Introduction to DBMS and SQL

Database:

A **database** is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning.

Database Management System:

A **database management system (DBMS)** is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

SQL

SQL stands for Structured Query Language. It is an ANSI (American National Standards Institute) standard. SQL is a standard language for storing, manipulating and retrieving data in databases. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

SQL is widely popular because it offers the following advantages:

- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

SQL Commands

- The standard SQL commands to interact with relational databases are classified into the following groups based on their nature

1. DDL - Data Definition Language
2. DML - Data Manipulation Language
3. DCL - Data Control Language

1. DDL - Data Definition Language

- Used to define, change and drop the structure of a table

Sl. No.	Command & Description
1	**CREATE** Creates a new table, a view of a table, or other object in the database.
2	**ALTER** Modifies an existing database object, such as a table.
3	**DROP** Deletes an entire table, a view of a table or other objects in the database.

2. DML - Data Manipulation Language

- Used to insert, modify, delete and retrieve data from a table

Sl.No.	Command & Description
1	**SELECT** **Retrieves certain records from one or more tables.**
2	**INSERT** **Creates a record.**
3	**UPDATE** **Modifies records.**
4	**DELETE** **Deletes records.**

3. DCL - Data Control Language

- used to provide control over the data in a database

Sl.No.	Command & Description
1	**GRANT** Gives a privilege to user.
2	**REVOKE** Takes back privileges granted from user.

SQL Data Types

- **Numeric data types**

  - integer numbers of various sizes (INTEGER or INT, and SMALLINT)
  - floating-point (real) numbers of various precision (FLOAT or REAL, and DOUBLE PRECISION).
  - Formatted numbers can be declared by using DECIMAL(i,j)—or DEC(i,j) or NUMERIC(i,j)—where

    i -precision, total number of decimal digits
    j - scale, number of digits after the decimal point.

- **Character-string data types**

  - fixed length—CHAR(n) or CHARACTER(n), where n is the number of characters
  - varying length—VARCHAR(n) or CHAR VARYING(n) or CHARACTER VARYING(n), where n is the maximum number of characters
  - When specifying a literal string value, it is placed between single quotation marks (apostrophes), and it is case sensitive

- A **Boolean** data type has the traditional values of TRUE or FALSE
- In SQL, because of the presence of NULL values, a three-valued logic is used, so a third possible value for a Boolean data type is UNKNOWN

- The **DATE** data type has ten positions, and its components are YEAR, MONTH, and DAY in the form YYYY-MM-DD
- The **TIME** data type has at least eight positions, with the components HOUR, MINUTE, and SECOND in the form HH:MM:SS.

Specifying Key and Referential Integrity Constraints

- Constraints are the rules enforced on data columns on a table
  - Primary key: uniquely identifies each row/record in a database table
  - Foreign key: uniquely identifies a row/record in another table
  - NOT NULL: ensures that a column cannot have a NULL value

CREATE Table

- used to create a new table.
- Syntax:

```
TABLE table_name (column1 datatype,column2 datatype,column3
 ., columnN datatype, PRIMARY KEY( one or more columns ) );
```

INSERT

- used to add new rows of data to a table in the database.
- Syntax:

```
INSERT INTO TABLE_NAME (column1, column2, column3, ...columnN)  VALUES
(value1, value2, value3,...valueN);
```

- Here, column1, column2, column3,...columnN are the names of the columns in the table into which you want to insert the data.

SELECT

- used to fetch the data from a database table which returns this data in the form of a result table.
- Syntax:

```
SELECT column1, column2, columnN FROM table_name;
```

Here, column1, column2... are the fields of a table whose values you want to fetch.

- If you want to fetch all the fields available in the field, then you can use the following syntax.

```
SELECT * FROM table_name;
```

WHERE

- clause is used to specify a condition while fetching the data from a single table or by joining with multiple tables.
- Syntax

SELECT column1, column2, columnN FROM table_name WHERE [condition]

UPDATE

- used to modify the existing records in a table. You can use the WHERE clause with the UPDATE query to update the selected rows, otherwise all the rows would be affected.
- Syntax

```
UPDATE table_name SET column1 = value1, column2 = value2...., columnN = valueN
WHERE [condition];
```

- You can combine N number of conditions using the AND or the OR operators.

DROP TABLE

- used to remove a table definition and all the data, indexes, triggers, constraints and permission specifications for that table.
- Syntax:

```
DROP TABLE table_name;
```

Lab Programs

1. **Consider the following schema for a Library Database:**
   BOOK (BOOK_ID, TITLE, PUBLISHER_NAME, PUB_YEAR)
   BOOK_AUTHORS (BOOK_ID, AUTHOR_NAME)
   PUBLISHER (NAME, ADDRESS, PHONE)
   BOOK_COPIES (BOOK_ID, BRANCH_ID, NO_OF_COPIES)
   BOOK_LENDING (BOOK_ID, BRANCH_ID, CARD_NO, DATE_OUT, DUE_DATE)
   LIBRARY_BRANCH (BRANCH_ID, BRANCH_NAME, ADDRESS)
   CARD (CARD_NO)
   Write SQL queries to

a) Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch.
b) Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.
c) Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
d) Create a view of all books and its number of copies that are currently available in the library**.**
e) Delete a book in the BOOK table. Update the contents of other tables to reflect this data manipulation operation.

