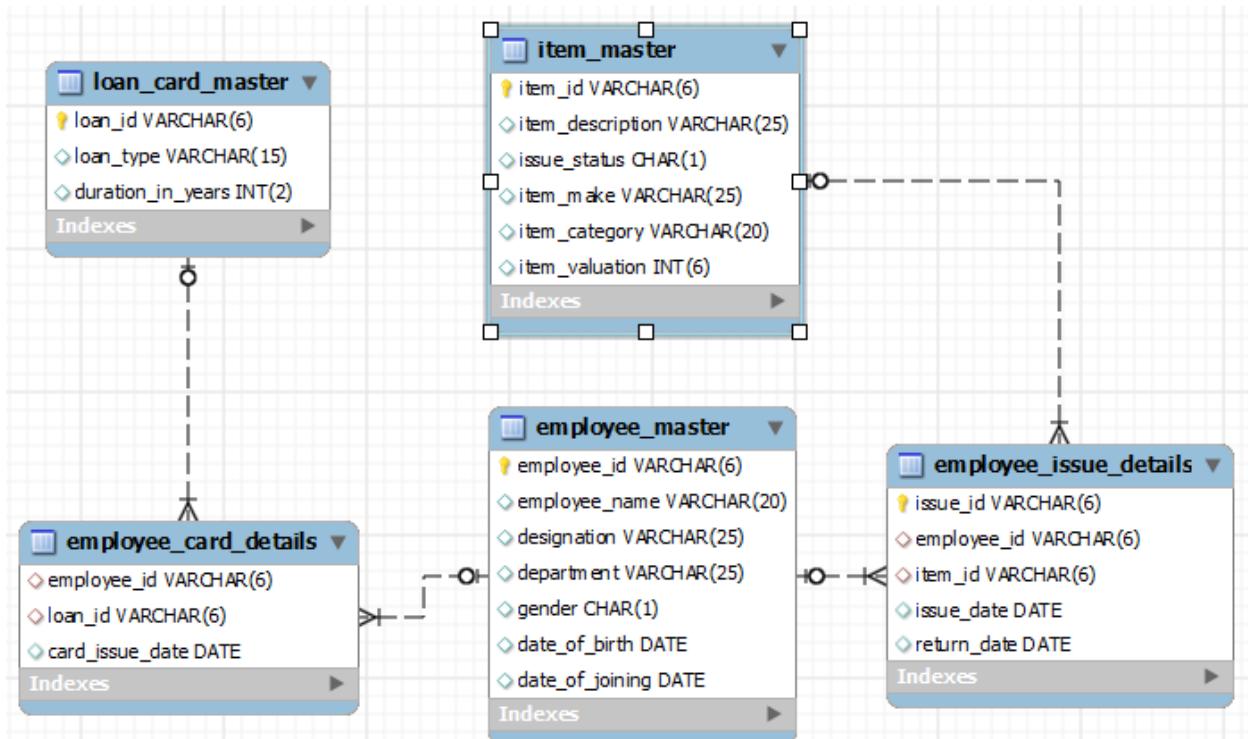
```
avg(afd.price) as Average_Price from air_flight af join air_flight_details afd
```

```
on af.flight_id=afd.flight_id group by af.flight_id having avg(afd.price)>
```

```
(select avg(afd.price) from air_flight_details afd) order by afd.price;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_SERVICES	AVERAGE
1262	HYDERABAD	CHENNAI	2	3444.5000
1265	CHENNAI	HYDERABAD	4	3499.2500
916	CHENNAI	HYDERABAD	4	3699.5000
1011	HYDERABAD	CHENNAI	3	4108.3333

ANSI SQL Item Loan Management System Schema



Item Loan Database Queries

1. Please follow instructions given below.

Write a query to display category and number of items in that category. Give the count an alias name of Count_category. Display the details on the sorted order of count in descending order.

3 rows

```
SELECT item_category , count(item_id) Count_category
FROM item_master
GROUP BY item_category order by count_category DESC;
```

Filter:		
	item_category	Count_category
▶	furniture	15
	Crockery	4
	Stationary	3

2. Please follow instructions given below.

Write a query to display the number of employees in HR department. Give the alias name as No_of_Employees.

1 row

```
SELECT count(employee_id) AS No_of_Employees
```

```
FROM employee_master  
WHERE department= 'HR'
```

Filter:	
No_of_Employees	
▶	2

3. Please follow instructions given below.

Write a query to display employee id, employee name, designation and department for employees who have never been issued an item as a loan from the company. Display the records sorted in ascending order based on employee id.

1 row

```
select employee_id,employee_name,designation,department from employee_master  
where employee_id  
not in (select employee_id from employee_issue_details) order by employee_id;
```

employee_id	employee_name	designation	department
E00005	Radica	Manager	HR
NULL	NULL	NULL	NULL

4. Please follow instructions given below.

Write a query to display the employee id, employee name who was issued an item of highest valuation.

In case of multiple records, display the records sorted in ascending order based on employee id.

[Hint Suppose an item called dinning table is of 22000 and that is the highest price of the item that has been issued. So display the employee id and employee name who issued dinning table whose price is 22000.]

1 row

```
select em.employee_id,em.employee_name from employee_master em join  
employee_issue_details eid  
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id  
and im.item_valuation>=all(select im.item_valuation from employee_master em  
join employee_issue_details eid  
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id)  
order by employee_id;
```

	employee_id	employee_name
▶	E00004	Zuben

5. Please follow instructions given below.

Write a query to display issue_id, employee_id, employee_name.
Display the records sorted in ascending order based on issue id.

9 rows

```
select eid.issue_id,eid.employee_id,em.employee_name from employee_issue_details eid join employee_master em on eid.employee_id=em.employee_id group by eid.issue_id,eid.employee_id
```

	issue_id	employee_id	employee_name
▶	ISS001	E00001	Ram
	ISS002	E00001	Ram
	ISS003	E00002	Abhay
	ISS004	E00003	Anita
	ISS005	E00003	Anita
	ISS006	E00003	Anita
	ISS007	E00004	Zuben
	ISS008	E00006	John
	ISS009	E00004	Zuben

```
order by eid.issue_id;
```

6. Please follow instructions given below.

Write a query to display employee id, employee name who don't have loan cards.
Display the records sorted in ascending order based on employee id.

3 rows

```
SELECT employee_id, employee_name  
FROM employee_master  
WHERE employee_id NOT IN ( SELECT employee_id FROM employee_card_details )  
order by employee_id;
```

	employee_id	employee_name
▶	E00004	Zuben
	E00005	Radica
	E00006	John
*	NULL	NULL

7. Please follow instructions given below.

Write a query to count the number of cards issued to an employee "Ram". Give the count an alias name as No_of_Cards.

1 row

```
select count(eid.loan_id) as No_of_Cards from employee_card_details eid join  
employee_master em  
on eid.employee_id=em.employee_id where em.employee_name='Ram'
```

Filter:

	No_of_Cards
▶	3

8. Please follow instructions given below.

Write a query to display the count of customers who have gone for loan type stationary. Give the count an alias name as Count_stationary.

1 row

```
select count(ecd.employee_id) as Count_Stationary from employee_card_details ecd  
join loan_card_master lcm on ecd.loan_id=lcm.loan_id where lcm.loan_type='Stationary'
```

Filter:

	Count_stationary
▶	3

9. Please follow instructions given below.

Write a query to display the employee id, employee name and number of items issued to them. Give the number of items an alias name as Count. Display the details in descending order of count and then by employee id in ascending order. Consider only employees who have been issued atleast 1 item.

5 rows

```
select em.employee_id,em.employee_name,count(eid.item_id) as Count from  
employee_master em join  
employee_issue_details eid on em.employee_id=eid.employee_id group by em.employee_id  
having  
count(eid.item_id)>=1 order by Count desc,employee_id asc;
```

	employee_id	employee_name	Count
▶	E00003	Anita	3
	E00001	Ram	2
	E00004	Zuben	2
	E00002	Abhay	1
	E00006	John	1

10. Please follow instructions given below.

Write a query to display the employee id, employee name who was issued an item of minimum valuation.

In case of multiple records, display them sorted in ascending order based on employee id.

[Hint Suppose an item called pen is of rupees 20 and that is the lowest price. So display the employee id and employee name who issued pen where the valuation is 20.]

2 rows

```
select em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id
and im.item_valuation<=all (select im.item_valuation from employee_master em join
employee_issue_details eid
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id) order by
employee_id;
```

	employee_id	employee_name
▶	E00002	Abhay
	E00003	Anita

11. Please follow instructions given below.

Write a query to display the employee id, employee name and total valuation of the product issued to each employee. Give the alias name as TOTAL_VALUATION.

Display the records sorted in ascending order based on employee id.

Consider only employees who have been issued atleast 1 item.

5 rows

```
select em.employee_id,em.employee_name,sum(im.item_valuation) as TOTAL_VALUATION
from employee_master em
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im
on eid.item_id=im.item_id group by em.employee_id having count(im.item_valuation)>=1
order by em.employee_id;
```

	employee_id	employee_name	TOTAL_VALUATION
▶	E00001	Ram	7000.00
	E00002	Abhay	1500.00
	E00003	Anita	15500.00
	E00004	Zuben	25500.00
	E00006	John	4500.00

12. Please follow instructions given below.

Write a query to display distinct employee id, employee name who kept the item issued for more than a year. Hint: Use Date time function to calculate the difference between item issue and return date. Display the records only if it is more than 365 Days.

Display the records sorted in ascending order based on employee id.

5 rows

```
select distinct em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid
```

on em.employee_id=eid.employee_id where datediff(return_date,issue_date)>365 order by

	employee_id	employee_name
▶	E00001	Ram
	E00002	Abhay
	E00003	Anita
	E00004	Zuben
	E00006	John

employee_id;

13. Please follow instructions given below.

Write a query to display employee id, employee name and count of items of those who asked for more than 1 furniture. Give the alias name for count of items as COUNT_ITEMS.

Display the records sorted in ascending order on employee id.

2 rows

```
select em.employee_id,em.employee_name,count(im.item_id) as COUNT_ITEMS from
employee_master em
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im
on eid.item_id=im.item_id where item_category='furniture' group by employee_id having
count(COUNT_ITEMS)>1 order by employee_id;
```

	employee_id	employee_name	COUNT_ITEMS
▶	E00001	Ram	2
	E00003	Anita	3

14. Please follow instructions given below.

Write a query to display the number of men & women Employees. The query should display the gender and number of Employees as No_of_Employees. Display the records sorted in ascending order based on gender.

2 rows

```
select gender,count(employee_id) as No_of_Employees from employee_master group by
```

	gender	No_of_Employees
▶	F	2
	M	4

gender order by gender;

15. Please follow instructions given below.

Write a query to display employee id, employee name who joined the company after 2005.
Display the records sorted in ascending order based on employee id.

3 rows

```
select employee_id,employee_name from employee_master where year(date_of_joining)>2005
```

	employee_id	employee_name
	E00002	Abhay
	E00003	Anita
	E00006	John

```
order by employee_id;
```

16. Please follow instructions given below.

Write a query to get the number of items of the furniture category issued and not issued. The query should display issue status and the number of furniture as No_of_Furnitures.

Display the records sorted in ascending order based on issue_status.

2 rows

```
select issue_status,count(item_id) as No_of_Furnitures from item_master where  
item_category='furniture' group by issue_status order by  
issue_status;
```

	issue_status	No_of_Furnitures
▶	N	6
	Y	9

17. Please follow instructions given below.

Write a query to find the number of items in each category, make and description. The Query should display Item Category, Make, description and the number of items as No_of_Items.

Display the records in ascending order based on Item Category, then by item make and then by item description.

16 rows

```
select item_category,item_make,item_description,count(item_id) as No_of_Items from  
item_master im group by item_category,item_make,item_description order by  
item_category,item_make,item_description;
```

	item_category	item_make	item_description	No_of_Items
▶	Crockery	Bonechina	Dining Set	1
	Crockery	Bonechina	Tea Set	1
	Crockery	Glass	Dining Set	1
	Crockery	Glass	Tea Set	1
	furniture	Steel	Cupboard	2
	furniture	Steel	Side Table	1
	furniture	Steel	Single Bed	2
	furniture	Steel	Tea Table	2
	furniture	Wooden	Dining Chair	1
	furniture	Wooden	Dining Table	1
	furniture	Wooden	Double Bed	2
	furniture	Wooden	Side Table	1
	furniture	Wooden	Sofa	1
	furniture	Wooden	Tea Table	2
	Stationary	Plastic	Pen	2
	Stationary	Wooden	Pencil	1

18. Please follow instructions given below.

Write a query to display employee id, employee name, item id and item description of employees who were issued item(s) in the month of January 2013. Display the records sorted in order based on employee id and then by item id in ascending order.

1 row

```
select em.employee_id,em.employee_name,im.item_id,im.item_description from
employee_master em join
employee_issue_details eid on em.employee_id=eid.employee_id join item_master im on
eid.item_id=im.item_id where year(eid.issue_date)=2013 and month(eid.issue_date)=01 order
by
```

	employee_id	employee_name	item_id	item_description
▶	E00002	Abhay	I00005	Side Table

em.employee_id,im.item_id;

19. Please follow instructions given below.

Write a query to display the employee id, employee name and count of item category of the employees who have been issued items in at least 2 different categories.

Give the alias name for category count as COUNT_CATEGORY.

Display the records sorted in ascending order based on employee id.

1 row

```
select em.employee_id,em.employee_name,count(distinct im.item_category) as  
COUNT_CATEGORY from employee_master em  
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im  
on eid.item_id=im.item_id group by em.employee_id having COUNT_CATEGORY>=2  
order by em.employee_id;
```

Filter: <input type="text"/>				Export:	Autosize:	
	employee_id	employee_name	COUNT_CATEGORY			
▶	E00004	Zuben	2			

20. Please follow instructions given below.

Write a query to display the item id , item description which was never issued to any employee.
Display the records sorted in ascending order based on item id.

14 rows

```
select item_id,item_description from item_master where item_id not in (select item_id  
from employee_issue_details) order by item_id;
```

Filter: <input type="text"/>		
	item_id	item_description
▶	I00002	Dining Table
	I00003	Tea Table
	I00006	Tea Table
	I00009	Sofa
	I00011	Cupboard
	I00013	Double Bed
	I00014	Single Bed
	I00015	Single Bed
	I00016	Tea Set
	I00017	Tea Set
	I00019	Dining Set
	I00020	Pencil
	I00021	Pen
	I00022	Pen
▼	NUL	NUL

21. Please follow instructions given below.

Write a query to display the employee id, employee name and total valuation for the employees who has issued minimum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION.

[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000

and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed.]

1 row

```
select em.employee_id,em.employee_name,sum(im.item_valuation) as TOTAL_VALUATION  
from employee_master em  
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im on  
eid.item_id=im.item_id group by em.employee_id having sum(im.item_valuation) <= all  
(select sum(im.item_valuation) from employee_master em  
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im on  
eid.item_id=im.item_id group by em.employee_id) order by employee_id;
```

Filter: <input type="text"/>			Export:	Autosize:
	employee_id	employee_name	TOTAL_VALUATION	
▶	E00002	Abhay	1500.00	

22. Please follow instructions given below.

Write a query to display the employee id, employee name, card issue date and card valid date. Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD_VALID_DATE.

[Hint: Validity in years for the loan card is given in loan_card_master table. Validity date is calculated by adding number of years in the loan card issue date. If the duration of year is zero then display AS 'No Validity Date'.]

```
SELECT ecd.employee_id,employee_name,  
card_issue_date, if(lcd.duration_in_years=0, 'NO-VALIDITY DATE',  
date_add(ec.card_issue_date, interval duration_in_years year)) as CARD_VALIDITY_DATE  
FROM employee_master em INNER JOIN  
employee_card_details ecd  
ON em.employee_id=ecd.employee_id  
INNER JOIN loan_card_master lcd  
ON ecd.loan_id=lcd.loan_id  
order by employee_name, CARD_VALID_DATE;
```

Filter: Export: Autosize:

	employee_id	employee_name	card_issue_date	CARD_VALID_DATE
▶	E00002	Abhay	2007-02-01	2012-02-01
	E00002	Abhay	2007-03-11	No Validity Date
	E00003	Anita	2007-04-15	2008-04-15
	E00003	Anita	2007-04-15	2012-04-15
	E00003	Anita	2007-04-15	No Validity Date
	E00001	Ram	2002-12-14	2003-12-14
	E00001	Ram	2000-01-01	2005-01-01
	E00001	Ram	2000-01-01	No Validity Date

23. Please follow instructions given below.

Write a query to display the employee id, employee name who have not issued with any item in the year 2013. Hint: Exclude those employees who was never issued with any of the items in all the years. Display the records sorted in ascending order based on employee id.

3 rows

```
select distinct em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid on
em.employee_id=eid.employee_id where em.employee_id not in
(select employee_id from employee_issue_details where year(issue_date)=2013)
order by employee_id;
```

Filter:

	employee_id	employee_name
▶	E00001	Ram
	E00003	Anita
	E00006	John

24. Please follow instructions given below.

Write a query to display issue id, employee id, employee name, item id, item description and issue date. Display the data in descending order of date and then by issue id in ascending order.

9 rows

```
select
eid.issue_id,em.employee_id,em.employee_name,im.item_id,im.item_description,eid.issue_dat
e
from employee_issue_details eid join employee_master em on
eid.employee_id=em.employee_id
```

```
join item_master im on eid.item_id=im.item_id order by eid.issue_date desc,eid.issue_id;
```

issue_id	employee_id	employee_name	item_id	item_description	issue_date
ISS009	E00004	Zuben	I00018	Dining Set	2013-04-18
ISS007	E00004	Zuben	I00012	Double Bed	2013-04-14
ISS003	E00002	Abhay	I00005	Side Table	2013-01-03
ISS008	E00006	John	I00018	Dining Set	2012-08-18
ISS006	E00003	Anita	I00010	Cupboard	2012-03-14
ISS001	E00001	Ram	I00001	Tea Table	2012-02-03
ISS002	E00001	Ram	I00004	Side Table	2012-02-03
ISS004	E00003	Anita	I00007	Dining Chair	2010-07-04
ISS005	E00003	Anita	I00008	Tea Table	2010-07-04

25. Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION.

[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issued item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name and total valuation of E00019 should display.]

1 row

```
select em.employee_id,em.employee_name,sum(im.item_valuation) as TOTAL_VALUATION  
from employee_master em join employee_issue_details eid on  
em.employee_id=eid.employee_id  
join item_master im on eid.item_id=im.item_id group by em.employee_id having  
sum(im.item_valuation)  
>= all (select sum(im.item_valuation) from employee_master em join employee_issue_details  
eid on em.employee_id=eid.employee_id  
join item_master im on eid.item_id=im.item_id group by em.employee_id);;
```

employee_id	employee_name	TOTAL_VALUATION
E00004	Zuben	25500.00

Oracle PLSQL V10 g Assessment for Learners - ATKDW253

Course Overview: This assessment will ensure the knowledge level assurance in Oracle PL/SQL

About the Test:

Test Duration: 120 mins

Number of Questions: 60

Pass Percentage: 70%

Target Audience: Employees

Topics Coverage:

Sl. No.	Topics Covered
1.	Oracle-Bind Variables
2.	Oracle-Compiler Directives
3.	Oracle-Data Dictionary
4.	Oracle-Oracle Clauses
5.	Oracle-Oracle Collections
6.	Oracle-Oracle Cursors
7.	Oracle-Oracle Exception Handling
8.	Oracle-Oracle Function, Procedure, Packages
9.	Oracle-Oracle Functions and Expressions
10.	Oracle-Oracle Indices, Keys and Constraints
11.	Oracle-Oracle Joins
12.	Oracle-Oracle Operators
13.	Oracle-Oracle SQL Concepts and Statements
14.	Oracle-Oracle Subqueries
15.	Oracle-Oracle Synonyms and Sequences
16.	Oracle-Oracle Triggers

17. Oracle-Oracle Views

18. Oracle-SQL Loader, Dynamic SQL and Performance Tuning

HCM Competency to be achieved after completing the assessment:

00002756 – Oracle

Mandatory Prerequisites:

NA

Reference Materials:

PLS00686	Oracle PL/SQL V10 g
----------	---------------------

Answers are highlighted in green. These are answers marked by me. If wrong means update it and let everyone know.

1. Select trunc(TO_DATE('25-JUL-2013'),'YEAR') FROM DUAL;
 - 25/7/2013
 - 42175
 - 42181
 - SYNTAX ERROR

2. SELECT ROUND('30-OCT-2013','YEAR') FROM DUAL;
 - 42184
 - 42640
 - Syntax error
 - 42175

3. Pragma that will call itself independently and can separate main user query from it is ...
 - Pragma autonomous transaction
 - Pragma exception init
 - None of options
 - Pragma restrict references

4. Pragma that will call an user exception and oracle will throw the error and assign a number
 - Pragma exception init
 - Pragma autonomous transaction
 - None of options
 - Pragma restrict references

5. Display date from 30, March , 2001 to ddmm-yyyy format
 - Select to_char(sysdate,'ddmm/yyyy') from dual;
*(For these type of questions understand to_date and to_char syntax correctly
because to_date will convert from char to date
To_char converts from other datatype to character.
Study functions like to_date, to_char, round, trunc)*

6. Which are equivalent to If then else statements:
 - Case when
 - Convert
 - Extract
 - Decode

7. Select year 1998 from a sysdate and display in environment:
 - Select to_char(sysdate,'year') from dual;

8. Update statement question :
Update departments set department_id=(select department_id from departments);
This statement if department_id is returned multiple means what it will do?
 - Throw syntax error
 - No rows are updated

- It will update with department_id value
 - None
9. Update employee where ename,eid=(select ename from employee where ename='Ruthvin') Where ename='Kavin'. This statement will do what?
- Both ename and eid are updated with same value
 - Ename is updated with this value and eid is retained with previous value
 - Ename, eid both not updated
 - Throw syntax error
10. Which is the dynamic view that lists the session information for each current session
- \$v\$session
 - \$v\$sessiononevent
 - \$v\$sessionwait
 - \$v\$sessiontrack
11. Examine the code:
- ```
CREATE OR REPLACE TRIGGER update_emp
AFTER UPDATE ON emp
BEGIN
 INSERT INTO audit_table (who, audited)
 VALUES(USER, SYSDATE);
END;
```
- You issue an update command on the EMP table that result in changing ten rows. How many rows are inserted into the AUDIT\_TABLE?
- 1
  - 10
  - none
  - Value equal to the number of rows in the EMP table
12. . Examine this function: (check box question)
- ```
CREATE OR REPLACE FUNCTION CALC_PLAYER_AVG
(V_ID IN PLAYER_BAT_STAT.PLAYER_ID%TYPE)
  RETURN NUMBER IS V_AVG NUMBER;
BEGIN
  SELECT HITS / AT_BATS INTO V_AVG
  FROM PLAYER_BAT_STAT
  WHERE PLAYER_ID = V_ID;
  RETURN (V_AVG);
END;
```
- Which statement will successfully invoke this function in SQL *Plus?
- SELECT CALC_PLAYER_AVG(PLAYER_ID) FROM PLAYER_BAT_STAT;
 - EXECUTE CALC_PLAYER_AVG(31);
 - CALC_PLAYER('RUTH');
 - CALC_PLAYER_AVG(31);
 - START CALC_PLAYER_AVG(31)
13. Database trigger is triggered when a table is truncated.

- True
- False

14. How do you disable a constraint :

Alter table tablename disable constraint constraint_name; is the syntax

15. What an inner join is called in other words:

- Equi join

16. Outer join main usage is:

- To display unmatched records from both the tables

17. Which operation is used for joining two tables which have same number of columns:

- Union

18. Select 1 from dual

Union

Select 1 from dual

Union

Select 3 from dual;

Ans:

- 1
- 3 (because union, intersection and minus operations will display only distinct records)

- 1
- 3
- None
- 0

19. Select 1 from dual

Minus

Select 3 from dual;

Ans:

- 1
- 3 (generally if we give a-b means records of b that are not present in a
Will be returned)
- 0
- None

20. If two tables are joined using Cartesian join and A has 5 rows and B 10 rows then output has how many rows?

50

21. Under which two circumstances do you design database triggers? (Choose two)

- A. To duplicate the functionality of other triggers.
- B. To replicate built-in constraints in the Oracle server such as primary key and foreign key.
- C. To guarantee that when a specific operation is performed, related actions are performed.
- D. For centralized, global operations that should be fired for the triggering statement, regardless of which user or application issues the statement.

22. Some question like...

If you want to display 2000 as \$2000.000 how you will display?
This is most confusing with options, someone find ans for this...

- Select to_char(2000,'\$#####') from dual;
- Select to_char(2000,'\$0000.000') from dual;
- Select to_char(2000,'\$9999.999') from dual;
- Select to_char(2000,'\$NNNN.NN') from dual;

23. If we trim S from SSNNSSS, then output:

- NN
- NNSSS
- BLANK OUTPUT
- NONE

24. IF AN INTERSECT STATEMENT IS issued between two tables one table customer with 4 rows and another table customer_2 with 10 rows. But none of the customer_id are matching between both tables. Then how many rows are returned.

- 0
- 40
- 10
- 4

Question 27

Answer saved

Marked out of 1

Flag question

In which of the following scenarios would a user recommend using associative arrays?

Select one:

- a. When the user wants to create a separate lookup table with multiple entries for each row of the main table, and access it through join queries
- b. When the user knows the number of elements in advance and the elements are usually accessed sequentially
- c. When the user wants to create a relatively small lookup table, which allows him/her to look up elements based on arbitrary numbers and strings
- d. When the user wants to retrieve an entire row from a table and perform calculations

Question 28

Answer saved

Marked out of 1

Flag question

