

- SQL statements are divided into four sub languages:
- Data Definition Language (DDL)** It is used to define the structure of tables in the database. It contains the necessary statements to CREATE, RENAME, ALTER and DROP the tables.
 - Data Manipulation Language (DML)** It is used to manipulate the data in the database. It contains statements to UPDATE, DELETE, INSERT and SELECT data that is stored in the database.
 - Data Control Language (DCL)** It is used to control data stored in the database. It contains statements give permission to access the data in the database. These statements are GRANT and REVOKE.
 - Transaction Control Language (TCL)** It is used to control the transactions in a database system. It contains statements like COMMIT, ROLLBACK and SAVEPOINT.

Rules for SQL Commands/Statements

Rules for SQL commands are given below:

- SQL statements are not case sensitive.
- SQL statements can be executed on one or more tables.
- Keywords cannot be abbreviated.
- The statement can be typed in single line or multiple lines.
- Place a semicolon at the end of the SQL statements.
- A comma (,) is used to separate parameters without a clause.
- Characters and date constants or literals must be enclosed in single quotes('').

SQL Data Types

Data types are declared to identify the type of data that will be hold or stored for a particular column or variable. Some SQL data types are as follows:

Data Type	Description
1. CHAR(n) or VARCHAR(n) or VARCHAR 2 (n)	Stores character string of n characters.
2. NUMBER (n, d)	Where n is a precision value with maximum precision of 38 digit and d is a scale value.
3. DECIMAL (p,s)	Where p is a precision value and s is a scale value.
4. DATE	Contain valid date.
5. TIME	Contain valid time.

DDL Commands/Statements in SQL

Some DDL statements are as follows:

- (i) **Create table** A table or relation can be created in a database by CREATE TABLE statement of SQL.

e.g.

```
CREATE TABLE <table_name>
(  <attribute_name> <datatype> [size] [constraint],
  <attribute_name> <datatype> [size] [constraint],
  :
  :
  );
```

Constraints are defined to enforce certain rules, either within a table or between two tables.

Type	Description
NULL/NOT NULL	Specifies, if a column can or cannot have NULL values.
UNIQUE	Values of a column have to be unique.
PRIMARY KEY	Defines it to be a primary key.
DEFAULT	It inserts default value into a column.
CHECK	Explicitly defines a condition that each row must satisfy.
FOREIGN	It designates a column as a foreign key.

- (ii) **Alter table** This command is used to add, delete or modify columns and constraints in an existing table. It is also used to rename a column name of a table.

To ADD a Column ALTER TABLE <table_name> ADD <column_name> <data_type>;

To DROP a Column ALTER TABLE <table_name> DROP COLUMN <column_name>;

To MODIFY Column Data Type ALTER TABLE <table_name> MODIFY <column_name> <datatype>;

To DELETE a Constraint ALTER TABLE <table_name> DROP <constraint_name>;

To RENAME a Column ALTER TABLE <table_name> RENAME COLUMN <old_name> To <new_name>;

- (iii) **Drop table** To remove the table definition and all data, constraints and permissions specified for that table.

DROP TABLE <table_name>;

DML Commands/Statements in SQL

Some DML statements are as follows

- (i) **INSERT** To add new records into the table or to the end of an existing table.

INSERT INTO <table_name>(<col_name1>,<col_name2>, ...)VALUES (<value1>,<value2>,...)
or

INSERT INTO <table_name> VALUES (<value 1>,<value 2>,...);

- (ii) **DELETE** This command is used to delete rows of a table.

DELETE FROM <table_name> WHERE <column_name> = <some_value>;

- (iii) **SELECT** This command is used to retrieve data from an existing table or view table information. SELECT clause is equivalent to a projection operation of relational algebra.

SELECT <attribute_name1>,<attribute_name2>...FROM <table_name>;

SELECT * It is a quick way to select all the columns from the table.

SELECT * FROM <table_name>;

Consider the following table STUDENT for further queries:

Table: STUDENT

Roll	Name	Class	Section	Marks
1	Ajay	12	A	98
2	Pooja	11	C	45
3	Kajal	10	B	NULL
4	Ram	10	B	45
5	Shyam	12	A	79
6	Esha	10	C	61
7	Gungun	12	C	82
8	Sakshi	12	B	29
9	Mohit	10	A	71
10	Daksh	11	A	89

(iv) **DISTINCT Keyword** It is used to eliminate the duplicate rows from a query result.

e.g. SELECT DISTINCT Class FROM STUDENT;

Output

Class
12
11
10

(v) **WHERE Clause** This helps you to define the criteria to determine which rows are to be selected for output.

SELECT Name, Roll FROM STUDENT WHERE Class = 12;

Output

Name	Roll
Ajay	1
Shyam	5
Gungun	7
Sakshi	8

(vi) **Special Operator NULL** To search the column whose value is NULL in the table.

SELECT Roll, Name FROM STUDENT WHERE Marks IS NULL;

Output

Roll	Name
3	Kajal

Some operators used in WHERE clause are as follows

(a) **LIKE Operator** It is used to match a pattern with the help of the character "%" or "_". Here, % is used for any number of characters while _ is used for a single character.

(i) SELECT Name, Roll FROM STUDENT WHERE Name LIKE "S%";

Output

Name	Roll
Shyam	5
Sakshi	8

(ii) SELECT Name, Roll FROM STUDENT WHERE Name LIKE "S_____";

Output

Name	Roll
Shyam	5

(b) **BETWEEN Operator** This is used to specify the range of column values.

SELECT Name FROM STUDENT WHERE Marks BETWEEN 45 AND 68;

Output

Name
Pooja
Ram
Esha

(c) **AND Operator** AND is a logical operator which takes two arguments as input and retains true if both the arguments are true.

SELECT Roll FROM STUDENT WHERE Name = Mohit AND Class=10;

Output

Name
9

(d) **OR Operator** OR is a logical operator which takes two arguments as input and retains true if any one of the arguments is true.