Table Creation

```
CREATE TABLE PUBLISHER (NAME VARCHAR (20) PRIMARY KEY,PHONE INTEGER,
ADDRESS VARCHAR (20));

CREATE TABLE BOOK (BOOK_ID INTEGER PRIMARY KEY,TITLE VARCHAR (20),
PUB_YEAR VARCHAR (20), PUBLISHER_NAME VARCHAR(20) REFERENCES
PUBLISHER (NAME) ON DELETE CASCADE);

CREATE TABLE BOOK_AUTHORS(AUTHOR_NAME VARCHAR (20),
BOOK_ID INT REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,PRIMARY KEY
(BOOK_ID, AUTHOR_NAME));

CREATE TABLE LIBRARY_BRANCH (BRANCH_ID INTEGER PRIMARY KEY,
BRANCH_NAME VARCHAR (50), ADDRESS VARCHAR (50));

CREATE TABLE BOOK_COPIES(NO_OF_COPIES INTEGER,
BOOK_ID INT REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
BRANCH_ID INT REFERENCES LIBRARY_BRANCH (BRANCH_ID) ON DELETE
CASCADE, PRIMARY KEY (BOOK_ID, BRANCH_ID));

CREATE TABLE CARD (CARD_NO INTEGER PRIMARY KEY);
```

CREATE TABLE BOOK_LENDING(DATE_OUT DATE, DUE_DATE DATE,

BOOK_ID INT REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE, BRANCH_ID INT REFERENCES LIBRARY_BRANCH (BRANCH_ID) ON DELETECASCADE, CARD_NO INT REFERENCES CARD (CARD_NO) ON DELETE CASCADE,PRIMARY KEY (BOOK_ID, BRANCH_ID, CARD_NO));

```
SQL> select * from publisher;

NAME                      PHONE ADDRESS
-------------------- ---------- --------------------
MCGRAW-HILL          9989076587 BANGALORE
PEARSON              9889076565 NEWDELHI
RANDOM HOUSE         7455679345 HYDRABAD
HACHETTE LIVRE       8970862340 CHENAI
GRUPO PLANETA        7756120238 BANGALORE
```

```
SQL> SELECT * FROM BOOK;

   BOOK_ID TITLE                PUB_YEAR             PUBLISHER_NAME
---------- -------------------- -------------------- --------------------
         1 DBMS                 JAN-2017             MCGRAW-HILL
         2 ADBMS                JUN-2016             MCGRAW-HILL
         3 CN                   SEP-2016             PEARSON
         4 CG                   SEP-2015             GRUPO PLANETA
         5 OS                   MAY-2016             PEARSON
```

```
SQL> SELECT * FROM BOOK_AUTHORS;

AUTHOR_NAME             BOOK_ID
-------------------- ----------
NAVATHE                      1
NAVATHE                      2
TANENBAUM                    3
EDWARD ANGEL                 4
GALVIN                       5
```

```
SQL> SELECT * FROM LIBRARY_BRANCH;

 BRANCH_ID BRANCH_NAME                                        ADDRESS
---------- -------------------------------------------------- ------------------------------------
        10 RR NAGAR                                           BANGALORE
        11 RNSIT                                              BANGALORE
        12 RAJAJI NAGAR                                       BANGALORE
        13 NITTE                                              MANGALORE
        14 MANIPAL                                            UDUPI
```

```
SQL> SELECT * FROM BOOK_COPIES;

NO_OF_COPIES    BOOK_ID  BRANCH_ID
------------ ---------- ----------
          10          1         10
           5          1         11
           2          2         12
           5          2         13
           7          3         14
           1          5         10
           3          4         11
```

```
SQL> SELECT * FROM CARD;

   CARD_NO
----------
       100
       101
       102
       103
       104
```

```
SQL> select * from book_lending;

DATE_OUT  DUE_DATE     BOOK_ID  BRANCH_ID    CARD_NO
--------- --------- ---------- ---------- ----------
01-JAN-17 01-JUN-17          1         10        101
11-JAN-17 11-MAR-17          3         14        101
21-FEB-17 21-APR-17          2         13        101
15-MAR-17 15-JUL-17          4         11        101
12-APR-17 12-MAY-17          1         11        104
```

Queries:

a) Retrieve details of all books in the library – id, title, name of publisher, authors,number of copies in each branch, etc.

