2 MARK QUESTIONS

1. What is a relation? What is the difference between a tuple and an attribute?

Ans A relation is table having atomic values, unique rows and unordered rows and columns.

A row in a relation is known as **tuple** whereas a column of a table is known as an **attribute** .

2. What is primary key in a table?

Ans A **Primary Key** is a set of one or more attributes that can be uniquely identify tuples within the relation.

3. What is data redundancy? What are the problems associated with it?

Ans Duplication of data is data redundancy. It leads to the problems like wastage of space and data inconsistency.

4. Define the following terms: (i) **Degree** (ii) **Cardinality**.

Ans (i) **Degree:** The numbers of attributes (columns) in a relation determine the degree of a relation.

(ii) **Cardinality**: The number of tuples (rows) in a relation is called the cardinality of the relation.

5. What are views? How are they useful?

Ans A view is a virtual table that does not really exist in its own right but it instead derived from one and more underlying base table(s). The view is kind of table whose contents are taken upon other tables depending upon a given query condition. No stored file is created to store contents of a view rather its definition is stored only.

The usefulness of views lies in the fact that they provide an excellent way to give people access to some but not all of the information in a table.

6. Differentiate between **Candidate Key** and **Primary Key** in context of RBDMS.

Ans **Candidate Key**. A candidate key is the one that is capable of becoming primary key. i.e., a field or attribute that has unique value for each row in the relation.

Primary Key is a designed attribute or a group of attributes whose values can uniquely identify the tuples in the relation.

7. Differentiate between Candidate key and **Alternate key** in context of RDBMS.

Ans Candidate Key. A candidate key is the one that is capable of becoming primary key i.e., a field or attribute that has unique value for each row in the relation.

A candidate key that is not a primary key is called an Alternate key.

8. Differentiate between **primary key** and **alternate key**.

Ans Primary Key. It is the set of one or more attributes that can uniquely identify tuples within a relation.

Alternate Key. It is a candidate key which is not primary key.

9. What are candidate keys in a table? Give a suitable example of candidate keys in a table.

Ans A candidate key is the one that is capable of becoming primary key i., a field or attribute that has unique value for each row in the relation.

Example Table: ITEM

Ino	**Item**	**Quantity**
101	Pen	560
102	Pencil	340
104	CD	540
10	DVD	200
110	Floppy	400

{ Candidate Keys}

10. Differentiate between Data Definition language and Data Manipulation language.

Ans The SQL DDL provides commands for defining relation schemas, deleting relationship, creating indexes and modifying schemas.

The SQL DML includes a query language to insert, delete and modify tuples in the database.

Data Manipulation Language (DML) is used to put values and manipulate them in tables and other database objects and Data Definition language (DDL) is used to create tables and other database objects.

11. What is the different between WHERE and HAVING clause?

Ans The HAVING clause places conditions on groups in contrast to WHERE clause, which places conditions on individual rows.

---

12. Write the SQL statement to create EMPLOYEE relation which contains EMPNO, Name, Skill, PayRate.

```
Ans CREATE TABLE Employee
        (       EmpNo                   CHAR(4)     NOT NULL PRIMARY KEY,
                Name            CHAR(20)   NOT NULL,
                Skill           CHAR(1),
                PayRate         DECIMAL(8,2));
```

13. Create a table with under mentioned structure (Table name is EMP)

EMPNo	NUMBER(4)	
DeptNo	NUMBER(2)	
EmpName	CHAR(10)	
Job	CHAR(10)	
Manager	NUMBER(4)	
Hiredate	DATE	
Salary	NUMBER(7,2)	>0
Commission	NUMBER(7,2)	

```
Ans CREATE TABLE Emp
        (       EmpNo           Number(4) NOT NULL PRIMARY KEY
                DeptNo          Number(2),
                EmpName         Char(10),
                Job             Char(10),
                Manager         Number(4),
                Hiredate        Date,
                Salary          Decimal(7,2) check (Salary>0),
                Commission      Decimal(7,2) );
```

14. Create a table with the under mentioned structure (Table name is DEPT)

DeptNo	NUMBER(2)
DeptName	CHAR(12)
Location	CHAR(12)

```
Ans CREATE TABLE Dept
        (       DeptNo          NUMBER(2) NOT NULL PRIMARY KEY,
                DeptName        CHAR(12),
                Location        CHAR(12);
```

15. Create a table called PROJECT with the columns specified below.

ProjId	NUMBER(4)	
ProjDesig	CHAR(20)	
ProjStartDT	DATE	
ProjEndDT	DATE	
BudgetAmount	NUMBER(7)	0
MaxNoStaff	NUMBER(2)	

Ans
```
CREATE TABLE Project
        (       ProjId          Number(4) NOT NULL PRIMARY KEY,
                ProjDesig       Char (20) NOT NULL,
                ProjStartDT     Date,
                ProjEndDT       DATE,
                BudgetAmount    Decimal(7,2)        default=0,
                MaxNoStaff      Number(2) );
```

16. Create a table called SALGRADE with the columns specified below:

LowSal	NUMBER(7,2)
HighSal	NUMBER(7,2)
Grade	NUMBER(2)

Ans
```
CREATE TABLE Salgrade
        (LowSal     Decimal(7,2),
        HighSal     DECIMAL(7,2),
        Grade       NUMBER(2) );
```

17. Insert a record with suitable data in the table EMP, having system date as the Hiredate.

Ans Date ( ) function gives the system date.

```
        INSERT INTO Emp
        VALUES (3008, 18, “XAVIER”, “Manager”, Date( ), 3250, NULL);
```

18. Illustrate Cartesian product operation between the two tables/relations using a suitable example.

Ans The two table GABS1 and GABS are as follows:

GAB 1

ROLL NO	**NAME**
1	ABC
2	GABS

GAB 2

MARKS	**SROLL NO**	**AGE**
90	1	19
92	3	17

The certesian product of above two tables is as follows:

Cartesian Product

RollNo	Name	Marks		SRollNo	Age
1	ABC	90	1	19	
1	ABC	92	3	17	
2	GABS	90	1	19	
2	GABS	92	3	17	

19. What is the purpose of key in a table? Give an example of key in a table.

Ans A key is used to identify a tuple uniquely with in the relation. The value of key is unique. No rows in the relation can have same value.

e.g. In an Employee relation EmpCode is a key using EmpCode one can obtain the information of a particular employee.

20. Explain the concept UNION between two tables, with the help of appropriate example.

Ans The UNION operator is used to combine the result-set of two or more tables, without returning any duplicate rows.

e.g.

Table CUSTOMERS

ID	SNAME	CITY
1	A	London
2	B	Berlin
3	C	Mexico

Table SUPPLIER

ID	SNAME	CITY
3	D	Mexico
4	E	London
5	F	UK
6	G	Germany

```
SELECT CITY FROM CUSTOMERS UNION
SELECT CITY FROM SUPPLIER:
```

The resultant table will be:

CITY
London
Berlin
Mexico
UK
Germany

6 MARKS QUESTIONS

1. Note: Write SQL commands for (b) to (e) and write the outputs for (f) on the basis of table GRADUATE.

Table: GRADUATE

S.NO.	NAME	STIPEND	SUBJECT	AVERAGE	DIV
1	KARAN	400	PHYSICS	68	1
2	DIVAKAR	450	COMPUTER SC	68	1
3	DIVYA	300	CHEMISTRY	62	2
4	ARUN	350	PHYSICS	63	1
5	SABINA	500	MATHEMATICS	70	1
6	JOHN	400	CHEMISTRY	55	2
7	ROBERT	250	PHYSICS	64	1
8	RUBINA	450	MATHEMATICS	68	1
9	VIKAS	500	COMPUTER SC	62	1
10.	MOHAN	300	MATHEMATICS	57	2

(a) List the names of those students who have obtained **DIV 1** sorted by **NAME**.

(b) Display a report, listing **NAME, STIPEND, SUBJEZCT** and amount of stipend received in a year assuming that the **STIPEND** is paid every month.

