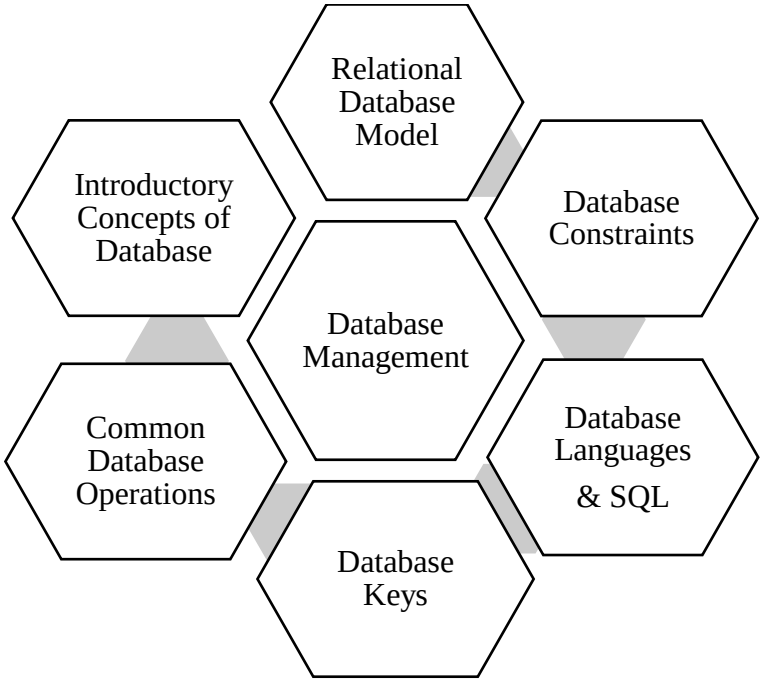
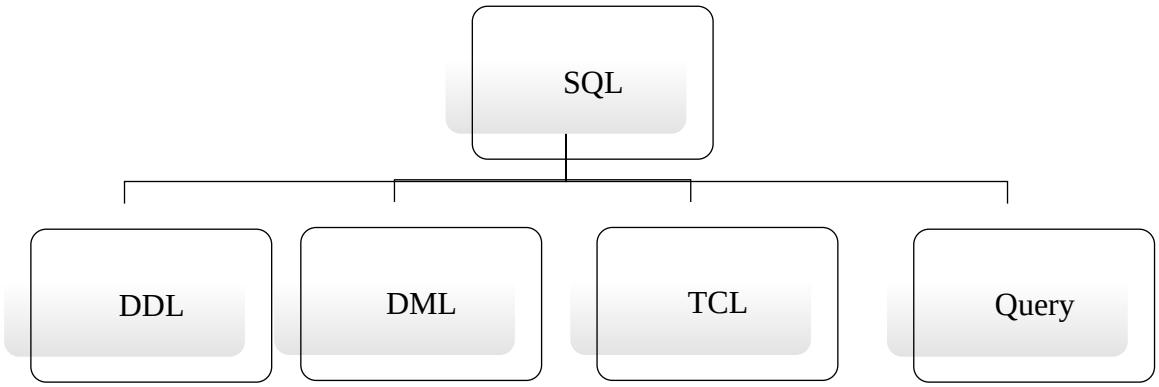
Unit : 3 Database Management and Mysql



Structured Query Language (SQL)



Operations in SQL

SELECT		ALIAS		ORDER BY	
	MATHS OPERATION		RELATIONAL OPS		LOGIC OPS
IN / NOT IN		IS / IS NOT		LIKE / NOT LIKE	
	DISTINCT		GROUP BY		HAVING
EQUI JOIN		NATURAL JOIN		CARTESIAN PRODUCT	

Database Management System (DBMS)

Common Terminologies related to database:

- **Data** -> Raw facts or figures
- **Database** -> A collection of interrelated data.
- **DBMS** -> A collection of files and a set of programs allowing users to access/modify these files are known as Database Management System.
- **Data Redundancy** - > Duplication of data.
- **Data Security** -> Protection of data against accidental/intentional disclosure to unauthorized person or unauthorized modification/destruction.
- **Data Privacy** -> Right of individual/organization to determine when/how/what information to be transmitted to others.

Need of using Database:

* Helps to store data in a structured manner
* Query in the Database (i.e. ask questions about the data)
* Sort and Manipulate Data in the Database
* Validate the Data Entered and check for inconsistencies
* Produce Flexible Reports

Advantage of using Database:

- Reduce data redundancy
- Control inconsistency
- Facilitates sharing of data
- Enforce standards
- Ensure data security
- Maintain integrity

Limitations of implementing Database:

⊗ Compromise of Security and Integrity without good control
⊗ Performance overhead
⊗ Extra hardware required sometimes
⊗ Complex system

Types of DBMS:

- Hierarchical DBMS
- Network Based DBMS
- Object Based DBMS
- Relational DBMS

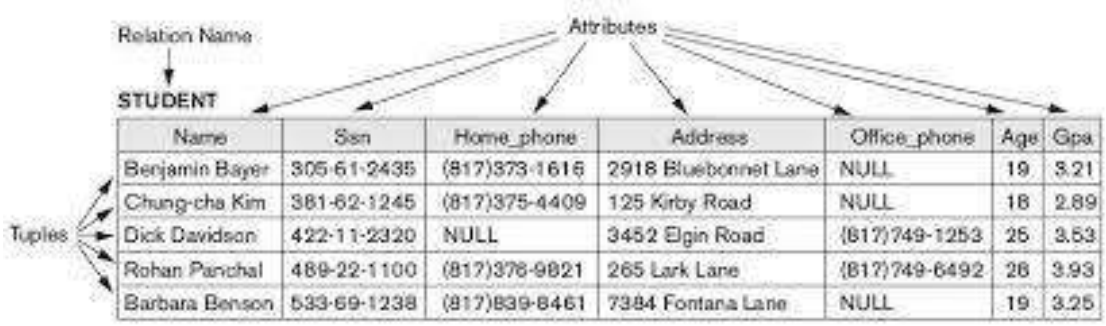
RELATIONAL DATA MODEL

Relational Data Model is defined as a model of defining a database as a collection of tables/relations i.e. arrangement of values in rows/tuples and columns/fields/attributes.