SELECT Roll, Name FROM STUDENT WHERE Name = 'Mohit' OR Name = 'Ram';

Output

Roll	Name
4	Ram
9	Mohit

- (e) **IN Operator** It is used to specify condition based on a list. The IN operator selects values that match any value present in the list of values.

```
SELECT Name, Section FROM STUDENT WHERE Section IN ('B', 'C');
```

Output

Name	Roll
Pooja	C
Kajal	B
Ram	B
Esha	C
Gungun	C
Sakshi	B

- (vii) **Using NOT with Special Operators** The NOT operator is used to negate a condition.

```
(a) SELECT Name, Section FROM STUDENT WHERE Section NOT IN ('B', 'C');
```

Name	Roll
Ajay	A
Shyam	A
Mohit	A
Daksh	A

```
(b) SELECT * FROM STUDENT WHERE Marks IS NOT NULL AND Marks <= 45;
```

Name
Pooja
Ram
Sakshi

- (viii) **ORDER BY Clause** To display a sorted result of data in ascending or descending order. By default, the data is sorted in ascending order.

```
SELECT Name, Marks FROM STUDENT WHERE Marks <= 61 ORDER BY Name;
```

Name	Roll
Pooja	45
Ram	45
Sakshi	29

- (ix) **The UPDATE Command** This command is used to update existing records in a table.

```
UPDATE <table_name> SET column1 = value1 WHERE column2 = VALUE;
```

- e.g. (i) UPDATE STUDENT SET Marks = Marks + 15 WHERE Name = "Ram";
(ii) UPDATE STUDENT SET Marks = 80, Section = "C" WHERE Section = "A";
(iii) UPDATE STUDENT SET Marks = NULL WHERE Class = 12;

The Group Functions

These are also called as **aggregate functions**. They return a single value based on group of rows. The group functions are as follows:

- (i) **AVG Function** This function is used to calculate the average of a specified column.

```
SELECT AVG(Marks) FROM STUDENT;
```

AVG (Marks)
66.5556

- (ii) **MAX Function** This function is used to find the maximum value of a specified column.

```
SELECT MAX(Marks) FROM STUDENT;
```

MAX (Marks)
98

- (iii) **MIN Function** This function is used to find the minimum value of a specified column.

```
SELECT MIN(Marks) FROM STUDENT;
```

MIN (Marks)
29

- (iv) **SUM Function** This function is used to calculate sum of a specified column.

```
SELECT SUM(Marks) FROM STUDENT;
```

SUM (Marks)
599

(v) **COUNT Function** This function is used to count the number of values in a given column.

SELECT COUNT(Name) FROM STUDENT;

COUNT (Name)
10

GROUP BY Clause

This clause groups the rows in the resulting table by columns that have the same values, so that each group is reduced to a single row.

SELECT MAX(Marks) FROM STUDENT GROUP BY Section;

MAX (Marks)
98
45
82

HAVING Clause

This clause restricts grouped rows that appear in the resulting table. HAVING clause works as a WHERE clause with aggregate functions.

SELECT MAX(Marks) FROM STUDENT GROUP BY Section HAVING MAX (Marks) < 82;

MAX (Marks)
45

Equi Join

Equi join is a simple SQL join condition that uses equal sign as a comparison operator.

Syntax

```
SELECT col1,col2,col3
  FROM table1, table2
 WHERE table1.col1=table2.col1;
```

Consider the following tables PERSON and ORDERS:

PERSON

P_Id	Last_Name	First_Name	City
1	Sharma	Abhay	Mumbai
2	Gupta	Mohan	Delhi
3	Verma	Akhil	Mumbai

ORDERS

O_Id	Order_No	P_Id
1	10050	3
2	25000	3
3	5687	1
4	45000	1
5	35000	15

```
SELECT Last_Name, First_Name, Order_No
  FROM PERSON, ORDERS
 WHERE PERSON.P_Id = ORDERS.P_Id
 ORDER BY PERSON.Last_Name;
```

The query will give us the below result:

Last_Name	First_Name	Order_No
Sharma	Abhay	5687
Sharma	Abhay	45000
Verma	Akhil	10050
Verma	Akhil	25000

Cartesian Product

The cartesian product is also referred as **cross-join**. It returns all the rows in all the tables listed in the query. Each row in the first table is paired with all the rows in the second table. We can specify a CROSS JOIN in two ways:

(i) Using the JOIN syntax

```
SELECT * FROM <table_1> CROSS JOIN
<table_2>;
```

(ii) Without using the JOIN syntax

```
SELECT * FROM <table_1>, <table_2>;
```

Consider the following tables GAMESCORES and DEPARTMENTS

Table : Gamescores

PlayerName	DeptID	Scores
Jason	1	3000
Irene	1	1500
Jane	2	1000

Table : Departments

DeptID	DeptName
1	IT
2	Marketing

```
SELECT * FROM GAMESCORES CROSS JOIN
DEPARTMENTS;
```

The result set will look like this:

Player Name	DeptID	Scores	DeptID	Dept Name
Jason	1	3000	1	IT
Irene	1	1500	1	IT
Jane	2	1000	1	IT
Jason	1	3000	2	Marketing
Irene	1	1500	2	Marketing
Jane	2	1000	2	Marketing

UNION

The UNION is used to combine and return all the distinct rows selected by query. Tables are said to be union compatible if

- (i) Both the tables are of same degree.
- (ii) The domains of corresponding attributes in both the tables should be same.