(c) To count the number of students who are either **PHYSICS** or **COMPUTER SC** graduates.

(d) To insert a new row in the **GRADUATE** table:
11, “KAJOL”, 300, “COMPUTER SC”, 75, 1

(e) Give the output of following SQL statement based on table **GRADUATE**:
(I) Select **MIN(AVERAGE)** from **GRADUATE** where **SUBJECT= “PHYSICS”;**
(II) Select **SUM(STIPEND**) from **GRADUATE** where **DIV=2;**
(III) Select **AVG(STIPEND**) from **GRADUATE** where **AVERAGE>=65;**
(IV) Select **COUNT**(distinct **SUBJECT**) from **GRADUATE**;

(f) Assume that there is one more table **GUIDE** in the database as shown below:

Table: GUIDE

MAINAREA	ADVISOR
PHYSICS	VINOD
COMPUTER SC	ALOK
CHEMISTRY	RAJAN
MATHEMATICS	MAHESH

What will be the output of the following query:

```
SELECT      NAME, ADVISOR
FROM        GRADUATE, GUIDE
WHERE       SUBJECT = MAINAREA
```

Ans 1. (a) Select Name From GRADUATE
Where DIV = 1
Order by Name;

(b) Select Name, stipend, subject, stepend * 12
From GRADUATE

(c) Select count (*)
From GRADUATE
Where subject IN (“PHYSICS”, “COMPUTER SC”);

(d) Insert into GRADUATE
Values (11, “KAJOL”, 300, “COMPUTER SC”, 75, 1);

(e) (i) 63 (ii) 1000 (iii) 450 (iv) 4

(f)
```
KARAN       VINOD
DIVAKAR     ALOK
DIVYA       RAJAN
ARUN        VINOD
SABINA      MAHESH
JOHN        RAJAN
ROBERT      VINOD
RUBINA      MAHESH
VIKAS       ALOK
MOHAN       MAHESH
```

2. Write SQL commands for (a) to (d) and write the outputs for (f) on the basis of table CLUB.

Table: CLUB

COACH ID	**COACH NAME**	**AGE**	**SPORTS**	**DATEOFAPP**	**PAY**	**SEX**
1.	KUKREJA	35	KARATE	27/03/1997	1000	M
2.	RAVINA	34	KARATE	20/01/1998	1200	F
3.	KARAN	34	SQUASH	19/02/1998	2000	M

4.	TARUN	33	BASKETBALL	01/01/1998	1500	M
5.	ZUBIN	36	SWIMMING	12/01/1998	750	M
6.	KETAKI	36	SWIMMING	24/02/1998	800	F
7.	ANKITA	39	SQUASH	20/02/1998	2200	F
8.	ZAREEN	37	KARATE	20/02/1998	1100	F
9.	KUSH	41	SWIMMING	13/01/1998	900	M
10.	SHAILYA	37	BASKETBALL	19/02/1998	1700	M

(a) To show all information about the swimming coaches in the club.

(b) To list names of all coaches with their date of appointment **(DATOFAPP)** in descending order.

(c) To display a report, showing coachname, pay, age and bonus (15% of pay) for all the coaches.

(d) To insert in a new row in the **CLUB** table with the following data:
11, “PRAKASH”, 37, “SQUASH”, {25/02/98}, 2500, “M”

(e) Give the output of following SQL statements:
(i) Select COUNT(distinct SPORTS) from CLUB:
(ii) Select MIN(AGE) from CLUB where SEX = “F”;
(iii) Select AVG(PAY) from CLUB where SPORTS = “KARATE”;
(iv) Select SUM(PAY) from CLUB where DATOFAPP > {31/01/98};

(f) Assume that there is one more table **COACHES** in the database as shown below:

Table: COACHES

SPORTS PERSON	SEX	COACH_NO
AJAY	M	1
SEEMA	F	2
VINOD	M	1
TANEJA	F	3

What will be the output of the following query:
SELECT SPORTSPERSON, COACHNAME FROM CLUB, COACHES
WHERE COACH_ID = COACH_NO

Ans 2. (a) Select * From CLUB
Where sports = “SWIMMING”;
(b) Select COACHNAME From CLUB
Order by DATOFAPP desc ;
(c) Select coachname, pay, age, 0.15 * pay From CLUB;
(d) Insert into CLUB values (11, “PRAKASH”, 37, “SQUASH”, {25/02/98], 2500, “M”);
(e) (i) 4 (ii) 34 (iii) 1100 (iv) 7800
(f) AJAY KUKREJA
SEEEMA RAVINA
VINOD KUKREJA
TANEJA KARAN

3. (a) Write SQL commands for (i) to (vii) on the basis of the table SPORTS

Table: SPORTS

Student No.	**Class**	**Name**	**Game1**	**Grade**	**Game2**	**Grade**
10	7	Sammer	Cricket	B	Swimming	A
11	8	Sujit	Tennis	A	Skating	C

12	7	Kamal	Swimming	B	Football	B
13	7	Venna	Tennis	C	Tennis	A
14	9	Archana	Basketball	A	Athletic	C

(i) Display the names of the students who have grade ‘C’ in either Game1 or Game2 or both.
(ii) Display the number of students getting grade ‘A’ in Cricket.
(iii) Display the names of the students who have same game for both Game1 and Game2.
(iv) Display the games taken up by the students, whose name starts with ‘A’.
(v) Add a new column named ‘Marks’.
(vi) Assign a value 200 Marks for all those who are getting grade ‘B’ or grade ‘A’ in both Game1 and Game2.
(vii) Arrange the whole table in the alphabetical order of Name.

(b) Explain Cartesian product of two relations.

Ans (a) Note: In a given table, two fields are having the same name GRADE, which is a mistake in the paper. So, we are assuming these names to be GRADE1 and GRADE2 respectively where GRADE1 pertains to grade of GAME1 and GRADE2 pertains to grade of GAME2.

```
(i)     SELECT Name
                FROM Sports
                WHERE Grade1 = “C” OR
                        Grade2 = “C”;
(ii)    SELECT Count (*)
                FROM Sports
                WHERE (Grade1 = “A”)
                AND Game1 = “Cricket”)
                OR (Grade2 = “A” and Game2 = “Cricket”);
(iii)   SELECT Name
        FROM Sports Game1 = Game2;
        Where Game1 = Game2
(iv)    SELECT Game1, Game2
        FROM Sports
        WHERE Name like “A”;
(v)     ALTER TABLE Student
        ADD Marks float (6, 2);
(vi)    UPDATE Student
        SET Marks = 200
        Where grade1 < = “B” AND
        grad2 < = “B”;
(vii)   SELECT *
        FROM Sports
        ORDER BY Name;
```

(b) The *Cartesian product* is a binary operation and is denoted by a cross(x). The Cartesian product of two relations **A** and **B** is written as **A x B**. The Cartesian product yields a new relation which has (degree number of attributes) equal to the sum of the degrees of the two relations operated upon. The number of tuples (cardinality) of the new relation of the product of the number of tuples of the two relations operated upon. The *Cartesian product* of two relations yields a relation with all possible combinations of the tuples of the two relations operated upon.

4. Given the following Teacher relation: Write SQL commands for question (a) to (f)

No.	Name	Department	Dateofjoining	Salary	Sex
1.	Raja	Computer	21/05/98	80000	M

2.	Sangita	History	21/05/97	9000	F
3.	Ritu	Sociology	29/08/98	8000	F
4.	Kumar	Linguistics	13/06/96	10000	M
5.	Venkatraman	History	31/10/99	8000	M
6.	Sidhu	Computer	21/05/86	14000	M
7.	Aishwarya	Sociology	11/1/98	12000	F

(a) To select all the information of teacher in computer department.
(b) To list the name of the female teacher in History department.
(c) To list all names of teachers with date of admission in ascending order.
(d) To display Teacher's name, Department, and Salary of female teachers.
(e) To count the number of teachers whose salary is less than 10,000.
(f) To insert a new record in the Teachers table with the following data:
8, "Mersa", "Computer", {1/1/2000}, 12000, "M".
(g) Give the output of the following SQL commands:
(i) SELECT MIN(DISTINCT Salary) FROM Teacher
(ii) SELECT MIN(Salary) FROM Teacher WHERE Sex = "M"
(iii) SELECT SUM(Salary) FROM Teacher WHERE Department = "History"
(iv) SELECT ACG(Salary) FROM Teacher WHERE dateofjoining < {1/1/98}.