Common Terminologies related to Relational Data Model:

- **Relation**: Collection of data organized in rows and columns where each cell has atomic value. (same as Table)
- **Tuple**: Row of a table (same as Record)
- **Attribute**: Column of a table (same as Field)
- **Domain**: Range of values (data types) allowed for an attribute
- **Degree**: No. of attributes/columns/fields in a table
- **Cardinality**: No. of tuples/rows/records in a table
- **View**: Virtual table (no physical existence) derived from one or more base table for ease of query only.
- **Referential Integrity:** Property of database that requires every value of one attribute of a Relation must be present in another attribute (same datatype) in a different (or the same) relation.

Example:



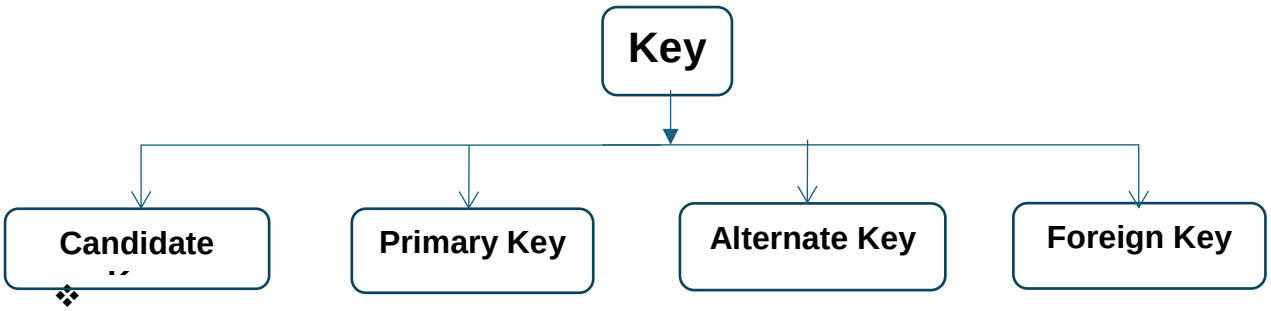
Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

In the above table STUDENT, degree = 7 and cardinality = 5.

Database Key

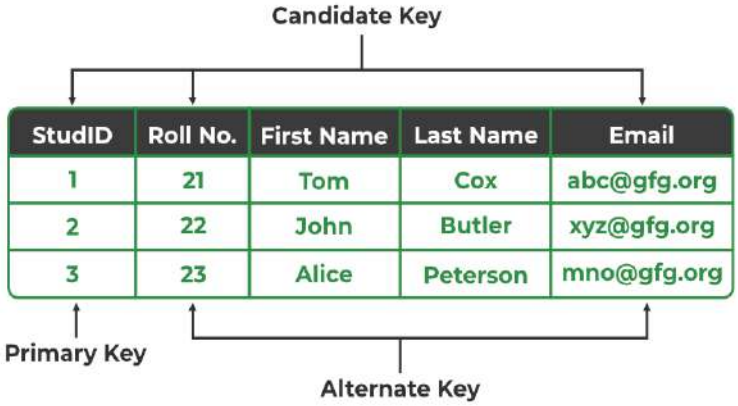
Key in a database is defined as a set of attributes to identify each record uniquely in a table. A Key must be unique and not null.

Classification of Keys:



- **Candidate Key:** Candidate key is defined as a set of minimum no. of attributes to uniquely identify a record in a table. A table may have multiple candidate keys.
- **Primary Key:** The one candidate key chosen by Database Administrator for a table to uniquely identify a record in a table is said to be Primary Key of that table. A table can have exactly one Primary Key.
- **Alternate Key:** Candidate key(s) not chosen by Database Administrator in a table is/are defined as alternate key(s). A table can have 0 or more alternate keys.

❖ **Foreign Key:** Foreign Key is a non-key attribute derived from primary key of some other table. A table can have 0 or more foreign keys.



Student_Details:

Roll No.	Name	Course_Id
1	Aryan	101
2	Sachin	102
3	Prince	103

FOREIGN KEY

Student_Marks:

PRIMARY KEY

Course_Id	Gpa
101	3
102	4
103	2.5

Data types in SQL

- Numeric data types: Used for representing number in a column
  e.g. –
  int(m) – Integer of maximum length m i.e. maximum number of digits allowed in m.
  float(m,d), decimal (m,d), numeric(m,d) – Real number of maximum length m where maximum number of digits permissible after decimal point is d and before decimal point is m-d.

- Date & Time data types: Used to represent date, time, or both in a column. Data is enclosed with quotes ‘ ’ or “ “.
  e.g. - **date, datetime, time**

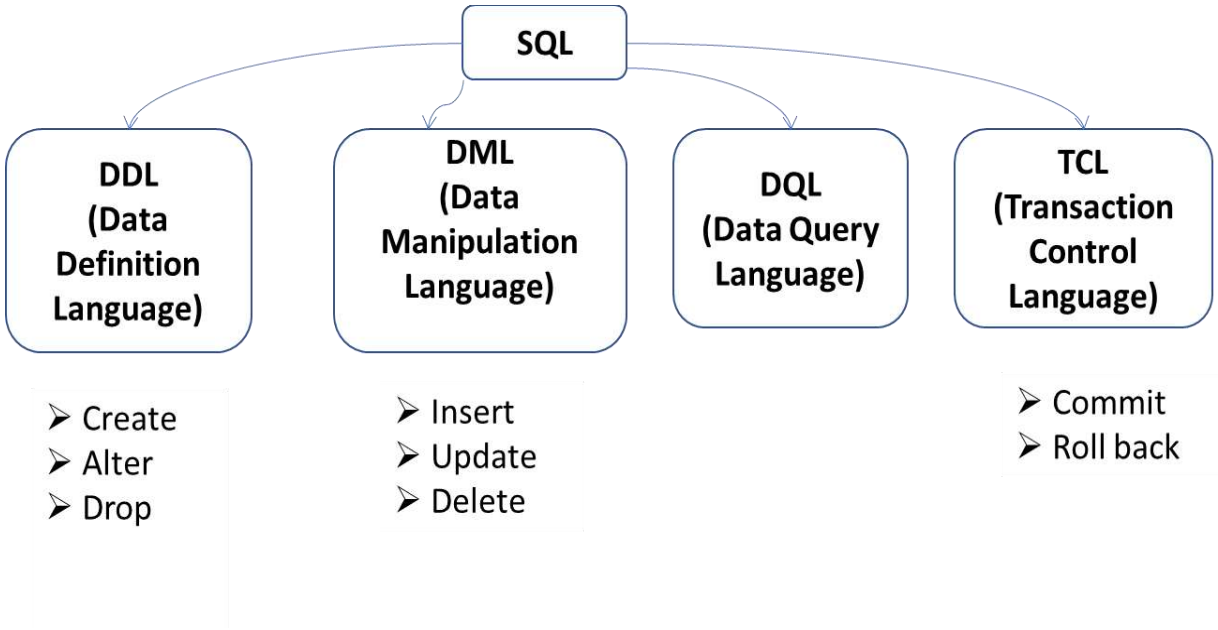
- String / Text data types: Used to represent text in a column. Data is enclosed with quotes ‘ ’ or “ “.
  e.g. –
  char(m) – Fixed length character of length m where 1 character takes 1 Byte in memory. Memory space is wasted if text size is less than m.