Syntax

SELECT column1 [, column2] FROM table1 [, table2] [WHERE condition] UNION

SELECT column1 [, column2] FROM table1 [, table2] [WHERE condition]

Consider the following tables COURSE1 and COURSE2:

COURSE1

CourseID	CourseName
C001	Computer
C002	Science
C003	Math

COURSE2

CourseID	CourseName
C002	Science
C003	Math
C004	Accounts

SELECT * FROM COURSE1 UNION SELECT * FROM COURSE 2

The result set will look like this:

CourseID	CourseName
C001	Computer
C002	Science
C003	Math
C004	Accounts

PREVIOUS YEARS' EXAMINATION QUESTIONS (TOPIC 1)

2 Marks Question

1. Explain the concept UNION between two tables, with the help of appropriate example.
Delhi 2014

6 Marks Questions

2. Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables.All India 2017

DVD

DCODE	DTITLE	DTYPE
F101	Henry Martin	Folk
C102	Dhrupad	Classical
C101	The Planets	Classical
F102	Universal Soldier	Folk
R102	A day in life	Rock

MEMBER

MID	NAME	DCODE	ISSUEDATE
101	AGAM SINGH	R102	2017-11-30
103	ARTH JOSEPH	F102	2016-12-13
102	NISHA HANS	C101	2017-07-24

- To display all details from the table MEMBER in descending order of ISSUEDATE.
- To display the DCODE and DTITLE of all Folk Type DVDs from the table DVD.
- To display the DTTYPE and number of DVDs in each DTTYPE from the table DVD.
- To display all NAME and ISSUEDATE of those members from the table MEMBER who have DVDs issued (i.e., ISSUEDATE) in the year 2017.
- SELECT MIN (ISSUEDATE) FROM MEMBER;
- SELECT DISTINCT DTTYPE FROM DVD;
- SELECT D.DCODE, NAME, DTITLE
FROM DVD D, MEMBER M WHERE D.DCODE=M.DCODE;
- SELECT DTITLE FROM DVD
WHERE DTTYPE NOT IN ("Folk", "Classical");

3. Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables. All India 2016

Table : VEHICLE

VCODE	VEHICLETYPE	PERKM
V01	VOLVO BUS	150
V02	AC DELUXE BUS	125
V03	ORDINARY BUS	80
V05	SUV	30
V04	CAR	18

NOTE PERKM is Freight Charges per kilometre

Table : TRAVEL

CNO	CNAME	TRAVELDATE	KM	VCODE	NOP
101	K. Niwal	2015-12-13	200	V01	32
103	Fredrick Sym	2016-03-21	120	V03	45
105	Hitesh Jain	2016-04-23	450	V02	42
102	Ravi Anish	2016-01-13	80	V02	40
107	John Malina	2015-02-10	65	V04	2
104	Sahanubhuti	2016-01-28	90	V05	4
106	Ramesh Jaya	2016-04-06	100	V01	25

NOTE • KM is Kilometres travelled

• NOP is number of passengers travelled in vehicle.

- To display CNO, CNAME, TRAVELDATE from the table TRAVEL in descending order of CNO
- To display the CNAME of all the customers from the table TRAVEL who are traveling by vehicle with code V01 or V02.
- To display the CNO and CNAME of those customers from the table TRAVEL who travelled between '2015-12-31' and '2015-05-01'.
- To display all the details from table TRAVEL for the customers, who have travel distance more than 120 KM in ascending order of NOP.
- SELECT COUNT(*), VCODE FROM TRAVEL
GROUP BY VCODE HAVING COUNT(*)>1;
- SELECT DISTINCT VCODE FROM TRAVEL;
- SELECT VCODE,CNAME,VEHICLETYPE
FROM TRAVEL A, VEHICLE B
WHERE A.VCODE=B.VCODE AND KM<90;
- SELECT CNAME, KM*PERKM
FROM TRAVEL A, VEHICLE B
WHERE A.VCODE=B.VCODE AND A.VCODE='V05';

4. Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables. Delhi 2016

Table : VEHICLE

CODE	VTYPE	PERKM
101	VOLVO BUS	
102	AC DELUXE BUS	160
103	ORDINARY BUS	150
105	SUV	90
104	CAR	40
		20

- NOTE**
- PERKM is Freight Charges per kilometre
 - VTYPE is Vehicle Type

Table : TRAVEL

NO	NAME	TDATE	KM	CODE	NOP
101	Janish Kin	2015-11-13	200	101	32
103	Vedika Sahai	2016-04-21	100	103	45
105	Tarun Ram	2016-03-23	350	102	42
102	John Fen	2016-02-13	90	102	40
107	Ahmed Khan	2015-01-10	75	104	2
104	Raveena	2016-05-28	80	105	4
106	Kripal Anya	2016-02-06	200	101	25

- NOTE**
- NO is Traveller Number
 - KM is Kilometre travelled
 - NOP is number of travellers travelled in vehicle
 - TDATE is Travel Date

- To display NO, NAME, TDATE from the table TRAVEL in descending order of NO.
- To display the NAME of all the travellers from the table TRAVEL who are travelling by vehicle with code 101 or 102.
- To display the NO and NAME of those travellers from the table TRAVEL who travelled between '2015-12-31' and '2015-04-01'.
- To display all the details from table TRAVEL for the travellers, who have travelled distance more than 100 KM in ascending order of NOP.
- SELECT COUNT(*), CODE FROM TRAVEL GROUP BY CODE HAVING COUNT(*) >1;
- SELECT DISTINCT CODE FROM TRAVEL;
- SELECT CODE, NAME, VTYPE FROM TRAVEL A, VEHICLE B WHERE A.CODE=B.CODE AND KM<90;
- SELECT NAME, KM*PERKM FROM TRAVEL A, VEHICLE B WHERE A.CODE=B.CODE AND A.CODE= '105' ;

5. Consider the following DEPT and WORKER tables. Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii): Delhi 2015

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;`

6. Consider the following DEPT and EMPLOYEE tables. Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii). All India 2015

Table : DEPT

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

Table : EMPLOYEE

ENO	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

ENO	NAME	DOJ	DOB	GENDER	DCODE
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 employees