Ans 4. (a) SELECT * FROM Teacher
WHERE Department = "Computer";
(b) SELECT Name FROM Teacher
WHERE Department = "History" and Sex = "F";
(c) SELECT Name FROM Teacher
ORDERBY Dateofjoining;
(d) SELECT Name, Department, Salary, FROM Teacher
WHERE Sex = "F";
(e) SELECT Count(*), FROM Teacher
WHERE Salary < 10,000;
(f) INSERT into Teacher Values (8, "Mersha", "Computer", {1/1/2000}, 12000, "M");
(g) (i) 8000 (ii) 8000 (iii) 17000 (iv) 11250

5. Given the following tables for a database INTERIORS :
Note: Write SQL command for (a) to (f) and write the outputs for (g) on the basis of tables INTERIORS and NEWONES.

Table: INTERIORS

NO.	ITEMNAME	TYPE	DATEOFSTOCK	PRICE	DISCOUNT
1	Red rose	Double bed	23/02/02	32000	15
2	Soft touch	Baby cot	20/01/02	9000	10
3	Jerry's home	Baby cot	19/02/02	8500	10
4	Rough wood	Office Table	01/01/02	20000	20
5	Comfort zone	Double bed	12/01/02	15000	20
6	Jerry look	Baby cot	24/02/02	7000	19
7	Lion king	Office Table	20/02/02	16000	20
8	Royal tiger	Sofa	22/02/02	30000	25
9	Park sitting	Sofa	13/12/01	9000	15
10	Dine Paradise	Dining Table	19/02/02	11000	15

Table: NEWONES

NO.	ITEMNAME	TYPE	DATEOFSTOCKS	PRICE	DISCOUNT

11	White wood	Double bed	23/03/03	20000	20
12	James 007	Sofa	20/02/03	15000	15
13	Tom look	Baby cot	21/02/13	7000	10

(a) To show all information about the sofas from the **INTERIORS** table.

(b) To list the **ITEMNAME** which are priced at more than 10,000 from the **INTERIORS** table.

(c) To list **ITEMNAME** and TYPE of those items, in which **DATEOFSTOCK** is before 22/01/02 from the **INTERIERS** table in the descending order of **ITEMNAME**.

(d) To display **ITEMNAME** and **DATEOFSTOCK** of those items, in which the discount percentage is more than 15 from **INTERIORS** table.

(e) To count the number of items, whose type is "**Double Bed**" from **INTERIOR** table.

(f) To insert a new row in the **NEWONES** table with the following data:
14, "True Indian", "Office Table",{28/03/03}, 15000,20

(g) Give the output of following SQL statement:

Note: outputs of the below mentioned queries should be based in original data given in both the tables i.e., without considering the insertion done in (f) part of this question.

(i) Select COUNT(distinct TYPE) from INTERIORS;

(ii) Select AVG(DISCOUNT) from INTERIORS, where TYPE = "Baby cot",

(iii) Select SUM(Price) from INTERIORS where DATEOFSTOCK < {12/02/02}.

```
Ans 5. (a)   Select * From INTERIORS Where TYPE = "Sofa";
       (b)   Select ITEMNAME From INTERIORS Where PRICE > 10000;
       (c)   Select ITEMNAME, TYPE From INTERIORS
             Where DATEOFSTOCK < {22/01/02} Order by ITEMNAME;
       (d)   Select ITEMNAME, DATEOFSTOCK From INTERIORS Where DISCOUNT > 15;
       (e)   Select Count (*) From INFERIORS Where TYPE = "Double Bed";
       (f)   Insert into NEWONES Values
                   (14, "True Indian", "Office Table", {28/03/03}, 15000, 20};
       (g)   (i) 5   (ii) 13   (iii)   43000
```

6. Given the following tables for a database FURNITURE :
NOTE: Write SQL command for (a) to (f) and write the outputs for (g) on the bases of tables **FURNITURE** AND **ARRIVALS.**

Table: FURNITURE

NO.	**ITEMNAME**	**TYPE**	**DATEOFSTOCK**	**PRICE**	**DISCOUNT**
1	White lotus	Double Bed	23/02/02	30000	25
2	Pink feather	Baby cot	20//01/02	7000	20
3	Dolphin	Baby cot	19/02/02	9500	20
4	Decent	Office Table	01/01/02	25000	30
5	Comfort zone	Double Bed	12/01/02	25000	25
6	Donald	Baby cot	24/02/02	6500	15
7	Royal Finish	Office Table	20/02/02	18000	30
8	Royal tiger	Sofa	22/02/02	31000	30
9	Econo sitting	Sofa	13/12/01	9500	25
10	Eating paradise	Dining Table	19/02/02	11500	25

Table: ARRIVALS

NO.	**ITEMNAME**	**TYPE**	**DATEOFSTOCK**	**PRICE**	**DISCOUNT**
11	Wood Comfort	Double Bed	23/03/03	25000	25

12	Old Fox	Sofa	20/02/03	17000	20
13	Micky	Baby cot	21/02/02	7500	15

(a) To show all information about the baby cots from the FURNITURE table.

(b) To list the ITEMNAME which are priced at more than 15000 from the FURNITURE table.

(c) To list ITEMNAME AND TYPE of those items, in which DATEOFSTOCK is before 22/01/02 from the FURNITURE table in descending order of ITEMNAME.

(d) To display ITEMNAME and DATEOFSTOCK of those items, in which the DISCOUNT percentage is more than 25 from FURNITURE table.

(e) To count the number of items, whose TYPE is “Sofa” from FURNITURE table.

(f) To insert a new row in the ARRIVALS table with the following data:
14, “Velvet touch”, Double bed”, {25/03/03}, 25000, 30

(g) Give the output of following SQL statement:

Note: outputs of the below mentioned queries should be based on original data given in both the tables i.e., without considering the insertion done in (g) part of this question.

(i) Select COUNT(distinct TYPE) from FURNITURE;
(ii) Select MAX(DISCOUNT) from FURNITURE,ARRIVALS;
(iii) Select AVG(DISCOUNT) from FURNITURE where TYPE = “Baby cot”;
(iv) Select SUM(PRICE) from FURNITURE where DATEOFSTOCK < {12/02/02}.

Ans 6. (a) Select * From FURNITURE Where TYPE = “Baby cot”;
(b) Select ITEMNAME From FURNITURE Where PRICE > 15000;
(c) Select ITEMNAME, TYPE From FURNITURE
Where DATEOFSTOCK < {22/01/02} Order by ITEMNAME;
(d) Select ITEMNAME, DATEOFSTOCK From FURNITURE Where DISCOUNT > 25.
(e) Select Count (*) From FURNITURE Where TYPE = “Sofa”;
(f) Insert Into ARRIVALS Values (14, “Velvet touch”, “Double bed”, {25/03/03}, 25000, 30);
(g) (i) 5 (ii) 30 (iii) 18.33 (iv) 66500.

7. Given the following tables for a database LIBERARY:

Table: Books

Book_Id	**Book_Name**	**Author_Name**	**Publishers**	**Price**	**Type**	**Qty.**
F0001	The Tears	William Hopkins	First Publ.	750	Fiction	10
F0002	Thunderbolts	Anna Roberts	First Publ.	700	Fiction	5
T0001	My First C++	Brian & Brooke	EPB	250	Text	10
T0002	C++ Brainworks	A.W.Rossaine	TDH	325	Text	5
C0001	Fast Cook	Lata Kapoor	EPB	350	Cookery	8

Table: Issued

Book_Id	**Quantity Issued**
F0001	3
T0001	1
C0001	5

Write SQL queries for (a) to (f):

(a) To show Book name, Author name and Price of books of EPB publishers.