varchar(m) – Variable length character allowing maximum number of characters m. It saves memory allocation for text having lesser size than m.

blob – Binary Large object for huge size text.

* **NULL** – NULL is said to be the absence of any value in a column. No arithmetic or comparison operation can be performed on NULL value.

Classification of Database Language



Data Definition Language (DDL):

Data Definition Language (DDL) defines the different structures in a database like table, view, index etc.

DDL statements are used to create structure of a table, modify the existing structure of the table and remove the existing table.

e.g. - CREATE, ALTER, DROP

Syntax of DDL statements:

- CREATE TABLE table_name
  (column_name datatype constraint)
- ALTER TABLE table_name
  ADD column datatype constraint (if any)
  MODIFY column new_datatype new_constraint (if any)
  DROP column
- DROP TABLE table_name

Data Manipulation Language (DML):

Data Manipulation Language (DML) statements are used to access and manipulate

data in existing tables.

The manipulation includes inserting data into tables, deleting data from the tables and modifying the existing data.

e.g. – INSERT, UPDATE, DELETE

Types of DML statements:

- INSERT record

  INSERT INTO table_name(columns) VALUES (1 or more comma separated values)

- UPDATE one or more columns in already existing record(s)

  UPDATE table_name SET column = value or expression

  (comma separated if multiple columns updated) WHERE condition

- DELETE record

  DELETE FROM table_name WHERE condition

Transaction Control Language (TCL):

Database ensures that a database transaction i.e. complete set of records involved in a transaction either fully completed or not taken place at all to maintain data consistency. Transaction Control Language (TCL) statements allows to save or revert database transactions.

e.g. –

COMMIT – Save the changes permanently in the database

ROLL BACK – Revert back the changes made in database

Query:

Query is a type of SQL commands which accepts tables (relations), columns (fields or attributes) and conditions or specifications if any and display the output by means of a temporary table which consists of data represented through fields and records.

Structure of Query:

SELECT < 1, multiple (comma i.e. , separated) or all columns >

FROM < 1 table or multiple tables (comma i.e. , separated) in case of join >

WHERE <condition on column(s)>

GROUP BY <1 column>

HAVING < condition on aggregate function on a column only if group by exists >

ORDER BY <0, 1 or more (comma i.e. , separated) columns >

Note:

I. Among above SELECT and FROM are mandatory statements in a query and all other statements are optional.

II. SELECT statement contains one or more columns. * should be used to display all columns. Functions or expressions on columns can also be done.

III. FROM statement contains multiple tables only if columns from more than one tables are displayed through SELECT statement in case of product or join operations. Here records can be fetched from multiple tables.

IV. WHERE clause may contain multiple conditions related with logical OR / AND operators. Logical NOT operator can be used on a condition also.

V. GROUP BY clause is used if statistical records are to be displayed based on a field/column. In this case SELECT statements should contain GROUP BY field and aggregate function on another column at least. Once a group is formed individual records cannot be accessed in the same query.

VI. ORDER BY clause can be used to arrange the output records in ascending (by default) or descending order of values in one or more columns.

Order of execution of a query

Step 1: Identify table(s) with FROM clause

Step 2: Filter records using WHERE clause

Step 3: Form group if any using GROUP BY clause

Step 4: Filter groups using HAVING clause only if GROUP BY is used

Step 5: Arrange the output records in ascending or descending order using ORDER BY

Step 6: Display the fields mentioned in SELECT clause.

Database Constraints

Rules imposed on the values of one or more columns in the tables are called database constraints.

The database constraints are:

UNIQUE	Ensures that all values in a column are different. No two records have same values in that column.
NOT NULL	Ensures that a column can never have NULL values.
PRIMARY KEY	Uniquely identify a record. It is a combination of UNIQUE and NOT NULL.
CHECK	Specify the domain of values with certain criteria for a column.
DEFAULT	Provides default value for a column when no value is specified.
REFERENCES / FOREIGN KEY	Ensures referential integrity between the foreign key of dependent / referencing table and primary key of independent / referenced table.

SQL STATEMENTS WITH EXAMPLES

Create two tables EMPL and DEPT as follows:

Table : DEPT

DEPT_ID	DNAME	DLOC	MAX_STRENGTH
D01	FINANCE	MUMBAI	20
D02	ADMIN	KOLKATA	15
D03	IT	CHENNAI	5

Table : EMPL

EID	ENAME	GEN	DOJ	HOMETOWN	SALARY	MGR_ID	DEPT_ID
E0001	RITU SEN	F	20/06/2002	KOLKATA	40000.00		D03
E0002	MALCOM RAY	M	12/11/1998	BANGALORE	50000.00		D02
E0003	SUNDAR P	M	9/12/2008	BANGALORE	40000.00		D01
E0004	ANISHA RAWAT	F	4/09/2019	DELHI	20000.00	E0001	D03
E0005	SANA KHAN	F	31/08/2017	DELHI	30000.00	E0003	D01

In DEPT table:

1. DEPT_ID is primary key
2. DNAME is not null
3. MAX_STRENGTH should be minimum 1

In EMPL table:

1. EID is primary key
2. ENAME is not null
3. HOMETOWN is 'BANGALORE' by default
4. SALARY is between 5000.00 and 300000.00
5. MGR_ID refers to EID of manager
6. DEPT_ID refers to DEPT_ID of table DEPT

A. Write DDL statement to create a database OFFICE and define two tables mentioned as above under OFFICE database.

Create new database OFFICE in MySQL as following:

CREATE DATABASE OFFICE;

Work inside the database OFFICE as following:

USE OFFICE;

Note: By default, TEST database is used which is in-built database in MySQL. So no need to create test. Only 'use test;' statement can be written to enter test.

