

DATABASE MANAGEMENT SYSTEM

ASSIGNMENT

ONLINE MOVIE TICKET BOOKING SYSTEM

Tables Structure:

1.Admin

```
SQL> desc admin
```

Name	Null?	Type
-----	-----	-----
USERNAME	NOT NULL	VARCHAR2(20)
PASSWORD	NOT NULL	VARCHAR2(20)

2.Movie

```
SQL> desc movie
```

Name	Null?	Type
-----	-----	-----
MOVIEID	NOT NULL	NUMBER(11)
MOVIEName		VARCHAR2(20)
IMAGEPATH		VARCHAR2(11)
DIRECTOR		VARCHAR2(26)
PRODUCER		VARCHAR2(26)
CAST		VARCHAR2(26)
DURATION		NUMBER
STORY		VARCHAR2(26)
TYPE		VARCHAR2(20)

3.Book

```
SQL> desc book
```

Name	Null?	Type
-----	-----	-----
STATUSID	NOT NULL	NUMBER
SEATID		NUMBER
SEATNAME		NUMBER
SHOWID		NUMBER
STATUS		NUMBER

4.Customer

```
SQL> desc customer
```

Name	Null?	Type
CUSTOMERID	NOT NULL	NUMBER
USERNAME		VARCHAR2(20)
PASSWD		VARCHAR2(26)
FNAME		VARCHAR2(11)
LNAME		VARCHAR2(20)
GENDER		VARCHAR2(26)
ADDRESS		VARCHAR2(26)
CITY		VARCHAR2(26)
PINCODE		NUMBER
STATE		VARCHAR2(26)
EMAIL		VARCHAR2(26)
MOBILE		NUMBER

5.Screen

```
SQL> desc screen
```

Name	Null?	Type
SCREENID	NOT NULL	NUMBER(26)
SCREENNO		NUMBER
THEATREID		NUMBER

6.Seat

```
SQL> desc seat
```

Name	Null?	Type
SEATID	NOT NULL	NUMBER
SCREENID		NUMBER
ROWNAME		VARCHAR2(30)
SEATNAME		VARCHAR2(20)
IMAGEURLCUSTOMER		VARCHAR2(30)

7.Show

```
SQL> desc show
```

Name	Null?	Type
SHOWID	NOT NULL	NUMBER
MOVIEID		NUMBER
THEATREID		NUMBER
SCREENID		NUMBER
STARTTIME		TIMESTAMP(6)
ENDTIME		TIMESTAMP(6)
DATES		DATE
PRATE		NUMBER
GRATE		NUMBER
SRATE		NUMBER

8.Theatre

```
SQL> desc theatre
```

Name	Null?	Type
-----	-----	-----
THEATREID	NOT NULL	NUMBER
THEATRENAME		VARCHAR2(20)
CITYID		NUMBER
ADDRESS		VARCHAR2(16)
PINCODE		NUMBER
NOSSSHOW		NUMBER

9.Ticket

```
SQL> desc ticket
```

Name	Null?	Type
-----	-----	-----
TICKETID	NOT NULL	NUMBER
SHOWID		NUMBER
SEATS		VARCHAR2(20)
SEATTYPE		VARCHAR2(20)
NOSEATS		NUMBER
AMOUNT		NUMBER
ISPAID		VARCHAR2(20)
MOBILE		NUMBER(11)
EMAIL		VARCHAR2(20)

10.Feedback

```
SQL> desc feedback
```

Name	Null?	Type
-----	-----	-----
FEEDBACKID	NOT NULL	NUMBER
USERNAME		VARCHAR2(20)
EMAILID		VARCHAR2(20)
FEEDBACK		VARCHAR2(30)

INSERTION:

1.Admin

```
SQL> select * from admin;
```

USERNAME	PASSWORD
-----	-----
arun	13579
ajay	02468
ananth	12345
akash	67890
bala	77777

2.Movie

```
SQL> select * from movie;
```

MOVIEID	MOVIE NAME	IMAGEPATH	DIRECTOR
1	jilla		neason
2	kathi		an murugadoss
3	sivaji		shankar
4	petta		karthi subraj