SELECT B.BOOK_ID, B.TITLE, B.PUBLISHER_NAME, A.AUTHOR_NAME, C.NO_OF_COPIES, L.BRANCH_ID FROM BOOK B, BOOK_AUTHORS A, BOOK_COPIES C, LIBRARY_BRANCH LWHERE B.BOOK_ID=A.BOOK_ID AND B.BOOK_ID=C.BOOK_ID AND L.BRANCH_ID=C.BRANCH_ID;

BOOK_ID	TITLE	PUBLISHER_NAME	AUTHOR_NAME	NO_OF_COPIES	BRANCH_ID
1	DBMS	MCGRAW-HILL	NAVATHE	10	10
1	DBMS	MCGRAW-HILL	NAVATHE	5	11
2	ADBMS	MCGRAW-HILL	NAVATHE	2	12
2	ADBMS	MCGRAW-HILL	NAVATHE	5	13
3	CN	PEARSON	TANENBAUM	7	14
5	OS	PEARSON	GALVIN	1	10
4	CG	GRUPO PLANETA	EDWARD ANGEL	3	11

b) Get the particulars of borrowers who have borrowed more than 3 books, but fromJan 2017 to Jun 2017.

SELECT CARD_NO FROM BOOK_LENDING
WHERE DATE_OUT BETWEEN '2017-01-01' AND '2017-07-31' GROUP BY CARD_NO
HAVING COUNT (*)>3;

CARD_NO
101

c) Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.

DELETE FROM BOOKWHERE BOOK_ID=3;

```
SQL> DELETE FROM BOOK
  2  WHERE BOOK_ID=3;

1 row deleted.

SQL> SELECT * FROM BOOK;
```

BOOK_ID	TITLE	PUB_YEAR	PUBLISHER_NAME
1	DBMS	JAN-2017	MCGRAW-HILL
2	ADBMS	JUN-2016	MCGRAW-HILL
4	CG	SEP-2015	GRUPO PLANETA
5	OS	MAY-2016	PEARSON

d) Partition the BOOK table based on year of publication. Demonstrate its working with asimple query.

CREATE VIEW V_PUBLICATION AS SELECT PUB_YEAR FROM BOOK;

SELECT *FROM V_PUBLICATION;

```
PUB_YEAR
-------------
JAN-2017
JUN-2016
SEP-2016
SEP-2015
MAY-2016
```

e) Create a view of all books and its number of copies that are currently available in the Library.

```
CREATE VIEW V_BOOKS AS
SELECT B.BOOK_ID, B.TITLE, C.NO_OF_COPIES
FROM BOOK B, BOOK_COPIES C, LIBRARY_BRANCH LWHERE
B.BOOK_ID=C.BOOK_ID
AND C.BRANCH_ID=L.BRANCH_ID;
```

BOOK_ID	TITLE	NO_OF_COPIES
1	DBMS	10
1	DBMS	5
2	ADBMS	2
2	ADBMS	5
3	CN	7
5	OS	1
4	CG	3

```
SELECT *FROM V_BOOKS;
```

2. Consider the following schema for Order Database:

SALESMAN (SALESMAN_ID, NAME, CITY, COMMISSION)
CUSTOMER (CUSTOMER_ID, CUST_NAME, CITY, GRADE, SALESMAN_ID)
ORDERS (ORD_NO, PURCHASE_AMT, ORD_DATE, CUSTOMER_ID, SALESMAN_ID)

Write SQL queries to
a) Count the customers with grades above Bangalore's average.
b) Find the name and number of all salesmen who had more than one customer.
c) List all the salesman and indicate those who have and do not have customers in their cities (Use UNION operation.)
d) Create a view that finds the salesman who has the customer with the highest order of a day.
e) Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

Table Creation

```
create table Salesman(
Salesman_ID int primary key,
Name varchar(20),
City varchar(20),
Commission varchar(20));
```

```
//insert into Salesman values(&Salesman_ID,'&Name','&City','&Commission');
```

SALESMAN_ID	NAME	CITY	COMMISSION
1000	John	Bangalore	25 %
2000	Ravi	Bangalore	20 %
3000	Kumar	Mysuru	15 %
4000	Smith	Delhi	30 %
5000	Harsha	Hyderabad	15 %

```
Create table Customer(Customer_ID int primary key,
      Cust_Name varchar(20),
      City varchar(20),
      Grade int,
      Salesman_ID int,
     foreignkey(Salesman_ID) references Salesman(Salesman_ID) on delete set null);

//insert into Customer values(&Customer_ID,'&Cust_Name','&City',&Grade,&Salesman_ID);
```

CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID
10	Preethi	Bangalore	100	1000
11	Vivek	Mangalore	300	1000
12	Bhaskar	Chennai	400	2000
13	Chethan	Bangalore	200	2000
14	Mamatha	Bangalore	400	3000

```
create table Orders(Ord_No int primary key,
      Purchase_Amt float,
      Ord_Date date,
      Customer_ID int references Customer(Customer_ID) on delete set null,
      Salesman_ID  int references Salesman(Salesman_ID) on delete set null);

//insert into Orders values(&Ord_No,&Purchase,'&Ord_Date',&Customer_ID,&Salesman_ID);
```

ORD_NO	PURCHASE_AMT	ORD_DATE	CUSTOMER_ID	SALESMAN_ID
50	5000	04-MAY-17	10	1000
51	450	20-JAN-17	10	2000
52	1000	20-JAN-17	13	2000

53	3500	13-APR-17	14	3000
54	550	9-MAR-17	12	2000

Queries :

a. select count(*) from Customer where Grade > (select avg(Grade) from Customer where city = 'Bangalore');

COUNT(*)
3

select grade,count(Customer_ID) from customer having grade >

(Select avg(grade) from Customer where city='Bangalore') group by grade;

GRADE	COUNT(CUSTOMER_ID)
400	2
300	1

b. select Salesman_ID, Name from Salesman S having

(select count(*) from Customer where Salesman_ID=S.Salesman_ID) > 1 group by Name,

Salesman_ID;

SALESMAN_ID	NAME
1000	John
2000	Ravi

c.

select S.Salesman_ID, S.Name, C.Cust_Name from Salesman S, Customer C where S.City=C.City

union

SELECT S.SALESMAN_ID, S.NAME, 'No-Match'

FROM SALESMAN S

WHERE S.CITY NOT IN (SELECT DISTINCT CITY FROM CUSTOMER);

SALESMAN_ID	NAME	CUST_NAME
1000	John	Chethan
1000	John	Mamatha
1000	John	Preethi
2000	Ravi	Chethan
2000	Ravi	Mamatha
2000	Ravi	Preethi
3000	Kumar	No-Match
4000	Smith	No-Match
5000	Harsha	No-Match

d. create view salesman_order as select b.Ord_Date, a.Salesman_ID,a.Name from Salesman a, Orders b where a.Salesman_ID=b.Salesman_ID and b.Purchase_Amt= (select max(Purchase_Amt) from Orders);

select *from salesman_order;

e. delete from Salesman where Salesman_ID=1000;

SALESMAN_ID	NAME	CITY	COMMISSION
2000	Ravi	Bangalore	20 %
3000	Kumar	Mysuru	15 %

4000	Smith	Delhi	30 %
5000	Harsha	Hyderabad	15 %

CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID
10	Preethi	Bangalore	100	
11	Vivek	Mangalore	300	
12	Bhaskar	Chennai	400	2000
13	Chethan	Bangalore	200	2000
14	Mamatha	Bangalore	400	3000

ORD_NO	PURCHASE_AMT	ORD_DATE	CUSTOMER_ID	SALESMAN_ID
50	5000	04-MAY-17	10	
51	450	20-JAN-17	10	2000
52	1000	20-JAN-17	13	2000
53	3500	13-APR-17	14	3000
54	550	09-MAR-17	12	2000

3. Consider the schema for Company Database: Exercise

EMPLOYEE(SSN, NAME, ADDRESS, SEX, SALARY, SUPERSSN, DNO)
DEPARTMENT(DNO, DNAME, MGRSSN, MGRSTARTDATE)
DLOCATION(DNO,DLOC)
PROJECT(PNO, PNAME, PLOCATION, DNO)
WORKS_ON(SSN, PNO, HOURS)
Write SQL queries to
a) Make a list of all project numbers for projects that involve an employee whose last name is 'Scott', either as a worker or as a manager of the department that Controls the project.
b) Show the resulting salaries if every employee working on the 'IoT' project is Given a 10 percent raise.
c) Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department
d) Retrieve the name of each employee who works on all projects controlled by department number 5.
e) For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.

Create table Depts (Dname varchar(20),Dno int Primary Key, Mgrstrdate date, Mgrssn int);

insert into Depts values ('Accounts', 1, '1998-08-12', 333), ('Research', 2, '2000-01-24',111), ('Production', 3,'2009-02-14', 444), ('sales',4,'2000-03-15', 555),('Testing', 5, '2002-04-23', 777);

DNAME	DNO	MGRSTRDAT	MGRSSN
Accounts	1	12-AUG-98	333
Research	2	24-JAN-00	111
production	3	14-FEB-09	444
sales	4	15-MAR-00	555
Testing	5	23-APR-02	777

Alter table Depts add constraint foreign key(Mgrssn) references Emp(SSN) on delete set NULL;