```
DECLARE
p_emp_no NUMBER :=1 ;
BEGIN
FOR i IN 1..10 LOOP
update employee
set ename = 'John'
WHERE emp_no = p_emp_no;
END LOOP;
END;
```

25.

In this second question, how many execution plans generated for this statement.

- 1
- 10
- 5
- 1

26. Which of the following are used to release all database locks?

- Commit
- Rollback
- Save

- Savepoint

27 and 28.

Question 35

Answer saved

Marked out of 1

Flag question

Marilyn wants to eliminate the need to type the full table name when querying the "TRANSACTION_HISTORY" table existing in her schema. All other database users should use the schema and full table name when referencing this table. Which statement should Marilyn execute?

Select one:

- a. CREATE PRIVATE SYNONYM trans_hist FOR mark.transaction_history;
- b. CREATE PUBLIC SYNONYM trans_hist FOR mark;
- c. CREATE SYNONYM trans_hist FOR transaction_history;
- d. CREATE PUBLIC trans_hist SYNONYM FOR mark.transaction_history;

Question 36

Answer saved

Marked out of 1

Flag question

The user needs to perform certain data manipulation operations through a view called EMP_DEPT_VU, which he/she previously created. The user wants to look at the definition of the view (the SELECT statement on which the view was created). How does he/she obtain the definition of the view using a query?

Select one:

- a. Query the USER_VIEWS data dictionary to search for the EMP_DEPT_VU view
- b. Query the USER_OBJECTS data dictionary view to search for the EMP_DEPT_VU view
- c. Query the USER_SOURCE data dictionary view to search for the EMP_DEPT_VU view
- d. Use the DEFINE VIEW command on the EMP_DEPT_VU view

29. wildcard question came...

Question 53

Answer saved

Marked out of 1

Flag question

When is the wildcard in a WHERE clause useful?

Select one:

- a. An exact match is not possible in a CREATE statement.
- b. An exact match is necessary in a CREATE statement.
- c. An exact match is not possible in a SELECT statement.
- d. An exact match is necessary in a SELECT statement.

Question 54

Answer saved

Marked out of 1

Flag question

What is 'Shadow' file?

Select one:

- a. Shell initialization information required by all users
- b. Contains the group information for each account
- c. Holds the encrypted password of the corresponding account
- d. None of the listed options
- e. Contains secure group account information

30.

Question 55

Answer saved

Marked out of 1

Flag question

The user can use combination of GROUP BY clause, HAVING clause and WHERE clause in one SQL statement.

State True or False.

Select one:

- True
 False

Question 56

Answer saved

Marked out of 1

Flag question

Which of the following are true about aggregate functions?

Select one or more:

- a. You can pass column names, expressions, constants, or functions as parameters to an aggregate function.
 b. You can use aggregate functions only in the column list of the SELECT clause and in the WHERE clause of a SELECT statement.
 c. You can use aggregate functions in any clause of a SELECT statement.
 d. You can mix single row columns with aggregate functions in the column list of a SELECT statement by grouping on the single row columns.

Aggregate question came... a and d

31. a view stores the data physically in the database .

True or false

(**False**, only materialized view can store values not a simple view)

32. what is oracle views:

View is a virtual table that can be accessed via SQL commands

33. datatype question came...

Question 65

Answer saved

Marked out of 1

Flag question

Which of the below datatypes are supported by Oracle?

- I. Scalar datatype
II. Composite datatype
III. Large Object (LOB) datatype

Select one:

- a. I & II
 b. I, II & III
 c. I
 d. I & III

Question 66

Answer saved

Marked out of 1

Flag question

You cannot add a subquery to a SELECT clause as a column expression in the SELECT list.

State True or False.

Select one:

- True
 False

34. what happens if commit used 'FORUPDATE' cursor is open?

- Subsequent fetch throws an exception**

- Loop runs normally
- Cursor gets closed

35. some questions related to how order of statements will execute

1. Select 2. from 3. where 4.group by 5.having 6.order by

36. how to sort rows in sql? Order by

37. check box questions related to primary key, unique key and null

38. one question related to sql loader utility

39. which oracle access method make query processing easier.

- Creating unique index
- Primary key access
- Access with rowid
- Full table scan

ORACLE PL/SQL 10G

TEST 2016

October 2, 2015

1. Which two among the following programming constructs can be grouped within a package?

Answers:

1. Cursor
2. Constant
3. Trigger
4. Sequence
5. View

2. Which two statements, among the following, describe the state of a package variable after executing the package in which it is declared?

Answers:

1. It persists across transactions within a session
2. It persists from session to session for the same user
3. It does not persist across transaction within a session
4. It persists from user to user when the package is invoked
5. It does not persist from session to session for the same user

3. Which of the following is not a legal declaration?

Answers:

1. declare v number(4);
2. declare x,y varchar2(10);
3. declare birthdate date not null;
4. declare Sex boolean:=1;

4. The oracle server implicitly opens a cursor to process:

Answers:

1. A Sql select statement
2. A PL/SQL Select statement
3. DML Statements
4. DDL Statements

5. Which two statements out of the following regarding packages are true?

Answers:

1. Both the specification and body are required components of a package
2. The package specification is optional, but the package body is required
3. The package specification is required, but the package body is optional
4. The specification and body of the package are stored separately in the database

6. Examine the following trigger:

```
CREATE OR REPLACE TRIGGER Emp_count
AFTER DELETE ON Employee
FOR EACH ROW
DECLARE
n INTEGER;
BEGIN
SELECT COUNT(*) INTO n FROM employee;
DBMS_OUTPUT.PUT_LINE( 'There are now' || n || 'employees');
END;
```

This trigger results in an error after this SQL statement is entered: **DELETE FROM Employee WHERE Empno = 7499;**

How should the error be corrected?

Answers:

1. Change the trigger type to a BEFORE DELETE
2. Take out the COUNT function because it is not allowed in a trigger
3. Remove the DBMS_OUTPUT statement because it is not allowed in a trigger
4. Change the trigger to a statement-level trigger by removing FOR EACH ROW

7. Which Section deals with handling of errors that arise during execution of the data manipulation statements, which makeup the PL/SQL Block?

Answers:

1. Declare
2. Exception
3. Begin
4. End

8. Which of the following statements is true?

Answers:

1. Stored functions can be called from the SELECT and WHERE clauses only
2. Stored functions do not permit calculations that involve database links in a distributed environment
3. Stored functions cannot manipulate new types of data, such as longitude and latitude
4. Stored functions can increase the efficiency of queries by performing functions in the query rather than in the application

9. Examine the following code:

```
CREATE OR REPLACE FUNCTION gen_email (first_name VARCHAR2, last_name
VARCHAR2,
id NUMBER)
RETURN VARCHAR2 IS
email_name VARCHAR2(19);
BEGIN
email_name := SUBSTR(first_name, 1, 1) ||
SUBSTR(last_name, 1, 7) || '@Oracle.com .';
UPDATE employees SET email = email_name
WHERE employee_id = id;
RETURN email_name;
```

END;

Which of the following statements removes the function?

Answers:

1. DROP gen_email;
2. REMOVE gen_email;
3. DELETE gen_email;
4. **DROP FUNCTION gen_email;**

10. A table has to be dropped from within a stored procedure. How can this be implemented?

Answers:

1. A table cannot be dropped from a stored procedure
2. Use the DROP command in the procedure to drop the table
3. **Use the DBMS_DDL packaged routines in the procedure to drop the table**
4. Use the DBMS_DROP packaged routines in the procedure to drop the table.

11. In PI/Sql, if the where clause evaluates to a set of data, which lock is used?

Answers:

1. Row Level Lock
2. **Page Level lock**
3. Column Level lock
4. Exclusive Lock

12. If user defined error condition exists, Which of the following statements made a call to that exception?

Answers:

1. Trap
2. **Raise**
3. call
4. call Exception

13. Which of the following are identified by the “INSTEAD OF” clause in a trigger?

Answers:

1. **The view associated with the trigger**
2. The table associated with the trigger
3. The event associated with the trigger
4. The package associated with the trigger
5. The statement level or for each row association to the trigger

14. What type of trigger is created on the EMP table that monitors every row that is changed, and places this information into the AUDIT_TABLE?

Answers:

1. **FOR EACH ROW trigger on the EMP table**
2. Statement-level trigger on the EMP table
3. FOR EACH ROW trigger on the AUDIT_TABLE table
4. Statement-level trigger on the AUDIT_TABLE table

```

15. CREATE OR REPLACE PACKAGE manage_emp IS
tax_rate CONSTANT NUMBER(5,2) := .28;
v_id NUMBER;
PROCEDURE insert_emp (p_deptno NUMBER, p_sal NUMBER);
PROCEDURE delete_emp;
PROCEDURE update_emp;
FUNCTION cal_tax (p_sal NUMBER) RETURN NUMBER;
END manage_emp;
/
CREATE OR REPLACE PACKAGE BODY manage_emp IS
PROCEDURE update_sal (p_raise_amt NUMBER) IS
BEGIN
UPDATE emp SET sal = (sal * p_raise_emt) + sal
WHERE empno = v_id;
END;
PROCEDURE insert_emp (p_deptno NUMBER, p_sal NUMBER) IS
BEGIN
INSERT INTO emp(empno, deptno, sal) VALUES
(v_id, p_depntno, p_sal);
END insert_emp;
PROCEDURE delete_emp IS
BEGIN
DELETE FROM emp WHERE empno = v_id;
END delete_emp;
PROCEDURE update_emp IS
v_sal NUMBER(10,2);
v_raise NUMBER(10, 2);
BEGIN
SELECT sal INTO v_sal FROM emp WHERE empno = v_id;
IF v_sal < 500 THEN v_raise := .05;
ELSIP v_sal < 1000 THEN v_raise := .07;
ELSE v_raise := .04;
END IF;
update_sal(v_raise);
END update_emp;
FUNCTION cal_tax (p_sal NUMBER)RETURN NUMBER IS
BEGIN
RETURN p_sal * tax_rate;
END cal_tax;
END manage_emp;
/

```

What is the name of the private procedure in this package?

Answers:

1. CAL_TAX
2. INSERT_EMP
3. UPDATE_SAL

4. DELETE_EMP
5. UPDATE_EMP

16. Which procedure is called after a row has been fetched to transfer the value, from the select list of the cursor into a local variable?

Answers:

1. Row_value
2. Column_value
3. Raise_application
4. Exception_init

17. An internal LOB is _____.

Answers:

1. A table
2. A column that is a primary key
3. Stored in the database
4. A file stored outside of the database, with an internal pointer to it from a database column.

18. What is the maximum number of handlers processed before the PL/SQL block is exited, when an exception occurs?

Answers:

1. Only one
2. All that apply
3. All referenced
4. None

19. The technique employed by the Oracle engine to protect table data, when several people are accessing it is called:

Answers:

1. Concurrency Control
2. Program Control
3. PL/SQL Control
4. Locking

20. Which table should be queried to determine when the procedure was last compiled?

Answers:

1. USER_PROCEDURES
2. USER_PROCS
3. USER_OBJECTS
4. USER_PLSQL_UNITS

21. Which cursor dynamically allows passing values to a cursor while opening another cursor?

Answers:

1. Implicit Cursor
2. User Defined Cursor

3. Parameterized Cursor
4. Explicit Cursor

22. When the procedure or function is invoked, the Oracle engine loads the compiled procedure or function in the memory area called:

Answers:

1. PGA
2. SGA
3. Redo Log Buffer
4. Data base buffer cache

23. Which precompiled word is called, which when encountered, immediately binds the numbered exception handler to a name?

Answers:

1. Pragma
2. Raise
3. Trap
4. Exception_init

24. What happens during the execute phase with dynamic SQL for INSERT, UPDATE, and DELETE operations?

Answers:

1. The validity of the SQL statement is established
2. An area of memory is established to process the SQL statement
3. The SQL statement is run and the number of rows processed is returned
4. The area of memory established to process the SQL statement is released

25. Examine the following package specification:

CREATE OR REPLACE PACKAGE combine_all

IS

```
v_string VARCHAR2(100);
PROCEDURE combine (p_num_val NUMBER);
PROCEDURE combine (p_date_val DATE);
PROCEDURE combine (p_char_val VARCHAR2, p_num_val NUMBER);
END combine_all;
/
```

Which overloaded COMBINE procedure declaration can be added to this package specification?

Answers:

1. PROCEDURE combine;
2. PROCEDURE combine (p_no NUMBER);
3. PROCEDURE combine (p_val_1 VARCHAR2, p_val_2 NUMBER);
4. PROCEDURE combine_all(p_num_val VARCHAR2, p_char_val NUMBER);

26. Which part of a database trigger determines the number of times the trigger body executes?

Answers:

1. Trigger type

2. Trigger body
3. Trigger event
4. Trigger timing

27. Which table should be queried to check the status of a function?

Answers:

1. USER_PROCS
2. **USER_OBJECTS**
3. USER_PROCEDURES
4. USER_PLSQL_UNITS

28. Which of the following statements is true regarding stored procedures?

Answers:

1. A stored procedure uses the DECLARE keyword in the procedure specification to declare formal parameters
2. A stored procedure is named PL/SQL block with at least one parameter declaration in the procedure specification
3. **A stored procedure must have at least one executable statement in the procedure body**
4. A stored procedure uses the DECLARE keyword in the procedure body to declare formal parameters

29. Examine the following code:

```
CREATE OR REPLACE TRIGGER secure_emp
BEFORE LOGON ON employees
BEGIN
IF (TO_CHAR(SYSDATE, 'DY') IN ('SAT', 'SUN')) OR
(TO_CHAR(SYSDATE, 'HH24:MI')
NOT BETWEEN '08:00' AND '18:00')
THEN RAISE_APPLICATION_ERROR (-20500, 'You may
insert into the EMPLOYEES table only during
business hours.');
END IF;
END;
/
```

What type of trigger is it?

Answers:

1. DML trigger
2. INSTEAD OF trigger
3. Application trigger
4. **This is an invalid trigger**

30. Which code is stored in the database when a procedure or function is created in SQL*PLUS?

Answers:

1. **Only P-CODE**
2. Only SOURCE code

3. Both SOURCE CODE and P-CODE
4. Neither SOURCE CODE or P-CODE

31. Evaluate the following PL/SQL block:

```
DECLARE
v_low NUMBER:=2;
v_upp NUMBER:=100;
v_count NUMBER:=1;
BEGIN
FOR i IN v_low..v_low LOOP
INSERT INTO test(results)
VALUES (v_count)
v_count:=v_count+1;
END LOOP;
END;
```

How many times will the executable statements inside the FOR LOOP execute?

Answers:

1. 0
2. **1**
3. 2
4. 98
5. 100

32. What can be done with the DBMS_LOB package?

Answers:

1. Use the DBMS_LOB.WRITE procedure to write data to a BFILE
2. Use the DBMS_LOB.BFILENAME function to locate an external BFILE
3. **Use the DBMS_LOB.FILEEXISTS function to find the location of a BFILE**
4. Use the DBMS_LOB.FILECLOSE procedure to close the file being accessed

33. Examine the following code:

```
CREATE OR REPLACE TRIGGER UPD_SALARY
FOR EACH ROW
BEGIN
UPDATE TEAM
SET SALARY=SALARY+:NEW.SALARY
WHERE ID=:NEW.TEAM_ID
END;
```

Which statement must be added to make this trigger executable after updating the SALARY column of the PLAYER table?

Answers:

1. **AFTER UPDATE ON PLAYER**
2. AFTER SALARY UPDATE OF PLAYER
3. AFTER UPDATE(SALARY) ON PLAYER
4. AFTER UPDATE OF SALARY ON PLAYER

34. How can migration be done from a LONG to a LOB data type for a column?

Answers:

1. Use the DBMS_MANAGE_LOB.MIGRATE procedure
2. Use the UTL_MANAGE_LOB.MIGRATE procedure
3. Use the DBMS_LOB.MIGRATE procedure
4. You cannot migrate from a LONG to a LOB data type for a column
5. Using ALTER TABLE statement

35. Examine the following code:

```
CREATE OR REPLACE PACKAGE comm_package IS  
g_comm NUMBER := 10;  
PROCEDURE reset_comm(p_comm IN NUMBER);  
END comm_package;
```

User MILLER executes the following code at 9:01am:

```
EXECUTE comm_package.g_comm := 15
```

User Smith executes the following code at 9:05am:

```
EXECUTE comm_package.g_comm := 20
```

Which of the following statement is true?

Answers:

1. g_comm has a value of 15 at 9:06am for Smith
2. g_comm has a value of 15 at 9:06am for Miller
3. g_comm has a value of 20 at 9:06am for both Miller and Smith
4. g_comm has a value of 15 at 9:03 am for both Miller and Smith

36. The CHECK_SAL procedure calls the UPD_SAL procedure. Both procedures are INVALID.Which command can be issued to recompile both procedures?

Answers:

1. COMPILE PROCEDURE CHECK_SAL;
2. COMPILE PROCEDURE UPD_SAL;
3. Using ALTER PROCEDURE CHECK_SAL compile
4. ALTER PROCEDURE UPD_SAL compile

37. Examine the following procedure:

```
PROCEDURE emp_salary  
(v_bonus BOOLEAN,  
V_raise BOOLEAN,  
V_issue_check in out BOOLEAN)  
is  
BEGIN  
v_issue_check:=v_bonus or v_raise;  
END;
```

If v_bonus=TRUE and v_raise=NULL,which value is assigned to v_issue_check?

Answers:

1. TRUE
2. FALSE
3. NULL
4. none

38. Which package construct must be declared and defined within the packages body?

Answers:

1. Exception
2. Boolean Variable
3. Public Procedure
4. **Private Procedure**

39. What happens when rows are found using a FETCH statement?

Answers:

1. The cursor opens
2. The cursor closes
3. **The current row values are loaded into variables**
4. Variables are created to hold the current row values

40. Evaluate the following PL/SQL block:

```
DECLARE  
result BOOLEAN;  
BEGIN  
DELETE FROM EMPLOYEE  
WHERE dept_id IN (10,40,50);  
result:=SQL%ISOPEN;  
COMMIT;  
END;
```

What will be the value of RESULT if three rows are deleted?

Answers:

1. 0
2. 3
3. TRUE
4. NULL
5. **FALSE**

41. Which two statements among the following, regarding oracle database 10g PL/SQL support for LOB migration, are true?

Answers:

1. Standard package functions accept LOBs as parameters
2. Standard package function do not accept LOBs as parameters
3. Implicit data conversion is supported for converting LOB to RAW
4. Implicit data conversion is not supported for converting varchar to LOB

42. Which command is used to disable all triggers on the EMPLOYEES table?

Answers:

1. Multiple triggers on a table in one command cannot be disabled
2. ALTER TRIGGERS ON TABLE employees DISABLE;
3. ALTER employees DISABLE ALL TRIGGERS;
4. **ALTER TABLE employees DISABLE ALL TRIGGERS;**

43. Which table and column can be queried to see all procedures and functions that have been marked invalid?

Answers:

1. USER_ERRORS table, STATUS column
2. **USER_OBJECTS** table, STATUS column
3. USER_ERRORS table, INVALID column
4. USER_OBJECTS table, INVALID column

44. SQL%ISOPEN always evaluates to false in case of a/an:

Answers:

1. Explicit Cursor
2. **Implicit Cursor**
3. Paramertized Cursor
4. Cursor with Arguments

45. Which datatype does the cursor attribute '%ISOPEN' return?

Answers:

1. **BOOLEAN**
2. INTEGER
3. NUMBER
4. VARCHAR2

46. Which of the following is a benefit of using procedures and functions?

Answers:

1. Procedures and Function increases the number of calls to the database
2. Procedures and Function are reparsed for multiple users by exploiting shared SQL area
3. **Procedures and Function avoid reparsing for multiple users by exploiting shared SQL areas**
4. Testing of procedures and functions requires the database to be restarted, to clear out shared SQL areas and future access

47. All packages can be recompiled by using an Oracle utility called:

Answers:

1. Dbms_Output
2. Dbms_Lob
3. **Dbms_utility**
4. Dbms_Error

48. Which type of variable should be used to assign the value TRUE, FALSE?

Answers:

1. Constant
2. **Scalar**
3. Reference
4. Composite

49. In which type of trigger can the OLD and NEW qualifiers can be used?

Answers:

1. Row level DML trigger
2. Row level system trigger
3. Statement level DML trigger
4. Row level application trigger

50. Examine the following code:

```
CREATE OR REPLACE TRIGGER update_emp
AFTER UPDATE ON emp
BEGIN
INSERT INTO audit_table (who, dated) VALUES (USER, SYSDATE);
END;
/
```

An UPDATE command is issued in the EMP table that results in changing 10 rows
How many rows are inserted into the AUDIT_TABLE ?

Answers:

1. 1
2. 10
3. None
4. A value equal to the number of rows in the EMP table

SELF TEST Position the Server Technologies

1. What components of the IT environment can Oracle Enterprise Manager Cloud Control manage? (Choose the best answer.)

- A. Oracle databases
- B. Oracle application servers
- C. Third-party products
- D. The server machines

E. All of the above (cloud control)

2. What languages can run within the database? (Choose all that apply.)

- A. SQL**
- B. C**
- C. PL/SQL
- D. Java**

E. Any other language linked to the OCI libraries

Understand Relational Structures

3. Data that is modeled into a form suitable for processing in a relational database may be described as being (Choose the best answer.)

- A. First normal form
- B. Third normal form**

- C. Abnormal form
- D. Paranormal form

4. An entity-relationship diagram shows data modeled into (Choose the best answer.)

- A. Two-dimensional tables**
- B. Multidimensional tables
- C. Hierarchical structures
- D. Object-oriented structures

5. SQL is a set-oriented language. Which of these features is a consequence of this? (Choose the best answer.)

- A. Individual rows must have a unique identifier.

- B. Sets of users can be managed in groups.

C. SQL statements can be placed within blocks of code in other languages, such as Java and PL/SQL.

D. One statement can affect multiple rows.

6. Which of these constructs is not part of the SQL language? (Choose all that apply.)

A. Iteration, based on WHILE..

B. Iteration, based on FOR..DO

C. Branching, based on IF..THEN..ELSE

D. Transaction control, based on COMMIT

E. Transaction control, based on ROLLBACK

Use the Client Tools

7. Which of these statements regarding SQL Developer are correct? (Choose two answers.)

A. SQL Developer cannot connect to databases earlier than release 10g.

B. SQL Developer can be installed outside an Oracle Home.

C. SQL Developer can store passwords.

D. SQL Developer relies on an LDAP directory for name resolution.

8. Which of the following are requirements for using SQL Developer? (Choose two correct answers.)

A. A Java Runtime Environment

- B. The OCI libraries
- C. A name resolution method such as LDAP or a TNSNAMES.ORA file
- D. The SQL*Plus libraries
- E. A graphical terminal**

Create the Demonstration Schemas

- 9. Where may the demonstration schemas be created? (Choose the best answer.)
 - A. The demonstration schemas must be created in a demonstration database.
 - B. The demonstration schemas cannot be created in a production database.
 - C. The demonstration schemas can be created in any database.**
 - D. The demonstration schemas can be created in any database if the demonstration user is created first.
- 10. How can you move a schema from one user to another? (Choose the best answer.)
 - A. Use the ALTER SCHEMA MOVE... command.**
 - B. You cannot move a schema from one user to another.**
 - C. A schema can only be moved if it is empty (or if all objects within it have been dropped).
 - D. Attach the new user to the schema, then detach the old user from the schema.

Self test 2

- 1. Which query creates a projection of the DEPARTMENT_NAME and LOCATION_ID columns from the DEPARTMENTS table? (Choose the best answer.)
 - A. SELECT DISTINCT DEPARTMENT_NAME, LOCATION_ID FROM DEPARTMENTS;
 - B. SELECT DEPARTMENT_NAME, LOCATION_ID FROM DEPARTMENTS;**
 - C. SELECT DEPT_NAME, LOC_ID FROM DEPT;
 - D. SELECT DEPARTMENT_NAME AS "LOCATION_ID" FROM DEPARTMENTS;
- 2. After describing the EMPLOYEES table, you discover that the SALARY column has a data type of NUMBER(8,2). Which SALARY value(s) will not be permitted in this column? (Choose all that apply.)
 - A. SALARY=12345678**
 - B. SALARY=123456.78
 - C. SALARY=12345.678
 - D. SALARY=123456
 - E. SALARY=12.34
- 3. After describing the JOB_HISTORY table, you discover that the START_DATE and END_DATE columns have a data type of DATE. Consider the expression END_DATE-START_DATE. (Choose two correct statements.)
 - A. A value of DATE data type is returned.
 - B. A value of type NUMBER is returned.**
 - C. A value of type VARCHAR2 is returned.
 - D. The expression is invalid since arithmetic cannot be performed on columns with DATE data types.
 - E. The expression represents the days between the END_DATE and START_DATE less one day.**
- 4. The DEPARTMENTS table contains a DEPARTMENT_NAME column with data type VARCHAR2(30). (Choose two true statements about this column.)
 - A. This column can store character data up to a maximum of 30 characters.**
 - B. This column must store character data that is at least 30 characters long.
 - C. The VARCHAR2 data type is replaced by the CHAR data type.

D. This column can store data in a column with data type VARCHAR2(50) provided that the contents are at most 30 characters long.

Execute a Basic SELECT Statement

5. Which statement reports on unique JOB_ID values from the EMPLOYEES table? (Choose all that apply.)

A. SELECT JOB_ID FROM EMPLOYEES;

B. SELECT UNIQUE JOB_ID FROM EMPLOYEES;

C. SELECT DISTINCT JOB_ID, EMPLOYEE_ID FROM EMPLOYEES;

D. SELECT DISTINCT JOB_ID FROM EMPLOYEES;

6. Choose the two illegal statements. The two correct statements produce identical results.

The two illegal statements will cause an error to be raised:

A. SELECT DEPARTMENT_ID||' represents the'|| DEPARTMENT_NAME||' Department' as "Department Info" FROM DEPARTMENTS;

B. SELECT DEPARTMENT_ID||' represents the || DEPARTMENT_NAME||' Department' as "Department Info" FROM DEPARTMENTS;

C. select department_id||' represents the'||department_name||' Department' "Department Info" from departments;

D. SELECT DEPARTMENT_ID represents the DEPARTMENT_NAME Department as "Department Info" FROM DEPARTMENTS;

7. Which expressions do not return NULL values? (Choose all that apply.)

A. select ((10 + 20) * 50) + null from dual;

B. select 'this is a'||null||'test with nulls' from dual;

C. select null/0 from dual;

D. select null||'test'||null as "Test" from dual;

8. Choose the correct syntax to return all columns and rows of data from the EMPLOYEES table.

A. select all from employees;

B. select employee_id, first_name, last_name, first_name, department_id from employees;

C. select % from employees;

D. select * from employees;

E. select *.* from employees;

9. The following character literal expression is selected from the DUAL table: SELECT 'Coda"s favorite fetch toy is his orange ring' FROM DUAL; (Choose the result that is returned.)

A. An error would be returned due to the presence of two adjacent quotes

B. Coda's favorite fetch toy is his orange ring

C. Coda"s favorite fetch toy is his orange ring

D. Coda"s favorite fetch toy is his orange ring'

10. There are four rows of data in the REGIONS table. Consider the following SQL statement: SELECT '6 * 6' "Area" FROM REGIONS; How many rows of results are returned and what value is returned by the Area column? (Choose the best answer.)

A. 1 row returned, Area column contains value 36

B. 4 rows returned, Area column contains value 36 for all 4 rows

C. 1 row returned, Area column contains value 6 * 6

D. 4 rows returned, Area column contains value 6 * 6 for all 4 rows

E. A syntax error is returned.

Self test 3

1. Which two clauses of the SELECT statement facilitate selection and projection?
 - A. SELECT, FROM
 - B. ORDER BY, WHERE
 - C. SELECT, WHERE
 - D. SELECT, ORDER BY
2. Choose the query that extracts the LAST_NAME, JOB_ID, and SALARY values from the EMPLOYEES table for records having JOB_ID values of either SA_REP or MK_MAN and having SALARY values in the range of \$1,000 to \$4,000. The SELECT and FROM clauses are `SELECT LAST_NAME, JOB_ID, SALARY FROM EMPLOYEES`:
 - A. WHERE JOB_ID IN ('SA_REP', 'MK_MAN') AND SALARY > 1000 AND SALARY < 4000;
 - B. WHERE JOB_ID IN ('SA_REP', 'MK_MAN') AND SALARY BETWEEN 1000 AND 4000;
 - C. WHERE JOB_ID LIKE 'SA_REP%' AND 'MK_MAN%' AND SALARY > 1000 AND SALARY < 4000;
 - D. WHERE JOB_ID = 'SA_REP' AND SALARY BETWEEN 1000 AND 4000 OR JOB_ID='MK_MAN';
3. Which of the following WHERE clauses contains an error? The SELECT and FROM clauses are `SELECT * FROM EMPLOYEES`:
 - A. WHERE HIRE_DATE IN ('02-JUN-2004');
 - B. WHERE SALARY IN ('1000', '4000', '2000');
 - C. WHERE JOB_ID IN (SA_REP, MK_MAN);
 - D. WHERE COMMISSION_PCT BETWEEN 0.1 AND 0.5;
4. Choose the WHERE clause that extracts the DEPARTMENT_NAME values containing the character literal "er" from the DEPARTMENTS table. The SELECT and FROM clauses are `SELECT DEPARTMENT_NAME FROM DEPARTMENTS`:
 - A. WHERE DEPARTMENT_NAME IN ('%e%r');
 - B. WHERE DEPARTMENT_NAME LIKE '%er%';
 - C. WHERE DEPARTMENT_NAME BETWEEN 'e' AND 'r';
 - D. WHERE DEPARTMENT_NAME CONTAINS 'e%r';
5. Which two of the following conditions are equivalent to each other?
 - A. WHERE COMMISSION_PCT IS NULL
 - B. WHERE COMMISSION_PCT = NULL
 - C. WHERE COMMISSION_PCT IN (NULL)
 - D. WHERE NOT(COMMISSION_PCT IS NOT NULL)
6. Which two of the following conditions are equivalent to each other?
 - A. WHERE SALARY <=5000 AND SALARY >=2000
 - B. WHERE SALARY IN (2000, 3000, 4000, 5000)
 - C. WHERE SALARY BETWEEN 2000 AND 5000
 - D. WHERE SALARY > 2000 AND SALARY < 5000
 - E. WHERE SALARY >=2000 AND <=5000
- Sort the Rows Retrieved by a Query
7. Choose one false statement about the ORDER BY clause.
 - A. When using the ORDER BY clause, it always appears as the last clause in a SELECT statement.
 - B. The ORDER BY clause may appear in a SELECT statement that does not contain a WHERE clause.
 - C. The ORDER BY clause specifies one or more terms by which the retrieved rows are sorted. These terms can only be column names.

- D. Positional sorting is accomplished by specifying the numeric position of a column as it appears in the SELECT list, in the ORDER BY clause.
8. The following query retrieves the LAST_NAME, SALARY, and COMMISSION_PCT values for employees whose LAST_NAME begins with the letter "R". Based on the following query, choose the ORDER BY clause that first sorts the results by the COMMISSION_PCT column, listing highest commission earners first, and then sorts the results in ascending order by the SALARY column. Any records with NULL COMMISSION_PCT must appear last: `SELECT LAST_NAME, SALARY, COMMISSION_PCT FROM EMPLOYEES WHERE LAST_NAME LIKE 'R%'`
- ORDER BY COMMISSION_PCT DESC, 2;
 - ORDER BY 3 DESC, 2 ASC NULLS LAST;
 - C. ORDER BY 3 DESC NULLS LAST, 2 ASC;**
 - ORDER BY COMMISSION_PCT DESC, SALARY ASC;
- Ampersand Substitution
9. The DEFINE command explicitly declares a session-persistent substitution variable with a specific value. How is this variable referenced in an SQL statement? Consider an expression that calculates tax on an employee's SALARY based on the current tax rate. For the following session-persistent substitution variable, which statement correctly references the TAX_RATE variable? `DEFINE TAX_RATE=0.14`
- `SELECT SALARY * :TAX_RATE TAX FROM EMPLOYEES;`
 - B. `SELECT SALARY * &TAX_RATE TAX FROM EMPLOYEES;`**
 - `SELECT SALARY * :&TAX TAX FROM EMPLOYEES;`
 - `SELECT SALARY * TAX_RATE TAX FROM EMPLOYEES;`
10. When using ampersand substitution variables in the following query, how many times will you be prompted to input a value for the variable called JOB the first time this query is executed? `SELECT FIRST_NAME, '&JOB' FROM EMPLOYEES WHERE JOB_ID LIKE '%'||&JOB|| '%' AND '&&JOB' BETWEEN 'A' AND 'Z';`
- 0
 - 1
 - 2
 - D. 3**

Self test 3

Describe Various Types of Functions Available in SQL

- Which statements regarding single-row functions are true? (Choose all that apply.)
- They may return more than one result.
 - B. They execute once for each record processed.**
 - C. They may have zero or more input parameters.**
 - They must have at least one mandatory parameter.
- Which of these are single-row character-case conversion functions? (Choose all that apply.)
- A. LOWER**
 - B. SMALLER
 - C. INITCASE
 - D. INITCAP**

Use Character, Number, and Date Functions in SELECT Statements

3. What value is returned after executing the following statement? (Choose the best answer.)
SELECT LENGTH('How_long_is_a_piece_of_string?') FROM DUAL;
- A. 29
B. 30
C. 24
D. None of the above
4. What value is returned after executing the following statement? (Choose the best answer.)
SELECT SUBSTR('How_long_is_a_piece_of_string?', 5,4) FROM DUAL;
- A. long**
B. _long
C. ring?
D. None of the above
5. What value is returned after executing the following statement? (Choose the best answer.)
SELECT INSTR('How_long_is_a_piece_of_string?','_',5,3) FROM DUAL;
- A. 4
B. 14
C. 12
D. None of the above
6. What value is returned after executing the following statement? (Choose the best answer.)
SELECT REPLACE('How_long_is_a_piece_of_string?','_','') FROM DUAL;
- A. How long is a piece of string?
B. How_long_is_a_piece_of_string?
C. Howlongisapieceofstring?
D. None of the above
7. What value is returned after executing the following statement? (Choose the best answer.)
SELECT MOD(14,3) FROM DUAL;
- A. 3
B. 42
C. 2
D. None of the above
8. Assuming SYSDATE=07-JUN-1996 12:05pm, what value is returned after executing the following statement? (Choose the best answer.)
SELECT ADD_MONTHS(SYSDATE, -1)
FROM DUAL;
- A. 07-MAY-1996 12:05pm
B. 06-JUN-1996 12:05pm
C. 07-JUL-1996 12:05pm
D. None of the above
9. What value is returned after executing the following statement? Take note that 01-JAN-2009 occurs on a Thursday. (Choose the best answer.)
SELECT NEXT_DAY('01-JAN-2009','wed') FROM DUAL;
- A. 07-JAN-2009**
B. 31-JAN-2009
C. Wednesday
D. None of the above
10. Assuming SYSDATE=30-DEC-2007, what value is returned after executing the following statement? (Choose the best answer.)
SELECT TRUNC(SYSDATE,'YEAR') FROM DUAL;
- A. 31-DEC-2007
B. 01-JAN-2008

C. 01-JAN-2007

D. None of the above

Self test 4:-

Describe Various Types of Conversion Functions Available in SQL

1. What type of conversion is performed by the following statement? (Choose the best answer.) `SELECT LENGTH(3.14285) FROM DUAL;`

A. Explicit conversion

B. Implicit conversion

C. 7

D. None of the above

2. Choose any incorrect statements regarding conversion functions. (Choose all that apply.)

A. `TO_CHAR` may convert date items to character items.

B. `TO_DATE` may convert character items to date items.

C. `TO_CHAR` may convert numbers to character items.

D. `TO_DATE` may convert date items to character items.

Use the `TO_CHAR`, `TO_NUMBER`, and `TO_DATE` Conversion Functions

3. What value is returned after executing the following statement? (Choose the best answer.) `SELECT TO_NUMBER(1234.49, 999999.9) FROM DUAL;`

A. 1234.49

B. 001234.5

C. 1234.5

D. None of the above

4. What value is returned after executing the following statement? (Choose the best answer.) `SELECT TO_CHAR(1234.49, '999999.9') FROM DUAL;`

A. 1234.49

B. 001234.5

C. 1234.5

D. None of the above

5. If `SYSDATE` returns 12-JUL-2009, what is returned by the following statement? (Choose the best answer.) `SELECT TO_CHAR(SYSDATE, 'fmMONTH, YEAR') FROM DUAL;`

A. JUL, 2009

B. JULY, TWO THOUSAND NINE

C. JUL-09

D. None of the above

6. If `SYSDATE` returns 12-JUL-2009, what is returned by the following statement? (Choose the best answer.) `SELECT TO_CHAR(SYSDATE, 'fmDDth MONTH') FROM DUAL;`

A. 12TH JULY

B. 12th July

C. TWELFTH JULY

D. None of the above

Apply Conditional Expressions in a SELECT Statement

7. If `SYSDATE` returns 12-JUL-2009, what is returned by the following statement? (Choose the best answer.) `SELECT TO_CHAR(TO_DATE(TO_CHAR(SYSDATE,'DD'),'DD'),'YEAR') FROM DUAL;`

A. 2009

B. TWO THOUSAND NINE

C. 12-JUL-2009

D. None of the above

8. What value is returned after executing the following statement? (Choose the best answer.)

SELECT NVL2(NULLIF('CODA','SID'),'SPANIEL','TERRIER') FROM DUAL;

A. SPANIEL

B. TERRIER

C. NULL

D. None of the above

9. What value is returned after executing the following statement? (Choose the best answer.)

SELECT NVL(SUBSTR('AM I NULL',10),'YES I AM') FROM DUAL;

A. NO

B. NULL

C. YES I AM

D. None of the above

10. If SYSDATE returns 12-JUL-2009, what is returned by the following statement? (Choose the best answer.) SELECT DECODE(TO_CHAR(SYSDATE,'MM'),'02','TAX DUE','PARTY')

FROM DUAL;

A. TAX DUE

B. PARTY

C. 02

D. None of the above

SELF test:-

Describe the Group Functions

1. What result is returned by the following statement? (Choose the best answer.) SELECT COUNT(*) FROM DUAL;

A. NULL

B. 0

C. 1

D. None of the above

2. Choose one correct statement regarding group functions.

A. Group functions may only be used when a GROUP BY clause is present.

B. Group functions can operate on multiple rows at a time.

C. Group functions only operate on a single row at a time.

D. Group functions can execute multiple times within a single group.

Identify the Available Group Functions

3. What value is returned after executing the following statement? (Choose the best answer.) SELECT SUM(SALARY) FROM EMPLOYEES; Assume there are ten employee records and each contains a SALARY value of 100, except for one, which has a null value in the SALARY field.

A. 900

B. 1000

C. NULL

D. None of the above

4. Which values are returned after executing the following statement? (Choose all that apply.) SELECT COUNT(*), COUNT(SALARY) FROM EMPLOYEES; Assume there are ten

employee records and each contains a SALARY value of 100, except for one, which has a null value in their SALARY field.

- A. 10 and 10
- B. 10 and NULL
- C. 10 and 9**

D. None of the above

5. What value is returned after executing the following statement? (Choose the best answer.)
SELECT AVG(NVL(SALARY,100)) FROM EMPLOYEES; Assume there are ten employee records and each contains a SALARY value of 100, except for one employee, who has a null value in the SALARY field.

- A. NULL
- B. 90
- C. 100**

D. None of the above

Group Data Using the GROUP BY Clause

6. What value is returned after executing the following statement? (Choose the best answer.)
SELECT SUM((AVG(LENGTH(NVL(SALARY,0))))) FROM EMPLOYEES GROUP BY SALARY; Assume there are ten employee records and each contains a SALARY value of 100, except for one, which has a null value in the SALARY field.

- A. An error is returned
- B. 3
- C. 4**

D. None of the above

7. How many records are returned by the following query? (Choose the best answer.)
SELECT SUM(SALARY), DEPARTMENT_ID FROM EMPLOYEES GROUP BY DEPARTMENT_ID; Assume there are 11 nonnull and 1 null unique DEPARTMENT_ID values. All records have a nonnull SALARY value.

- A. 12**
- B. 11
- C. NULL

D. None of the above

8. What values are returned after executing the following statement? (Choose the best answer.)
SELECT JOB_ID, MAX_SALARY FROM JOBS GROUP BY MAX_SALARY; Assume that the JOBS table has ten records with the same JOB_ID value of DBA and the same MAX_SALARY value of 100.

- A. One row of output with the values DBA, 100
- B. Ten rows of output with the values DBA, 100
- C. An error is returned**

D. None of the above

Include or Exclude Grouped Rows Using the HAVING Clause

9. How many rows of data are returned after executing the following statement? (Choose the best answer.)
SELECT DEPT_ID, SUM(NVL(SALARY,100)) FROM EMP GROUP BY DEPT_ID HAVING SUM(SALARY) > 400; Assume the EMP table has ten rows and each contains a SALARY value of 100, except for one, which has a null value in the SALARY field. The first and second five rows have DEPT_ID values of 10 and 20, respectively.

- A. Two rows
- B. One row**
- C. Zero rows

D. None of the above

10. How many rows of data are returned after executing the following statement? (Choose the best answer.)
SELECT DEPT_ID, SUM(SALARY) FROM EMP GROUP BY DEPT_ID
HAVING SUM(NVL(SALARY,100)) > 400; Assume the EMP table has ten rows and each contains a SALARY value of 100, except for one, which has a null value in the SALARY field. The first and second five rows have DEPT_ID values of 10 and 20, respectively.

A. Two rows

B. One row

C. Zero rows

D. None of the above

Self test:-

Write SELECT Statements to Access Data from More Than One Table Using Equijoins and Nonequijoins

1. The EMPLOYEES and DEPARTMENTS tables have two identically named columns: DEPARTMENT_ID and MANAGER_ID. Which of these statements joins these tables based only on common DEPARTMENT_ID values? (Choose all that apply.)

A. SELECT * FROM EMPLOYEES NATURAL JOIN DEPARTMENTS;

B. SELECT * FROM EMPLOYEES E NATURAL JOIN DEPARTMENTS D ON E.DEPARTMENT_ID=D.DEPARTMENT_ID;

C. SELECT * FROM EMPLOYEES NATURAL JOIN DEPARTMENTS USING (DEPARTMENT_ID);

D. None of the above

2. The EMPLOYEES and DEPARTMENTS tables have two identically named columns: DEPARTMENT_ID and MANAGER_ID. Which statements join these tables based on both column values? (Choose all that apply.)

A. SELECT * FROM EMPLOYEES NATURAL JOIN DEPARTMENTS;

B. SELECT * FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID,MANAGER_ID);

C. SELECT * FROM EMPLOYEES E JOIN DEPARTMENTS D ON E.DEPARTMENT_ID=D.DEPARTMENT_ID AND E.MANAGER_ID=D.MANAGER_ID;

D. None of the above

3. Which join is performed by the following query? (Choose the best answer.)
SELECT E.JOB_ID,J.JOB_ID FROM EMPLOYEES E JOIN JOBS J ON (E.SALARY < J.MAX_SALARY);

A. Equijoin

B. Nonequijoin

C. Cross join

D. Outer join

4. Which of the following statements are syntactically correct? (Choose all that apply.)

A. SELECT * FROM EMPLOYEES E JOIN DEPARTMENTS D USING (DEPARTMENT_ID);

B. SELECT * FROM EMPLOYEES JOIN DEPARTMENTS D USING (D.DEPARTMENT_ID);

C. SELECT D.DEPARTMENT_ID FROM EMPLOYEES JOIN DEPARTMENTS D USING (DEPARTMENT_ID);

D. None of the above

5. Which of the following statements are syntactically correct? (Choose all that apply.)

- A. SELECT E.EMPLOYEE_ID, J.JOB_ID PREVIOUS_JOB, E.JOB_ID CURRENT_JOB
FROM JOB_HISTORY J CROSS JOIN EMPLOYEES E ON
(J.START_DATE=E.HIRE_DATE);
**B. SELECT E.EMPLOYEE_ID, J.JOB_ID PREVIOUS_JOB, E.JOB_ID CURRENT_JOB
FROM JOB_HISTORY J JOIN EMPLOYEES E ON (J.START_DATE=E.HIRE_DATE);**
C. SELECT E.EMPLOYEE_ID, J.JOB_ID PREVIOUS_JOB, E.JOB_ID CURRENT_JOB
FROM JOB_HISTORY J OUTER JOIN EMPLOYEES E ON
(J.START_DATE=E.HIRE_DATE);
D. None of the above
6. Choose one correct statement regarding the following query: `SELECT * FROM EMPLOYEES E JOIN DEPARTMENTS D ON (D.DEPARTMENT_ID=E.DEPARTMENT_ID) JOIN LOCATIONS L ON (L.LOCATION_ID =D.LOCATION_ID);`
- A. Joining three tables is not permitted.
B. A Cartesian product is generated.
C. The JOIN...ON clause may be used for joins between multiple tables.
D. None of the above
- Join a Table to Itself Using a Self-Join
7. How many rows are returned after executing the following statement? (Choose the best answer.) `SELECT * FROM REGIONS R1 JOIN REGIONS R2 ON (R1.REGION_ID=LENGTH(R2.REGION_NAME)/2);` The REGIONS table contains the following row data:
- | REGION_ID | REGION_NAME |
|-----------|------------------------|
| 1 | Europe |
| 2 | Americas |
| 3 | Asia |
| 4 | Middle East and Africa |
- A. 2
B. 3
C. 4
D. None of the above
- View Data that Does Not Meet a Join Condition Using Outer Joins
8. Choose one correct statement regarding the following query. `SELECT C.COUNTRY_ID
FROM LOCATIONS L RIGHT OUTER JOIN COUNTRIES C ON
(L.COUNTRY_ID=C.COUNTRY_ID) WHERE L.COUNTRY_ID is NULL;`
- A. No rows in the LOCATIONS table have the COUNTRY_ID values returned.**
B. No rows in the COUNTRIES table have the COUNTRY_ID values returned.
C. The rows returned represent the COUNTRY_ID values for all the rows in the LOCATIONS table.
D. None of the above
9. Which of the following statements are syntactically correct? (Choose all that apply.)
- A. SELECT JH.JOB_ID FROM JOB_HISTORY JH RIGHT OUTER JOIN JOBS J ON
JH.JOB_ID=J.JOB_ID;**
B. SELECT JOB_ID FROM JOB_HISTORY JH RIGHT OUTER JOIN JOBS J ON
(JH.JOB_ID=J.JOB_ID);
C. SELECT JOB_HISTORY.JOB_ID FROM JOB_HISTORY OUTER JOIN JOBS ON
JOB_HISTORY.JOB_ID=JOBS.JOB_ID;
D. None of the above
- Generate a Cartesian Product of Two or More Tables
10. If the REGIONS table, which contains 4 rows, is cross joined to the COUNTRIES table, which contains 25 rows, how many rows appear in the final results set? (Choose the best answer.)
- A. 100 rows**

- B. 4 rows
- C. 25 rows
- D. None of the above

Self test :-

Define Subqueries

1. Consider this generic description of a SELECT statement: SELECT select_list FROM table WHERE condition GROUP BY expression_1 HAVING expression_2 ORDER BY expression_3 ; Where could subqueries be used? (Choose all correct answers.)

- A. select_list
- B. table
- C. condition
- D. expression_1
- E. expression_2
- F. expression_3

2. A query can have a subquery embedded within it. Under what circumstances could there be more than one subquery? (Choose the best answer.)

- A. The outer query can include an inner query. It is not possible to have another query within the inner query.
- B. It is possible to embed a single-row subquery inside a multiple-row subquery, but not the other way around.
- C. The outer query can have multiple inner queries, but they must not be embedded within each other.

D. Subqueries can be embedded within each other with no practical limitations on depth.

3. Consider this statement: SELECT employee_id, last_name FROM employees WHERE salary > (SELECT avg(salary) FROM employees); When will the subquery be executed? (Choose the best answer.)

- A. It will be executed before the outer query.
- B. It will be executed after the outer query.
- C. It will be executed concurrently with the outer query.
- D. It will be executed once for every row in the EMPLOYEES table.

4. Consider this statement: SELECT o.employee_id, o.last_name FROM employees o WHERE o.salary > (SELECT avg(i.salary) FROM employees i WHERE i.department_id=o.department_id); When will the subquery be executed? (Choose the best answer.)

- A. It will be executed before the outer query.
- B. It will be executed after the outer query.
- C. It will be executed concurrently with the outer query.
- D. It will be executed once for every row in the EMPLOYEES table.

Describe the Types of Problems That the Subqueries Can Solve

5. Consider the following statement: SELECT last_name FROM employees JOIN departments ON employees.department_id = departments.department_id WHERE department_name='Executive'; and this statement: SELECT last_name FROM employees WHERE department_id IN (SELECT department_id FROM departments WHERE department_name='Executive'); What can be said about the two statements? (Choose two correct answers.)

- A. The two statements should generate the same result.
- B. The two statements could generate different results.
- C. The first statement will always run successfully; the second statement will error if there are two departments with DEPARTMENT_NAME='Executive'.
- D. Both statements will always run successfully, even if there are two departments with DEPARTMENT_NAME='Executive'.

List the Types of Subqueries

6. What are the distinguishing characteristics of a scalar subquery? (Choose two correct answers.)

- A. A scalar subquery returns one row.
 - B. A scalar subquery returns one column.
 - C. A scalar subquery cannot be used in the SELECT LIST of the parent query.
 - D. A scalar subquery cannot be used as a correlated subquery.
7. Which comparison operator cannot be used with multiple-row subqueries? (Choose the best answer.)
- A. ALL
 - B. ANY
 - C. IN
 - D. NOT IN
 - E. All the above can be used.

Write Single-Row and Multiple-Row Subqueries

8. Consider this statement: `SELECT last_name, (SELECT count(*)
FROM departments) FROM employees WHERE salary = (SELECT salary
FROM employees);` What is wrong with it? (Choose the best answer.)
- A. Nothing is wrong—the statement should run without error.
 - B. The statement will fail because the subquery in the SELECT list references a table that is not listed in the FROM clause.
 - C. The statement will fail if the second query returns more than one row.
 - D. The statement will run but is extremely inefficient because of the need to run the second subquery once for every row in EMPLOYEES.

9. Which of the following statements are equivalent? (Choose two answers.)
- A. `SELECT employee_id FROM employees WHERE salary < ALL (SELECT salary
FROM employees WHERE department_id=10);`
 - B. `SELECT employee_id FROM employees WHERE salary < (SELECT min(salary)
FROM employees WHERE department_id=10);`

- C. `SELECT employee_id FROM employees WHERE salary NOT >= ANY (SELECT
salary FROM employees WHERE department_id=10);`
- D. `SELECT employee_id FROM employees e JOIN departments d ON
e.department_id=d.department_id WHERE e.salary < (SELECT min(salary) FROM
employees) AND d.department_id=10;`

10. Consider this statement, which is intended to prompt for an employee's name and then find all employees who have the same job as the first employee: `SELECT
last_name,employee_id FROM employees WHERE job_id = (SELECT job_id
FROM employees WHERE last_name = '&Name');` What would happen if a value were given for &Name that did not match with any row in EMPLOYEES? (Choose the best answer.)

- A. The statement would fail with an error.
- B. The statement would return every row in the table.
- C. The statement would return no rows.

- D. The statement would return all rows where JOB_ID is NULL.

Self test:-

Describe the Set Operators

1. Which of these set operators will not sort the rows? (Choose the best answer.)

A. INTERSECT

B. MINUS

C. UNION

D. UNION ALL

2. Which of these operators will remove duplicate rows from the final result? (Choose all that apply.)

A. INTERSECT

B. MINUS

C. UNION

D. UNION ALL

Use a Set Operator to Combine Multiple Queries into a Single Query

3. If a compound query contains both a MINUS and an INTERSECT operator, which will be applied first? (Choose the best answer.)

A. The INTERSECT, because INTERSECT has higher precedence than MINUS.

B. The MINUS, because MINUS has a higher precedence than INTERSECT.

C. The precedence is determined by the order in which they are specified.

D. It is not possible for a compound query to include both MINUS and INTERSECT.

4. There are four rows in the REGIONS table. Consider the following statements and choose how many rows will be returned for each: 0, 4, 8, or 16.

A. SELECT * FROM regions UNION SELECT * FROM regions;

B. SELECT * FROM regions UNION ALL SELECT * FROM regions;

C. SELECT * FROM regions MINUS SELECT * FROM regions;

D. SELECT * FROM regions INTERSECT SELECT * FROM regions;

5. Consider this compound query: `SELECT empno, hired FROM emp UNION ALL
SELECT emp_id,hired,fired FROM ex_emp;` The columns EMP.EMPNO and EX_EMP.EMP_ID are integer; the column EMP.HIRED is timestamp; the columns EX_EMP.HIRED and EX_EMP.FIRED are date. Why will the statement fail? (Choose the best answer.)

A. Because the columns EMPNO and EMP_ID have different names

B. Because the columns EMP.HIRED and EX_EMP.HIRED are different data types

C. Because there are two columns in the first query and three columns in the second query

D. For all the reasons above E. The query will succeed.

Control the Order of Rows Returned

6. Which line of this statement will cause it to fail? (Choose the best answer.)

A. `SELECT ename, hired FROM current_staff`

B. ORDER BY ename

C. MINUS

D. `SELECT ename, hired FROM current staff E. WHERE deptno=10 F. ORDER BY ename;`

7. Study this statement: `SELECT ename FROM emp UNION ALL SELECT ename FROM ex_emp;` In what order will the rows be returned? (Choose the best answer.)

- A. The rows from each table will be grouped and within each group will be sorted on ENAME.
- B. The rows from each table will be grouped but not sorted.**
- C. The rows will not be grouped but will all be sorted on ENAME.
- D. The rows will be neither grouped nor sorted.

Self test:-

Describe Each Data Manipulation Language (DML) Statement

- 1. Which of the following commands can be rolled back?
- A. COMMIT
- B. DELETE**
- C. INSERT**
- D. MERGE**
- E. TRUNCATE
- F. UPDATE**
- 2. How can you change the primary key value of a row? (Choose the best answer.)
- A. You cannot change the primary key value.
- B. Change it with a simple UPDATE statement.**
- C. The row must be removed with a DELETE and reentered with an INSERT.
- D. This is only possible if the row is first locked with a SELECT FOR UPDATE.
- 3. If an UPDATE or DELETE command has a WHERE clause that gives it a scope of several rows, what will happen if there is an error part way through execution? The command is one of several in a multistatement transaction. (Choose the best answer.)
- A. The command will skip the row that caused the error and continue.
- B. The command will stop at the error, and the rows that have been updated or deleted will remain updated or deleted.
- C. Whatever work the command had done before hitting the error will be rolled back, but work done already by the transaction will remain.**
- D. The whole transaction will be rolled back.

Insert Rows into a Table

- 4. If a table T1 has four numeric columns, C1, C2, C3, and C4, which of these statements will succeed? (Choose the best answer.)
- A. INSERT INTO T1 VALUES (1,2,3,null);
- B. INSERT INTO T1 values ('1','2','3','4');
- C. INSERT INTO T1 SELECT * FROM T1;
- D. All the statements (A, B, and C) will succeed.**
- E. None of the statements (A, B, or C) will succeed.

5. Study the result of this SELECT statement:

```
SELECT * FROM t1;    C1      C2      C3      C4 -----
1      2      3      4      5      6      7      8
```

If you issue this statement:

INSERT INTO t1 (c1,c2) VALUES (SELECT c1,c2 FROM t1); why will it fail? (Choose the best answer.)

- A. Because values are not provided for all the table's columns: there should be NULLs for C3 and C4.
- B. Because the subquery returns multiple rows: it requires a WHERE clause to restrict the number of rows returned to one.

- C. Because the subquery is not scalar: it should use MAX or MIN to generate scalar values.
- D. Because the VALUES keyword is not used with a subquery.
- E. It will succeed, inserting two rows with NULLs for C3 and C4.
6. Consider this statement: INSERT INTO regions (region_id,region_name) VALUES ((SELECT max(region_id)+1 FROM regions), 'Great Britain'); What will the result be? (Choose the best answer.)
- A. The statement will not succeed if the value generated for REGION_ID is not unique, because REGION_ID is the primary key of the REGIONS table.
- B. The statement has a syntax error because you cannot use the VALUES keyword with a subquery.
- C. The statement will execute without error.
- D. The statement will fail if the REGIONS table has a third column.
- Update Rows in a Table
7. You want to insert a row and then update it. What sequence of steps should you follow? (Choose the best answer.)
- A. INSERT, UPDATE, COMMIT
- B. INSERT, COMMIT, UPDATE, COMMIT
- C. INSERT, SELECT FOR UPDATE, UPDATE, COMMIT
- D. INSERT, COMMIT, SELECT FOR UPDATE, UPDATE, COMMIT
8. If you issue this command: UPDATE employees SET salary=salary * 1.1; what will be the result? (Choose the best answer.)
- A. The statement will fail because there is no WHERE clause to restrict the rows affected.
- B. The first row in the table will be updated.
- C. There will be an error if any row has its SALARY column NULL.
- D. Every row will have SALARY incremented by 10 percent, unless SALARY was NULL.
- UPDATE command, setting every column to NULL and with no WHERE clause
9. How can you delete the values from one column of every row in a table? (Choose the best answer.)
- A. Use the DELETE COLUMN command.
- B. Use the TRUNCATE COLUMN command.
- C. Use the UPDATE command.
- D. Use the DROP COLUMN command.
10. Which of these commands will remove every row in a table while keeping its structure intact? (Choose one or more correct answers.)
- A. A DELETE command with no WHERE clause
- B. A DROP TABLE command
- C. A TRUNCATE command
- D. An UPDATE command, setting every column to NULL and with no WHERE clause
- Control Transactions
11. User JOHN updates some rows and asks user ROOPESH to log in and check the changes before he commits them. Which of the following statements is true? (Choose the best answer.)
- A. ROOPESH can see the changes but cannot alter them because JOHN will have locked the rows.
- B. ROOPESH will not be able to see the changes.
- C. JOHN must commit the changes so that ROOPESH can see them and, if necessary, roll them back.

- D. JOHN must commit the changes so that ROOPESH can see them, but only JOHN can roll them back.
12. User JOHN updates some rows but does not commit the changes. User ROOPESH queries the rows that JOHN updated. Which of the following statements is true? (Choose the best answer.)
- A. ROOPESH will not be able to see the rows because they will be locked.
 - B. ROOPESH will be able to see the new values, but only if he logs in as JOHN.
 - C. ROOPESH will see the old versions of the rows.**
 - D. ROOPESH will see the state of the data as it was when JOHN last created a SAVEPOINT.
13. Which of these commands will terminate a transaction? (Choose three correct answers.)
- A. COMMIT**
 - B. DELETE
 - C. ROLLBACK**
 - D. ROLLBACK TO SAVEPOINT
 - E. SAVEPOINT
 - F. TRUNCATE**

Self test:-

Categorize the Main Database Objects

1. If a table is created without specifying a schema, in which schema will it be? (Choose the best answer.)
- A. It will be an orphaned table, without a schema.
 - B. The creation will fail.
 - C. It will be in the SYS schema.
 - D. It will be in the schema of the user creating it.**
 - E. It will be in the PUBLIC schema.
2. Several object types share the same namespace, and therefore cannot have the same name in the same schema. Which of the following object types is not in the same namespace as the others? (Choose the best answer.)
- A. Index**
 - B. PL/SQL stored procedure
 - C. Synonym
 - D. Table
 - E. View
3. Which of these statements will fail because the table name is not legal? (Choose two answers.)
- A. create table "SELECT" (col1 date);
 - B. create table "lowercase" (col1 date);
 - C. create table number1 (col1 date);
 - D. create table 1number (col1 date);**
 - E. create table update (col1 date);**

Review the Table Structure

4. What are distinguishing characteristics of heap tables? (Choose two answers.)
- A. A heap can store variable length rows.**
 - B. More than one table can store rows in a single heap.
 - C. Rows in a heap are in random order.**
 - D. Heap tables cannot be indexed.

E. Tables in a heap do not have a primary key.

List the Data Types That Are Available for Columns

5. Which of the following data types are variable length? (Choose all correct answers.)

- A. BLOB
- B. CHAR
- C. LONG
- D. NUMBER
- E. RAW
- F. VARCHAR2

6. Study these statements: CREATE TABLE tab1 (c1 NUMBER(1), c2 DATE); ALTER SESSION SET nls_date_format='dd-mm-yy'; INSERT INTO tab1 VALUES (1.1,'31-01-07');

Will the insert succeed? (Choose the best answer)

- A. The insert will fail because the 1.1 is too long.
- B. The insert will fail because the '31-01-07' is a string, not a date.
- C. The insert will fail for both reasons A and B.

D. The insert will succeed.

7. Which of the following is not supported by Oracle as an internal data type? (Choose the best answer.)

- A. CHAR
- B. FLOAT
- C. INTEGER
- D. STRING**

Create a Simple Table

8. Consider this statement: CREATE TABLE t1 AS SELECT * FROM regions WHERE 1=2; What will be the result? (Choose the best answer.)

- A. There will be an error because of the impossible condition.
- B. No table will be created because the condition returns FALSE.
- C. The table T1 will be created but no rows inserted because the condition returns FALSE.**
- D. The table T1 will be created and every row in REGIONS inserted because the condition returns a NULL as a row filter.

9. When a table is created with a statement such as the following: CREATE TABLE newtab AS SELECT * FROM tab; will there be any constraints on the new table? (Choose the best answer.)

- A. The new table will have no constraints, because constraints are not copied when creating tables with a subquery.
- B. All the constraints on TAB will be copied to NEWTAB.
- C. Primary key and unique constraints will be copied but not check and not null constraints.
- D. Check and not null constraints will be copied but not unique or primary key.**
- E. All constraints will be copied, except foreign key constraints.

Explain How Constraints Are Created at the Time of Table Creation

10. Which types of constraint require an index? (Choose all that apply.)

- A. CHECK
- B. NOT NULL
- C. PRIMARY KEY**
- D. UNIQUE**

11. A transaction consists of two statements. The first succeeds, but the second (which updates several rows) fails partway through because of a constraint violation. What will happen? (Choose the best answer.)

- A. The whole transaction will be rolled back.
- B. The second statement will be rolled back completely, and the first will be committed.
- C. The second statement will be rolled back completely, and the first will remain uncommitted.
- D. Only the one update that caused the violation will be rolled back; everything else will be committed.
- E. Only the one update that caused the violation will be rolled back; everything else will remain uncommitted.

Self test:-

PL/SQL IQA 2

1. The number of cascading triggers is limited by which data base initialization parameter?
 - a. CASCADE_TRIGGER_CNT
 - b. **OPEN_CURSORS**
 - c. OPEN_TRIGGERs
 - d. OPEN_DB_TRIGGERs
2. Which type of package construct must be declared both within package specifications and package body?
 - a. All package variables.
 - b. Boolean variables.
 - c. Private procedures and functions.
 - d. **Public procedures and functions.**
3. Why do stored procedures and functions improve performance? (choose two)
 - a. They reduce network round trips.
 - b. They postpone PL/SQL parsing until run time.
 - c. They allow the application to perform high speed processing locally.
 - d. They reduce the number of calls to the database and decrease network traffic by bundling commands.
 - e. They reduce the number of calls to the database and decrease network traffic by using the local PL/SQL engine.
4. When creating stored procedures and functions, which construct allows you to transfer values to and from the calling environment?
 - a. Local variables.
 - b. **Formal arguments.**
 - c. Boolean variables.
 - d. Substitution variables.
5. You need to remove the database trigger, BUSINESS_RULE. Which command do you use to remove the trigger in the SQL*Plus environment?
 - a. **DROP TRIGGER business_rule;**
 - b. DELETE TRIGGER business_rule;
 - c. REMOVE TRIGGER business_rule;
 - d. ALTER TRIGGER business_rule;
 - e. DELETE FROM USER_TRIGGER WHERE TRIGGER_NAME= 'BUSINESS_RULE';
6. Which two tables are fused track object dependencies?
 - a. **USER_DEPENDENCIES.**
 - b. USER_IDEPTREE.
 - c. IDEPTREE.
 - d. USER_DEPTREE.
 - e. USER_DEPENDS.
7. The QUERY_PRODUCT procedure directly references the product table. There is a NEW_PRODUCT_VIEW view created based on the NOT NULL columns of the table. The ADD_PRODUCT procedure updates the table indirectly by the way of NEW_PRODUCT_VIEW view. Under which circumstances does the procedure ADD_PRODUCT get invalidated but automatically get complied when invoked?
 - a. When the NEW_PRODUCT_VIEW is dropped.
 - b. When rows of the product table are updated through SQL Plus.
 - c. When the internal logic of the QUERY_PRODUCT procedure is modified.

- d. When a new column that can contain null values is added to the product table.
 - e. When a new procedure is created that updates rows in the product table directly.
8. You need to recompile several program units you have recently modified through a PL/SQL program. Which statement is true?
- a. You cannot recompile program units using a PL/SQL program.
 - b. You can use the DBMS_DDL.RECOMPILE package procedure to recompile the program units.
 - c. You can use the DBMS_ALTER_COMPILE packaged procedure to recompile the program units.
 - d. **You can use the DBMS_DDL.ALTER_COMPILE packaged procedure to recompile the program units.**
 - e. You can use the DBMS_SQL.ALTER_COMPILE packaged procedure to recompile the program units.
9. Which type of argument passes a value from a calling environment?
- a. VARCHAR2
 - b. BOOLEAN
 - c. OUT
 - d. IN
10. You need to create a trigger on the EMP table that monitors every row that is changed and places this information into the AUDIT_TABLE. Which type of trigger do you create?
- a. Statement-level trigger on the EMP table.
 - b. **For each row trigger on the EMP table.**
 - c. Statement-level trigger on the AUDIT_TABLE table.
 - d. For each row statement level trigger on the EMP table.
 - e. For each row trigger on the AUDIT_TABLE table.
11. In order for you to create a run package, MAINTAIN_DATA, which privilege do you need?
- a. **EXECUTE privilege on the MAINTAIN_DATA package.**
 - b. INVOKER privilege on the MAINTAIN_DATA package.
 - c. EXECUTE privilege on the program units in the MAINTAIN_DATA package.
 - d. Object privilege on all of the objects that the MAINTAIN_DATA package is accessing.
 - e. Execute privilege on the program units inside the MAINTAIN_DATA package and execute privilege on the MAINTAIN_DATA package.
12. You have created a script file EMP_PROC.SQL that holds text to create a procedure, PROCESS_EMP. You have compiled the procedure for the SQL Plus environment by running the script file EMP_PROC.SQL. What happens if there are syntax errors in the procedure PROCESS_EMP?
- a. The errors are stored in the EMP_PROC.ERR file.
 - b. The errors are displayed to the screen when the script file is run.
 - c. The errors are stored in the PROCEDURE_ERRORS data dictionary view.
 - d. **You need to issue the SHOWERRORS command in the SQL Plus environment to see the errors.**
 - e. You need to issue the DISPLAY ERRORS command in the SQL Plus environment to see the errors.

13. Which statement about local dependent objects is true?

- a. They are on different nodes.
- b. They are in a different database.
- c. **They are on the same node in the same database.**
- d. They are on the same node in a different database.

14. You need to create a stored procedure that deletes rows from a table. The name of the table from which the rows are to be deleted is unknown until run time. Which method do you implement while creating such a procedure?

- a. Use SQL command DELETE in the procedure to delete the rows.
- b. **Use DBMS_SQL packaged routines in the procedure to delete the rows.**
- c. Use DBMS_DML packaged routines in the procedure to delete the rows.
- d. Use DBMSDELETE packaged routines in the procedure to delete the rows.
- e. YOU cannot have a delete statement without providing a table name before compile time.

15. Under which situation do you create a server-side procedure?

- a. When the procedure contains no SQL statements.
- b. When the procedure contains no PL/SQL commands.
- c. When the procedure needs to be used by many client applications accessing several remote databases.
- d. **When the procedure needs to be used by many users accessing the same schema objects on a local database.**

16. Examine this procedure:

```
CREATE OR REPLACE PROCEDURE ADD_PLAYER
(V_ID IN NUMBER, V_LAST_NAME VARCHAR2)
IS
BEGIN
  INSERT INTO PLAYER(ID,LAST_NAME).
  VALUES(V_ID,V_LAST_NAME);
  COMMIT;
END;
```

This procedure must invoke the UPD-STAT procedure and pass a parameter. Which statement will successfully invoke this procedure?

- a. EXECUTE UPD_BAT_STAT(V_ID);
- b. **UPD_BAT_STAT(V_ID);**
- c. RUN UPD_BAT_STAT(V_ID);
- d. START UPD_BAT_STAT(V_ID);

17. Match the purity levels to their correct definitions:

Terms

RNTS

RNPS

WNDS

WNPS

Definitions

The function cannot modify the database tables.

The function cannot change the values of the package variables.

The function cannot query database tables.

The function cannot reference the value of public packaged variables.

18. Examine this function:

```
CREATE OR REPLACE FUNCTION CALC_PLAYER_AVG
(V_ID IN PLAYER_BAT_STAT.PLAYER_ID%TYPE)
RETURN NUMBER
IS
V_AVG NUMBER;
SELECTS HITS/AT_BATS
INTO V_AVG
FROM PLAYER_BAT_STAT
WHERE PLAYER_ID=V_ID;
RETURN(V_AVG);
END;
```

This function must be moved to a package. Which additional statement must be added to the function to allow you to continue using the function in the GROUP BY clause of a SELECT statement?

- a. PRAGMA RESTRICT_REFERENCES (CALC_PLAYER_AVG, WNDS, WNPS);
- b. PRAGMA RESTRICT_REFERENCES (CALC_PLAYER_AVG, WNPS);
- c. PRAGMA RESTRICT_REFERENCES (CALC_PLAYER_AVG, RNPS, WNPS);
- d. PRAGMA RESTRICT_REFERENCES (CALC_PLAYER_AVG, ALLOW_GROUP_BY);

19. A programmer develops a procedure, ACCOUNT_TRANSACTION, and has left your company. You are assigned a task to modify this procedure. You want to find all the program units invoking the ACCOUNT_TRANSACTION procedure. How can you find this information?

- a. Query the USER_SOURCE data dictionary view.
- b. Query the USER PROCEDURES data dictionary view.
- c. **Query the USER_DEPENDENCIES data dictionary views.**
- d. Set the SQL Plus environment variable trade code=true and run the ACCOUNT_TRANSACTION procedure.
- e. Set the SQL Plus environment variable DEPENDENCIES=TRUE and run the Account_Transaction procedure.

20. Examine this package.

```
CREATE OR REPLACE PACKAGE BB_PACK
IS
V_MAX_TEAM_SALARY NUMBER(12,2);
PROCEDURE ADD_PLAYER(V_ID NUMBER,V_LAST_NAME
VARCHAR2,V_SALARY NUMBER);
END BB_PACK;
/
CREATE OR REPLACE PACKAGE BODY BB_PACK
IS
PROCEDURE UPD_PLAYER_STAT
(V_ID IN NUMBER,V_AB_IN NUMBER DEFAULT4,V_HITS IN NUMBER)
IS
BEGIN
UPDATE PLAYER_BAT_STAT
SET AT_BATS+V_AB,
HITS=HITS+V_HITS
WHERE PLAYER_ID=V_ID;
COMMIT;
END UPD_PLAYER_STAT;
PROCEDURE ADD_PLAYER
(V_ID IN NUMBER,V_LAST_NAME VARCHAR2,V_SALARY NUMBER)
IS
```

```

BEGIN
INSERT INTO PLAYER(ID,LAST_NAME,SALARY);
UPD_PLAYER_STAT(V_ID,0,0);
END ADD PLAYER;
END BB_PACK;

```

Which statement successfully assigns \$75000000 to the V_MAX_TEAM_SALARY variable from within a stand alone procedure?

- a. V_MAX_TEAM_SALARY := 75000000;
- b. BB_PACK.ADD_PLAYER. V_MAX_TEAM_SALARY := 75000000;
- c. **BB_PACK.V_MAX_TEAM_SALARY := 75000000;**
- d. This variable cannot be assigned a value from outside the package.

21. Which two statements about the overloading feature of packages are true?

- a. **Only local or packaged sub programs can be overloaded.**
- b. Overloading allows different functions with the same name that differ only in their return types.
- c. Overloading allows different subprograms with the same number, type and order of the parameter.
- d. Overloading allows different subprograms with the same name and same number or type of the parameters.
- e. **Overloading allows different subprograms with the same name but different in either number or type or order of parameters.**

22. **Examine this package:**

```

CREATE OR REPLACE manag_emps
IS
Tax_rate CONSTANT NUMBER(5,2):= .28;;
V_id NUMBER;
PROCEDURE insert_emp(p_deptno NUMBER,p-sal NUMBER);
PROCEDURE delete_emp;
PROCEDURE update_emp;
FUNCTION calc_tax(o_sal NUMBER)
    RETURN NUMBER;
END manag_emps;
/
CREATE REPLACE PACKAGE BODY manage_emps
IS
BEGIN
Update emp
SET sal=(sal+p-raise_amt)+sal WHERE empno= v_id;
END;
PROCEDURE insert_emp
    (p_deptno NUMBER, p-sal NUMBER)
IS
BEGIN
INSERT INTO emp(empno, deptno,sal) VALUES(v_id, p_deptno, p_sal);
END insert emp;
PROCEDURE delete_emp
IS
BEGIN
DELETE FROM emp WHERE empno=v_id
END delete_emp;
PROCEDURE Update_emp.
IS

```

```

V_sal NUMBER (10,2);
V_raise NUMBER(10,2);
BEGIN
SELECT Sal INTO v_sal FROM emp WHERE empno=v_id;
IF      v_sal<500 THEN      V_raise:=0. 05;
ELSIF  v_sal<1000 THEN     V_raise:=0. 07;
ELSE    V_raise:=0. 04
END IF;
Update_sal(v_raise);
END update_emp.
FUNCTION calc_tax
(p_sal NUMBER)
RETURN NUMBER
IS
BEGIN
    RETURN p_sal* tax-rate;
    END calc_tax;
END MANAGE_emp;
/

```

What is the name of the private procedure in the package?

- a. CALC_TAX
- b. INSERT_EMP
- c. **UPDATE_SAL**
- d. DELETE_EMP
- e. UPDATE_EMP
- f. MANAGE_EMP

23. Examine the code:

```

CREATE OR REPLACE TRIGGER update_emp
AFTER UPDATE ON emp
BEGIN
INSERT INTO audit_table (who, audited)
VALUES(USER, SYSDATE);
END;

```

You issue an update command on the EMP table that result in changing ten rows. How many rows are inserted into the AUDIT_TABLE?

- a. 1
- b. 10
- c. none
- d. Value equal to the number of rows in the EMP table

24. All users currently have INSERT privileges on the PLAYER table. You want only your users to insert into this table using the ADD_PLAYER procedure. Which two actions must you take?

- a. GRANT SELECT ON ADD_PLAYER TO PUBLIC;
- b. **GRANT EXECUTE ON ADD_PLAYER TO PUBLIC;**
- c. GRANT INSERT ON PLAYER TO PUBLIC;
- d. GRANT EXECUTE, INSERT ON ADD_PLAYER TO PUBLIC;
- e. REVOKE INSERT ON PLAYER FROM PUBLIC;

25. Which oracle supply package allows you to run jobs that use defined times?

- a. **DBMS_JOB**
- b. DBMS_RUN

- c. DBMS_PIPE
- d. DBMS_SQL

26. You need to drop a table from within a stored procedure. How do you implement this?

- a. You cannot drop a table from a stored procedure.
- b. Use the DROP command in the procedure to drop the table.
- c. Use the DBMS_DDL packaged routines in the procedure to drop the table.
- d. Use the DBMS_SQL packaged routines in the procedure to drop the table. ans
- e. Use the DBMS_DROP packaged routines in the procedure to drop the table.

27. Which data dictionary view gives you the names and the source code of all the procedures you have created?

- a. USER_SOURCE ans
- b. USER_OBJECTS
- c. USER_PROCEDURES
- d. USER_SUBPROGRAMS

28. Examine this package:

```
CREATE OR REPLACE PACKAGE BB_PACK
IS
V_MAX_TEAM_SALARY NUMBER(12,2);
PROCEDURE ADD_PLAYER(V_ID IN NUMBER, V_LAST_NAME)
VARCHAR2(V_SALARY NUMBER);
END BB_PACK;
/
CREATE OR REPLACE PACKAGE BODY BB_PACK
IS
V_PLAYER_AVG NUMBER(4,3);
PROCEDURE UPD_PLAYER_STAT
V_ID IN NUMBER, V_AB IN NUMBER DEFAULT4, V_HITS IN NUMBER)
IS
BEGIN
UPDATE PLAYER_BAT_STAT
SET ADD_BAT=ADD_BATS+V_AB,
HITS=HITS+V_HITS
WHERE PLAYER_ID=V_ID;
COMMIT;
VALIDATE_PLAYER_STAT(V_ID);
END UPD_PLAYER_STAT;
PROCEDURE ADD_PLAYER
(V_ID IN NUMBER, V_LAST_NAME, VARCHAR2, V_SALARY IN NUMBER);
IS
BEGIN
INSERT INTO PLAYER (ID, LAST_NAME, SALARY) VALUES(V_ID, V_LAST_NAME, V_SALARY);
UPD_PLAYER_STAT(V_ID,0,0);
END ADD_PLAYER;
END BB_PACK;
```

Which kind of packaged variable is V_MAX_TEAM_SALARY?

- a. PRIVATE
- b. PUBLIC ans
- c. IN
- d. OUT

29. Examine this trigger:

```
CREATE OR REPLACE TRIGGER UPD_TEAM_SALARY
AFTER INSERT OR UPDATE OR DELETE ON PLAYER
FOR EACH ROW
BEGIN
UPDATE TEAM
SET TOT_SALARY=TOT_SALARY+:NEW SALARY.
WHERE ID=:NEW:TEAM_ID;
```

You will be adding additional code later but for now you want the current block to fire when updating the salary column. Which solution should you use to verify that the user is performing an update on the salary column?

- a. ROW_UPDATE('SALARY')
- b. UPDATING('SALARY') ans
- c. CHANGING('SALARY')
- d. COLUMN_UPDATE('SALARY')

30. Examine this package:

```
CREATE OR REPLACE PACKAGE discounts IS
G_ID NUMBER:=7839;
DISCOUNT_RATE NUMBER 0. 00;
PROCEDURE DISPLAY_PRICE (V_PRICE NUMBER);
END DISCOUNTS;
/
CREATE OR REPLACE PACKAGE BODY discounts
IS
PROCEDURE DISPLAY_PRICE (V_PRICE_NUMBER)
IS
BEGIN DBMS_OUTPUT.PUT_LINE('DISCOUNTED'||2_4 (V_PRICENVL(DISCOUNT_RATE, 1)))
END DISPLAY_PRICE;
BEGIN DISCOUNT_RATE:=0. 10;
END DISCOUNTS;
/
```

Which statement is true?

- a. The value of DISCOUNT_RATE always remain 0.00 in a session.
- b. The value of DISCOUNT_RATE is set to 0.10 each time the package is invoked in a session.
- c. The value of DISCOUNT_RATE is set to 1 each time the procedure DISPLAY_PRICE is invoked.
- d. The value of DISCOUNT_RATE is set to 0.10 when the package is invoked for the first time in a session. ans

31. Examine this package:

```
CREATE OR REPLACE PACKAGE BB_PACK
V_MAX_TEAM_SALARY NUMBER(12,2);
PROCEDURE ADD_PLAYER(V_ID IN NUMBER, V_LAST_NAME
VARCHAR2, V_SALARY NUMBER);
DB_PACK;/ CREATE OR REPLACE PACKAGE BODY BB_PACK
IS
V_WHERE_AVG NUMBER(4,3);
PROCEDURE UPD_PLAYER_STAT
(V_ID IN NUMBER, V_AVG IN NUMBER DEFAULT 4,V_HITS IN NUMBER)
IS
BEGIN
```

```

UPDATE PLAYER_BAT_STAT
SET AT_BATS=AT_BATS+V_AB,
HITS=HITS+V_HITS
WHERE PLAYER_ID=V_ID;
COMMIT;
VALIDATE_PLAYER_STAT(V_ID);
END UPD_PLAYER_STAT;
PROCEDURE ADD-PLAYER
(V_ID IN NUMBER, V_LAST_NAME VARCHAR2, V_SALARY NUMBER)
IS
BEGIN
INSERT INTO PLAYER(ID, LAST_NAME, SALARY) VALUES(V_ID, V_LAST_NAME, V_SALARY);
UPD_PLAYER_STAT(V_ID,0,0);
END ADD-PLAYER;
END BB_PACK;

```

An outside procedure VALIDATE_PLAYER_STAT is executed from this package. What will happen when this procedure changes?

- a. The package specification is dropped.
- b. The package specification is invalidated.
- c. The package is invalid to begin with.
- d. The package body is invalidated

1. Examine this function:

```

CREATE OR REPLACE FUNCTION CALC_PLAYER_AVG
(V_ID in PLAYER_BAT_STAT.PLAYER_ID%TYPE)
RETURN NUMBER IS V_AVG NUMBER;
BEGIN
SELECT HITS / AT_BATS INTO V_AVG
FROM PLAYER_BAT_STAT
WHERE PLAYER_ID = V_ID;
RETURN (V_AVG);
END;

```

Which statement will successfully invoke this function in SQL *Plus?

- A. SELECT CALC_PLAYER_AVG(PLAYER_ID) FROM PLAYER_BAT_STAT;
- B. EXECUTE CALC_PLAYER_AVG(31);
- C. CALC_PLAYER('RUTH');
- D. CALC_PLAYER_AVG(31);
- E. START CALC_PLAYER_AVG(31) **Ans: a**

2. Which three are true statements about dependent objects? (Choose three)

- A. Invalid objects cannot be described.
- B. An object with status of invalid cannot be a referenced object.
- C. The Oracle server automatically records dependencies among objects.
- D. All schema objects have a status that is recorded in the data dictionary.
- E. You can view whether an object is valid or invalid in the USER_STATUS data dictionary view.
- F. You can view whether an object is valid or invalid in the USER_OBJECTS data dictionary view. **Ans: cdf**

3. You have created a stored procedure DELETE_TEMP_TABLE that uses dynamic SQL to remove a table in your schema. You have granted the EXECUTE privilege to user A on this procedure. When user A executes the DELETE_TEMP_TABLE procedure, under whose privileges are the operations performed by default?

- A. SYS privileges
- B. Your privileges
- C. Public privileges
- D. User A's privileges
- E. User A cannot execute your procedure that has dynamic SQL. **Ans: b**

4. Examine this code:

```
CREATE OR REPLACE PROCEDURE add_dept (p_dept_name VARCHAR2 DEFAULT 'placeholder',
p_location VARCHAR2 DEFAULT 'Boston')
IS BEGIN INSERT INTO departments VALUES (dept_id_seq.NEXTVAL, p_dept_name, p_location); END
add_dept; /
```

Which three are valid calls to the add_dept procedure? (Choose three)

- A. add_dept;
- B. add_dept('Accounting');
- C. add_dept(, 'New York');
- D. add_dept(p_location=>'New York'); **Ans: abd**

5. Which two statements about packages are true? (Choose two)

- A. Packages can be nested.
- B. You can pass parameters to packages.
- C. A package is loaded into memory each time it is invoked.
- D. The contents of packages can be shared by many applications.
- E. You can achieve information hiding by making package constructs private. **Ans: de**

6. Which two programming constructs can be grouped within a package? (Choose two)

- A. Cursor
- B. Constant
- C. Trigger
- D. Sequence

View

Ans: ab

7. Which two statements describe the state of a package variable after executing the package in which it is declared? (Choose two)

- A. It persists across transactions within a session.
- B. It persists from session to session for the same user.
- C. It does not persist across transaction within a session.
- D. It persists from user to user when the package is invoked.
- E. It does not persist from session to session for the same user. **Ans: ae**

8. Which code can you use to ensure that the salary is not increased by more than 10% at a time nor is it ever decreased?

- A. ALTER TABLE emp ADD CONSTRAINT ck_sal CHECK (sal BETWEEN sal AND sal * 1.1);
- B. CREATE OR REPLACE TRIGGER check_sal BEFORE UPDATE OF sal ON emp FOR EACH ROW WHEN (new.sal < old.sal OR new.sal > old.sal * 1.1) BEGIN RAISE_APPLICATION_ERROR (-20508, 'Do not decrease salary not increase by more than 10%'); END;

C. CREATE OR REPLACE TRIGGER check_sal BEFORE UPDATE OF sal ON emp WHEN (new.sal < old.sal OR new.sal > old.sal * 1.1) BEGIN RAISE_APPLICATION_ERROR (-20508, 'Do not decrease salary not increase by more than 10%'); END;

D. CREATE OR REPLACE TRIGGER check_sal AFTER UPDATE OF sal ON emp WHEN (new.sal < old.sal OR -new.sal > old.sal * 1.1) BEGIN RAISE_APPLICATION_ERROR (-20508, 'Do not decrease salary not increase by more than 10%'); END; **Ans: b**

9. Examine this code:

```
CREATE OR REPLACE PACKAGE bonus IS g_max_bonus NUMBER := .99;
FUNCTION calc_bonus (p_emp_id NUMBER) RETURN NUMBER;
FUNCTION calc_salary (p_emp_id NUMBER) RETURN NUMBER; END;
```

```
CREATE OR REPLACE PACKAGE BODY bonus IS v_salary employees.salary%TYPE;
v_bonus employees.commission_pct%TYPE; FUNCTION calc_bonus (p_emp_id NUMBER) RETURN NUMBER IS
BEGIN
SELECT salary, commission_pct INTO v_salary, v_bonus FROM employees WHERE employee_id = p_emp_id; RETURN v_bonus * v_salary; END calc_bonus FUNCTION calc_salary (p_emp_id NUMBER) RETURN NUMBER IS
BEGIN
SELECT salary, commission_pct INTO v_salary, v_bonus FROM employees WHERE employees RETURN v_bonus * v_salary + v_salary;
END calc_salary;
END bonus; /
```

Which statement is true?

- A. You can call the BONUS.CALC_SALARY packaged function from an INSERT command against EMPLOYEES table.
- B. You can call the BONUS.CALC_SALARY packaged function from a SELECT command against EMPLOYEES table.
- C. You can call the BONUS.CALC_SALARY packaged function from a DELETE command against EMPLOYEES table.
- D. You can call the BONUS.CALC_SALARY packaged function from an UPDATE command against EMPLOYEES table. **Ans: b**

10. Which statement is valid when removing procedures?

- A. Use a drop procedure statement to drop a standalone procedure.
- B. Use a drop procedure statement to drop a procedure that is part of a package. Then recompile the package specification.
- C. Use a drop procedure statement to drop a procedure that is part of a package. Then recompile the package body.
- D. For faster removal and re-creation, do not use a drop procedure statement. Instead, recompile the procedure using the alter procedure statement with the REUSE SETTINGS clause. **Ans: a**

11. Examine this package:

```
CREATE OR REPLACE PACKAGE BB_PACK IS V_MAX_TEAM:SALAR NUMBER(12,2);
PROCEDURE ADD_PLAYER(V_ID IN NUMBER, V_LAST_NAME VARCHAR2, V_SALARY NUMBER);
END BB_PACK; /
CREATE OR REPLACE PACKAGE BODY BB_PACK IS PROCEDURE UPD_PLAYER_STAT (V_ID IN NUMBER, V_AB IN NUMBER DEFAULT 4, V_HITS IN NUMBER) IS BEGIN UPDATE
PLAYER_BAT_STAT SET AT_BATS = AT_BATS + V_AB, HITS = HITS + V_HITS WHERE PLAYER_ID = V_ID;
COMMIT;
```

```

END UPD_PLAYER_STAT;
PROCEDURE ADD_PLAYER (V_ID IN NUMBER, V_LAST_NAME VARCHAR2, V_SALARY NUMBER)
IS BEGIN INSERT INTO PLAYER(ID,LAST_NAME,SALARY) VALUES (V_ID, V_LAST_NAME,
V_SALARY);
UPD_PLAYER_STAT(V_ID,0,0); END ADD_PLAYER;
END BB_PACK;

```

**You make a change to the body of the BB_PACK package. The BB_PACK body is recompiled.
What happens if the stand alone procedure VALIDATE_PLAYER_STAT references this package?**

- a. VALIDATE_PLAYER_STAT cannot recompile and must be recreated.
- b. VALIDATE_PLAYER_STAT is not invalidated.
- c. VALIDATE_PLAYER_STAT is invalidated.
- d. VALIDATE_PLAYER_STAT and BB_PACK are invalidated. **Ans: b**

12. You need to create a trigger on the EMP table that monitors every row that is changed and places this information into the AUDIT_TABLE. What type of trigger do you create?

- A. FOR EACH ROW trigger on the EMP table.
- B. Statement-level trigger on the EMP table.
- C. FOR EACH ROW trigger on the AUDIT_TABLE table.
- D. Statement-level trigger on the AUDIT_TABLE table.
- E. FOR EACH ROW statement-level trigger on the EMP table. **Ans: a**

13. Which statements are true? (Choose all that apply)

- A. If errors occur during the compilation of a trigger, the trigger is still created.
- B. If errors occur during the compilation of a trigger you can go into SQL *Plus and query the USER_TRIGGERS data dictionary view to see the compilation errors.
- C. If errors occur during the compilation of a trigger you can use the SHOW ERRORS command within iSQL *Plus to see the compilation errors.
- D. If errors occur during the compilation of a trigger you can go into SQL *Plus and query the USER_ERRORS data dictionary view to see compilation errors. **Ans: acd**

14. Which two dictionary views track dependencies? (Choose two)

- A. USER_SOURCE
- B. UTL_DEPTREE.
- C. USER_OBJECTS
- D. DEPTREE_TEMPTAB
- E. USER_DEPENDENCIES
- F. DBA_DEPENDENT_OBJECTS **Ans: de**

15. Given a function CALCTAX:

```

CREATE OR REPLACE FUNCTION calctax (sal NUMBER) RETURN NUMBER IS
BEGIN RETURN (sal * 0.05); END;

```

If you want to run the above function from the SQL *Plus prompt, which statement is true?

- A. You need to execute the command CALCTAX(1000);
- B. You need to execute the command EXECUTE FUNCTION calctax;
- C. You need to create a SQL *Plus environment variable X and issue the command :X := CALCTAX(1000);
- D. You need to create a SQL *Plus environment variable X and issue the command EXECUTE :X := CALCTAX;
- E. You need to create a SQL *Plus environment variable X and issue the command EXECUTE :X := CALCTAX(1000); **Ans: d**

16. What happens during the execute phase with dynamic SQL for INSERT, UPDATE, and DELETE operations?

- A. The rows are selected and ordered.
- B. The validity of the SQL statement is established.
- C. An area of memory is established to process the SQL statement.
- D. The SQL statement is run and the number of rows processed is returned.
- E. The area of memory established to process the SQL statement is released. **Ans: d**

17. What part of a database trigger determines the number of times the trigger body executes?

- A. Trigger type
- B. Trigger body
- C. Trigger event
- D. Trigger timing **Ans: a**

18. Examine this code:

```
CREATE OR REPLACE FUNCTION gen_email_name (p_first_name VARCHAR2, p_last_name VARCHAR2, p_id NUMBER) RETURN VARCHAR2 is v_email_name VARCHAR2(19);
BEGIN v_email_home := SUBSTR(p_first_name, 1, 1) || SUBSTR(p_last_name, 1, 7) || '@Oracle.com';
UPDATE employees SET email = v_email_name WHERE employee_id = p_id;
RETURN v_email_name; END;
```

You run this SELECT statement:

```
SELECT first_name, last_name gen_email_name(first_name, last_name, 108) EMAIL FROM employees;
```

What occurs?

- A. Employee 108 has his email name updated based on the return result of the function.
- B. The statement fails because functions called from SQL expressions cannot perform DML.
- C. The statement fails because the functions does not contain code to end the transaction.
- D. The SQL statement executes successfully, because UPDATE and DELETE statements are ignoring in stored functions called from SQL expressions.
- E. The SQL statement executes successfully and control is passed to the calling environment. **Ans: b**

19. Which table should you query to determine when your procedure was last compiled?

- A. USER_PROCEDURES
- B. USER_PROCS
- C. USER_OBJECTS
- D. USER_PLSQL_UNITS **Ans: c**

20. Examine this code:

```
CREATE OR REPLACE TRIGGER secure_emp BEFORE LOGON ON employees
BEGIN
IF (TO_CHAR(SYSDATE, 'DY') IN ('SAT', 'SUN')) OR (TO_CHAR(SYSDATE, 'HH24:MI') NOT BETWEEN '08:00' AND '18:00')
THEN RAISE_APPLICATION_ERROR (-20500, 'You may insert into the EMPLOYEES table only during business hours.');
END IF; END; /
```

What type of trigger is it?

- A. DML trigger
- B. INSTEAD OF trigger
- C. Application trigger
- D. System event trigger
- E. This is an invalid trigger. **Ans: e**

21. Examine this package:

```
CREATE OR REPLACE PACKAGE discounts IS g_id NUMBER := 7829;
discount_rate NUMBER := 0.00;
PROCEDURE display_price (p_price NUMBER);
END discounts; /
CREATE OR REPLACE PACKAGE BODY discounts IS PROCEDURE display_price (p_price NUMBER)
IS BEGIN DBMS_OUTPUT.PUT_LINE('Discounted'|| TO_CHAR(p_price*NVL(discount_rate, 1)));
END display_price;
BEGIN discount_rate := 0.10;
END discounts; /
```

Which statement is true?

- A. The value of DISCOUNT_RATE always remains 0.00 in a session.
- B. The value of DISCOUNT_RATE is set to 0.10 each time the package is invoked in a session.
- C. The value of DISCOUNT_RATE is set to 1.00 each time the procedure DISPLAY_PRICE is invoked.
- D. The value of DISCOUNT_RATE is set to 0.10 when the package is invoked for the first time in a session. **Ans: d**

22. Examine this code:

```
CREATE OR REPLACE TRIGGER update_emp AFTER UPDATE ON emp BEGIN INSERT INTO
audit_table (who, dated) VALUES (USER, SYSDATE); END;
```

You issue an UPDATE command in the EMP table that result in changing 10 rows. How many rows are inserted into the AUDIT_TABLE?

- A. 1
- B. 10
- C. None
- D. A value equal to the number of rows in the EMP table. **Ans: a**

23. Examine this package:

```
CREATE OR REPLACE PACKAGE BB_PACK IS V_MAX_TEAM_SALARY NUMBER(12,2);
PROCEDURE ADD_PLAYER(V_ID IN NUMBER, V_LAST_NAME VARCHAR2, V_SALARY NUMBER;
END BB_PACK; /
CREATE OR REPLACE PACKAGE BODY BB_PACK IS
PROCEDURE UPD_PLAYER_STAT (V_ID IN NUMBER, V_AB IN NUMBER DEFAULT 4, V_HITS IN
NUMBER) IS
BEGIN UPDATE PLAYER_BAT_STAT SET AT_BATS = AT_BATS + V_AB, HITS = HITS + V_HITS
WHERE PLAYER_ID = V_ID) COMMIT;
END UPD_PLAYER_STAT;
PROCEDURE ADD_PLAYER (V_ID IN NUMBER, V_LAST_NAME VARCHAR2, V_SALARY NUMBER)
IS BEGIN INSERT INTO PLAYER(ID,LAST_NAME,SALARY) VALUES (V_ID, V_LAST_NAME,
V_SALARY);
UPD_PLAYER_STAT(V_ID,0.0);
END ADD_PLAYER; END BB_PACK;
```

Which statement will successfully assign \$75,000,000 to the V_MAX_TEAM_SALARY variable from within a stand-alone procedure?

- A. V_MAX_TEAM_SALARY := 7500000;
- B. BB_PACK.ADD_PLAYER.V_MAX_TEAM_SALARY := 75000000;
- C. BB_PACK.V_MAX_TEAM_SALARY := 75000000;
- D. This variable cannot be assigned a value from outside the package. **Ans: c**

- 24. There is a CUSTOMER table in a schema that has a public synonym CUSTOMER and you are granted all object privileges on it. You have a procedure PROCESS_CUSTOMER that processes customer information that is in the public synonym CUSTOMER table. You have just created a new table called CUSTOMER within your schema. Which statement is true?**

- A. Creating the table has no effect and procedure PROCESS_CUSTOMER still accesses data from public synonym CUSTOMER table.
- B. If the structure of your CUSTOMER table is the same as the public synonym CUSTOMER table then the procedure PROCESS_CUSTOMER is invalidated and gives compilation errors.
- C. If the structure of your CUSTOMER table is entirely different from the public synonym CUSTOMER table then the procedure PROCESS_CUSTOMER successfully recompiles and accesses your CUSTOMER table.
- D. If the structure of your CUSTOMER table is the same as the public synonym CUSTOMER table then the procedure PROCESS_CUSTOMER successfully recompiles when invoked and accesses your CUSTOMER table. **Ans: d**

25. Which two statements about packages are true? (Choose two)

- A. Both the specification and body are required components of a package.
- B. The package specification is optional, but the package body is required.
- C. The package specification is required, but the package body is optional.
- D. The specification and body of the package are stored together in the database.
- E. The specification and body of the package are stored separately in the database. **Ans: ce**

26. When creating a function in SQL *Plus, you receive this message:

"Warning: Function created with compilation errors."

Which command can you issue to see the actual error message?

- A. SHOW FUNCTION_ERROR
- B. SHOW USER_ERRORS
- C. SHOW ERRORS
- D. SHOW ALL_ERRORS **Ans: c**

27. Which four triggering events can cause a trigger to fire? (Choose four)

- A. A specific error or any errors occurs.
- B. A database is shut down or started up.
- C. A specific user or any user logs on or off.
- D. A user executes a CREATE or an ALTER table statement.
- E. A user executes a SELECT statement with an ORDER BY clause.
- F. A user executes a JOIN statement that uses four or more tables. **Ans: abcd**

28. Examine this procedure:

```
CREATE OR REPLACE PROCEDURE ADD_PLAYER (V_ID IN NUMBER, V_LAST_NAME
VARCHAR2) IS
BEGIN
    INSERT INTO PLAYER (ID,LAST_NAME) VALUES (V_ID,
V_LAST_NAME); COMMIT; END;
```

This procedure must invoke the UPD_BAT_STAT procedure and pass a parameter.

Which statement, when added to the above procedure will successfully invoke the UPD_BAT_STAT procedure?

- A. EXECUTE UPD_BAT_STAT(V_ID);
- B. UPD_BAT_STAT(V_ID);
- C. RUN UPD_BAT_STAT(V_ID);
- D. START UPD_BAT_STAT(V_ID); **Ans: b**

29. Which statement about triggers is true?

- A. You use an application trigger to fire when a DELETE statement occurs.
- B. You use a database trigger to fire when an INSERT statement occurs.

- C. You use a system event trigger to fire when an UPDATE statement occurs.
- D. You use INSTEAD OF trigger to fire when a SELECT statement occurs. **Ans: b**

30. You want to create a PL/SQL block of code that calculates discounts on customer orders. This code will be invoked from several places, but only within the program unit ORDERTOTAL. What is the most appropriate location to store the code that calculates the discounts?

- A. A stored procedure on the server.
- B. A block of code in a PL/SQL library.
- C. A standalone procedure on the client machine.
- D. A block of code in the body of the program unit ORDERTOTAL.
- E. A local subprogram defined within the program unit ORDERTOTAL. **Ans: e**

31. Which type of argument passes a value from a procedure to the calling environment?

- A. VARCHAR2
- B. BOOLEAN
- C. OUT
- D. IN **Ans: c**

32. You create a DML trigger. For the timing information, which is valid with a DML trigger?

- A. DURING
- B. INSTEAD
- C. ON SHUTDOWN
- D. BEFORE
- E. ON STATEMENT EXECUTION **Ans: d**

33. You are about to change the arguments of the CALC_TEAM_AVG function. Which dictionary view can you query to determine the names of the procedures and functions that invoke the CALC_TEAM_AVG function?

- A. USER_PROC_DEPENDS
- B. USER_DEPENDENCIES
- C. USER_REFERENCES
- D. USER_SOURCE **Ans: b**

34. A CALL statement inside the trigger body enables you to call _____.

- A. A package.
- B. A stored function.
- C. A stored procedure.
- D. Another database trigger. **Ans: c**

35. You need to remove the database trigger BUSINESS_HOUR. Which command do you use to remove the trigger in the SQL *Plus environment?

- A. DROP TRIGGER business_hour;
- B. DELETE TRIGGER business_hour;
- C. REMOVE TRIGGER business_hour;
- D. ALTER TRIGGER business_hour REMOVE;
- E. DELETE FROM USER_TRIGGERS WHERE TRIGGER_NAME = 'BUSINESS_HOUR'; **Ans: a**

36. How can you migrate from a LONG to a LOB data type for a column?

- A. Use the DBMS_MANAGE_LOB.MIGRATE procedure.
- B. Use the UTL_MANAGE_LOB.MIGRATE procedure.
- C. Use the DBMS_LOB.MIGRATE procedure.
- D. Use the ALTER TABLE command.
- E. You cannot migrate from a LONG to a LOB date type for a column. **Ans: d**

37. Examine this procedure:

```
CREATE OR REPLACE PROCEDURE INSERT_TEAM (V_ID in NUMBER, V_CITY in VARCHAR2
DEFAULT 'AUSTIN', V_NAME in VARCHAR2) IS
BEGIN
    INSERT INTO TEAM (id, city, name) VALUES (v_id, v_city, v_name); COMMIT;
END
```

Which two statements will successfully invoke this procedure in SQL *Plus? (Choose two)

- A. EXECUTE INSERT_TEAM;
- B. EXECUTE INSERT_TEAM(3, V_NAME=>'LONGHORNS', V_CITY=>'AUSTIN');
- C. EXECUTE INSERT_TEAM(3, 'AUSTIN','LONGHORNS');
- D. EXECUTE INSERT_TEAM (V_ID := V_NAME := 'LONGHORNS', V_CITY := 'AUSTIN');
- E. EXECUTE INSERT_TEAM (3, 'LONGHORNS'); **Ans: bc**

38. To be callable from a SQL expression, a user-defined function must do what?

- A. Be stored only in the database.
- B. Have both IN and OUT parameters.
- C. Use the positional notation for parameters.
- D. Return a BOOLEAN or VARCHAR2 data type. **Ans: c**

39. Which two describe a stored procedure? (Choose two)

- A. A stored procedure is typically written in SQL.
- B. A stored procedure is a named PL/SQL block that can accept parameters.
- C. A stored procedure is a type of PL/SQL subprogram that performs an action.
- D. A stored procedure has three parts: the specification, the body, and the exception handler part.
- E. The executable section of a stored procedure contains statements that assign values, control execution, and return values to the calling environment. **Ans: bc**

40. Examine this code:

```
CREATE OR REPLACE PROCEDURE add_dept (p_name departments.department_name%TYPE
DEFAULT 'unknown', p_loc departments.location_id%TYPE DEFAULT 1700) IS
BEGIN
    INSERT      INTO      departments(department_id,      department_name,      location_id)
VALUES(dept_seq.NEXTVAL,p_name, p_loc); END add_dept; /
```

You created the add_dept procedure above, and you now invoke the procedure in SQL *Plus.

Which four are valid invocations? (Choose four)

- A. EXECUTE add_dept(p_loc=>2500)
- B. EXECUTE add_dept('Education', 2500)
- C. EXECUTE add_dept('2500', p_loc =>2500)
- D. EXECUTE add_dept(p_name=>'Education', 2500)
- E. EXECUTE add_dept(p_loc=>2500, p_name=>'Education') **Ans: abce**

41. Which three are valid ways to minimize dependency failure? (Choose three)

- A. Querying with the SELECT * notification.
- B. Declaring variables with the %TYPE attribute.
- C. Specifying schema names when referencing objects.
- D. Declaring records by using the %ROWTYPE attribute.

- E. Specifying package.procedure notation while executing procedures. **Ans: abd**

42. Which two dopes the INSTEAD OF clause in a trigger identify? (Choose two)

- A. The view associated with the trigger.
- B. The table associated with the trigger.
- C. The event associated with the trigger.
- D. The package associated with the trigger.
- E. The statement level or for each row association to the trigger. **Ans: ac**

43. Examine this package:

```
CREATE OR REPLACE PACKAGE manage_emps IS tax_rate CONSTANT NUMBER(5,2) := .28;
v_id NUMBER;
PROCEDURE insert_emp (p_deptno NUMBER, p_sal NUMBER);
PROCEDURE delete_emp; PROCEDURE update_emp;
FUNCTION calc_tax (p_sal NUMBER) RETURN NUMBER;
END manage_emps;

CREATE OR REPLACE PACKAGE BODY manage_emps IS
PROCEDURE update_sal (p_raise_amt NUMBER) IS BEGIN
    UPDATE emp SET sal = (sal * p_raise_amt) + sal WHERE empno = v_id;
END;
PROCEDURE insert_emp (p_deptno NUMBER, p_sal NUMBER) IS BEGIN
    INSERT INTO emp(empno, deptno, sal) VALUES(v_id, p_deptno, p_sal);
END insert_emp;
PROCEDURE delete_emp IS BEGIN
    DELETE FROM emp WHERE empno = v_id;
END delete_emp;
PROCEDURE update_emp IS
    v_sal NUMBER(10, 2); v_raise NUMBER(10, 2);
BEGIN
    SELECT sal INTO v_sal FROM emp WHERE empno = v_id;
    IF v_sal < 500 THEN v_raise := .05;
    ELSIF v_sal < 1000 THEN v_raise := .07;
    ELSE v_raise := .04;
    END IF;
    update_sal(v_raise);
END update_emp;
FUNCTION calc_tax (p_sal NUMBER) RETURN NUMBER IS BEGIN
    RETURN p_sal * tax_rate;
END calc_tax;
END manage_emps; /
```

What is the name of the private procedure in this package?

- A. CALC_TAX
- B. INSERT_EMP
- C. UPDATE_SAL
- D. DELETE_EMP
- E. UPDATE_EMP
- F. MANAGE_EMPS **Ans: c**

44. What can you do with the DBMS_LOB package?

- A. Use the DBMS_LOB.WRITE procedure to write data to a BFILE.
- B. Use the DBMS_LOB.BFILENAME function to locate an external BFILE.
- C. Use the DBMS_LOB.FILEEXISTS function to find the location of a BFILE.
- D. Use the DBMS_LOB.FILECLOSE procedure to close the file being accessed. **Ans: d**

45. Examine this package:

```
CREATE OR REPLACE PACKAGE BB_PACK IS
    V_MAX_TEAM_SALARY NUMBER(12,2);
    PROCEDURE ADD_PLAYER(V_ID IN NUMBER, V_LAST_NAME VARCHAR2, V_SALARY NUMBER);
END BB_PACK;
```

```

CREATE OR REPLACE PACKAGE BODY BB_PACK IS V_PLAYER_AVG NUMBER(4,3); PROCEDURE
UPD_PLAYER_STAT V_ID IN NUMBER, V_AB IN NUMBER DEFAULT 4, V_HITS IN NUMBER) IS BEGIN
UPDATE PLAYER_BAT_STAT SET AT_BATS = AT_BATS + V_AB, HITS = HITS + V_HITS WHERE
PLAYER_ID = V_ID; COMMIT;
VALIDATE_PLAYER_STAT(V_ID);
END UPD_PLAYER_STAT;
PROCEDURE ADD_PLAYER (V_ID IN NUMBER, V_LAST_NAME VARCHAR2, V_SALARY NUMBER)
IS BEGIN INSERT INTO PLAYER(ID,LAST_NAME,SALARY) VALUES (V_ID, V_LAST_NAME,
V_SALARY);
UPD_PLAYER_STAT(V_ID,0,0);
END ADD_PLAYER;
END BB_PACK /

```

Which statement will successfully assign .333 to the V_PLAYER_AVG variable from a procedure outside the package?

- A. V_PLAYER_AVG := .333;
- B. BB_PACK.UPD_PLAYER_STAT.V_PLAYER_AVG := .333;
- C. BB_PACK.V_PLAYER_AVG := .333;
- D. This variable cannot be assigned a value from outside of the package. **Ans: d**

46. Examine this code:

```

CREATE OR REPLACE PACKAGE comm_package IS g_comm NUMBER := 10;
PROCEDURE reset_comm(p_comm IN NUMBER);
END comm_package;

```

User Jones executes the following code at 9:01am:

```
EXECUTE comm_package.g_comm := 15
```

User Smith executes the following code at 9:05am:

```
EXECUTE comm_package.g_comm := 20
```

Which statement is true?

- A. g_comm has a value of 15 at 9:06am for Smith.
- B. g_comm has a value of 15 at 9:06am for Jones.
- C. g_comm has a value of 20 at 9:06am for both Jones and Smith.
- D. g_comm has a value of 15 at 9:03 am for both Jones and Smith.
- E. g_comm has a value of 10 at 9:06am for both Jones and Smith.
- F. g_comm has a value of 10 at 9:03am for both Jones and Smith **Ans: b**

47. Examine this code:

```

CREATE OR REPLACE FUNCTION gen_email_name (p_first_name VARCHAR2, p_last_name
VARCHAR2, p_id NUMBER) RETURN VARCHAR2 IS v_email_name VARCHAR2(19=; BEGIN
v_email_name := SUBSTR(p_first_name, 1, 1) || SUBSTR(p_last_name, 1, 7) || '@Oracle.com'; UPDATE
employees SET email = v_email_name WHERE employee_id = p_id; RETURN v_email_name; END;

```

Which statement removes the function?

- A. DROP gen_email_name;
- B. REMOVE gen_email_name;
- C. DELETE gen_email_name;
- D. ***MISSING*** **Ans: d**

48. Examine this procedure:

```

CREATE OR REPLACE PROCEDURE UPD_BAT_STAT (V_ID IN NUMBER DEFAULT 10, V_AB IN
NUMBER DEFAULT 4) IS BEGIN UPDATE PLAYER_BAT_STAT SET AT_BATS = AT_BATS + V_AB
WHERE PLAYER_ID = V_ID; COMMIT; END;

```

Which two statements will successfully invoke this procedure in SQL *Plus? (Choose two)

- A. EXECUTE UPD_BAT_STAT;

- B. EXECUTE UPD_BAT_STAT(V_AB=>10, V_ID=>31);
- C. EXECUTE UPD_BAT_STAT(31, 'FOUR','TWO');
- D. UPD_BAT_STAT(V_AB=>10, V_ID=>31);
- E. RUN UPD_BAT_STAT; **Ans: ab**

49. Examine this code:

```
CREATE OR REPLACE PROCEDURE audit_action (p_who VARCHAR2) AS BEGIN INSERT INTO audit(schema_user) VALUES(p_who); END audit_action; /
CREATE OR REPLACE TRIGGER watch_it AFTER LOGON ON DATABASE.
CALL audit_action(ora_login_user) ;
```

What does this trigger do?

- A. The trigger records an audit trail when a user makes changes to the database.
- B. The trigger marks the user as logged on to the database before an audit statement is issued.
- C. The trigger invoked the procedure audit_action each time a user logs on to his/her schema and adds the username to the audit table.
- D. The trigger invokes the procedure audit_action each time a user logs on to the database and adds the username to the audit table. **Ans: d**

50. Which view displays indirect dependencies, indenting each dependency?

- A. DEPTREE
- B. IDEPTREE
- C. INDENT_TREE
- D. I_DEPT_TREE **Ans: b**

51. The OLD and NEW qualifiers can be used in which type of trigger?

- A. Row level DML trigger
- B. Row level system trigger
- C. Statement level DML trigger
- D. Row level application trigger
- E. Statement level system trigger
- F. Statement level application trigger **Ans: a**

52. Which statement is true?

- A. Stored functions can be called from the SELECT and WHERE clauses only.
- B. Stored functions do not permit calculations that involve database links in a distributed environment.
- C. Stored functions cannot manipulate new types of data, such as longitude and latitude.
- D. Stored functions can increase the efficiency of queries by performing functions in the query rather than in the application. **Ans: d**

53. Examine the trigger:

```
CREATE OR REPLACE TRIGGER Emp_count AFTER DELETE ON Emp_tab FOR EACH ROW
DELCARE n INTEGER;
BEGIN SELECT COUNT(*) INTO n FROM Emp_tab;
DMBS_OUTPUT.PUT_LINE(' There are now ' || a || ' employees,');
END;
```

This trigger results in an error after this SQL statement is entered:

```
DELETE FROM Emp_tab WHERE Empno = 7499;
```

How do you correct the error?

- A. Change the trigger type to a BEFORE DELETE .

- B. Take out the COUNT function because it is not allowed in a trigger.
- C. Remove the DBMS_OUTPUT statement because it is not allowed in a trigger.
- D. Change the trigger to a statement-level trigger by removing FOR EACH ROW. **Ans: d**

54. What is true about stored procedures?

- A. A stored procedure uses the DECLARE keyword in the procedure specification to declare formal parameters.
- B. A stored procedure is named PL/SQL block with at least one parameter declaration in the procedure specification.
- C. A stored procedure must have at least one executable statement in the procedure body.
- D. A stored procedure uses the DECLARE keyword in the procedure body to declare formal parameters. **Ans: c**

55. Examine this code:

```

CREATE OR REPLACE PROCEDURE insert_dept (p_location_id NUMBER) IS v_dept_id NUMBER(4);
BEGIN
  INSERT INTO departments VALUES (5, 'Education', 150, p_location_id);
  SELECT department_id INTO v_dept_id FROM employees WHERE employee_id=99999; END
insert_dept; /
CREATE OR REPLACE PROCEDURE insert_location ( p_location_id NUMBER, p_city VARCHAR2) IS
BEGIN
  INSERT INTO locations(location_id, city) VALUES (p_location_id, p_city);
  insert_dept(p_location_id);
END insert_location; /

```

You just created the departments, the locations, and the employees table. You did not insert any rows. Next you created both procedures. You now invoke the insert_location procedure using the following command:

EXECUTE insert_location (19, 'San Francisco')

What is the result in this EXECUTE command?

- A. The locations, departments, and employees tables are empty.
- B. The departments table has one row. The locations and the employees tables are empty.
- C. The location table has one row. The departments and the employees tables are empty.
- D. The locations table and the departments table both have one row. The employees table is empty. **Ans: a**

56. The creation of which four database objects will cause a DDL trigger to fire? (Choose four)

- A. Index
- B. Cluster
- C. Package
- D. Function
- E. Synonyms
- F. Dimensions
- G. Database links **Ans: abcd**

57. Which two program declarations are correct for a stored program unit? (Choose two)

- A. CREATE OR REPLACE FUNCTION tax_amt (p_id NUMBER) RETURN NUMBER
- B. CREATE OR REPLACE PROCEDURE tax_amt (p_id NUMBER) RETURN NUMBER
- C. CREATE OR REPLACE PROCEDURE tax_amt (p_id NUMBER, p_amount OUT NUMBER)
- D. CREATE OR REPLACE FUNCTION tax_amt (p_id NUMBER) RETURN NUMBER(10,2)
- E. CREATE OR REPLACE PROCEDURE tax_amt (p_id NUMBER, p_amount OUT NUMBER(10, 2)) **Ans: ac**

58. You need to implement a virtual private database (vpd). In order to have the vpd functionality, a trigger is required to fire when every user initiates a session in the database. What type of trigger needs to be created?

- a. DML trigger
- b. System event trigger
- c. INSTEAD OF trigger
- d. Application trigger **Ans: b**

59. You have a row level BEFORE UPDATE trigger on the EMP table. This trigger contains a SELECT statement on the EMP table to ensure that the new salary value falls within the minimum and maximum salary for a given job title. What happens when you try to update a salary value in the EMP table?

- A. The trigger fires successfully.
- B. The trigger fails because it needs to be a row level AFTER UPDATE trigger.
- C. The trigger fails because a SELECT statement on the table being updated is not allowed.
- D. The trigger fails because you cannot use the minimum and maximum functions in a BEFORE UPDATE trigger. **Ans: c**

60. Examine this code:

```
CREATE OR REPLACE STORED FUNCTION get_sal (p_raise_amt NUMBER, p_employee_id employees.employee_id%TYPE) RETURN NUMBER;
IS v_salary NUMBER; v_raise NUMBER(8,2); BEGIN SELECT salary INTO v_salary FROM employees
WHERE employee_id = p_employee_id; v_raise := p_raise_amt * v_salary; RETURN v_raise; END;
```

Which statement is true?

- A. This statement creates a stored procedure named get_sal.
- B. This statement returns a raise amount based on an employee id.
- C. This statement creates a stored function named get_sal with a status of invalid.
- D. This statement creates a stored function named get_sal.
- E. This statement fails. **Ans: e**

61. You need to disable all triggers on the EMPLOYEES table. Which command accomplishes this?

- A. None of these commands; you cannot disable multiple triggers on a table in one command.
- B. ALTER TRIGGERS ON TABLE employees DISABLE;
- C. ALTER employees DISABLE ALL TRIGGERS;
- D. ALTER TABLE employees DISABLE ALL TRIGGERS; **Ans: d**

62. An internal LOB is _____.

- A. A table.
- B. A column that is a primary key.
- C. Stored in the database.
- D. A file stored outside of the database, with an internal pointer to it from a database column. **Ans: c**

63. Examine this code:

```
CREATE OR REPLACE FUNCTION calc_sal (p_salary NUMBER) RETURN NUMBER IS v_raise
NUMBER(4,2) DEFAULT 1.08;
BEGIN
  RETURN v_raise * p_salary;
END calc_sal; /
```

Which statement accurately call the stored function CALC_SAL ? (Choose two)

- A. UPDATE employees (calc_sal(salary)) SET salary = salary * calc_sal(salary);

- B. INSERT calc_sal(salary) NOT employees WHERE department_id = 60;
- C. DELETE FROM employees(calc_sal(salary)) WHERE calc_sal(salary) > 1000;
- D. SELECT salary, calc_sal(salary) FROM employees WHERE department_id = 60;
- E. SELECT last_name, salary, calc_sal(salary) FROM employees ORDER BY calc_sal(salary); **Ans: de**

64. This statement fails when executed:

```
CREATE OR REPLACE TRIGGER CALC_TEAM_AVG AFTER INSERT ON PLAYER BEGIN
INSERT INTO PLAYER_BATSTAT (PLAYER_ID, SEASON_YEAR, AT_BATS, HITS) VALUES (:NEW.ID,
1997, 0, 0); END;
```

To which type must you convert the trigger to correct the error?

- A. Row.
- B. Statement
- C. ORACLE FORM trigger
- D. Before **Ans: a**

65. Examine this code:

```
CREATE OR REPLACE PROCEDURE audit_emp (p_id IN emp.empno%TYPE) IS
v_id NUMBER;
    PROCEDURE log_exec IS BEGIN INSERT INTO log_table (user_id, log_delete) VALUES (USER,
SYSDATE); END log_exec;
    v_name VARCHAR2(20);
BEGIN
    DELETE FROM emp WHERE empno = p_id; log_exec; SELECT ename, empno INTO v_name, v_id
FROM emp WHERE empno = p_id; END audit_temp;
```

Why does this code cause an error when compiled?

- A. An statement is not allowed in a subprogram declaration.
- B. Procedure LOG_EXEC should be declared before any identifiers.
- C. Variable v_name should be declared before declaring the LOG_EXEC procedure.
- D. The LOG_EXEC procedure should be invoked as EXECUTE log_exec with the AUDIT_EMP procedure. **Ans: c**

66. Examine this code:

```
CREATE OR REPLACE PACKAGE metric_converter IS
c_height CONSTRAINT NUMBER := 2.54;
c_weight CONSTRAINT NUMBER := .454;
FUNCTION calc_height (p_height_in_inches NUMBER)
RETURN NUMBER;
FUNCTION calc_weight (p_weight_in_pounds NUMBER)
RETURN NUMBER;
END;
CREATE OR REPLACE PACKAGE BODY metric_converter IS
FUNCTION calc_height (p_height_in_inches NUMBER)
RETURN NUMBER IS
BEGIN
    RETURN p_height_in_inches * c_height;
END calc_height;
FUNCTION calc_weight (p_weight_in_pounds NUMBER)
RETURN NUMBER IS
BEGIN
    RETURN p_weight_in_pounds * c_weight;
END calc_weight;
END metric_converter;
CREATE OR REPLACE FUNCTION calc_height (p_height_in_inches NUMBER)
RETURN NUMBER IS
BEGIN
    RETURN p_height_in_inches * metric_converter.c_height;
END calc_height;
```

Which statement is true?

- a. If you remove the package specification, then the package body and the stand alone stored function CALC_HEIGHT are removed.
- b. If you remove the package body, then the package specification and the stand alone stored function CALC_HEIGHT are removed.
- c. If you remove the package specification, then the package body is removed.
- d. If you remove the package body, then the package specification is removed.

- e. If you remove the stand alone stored function CALC_HEIGHT, then the METRIC_CONVERTER package body and the package specification are removed.
- f. The stand alone function CALC_HEIGHT cannot be created because its name is used in a packaged function. **Ans: c**

67. What is a condition predicate in a DML trigger?

- A conditional predicate allows you to specify a WHEN-LOGGING-ON condition in the trigger body.
- B. A conditional predicate means you use the NEW and OLD qualifiers in the trigger body as a condition.
- C. A conditional predicate allows you to combine several DBM triggering events into one in the trigger body.
- D. A conditional predicate allows you to specify a SHUTDOWN or STARTUP condition in trigger body. **Ans: c**

68. Examine this package specification:

```
CREATE OR REPLACE PACKAGE concat_all IS v_string VARCHAR2(100);
PROCEDURE combine (p_num_val NUMBER);
PROCEDURE combine (p_date_val DATE);
PROCEDURE combine (p_char_val VARCHAR2, p_num_val NUMBER);
END concat_all; /
```

Which overloaded COMBINE procedure declaration can be added to this package specification?

- a. PROCEDURE combine;
- b. PROCEDURE combine (p_no NUMBER);
- c. PROCEDURE combine (p_val_1 VARCHAR2, p_val_2 NUMBER);
- d. PROCEDURE concat_all (p_num_val VARCHAR2, p_char_val NUMBER); **Ans: a**

- 69. Local procedure A calls remote procedure B. Procedure B was compiled at 8 A.M. Procedure A was modified and recompiled at 9 A.M. Remote procedure B was later modified and recompiled at 11 A.M.**

The dependency mode is set to TIMESTAMP. What happens when procedure A is invoked at 1 P.M?

- A. There is no affect on procedure A and it runs successfully.
- B. Procedure B is invalidated and recompiles when invoked.
- C. Procedure A is invalidated and recompiles for the first time it is invoked.
- D. Procedure A is invalidated and recompiles for the second time it is invoked. **Answer**

70. Under which two circumstances do you design database triggers? (Choose two)

- E. To duplicate the functionality of other triggers.
- F. To replicate built-in constraints in the Oracle server such as primary key and foreign key.
- G. To guarantee that when a specific operation is performed, related actions are performed.
- H. For centralized, global operations that should be fired for the triggering statement, regardless of which user or application issues the statement. **Ans: cd**

71. Examine this procedure

```
CREATE OR REPLACE PROCEDURE DELETE_PLAYER (V_ID IN NUMBER) IS BEGIN DELETE FROM
PLAYER WHERE ID = V_ID EXCEPTION
WHEN STATS_EXISTS_EXCEPTION THEN
DBMS_OUTPUT.PUT_LINE ('Cannot delete this player, child records exist in PLAYER_BAT_STAT table');
END;
```

What prevents this procedure from being created successfully?

- A. A comma has been left after the STATS_EXIST_EXCEPTION exception.
- B. The STATS_EXIST_EXCEPTION has not been declared as a number.
- C. The STATS_EXIST_EXCEPTION has not been declared as an exception.
- D. Only predefined exceptions are allowed in the EXCEPTION section. **Ans: c**

1. Consider the following code snippet:

```
declare
n1 number:=20;
n2 number:=10;
n3 number;
begin
n1:=30;
n2:=n1;
n3:=n1*n2;
dbms_output.put_line(n1);
dbms_output.put_line(n2);
dbms_output.put_line(n3);
end;
```

What will be the output?

- a. 10 10 20 b. 20 20 40 **c. 30 30 900** d. 30 10 300

2. Consider the following if statement

```
if v_value>100 then
v_new_value:=2*v_value;
elsif v_value>200 then
v_new_value:=3*v_value;
elsif v_value>300 then
v_new_value:=4*v_value;
else
v_new_value:=5*v_value;
end if;
```

what value would be assigned to v_new_value if v_value=250?

- a. 500** b. 250 c. 1000 d.750

3. declare

```
j number;
begin
for index in 1..6 loop
j:=index;
goto outer;
end loop;
<<outer>>
j:=j+1;
end;
```

What is the value of j after the above code snippet is executed?

- a. 1 b. 6 **c. 2** d. 7 e. null

4. Examine the following code:

```
declare
j number:=100;
result number:=0;
begin
```

```
--other statements here
begin
result:=j/0;
Exception
when others then
dbms_output.put_line('when others handler');
end;
exception
when zero_divide then
dbms_output.put_line('zero divide handler');
end;
```

Which of the following gives the output of above code?

a. when others handler is displayed

- b. program issues error
- c. no message is displayed
- d. zero divide handler is displayed
- e. when others handler is displayed after which zero divide handler is displayed

5. Consider the following statements that describe a PL/SQL block structure:

Statement 1: Declare and Exception sections are mandatory.

Statement 2: Begin and End sections are optional.

a. only statement 2 is true.

- b. only statement 1 is true.
- c. both are true. d. both are false.

6. What does the method extend (n,i) do?

- a. adds i elements at nth position.
- b. adds a single null element at ith position.
- c. appends n elements and sets each to the same value as the ith element.**
- d. add n null elements at the ith position
- e. appends n elements and sets each to a null value.

7. Given the following declaration:

declare

```
type emp_salinfo_t is record(e_no emp.empno%type, sal emp.sal%type,comm
emp.comm%type);
emp_sal_rec emp_salinfo_t;
```

which of the following is the correct declaration to create a collection of the above record type?

- a. type emp_sal_col_t is table of emp_sal_rec;**
- b. emp_sal_col_t is table of emp_salinfo_t;
- c. type emp_sal_col_t is of table of emp_salinfo_t;
- d. type emp_sal_col_t is table of record emp_salinfo_t;
- e.emp_sal_col_t is table of emp_sal_rec;

8. Which of the following statement can be executed in PL/SQL?

- a. commit b. update c. delete d. select e. all of the listed options.

9. In SQL, varchar data type can store

- a. only numbers

b. both characters and number

c. only decimal values

d. only characters.

10. In SQL, a NULL value can be entered into the table by

a. specifying /0 b. specifying a blank space c. **using NULL keyword** d. specifying 0.

11. create or replace function cal_stud_marks_avg(v_id in Student.student_id%type)
return number

is

v_avg number;

begin

select (sum)marks/count(subject_id) into v_avg from student where student-id=v_id;

return (v_avg);

end;

Which of the following gives the correct way of calling the above function for student id 150?

a. execute cal_stud_marks_avg(150)

b. exec cal_stud_marks_avg(150)

c. **select cal_stud_marks_avg(150) from student**

d. run cal_stud_marks_avg(150)

12. The correct syntax for defining user defined exception, called myExp is

a. Exception myExp;

b. TYPE Exception myExp;

c. create exception myExp;

d. nyExp new Exception;

e. **myExp Exception;**

13. Consider the following code snippet:

declare

flag boolean;

begin

if (not flag) then

dbms_output.put_line('hai');

else

dbms_output.put_line('hello');

end if;

end;

Which of the following gives the outcome of the above code?

a. prints hello.

b. prints null

c. gives error.

d. prints hai

14. In PL/SQL, the unnamed blocks that are embedded within an application having no header section are defined as

a. **anonymous block.** b. procedure c. named blocks. d. function e. trigger

15. Consider the following cursor definition based on the products table (product_id,name,price)

```

declare
v_product_id products.product_id%type;
v_name products.name%type;
v_price products.price%type;
cursor v_product_cursor is select product_id,name,price from products order by product_id;
begin
open v_product_cursor;
loop
fetch v_product_cursor into v_product_id,v_name,v_price;
dbms_output.put_line(v_product_id || v_name || v_price);
end loop;
close v_product_cursor;
end;

```

The above code segment when executed results in infinite loop. Which of the following code, if added, will execute the code segment properly?

a. exit when v_product_cursor%notfound

b.exit when v_product_cursor%isopen;

c. exit when others;

d.exit when v_product_cursor%rowcount;

16. In PL/SQL, boolean data type is an example of

a. LOB data type

b. composite data type

c. scalar data type

d. reference data type

17. A relational database uses two-dimensional tables to store information

a. true b. false

18. declare

count number:=0;

begin

for j in reverse 3..1 loop

count:=j;

end loop;

end;

What is the value of count after the loop executes?

a. 1 **b. 0** c. Null d. 2 e. 3

19. declare

ctr number:=0;

begin

ctr:=10;

for ctr in reverse 1..10

loop

dbms_output.put_line(ctr);

end loop;

end;

Which of the following output will be displayed if the above code is executed?

- a. prints 0 to 10
- b. prints 10 to 0
- c. prints 1 to 10
- d. prints 10 to 1**

20. In PL/SQL, a function can return a record.

- a. **true**
- b. false.

21. Which of the following will allow you to run the loop at least once?

- a. numeric for loop
- b. while loop
- c. do-while-loop
- d. simple loop**
- e. cursor for loop

22. Consider the following cursor declaration based on emp table (empno,ename,deptno,job)
declare

```
cursor emp_cursor(p_deptno number,p_job varchar2)
is
select empno,ename from emp where deptno=p_deptno;
begin
statements
---
```

which of the following statement opens the cursor successfully?

- a. open emp_cursor(10,'analyst');**
- b. open emp_cursor;
- c. open emp_cursor(10);
- d.open emp_cursor('analyst');

23. Which of the following events can cause a trigger to fire? (choose 3)

- a. An user executes a create or alter table statement.
- b. a database is shut down or started up.
- c. A user executes a query that joins multiple tables
- d. An user executes a select statement
- e. An user logs on or logs off the database.

24. Given the header of a procedure,

```
create or replace procedure account_transaction
```

is

begin

--statements

end;

Which command will execute the procedure from sql plus prompt?

- a. account_transaction
- b. execute account_transaction**
- c. start account_transaction
- d. run account_transaction

25. Following section in the declaration section throws error:

PI constant number;

- a. There is not enough memory in the program for this constant
- b. There is no datatype associated with the constant
- c. PI is a reserved word
- d. There is no value associated with the constant.

26. What is done in the following PL/SQL code?

```
INVALID_DATE EXCEPTION;
```

- a. An exception is raised
- b. An exception is associated
- c. An exception is declared**

- d. an exception is handled
27. Under which situation do you create a server side procedure?
- When the procedure needs to be used by many client applications accessing several remote databases.
 - When the procedure contains no SQL statements
 - When the procedure contains no PL/SQL commands
 - When the procedure needs to be used by many users accessing the same schema objects on a local database.
28. You are developing PL/SQL process flow into your program. The command use to open a cursor for loop is which of the following keywords?
- parse
 - open
 - fetch
 - none of these**
29. Which two statements about the overloading feature of packages are true? (Choose two)
- Overloading allows different functions with the same name that differ only in their return types.
 - Only local or packaged sub programs can be overloaded**
 - Overloading allows different subprograms with the same name and same number or type of the parameters.
 - Overloading allows different subprograms with the same name but different number or type or order of the parameters.**
30. Which part of a database trigger determines the number of times the trigger body executes?
- Trigger body
 - Trigger timing
 - Trigger type**
 - Trigger event

1) Declare

```
TYPE CITY
is TABLE of
VARCHAR2(20) index by PLS_INTEGER;
v1.CITY;
begin
v1(1):='Bangalore';
v1(6):='Mumbai';
v1(2):='Delhi';
dbms_output.put_line(v1.c ount());
dbms_output.put_line(v1(2));
dbms_output.put_line(v1.next(6));
end;
```

2)Declare

```
Num number;
num:=12;
Declare
Num number(line1);
num:=13;
begin
loop
```

```
dbms.output.put_line(num+ +);
end loop;
end;
if num
  dbms.output.put_line("less");
end if;
end;
```

3) select decode(Ecode,5,'Department',4,'Marketing',3,'Marketing','Sales') as result from Employee;

4) Which of the following statements about packages are true?(choose 3)

- a) The specification is required, but the body is optional.
- b) To reference a packaged procedure, you have to use the <packagename>.<procedurename> notation
- c) The specification and body are stored separately in the database
- d) Both the specification and body are required components of a package.
- e) The specification and the body are stored together in the database.

ans:a,b,c

5) select * from emp where empno=100;

Oracle opens an explicit cursor for the above statement. State true or false.

ans:F

6) In the following declaration, birthday_list is of type NESTED TABLE. State true or false.

TYPE birthday_list is TABLE of date INDEX BY POSITIVE;

ans: True

7) In PL/SQL, the unnamed blocks that are embedded within an application having no header section are defined as

- a) Anonymous blocks
- b) Procedure
- c) Trigger
- d) Function
- e) Named blocks

8) If the condition evaluates to NULL in a while loop, the loop body is executed.

- a) True b) False

9) Consider the following declarations in PL/SQL.

declare

n1,n2 number; --1

n3 number NOT NULL;--2

n4 number:=12345; --3

```
n5:=12345; --4  
c1 char(1):='Y'; --5
```

Which of the following gives the lines that have the right declaration?

- a) 3 and 5
- b) 4 and 5
- c) 1 and 3 -ans
- d) 2 and 3
- e) 1 and 2

10) declare

```
TYPE list_of_names_t is TABLE OF emp.ename%type INDEX BY BINARY_INTEGER;  
family_mem list_of_names_t;  
l_row BINARY_INTEGER;  
begin  
family_mem(100):='Veena';  
family_mem(-15):='Sheela';  
family_mem(-30):='Syed';  
family_mem(88):='Raji';  
l_row:=family_mem.FIRST;  
while(l_row is not null)  
loop  
dbms_output.put_line(family_mem(l_row));  
l_row:=family_mem.NEXT(l_row);  
end loop;  
end;
```

- a) Syed, Sheela, Raji, Veena -ans
- b) No output
- c) Veena, Syed, Sheela, Raji
- d) Raji, Syed, Sheela, Veena
- e) Sheela, Syed, Veena, Raji

11) In a banking application, which of the following data types can be used for interest rates with varying and unpredictable decimal places such as 1234,3.4,5 and 123?

- a) long
- b) number(p,s)
- c) double
- d) number

12) which of the following functions are used to obtain additional information about an error?(choose 2)

- a) sqlcode
- b) sqlerrm
- c) raise_application_error
- d) pragma_exception_init

13) Which of the following statement can be executed in PL/SQL?

- a)delete
- b)select
- c)update
- d)commit
- e)all of the above-ans

14) Which of the following loop construct will allow you to run the loop atleast once?

- a)Numeric for loop
- b)while-loop
- c)do-while loop
- d)loop -ans
- d)cursor for loop

15) Which of the following statement can be used to turn off Query Flashback option in Oracle 10g?

- a)EXECUTE DBMS_FLASHBACK.Remove
- b)EXECUTE DBMS_FLASHBACK.Disable-ans
- c)EXECUTE DBMS_FLASHBACK.End
- d)EXECUTE DBMS_FLASHBACK.Turnoff

16) The maximum length of varchar2 data type in Oracle 10g is

- a)1000 bytes
- b)6000 bytes
- c)4000 bytes
- d)2000 bytes -ans

17) Which of the following events can cause a trigger to fire?(choose 3).

- a) An user executes a CREATE or ALTER table statement.
- b) A database is shut down or started up
- c)A user executes a query that joins multiple tables
- d)An user executes a SELECT statement
- e) An user logs on or logs off the database.

a,b,e

18) Assume there is a table called emp_log with the columns,empno,ename,trans_date.

Assume there is a trigger on emp table as follows:

```
create or replace trigger emp_trig after insert on emp for each row
declare
pragma autonomous_transaction;
begin
insert into emp_log(empno,ename,trans_date) values
(:new.empno,:new.ename,sysdate);
```

```
commit;  
end;
```

What would be the result after executing the following procedure:

```
create or replace procedure ins_rec as  
begin  
insert into emp(empno,ename) values (1,'aa');  
commit;  
insert into emp(empno,ename) values(2,'bb');  
rollback;  
end;
```

- a)1 record inserted in both emp and emp_log tables with empno 1 -ans
 - b)2 records inserted into emp table-empno1 and empno 2
 - c)everything is rolled back
 - d)2 records inserted in both emp and emp_log tables
 - e)1 record is inserted into emp table -empno 1
- 2 records are inserted into emp_log table - empno 1 and empno 2

In oracle database, number data type allows both number and decimal digits.

- a) True b) False

In PL/SQL function can return a record

- a) True b) False

Which of the following statements correctly describe functions that are available in SQL(choose 3).

- a) Truncate rounds the column,expression,or value to n decimal places
 - b)Decode translates an expression after comparing it to each search value.
 - c)INSTR returns the numeric position of a named character.
 - d)NVL compares two expressions and returns null if they are equal, or the first expression if they are not.
 - e)TRIM trims the leading or trailing characters(or both) from a character string.
- b,c,e

The SQL query that builds the view is stored in the data dictionary and need not be reassemble every time the view is used.

- a) true-ans b)false

Which of the following is declaration for retrieving multiple employees of a specific department?assume department number is not known. The rows retrieved must be stored in a construct that will allow for row by row processing at a later stage.

- a)cursor emp_cur is select * from emp;
- b)cursor emp_cur(p_deptno varchar2) is select * from emp where deptno=p_deptno

- c)select * from department
- d)cursor emp_cur is select * from emp where deptno=p_deptno;
- e)select * from emp;

which of the following will allow you to pass a value from a procedure to the calling environment?

- a)varchar2
- b)return
- c)in
- d>out
- e)number

Which of the following can be defined as a collection of the most recently used definitions in the database?

- a)data dictionary cache
- b)redo log buffer cache
- c)library cache
- d)database buffer cache-ans

Which of the following are main types of files used in the physical architecture of an Oracle database?

- a)trace files
- b)redo log files
- c)control files
- d)parameter files
- d)data files
- b,c,e

Statement A:Listener process scans for connection requests to an Oracle Instance

Statement B:Listener process links up a dispatcher process to user process.

which of the following is true?

Both Statements A and B are true

Statement A is true and Statement B is false

Statement A is false and Statement B is true

Both Statements A and B are false

Both Statements A and B are true

Examine the PL/SQL Block

declare

```

result Number;
num1 Number:=&number1;
num2 Number:=&number2;
begin
select num1/num2 into result from dual;
dbms_output.put_line('Result is'||result);
exception
when ZEROS_DIVIDE then
dbms_output.put_line('Exception Occurred Divide by Zero');
end;
/

```

What will be the output of the above code when user passes the value for NUM1 as 0 and NUM2 as 10

Exception Occurred Divide by Zero
Result is 10
Result is 0
Compilation fails
ans:-- Result is 0
Compiles successfully and terminates the program due to an Exception

```

DECLARE
A BOOLEAN := NULL;
B BOOLEAN;
BEGIN
B := A IS NULL;
IF B IS NULL THEN
DBMS_OUTPUT.PUT_LINE('HI');
END IF;
END

```

ans -false

Carefully read the question and answer accordingly.
Which trigger is used to display the salary change information whenever a DELETE , INSERT or UPDATE statement affects the tbl_employees table(Salary Column)- The trigger should be FOR EACH ROW trigger.
SAMPLE OUTPUT : Fahan,Karn: Old salary = 3500, New

salary = 4500, Difference: 1000

```
CREATE OR REPLACE TRIGGER salary_changes BEFORE DE  
LETE OR INSERT OR UPDATE ON TBL_EMPLOYEES FOR EACH  
ROW DECLARE sal_diff NUMBER; BEGIN sal_diff := :N  
EW.salary - :OLD.salary;
```

Dovetail ques consolidated:

1. Salary greater than 5000

Ans: where sal>5000

10. DML statement

Ans: UPDATE

11. Order by

- a) Default in ascending order
- b) Last in select statement

Ans: Both are true, if radio button then chose 'a'

12. We can have multiple document & universe domain in repository?

Ans: True

13. for LOOP-

Ans: EXIT condition is not required

14. When pmcmd is incorrect what it will return:

Ans: Non Zero return code

15. Transformation to move top 3 records

Ans: Rank Transformation

22. Foreign key query: (**checked**)

Ans:

```
CREATE TABLE EMP  
(empno NUMBER(4),  
ename VARCHAR2(35),  
deptno NUMBER(7,2)  
CONSTRAINT emp_deptno_fk REFERENCES dept (deptno);
```

25. View is created with

- a. by joining two tables
- b. some restriction on rows in one table

Ans: by joining two tables

26. What is output of full outer join?

- a. only matched columns in both tables
- b. unmatched columns in both tables
- c. matched columns in table in one table unmatched in other table

Ans: unmatched columns in both tables

27. Result for round(trunc(mod(1600,10),-2))

- a. 0
- b. error
- c. 1
- d. none

Ans: 0

29. Which of the following is true about group by clause?

- a. It has to use before having clause
- b. It has to use after having clause
- c. Last in select statement
- d. none

Ans: It has to use before having clause

Oracle

33. What is true about Cursor for loop?

- a. Does not require a exit condition
- b. Open Cursor
- c. Close Cursor
- d. Fetch Cursor

Ans: Does not require a exit condition

34. Not true abt Cursor For Loop?

- a. Open Cursor
- b. Close Cursor
- c. Fetch Cursor
- d. All the Above

Ans: All the above

33. Calculate 12*sal*commission for every record of the emp table chaoose the appropriate query?

- a) Select 12*sal*nvl(commission,0) from emp ;
- b) Select 12*sal*decode(commission,0) from emp ;

Ans: a

1. no of modes in pmcmd?

Ans: 2 (Commandline & Interactive Mode)

2. No. of designer tools in informatica designer?

Ans: 5

3. No. of tools in w.f. manager?

Ans: 3

4. Top down approach which is built first?

Ans: EDW & then Data mart

5. Bottom Up approach

Ans: Data Mart and then EDW

6. Oracle: If you delete one constraint, what will happen?

Ans: It will affects child also

7. Reusable tasks were developed in

Ans: Task developer

8. Oracle: round(trun(34.67,2),3)

Ans: Value is 34.670... similar type.

21. To display the string 'Hello World' as ello world...

- a. select lower(trim('H' from 'Hello World')) from dual
- b. select lower(substr('Hello World',1,1) from dual
- c. select lower(substr('Hello World',2,1) from dual
- d. None of the listed options

Ans: select lower(trim('H' from 'Hello World')) from dual

26. which line in the option is incorrect

- a. from action_table;
- b. select name, action, item
- c. Cursor action_cursor is
- d. **Into action_records**
- e. There is no errors in the statement

Ans: into action_records

Because if we want to use 'into' command then first we have to use OPEN n FETCH command during cursor usage and there is no fetch command over here. The correct syntax follows like this

"Cursor action_cursor Is Select name, action, item From action_table"

1. Which is not cursor attributes?

- a. %ROWCOUNT
- b. %ISOPEN
- c. %NOTFOUND
- d. %FOUND
- e. %TOO_MANY_ROWS

Ans: %TOO_MANY_ROWS

NOTE: too_many_rows is an EXCEPTION

18. How to over ride the order by clause in look up query?

- a. Write an order by clause in look up query and add – (double hiphen) after order by clause
- b. Write an order by clause in look up query over ride order by
- c. Write an over ride query.
- d. Something similar

Ans: Write an order by clause in look up query over ride and add – (double hiphen) after order by clause

20. Select (to_char(nvl(sqrt(43562)," invalid")) from DUAL is valid statement or not?

Ans: Invalid Parenthesis not closed properly, contain "invalid" statement in the sql.

9. Which statement in oracle will cause implicit COMMIT?

Ans: Only DDL commands cause implicit commit.

DDL commands – Create, Alter, Rename, Truncate. etc

10. Create table dept(name varchar2(20), no number);
rollback;
describe dept;

What happens when u describe table dept?

Ans: Structure of table dept will appear because create is a DDL command so it get automatically committed.

26. A User desires to join two relational sources with a single Source Qualifier have been correctly imported into the mapping. Next, she must establish a primary/foreign key relationship between two ports in the source definitions. Assuming that the relationship does not currently exist on the database server, what are the required steps to do this?

- a. Use the source Analyser to establish that primary/foreign key relationship and save the repository
- b. Use the source Analyser to establish that primary/foreign key relationship, save the repository, and run the appropriate SQL statement on the database tables to create the actual primary / foreign key relationship
- c. **Use the source Analyser to establish that primary/foreign key relationship, open the SQL Editor in the Source qualifier and generate the join statement, modify the join statement if required and save the repository.**
- d. Use the source analyzer to establish that primary /foreign key relationship , regenerate the source Qualifier SQL using the SQL editor, press the okay button and save the repository.

Ans: c

11. Which of the following is true?

- a. single row query will fetch the rows from only one table
- b. subqueries can have aggregate function
- c. subqueries can have analytical function
- d. multiple sub query can have like function

Ans: single row will fetch the rows from only one table

12. Which of the following is possible to a date variable

- a. min
- b. sum
- c. avg
- d. none

Ans: min

13. What are implicit cursors?

- a. Implicit Cursor is not used in Oracle
- b. User should be very careful in using Implicit Cursor
- c. **Implicit cursor used for SQL statements that are not named**
- d. None
- e.

14. In which line will an error occur?

- a. create function xyz (var_a in number,
- b. return(var_a + var_b)
- c. end;
- d. var_b in out number) return number

Ans: `create function xyz (var_a in number,`

Oracle:

1. Which of the following one is true about order by clause?

- a. By default it sorts in ascending order.
- b. It comes at the last of select clause.
- c. and d. options are irrelevant.

Ans: 'a'.. For 'b', in a select clause order by can come inside a sub query also.

3. Using which command it is possible to append data from one table to another which has same column data type but with different data. Existing data should be updated and new data should be inserted.

- a. Merge
- b. Not possible with single command.
- c. None of the options
- d. INSERT

Ans: Merge

1. Which privilege is an object privilege?

- a. index
- b. drop user
- c. create session
- d. back up any table

Ans: index

Create index, Create synonym is a SYSTEM privileges.

Delete, Index, Execute, Flush, Insert, Load, References, Refresh, Select, Unload, Update---
these are the OBJECT privileges. If any of these comes, tick those..

6. What does the following query does?

`select a.mid, b.maxsal from emp a, (select mid, max(sal) maxsal from emp group by deptname) b where a.mid=b.mid ;`

a. maximum salaried emp within each dept.

- b. maximum salaried emp within company.
- c. Error in statement.
- d. I don't remember.

Actual question: WHAT I GOT IN DOVETAIL

1. `SELECT a.emp_name, a.sal, a.dept_id, b.maxsal`
2. `FROM employees a,`
3. `(SELECT dept_id, MAX(sal) maxsal`
4. `FROM employees`
5. `GROUP BY dept_id) b`
6. `WHERE a.dept_id = b.dept_id`
7. `AND a.sal < b.maxsal;`

What is the result of the statement?

- A. The statement produces an error at line 1.
- B. The statement produces an error at line 3.
- C. The statement produces an error at line 6.
- D. The statement returns the employee name, salary, department ID, and maximum salary earned in the department of the employee for all departments that pay less salary than the maximum salary paid in the company.
- E. The statement returns the employee name, salary, department ID, and maximum salary earned in the department of the employee for all employees who earn less than the maximum salary in their department.**

7. Which one is a different oracle constraint?

- a. Not Null
- b. Unique
- c. Primary Key
- d. None of the above

Ans: NOTA. If they meant 'not present' for 'different' word. Because all CONSTRAINT are there in oracle..

4. State the Correct one

- a. Group by should be placed before having clause
- b. Group by should be placed after having clause
- c. oderby should be placed before select clause

Ans: Group by should be placed before having clause

5. Find out error in following query

```
select o.productname, p.ordername, o.location
```

where

```
product.product_id = order.order_id
```

and

```
order by o.sale
```

- a. table alias should be used in where clause
- b. o.sale should be used in select clause
- c. alias in oder must be changed

Ans: table alias should be used in where clause

6. What will be situation when raise_ exception is raised and exception handler block is not provided

- a. it will cause a memory leak
- b. it will raise % not found error
- c. the control will pass through sql block

Ans: It will cause a memory leak

Oracle-8q

1. There are two tables. You need to copy one table to another. Incase the row already exists, update the values else insert the row. How will you do this?

- a) Merge command

- b) Update statement
- c) Not possible
- d) Insert

Ans: Merge Command

2. To work with a view what should you have?

Ans: Select privilege on the view

3. OTHERS exception is for handling an exception by name "OTHERS". True or false.

Ans: False

4. Which of the following are necessary for Cursor FOR loop?

- a) Open cursor
- b) Parse cursor
- c) Fetch cursor
- d) None of the above

Ans: None of the above

10) Which one this is a valid table name?

- a. COLUMN
- b. NUMBER#1
- c. 1TABLE
- d. TABLE_MANY_COLUMN_ (goes on like that up to 34 characters.)

Ans: NUMBER#1

21) To which of the following does the oracle create index implicitly? **(checked)**

- a. Primary key
- b. Unique
- c. Not Null
- d. Check

22) Which is the return value for pmcmd when the value is true.

- a) 1
- b) 0
- c) 5
- d) 6
- e) 4

Ans: 0

23) Which of the following are true?

- a. Single-row functions can be used only in select and where clause.
- b. character functions works on character data type column only.
- c. date functions return date data type only.
- d. Conversion function will convert the datatype of column from one datatype to another datatype
- e. None of the above.

Ans: Conversion function will convert the datatype of column from one datatype to another datatype.

2. Calculate 12*sal*commission for every record of the emp table choose the appropriate query?

- a) **Select 12*sal*nvl(commission,0) from emp ;**
- b) Select 12*sal*decode(commission,0) from emp ;

- c) Select 12*sal*nvl(commission) from emp;
 - c) Select 12*sal*commission from emp ;
2. If left out, which of the following will cause infinite execution of LOOP
- a. exit
 - b. end
 - c. begin

20) Which will do automatic commit

- a) select
- b) update
- c) none

(DDL statements only have auto commit so create, drop, alter, rename, truncate)
If any DDL command is there, choose that option.

1. What is stored in the .key file?

- a. repository server name
- b. user name
- c. password**
- d. all the above

30. To search for the employees whose last name has 'a' as its second letter.

Four queries were given.

The correct condition is: **last_name like “_a%”**

31. To use the derived datatype for a column :

Four queries were given.

The correct condition is: **emp.lname%type**

1. *Which is not a correct representation of 1 to many relation in framework manager?*
 - a) 1:0 to 1:n
 - b) 1:0 to 0:n
 - c) 1:1 to 0:n**
 - d) 0:1 to 0:n

1. You granted user Joe the INDEX and REFERENCES privileges on the INVENTORY table. Which statement did you use?

**GRANT ALL
ON inventory
TO joe;**

**GRANT ANY PRIVILEGE
ON inventory
TO joe;**

**GRANT INDEX AND REFERENCES
ON inventory
TO joe;**

GRANT ALL WITH GRANT OPTION

ON inventory
TO joe;

Ans: GRANT ALL
ON inventory
TO joe;

13. What is the use of grant clause?(**checked**)

- a. **Grant privileges on objects to users and groups**
- b. Grant privileges to objects.
- c. allows the grantee to give privileges to groups and users

3. select emp_name, salary, desig
from emp
order by salary desc;

```
select emp_name, salary, desig  
from emp  
order by 2 desc;
```

Result of the query?

- a) **Two identical results**
- b) two identical errors
- c) Make some changes in column names and then there will be identical results
- d) none

2) Syntax for adding a primary key constraint

1. Select a.emp, a.name b. emp from employee a, select(emp from empl where sal =max(sal)) b;
(check ansr)

Ans: this is throw an error in the first line itself is wrong but there was no choice like that.

2. Decode("1611", aaaa, 1444, exists, non existent)
What will the ouput be if the column has the value 1444

Ans: exists

3. If you want the datatype of a column existing in some other table what will you use
Ans: %rowtype

8) For loop requires

Ans: EXIT to terminate

9) Which is used to open a cursor?

Ans:Open statement

33	34	35	36
37	38	39	40
41	42	43	44
45			

State whether the sentence below is True or false:
The OTHERS exception handler is used to handle the exception called OTHERS

Answer:

True
 False

17 of
Item colle

Ans: False

Time Remaining 0:03:01 Which of the following is necessary for executing your query on an existing view successfully?

1	2	3	4
5	6	7	8
9	10	11	12

Question Status

- ☆:Unanswered
- ★:Answered

a. None of the listed options

b. You need SELECT privileges on the view.

c. The underlying tables must be in the same schema.

d. You need SELECT privileges only on the underlying tables.

e. The underlying tables must have data.

Ans: You need SELECT privileges on the view

1	2	3	4
33	34	35	36
37	38	39	40
41	42	43	44
45			

Which of the following syntax turns an existing constraint on?

a. ALTER TABLE table_name STATUS ENABLE CONSTRAINT constraint_name;

b. None of the listed options

c. ALTER TABLE table_name ENABLE constraint_name;

21 of
Item colle

**Ans: ALTER TABLE table_name
ENABLE CONSTRAINT constraint_name;**

Which logical condition operator, used in a WHERE clause, returns TRUE only when both of the conditions are true?

a. AND

b. NOT

c. IN

d. OR

e. None of the listed options

Ans: EXIT

23 With reference to Oracle data storage structures, a cluster is defined as?

- a. A group of one or more tables which resides in a tablespace that is striped across multiple disks.**
- b. A group of table that each have more than 2 low cardinality columns.**
- c. A data structure where a group of one or more tables have their own dedicated tablespaces.**
- d. None of the listed options**
- e. A group of one or more tables that share the same data blocks because they share common columns and are often used together in join queries.**

Ans: A group of one or more tables that share the same data blocks because they share common columns and are often used together in join queries.

Identify the different Oracle constraints from the below given list.

1	2	3	4
---	---	---	---

a. UNIQUE

b. CHECK

c. NOT NULL

d. None of the listed options

e. PRIMARY KEY

Ans: all.