PRODUCER	CAST	DURATION
1	49 crore	2.16
2	69 crore	2316
3	89 crore	2.36
4	64 crore	2.46

STORY	TYPE
1	action
2	drama
3	drama
4	

3.Booking

```
SQL> select * from booking;
```

STATUSID	SEATID	SEATNAME	SHOWID	STATUS
3	1	13	3	101
1	4	7	2	1
2	3	13	4	1001
4	2	13	3	18
5	1	13	5	28

4.Customer

```
SQL> select * from customer;
```

CUSTOMERID	USERNAME	PASSWD	FNAME
LNAME	GENDER	ADDRESS	
CITY	PINCODE	STATE	
EMAIL	MOBILE		
102	pazani	banana	muruga
vel	male	pazani	
pazan		tamilnadu	
murugavel@gmail.com	9868462123		

CUSTOMERID	USERNAME	PASSWD	FNAME
LNAME	GENDER	ADDRESS	
CITY	PINCODE	STATE	
EMAIL	MOBILE		
104	vijay	mersal	sun
king	male	coimbatore	
coimbatore		641035 tamilnadu	
sunking@gmail.com	8364268462		

CUSTOMERID	USERNAME	PASSWD	FNAME
------------	----------	--------	-------

CUSTOMERID	USERNAME	PASSWD	FNAME
LNAME	GENDER	ADDRESS	
CITY	PINCODE	STATE	
EMAIL	MOBILE		
106	welcome	welcome	karthik
raja	male	chennai	
chennai	600012	tamilnadu	
karthik@gmail.com	9726268462		
CUSTOMERID	USERNAME	PASSWD	FNAME
LNAME	GENDER	ADDRESS	
CITY	PINCODE	STATE	
EMAIL	MOBILE		
108	system	wakeup	yogesh
kumar	male	mumbai	
mumbai	8726268412	tamilnadu	
CUSTOMERID	USERNAME	PASSWD	FNAME