Create table Emp(SSN int Primary Key,

FName varchar(20),LName varchar(20),

Address varchar(20),

Sex char(1),

Salary int, superssn int, Dno int, foreign key(superssn) references Emp(ssn) on delete cascade, foreign key(Dno) references Depts(Dno) on delete cascade);

insert into Emp values(111, 'james', 'scott', 'mysore', 'M', 123456, 111, 1), (222, 'ramesh', 'aravnid', 'banglore', 'M', 345672, 111, 1), (333,'david', 'hush', 'Honalulu', 'M', 70000, 222,1), (444, 'kavya',

'bind', 'guwahathi', 'F', 750000, 444,1), (555, 'hashmi', 'khan', 'mumbai', 'F', 450000, 333,1), (666, 'pniky', 'khanna', 'punjab', 'F', 560000, 666, 1), (777,'kushal', 'kundan', 'assam', 'M', 35000, 555,4), (888, 'Arjith', 'gundal', 'assam', 'M', 660000, 444, 5);

SSN	FNAME	LNAME	ADDRESS	S	SALARY	SUPERSSN	DNO
111	james	scott	mysore	M	123456	111	1
222	Ramesh	Aravind	Bangalore	M	345672	111	1
333	David	Hush	Honululu	M	70000	222	1
444	Kavya	Bind	Guwahati	F	750000	444	1
555	Hashmi	Khan	Mumbai	F	450000	333	1
666	Pinky	Khanna	Punjab	F	560000	666	1
777	Kushal	Kundan	Assam	M	35000	555	4
888	Arijith	Gundal	Assam	M	660000	444	5

Create table Deptloc (Dno int references Depts(dno) , Dloc varchar(20));

insert into Deptloc values(1, 'Babglore'), (2,'Mysore'), (3,'hubli'), (4, 'punjab'), (5,' guwahathi');

DNO	DLOC
1	Banglore
2	Mysore
3	Hubli
4	Punjab
5	Guwahati

Create table Prj (Pno int Primary Key , Pname varchar(20), Ploc varchar(20), Dno int, foreign key(Dno) references Depts(dno));

insert into Prj values(10,'p1', 'banglore',1), (20,'p2','mysore',3), (30,'p3','mumbai',4), (40,'p4', 'assam', 4), (50,'IOT','punjab',5), (60,'p6','honalulu',1);

PNO	PNAME	PLOC	DNO
10	P1	Blore	1
20	P2	Mysore	3
30	P3	Mumbai	4
40	P4	Assam	4
50	IOT	Punjab	5
60	P6	Honululu	1

Create table Work(SSN int references Emp(SSN), Pno int, foreign key (Pno) references Prj(pno), hours float);

insert into work values(111,50,5), (222, 50,10), (333, 50,7), (444,20,8), (555, 10,3), (666, 30,1), (777,40,12);

SSN	PNO	HOURS
111	50	5
222	50	10
333	50	7
444	20	8
555	10	3
666	30	1
777	40	12

Queries:

a. Select distinct p.pno from Prj p, Depts d, Emp e where e.dno=d.dno and d.dno = p.dno and (e.lname = 'scott' or d.mgrssn in (select ssn from Emp where lname='scott'));

PNO
60
10

b. Select e.ssn , e.fname,e.lname, 1.1*e.salary as raisedsal from emp e , prj p, work w where p.pname = 'IOT' and p.pno = w.pno and e.ssn = w.ssn;

SSN	FNAME	LNAME	RAISEDSAL
111	james	scott	135801.6
222	Ramesh	Aravind	380239.2
333	David	Hush	77000

c. Select sum(salary) as sumsal, avg(salary) as avgsal, min(salary) as minsal, max(salary) as maxsal from Emp e, Depts d where e.dno = d.dno and d.dname='Accounts';

SUMSAL	AVGSAL	MINSAL	MAXSAL
2299128	383188	70000	750000

d. select e.fname, e.lname from emp e join work w on e.ssn=w.ssn join prj p on w.pno=p.pno where p.dno=5;

FNAME	LNAME
james	scott
Ramesh	Aravind
David	Hush

e. select e.dno, count(*) as Num_emp from emp e where e.salary>600000 and e.dno in (select dno from emp group by dno having count(dno)>5) group by e.dno;

4. **The following relations keep track of airline flight information: Exercise**
FLIGHTS (FLNO: INTEGER, SOURCE: STRING, DESTINATION: STRING, DISTANCE: INTEGER, DEPARTS:TIME, ARRIVES: TIME, PRICE: INTEGER)

AIRCRAFT (AID: INTEGER, ANAME: STRING, CRUISINGRANGE: INTEGER)
CERTIFIED (EID: INTEGER, AID: INTEGER)
EMPLOYEES (EID: INTEGER, ENAME: STRING, SALARY: INTEGER)
Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft, and only pilots are certified to fly.
Write SQL queries to
a) Find the names of aircraft such that all pilots certified to operate them earn more than $80,000.
b) For each pilot who is certified for more than three aircraft, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
c) For all aircraft with cruisingrange over 1000 miles, find the name of the aircraft and the average salary of all pilots certified for this aircraft.
d) Print the enames of pilots who can operate planes with cruising range greater than 1000 miles but are not certified on any Goibibo aircraft.
e) Print the name and salary of every nonpilot whose salary is more than the average salary for pilots.