- (i) To display ENO, NAME, GENDER from the table EMPLOYEE in ascending order of ENO.
- (ii) To display the NAME of all the MALE employees from the table EMPLOYEE.
- (iii) To display the ENO and NAME of those employees from the table EMPLOYEE who are born between '1987-01-01' and '1991-12-01'.
- (iv) To count and display FEMALE employees who have joined after '1986-01-01'.
- (v) SELECT COUNT (*), DCODE FROM EMPLOYEE
GROUP BY DCODE HAVING COUNT(*)>1;
- (vi) SELECT DISTINCT DEPARTMENT FROM DEPT;
- (vii) SELECT NAME, DEPARTMENT FROM EMPLOYEE E,DEPT D
WHERE E.DCODE = D.DCODE AND ENO<1003;
- (viii) SELECT MAX(DOJ), MIN(DOB) FROM EMPLOYEE;

7. Consider the following tables SCHOOL and ADMIN and answer (a) and (b) parts of this question : All India 2014 C

Table : SCHOOL

Code	TeacherName	Subject	DOJ	Periods	Experience
1001	Ravi Shanker	English	12/03/2000	24	10
1009	Priya Rai	Physics	03/09/1998	26	12
1203	Lisa Anand	English	09/04/2000	27	5
1045	Yashraj	Maths	24/08/2000	24	15
1123	Ganan	Physics	16/07/1999	28	3
1167	Harish B	Chemistry	19/10/1999	27	5
1215	Umesh	Physics	11/05/1998	22	16

Table : ADMIN

CODE	GENDER	DESIGNATION
1001	Male	Vice Principal
1009	Female	Coordinator
1203	Female	Coordinator
1045	Male	HOD
1123	Male	Senior Teacher
1167	Male	Senior Teacher
1215	Male	HOD

(a) Write SQL statements for the following:

- (i) To display TEACHERNAME, PERIODS of all teachers whose periods are more than 25.
- (ii) To display all the information from the table SCHOOL in descending order of experience.
- (iii) To display DESIGNATION without duplicate entries from the table ADMIN.
- (iv) To display TEACHERNAME, CODE and corresponding DESIGNATION from tables SCHOOL and ADMIN of Male teachers.

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

- SELECT DESIGNATION, COUNT (*) FROM ADMIN GROUP BY DESIGNATION HAVING COUNT (*) < 2;
- SELECT MAX (EXPERIENCE) FROM SCHOOL;
- SELECT TEACHERNAME FROM SCHOOL WHERE EXPERIENCE > 12 ORDER BY TEACHERNAME;
- SELECT COUNT (*), GENDER FROM ADMIN GROUP BY GENDER;

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

Delhi 2014

TABLE : STORE

SNo	SName	Area
S01	ABC Computronics	GK II
S02	All Infotech Media	CP
S03	Tech Shoppe	Nehru Place
S04	Geeks Tecno 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):

- To display IName and Price of all the Items in ascending order of their Price.
- To display SNo and SName of all Store located in CP.
- To display Minimum and Maximum Price of each IName from the table ITEM.
- 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):

- SELECT DISTINCT IName FROM ITEM
WHERE Price >= 5000;
- SELECT Area, COUNT(*)
FROM STORE GROUP BY Area;
- SELECT COUNT(DISTINCT Area) FROM STORE;
- SELECT IName, Price * 0.05 DISCOUNT FROM ITEM
WHERE SNo IN (S02, S03);

9. Answer the questions (a) and (b) on the basis of the following tables **SHOPPE** and **ACCESSORIES**. All India 2014

TABLE : SHOPPE

ID	SName	Area
S001	ABC Computronics	CP
S002	All Infotech Media	GK II
S003	Tech Shoppe	CP
S004	Geeks Tecno Soft	Nehru Place
S005	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 located 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) Write the output of the following SQL commands:

- (i) SELECT DISTINCT Name FROM ACCESSORIES WHERE Price >= 5000;
- (ii) SELECT Area, COUNT(*) FROM SHOPPE GROUP BY Area;
- (iii) SELECT COUNT(DISTINCT Area) FROM SHOPPE;
- (iv) SELECT Name, Price * 0.05 DISCOUNT FROM ACCESSORIES WHERE SNo IN (S02, S03);

10. Write SQL queries for (a) to (f) and write the outputs for the SQL queries mentioned shown in (i) to (iv) parts on the basis of tables **PRODUCTS** and **SUPPLIERS**. All India 2013

TABLE : PRODUCTS

PID	PNAME	QTY	PRICE	COMPANY	SUPCODE
101	DIGITAL CAMERA 14X	120	12000	RENIX	S01
102	DIGITAL PAD 11i	100	22000	DIGI POP	S02
104	PEN DRIVE 16 GB	500	1100	STOREKING	S01
106	LED SCREEN 32	70	28000	DISPEXPERTS	S02
105	CAR GPS SYSTEM	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 products 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:
 (i) SELECT DISTINCT SUPCODE FROM PRODUCTS;
 (ii) SELECT MAX(PRICE), MIN(PRICE) FROM PRODUCTS;
 (iii) SELECT PRICE * QTY AMOUNT FROM PRODUCTS WHERE PID = 104;
 (iv) SELECT PNAME, SNAME FROM PRODUCTS P, SUPPLIERS S
 WHERE P.SUPCODE = S.SUPCODE AND QTY > 100;

11. Write SQL queries for (a) to (f) and write the outputs for the SQL queries mentioned shown in (i) to (iv) parts on the basis of tables ITEMS and TRADERS. Delhi 2013