5.Screen

```
SQL> select * from screen;
```

SCREENID	SCREENNO	THEATREID
1	46	106
2	68	104
3	84	104
4	74	108
5	74	110

6.Seats

```
SQL> select * from seats;
```

SEATID	SCREENID	ROWNAME	SEATNAME
--------	----------	---------	----------

IMAGEURLCUSTOMER			

1	23	7	
---	----	---	--

2	46	3	
---	----	---	--

3	66	1	
---	----	---	--

SEATID	SCREENID	ROWNAME	SEATNAME
--------	----------	---------	----------

IMAGEURLCUSTOMER			

4	92	2	
---	----	---	--

5	48	4	
---	----	---	--

7.Show

```
SQL> select * from show;
```

SHOWID	MOVIEID	THEATREID	SCREENID
--------	---------	-----------	----------

STARTTIME			

ENDTIME			

DATES	PRATE	GRATE	SRATE
-------	-------	-------	-------

1	3	2	1
---	---	---	---

02-APR-19	01.00.00.000000	AM	
-----------	-----------------	----	--

02-APR-19	03.36.00.000000	AM	
-----------	-----------------	----	--

04-APR-19	4	8	1
-----------	---	---	---

SHOWID	MOVIEID	THEATREID	SCREENID
--------	---------	-----------	----------

STARTTIME			

ENDTIME			

DATES	PRATE	GRATE	SRATE
-------	-------	-------	-------

2	1	4	2
---	---	---	---

02-APR-19	04.00.00.000000	AM	
-----------	-----------------	----	--

02-APR-19	07.26.00.000000	AM	
-----------	-----------------	----	--

04-APR-19	2	1	7
-----------	---	---	---

SHOWID	MOVIEID	THEATREID	SCREENID

STARTTIME			

ENDTIME			

DATES	PRATE	GRATE	SRATE

3	5	2	3
02-APR-19	08.00.00.000000	AM	
02-APR-19	11.46.00.000000	AM	
04-APR-19	2	3	9

SHOWID	MOVIEID	THEATREID	SCREENID

STARTTIME			

ENDTIME			

DATES	PRATE	GRATE	SRATE

4	2	3	4
02-APR-19	02.00.00.000000	AM	
02-APR-19	05.16.00.000000	AM	
04-APR-19	1	4	6

SHOWID	MOVIEID	THEATREID	SCREENID

STARTTIME			

ENDTIME			

DATES	PRATE	GRATE	SRATE

5	3	1	5
22-MAR-19	01.00.00.000000	PM	
22-MAR-19	05.46.00.000000	PM	
22-MAR-19	1	3	7

8.Theatre

```
SQL> select * from theatre;
```

THEATREID	THEATRENAME	CITYID	ADDRESS	PINCODE

NOSSSHOW				

1	sathyam	3	chennai	600012
2				
2	lenaa	1	villupuram	606305
1				
3	navaneetham	5	villupuram	606305
14				

THEATREID	THEATRENAME	CITYID	ADDRESS	PINCODE

NOSSSHOW				

4	star	2	coimbatore	600035
8				
5	actors	4	coimbatore	600035
18				

9.Ticket

```
SQL> select * from ticket;
```

TICKETID	SHOWID	SEATS	SEATTYPE	NOSEATS
AMOUNT	ISPAID	MOBILE	EMAIL	
1	550 yes	4 three	962472214	first
2	1050 no	1 seven	8624721674	first
3	750 yes	5 seven	7724721674	second
TICKETID	SHOWID	SEATS	SEATTYPE	NOSEATS
AMOUNT	ISPAID	MOBILE	EMAIL	
4	1750 yes	1 ten	6724721674	first
5	250 no	2 two	9324721674	second

10.Feedback

```
SQL> select * from feedback;
```

FEEDBACKID	USERNAME	EMAILID
FEEDBACK		
1	varun	
nice		
2	karthi	karthi@gmail.com
average		
3	ajith	ak@gmail.com
super		
FEEDBACKID	USERNAME	EMAILID
FEEDBACK		
4	yogesh	yogesh@gmail.com
super		
5	pranav	
average		

2.2 b) DML (DATA MANIPULATION LANGUAGE)

Data Manipulation Language (DML) statements or commands are used for managing data within tables. Some commands of DML are:

Write the Queries for the following:

- 1) Update the value for the column TicketId and No.of Seats in relation **Ticket**.

```
SQL> update ticket set ticketid=1,seats='five' where showid=4;
1 row updated.
```

- 2) Display all the records from the relation **Customer_details**.

```
SQL> select * from cus_details;
```

CUSTOMERID	USERNAME	PASSWD	FNAME
LNAME	GENDER	ADDRESS	
CITY	PINCODE	STATE	
EMAIL	MOBILE		
102 pazani	male	banana	muruga
vel		pazani	
pazan		tamilnadu	
murugavel@gmail.com	9868462123		

CUSTOMERID	USERNAME	PASSWD	FNAME
LNAME	GENDER	ADDRESS	
CITY	PINCODE	STATE	
EMAIL	MOBILE		
104 vijay	male	mersal	sun
king		coimbatore	
coimbatore	641035	tamilnadu	
sunking@gmail.com	8364268462		

- 3) List only first three tuple from the relation **Ticket**.

```
SQL> select * from ticket where rownum<4;
```

TICKETID	SHOWID	SEATS	SEATTYPE	NOSEATS
AMOUNT	ISPAID		MOBILE	EMAIL
1	550	yes	4 five	962472214
2	1050	no	1 seven	8624721674
3	750	yes	5 seven	7724721674
				first
				first
				second

- 4) Display the column TheatreId,theatrename,address from the Relation Theatre.

```
SQL> select theatreid,theatrename,address from theatre;
```

THEATREID	THEATRENAME	ADDRESS
1	sathyam	chennai
2	lenaa	villupuram
3	navaneetham	villupuram
4	star	coimbatore
5	actors	coimbatore

- 5) Insert a new customer into customer with all the required details.

```
SQL> insert into cus_details(customerid,username,passwd,fname,lname,gender,address,city,pincode,state,email,mobile)values(7,'system','oracle','santhosh','kumar','male','thiruppur','thiruppur',641654,'tamilnadu','kumar123@gmail.com',8746431267);
1 row created.
```