create table Flights(Flno int primary key, Source varchar(20), Destination varchar(20),Distance int,DepartTime time,ArrivalTime time, Price int);

insert into Flights values(101, 'mysore', 'banglore', 140, '09:30:00', '10:30:00', 5000);

FLNO	SOURCE	DESTINATION	DISTANCE	DEPARTTIME	ARRIVALTIM	PRICE
101	Mysuru	Banglore	140	09:30:00	10:30:00	5000
102	Mysuru	Hassan	120	13:00:00	14:00:00	4500
103	Mysuru	Manglore	600	06:00:00	09:00:00	15000
104	Mysuru	Ballary	750	07:00:00	12:00:00	19500
105	Banglore	Hyderabad	850	12:00:00	16:00:00	20000

create table AirCraft(Aid int primary key, Aname varchar(20), CrusingRange int);

AID	ANAME	CRUSINGRANGE
1001	KingFisher	2500
2002	Deccan	2800
3003	Dishan	3000
4004	GoIbibo	750
5005	Local	900

create table Employees(Eid int primary key, Ename varchar(20), Salary int);

EID	ENAME	SALARY
11	Shivakumar	150000
12	Bhaskar	75000
13	DishanKrishna	200000
15	Vishnu	80000
20	Dhanu	90000

create table Certified(Eid int, Aid int, foreign key(Eid) references Employees(Eid), foreign key(Aid) references AirCraft(Aid));

EID	AID
11	3003
11	1001
11	2002
13	4004
12	2002

Queries:

a. **select Aname from AirCraft A, Employees E, Certified C where A.Aid = C.Aid and E.Eid = C.Eid and E.Salary > 80000;**

ANAME

Dishan

KingFisher

Deccan

GoIbibo

b. select E.Eid, Max(CrusingRange) from Employees E, Certified C, AirCraft A where E.Eid=C.Eid and A.Aid = C.Aid group by E.Eid having count(*) > 3;

EID	MAX(CRUSINGRANGE)
11	3000

c. select A.Aname ,avg(Salary) from Employees E, Certified C , Aircraft A where C.Eid = E.Eid and A.Aid = C.Aid and A.CrusingRange>1000 group by A.Aname;

ANAME	AVG(SALARY)
Deccan	112500
KingFisher	90000
Dishan	150000

d. (select Ename from Employees E, Certified C where C.Aid in

(select A.Aid from Aircraft A, Certified C where A.Aid not in

(select Aid from Aircraft where CrusingRange>1000)

group by A.Aid) and E.Eid = C.Eid)

Minus

(select ename from Employees E, Certified C where C.Aid in (

select A.Aid from Aircraft A, Certified C where A.Aid in

(select Aid from Aircraft where Aname = 'Goibibo')

group by A.Aid) and E.Eid = C.Eid);

e. select Eid, Ename, Salary from Employees where salary > (select avg(salary) from Employees) and Eid not in (select Eid from Certified);

5. **Consider the following relations Structured Enquiry**

STUDENT(SNUM: INTEGER, SNAME: STRING, MAJOR: STRING, LEVEL: STRING, AGE: INTEGER)
CLASS(CNAME: STRING, MEETS AT: STRING, ROOM: STRING, FID: INTEGER)
ENROLLED(SNUM: INTEGER, CNAME: STRING)
FACULTY(FID: INTEGER, FNAME: STRING, DEPTID: INTEGER)
The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class.
Write the following queries in SQL. No duplicates should be printed in any of the answers.
a) Find the names of all Juniors (level = JR) who are enrolled in a class taught by Rakesh.
b) Find the age of the oldest student who is either a history major or enrolled in a course taught by Ravi.

c) Find the names of all students who are enrolled in two classes that meet at the same time.
d) For each faculty member that has taught classes only in room R128, print the faculty member's name and the total number of classes she or he has taught.
e) Create a view that contains the details of students along with the name of the courses enrolled.