(b) To list the names of the books of Fiction Type.

(c) To display the names and price of the books in descending order of their price.

(d) To increase the price of all books of first publisher by 50.

(e) To display the Book_Id, Book_Name and Quantity issued for all books which have been issued. (The query will require contents from both the tables).

(f) To insert a new row in the table Issued following the data: “F0002”,4

(g) Give the output of the following queries based on the above tables:

(i) SELECT COUNT(DISTINCT Publishers) FROM Books.

(ii) SELECT SUM(Price) FROM Books WHERE Quantity > 5.

(iii) SELECT BOOK_NAME, AUTHOR_NAME FROM Books WHERE Price < 500.

(iv) SELECT COUNT (*) FROM Books.

Ans 7. (a)
```
SELECT Book_Name, Author_Name, Price
FROM Books
WHERE Publishers = “EPB”;
```
(b)
```
SELECT Book_Name
FROM Books
WHERE Type = “Fiction”;
```
(c)
```
SELECT Book_Name, Price
FROM Books
ORDER BY Price DESC;
```
(d)
```
UPDATE Book
SET Price = Price + 50
WHERE Publishers = “First Publ.”;
```
(e)
```
SELECT Books.Book_Id, Book_Name, Quantity_Issued
FROM Books, Issued
WHERE books.Book_Id = Issued.Book_Idf;
```
(f)
```
INSERT INTO Issued
VALUES(“F0002”,4);
```
(g) (i) 3 (ii) 1350

(iii)
```
MY First C++          Brain & Brooke
C++ Brainworks     A.W. Rosssaine
Fast Cook            Lata Kapoor
```
(iv) 5

8. Write SQL commands for (a) to (f) and write output for (g) on the basis of Teacher relation given below:

relation Teacher

No.	Name	Age	Department	Date of join	Salary	Sex
1.	Jugal	34	Computer	10/01/97	12000	M
2.	Sharmila	31	History	24/03/98	20000	F
3.	Sandeep	32	Maths	12/12/96	30000	M
4.	Sangeeta	35	History	01/07/99	40000	F
5.	Rakesh	42	Maths	05/09/97	25000	M
6.	Shyam	50	History	27/06/98	30000	M
7.	Shiv Om	44	Computer	25/02/97	21000	M
8.	Shalakha	33	Maths	31/07/97	20000	F

(a) To show all information about the teacher of history department

(b) To list the names of female teacher who are in Hindi department

(c) To list names of all teachers with their date of joining in ascending order.

(d) To display student’s Name, Fee, Age for male teacher only

(e) To count the number of teachers with Age>23.

(f) To inset a new row in the TEACHER table with the following data:

9, “Raja”, 26, “Computer”, {13/05/95}, 2300, “M”

(g) Give the output of following SQL statements:

(i) Select COUNT (distinct department) from TEACHER;
(II) Select MAX (Age) from TEACHER where Sex = “F”
(iii) Select AVG (Salary) from TEACHER where Date of join < {12/07/96};
(v) Select SUM (Salary) from TEACHER where Date of join < {12/07/96};

Ans 8. (a) SELECT * FROM Teacher
WHERE Department = “History”;

(b) SELECT Name FROM Teacher
WHERE Department = “Hindi” and Sex = “F”;

(c) SELECT Name, Dateofjoin
FROM Teacher
ORDER BY Dateofjoin;

(d) (The given query is wrong as no. information about students and fee etc. is available. The query should actually be
To display teacher’s Name, Salary, Age for male teacher only)
SELECT Name, Salary, Age FROM Teacher
WHERE Age > 23 AND Sex = ‘M’;

(e) SELECT COUNT (*) FROM Teacher
WHERE Age > 23;

(f) INSERT INTO Teacher
VALUES (9, “Raja”, 26, “Computer”, {13/05/95}, 2300, “M”);
(g) (i) 3 (ii) 35 (iii) 23600 (AVG (Salary))
(iv) 2300 – after insertion (It is SUM (Salary))

9. Write SQL commands for (a) to (f) and Write the outputs for (g) on the basis of table HOSPITAL

Table: HOSPITAL

No.	Name	Age	Department	Dateofadm	Charges	Sex
1	Arpit	62	Surgery	21/01/98	300	M
2	Zarina	22	ENT	12/12/97	250	F
3	Kareem	32	Orthopedic	19/02/98	200	M
4	Arun	12	Surgery	11/01/98	300	M
5	Zubin	30	ENT	24/02/98	250	M
6	Ketaki	16	ENT	12/01/98	250	M
7	Ankita	29	Cardiology	20/02/98	800	F
8	Zareen	45	Gynecology	22/02/98	300	F
9	Kush	19	Cardiology	13/01/98	800	M
10	Shilpa	23	Nuclear Medicine	21/02/98	400	F

(a) To select all the information of patients of cardiology department.
(b) To list the names of female patients who are in ENT department.
(c) To list name of all patients with their date of admission in ascending order.
(d) To display Patient’s Name, Charges, Age for only female patients.
(e) To count the number of patients with Age<30.
(f) To inset in a new row in the HOSPITAL table with the following data:
11, “Aftab”, 24, “Surgery”, {25/02/98}, 300, “M”
(g) Give the output of following SQL statements:
(i) Select COUNT (DISTINCT charges) from HOSPITAL;
(ii) Select MIN (Age) from HOSPITAL where Sex = “F”
(iii) Select SUM (Charges) from HOSPITAL where Department = “ENT”
(iv) Select AVG (Charges) from HOSPITAL where Datofadm < {12/08/98}

Ans 9. (a) SELECT * FROM Hospital
WHERE Department = “Cardiology;

(b) SELECT Name FROM Hospital

WHERE Department = “ENT” AND Sex = “F”;
(c) SELECT Name, Datofadm FROM Hospital
ORDER BY Datofadm;
(d) SELECT Name, Charges, Age FROM Hospital
WHERE Sex = “F”;
(e) SELECT COUNT (*) FROM Hospital
WHERE Age < 30;
(f) INSERT INTO Hospital
VALUES (11, “Aftab”, 24, “Surgery”, {25/02/98}, 300, “M”;
(g) (i) 5 (ii) 16 (iii) 750 (iv) 340.

10. Answer the questions (a) and (b) on the basis of the following tables **STORE** and **ITEM.**

TABLE STORE

SNo	SName	Area
S01	ABC Computronics	GK II
S02	All Infotech Media	CP
S03	Tech Shoppe	Nehru Place
S04	Geeks Techno Soft	Nehru Place
S05	Hitech Tech Store	CP

TABLE ITEM

INo	IName	Price	SNo
T01	Mother Board	12000	S01
T02	Hard Disk	5000	S01
T03	Keyboard	500	S02
T04	Mouse	300	S01
T05	Mother Board	13000	S02
T06	Keyboard	400	S03
T07	LCD	6000	S04
T08	LCD	5500	S05
T09	Mouse	350	S05
T10	Hard Disk	4500	S03

(a) Write the SQL queries (i) to (iv):
(i) To display IName and Price of all the items in ascending order of their Price.
(ii) To display SNo and SName of all store location in CP.
(iii) To display Minimum and maximum Price of each IName from the table ITEM.
(iv) To display IName, Price of all items and their respective SName where they are available.
(b) Write the output of the following SQL commands (i) to (iv):
(i) SELECT DISTINCT IName FROM ITEM
WHERE Price >=5000;
(ii) SELECT Area, COUNT (*)
FROM STORE GROUP BY Area;
(iii) SELECT COUNT (DISTINCT Area)
FROM STORE:
(iv) SELECT IName, Price * 0.05
DISCOUNT FROM ITEM
WHERE SNo IN (‘S02’, ‘S03’);

Ans 10. (a) (i) SELECT IName, Price
FROM ITEM
ORDER BY Price ASC;