TABLE : ITEMS

CODE	INAME	QTY	PRICE	COMPANY	TCODE
1001	DIGITAL PAD 12i	120	11000	XENITA	T01
1006	LED SCREEN 40	70	38000	SANTORA	T02
1004	CAR GPS SYSTEM	50	21500	GEOKNOW	T01
1003	DIGITAL CAMERA 12X	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 name 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:
- SELECT MAX(PRICE), MIN(PRICE) FROM ITEMS;
 - SELECT PRICE * QTY AMOUNT FROM ITEMS WHERE CODE = 1004;
 - SELECT DISTINCT TCODE FROM ITEMS;
 - SELECT INAME, TNAME FROM ITEMS I, TRADERS T
WHERE I.TCODE = T.TCODE AND QTY<100;

12. Write SQL queries for (a) to (f) and write the outputs for the SQL queries mentioned shown in (i) to (iv) parts on the basis of tables APPLICANTS and COURSES. Delhi (C) 2013

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, joineyear 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 names of all applicants in ascending order of their joineyear.
- (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 table APPLICANTS and COURSES.
- (g) Give the output of following SQL statements:
- SELECT NAME, JOINYEAR FROM APPLICANTS WHERE GENDER='F' and C_ID='A02';
 - SELECT MIN(JOINYEAR) FROM APPLICANTS WHERE Gender='M';
 - SELECT AVG(FEE) FROM APPLICANTS WHERE C_ID='A01' OR C_ID='A05';
 - SELECT SUM(FEE), C_ID FROM APPLICATIONS GROUP BY C_ID HAVING COUNT(*)=2;

13. Consider the following tables CABHUB and CUSTOMER and answer (a) and (b) parts of this question: Delhi 2012

TABLE : CABHUB

Vcode	VehicleName	Make	Color	Capacity	Charges
100	Innova	Toyota	WHITE	/	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

Ccode	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 and capacity of vehicles in ascending order of their seating Capacity.

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

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

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

(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;

14. Consider the following tables CUSTOMER and ONLINESHOP. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii). Delhi (C) 2012

TABLE : CUSTOMER

CID	CNAME	GENDER	SID	AREA
1001	R SHARMA	FEMALE	101	NORTH
1002	M R TIWARY	MALE	102	SOUTH
1003	M K KHAN	MALE	103	EAST
1004	A K SINGH	MALE	102	EAST
1005	S SEN	MALE	101	WEST
1006	R DUBEY	FEMALE	104	NORTH
1007	M AGARWAL	MALE	104	NORTH
1008	S DAS	FEMALE	103	SOUTH
1009	R K PATIL	FEMALE	102	NORTH
1010	N KRISHNA MURTY	MALE	102	SOUTH

TABLE : ONLINESHOP

SID	SHOP
101	MY BUY
102	ECO BUY
103	JUST SHOPPING
104	SHOPPING EASY

- (i) To display cname, area of all female customers from CUSTOMER table.
(ii) To display the details of all the customers in ascending order of CNAME within SID.
(iii) To display the total number of customers for each area from CUSTOMER table.
(iv) To display cname and corresponding shop from CUSTOMER table and ONLINESHOP table.
(v) SELECT COUNT(DATE), GENDER FROM CUSTOMER GROUP BY GENDER;
(vi) SELECT COUNT(*) FROM ONLINESHOP;
(vii) SELECT CNAME FROM CUSTOMER WHERE CNAME LIKE "L%";
(viii) SELECT DISTINCT AREA FROM CUSTOMER;

15. Consider the following tables CARDEN and CUSTOMER and answer (a) and (b) parts of this question: All India 2012

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	Feroza 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 names and the corresponding name of the cars hired by them.
- (b) Give the output of the following SQL queries:
- SELECT COUNT(DISTINCT Make) FROM CARDEN;
 - SELECT MAX(Charges), MIN (Charges) FROM CARDEN;
 - SELECT COUNT(*), Make FROM CARDEN;
 - SELECT CarName FROM CARDEN WHERE Capacity = 4;

16. Consider the following tables EMPLOYEE and SALGRADE and answer (a) and (b) parts of this question: All India 2011

TABLE : EMPLOYEE

E CODE	NAME	DESIG	S GRADE	DOJ	DOB
101	Abdul Ahmad	EXECUTIVE	S03	23-MAR-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

S GRADE	SALARY	HRA
S01	56000	18000
S02	32000	12000
S03	24000	8000

(a) Write SQL commands for the following statements:

- To display the details of all the EMPLOYEE in descending order of DOJ.
 - To display name and desig of those EMPLOYEE, whose sgrade is either S02 or S03.
 - To display the content of all the EMPLOYEE table, whose DOJ is in between '09-FEB-2009' and '08-AUG-2009'.
 - 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:
- SELECT COUNT(SGRADE), SGRADE FROM EMPLOYEE GROUP BY SGRADE;
 - SELECT MIN(DOB), MAX (DOJ) FROM EMPLOYEE;
 - SELECT NAME, SALARY FROM EMPLOYEE E, SALGRADE S
WHERE E.SGRADE = S.SGRADE AND E.ECODE < 103;
 - SELECT S GRADE, SALARY+HRA FROM SALGRADE WHERE S GRADE = 'S02';

17. Consider the following tables WORKER and PAYLEVEL and answer (a) and (b) parts of this question: Delhi 2011

TABLE : WORKER

E CODE	NAME	DESIG	P LEVEL	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	Ameen Ahmed	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:
- To display the details of all WORKER in descending order of DOB.
 - To display name and desig of those WORKER, whose plevel is either P001 or P002.
 - To display the content of all the WORKER table, whose DOB is in between '19-JAN-1984' and '18-JAN-1987'.
 - 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:
- SELECT COUNT(PLEVEL), PLEVEL FROM WORKER GROUP BY PLEVEL;
 - SELECT MAX(DOB), MIN(DOJ) FROM WORKER;
 - SELECT NAME, PAY FROM WORKER W, PAYLEVEL P
WHERE W.PLEVEL= P.PLEVEL AND W.ECODE<13;
 - SELECT PLEVEL, PAY+ALLOWANCE FROM PAYLEVEL WHERE PLEVEL = 'P003';