```
SELECT c.customer_id, o.order_id, o.order_date, p.product_name
FROM customer c, curr_order o, product p
WHERE customer.customer_id = curr_order.customer_id
AND o.product_id = p.product_id
ORDER BY o.order_amount;
```

If the above statement fails when executed, which of the following changes will correct the problem?

- a. Remove the table aliases from the WHERE clause.
- b. Include the ORDER_AMOUNT column in the SELECT list.
- c. Use the table aliases instead of the table names in the WHERE clause.
- d. Remove the table alias from the ORDER BY clause and use only the column name.
- e. None of the listed options

Ans: Use the table aliases instead of the table names in the WHERE clause

Which of the following clauses would you use in a SELECT statement to limit the display to those employees whose salary is greater than 5000?

- a. None of the listed options
- b. ORDER BY SALARY > 5000
- c. WHERE SALARY > 5000
- d. HAVING SALARY > 5000
- e. GROUP BY SALARY > 5000

Ans: WHERE SALARY>5000

Which of the following statements about constraint is true?

- a. You must provide a name for each constraint at the time of its creation.
- b. None of the listed options
- c. Constraint names are NOT required to follow the standard object-naming rules.
- d. Constraints only enforce rules at the table level.
- e. Constraints prevents a table with dependencies from being deleted.

Ans: You must provide a name for each constraint at the time of its creation

The STUDENT table contains these columns:
STU_ID NUMBER(9) NOT NULL
LAST_NAME VARCHAR2(30) NOT NULL
FIRST_NAME VARCHAR2(25) NOT NULL
DOB DATE
STU_TYPE_ID VARCHAR2(1) NOT NULL
ENROLL_DATE DATE

You create another table, named PT_STUDENT, with an identical structure. You want to insert all part-time students, who have a STU_TYPE_ID value of "P", into the new table. You execute this INSERT statement:

```
INSERT INTO pt_student
(SELECT stu_id, last_name, first_name, dob, sysdate
FROM student
WHERE UPPER(stu_type_id) = 'P');
```

What is the result of executing this INSERT statement?

- a. All part-time students are inserted into the PT_STUDENT table.
- b. An error occurs because the INSERT statement does NOT contain a VALUES clause.
- c. An error occurs because the STU_TYPE_ID column is NOT included in the subquery select list.
- d. None of the listed options
- e. An error occurs because both the STU_TYPE_ID and ENROLL_DATE columns are NOT included in the subquery select list.

Ans: An error occurs because the STU_TYPE_ID column is NOT included in the subquery select list.

Which SELECT statement implements a self join?

- a. `SELECT i.id_number, m.manufacturer_id
FROM inventory i NATURAL JOIN inventory m;`
- b. `SELECT i.id_number, m.manufacturer_id
FROM inventory i, manufacturer m
WHERE i.manufacturer_id = m.id_number;`
- c. `SELECT i.id_number, m.manufacturer_id
FROM inventory i, manufacturer m
WHERE i.manufacturer_id <> m.id_number;`
- d. `SELECT i.id_number, m.manufacturer_id
FROM inventory i, inventory m
WHERE i.manufacturer_id = m.id_number;`
- e. None of the listed options

Ans: D

Which of the following SELECT statement will result in 'ello world' from the string 'Hello World'?

- a. `SELECT LOWER(SUBSTR('Hello World', 1, 1) FROM dual;`
- b. `SELECT LOWER(SUBSTR('Hello World', 2, 1) FROM dual;`
- c. `SELECT SUBSTR('Hello World',1) FROM dual;`
- d. None of the listed options
- e. `SELECT LOWER(TRIM ('H' FROM 'Hello World')) FROM dual;`

Ans: `SELECT LOWER(TRIM('H' FROM 'Hello World')) FROM dual;`

Which of the following functions can be categorized as similar in function to an IF-THEN-ELSE statement?

- a. DECODE
- b. ROWIDTOCHAR
- c. NEW_TIME
- d. SQRT

Ans: DECODE

Which type of table is the best candidate to be cached?

- a. Large table rarely retrieved with a full table scan.
- b. None of the listed options
- c. Small table rarely retrieved with a full table scan.
- d. Small table frequently retrieved with a full table scan.
- e. Large table frequently retrieved with a full table scan.

Ans: Small table frequently retrieved with a full table scan.

- a. `ALTER TABLE inventory
ADD PRIMARY KEY (manufacturer_id);`
- b. None of the listed options
- c. `ALTER TABLE inventory
ADD CONSTRAINT manufacturer_id PRIMARY KEY;`
- d. `ALTER TABLE inventory
MODIFY manufacturer_id CONSTRAINT PRIMARY KEY;`
- e. `ALTER TABLE inventory
MODIFY CONSTRAINT PRIMARY KEY manufacturer_id;`

Ans: None of the listed options

Actual syntax:

```
ALTER TABLE inventory
  ADD CONSTRAINT man_id_pk PRIMARY KEY
INVENTORY(MANUFACTURER_ID)
```

Please verify the answers from your side and learn, the given answers might be wrong
And for some questions I don't know the answers.

- 1) SELECT trunc('30-OCT-2013','YEAR') FROM DUAL;
 - 42184
 - 42640
 - Syntax error
 - 42175
 - 2) Pragma that will call itself independently and can separate main user query from it is ...
 - Pragma autonomous transaction
 - Pragma exception init
 - None of options
 - Pragma restrict references
- ans: Pragma autonomous transaction
- 3) How do you display '01-01-2001' , from 01-jan-01
 - to_char('01-jan-01')
 - trunc(to_char('01-jan-01'))
 - round(to_char('01-jan-01'))
 - to_char(trunc('01-jan-01'))(not sure of the options, but something like this came)
 - 4) how to display "ello world" from Hello World
ans:lower(trim('H' from "Hello World"))

- 5) CREATE OR REPLACE FUNCTION CALC_PLAYER_AVG
(V_ID in PLAYER_BAT_STAT.PLAYER_ID%TYPE)
RETURN NUMBER IS V_AVG NUMBER;
BEGIN
 SELECT HITS / AT_BATS INTO V_AVG
 FROM PLAYER_BAT_STAT
 WHERE PLAYER_ID = V_ID;
 RETURN (V_AVG);
END;

Which statement will successfully invoke this function in SQL *Plus?

- SELECT CALC_PLAYER_AVG(PLAYER_ID) FROM PLAYER_BAT_STAT;
- EXECUTE CALC_PLAYER_AVG(31);
- CALC_PLAYER('RUTH');
- CALC_PLAYER_AVG(31);
- START CALC_PLAYER_AVG(31)

ans: SELECT CALC_PLAYER_AVG(PLAYER_ID) FROM PLAYER_BAT_STAT; and
CALC_PLAYER_AVG(31);

6) Which operation is used for joining two tables which have same number of columns:

- Union

7) Select 1 from dual

Union

Select 1 from dual

Union

Select 3 from dual;

Ans:

1

3 (because union, intersection and minus operations will display only distinct records)

8) how to you display the employee and department table, even if there are no matching departments (unable to frame question exactly)

ans: full outer join/outer join.

9) Which of the following are used to release all database locks? (check box)

- Commit
- Rollback
- Save
- Savepoint

ans: Rollback and Commit

10) DECLARE
p_emp_no NUMBER :=1;
BEGIN
FOR I IN 1....10 LOOP
update employee
set ename='John'
WHERE emp_no=p_emp_no;
END LOOP;
END;

How many execution plans generated for this statement.

- 1
- 10
- 5
- 1

ans: 1

11) When is the wildcard in a WHERE clause useful?

ans: An exact match is not possible in a SELECT statement

12) what is oracle views:

View is a virtual table that can be accessed via SQL commands

13) Which of the below datatypes are supported by Oracle?

- I. Scalar datatype
- II. Composite datatype
- III. Large Object (LOB) datatype

ans: all

14) which oracle access method make query processing easier.

- Creating unique index
- Primary key access
- Access with rowid
- Full table scan

ans: access with rowid

15) which of the following are true

a. group by clause is executed before having clause.

b. order by clause is executed after having clause.

(don't remember all options)

(order of execution of statements: from, where,group by, having ,select,order by)

16) The PRODUCT table contains these columns:

PRODUCT_ID NUMBER(9)
PRODUCT_NAME VARCHAR(25)
COST NUMBER(5,2)
LIST_PRICE NUMBER(5,2)
SUPPLIER_ID NUMBER(9)

Display product names, cost, supplier ids and average price.

(for this question they might change the ques , keeping the ans unchanged, read the ques carefully and ans accordingly)

ans: `SELECT product_name, cost, supplier_id , avg(list_price) from product group by product_name, cost, supplier_id;`

17) One question related to the above (16th question) type

Similarly, to display some columns along with that need to find the profit

Profit can be calculated as (retail amount - (retail amount *25%)) as profit

- 18) Which of the following is not correct about User_Defined Exceptions?
- a. Raised automatically in response to an Oracle Error
 - b. Must be declared
 - c. Must be handled by referencing the Exception name in the Exception Handler
 - d. Must be raised explicitly

19) (one more questions related to exceptions came)

- 20) Which of the statements about subqueries are true?
- a. A single row subquery can retrieve only one row but many columns
 - b. Subqueries can be correlated and non correlated
 - c. A multiple row subquery can be compared using the ">" operator
 - d. A multiple row subquery can use the "=" operator.

21) The user needs to perform certain data manipulation operations through a view called `EMP_DEPT_VU`, which he/she previously created. The user wants to look at the definition of the view. How he/she obtain the definition of the view using a query?

- a. Query the `USER_VIEWS` data dictionary to search for the `EMP_DEPT_VU` view.
- b. Query the `USER_OBJECTS` data dictionary view to search for the `EMP_DEPT_VU`
- c. Query the `USER_SOURCE` data dictionary view to search for the `EMP_DEPT_VU` view
- d. Use the `DEFINE VIEW` command on the `EMP_DEPT_VU` view.

22) STD_ID NUMBER(4)
COURSE_ID VARCHAR(10)
START_DATE DATE
END_DATE DATE

Which two aggregate functions are valid on the `START_DATE` column?

- a. COUNT(`start_date`)
- b. AVG(`start_date,end_date`)
- c. SUM(`start_date`)
- d. MIN(`start_date`)

ans: count,min

23) What is another name for simple join or inner join?
ans: Equijoin

24) SELECT to_char('05-09-2015',MONTH) from dual;

ans: September

25) All users currently have INSERT privileges on the LIBRARY table. You want only your users to insert into this table using the ADD_BOOK procedure. Which two actions must you take?

- a. GRANT EXECUTE ON ADD_BOOK TO PUBLIC
- b. REVOKE INSERT ON LIBRARY FROM PUBLIC

26) trunc(56.892)

ans: 56

27) Which of the following is an aggregate function?

ans: MIN

28) which function is used to add 4 zeros to the left of a column

ans: LPAD

29) select emp.ename, dept.dname from emp JOIN dept on emp.dno=dept.dno;
1. Natural Join 2. Inner Join 3. Equi Join 4. Outer Join

Ans: Equi Join

(similar question came)

30) 10. IF AN INTERSECT STATEMENT IS issued between two tables one table customer with 4 rows and another table customer_2 with 10 rows. But none of the customer_id are matching between both tables. Then how many rows are returned.

- 0
- 40
- 10
- 4

ans: 0

31) You cannot add a subquery to a SELECT clause as a column expression in the SELECT list.

ans: False

32) With SQL, how do you select all the records from a table named "Persons" where the value of the column " FirstName" starts with "a"?

ans: Select * from Persons where FirstName LIKE 'a%';

33) Consider a "DEPARTMENT" table with five records. How many rows will be updated by the below query?

UPDATE department SET department_id=5

ans: All Rows in the table
34) A view stores the data physically in database

ans: False

35) SELECT CUSTOMER.CUSTOMER_ID, ORDER.CUSTOMER_ID, NAME, ORDER_ID FROM CUSTOMER, ORDER (similar ques came)

ans: Cartesian join

36) If we trim S from SSNNSSS, then output:

- NN
- NNSSS
- BLANK OUTPUT
- NONE

ans: NN

37) Under which two circumstances do you design database triggers? (Choose two) (same ques different option came)

- A. To duplicate the functionality of other triggers.
- B. To replicate built-in constraints in the Oracle server such as primary key and foreign key.
- C. To guarantee that when a specific operation is performed, related actions are performed.
- D. For centralized, global operations that should be fired for the triggering statement, regardless of which user or application issues the statement.

ans: c and d

38) Update departments set department_id=(select department_id from departments); (same ques different options came)

This statement if department_id is returned multiple means what it will do?

- Throw syntax error
- No rows are updated
- It will update with department_id value
- None

ans: Throws syntax error

2-3 :ques related to types of joins
2 :questions find the error/ what is wrong in the given snippet
10 :ques on packages, triggers
2-3 :ques on subqueries
many :ques on trunc, round, date, aggregate function
1. :ques related to primary key, not null ,foreign key constrains

RDBMS Select Statements

Select Statements- Single, Multiple tables

2.Zee Rentals-Laptops hired in MayCoding exercise
2. Average Salary based on departmentCoding exercise
2.Staff details based on salaryCoding exercise
2.Supplier-Supply details of the goodsCoding exercise
2.Student details based on average marksCoding exercise
2.Shopify 1 - Maximum Shopping detailsCoding exercise
2.Sales- Data detailsCoding exercise
2.Shopify 2 Customers not placed ordersCoding exercise
2.Insurance-List of PoliciesCoding exercise
2.Hospital-Number of doctors based on shiftCoding exercise
2.Car pooling-Driver booking details based on nameCoding exercise
2.Cricket-Number of players in each cityCoding exercise
2. Dream Home-Maximum cost of flatCoding exercise
2. Event Hall-Number of booking customer wiseCoding exercise
2. Event Hall-Halls booked more than onceCoding exercise
2.Pizza-Highest Selling PizzaCoding exercise
2.Pizza-Delivery partner detailsCoding exercise
2.Pizza-Highest Business DateCoding exercise
Patient Appointment Details based on reasonCoding exercise
Student-Room DetailsCoding exercise
Movie details based on Certification and DurationCoding exercise
2.Review of hotel based on ratingCoding exercise
2.Movie details based on Language and DurationCoding exercise
2.Customer details based on pets purchasedCoding exercise
2.No of time rented by each carCoding exercise
Book detailsCoding exercise
Display User detailsCoding exercise
Customer city from addressCoding exercise
Customer DetailsCoding exercise
User informationCoding exercise
Display Junior Sailors detailsCoding exercise

Code Challenge - RDBMS DDL & DML

DDL & DML

Insurance- Alter table-add constraint(1.1)Coding exercise
Insurance-Update Agent details(1.2)Coding exercise
Hospital- Alter T_HOSPITAL table 1.1Coding exercise
Hospital-Update T_DOCTOR table 1.2Coding exercise
Car Pooling- Create BOOKING table 1.1Coding exercise
Car Pooling - Update booking table1.2Coding exercise
Cricket -Alter T_MATCH_SCORE_CARD table(1.1)Coding exercise
Cricket-Update T_PLAYER table(1.2)Coding exercise
Dream Home- Alter table t_flat_booking1.1Coding exercise

Dream Home- Update t_flat_details table 1.2 Coding exercise
Event Hall-Alter T_HALL_BOOKING table 1.1 Coding exercise
Event Hall -Update T_HALL_DETAILS table 1.2 Coding exercise
Event Hall- Alter table Hall Booking 1.1 Coding exercise
Event Hall- Update the event date 1.2 Coding exercise
Pizza Store- Alter table Pizza 1.1 Coding exercise
Pizza Store - Update PIZZA table 1.2 Coding exercise
Pizza Store- Alter table-Foreign key 1.1 Coding exercise
Pizza Store- Update PIZZA table discount 1.2 Coding exercise
Pizza Store- Alter table Pizza 1.1 Coding exercise
Pizza Store-Update PIZZA table 1.2 Coding exercise
Hospital- Add a new column set 1 Coding exercise
Hospital- Change the datatype/column Coding exercise
Create Movie_Master table set 1 Coding exercise
Movie - Modify the datatype Coding exercise
Hostel-Update Student Record Coding exercise
Hostel-Insert Student Records Coding exercise
ZeeRental- Alter table Laptop 1.1 Coding exercise
ZeeRental- Update ZEERENTALS table 1.2 Coding exercise
College- Alter table Department 1.1 Coding exercise
College- Modify Staff Salary 1.2 Coding exercise
College- Alter table Staff 1.1 Coding exercise
College - Update Staff Salary 1.2 Coding exercise
Supplier- Alter table Supplydetails 1.1 Coding exercise
School - Alter table- Foreign key 1.1 Coding exercise
Sales - Alter table SalesPerson 1.1 Coding exercise
Shopify 1- Alter table Customer 1.1 Coding exercise
Shopify 2- Alter table Customer 1.1 Coding exercise
Supplier- Update Price 1.2 Coding exercise
School-Update Grace Marks for Tenth Standard 1.2 Coding exercise
Sales - Update Price and Total Sale 1.2 Coding exercise
Shopify 1- Update amount based on month 1.2 Coding exercise
Shopify 2- Update amount 1.2 Coding exercise

2) 1 Hunger eats - Create table Coding exercise
• 2 Hunger eats - Update table Coding exercise

IV. 1 Insert records in to Theatre_master table Coding exercise
• 2 Alter Customer_master table Coding exercise

c. 1 Insert records into Pet_details table Coding exercise
e. 2 Alter Customer_Details table Coding exercise
f. Create Rentals Table Coding exercise
g. Update values in rental table Coding exercise

Insert records in to Book table Coding exercise
update Book table Coding exercise
Create buses table Coding exercise
Update buses table Coding exercise
Create Plan table Coding exercise
Update the data in GB per day Coding exercise

Insert customers tableCoding exercise
Hunger eats - Change the field nameCoding exercise
e. 1Update table ScheduleCoding exercise
e. 2Insert records into Buses tableCoding exercise
Insert records into Sailor_details tableCoding exercise
Update the seating capacity of Ultra Deluxe type boatsCoding exercise
Create Customer_Master tableCoding exercise
Insert Records - CustomersCoding exercise

Function-Scalar & Aggregate

Function-Scalar & Aggregate
3.Insurance-List of AgentsCoding exercise
3.Hospital-Total fees received based on gender and shiftCoding exercise
3.Car Pooling-Vehicle detailsCoding exercise
3.Cricket-Average runs of players based on nameCoding exercise
3.Dream Home-Customer name details based on total costCoding exercise
3. Event Hall-Average cost of not booked hallsCoding exercise
3.Event Hall-Average cost of booked hallsCoding exercise
3. Pizza-Low cost and High cost pizzaCoding exercise
3.Pizza-Framing Customer passwordCoding exercise
3.Pizza-Delivery Partner Details With RatingCoding exercise
Student Details In Capital CaseCoding exercise
Number of AppointmentsCoding exercise
Minimum & Maximum Discount AmountCoding exercise
3. Zee Rentals-Customers hiring laptop minimal timesCoding exercise
3. College C- Display the year based on more staff recruitedCoding exercise
3.College A-Staff detailsCoding exercise
3.Supplier- Maximum,Minimum and Average supply valuesCoding exercise
3. School -Overall marksCoding exercise
3.Sales-Region-wise sales summaryCoding exercise
3. Shopify 1Shopping DetailsCoding exercise
3.Shopify 2 Customer Details Based On Purchase AmountCoding exercise
3.Total rides taken by individual partnerCoding exercise
3.Customer details based on bookingCoding exercise
3.Display Pet detailsCoding exercise
Member details based on book countCoding exercise
3.Credential detailsCoding exercise
Display bus count based on durationCoding exercise
Customer details based on no of subscriptionsCoding exercise
Customer contact detailsCoding exercise
3.Number of buses from ChennaiCoding exercise
Display Welcome MessageCoding exercise

Code Challenge - Functions & SubQueries

Functions & SubQueries

- 4. Zee Rentals-laptop with minimum amountCoding exercise
- 4. College-C-Display Senior Staff and Junior staffCoding exercise
- 4. College A-Details of HOD in each departmentCoding exercise
- 4. Supplier-Supplier details based on supplyCoding exercise
- 4. School-Teacher detailsCoding exercise
- 4. Sales- person detailsCoding exercise
- 4. Shopify 1-Store Details Based On Orders ReceivedCoding exercise
- 4. Shopify 2-Store Details Based On Order DispatchCoding exercise
- 4. Insurance-Agent detailsCoding exercise
- 4. Hospital-Maximum fees paid patient detailsCoding exercise
- 4. Car Pooling-Maximum time driven driver detailsCoding exercise
- 4. Cricket-Player detailsCoding exercise
- 4. Dream Home -Flat details based on yearCoding exercise
- 4. Event Hall-Customer details with booking doneCoding exercise
- 4. Event Hall-Customer having Average paymentCoding exercise
- 4. Pizza-Extra Large PizzaCoding exercise
- 4. Pizza-Total Cost of Pizza OrderedCoding exercise
- 4. Pizza-Highest Business Customer DetailsCoding exercise
- Concatenating DetailsCoding exercise
- Patient Appointment details Based On MonthCoding exercise
- Room Details Based On LocationCoding exercise
- 4. Customer placed order more than 2 timesCoding exercise
- 4. Theatre - Movie DetailsCoding exercise
- 4. Purchase details of petsCoding exercise
- 4. Cars rented for maximum number of daysCoding exercise
- 4. Book details which are not returnedCoding exercise
- Display Bus booking detailsCoding exercise
- Customer recharge detailsCoding exercise
- Orders on weekendCoding exercise
- Users details based on ticket cancellationCoding exercise
- Display Boat details having Afternoon ShiftsCoding exercise