DDL statement to create DEPT table is as following:

SOLUTION 1	SOLUTION 2
CREATE TABLE DEPT **(** **DEPT_ID VARCHAR(4) PRIMARY KEY,** **DNAME VARCHAR(15) NOT NULL,** **DLOC VARCHAR(20),** **MAX_STRENGTH INT(2) CHECK (MAX_STRENGTH >= 1)** **);**	**CREATE TABLE DEPT** **(** **DEPT_ID VARCHAR(4),** **DNAME VARCHAR(15) NOT NULL,** **DLOC VARCHAR(20),** **MAX_STRENGTH INT(2),** **PRIMARY KEY(DEPT_ID),** **CHECK (MAX_STRENGTH >= 1)** **);**

Schema or structure of table DEPT is as follows:

DESC DEPT;

Field	Type	Null	Key	Default	Extra
DEPT_ID	varchar(4)	NO	PRI	NULL	
DNAME	varchar(15)	NO		NULL	
DLOC	varchar(20)	YES		NULL	
MAX_STRENGTH	int(2)	YES		NULL	

DDL statement to create EMPL table is as following:

```
CREATE TABLE EMPL
(
EID VARCHAR(6) PRIMARY KEY,
ENAME VARCHAR(30) NOT NULL,
GEN CHAR(1) CHECK (GEN IN ('M', 'F', 'T')),
DOJ DATE,
HOMETOWN VARCHAR(20) DEFAULT 'BANGALORE',
SALARY DECIMAL(8, 2) CHECK (SALARY BETWEEN 5000 AND 300000),
MGR_ID VARCHAR(6) REFERENCES EMPL(EID),
DEPT_ID VARCHAR(4) REFERENCES DEPT(DEPT_ID)
);
```

or,

```
CREATE TABLE EMPL
(
        EID VARCHAR(6),
        ENAME VARCHAR(30) NOT NULL,
        GEN CHAR,
        DOJ DATE,
        HOMETOWN VARCHAR(20) DEFAULT 'BANGALORE',
        SALARY DECIMAL(8, 2) ,
        MGR_ID VARCHAR(6) ,
        DEPT_ID VARCHAR(4) ,
        PRIMARY KEY(EID),
        CHECK (GEN IN ('M', 'F', 'T')),
        CHECK (SALARY BETWEEN 5000.00 AND 300000.00),
        FOREIGN KEY(MGR_ID) REFERENCES EMPL(EID),
        FOREIGN KEY(DEPT_ID) REFERENCES DEPT(DEPT_ID)
);
```

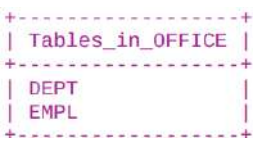
Schema or structure of table EMPL is as follows:

DESC EMPL;

Field	Type	Null	Key	Default	Extra
EID	varchar(6)	NO	PRI	NULL	
ENAME	varchar(30)	NO		NULL	
GEN	char(1)	YES		NULL	
DOJ	date	YES		NULL	
HOMETOWN	varchar(20)	YES		BANGALORE	
SALARY	decimal(8,2)	YES		NULL	
MGR_ID	varchar(6)	YES	MUL	NULL	
DEPT_ID	varchar(4)	YES	MUL	NULL	

Name of tables defined in current database so far.

SHOW TABLES;



```
+-----------------+
| Tables_in_OFFICE |
+-----------------+
| DEPT            |
| EMPL            |
+-----------------+
```

B. Write DML statements to insert records in two tables.

DML statements to insert records in DEPT are as follows:

INSERT INTO DEPT VALUES ('D01', 'FINANCE', 'MUMBAI', 20);

INSERT INTO DEPT VALUES ('D02', 'ADMIN', 'KOLKATA', 15);

INSERT INTO DEPT VALUES ('D03', 'IT', 'CHENNAI', 5);

DML statements to insert records in EMPL are as follows:

INSERT INTO EMPL VALUES ('E0001', 'RITU SEN', 'F', '2002-06-20', 'KOLKATA', 40000.00, NULL, 'D03');

INSERT INTO EMPL VALUES ('E0002', 'MALCOM RAY', 'M', '1998-11-12', 'BANGALORE', 50000.00, NULL, 'D02');

INSERT INTO EMPL(EID, ENAME, GEN, DOJ, HOMETOWN, SALARY, DEPT_ID) VALUES ('E0003', 'SUNDAR P', 'M', '2008-12-09', 'BANGALORE', 40000.00, 'D01');

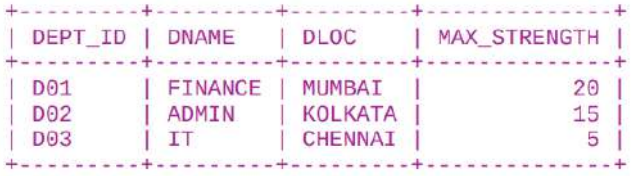
INSERT INTO EMPL VALUES ('E0004', 'ANISHA RAWAT', 'F', '2019-09-04', 'DELHI', 20000.00, 'E0001', 'D03');

INSERT INTO EMPL VALUES ('E0005', 'SANA KHAN', 'F', '2017-08-31', 'DELHI', 30000.00, 'E0003', 'D01');

C. Write SQL statements for the following queries and display their outputs.

1. Display all the records from table DEPT.

SELECT * FROM DEPT;



```
+---------+---------+---------+--------------+
| DEPT_ID | DNAME   | DLOC    | MAX_STRENGTH |
+---------+---------+---------+--------------+
| D01     | FINANCE | MUMBAI  |           20 |
| D02     | ADMIN   | KOLKATA |           15 |
| D03     | IT      | CHENNAI |            5 |
+---------+---------+---------+--------------+
```

2. Display name and salary of all the employeEs from table EMPL.

SELECT ENAME, SALARY
FROM EMPL;

ENAME	SALARY
RITU SEN	40000.00
MALCOM RAY	50000.00
SUNDAR P	40000.00
ANISHA RAWAT	20000.00
SANA KHAN	30000.00

4. Display DNAME in ascending order of MAX_STRENGTH.

SOLUTION 1	SOLUTION 2	OUTPUT
SELECT DNAME FROM DEPT ORDER BY MAX_STRENGTH;	**SELECT DNAME FROM DEPT ORDER BY MAX_STRENGTH ASC;**	DNAME IT ADMIN FINANCE

Note:

- ✓ Sorting in SQL is by default in ascending order of values be it numeric or alphabetical order. Hence ASC is default keyword and need not be used in ORDER BY statement.
- ✓ In case of arranging the output of query in descending order of values DESC keyword must be used in ORDER BY statement.