2.3 A) SELECTION OPERATIONS

- 1) Display the first name, last name, and customerId for all customers whose last name is “King”.

```
SQL> select fname,lname,customerid from cus_details where lname='king';
```

FNAME	LNAME	CUSTOMERID
sun	king	104

- 2) Find the TicketId for all customers where the amount is greater than 1000.

```
SQL> select ticketid from ticket where amount>1000;

TICKETID
-----
         2
         4
```

- 3) Display the records of the customer_details where the first name or last name starts with S.

```
SQL> select * from cus_details where fname like 's%' or lname like 's%';

CUSTOMERID USERNAME          PASSWD          FNAME
-----
LNAME          GENDER          ADDRESS
-----
CITY          PINCODE STATE
-----
EMAIL          MOBILE
-----
          104 vijay          mersal          sun
king          male          coimbatore
coimbatore          641035 tamilnadu
sunking@gmail.com          8364268462

CUSTOMERID USERNAME          PASSWD          FNAME
-----
LNAME          GENDER          ADDRESS
-----
CITY          PINCODE STATE
-----
EMAIL          MOBILE
-----
          7 system          oracle          santhosh
kumar          male          thiruppur
thiruppur          641654 tamilnadu
kumar123@gmail.com          8746431267
```

4. Display details of customer with ID 101 or 110.

```
SQL> select * from cus_details where customerid=101 or customerid=110;
```

CUSTOMERID	USERNAME	PASSWD	FNAME
LNAME	GENDER	PINCODE	STATE
CITY	ADDRESS	MOBILE	
110	raj	male	kumar
pulb	chennai	600012	772626832
			tamilnadu
			chennai
			pranav

- 4) Display the first name concatenated with the last name, separated by comma, and salary, for all customer whose customerId not in the range between 105 and 110. Sort the query in ascending order by the full name.

```
SQL> select fname||','||lname from cus_details where customerid not between 105 and 110 order by fname||lname asc;
```

FNAME ',' LNAME
muruga,vel
sun,king

- 5) Find all name of customers whose Address is Coimbatore.

```
SQL> select fname||lname from cus_details where address='coimbatore';
```

FNAME LNAME
sunking

- 6) Show all Feedbackid,feedback and corresponding emailId in FeedBack relation.

```
SQL> select feedbackid,feedback from feedback where emailid is not null;
```

FEEDBACKID	FEEDBACK
2	average
3	super
4	super

- 7) Display ticketId,the difference between minimum and maximum amount for tickets with max amount in the range 750 to 1000.

```
SQL> select ticketid,max(amount)-min(amount) from ticket group by ticketid having max(amount) between 750 and 1000;
```

TICKETID	MAX(AMOUNT)-MIN(AMOUNT)
3	0

- 8) Display the last name and address for all customers live in Chennai.

```
SQL> select lname,address from cus_details where address='chennai';
```

LNAME	ADDRESS
-----	-----
raja	chennai
pulb	chennai

2.3B) OPERATORS

- 1) Change amount of ticket from 100 to 150 if the existing amount is less than 600.

```
SQL> update ticket set amount=amount+50 where amount<600;
```

2 rows updated.

- 2) Display the customerid, first name ,state, city and mobile number for all customers whose customerId is not equal to 127, 110, or 103 .

```
SQL> select customerid, fname, state, city, mobile from cus_details where customerid not in(127,110,103);
```

CUSTOMERID	FNAME	STATE	CITY
-----	-----	-----	-----
MOBILE			

102 muruga	tamilnadu	pazan	
9868462123			
104 sun	tamilnadu	coimbatore	
8364268462			
106 karthik	tamilnadu	chennai	
9726268462			

CUSTOMERID	FNAME	STATE	CITY
-----	-----	-----	-----
MOBILE			

108 yogesh	tamilnadu	mumbai	
8726268412			

- 3) Display all the data from Customers table for all customers who have the letters: *l*, *j* or *h* in their last name. Order the query in descending order by customerId

```
SQL> select * from cus_details where lname like '%j' or lname like '%h';
no rows selected
```