(ii)

```
SELECT IName
FROM STORE
WHERE Area = ‘CP’;
```

(iii)

```
SELECT IName,
MIN (Price) “Minimum Price”,
MAZ (Price) “Maximum Price”
FROM ITEM
GROUP BY IName;
```

(iv)

```
SELECT IName, Price, SName
FROM ITEM I, STORE S
WHERE I, SNo = S.No
```

(b) (i)

INAME
Mother Board
Hard Disk
LCD

(ii)

AREA	**COUNT(*)**
GK II	1
CP	2
Nehru place	2

(iii)

Count (DISTINCT Area)
3

(iv)

INAME	**DISCOUNT**
Keyboard	25
Mother Board	650
Keyboard	20
Hard Disk	225

11. Answer the questions (a) and (b) on the basis of the following tables SHOPPE and ACCESSORIES.

TABLE SHOP

ID	SName	Area
S0001	ABC Computeronics	CP
S0002	All Infotech Media	GK II
S0003	Tech Shoppe	CP
S0004	Greeks Techno Soft	Nehru Place
S0005	Hitech Tech Store	Nehru Place

TABLE ACCESSORIES

No	Name	Price	ID
A01	Mother Board	12000	S01
A02	Hard Disk	5000	S01
A03	Keyboard	500	S02
A04	Mouse	300	S01
A05	Mother Board	13000	S02
A06	Keyboard	400	S03
A07	LCD	6000	S04
T08	LCD	5500	S05

T09	Mouse	350	S05
T10	Hard Disk	4500	S03

(a) Write the SQL queries:

(i) To display Name and Price of all the accessories in ascending order of their Price.

(ii) To display Id and SName of all Shoppe in Nehru Place.

(iii) To display Minimum and Maximum Price of each Name of accessories.

(iv) To display Name, Price of all accessories and their respective SName where they are available.

(b) (i) SELECT DISTINCT Name FROM ACCESSORIES WHERE Price>=500;

(ii) SELECT Area, COUNT (*) FROM GROUP BY Area;

(iii) SELECT COUNT (DISTINCT Area) FROM SHOPPE;

(iv) SELECT Name, Price*0.05 DISCOUNT FROM ACCESSORIES WHERE SNo IN (‘S02, ‘S03’);

Ans 11. (a) (i)

```
SELECT Name, Price
FROM ACCESSORIES
ORDER BY Price ASC;
```

(ii)

```
SELECT ID, Price
FROM SHOPPE
WHERE Area = ‘Nehru Place’;
```

(iii)

```
SELECT MIN (Price) “Minimum Price”,
MAX (Price) “Maximum Price”,
Name
FROM ACCESSORIES
GROUP BY Name;
```

(iv)

```
SELECT Name, Price, SName
FROM ACCESSORIES A. SHOPPE S
WHERE A. ID = S. ID
```

(b) (i)

NAME
Mother Board
Hard Disk
LCD

(ii)

AREA	COUNT(*)
CP	2
GK II	1
Nehru Place	2

(iii)

COUNT (DISTINCT Area)
3

(iv) The given query will result in an error as there is no column named SNo in Accessories table.

12. Write SQL queries for (a) to (f) and write the outputs for the SQL queries mentioned shown in (g$_{1)}$ to (g$_4$) parts on the basis of tables PRODUCTS AND SUPPLIERS

TABLE PRODUCTS

PID	SNAME	QTY	PRICE	COMPANY	SUPCODE
101	DIGITAL CAMERA14X	120	12000	RENIX	S01
102	DIGITAL PAD lli	100	22000	DIGI POP	S02
104	PEN DRIVE 16 GB	500	1100	STOREKING	S01
106	LED SCREEN	70	28000	DISEXPERTS	S02

105	CAR GPS SYSSTEM	60	12000	MOVEON	S03

TABLE SUPPLIERS

SUPCODE	SNAME	CITY
S01	GET ALL INC	KOLKATA
S03	EASY MARKET CORP	DELHI
S02	DIGI BUSY GROUP	CHENNAI

(a) To display the details of all the products in ascending order of product names (i.e. PNAME).

(b) To display product name and price of all those products, whose price is in the range of 10000 and 15000 (both values inclusive).

(c) To display the number of products which are supplied by each supplier i.e. the expected output should be

S01	2
S02	2
S03	1

(d) To display the price, product name (i.e. PName) and quantity (i.e. QTY) of those which have quantity more than 100.

(e) To display the names of those suppliers, who are either from DELHI or from CHENNAI.

(f) To display the name of the companies and the name of the products in descending order of company names.

(g) Obtain the outputs of the following SQL queries based on the data given in tables PRODUCTS and SUPPLIERS.

(g$_1$) SELECT DISTINCT SUPCODE FROM PRODUCTS;

(g$_2$) SELECT MAX (PRICE), MIN (PRICE) FROM PRODUCTS;

(g$_3$) SELECT PRICE * QTY AMOUNT FROM PRODUCTS WHERE PID = 104;

(g$_4$) SELECT PNAME, SNAME FROM PRODUCTS P, SUPPLIERS S
WHERE P. SUPCODE = S. SUPCODE AND QTY>100;

```
Ans 12.     (a)     SELECT *
            FROM PRODUCTS
            ORDER BY NAME;
        (b)     SELECT PNAME’ PRICE
            FROM PRODUCTS
                    WHERE PRICE BETWEEN 10000 AND 15000;
        (c)     SELECT SUPCODE, COUNT (*)
                    FROM PRODUCTS
                    GROUP BY SUPCODE;
            (d)     SELECT PRICE, PNAME, QTY
                    FROM PRODUCTS
                    WHERE QTY > 100;
        (e)     SELECT SNAME
                    FROM SUPPLIERS
                    WHERE CITY IN (‘DELHI’, CHENNAI’);
        (f)     SELECT COMPANY, PNAME
           FROM PRODUCTS
            ORDER BY COMPANY DESC;
```

(g) (g$_1$)

SUPCODE
S01
S02
S03

(g$_2$)

MAX (PRICE)	**MIN** (PRICE)
28000	1100

(g$_3$)

AMOUNT
550000

(g$_4$)

PNAME	**SNAME**
DIGITAL CAMERA 14X	GET ALL INC
PEN DRIVE 16 GB	GET ALL INC

13. Write SQL queries for (a) to (f) and write the outputs for the SQL queries mentioned shown in (g$_1$) to (g$_4$) parts on the basis of tables ITEMS and TRADERS.

TABLE ITEMS

CODE	**INAME**	**QTY**	**PRICE**	**COMPANY**	**TCODE**
1001	DIGITAL PAD12i	120	11000	XENITA	T01
1006	LED SCREEN 40	70	38000	SANTORA	T02
1004	CAR GPS SYSTEM	50	21500	GEOKNOW	T01
1003	DIGITAL CAMERA	160	8000	DIGICLICK	T02
1005	PEN DRIVE 32 GB	600	1200	STOREHOME	T03

TABLE TRADERS

TCODE	**TNAME**	**CITY**
T01	ELECTRONIC SALES	MUMBAI
T03	BUSY STORE CORP	DELHI
T02	DISP HOUSE INC	CHENNAI

(a) To display the details of all the items in ascending order of item names (i.e. INAME).

(b) To display item name and price of all those items, whose price is in the range of 10000 and 22000 (both values inclusive).

(c) To display the number of items, which are traded by each trader. The expected output of this query should be

T01	2
T02	2
T03	1

(d) To display the price, item name (i.e. INAME) and quantity (i.e. QTY) of those items which have quantity more than 150.

(e) To display the names of those traders, who are either from DELHI or from MUMBAI.

(f) To display the name of the companies and the bane of the items in descending order of company names.

(g) Obtain the outputs of the following SQL queries based on the data given in tables ITEMS and TRADERS.