Comparison operators

= > < >= <= <> !=

4. Display name and gender of employees whose hometown is BANGALORE.

SOLUTION	OUTPUT
SELECT ENAME, GEN FROM EMPL WHERE HOMETOWN = 'BANGALORE';	ENAME | GEN MALCOM RAY | M SUNDAR P | M

5. Display the name of departments which are not located in KOLKATA.

SOLUTION 1	SOLUTION 2	OUTPUT
SELECT DNAME FROM DEPT WHERE DLOC <> KOLKATA';	**SELECT DNAME FROM DEPT WHERE DLOC != 'KOLKATA';**	DNAME FINANCE IT

6. Display name of employees and salary in descending order of names where DEPT_ID is not 'D03'.

SOLUTION	OUTPUT

SELECT ENAME, SALARY FROM EMPL WHERE DEPT_ID != 'D03' ORDER BY ENAME DESC;	+------------+----------+ | ENAME | SALARY | +------------+----------+ | SUNDAR P | 40000.00 | | SANA KHAN | 30000.00 | | MALCOM RAY | 50000.00 | +------------+----------+

7. Display EID, ENAME of employees whose DOJ is after January, 2015.

SOLUTION	OUTPUT
SELECT EID, ENAME FROM EMPL WHERE DOJ > '2015-01-31' ;	+-------+--------------+ | EID | ENAME | +-------+--------------+ | E0004 | ANISHA RAWAT | | E0005 | SANA KHAN | +-------+--------------+

[Note: DATE should be preferably mentioned in 'yyyy-mm-dd' format.]

Logical Operators

OR AND NOT

Logical operators are used in where clause. AND, OR are binary operations which require 2 conditions. NOT is unary operator which requires one condition only.

- **AND** : c1 and c2 → If both c1 and c2 are true the overall condition is true.
- **OR** : c1 or c2 → If at least one between c1 or c2 are true the overall condition is true.
- **NOT** : not c1 → If c1 is true the overall condition is false and vice versa.

BETWEEN: BETWEEN operator can be used as a substitute of and operation where the minimum and maximum value is to be checked for a single column.

8. Display the records of those employees whose salary is between 35000 and 45000.

SOLUTION1	SOLUTION2
SELECT * FROM EMPL WHERE SALARY >=35000 AND SALARY <=45000;	**SELECT * FROM EMPL WHERE SALARY BETWEEN 35000 AND 45000;**

```
+-------+----------+------+------------+-----------+----------+--------+---------+
| EID   | ENAME    | GEN  | DOJ        | HOMETOWN  | SALARY   | MGR_ID | DEPT_ID |
+-------+----------+------+------------+-----------+----------+--------+---------+
| E0001 | RITU SEN | F    | 2002-06-20 | KOLKATA   | 40000.00 | NULL   | D03     |
| E0003 | SUNDAR P | M    | 2008-12-09 | BANGALORE | 40000.00 | NULL   | D01     |
+-------+----------+------+------------+-----------+----------+--------+---------+
```

Checking a list of values

IN: IN operator is a substitute of OR operation(s) among equality checking of a single column with multiple values.

NOT IN: NOT IN operator is used for non-equality checking of a column with multiple values.

9. Display name and hometown of employees who belong to departments 'D01' or 'D02'.

SOLUTION 1	SOLUTION 2	OUTPUT
SELECT ENAME, HOMETOWN FROM EMPL WHERE DEPT_ID = 'D01' OR DEPT_ID = 'D02';	**SELECT ENAME, HOMETOWN FROM EMPL WHERE DEPT_ID IN ('D01', 'D02');**	+-----------+----------+ | ENAME | HOMETOWN | +-----------+----------+ | MALCOM RAY | BANGALORE | | SUNDAR P | BANGALORE | | SANA KHAN | DELHI | +-----------+----------+

10. Display EID and SALARY of employees whose half of salary is neither 10000 nor 20000.

SOLUTION 1	SOLUTION 2	OUTPUT
SELECT EID, SALARY FROM EMPL WHERE NOT (SALARY/2 = 10000 OR SALARY/2 = 20000);	**SELECT EID, SALARY FROM EMPL WHERE SALARY/2 NOT IN (10000, 20000);**	+-------+----------+ | EID | SALARY | +-------+----------+ | E0002 | 50000.00 | | E0005 | 30000.00 | +-------+----------+

Wildcard Characters

A string constant to be checked with a value stored in a column may have one or more characters missing in case of sub string checking. Such placeholder can be of two types:

_ → Replacement or placeholder of exactly one character in the string constant value. (underscore)

% → Replacement or placeholder of 0 or more characters in the string constant value.

LIKE: A string constant containing one or more wildcard characters can be checked for equality with LIKE operator only, not =.

NOT LIKE: Likewise NOT LIKE operator checks inequality checking with a string constant containing one or more wildcard characters. It cannot be done using <> or !=.

11. List the name of employees whose name starts with 'S' and have length at least 5.

SOLUTION	OUTPUT
SELECT ENAME FROM EMPL WHERE ENAME LIKE 'S____%'; **[Hints: 4 underscores i.e. _ after S]**	+-----------+ | ENAME | +-----------+ | SUNDAR P | | SANA KHAN | +-----------+

12. List the name of employees whose name ends with 'N' or does not contain 'M' in it.

SOLUTION	OUTPUT
SELECT ENAME **FROM EMPL** **WHERE ENAME LIKE '%N'** **AND ENAME NOT LIKE '%M%';**	+-----------+ | ENAME | +-----------+ | RITU SEN | | SANA KHAN | +-----------+

NULL checking

IS: IS is a special operator which is used to check absence of value i.e. NULL in a column as no other comparison operator can be used on NULL values.