- 4) Display the first name, Mobile number and pincode for all customers without a pincode (pincode is null).

```
SQL>
      select fname, mobile, pincode from cus_details where pincode is null;

FNAME          MOBILE          PINCODE
-----
muruga         9868462123
yogesh         8726268412
pranav         772626832
```

- 5) Display the seattype, No of seats and mobile number for all ticketed whose mobile number is not null.

```
SQL> select seattype, seats, mobile from ticket where mobile is not null;

SEATTYPE          SEATS          MOBILE
-----
first             three          962472214
first             seven          8624721674
second            seven          7724721674
first             ten            6724721674
second            two            9324721674
```

- 6) Display the seatname, rowname whose seatId is A4, H5, and B2.

```
SQL> select seatname, rownum from seats where seatid in(4,5,2);

SEATNAME          ROWNUM
-----
1
2
3
```

- 7) Display the customerId where Fname starts with 'h' and ends with 'a'

```
SQL> select customerid from cus_details where fname like 'h%a';
no rows selected
```

- 8) Find all customers in the database who does not have the emailId.


```
SQL> select * from cus_details where email is null;
```

CUSTOMERID	USERNAME	PASSWD	FNAME
LNAME	GENDER	ADDRESS	
CITY	PINCODE	STATE	
EMAIL	MOBILE		
108	system	wakeup	yogesh
kumar	male	mumbai	
mumbai		tamilnadu	
		8726268412	

- 9) Increase the Ticket_amount by 20 % whose seat_type is 'Box'

```
SQL> update ticket set amount=amount+(0.2*amount) where seattype='box';
1 row updated.
```

- 10) Display the names of all customers whose street address includes the substring 'Main'

```
SQL> select fname from cus_details where address like '%main%';

FNAME
-----
sun
```

3.1 AGGREGATE FUNCTIONS

- 1) Display the minimum duration of movie.

```
SQL> select min(duration) from movie;

MIN(DURATION)
-----
2.16
```

- 2) Display the no of customer_id whose gender is 'male'.

```
SQL> select count(customerid) from cus_details where gender='male';
```

COUNT(CUSTOMERID)
5

- 3) Find the difference between the highest and lowest number of amount in ticket relation.

```
SQL> select max(amount)-min(amount) from ticket;
```

MAX(AMOUNT)-MIN(AMOUNT)
1500

- 4) Find the total no of grate and srate including null values

```
SQL> select sum(srate) srate,sum(grate) grade from show;
```

SRATE	GRADE
30	19

- 5) Find the maximum number of feedback for each customer

```
SQL> select max(feedback) from feedback group by feedbackid;
```

MAX(FEEDBACK)
nice
average
super
average
super

- 6) Display state, no.of customers in the state where the customer name contains the word 'raj'.

```
SQL> select state,count(customerid) from cus_details where username like '%raj%'group by state;
```

STATE	COUNT(CUSTOMERID)
tamilnadu	1

- 7) Display highest no.of tickets bought by a single customer.

```
SQL> select ticketid,max(trunc(amount/50)) from ticket group by ticketid;
```

TICKETID	MAX(TRUNC(AMOUNT/50))
1	11
2	21
4	35
5	5
3	15

- 8) Display how many seats are still not filled.

```
SQL> select 100-sum(amount)/50 from ticket;

100-SUM(AMOUNT)/50
-----
                13
```

3.2 BUILT IN FUNCTIONS

3.2 A) STRING FUNCTION

- 1) Display first name and last name after converting the first letter of each name to upper case and the rest to lower case.

```
SQL> select initcap(fname),initcap(lname) from cus_details;

INITCAP(FNA  INITCAP(LNAME)
-----
Muruga      Vel
Sun         King
Karthik     Raja
Yogesh      Kumar
Pranav      Pulb
```

- 2) Display the length of first name for customer where last name contain character 'b' after 3rd position.

```
SQL> select length(fname) from cus_details where lname like '___%b%';

LENGTH(FNAME)
-----
                6
```

- 3) Select first 3 characters of first_name from customer

```
SQL> select substr(fname,0,3) name from cus_details;

NAME
-----
mur
sun
kar
yog
pra
```

- 4) Display the bit length of cust_name.