(g$_1$) SELECT MAX (PRICE), MIN (PRICE) FROM ITEMS;

(g$_2$) SELECT PRICE * QTY AMOUNT FROM ITEMS WHERE CODE = 1004;

(g$_3$) SELE CT DISTINCT TCODE FROM ITEMS;

(g$_4$) SELECT INAME, TNAME FROM ITEMS I, TRASERS T
WHERE I, TCODE AND QTY<100;

Ans 13. (a)

```
SELECT *
FROM ITEMS ORDER BY INAME;
```

(b)

```
SELECT INAME, PRICE
```

```
        FROM ITEMS
        WHERE PRICE BETWEEN 10000
        AND 22000;
(c)     SELECT TCODE, COUNT (*)
        FROM ITEMS
        GROUP BY TCODE;
(d)     SELECT PRICE, INAME, QTY
        FROM ITEMS
        WHERE QTY >150;
(e)     SELECT TNAME
        FROM TRADERS
        WHERE CITY = ‘MUMBAI’ OR CITY =’DELHI’;
(f)     SELECT COMPANY, INAME
        FROM ITEMS
        OREDER BY COMPANY DESC;
```

(g) (g_1)

```
MAX (PRICE)          MIN (PRICE)
  38000              1200
```

(g_2)

```
AMOUNT
1075000
```

(g_3)

```
TCODE
 T01
 T02
 T03
```

(g_4)

```
   INAME              TNAME
LED SCREEN 40      DISP HOUSE INC
CAR GPS SYSTEM     ELECTRONIC SALES
```

14. Write SQL queries for (a) to (f) and write the outputs for the SQL queries mentioned shown in (g_1) to (g_4) parts on the basis of tables APPLICANTS and COURSES.

TABLE APPLICANTS

NO	**NAME**	**FEE**	**GENDER**	**C_ID**	**JOINYEAR**
1012	Amandeep	30000	M	A01	2012
1102	Avisha	25000	F	A02	2009
1103	Ekant	30000	M	A02	2011
1049	Arun	30000	M	A03	2009
1025	Amber	40000	M	A02	2011
1106	Ela	40000	F	A05	2010
1017	Nikita	35000	F	A03	2012
1108	Arleena	30000	F	A03	2012
2109	Shakti	35000	M	A04	2011
1101	Kirat	25000	M	A01	2012

TABLE COURSES

C_ID	**COURSE**
A01	FASHION DESIGN
A02	NETWORKING
A03	HOTEL MANAGEMENT

A04	EVENT MANAGEMENT
A05	OFFICE MANAGEMENT

(a) To display name, fee, gender, joinyear about the applicants, who have joined before 2010.

(b) To display the names of applicants, who are paying fee more than 30000.

(c) To display name of all applicants in ascending order of their joinyear.

(d) To display the year and the total number of applicants joined in each YEAR from the table APPLICANTS.

(e) To display the C_ID (i.e. Course ID) and the number of applicants registered in the course from the APPLICANTS table.

(f) To display the applicant's name with their respective course's name from the tables APPLICANTS and COURSES.

(g) Give the output of following SQL statements:

(g$_1$) SELECT NAME, JOIN YEAR FROM APPLICANTS WHERE GENDER= 'F' AND C_ID= '02';

(g$_2$) SELECT MIN (JOINYEAR) FROM APPLICANTS WHERE Gender= 'M';

(g$_3$) SELE CT AVG (FEE) FROM APPLICANTS WHERE C_ID= 'A01' OR C_ID= 'A05';

(g$_4$) SELECT SUM (FEE), C_ID FROM APPLICATIONS GROUP BY C_ID HAVING COUNT (*) =2;

Ans 14.

(a)
```
SELECT NAME, FEE, GENDER, JOINYEAR
FROM APPLICANTS
WHERE JOINYEAR<2010;
```

(b)
```
SELECT NAME
FROM APPLICANTS
WHERE FEE >30000;
```

(c)
```
SELECT NAME
FROM APPLICANTS
ORDER BY JOINYEAR;
```

(d)
```
SELECT JOINYEAR, COUNT (*)
FROM APPLICANTS
GROUP BY JOINYEAR
```

(e)
```
SELECT C_ID, COUNT (*)
FROM APPLICANTS
ORDER BY C_ID;
```

(f)
```
SELECT NAME, COURSE
FROM APPLICANTS, COURSES
WHERE APPLICANTS. C_ID=COURSES.C_ID;
```

(g) (g$_1$)

NAME	**JOINYEAR**
Avisha	2009

(g$_2$)

MIN (JOINYEAR)
2009

(g$_3$)

AVG(FEE)
31666.666

(g$_4$)

SUM(FEE)	**C_ID**
55000	A01

15. Consider the following tables CABHUB and CUSTOMER and answer (a) and (b) parts of this question:

TABLE CABHUB

Vcode	**VehicleName**	**Make**	**Color**	**Capacity**	**Charges**
100	Innova	Toyota	WHITE	7	15
102	SX4	Suzuki	BLUE	4	14
104	C-Class	Mercedes	RED	4	35
105	A-Star	Suzuki	WHITE	3	14
108	Indigo	Tata	SILVER	3	12

TABLE CUSTOMER

Code	**CName**	**VCode**
1	Hemant Sahu	101
2	Raj Lal	108
3	Feroza Shah	105
4	Ketan Dhal	104

(a) Write SQL commands for the following statements:
(i) To display the names of all the white colored vehicles.
(ii) To display name of vehicle, make the capacity of vehicles in ascending order of their sitting Capacity.
(iii) To display the highest charges at which a vehicle can be hired from CABHUB.
(iv) To display the customer and the corresponding name of the vehicle hired by them.
(b) (i) SELECT COUNT (DISTINCT Make) FROM CABHUB;
(ii) SELECT MAX (CHARGES), MIN (Charges) FROM CABHUB;
(iii) SELECT COUNT (*), Make FROM CABHUB;
(iv) SELECT VehicleName FROM CABHUB WHERE Capacity = 4;

Ans 15. (a) (i) SELECT VehicleName
FROM CABHUB WHERE Color = ‘WHITE’;
(ii) SELECT VehicleName, Make,
Capacity FROM CABHUB
ORDER BY Capacity;
(iii) SELECT MAX (Charges)
FROM CABHUB;
(iv) SELECT CName, VehicleName
FROM CABHUB, CUSTOMER
WHERE CABHUB, Vcode = CUSTOMER, Vcode;

(b) (i)

COUNT (DISTINCT Make)
4

(ii)

MAX (Charges)	**MIN** (Charges)
35	12

(iii) This query will execute but count (*) will result one row and Make will give more than one row so both are not compatible together. But on removing Make from select clause it will give following result.

COUNT (*)
5

(iv)

VehicleName

SX4
C-Class

16. Consider the following tables CARDEN and CUSTOMER and answer (a) and (b) parts of this question:

TABLE CARDEN

Ccode	**CarName**	**Make**	**Color**	**Capacity**	**Charges**
501	A-star	Suzuki	RED	3	14
503	Indigo	Tata	SILVER	3	12
502	Innova	Toyota	WHITE	7	15
509	SX4	Suzuki	SILVER	4	14
510	C-Class	Mercedes	RED	4	35

TABLE CUSTOMER

CCode	**Cname**	**Ccode**
1001	Hamant Sahu	501
1002	Raj Lal	509
1003	Feroja Shah	503
1004	Ketan Dhal	502

(a) Write SQL commands for the following statements:

(i) To display the name of all the SILVER colored cars.

(ii) To display name of car, make and capacity of cars in descending order of their sitting capacity.

(iii) To display the highest Charges at which a vehicle can be hired from CARDEN.

(iv) To display the customer name and the corresponding name of the cards hired by them.