IS NOT: Likewise, IS NOT is used to check the presence of values i.e. NOT NULL in a column.

13. Print ENAME and DEPT_ID of employees who do not have manager i.e. MGR_ID is blank.

SOLUTION	OUTPUT
SELECT ENAME, DEPT_ID **FROM EMPL** **WHERE MGR_ID IS NULL;**	+------------+---------+ | ENAME | DEPT_ID | +------------+---------+ | RITU SEN | D03 | | MALCOM RAY | D02 | | SUNDAR P | D01 | +------------+---------+

14. Print ENAME and DEPT_ID of employees who have manager i.e. MGR_ID is not empty.

SOLUTION	OUTPUT
SELECT ENAME, DEPT_ID **FROM EMPL** **WHERE MGR_ID IS NOT NULL;**	+--------------+---------+ | ENAME | DEPT_ID | +--------------+---------+ | ANISHA RAWAT | D03 | | SANA KHAN | D01 | +--------------+---------+

Display redundant or unique values

ALL: ALL keyword allows all the values occurring including duplicate values to be displayed in output. SQL allows duplicate values in output. ALL is by default used in SQL, so need not be used explicitly.

DISTINCT: By default, SQL does not remove any duplicate values in the output on its own. Hence DISTINCT keyword is used along with a column where redundant values need to be removed before displayed.

15. List the hometowns of all the employee (Including duplicate values).

SOLUTION 1	SOLUTION 2	OUTPUT
SELECT HOMETOWN FROM EMPL;	**SELECT ALL HOMETOWN FROM EMPL;**	+----------+ | HOMETOWN | +----------+ | KOLKATA | | BANGALORE | | BANGALORE | | DELHI | | DELHI | +----------+

16. List the name of places which are hometown of any employee. (No duplicate values)

SOLUTION	OUTPUT
SELECT DISTINCT HOMETOWN FROM EMPL;	+----------+ | HOMETOWN | +----------+ | KOLKATA | | BANGALORE | | DELHI | +----------+

Aggregate functions

SUM( ) AVG( ) MAX( ) MIN( ) COUNT( )

Aggregate or statistical functions can be used on a group of records.

Using GROUP BY clause: Display outputs regarding each group formed by the GROUP BY field.

Without using GROUP BY clause: Display output corresponding to the overall table may or may not be filtered by where clause.

For example, consider the following ITEM table:

ITEM_NAME	PRICE	TYPE
RICE	60	Crops
WHEAT	45	Crops
TEA		Leaves
RAJMA	300	Pulses

Functions	Query	Output	Explanation
SUM()	SELECT SUM(PRICE) FROM ITEM	405	60 + 45 + 300 = 405
AVG()	SELECT AVG(PRICE) FROM ITEM	135	(60 + 45 + 300) / 3 = 135
MAX()	SELECT MAX(PRICE) FROM ITEM	300	Maximum among 60, 45, 300 = 300
MIN()	SELECT MIN(PRICE) FROM ITEM	45	Minimum among 60, 45, 300 = 45
COUNT()	SELECT COUNT(PRICE) FROM ITEM SELECT COUNT(*) FROM ITEM	3 4	Number of records in output

GROUP BY: GROUP BY clause is used if statistical records of a table are to be displayed based on a field. Once the group is formed individual records cannot be accessed in that query. Several clusters or groups are formed based on the number of different values in the GROUP BY column present in the table.

For example, if GROUP BY is applied on TYPE field of ITEM table 3 groups are formed – Crops have 2 records, Leaves and Pulses have one record each.

Renaming field and table

AS is an optional keyword to rename a column a table in FROM clause or an expression on column(s) in SELECT clause. If there is blank space in alias then it must be surrounded by ' ' or " ".

- Column renaming is done for customized display of query output.
- Table renaming is done for giving convenient names to the tables in join operations for ease of access by programmers.

17. Display the number of distinct DLOC mentioned in table DEPT.

SOLUTION	OUTPUT
SELECT COUNT(DISTINCT DLOC) as 'NO. OF LOCATIONS' **FROM DEPT;**	+------------------+ | NO. OF LOCATIONS | +------------------+ | 3 | +------------------+

18. Display the earliest and latest DOJ of employees as per EMPL.

SOLUTION	OUTPUT
SELECT MIN(DOJ) 'EARLIEST', MAX(DOJ) 'LATEST' **FROM EMPL;**	+------------+------------+ | EARLIEST | LATEST | +------------+------------+ | 1998-11-12 | 2019-09-04 | +------------+------------+

19. Display the number of employees of each gender GEN.

SOLUTION	OUTPUT
SELECT GEN, COUNT(*) COUNT **FROM EMPL** **GROUP BY GEN;**	+------+-------+ | GEN | COUNT | +------+-------+ | F | 3 | | M | 2 | +------+-------+

20. Display the total SALARY paid by each department.

SOLUTION	OUTPUT
SELECT DEPT_ID, SUM(SALARY) 'TOTAL SALARY' **FROM EMPL** **GROUP BY DEPT_ID;**	+---------+--------------+ | DEPT_ID | TOTAL SALARY | +---------+--------------+ | D01 | 70000.00 | | D02 | 50000.00 | | D03 | 60000.00 | +---------+--------------+

HAVING: It is a conditional statement used along with group by clause only. It compares the values with the outcome of aggregate functions belonging to each group already formed by GROUP BY clause.

Difference between WHERE and HAVING:

WHERE	HAVING
Works on the entire table	Works on the groups formed by GROUP BY
Checks all records individually and filter the output	Checks the output of aggregate functions on each group and filter groups

21. Display the hometowns and no. of employees belonging to them if the headcount per hometown is at least 2.

SOLUTION	OUTPUT
SELECT HOMETOWN, COUNT(EID) 'NO OF EMPLOYEE' **FROM EMPL** **GROUP BY HOMETOWN** **HAVING COUNT(EID) >= 2;**	+-----------+----------------+ | HOMETOWN | NO OF EMPLOYEE | +-----------+----------------+ | BANGALORE | 2 | | DELHI | 2 | +-----------+----------------+

22. Display the number of employees working in each DEPT_ID excepting 'D01' where no. of employees in the DEPT_ID is more than 1.