Create table Student (Snum int Primary Key,Sname varchar(20),Major varchar(10), Lev varchar(20),age int);

SNUM	SNAME	MAJOR	LEV
1	Andy	CSE	JR
2	Helen	Arts	JR
3	Bob	Psychology	GR
4	Pham	History	SR
5	ZOla	CSE	SR
6	ABhi	CSE	JR
7	Tara	Arts	JR

Create table Faculty(Fid int Primary Key,Fname varchar(20),Depid int);

FID	FNAME	DEPID
22	Kera	88
23	Niall	77
24	Ravi	33
27	Lou	66
31	Jane	44
34	Payne	55
36	Rakesh	22

Create table Class(Cname varchar(20) Primary Key, Meets_at varchar(20),Room varchar(20),Fid int, foreign key(fid) references Faculty(Fid) on delete set NULL);

CNAME	MEETS_AT	ROOM	FID
HIS38	12:00	R123	34
CS293	12:00	R124	23
CS262	2:00	R125	24
PY331	9:00	R126	27
AR32	9:00	R127	36
AR36	10:00	R128	31
PY332	8:00	R128	22
HIS37	5:00	R128	34
HIS34	4:00	R128	31
CS289	2:00	R128	22

Create table Enrolled (Snum int, foreign key(Snum)
references Student(Snum),Cname varchar(10),foreign key(Cname) references Class(Cname));

```
 SNUM CNAME
----- --------
    1 CS262
    1 AR32
    2 AR32
    2 PY331
    3 PY331
    3 HIS34
    4 HIS38
    5 CS293
    5 HIS38
    6 CS262
```

drop table Class cascade constraints;

drop table Faculty cascade constraints;

Queries

a**. Select Distinct S.Snum, S.Sname from Student S,Class C, Enrolled E,Faculty F where S.Snum = E.Snum and E.Cname = C.Cname and C.Fid = F.Fid and F.Fname = 'Rakesh' and S.Lev = 'JR' order by Snum;**

```
 SNUM SNAME
--------------
     1 Andy
     2 Helen
```

b**. Select MAX(S.age) as Age from Student S where (S.Major = 'History') OR S.Snum in (Select E.Snum from Class C , Enrolled E, Faculty F where E.cname = C.cname and C.Fid = F.Fid and F.Fname = 'Ravi');**

```
  AGE
------
```

c. **Select Distinct S.Sname from Student S where S.Snum in (Select E1.Snum from Enrolled E1,Enrolled E2, Class C1, Class C2 where E1.snum = E2.snum and E1.cname <> E2.cname and E1.cname = C1.cname and E2.cname = C2.cname and C1.meets_at = C2.meets_at);**

SNAME

Helen

Zola

d. **Select Distinct F.Fname , count(*) as CourseCount from CLass C , Faculty F where C.FID not in (Select Fid from Class where Room IN (Select Room from Class where Room!='R128')) AND C.Fid = F.Fid group by F.Fname;**

FNAME	COURSECOUNT
Kera	2
Jane	2

Viva Questions

1. What is SQL?

Structured Query Language

2. What is database?

A database is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.

3. What is DBMS?

It is a collection of programs that enables user to create and maintain a database. In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.

4. What is a Database system?

The database and DBMS software together is called as Database system.

5. Advantages of DBMS?

- Redundancy is controlled.
- Unauthorized access is restricted.
- Providing multiple user interfaces.
- Enforcing integrity constraints.
- Providing backup and recovery.

6. Disadvantage in File Processing System?

- Data redundancy & inconsistency.
- Difficult in accessing data.
- Data isolation.
- Data integrity.
- Concurrent access is not possible.
- Security Problems.

7. Describe the three levels of data abstraction?

There are three levels of abstraction:

- Physical level: The lowest level of abstraction describes how data are stored.
- Logical level: The next higher level of abstraction, describes what data are stored indatabase and what relationship among those data.

- View level:The highest level of abstraction describes only part of entire database.

8. Define the "integrity rules"

There are two Integrity rules.

- Entity Integrity:States that –Primary key cannot have NULL value‖
- Referential Integrity:States that –Foreign Key can be either a NULL value or should be Primary Key value of other relation.

9. What is extension and intension?

Extension - It is the number of tuples present in a table at any instance. This is time dependent.

Intension -It is a constant value that gives the name, structure of table and the constraints laid on it.

10. What is Data Independence?

Data independence means that –the application is independent of the storage structure and access strategy of data‖. In other words, The ability to modify the schema definition in one level should not affect the schema definition in the next higher level.

Two types of Data Independence:

- Physical Data Independence: Modification in physical level should not affect the logical level.
- Logical Data Independence: Modification in logical level should affect the view level.

NOTE: Logical Data Independence is more difficult to achieve.

11. What is a view? How it is related to data independence?

A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base table. In other words, there is no stored file that direct represents the view instead a definition of view is stored in data dictionary.

Growth and restructuring of base tables is not reflected in views. Thus the view can insulate users from the effects of restructuring and growth in the database. Hence accounts for logical data independence.

12. What is Data Model?

A collection of conceptual tools for describing data, data relationships data semantics and constraints.

13. What is E-R model?

This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

14. What is Object Oriented model?

This model is based on collection of objects. An object contains values stored in instance variables within the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain same types of values and the same methods are grouped together into classes.

15. What is an Entity?

It is an 'object' in the real world with an independent existence.

16. What is an Entity type?

It is a collection (set) of entities that have same attributes.