(b) Give the output of the following SQL queries:

(i) SELECT COUNT (DISTINCT Make) FROM CARDEN;

(ii) SELECT MAX (Charges), MIN (Charges) FROM CARDEN;

(iii) SELECT COUNT (*), Make FROM CARDEN;

(iv) SELECT CarName FROM CARDEN WHERE Capacity = 4;

```
Ans 16.  (a)   (i)  SELECT CarName
                  FROM CARDEN
                  WHERE Color = 'SILVER';
           (ii)  SELECT CarName, Make, Capacity
                  FROM CARDEN;
                  ORDER BY Capacity DESC;
           (iii)  SELECT MAX (Charges)
                   FROM CARDEN;
           (iv)  SELECT CName, CarName
                   FROM CARDEN,     CUSTOMER
                   WHERE CARDEN.Ccode = CUSTOMER.Ccode;
```

(b) (i)

COUNT (DISTINCT Make)
4

(ii)

MAX (Charges)	**MIN** (Charges)
35	12

(iii) This query will execute but count (*) will result one row and Make will give more than one row so both are not compatible together. But on removing Make from select clause it will give compatible result:

COUNT (*)
5

(iv)

CarName
SX4
C-Class

17. Consider the following tables EMPLOYEE and SALGRADE and answer (a) and (b) parts of this question:

TABLE EMPLOYEE

ECODE	NAME	DESIG	SGRADE	DOJ	DOB
101	Abdul Ahmad	EXECUTIVE	S03	23-MARCH-2003	13-JAN-1980
102	Ravi Chander	HEAD-IT	S02	12-FEB-2010	22-JUL-1987
103	John Ken	Receptionist	S03	24-JUN-2009	24-FEB-1983
105	Nazar Ameen	GM	S02	11-AUG-2006	03-MAR-1984
108	Priyam Sen	CEO	S01	29-DEC-2004	19-JAN-1982

TABLE SALGRADE

SGRADE	SALARY	HRA
S01	56000	18000
S02	32000	12000
S03	24000	8000

(a) Write SQL commands for the following statements:

(i) To display the detail of all the EMPLOYEE in descending order of DOJ.

(ii) To display name and design of those EMPLOYEE, whose sgrade is either S02 or S03.

(iii) To display the content of all the EMPLOYEE table, whose DOJ is in between '09-FEB-2006' and '08-AUG-2009'.

(iv) TO add a new row in the EMPLOYEE table with the following data:
109, 'Harish Roy', 'HEAD-IT', 'S02', '09-SEP-2007', '21-APR-1983'.

(b) Give the output of the following SQL queries:

(i) SELECT COUNT (SGRADE), SGRADE FROM EMPLOYEE GROUP BY SGRADE;

(ii) SELECT MIN (DOB), MAX (DOJ) FROM EMPLOYEE;

(iii) SELECT NAME, SALARY FROM EMPLOYEE E, SALGRADE S
WHERE E. SGRADE = S. SGRADE AND E. ECODE<103;

(iv) SELECT SGRADE, SALARY+HRA FROM SALGRADE WHERE SGRADE = 'S02';

```
Ans 17.  (a)  (i)  SELECT  *
              FROM EMPLOYEE ORDER BY DOJ DESC;
         (ii) SELECT NAME, DESIG
              FROM EMPLOYEE
              WHERE SGRADE= 'S02'
                                 OR SGRADE = 'S03';
         (iii) SELECT *
              FROM EMPLOYEE
              WHERE DOJ BETWEEN   '09-FEB-2006'
              AND '08-AUG-2009';
          (iv) INSERT INTO EMPLOYEE VALUES
```

(109, ‘Harish Roy’, ‘HEAD-IT’, ‘S02’,
’09-SEP-2007’, ’21-APR-1983’);

(b) (i)

COUNT (SGRADE)	**SGRADE**
1	S01
2	S02
3	S03

(ii)

MIN(DOB)	**MAX**(DOJ)
13-JAN-1980	12-FEB-2010

(iii)

NAME	**SALARY**
Abdul Ahmad	24000
Ravi Chander	32000

(iv)

SGRADE	**SALARY+HRA**
S02	44000

18. Consider the following tables WORKER and PAYLAVEL and answer (a) and (b) parts of this question:

TABLE WORKER

ECODE	**NAME**	**DESIGN**	**PLEVEL**	**DOJ**	**DOB**
11	Radhe Shyam	Supervisor	P001	13-SEP-2004	23-AUG-1981
12	Chander Nath	Operator	P003	22-FEB-2010	12-JUL-1987
13	Fizza	Operator	P003	14-JUN-2009	14-OCT-1983
15	Ahmeen Ahmad	Mechanic	P002	21-AUG-2006	13-MAR-1984
18	Sanya	Clerk	P002	19-DEC-2005	09-JUN-1983

TABLE PAYLEVEL

PLEVEL	**PAY**	**ALLOWANCE**
P001	26000	12000
P002	22000	10000
P003	12000	6000

(a) Write SQL commands for the following statements:

(i) To display the detail of all WORKER in descending order of DOB.

(ii) To display name and design of those WORKER, whose plevel is either P001 to P002.

(iii) To display the content of all the WORKER table, whose DOB is in between ‘19-JAN-1984’ and ‘18-JAN-1987’.

(iv) To add a new row with the following:
19, ‘Daya Kishore’, ‘Operator’, ‘P003’, ’19-JUN-2008’, ’11-JUL-1984’.

(b) Give the output of the following SQL queries:

(i) SELECT COUNT (PLEVEL), PLEVEL FROM WORKER GROUP BY PLEVEL;

(ii) SELECT MAX (DOB), MIN (DOJ) FROM WORKER;

(iii) SELECT NAME, PAY FROM WORKER W, PAYLEVEL P
WHERE W. PLEVEL= P.LEVEL AND W. ECODE<13;

(iv) SELECT PLEVEL, PAY+ALLOWANCE FROM PLEVEL WHERE PLEVEL = ‘P003’;

Ans 18. (a) (i) SELECT *
FROM WORKER
ORDER BY DOB DESC;

```
(ii)SELECT NAME,  DESIG;
    FROM WORKER
    WHERE PLEVEL = ‘P001’ OR PLEVEL = ‘P002’;
(iii) SELECT *
    FROM WORKER
    WHERE DOB BETWEEN
    ’19-JAN-1984’ AND ’18-JAN-1987’;
(iv) INSERT INTO WORKER VALUES (19,
    ‘Daya Kishore’, Operator’, ‘P003’
      ’19-JUN-2008’, ’11-JUL-1984’);
```

(b) (i)

COUNT (PLEVEL)	**PLEVEL**	
	1	P001
	2	P002
	3	P003

(ii)

MAX (DOB)	**MIN** (DOJ)
12-JUL-1987	13-SEP-2004

(iii)

NAME	**PAY**
Radhe Shyam	26000
Chander Nath	12000

(iv)

PLEVEL	**PAY+ALLOWANCE**
P003	18000

19. Consider the following tables STORE and SUPPLIERS and answer (a) and (b) parts of this question:

TABLE STORE

iteemNo	**Item**	**Scode**	**Qty**	**Rate**	**LastBuy**
2005	Sharpener Classic	23	60	8	31-JUN-09
2003	Ball pen 0.25	22	50	25	01-FEB-09
2002	Gel Pen Premium	21	150	12	24-FEB-10
2006	Gel Pen Classic	21	250	20	11-MAY-09
2001	Eraser Small	22	220	6	19-JAN-09
2004	Eraser Big	22	110	8	02-DEC-09
2009	Ball Pen 0.5	21	180	18	03-NOV-09

TABLE SUPPLIERS

Scode	**Sname**
21	Premium Stationers
23	Soft Plastics
22	Tetra Supply

(a) Write SQL commands for the following statements:

(i) To display details of all the items in the Store table in ascending order of LastBuy.

(ii) To display ItemNo and Item name of those items from STORE table whose Rate is more than 15 Rupees.

(iii) To display the details of those items whose Supplier code (Scode) is 22 or Quantity in Store (Qty) is more than 110 from the table STORE.

(iv) To display minimum Rate of items for each supplier individually as per Scode from the table STORE.