- 5) Combine custome_id and last_name in a single column with label "combined_name".

```
SQL> select customerid||lname as "combined name" from cus_details;

combined name
-----
102vel
104king
106raja
108kumar
110pulb
```

- 6) Replace the theatre_name from 'sathiyam' to 'abrami'.

```
SQL> update theatre set theatrename='abirami' where theatrename='sathyam';

1 row updated.
```

- 7) Get first_name from customer table after removing white spaces from left side.

```
SQL> select ltrim(fname) from cus_details;

LTRIM(FNAME)
-----
muruga
sun
karthik
yogesh
pranav
```

- 8) Display the total number of characters in seatname.

```
SQL> select length(seatname) from seats;

LENGTH(SEATNAME)
-----
```

3.2 B) DATE FUNCTIONS

- 1) Get the current date in the following format.

- i) Thursday September 2014

```
SQL> select to_char(current_date,'day month yyyy') from dual;

TO_CHAR(CURRENT_DATE,'DAYMONTHYYYY')
-----
thursday april 2019
```

- 2) Display the movieId,ShowId in which shows have in the month of august.

```
SQL> select movieid,showid from show where extract(month from dates)=8;
```

MOVIEID	SHOWID
2	4

- 3) get srate, prate, and date of first show of the movie

```
SQL> select srate,prate,dates from show where screenid=1;
```

SRATE	PRATE	DATES
1	4	04-APR-19

- 4) Display the difference between the start time and end time of each show

```
SQL> select endtime-starttime from show;
```

ENDTIME-STARTTIME
+000000000 02:36:00.000000
+000000000 03:26:00.000000
+000000000 03:46:00.000000
+000000000 03:16:00.000000
+000000000 04:46:00.000000

- 5) Extract month from Date column in show relation.

```
SQL> select extract(month from dates) from show;
```

EXTRACT(MONTHFROMDATES)
4
4
4
8
3

- 6) Display the index of the week for the date 2016-05-19.

```
SQL> select round(dates,'day') from show;
```

ROUND(DAT
07-APR-19
07-APR-19
07-APR-19
04-AUG-19
24-MAR-19

- 7) Display show start time after subtracting 10 days from current date in show relation

```
SQL> select starttime-10 from show;

STARTTIME
-----
23-MAR-19
23-MAR-19
23-MAR-19
23-MAR-19
12-MAR-19
```

- 8) Display the number of days from year 0 to 2009-05-18.

```
SQL> select months_between('01-Jan-00','18-May-2009')*extract(day from last_day(dates)) from show;

MONTHS_BETWEEN('01-JAN-00','18-MAY-2009')*EXTRACT(DAYFROMLAST_DAY(DATES))
-----
-3376.4516
-3376.4516
-3376.4516
-3489
-3489
```

- 9) Display the day name from the date from show table.

```
SQL> select to_char(dates,'day') from show;

TO_CHAR(DATES,'DAY')
-----
thursday
thursday
thursday
sunday
friday
```

3.2 C) NUMBER FUNCTIONS AND CONTROL FLOW FUNCTIONS

- 1) Display the screenno column in decimal number format.

```
SQL> select round(screenno) from screen;
```

```
ROUND(SCREENNO)
```

```
-----
          46
          68
          84
          74
          74
```

- 2) Display the records randomly from feedback table.

```
SQL> select * from feedback where rownum=&n;
Enter value for n: 1
old 1: select * from feedback where rownum=&n
new 1: select * from feedback where rownum=1
```

```
FEEDBACKID USERNAME          EMAILID
```

```
-----
```

```
FEEDBACK
```

```
-----
          1 varun
nice
```

- 3) Find absolute value for Ticket_id from Ticket table.

```
SQL> select abs(ticketid) from ticket;
```

```
ABS(TICKETID)
```

```
-----
          1
          2
          3
          4
          5
```

- 4) Display the current user.

```
SQL> select user from dual;
```

```
USER
```

```
-----
```

```
SYSTEM
```