SOLUTION	OUTPUT
SELECT DEPT_ID, COUNT(*) AS 'NO OF EMPLOYEE' **FROM EMPL** **WHERE DEPT_ID != 'D01'** **GROUP BY DEPT_ID** **HAVING COUNT(*) > 1;**	+---------+----------------+ | DEPT_ID | NO OF EMPLOYEE | +---------+----------------+ | D03 | 2 | +---------+----------------+

Cartesian Product

Cartesian product is performed on two tables and it produces all the combination of records in both tables. It does not require any common column.

If tables A, B have m, n columns and p, q records respectively then resultant table A x B has m+n columns and p x q records.

23. Perform Cartesian Product between EMPL and DEPT.

SOLUTION 1	SOLUTION 2	SOLUTION 3
SELECT * **FROM EMPL, DEPT;** [RECOMMENDED STATEMENT]	**SELECT *** **FROM EMPL INNER JOIN DEPT;**	**SELECT *** **FROM EMPL JOIN DEPT;**

EID	ENAME	GEN	DOJ	HOMETOWN	SALARY	MGR_ID	DEPT_ID	DEPT_ID	DNAME	DLOC	MAX_STRENGTH
E0001	RITU SEN	F	2002-06-20	KOLKATA	40000.00	NULL	D03	D01	FINANCE	MUMBAI	20
E0001	RITU SEN	F	2002-06-20	KOLKATA	40000.00	NULL	D03	D02	ADMIN	KOLKATA	15
E0001	RITU SEN	F	2002-06-20	KOLKATA	40000.00	NULL	D03	D03	IT	CHENNAI	5
E0002	MALCOM RAY	M	1998-11-12	BANGALORE	50000.00	NULL	D02	D01	FINANCE	MUMBAI	20
E0002	MALCOM RAY	M	1998-11-12	BANGALORE	50000.00	NULL	D02	D02	ADMIN	KOLKATA	15
E0002	MALCOM RAY	M	1998-11-12	BANGALORE	50000.00	NULL	D02	D03	IT	CHENNAI	5
E0003	SUNDAR P	M	2008-12-09	BANGALORE	40000.00	NULL	D01	D01	FINANCE	MUMBAI	20
E0003	SUNDAR P	M	2008-12-09	BANGALORE	40000.00	NULL	D01	D02	ADMIN	KOLKATA	15
E0003	SUNDAR P	M	2008-12-09	BANGALORE	40000.00	NULL	D01	D03	IT	CHENNAI	5
E0004	ANISHA RAWAT	F	2019-09-04	DELHI	20000.00	E0001	D03	D01	FINANCE	MUMBAI	20
E0004	ANISHA RAWAT	F	2019-09-04	DELHI	20000.00	E0001	D03	D02	ADMIN	KOLKATA	15
E0004	ANISHA RAWAT	F	2019-09-04	DELHI	20000.00	E0001	D03	D03	IT	CHENNAI	5
E0005	SANA KHAN	F	2017-08-31	DELHI	30000.00	E0003	D01	D01	FINANCE	MUMBAI	20
E0005	SANA KHAN	F	2017-08-31	DELHI	30000.00	E0003	D01	D02	ADMIN	KOLKATA	15
E0005	SANA KHAN	F	2017-08-31	DELHI	30000.00	E0003	D01	D03	IT	CHENNAI	5

JOIN

NATURAL JOIN: Natural join is a binary operator which works on two tables. They should have one column which have same name and domain. It a combination of Cartesian product and a where clause with equality checking on the common columns.

- Other conditions in that query are ANDed with the join condition.
- Natural join is mostly done on Foreign key field of one table and Primary key field of another table.
- If tables A, B have m, n columns and p, q records respectively then resultant table has m+n columns and minimum(p,q) records.

EQUI JOIN: Equi join is a join operation which works on the equality condition of values in two columns from two tables having similar data type. NATURAL JOIN, EQUI JOIN are said to be INNER JOIN.

24. Perform Natural Join between these two tables.

SOLUTION 1	SOLUTION 2
SELECT * FROM EMPL NATURAL JOIN DEPT;	**SELECT * FROM EMPL, DEPT WHERE EMPL.DEPT_ID = DEPT.DEPT_ID;** [RECOMMENDED STATEMENT]

EID	ENAME	GEN	DOJ	HOMETOWN	SALARY	MGR_ID	DEPT_ID	DEPT_ID	DNAME	DLOC	MAX_STRENGTH
E0001	RITU SEN	F	2002-06-20	KOLKATA	40000.00	NULL	D03	D03	IT	CHENNAI	5
E0002	MALCOM RAY	M	1998-11-12	BANGALORE	50000.00	NULL	D02	D02	ADMIN	KOLKATA	15
E0003	SUNDAR P	M	2008-12-09	BANGALORE	40000.00	NULL	D01	D01	FINANCE	MUMBAI	20
E0004	ANISHA RAWAT	F	2019-09-04	DELHI	20000.00	E0001	D03	D03	IT	CHENNAI	5
E0005	SANA KHAN	F	2017-08-31	DELHI	30000.00	E0003	D01	D01	FINANCE	MUMBAI	20

25. Display every ENAME and their corresponding DNAME.

SOLUTION	OUTPUT
SELECT ENAME, DNAME FROM EMPL, DEPT WHERE EMPL.DEPT_ID = DEPT.DEPT_ID;	ENAME | DNAME RITU SEN | IT MALCOM RAY | ADMIN SUNDAR P | FINANCE ANISHA RAWAT | IT SANA KHAN | FINANCE

26. List the name of employees who work in ADMIN department.

SOLUTION	OUTPUT
SELECT ENAME FROM EMPL AS E, DEPT AS D WHERE E.DEPT_ID = D.DEPT_ID AND DNAME = 'ADMIN';	ENAME MALCOM RAY

27. Display no. of employees working in those departments whose DLOC is CHENNAI.

SOLUTION	OUTPUT

SELECT COUNT(*) 'NO. OF EMPLOYEES' FROM EMPL AS E, DEPT AS D WHERE E.DEPT_ID = D.DEPT_ID AND DLOC = 'CHENNAI';	+------------------+ | NO. OF EMPLOYEES | +------------------+ | 2 | +------------------+