8. Consider the following tables STORE and SUPPLIERS and answer (a) and (b) parts of this question: Delhi 2010

TABLE : STORE

ItemNo	Item	Scode	Qty	Rate	LastBuy
2005	Sharpener Classic	23	60	8	31-JUN-09
2003	Ball Pen 0.25	22	50	25	01-FEB-10
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:
- To display details of all the items in the STORE table in ascending order of LastBuy.
 - To display ItemNo and Item of those items from STORE table whose Rate is more than ₹15.
 - 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.
 - 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:
- SELECT COUNT(DISTINCT Scode) FROM STORE;
 - SELECT Rate * Qty FROM STORE WHERE ItemNo = 2004;
 - SELECT Rate * Qty FROM STORE S, SUPPLIERS P
WHERE S.Scode = P.Scode AND ItemNo = 2006;
 - SELECT MAX(LastBuy) FROM STORE;

19. Consider the following tables STOCK and DEALERS and answer (a) and (b) parts of this question: All India 2010

TABLE : STOCK

ItemNo	ItemName	Dcode	Qty	UnitPrice	StockDate
5005	Ball Pen 0.5	102	100	16	31-MAR-10
5003	Ball Pen 0.25	102	150	20	01-JAN-10
5002	Gel Pen Premium	101	125	14	14-FEB-10
5006	Gel Pen Classic	101	200	22	01-JAN-09
5001	Eraser Small	102	210	5	19-MAR-09
5004	Eraser Big	102	60	10	12-DEC-09
5009	Sharpener Classic	103	160	8	23-JAN-09

TABLE : DEALERS

Dcode	Dname
101	Reliable Stationers
103	Classic Plastics
102	Clear Deals

(a) Write SQL commands for the following statements:

- To display details of all the items in the STOCK table in ascending order of StockDate
- To display ItemNo and ItemName of those items from STOCK table whose UnitPrice is more than ₹ 10.
- To display the details of those items whose dealer code (Dcode) is 102 or quantity in stock (Qty) is more than 100 from the table STOCK.
- To display maximum UnitPrice of items for each dealer individually as per Dcode from the table STOCK.

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

- SELECT COUNT(DISTINCT Dcode) FROM STOCK;
- SELECT Qty * UnitPrice FROM STOCK WHERE ItemNo = 5006;
- SELECT ItemName, Dname FROM STOCK S, DEALERS D
WHERE S.Dcode = D.Dcode AND ItemNo = 5004;
- SELECT MIN(StockDate) FROM STOCK;

20. Consider the following tables GARMENT and FABRIC. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii). Delhi 2009

TABLE : GARMENT

GCODE	DESCRIPTION	PRICE	FCODE	READYDATE
10023	PENCIL SKIRT	1150	F03	19-DEC-08
10001	FORMAL SHIRT	1250	F01	12-JAN-08
10012	INFORMAL SHIRT	1550	F02	06-JUN-08
10024	BABY TOP	750	F03	07-APR-07
10090	TULIP SKIRT	850	F02	31-MAR-07
10019	EVENING GOWN	850	F03	06-JUN-08
10009	INFORMAL PANT	1500	F02	20-OCT-08
10007	FORMAL PANT	1350	F01	09-MAR-08
10020	FROCK	850	F04	09-SEP-07
10089	SLACKS	750	F03	20-OCT-08

TABLE : FABRIC

FCODE	TYPE
F04	POLYSTER
F02	COTTON
F03	SILK
F01	TERELENE

- (i) To display GCODE and DESCRIPTION of each GARMENT in ascending 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 of 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 alongwith 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;
11. Consider the following tables DRESS and MATERIAL. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii). All India 2009

TABLE : DRESS

DCODE	DESCRIPTION	PRICE	MPCODE	LAUNCHDATE
10001	FORMAL SHIRT	1250	M001	12-JAN-08
10020	FROCK	750	M004	09-SEP-07
10012	INFORMAL SHIRT	1450	M002	06-JUN-08
10019	EVENING GOWN	850	M003	06-JUN-08
10090	TULIP SKIRT	850	M002	31-MAR-07
10023	PENCIL SKIRT	1250	M003	19-DEC-08
10089	SLACKS	850	M003	20-OCT-08
10007	FORMAL PANT	1450	M001	09-MAR-08
10009	INFORMAL PANT	1400	M002	20-OCT-08
10024	BABY TOP	650	M003	07-APR-07

TABLE : MATERIAL

MPCODE	TYPE
M001	TERELENE
M002	COTTON
M004	POLYSTER
M003	SILK

- (i) To display DCODE and DESCRIPTION of each dress in ascending order of DCODE.
- (ii) To display the details of all the dresses which have LAUNCHDATE in between 05-DEC-07 and 20-JUN-08 (inclusive of both the dates).
- (iii) To display the average PRICE of all the dresses which are made up of material with MPCODE as M003.
- (iv) To display materialwise highest and lowest price of dresses from DRESS table. (display MPCODE of each dress alongwith highest and lowest price).