- 5) Display the Address if Customer_id is less than 10 else lists the Mobile_no.

```
SQL> select customerid,
2  (case
3  when customerid<10 then address
4  else fname
5  end) result from cus_details;
```

CUSTOMERID	RESULT
102	muruga
104	sun
106	karthik
108	yogesh
110	pranav

- 6) Display the sqrt of the SeatId.

```
SQL> select sqrt(seatid) from seats;
```

SQRT(SEATID)
1
1.41421356
1.73205081
2
2.23606798

- 7) Display the ASCII value of Last_Name from Customer table.

```
SQL> select ascii(lname) from cus_details;
```

ASCII(LNAME)
118
107
114
107
112

4.1 SUBQUERY

1. Display details of ticket who has booked max no of seats

```
SQL> select * from ticket where amount=(select max(amount) from ticket);
```

TICKETID	SHOWID	SEATS	SEATTYPE	NOSEATS
4	1	ten	first	
1750	yes		6724721674	

- 1) Display details of customers who bought any ticket worth more than 600


```
SQL> select * from cus_details where customerid in
2 (select ticketid+100 from ticket where amount>600);
```

CUSTOMERID	USERNAME	PASSWD	FNAME
102	pazani	banana	muruga
103	vel	male	pazani
104	pazan	tamilnadu	
105	murugavel@gmail.com	9868462123	

- 2) Display details of tickets that are purchased by customer 102

```
SQL> select * from ticket where ticketid in
2 (select customerid-100 from cus_details where customerid=102);
```

TICKETID	SHOWID	SEATS	SEATTYPE	NOSEATS
2	1	seven	first	
1050	no		8624721674	

- 3) List the theatre name,theatre id,address which has more than 3 shows per day

```
SQL> select theatreid,theatrename,address from theatre
2 where theatreid in(select theatreid-100 from screen where screenid>3);
```

THEATREID	THEATRENAME	ADDRESS
4	star	coimbatore

- 4) Find the customer details that they bought the highest amount of ticket.

```
SQL> select * from cus_details where customerid in
2 (select ticketid+100 from ticket where amount=(select max(amount) from ticket));
```

CUSTOMERID	USERNAME	PASSWD	FNAME
104	vijay	mersal	sun
105	king	male	coimbatore
106	coimbatore	641035 tamilnadu	
107	sunking@gmail.com	8364268462	

- 5) List the movieid,producername,type of movie where customerid is more than 106.

```
SQL> select movieid,producer,type from movie
2 where movieid in(select customerid-100 from cus_details);
```

MOVIEID	PRODUCER	TYPE
2	subaskaran	
4	kalanithi maran	

- 6) display details of show where theatre name ends with 'am' .

```
SQL> select * from show where
2 theatreid in (select theatreid from theatre where theatrename like '%am');
```

SHOWID	MOVIEID	THEATREID	SCREENID
STARTTIME			
ENDTIME			
DATES	PRATE	GRATE	SRATE
4	2	3	4
02-APR-19 02.00.00.000000 AM			
02-APR-19 05.16.00.000000 AM			
04-AUG-19	1	4	6

- 7) Display the ticketid,no of seats,seatttype where srate is greater than 1000.

```
SQL> select ticketid,seats,seatttype from ticket
2 where showid in(select showid from show where srate>1000);
```

TICKETID	SEATS	SEATTTYPE
5	two	second

4.2 JOIN

- 1) Prepare a list with cust_name,directorname and their story deatils of movie.

```
SQL> select cus_details.customerid, movie.moviename, movie.director from cus_details, movie;
```

CUSTOMERID	MOVIENAME	DIRECTOR
102	jilla	neason
104	jilla	neason
106	jilla	neason
108	jilla	neason
110	jilla	neason
102	kathi	ar murugadoss
104	kathi	ar murugadoss
106	kathi	ar murugadoss
108	kathi	ar murugadoss
110	kathi	ar murugadoss
102	sivaji	shankar

- 2) Make a list in ascending order of ticketid in which screenid and theatreid are same .

```
SQL> select screen.screenid, theatre.theatreid, ticket.ticketid from ticket, theatre, screen;
```