17. What is an Entity set?

It is a collection of all entities of particular entity type in the database.

18. What is an Extension of entity type?

The collections of entities of a particular entity type are grouped together into an entity set.

19. What is an attribute?

It is a particular property, which describes the entity.

20. What is a Relation Schema and a Relation?

A relation Schema denoted by R(A1, A2, …, An) is made up of the relation name R and the list of attributes A_i that it contains. A relation is defined as a set of tuples. Let r be the relation which contains set tuples (t1, t2, t3, ...,tn). Each tuple is an ordered list of n-values t=(v1,v2, ..., vn).

21. What is degree of a Relation?

It is the number of attribute of its relation schema.

22. What is Relationship?

It is an association among two or more entities.

23. What is Relationship set?

The collection (or set) of similar relationships.

24. What is Relationship type?

Relationship type defines a set of associations or a relationship set among a given set ofentity types.

25. What is degree of Relationship type?

It is the number of entity type participating.

26. What is DDL (Data Definition Language)?

A data base schema is specified by a set of definitions expressed by a special language called DDL.

27. What is VDL (View Definition Language)?

It specifies user views and their mappings to the conceptual schema.

28. What is SDL (Storage Definition Language)?

This language is to specify the internal schema. This language may specify the mapping between two schemas.

29. What is Data Storage - Definition Language?

The storage structures and access methods used by database system are specified by a set of definition in a special type of DDL called data storage- definition language.

30. What is DML (Data Manipulation Language)?

This language that enable user to access or manipulate data as organized by appropriate data model.

- Procedural DML or Low level: DML requires a user to specify what data are neededand how to get those data.
- Non-Procedural DML or High level: DML requires a user to specify what data areneeded without specifying how to get those data.

31. What is DML Compiler?

It translates DML statements in a query language into low-level instruction that the query evaluation engine can understand.

32. What is Relational Algebra?

It is a procedural query language. It consists of a set of operations that take one or two relations as input and produce a new relation.

33. What is Relational Calculus?

It is an applied predicate calculus specifically tailored for relational databases proposedby E.F. Codd. E.g. of languages based on it are DSL, ALPHA, QUEL.

34. What is normalization?

It is a process of analyzing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties

- Minimizing redundancy
- Minimizing insertion, deletion and update anomalies.

35. What is Functional Dependency?

A Functional dependency is denoted by X Y between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuple that can form a relation state r of R. The constraint is for any two tuples t1 and t2 in r if t1[X] = t2[X] then they have t1[Y] = t2[Y]. This means the value of X component of a tuple uniquely determines the value of component Y.

36. When is a functional dependency F said to be minimal?

- Every dependency in F has a single attribute for its right hand side.
- We cannot replace any dependency X A in F with a dependency Y A where Y is a proper subset of X and still have a set of dependency that is equivalent to F.
- We cannot remove any dependency from F and still have set of dependency that is equivalent to F.

37. What is Multivalued dependency?

Multivalued dependency denoted by X Y specified on relation schema R, where X and Y are both subsets of R, specifies the following constraint on any relation r of R: if two tuples t1 and t2 exist in r such that t1[X] = t2[X] then t3 and t4 should also exist in r with the following properties

- t3[x] = t4[X] = t1[X] = t2[X]
- t3[Y] = t1[Y] and t4[Y] = t2[Y]
- t3[Z] = t2[Z] and t4[Z] = t1[Z]

where [Z = (R-(X U Y))]

38. What is Lossless join property?

It guarantees that the spurious tuple generation does not occur with respect to relation schemas after decomposition.

39. What is 1 NF (Normal Form)?

The domain of attribute must include only atomic (simple, indivisible) values.

40. What is Fully Functional dependency?

It is based on concept of full functional dependency. A functional dependency X Y is fully functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

41. What is 2NF?

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

42. What is 3NF?

A relation schema R is in 3NF if it is in 2NF and for every FD X A either of the following is true

- X is a Super-key of R.
- A is a prime attribute of R.

In other words, if every non prime attribute is non-transitively dependent on primary key.

43. What is BCNF (Boyce-Codd Normal Form)?

A relation schema R is in BCNF if it is in 3NF and satisfies additional constraints that for every FD X A, X must be a candidate key.

44. What is 4NF?

A relation schema R is said to be in 4NF if for every Multivalued dependency X Y that holds over R, one of following is true

- X is subset or equal to (or) XY = R.
- X is a super key.

45. What is 5NF?

A Relation schema R is said to be 5NF if for every join dependency {R1, R2, ...,Rn} that holds R, one the following is true

- Ri = R for some i.
- The join dependency is implied by the set of FD, over R in which the left side is key of R.

46. What is Domain-Key Normal Form?

A relation is said to be in DKNF if all constraints and dependencies that should hold on the constraint can be enforced by simply enforcing the domain constraint and key constraint on the relation.