(v) SELECT SUM(PRICE) FROM DRESS WHERE MCODE = 'M001';

(vi) SELECT DESCRIPTION, TYPE FROM DRESS, MATERIAL

WHERE DRESS.DCODE = MATERIAL.MCODE AND DRESS.PRICE >= 1250;

(vii) SELECT MAX(MCODE) FROM MATERIAL;

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

22. Consider the following tables STUDENT and STREAM. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii). Delhi (C) 2009

TABLE : STUDENT

SCODE	NAME	AGE	STRCDE	POINTS	GRADE
101	Amit	16	1	6	NULL
102	Arjun	13	3	4	NULL
103	Zaheer	14	2	1	NULL
105	Gagan	15	5	2	NULL
108	Kumar	13	6	8	NULL
109	Rajesh	17	5	8	NULL
110	Naveen	13	3	9	NULL
113	Ajay	16	2	3	NULL
115	Kapil	14	3	2	NULL
120	Gurdeep	15	2	6	NULL

TABLE : STREAM

STRCDE	STRNAME
1	SCIENCE+COMP
2	SCIENCE+BIO
3	SCIENCE+ECO
4	COMMERCE+MATHS
5	COMMERCE+SOCIO
6	ARTS+MATHS
7	ARTS+SOCIO

- (i) To display the name of streams in alphabetical order from table STREAM.
- (ii) To display the number of students whose POINTS are more than 5.
- (iii) To update GRADE to 'A' for all those students who are getting more than 8 as POINTS.
- (iv) ARTS+MATHS stream is no more available. Make necessary change in table STREAM.
- (v) SELECT SUM(POINTS) FROM STUDENT WHERE AGE > 14;
- (vi) SELECT STRCDE, MAX(POINTS) FROM STUDENT
GROUP BY STRCDE HAVING SCODE BETWEEN 105 AND 130;
- (vii) SELECT AVG(AGE) FROM STUDENT WHERE SCODE IN (102,105, 110, 120);
- (viii) SELECT COUNT(STRNAME) FROM STREAM WHERE STRNAME LIKE "SCI%";

Answers

1. 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

2. (i) SELECT * FROM MEMBER ORDER BY ISSUEDATE DESC;
(ii) SELECT DCODE, DTITLE
FROM DVD
WHERE DTTYPE = "Folk";
(iii) SELECT DTTYPE, COUNT (*)
FROM DVD
GROUP BY DTTYPE;
(iv) SELECT NAME, ISSUEDATE
FROM MEMBER
WHERE ISSUEDATE LIKE '2017%';
(v)

MIN(ISSUEDATE)

2016-12-13

DTYPE
Folk
Classical
Rock

(vii)

DCODE	NAME	DTITLE
C101	NISHA HANS	The Planets
F102	ARTH JOSEPH	Universal Soldier
R102	AGAM SINGH	A day in life

(viii)

DTITLE
A day in life

3. (i) SELECT CNO, CNAME, TRAVELDATE
FROM TRAVEL
ORDER BY CNO DESC;
(ii) SELECT CNAME FROM TRAVEL
WHERE VCODE = "V01" OR VCODE="V02";
(iii) SELECT CNO, CNAME FROM TRAVEL
WHERE TRAVELDATE BETWEEN
'2015-12-31' AND '2015-05-01';
(iv) SELECT * FROM TRAVEL
WHERE KM>120 ORDER BY NOP;
(v)

COUNT(*)	VCODE
2	V01
2	V02

(vi)

VCODE
V01
V03
V02
V04
V05

(vii)

VCODE	CNAME	VEHICLETYPE
V02	Ravi Anish	AC DELUXE BUS
V04	John Malina	CAR

(viii)

CNAME	KM*PERKM
Sahanubhuti	2700

4. (i) SELECT NO, NAME, TDATE FROM TRAVEL
ORDER BY NO DESC;
(ii) SELECT NAME FROM TRAVEL
WHERE CODE = 101 OR CODE = 102;
(iii) SELECT NO, NAME FROM TRAVEL
WHERE TDATE BETWEEN '2015-12-31'
AND '2015-04-01';
(iv) SELECT * FROM TRAVEL WHERE KM > 100
ORDER BY NOP;

(v)

COUNT(*)	CODE
2	101
2	102

(vi)

CODE
101
103
102
104
105

(vii)

CODE	NAME	VTYPE
104	Ahmed Khan	CAR
105	Raveena	SUV

(viii)

NAME	KM*PERKM
Raveena	3200

5. (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 '1987-01-01'
AND '1991-12-01';

- (iv) SELECT COUNT(*) FROM WORKER
WHERE GENDER = "MALE"
AND DOJ > '1986-01-01';

(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

6. (i) SELECT ENO, NAME, GENDER
FROM EMPLOYEE ORDER BY ENO;

- (ii) SELECT NAME FROM EMPLOYEE
WHERE GENDER = 'MALE';

- (iii) SELECT ENO, NAME FROM EMPLOYEE
WHERE DOB BETWEEN '1987-01-01'
AND '1991-12-01';

- (iv) SELECT COUNT(*) FROM EMPLOYEE
WHERE GENDER = 'FEMALE'
AND DOJ > '1986-01-01';

(v)

COUNT (*)	DCODE
2	D01
2	D05

(vi)

DEPARTMENT
INFRASTRUCTURE
MARKETING
MEDIA
FINANCE
HUMAN RESOURCE

(vii)

NAME	DEPARTMENT
George K	INFRASTRUCTURE
Ryma Sen	MEDIA

(viii)

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

7. (a)
- SELECT TEACHERNAME, PERIODS FROM SCHOOL WHERE PERIODS > 25;
 - SELECT * FROM SCHOOL ORDER BY EXPERIENCE DESC;
 - SELECT DISTINCT DESIGNATION FROM ADMIN;
 - SELECT TEACHERNAME, CODE, DESIGNATION FROM SCHOOL S, ADMIN A WHERE S.CODE = A.CODE AND GENDER = "MALE";

(b) (i)

DESIGNATION	COUNT (*)
VICE PRINCIPAL	1

(ii)

MAX (EXPERIENCE)	
	16

(iii)

TEACHERNAME	
UMESH	
YASHRAJ	

(iv)

COUNT (*)	GENDER
5	MALE
2	FEMALE

8. (a) (i) SELECT IName, Price FROM ITEM ORDER BY Price;
- (ii) SELECT SNo, SName FROM STORE WHERE Area = 'CP';
- (iii) SELECT IName, MIN(Price) "Minimum Price", MAX(Price) "Maximum Price" FROM ITEM GROUP BY IName;
- (iv) SELECT IName, Price, SName FROM ITEM I, STORE S WHERE I.SNo = S.SNo;

(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

9. (a) (i) SELECT Name, Price FROM ACCESSORIES ORDER BY Price ASC;
- (ii) SELECT ID, SName FROM SHOPPE WHERE Area = 'Nehru Place';
- (iii) SELECT MIN(Price) "Minimum Price", MAX(Price) "Maximum Price", Name FROM ACCESSORIES GROUP BY Name;
- (iv) The query for this statement cannot be done because relation column, i.e. foreign key is not present.