SCREENID	THEATREID	TICKETID
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
1	2	1
2	2	1
3	2	1
4	2	1
5	2	1
1	3	1

- 3) List the showid, ticket_amount and their date of show

```
SQL> select ticket.showid, ticket.amount, show.dates from ticket, show;
```

SHOWID	AMOUNT	DATES
4	550	04-APR-19
1	1050	04-APR-19
5	750	04-APR-19
1	1750	04-APR-19
2	250	04-APR-19
4	550	04-APR-19
1	1050	04-APR-19
5	750	04-APR-19
1	1750	04-APR-19
2	250	04-APR-19
4	550	04-APR-19

- 4) Display all the theatre name and the seat_id by implementing a left outer join

```
SQL> select t.theatrename,s.seatid from theatre t left join seats s
2 on t.theatreid=s.seatid;
```

THEATRENAME	SEATID
abirami	1
lenaa	2
navaneetham	3
star	4
actors	5

- 5) Display details of customer for ticket with amount more than 1000

```
SQL> select c.* from cus_details c,ticket t
2 where t.amount>1000;
```

CUSTOMERID	USERNAME	PASSWD	FNAME
102	pazani	banana	muruga
104	vijay	mersal	sun

- 6) Make a list with ticket_id,firstname and amount for those ticket id whose amount is between 500 and 2000.

```
SQL> select t.ticketid,c.fname,t.amount from ticket t,cus_details c
2 where t.amount between 500 and 2000;
```

TICKETID	FNAME	AMOUNT
1	muruga	550
1	sun	550
1	karthik	550
1	yogesh	550
1	pranav	550
2	muruga	1050
2	sun	1050
2	karthik	1050
2	yogesh	1050
2	pranav	1050
3	muruga	750

- 7) Display customerid,customer name, where the customeraddress and theatre address are same.

```
SQL> select c.customerid,c.username from cus_details c,theatre t
2 where c.address=t.address;
```

CUSTOMERID	USERNAME
110	raj
106	welcome
104	vijay
104	vijay

- 8) Display address of theatre,theatre name where screened greater than 112.

```
SQL> select t.address,t.theatrename from theatre t,screen s
2 where screenid>112;
```

ADDRESS	THEATRENAME
chennai	abirami
villupuram	lenaa
villupuram	navaneetham
coimbatore	star
coimbatore	actors

- 9) Display customer name and total amount of ticket purchased by customer

```
SQL> select c.customerid,sum(t.amount) from cus_details c,ticket t
      2 group by c.customerid;
```

CUSTOMERID	SUM(T.AMOUNT)
102	4350
104	4350
106	4350
108	4350
110	4350

5.1 STORED PROCEDURE

- 1) Create a procedure to insert value for the table movie.

```
SQL> create or replace procedure movie_val
  2 is
  3 begin
  4 insert into movie values(&movieid,&moviename,&imagepath,&directr,&producer,&cast,&duration,&stort,&type);
  5 end;
  6 /
Enter value for movieid: 6
Enter value for moviename: 'theri'
Enter value for imagepath: null
Enter value for directr: 'atlee'
Enter value for producer: 'vijay'
Enter value for cast: '16 laks'
Enter value for duration: 3
Enter value for stort: 'action'
Enter value for type: null
old  4: insert into movie values(&movieid,&moviename,&imagepath,&directr,&producer,&cast,&duration,&stort,&type);
new  4: insert into movie values(6,'theri',null,'atlee','vijay','16 laks',3,'action',null);
Procedure created.
```

- 2) Create a procedure to display the show details that have show_date on the month of december. Pass the month as a parameter to the procedure.

```
SQL> create or replace procedure display(mo in number)
  2 is
  3 no number;
  4 detail show%rowtype;
  5 begin
  6 no:=mo;
  7 select * into detail from show where extract(month from dates)=no;
  8 dbms_output.put_line('movie id'||detail.movieid);
  9 dbms_output.put_line('show id'||detail.showid);
 10 end;
 11 /
Procedure created.
```