(b) Give the output of the following SQL queries:

(i) SELECT COUNT (DISTINCT Scode) FROM STORE;

(ii) SELECT Rate * Qty FROM STORE WHERE ItemNo = 2004;

(iii) SELECT Item, Sname FROM Store S, SUPPLIERS P
WHERE S. Scode = P. Scode AND ItemNo = 2006;

(iv) SELECT MAX (LastBuy) FROM STORE;

Ans 19.(a) (i) SELECT *
FROM STORE ORDER BY LastBuy;

(ii) SELECT itemNo. Item
FROM STORE WHERE Rate>15;

(iii) SELECT * FROM STORE
WHERE Scode = 22 OR Qty >110;

(iv) SELECT MIN (Rate)
FROM STORE GROUP BY Scode;

(b)

(i)

COUNT (DISTINCT Scode)
3

(ii)

Rate * Qty
880

(iii)

Item	**Sname**
Gel Pen Classic	Premier Stationers

(iv)

MAX (Lastbuy)
24-FEB-10

20. Consider the following table GARMENT and FABRIC, Write SQL commands for the statements (i) to (iv) and give outputs for the SQL queries (v) to (viii).

TABLE GARMENT

GCODE	DESCRIPTION	PRICE	FCODE	READYDATE
10023	PENCIL SKIRT	1150	F 03	19-DEC-08
10001	FORMAL SHIRT	1250	F 01	12-JAN-08
10012	INFORMAL SHIRT	1550	F 02	06-JUN-08
10024	BABY TOP	750	F 03	07-APR-07
10090	TULIP SKIRT	850	F 02	31-MAR-07
10019	EVENING GOWN	850	F 03	06-JUN-08
10009	INFORMAL PANT	1500	F 02	20-OCT-08
10007	FORMAL PANT	1350	F 01	09-MAR-08
10020	FROCK	850	F 04	09-SEP-07
10089	SLACKS	750	F 03	20-OCT-08

TABLE FABRIC

FCODE	TYPE
F 04	POLYSTER
F 02	COTTON
F 03	SILK
F01	TERELENE

(i) To display GCODE and DESCRIPTION of each GARMENT in descending order of GCODE.

(ii) To display the details of all the GARMENT, which have READYDATE in between 08-DEC-07 and 16-JUN-08 (inclusive if both the dates).

(iii) To display the average PRICE of all the GARMENT, which are made up of fabric with FCODE as F03.

(iv) To display fabric wise highest and lowest price of GARMENT from GARMENT table. (Display

FCODE of each GARMENT along with highest and lowest Price).

(v) SELECT SUM (PRICE) FROM GARMENT WHERE FCODE = 'F01';

(vi) SELECT DESCRIPTION, TYPE FROM GARMENT, FABRIC
WHERE GARMENT, FCODE = FABRIC.FCODE AND GARMENT.PRICE >=1260;

(vii) SELECT MAX (FCODE) FROM FABRIC;

(viii) SELECT COUNT (DISTINCT PRICE) FROM GARMENT;

Ans 20. (i) SELECT GCODE, DESCRIPTION
FROM GARMENT ORDER BY GCODE DESC;

(ii) SELECT * FROM GARMENT
WHERE READY DATE BETWEEN '08-DEC-07'
AND '16-JUN-08';

(iii) SELECT AVG (PRICE)
FROM GARMENT WHERE FCODE = 'F03';

(iv) SELECT FCODE, MAX (PRICE), MIN (PRICE)
FROM GARMENT GROUP BY FCODE;

(v)

SUM (PRICE)
2600

(vi)

DESCRIPTION	**TYPE**
INFORMAL SHIRT	COTTON
INFORMAL PANT	COTTON
FORMAL PANT	TERELENE

(vii)

MAX (FCODE)
F04

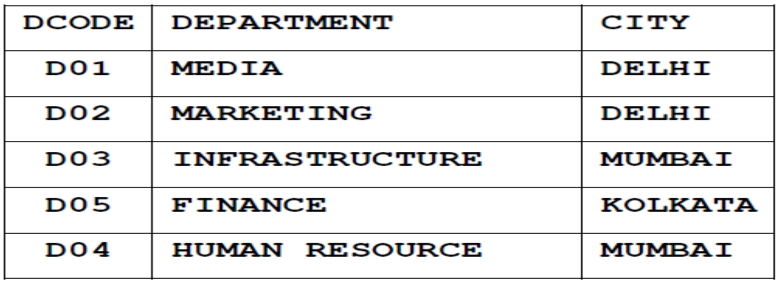
(vii)

COUNT (DISTINCT PRICE)
7

21.

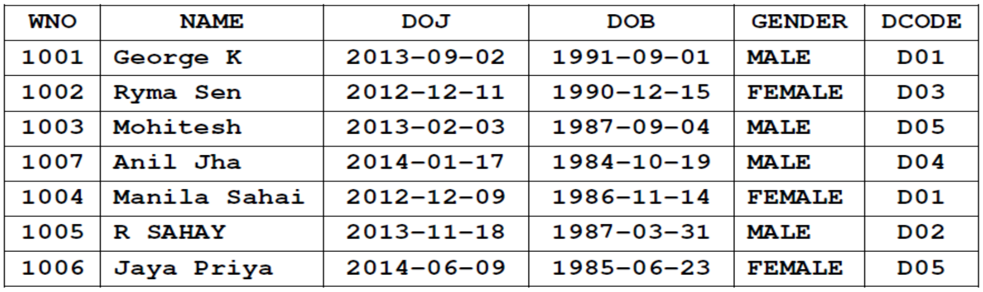
Consider the following DEPT and WORKER tables. Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii) :

Table : DEPT



DCODE	DEPARTMENT	CITY
D01	MEDIA	DELHI
D02	MARKETING	DELHI
D03	INFRASTRUCTURE	MUMBAI
D05	FINANCE	KOLKATA
D04	HUMAN RESOURCE	MUMBAI

Table : WORKER



WNO	NAME	DOJ	DOB	GENDER	DCODE
1001	George K	2013-09-02	1991-09-01	MALE	D01
1002	Ryma Sen	2012-12-11	1990-12-15	FEMALE	D03
1003	Mohitesh	2013-02-03	1987-09-04	MALE	D05
1007	Anil Jha	2014-01-17	1984-10-19	MALE	D04
1004	Manila Sahai	2012-12-09	1986-11-14	FEMALE	D01
1005	R SAHAY	2013-11-18	1987-03-31	MALE	D02
1006	Jaya Priya	2014-06-09	1985-06-23	FEMALE	D05

Note : DOJ refers to date of joining and DOB refers to date of Birth of workers.

(i) To display Wno, Name, Gender from the table WORKER in descending order of Wno.
(ii) To display the Name of all the FEMALE workers from the table WORKER.
(iii) To display the Wno and Name of those workers from the table WORKER who are born between ‘1987-01-01’ and ‘1991-12-01’.
(iv) To count and display MALE workers who have joined after ‘1986-01-01’.
(v) **SELECT COUNT(*), DCODE FROM WORKER GROUP BY DCODE HAVING COUNT(*)>1;**
(vi) **SELECT DISTINCT DEPARTMENT FROM DEPT;**
(vii) **SELECT NAME, DEPARTMENT, CITY FROM WORKER W,DEPT D WHERE W.DCODE=D.DCODE AND WNO<1003;**
(viii) **SELECT MAX(DOJ), MIN(DOB) FROM WORKER;**

Ans 21.

```
(i)   SELECT Wno,Name,Gender FROM Worker ORDER BY Wno DESC;
(ii)  SELECT Name FROM Worker WHERE Gender=’FEMALE’;
(iii) SELECT Wno, Name FROM Worker
WHERE DOB BETWEEN ‘ 19870101’ AND ‘ 19911201’;
(iv)  SELECT COUNT(*) FROM Worker
WHERE GENDER=’MALE’ AND DOJ > ‘19860101’;

(v) COUNT(*) DCODE
     2         D01
     2         D05

(vi) Department
MEDIA
MARKETING
INFRASTRUCTURE
FINANCE
HUMAN RESOURCE

(vii) NAME      DEPARTMENT     CITY
     George K  MEDIA          DELHI
     Ryma Sen  INFRASTRUCTURE MUMBAI

(viii)
          MAX(DOJ)       MIN(DOB)
          2014-06-09     1984-10-19
```