(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.

10. (a) SELECT * FROM PRODUCTS ORDER BY PNAME;
- (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 = 'DELHI'
OR CITY = 'CHENNAI';
(f) SELECT COMPANY, PNAME
FROM PRODUCTS
ORDER BY COMPANY DESC;

(g) (i)	SUPCODE	
	S01	
	S02	
	S03	

(ii)	MAX (PRICE)	MIN (PRICE)
	28000	1100

(iii)	AMOUNT	
	550000	

(iv)	PNAME	SNAME
	DIGITAL CAMERA 14X	GET ALL INC
	PEN DRIVE 16 GB	GET ALL INC

11. (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
ORDER BY COMPANY DESC;
(g) (i)

MAX (PRICE)	MIN (PRICE)
38000	1200

(ii)	AMOUNT	
	1075000	

(iii)	TCODE
	T01
	T02
	T03

(iv)	I NAME	T NAME
	LED SCREEN 40	DISP HOUSE IN.
	CAR GPS	ELECTRONIC
	SYSTEM	SALES

12. (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) (i)	NAME	JOINYEAR
	Avisha	2009
(ii)	MIN (JOINYEAR)	
	2009	
(iii)	AVG(FEE)	
	31666.666	
(iv)	SUM(FEE)	C_ID
	55000	A01

13. (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 C1, CUSTOMER C2
WHERE C1.Vcode = C2.Vcode;

(b) (i) COUNT(DISTINCT Make)

4

(ii)

MAX(Charges)	MIN(Charges)
35	12

(iii) This query will execute but COUNT (*) will give 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

14. (i) SELECT CNAME, AREA

FROM CUSTOMER

WHERE GENDER = 'FEMALE';

(ii) SELECT * FROM CUSTOMER
ORDER BY SID, CNAME;

(iii) SELECT COUNT(*)
FROM CUSTOMER GROUP BY AREA;

(iv) SELECT CNAME, SHOP
FROM CUSTOMER C, ONLINESHOP O
WHERE C.SID = O.SID;

(v)

COUNT(DATE)	GENDER
4	FEMALE
6	MALE

(vi)

COUNT(*)
4

(vii) EMPTY Set

(viii)

AREA
NORTH
SOUTH
EAST
WEST

15. (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 C1, CUSTOMER C2
WHERE C1.Ccode = C2.Ccode;

(b) (i)

COUNT(DISTINCT Make)

4

(ii)

MAX(Charges)	MIN(Charges)
35	12

(iii) This query will execute but COUNT (*) will give 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)

CarName
SX4
C-Class

16. (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
2	S03
2	S02
1	S01

	MIN(DOB)	MAX(DOJ)
(ii)	13-JAN-1980	12-FEB-2010
(iii)	NAME	SALARY
Abdul Ahmad	24000	
Ravi Chander	32000	
(iv)	SGRADE	SALARY + HRA
S02	44000	

17. (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	P003	
2	P002	
(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	

18. (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
Gel Pen Classic	Sname Premium Stationery
(iv)	MAX (LastBuy)
	24-FEB-10
19. (a) (i)	SELECT *
	FROM STOCK ORDER BY StockDate;
(ii)	SELECT ItemNo, ItemName
	FROM STOCK WHERE UnitPrice>10;
(iii)	SELECT * FROM STOCK
	WHERE Dcode = 102 OR Qty>100;
(iv)	SELECT MAX(UnitPrice)
	FROM STOCK GROUP BY Dcode;
(b) (i)	COUNT (DISTINCT Dcode)
	3
(ii)	Qty * UnitPrice
	4400
(iii)	ItemName
Eraser Big	Dname Clear Deals
(iv)	MIN (StockDate)
	01-JAN-09
20. (i)	SELECT GCODE, DESCRIPTION
	FROM GARMENT ORDER BY GCODE DESC;
(ii)	SELECT * FROM GARMENT
	WHERE READYDATE BETWEEN '08-DEC-08' 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	
(viii)	COUNT (DISTINCT PRICE)	
	7	

21. (i) SELECT DCODE, DESCRIPTION
FROM DRESS ORDER BY DCODE;
(ii) SELECT * FROM DRESS
WHERE LAUNCHDATE BETWEEN
'05-DEC-07' AND '20-JUN-08';
(iii) SELECT AVG(PRICE)
FROM DRESS WHERE MCODE = 'M003';
(iv) SELECT MCODE, MAX(PRICE),
MIN(PRICE)
FROM DRESS
GROUP BY MCODE;
- | (v) | SUM (PRICE) |
|-----|-------------|
| | 2700 |
- (vi) No Output
- | (vii) | MAX (MCODE) |
|-------|-------------|
| | M004 |

(viii)	COUNT (DISTINCT PRICE)
	6

22. (i) SELECT STRNAME
FROM STREAM
ORDER BY STRNAME;
(ii) SELECT COUNT(*)
FROM STUDENT
WHERE POINTS > 5;
(iii) UPDATE STUDENT
SET GRADE = 'A'
WHERE POINTS > 8;
(iv) DELETE FROM STREAM
WHERE STRNAME = "ARTS + MATHS";

(v)	SUM (POINTS)
	25

(vi)	STRCDE	MAX (POINTS)
	6	8
	5	8
	3	9
	2	6

(vii)	AVG (AGE)
	14

(viii)	COUNT (STRNAME)
	3