28. Display ENAME of employees who have manager along with that display ENAME of their corresponding manager as well.

SOLUTION	OUTPUT
SELECT E.ENAME 'EMPLOYEE', M.ENAME 'MANAGER' FROM EMPL E, EMPL M WHERE E.MGR_ID = M.EID;	+---------------+----------+ | EMPLOYEE | MANAGER | +---------------+----------+ | ANISHA RAWAT | RITU SEN | | SANA KHAN | SUNDAR P | +---------------+----------+

29. Display ENAME and the amount of bonus to be paid to that employee where bonus = 5000 + 5% of SALARY.

SOLUTION	OUTPUT
SELECT ENAME, SALARY, 5000 + 0.05 * SALARY 'BONUS' FROM EMPL;	+---------------+----------+------------+ | ENAME | SALARY | BONUS | +---------------+----------+------------+ | RITU SEN | 40000.00 | 7000.0000 | | MALCOM RAY | 50000.00 | 7500.0000 | | SUNDAR P | 40000.00 | 7000.0000 | | ANISHA RAWAT | 20000.00 | 6000.0000 | | SANA KHAN | 30000.00 | 6500.0000 | +---------------+----------+------------+

D. Write DML statements for the following purpose:

1. Assign DEPT_ID 'D03' to those employees who are presently working at 'D02'.

SOLUTION	OUTPUT
UPDATE EMPL SET DEPT_ID = 'D03' WHERE DEPT_ID = 'D02'; SELECT * FROM EMPL;	+------+-------------+-----+------------+-----------+---------+--------+---------+ | EID | ENAME | GEN | DOJ | HOMETOWN | SALARY | MGR_ID | DEPT_ID | +------+-------------+-----+------------+-----------+---------+--------+---------+ | E0001 | RITU SEN | F | 2002-06-20 | KOLKATA | 40000.00 | NULL | D03 | | E0002 | MALCOM RAY | M | 1998-11-12 | BANGALORE | 50000.00 | NULL | D03 | | E0003 | SUNDAR P | M | 2008-12-09 | BANGALORE | 40000.00 | NULL | D01 | | E0004 | ANISHA RAWAT | F | 2019-09-04 | DELHI | 20000.00 | E0001 | D03 | | E0005 | SANA KHAN | F | 2017-08-31 | DELHI | 30000.00 | E0003 | D01 | +------+-------------+-----+------------+-----------+---------+--------+---------+

2. Increase SALARY of all the employees by 10%.

SOLUTION	OUTPUT
UPDATE EMPL SET SALARY = 1.1 * SALARY; SELECT * FROM EMPL;	+------+-------------+-----+------------+-----------+---------+--------+---------+ | EID | ENAME | GEN | DOJ | HOMETOWN | SALARY | MGR_ID | DEPT_ID | +------+-------------+-----+------------+-----------+---------+--------+---------+ | E0001 | RITU SEN | F | 2002-06-20 | KOLKATA | 44000.00 | NULL | D03 | | E0002 | MALCOM RAY | M | 1998-11-12 | BANGALORE | 55000.00 | NULL | D03 | | E0003 | SUNDAR P | M | 2008-12-09 | BANGALORE | 44000.00 | NULL | D01 | | E0004 | ANISHA RAWAT | F | 2019-09-04 | DELHI | 22000.00 | E0001 | D03 | | E0005 | SANA KHAN | F | 2017-08-31 | DELHI | 33000.00 | E0003 | D01 | +------+-------------+-----+------------+-----------+---------+--------+---------+

3. Delete the department 'D02' from DEPT table.

SOLUTION	OUTPUT
DELETE FROM DEPT WHERE DEPT_ID = 'D02'; SELECT * FROM DEPT;	+---------+---------+---------+--------------+ | DEPT_ID | DNAME | DLOC | MAX_STRENGTH | +---------+---------+---------+--------------+ | D01 | FINANCE | MUMBAI | 20 | | D03 | IT | CHENNAI | 5 | +---------+---------+---------+--------------+

E. Write DDL statements for the following purpose:

1. Add DPHONE field to table DEPT which should be a number of 10 digits and unique for each department.

SOLUTION	OUTPUT
ALTER TABLE DEPT **ADD DPHONE INT(10) UNIQUE;** **DESC DEPT;**	+--------------+------------+------+-----+---------+-------+ | Field | Type | Null | Key | Default | Extra | +--------------+------------+------+-----+---------+-------+ | DEPT_ID | varchar(4) | NO | PRI | NULL | | | DNAME | varchar(15) | NO | | NULL | | | DLOC | varchar(20) | YES | | NULL | | | MAX_STRENGTH | int(2) | YES | | NULL | | | DPHONE | int(10) | YES | UNI | NULL | | +--------------+------------+------+-----+---------+-------+

2. Drop the column MAX_STRENGTH from DEPT.

SOLUTION	OUTPUT
ALTER TABLE DEPT **DROP MAX_STRENGTH;** **SELECT * FROM DEPT;**	+---------+---------+---------+---------+ | DEPT_ID | DNAME | DLOC | DPHONE | +---------+---------+---------+---------+ | D01 | FINANCE | MUMBAI | NULL | | D03 | IT | CHENNAI | NULL | +---------+---------+---------+---------+

3. Modify the datatype of SALARY in table EMPL to an integer of length 6 and drop the existing check constraint.

SOLUTION	OUTPUT
ALTER TABLE EMPL **MODIFY SALARY INT(6);** **DESC EMPL;**	+----------+-------------+------+-----+-----------+-------+ | Field | Type | Null | Key | Default | Extra | +----------+-------------+------+-----+-----------+-------+ | EID | varchar(6) | NO | PRI | NULL | | | ENAME | varchar(30) | NO | | NULL | | | GEN | char(1) | YES | | NULL | | | DOJ | date | YES | | NULL | | | HOMETOWN | varchar(20) | YES | | BANGALORE | | | SALARY | int(6) | YES | | NULL | | | MGR_ID | varchar(6) | YES | | NULL | | | DEPT_ID | varchar(4) | YES | | NULL | | +----------+-------------+------+-----+-----------+-------+

Questions ;

Q. No. 1 to 20 are MCQs of 1 mark each	
1.	An attribute in a table is foreign key if it is the _____key in any other table. a) Candidate b) Primary c) Unique d) Alternate
2.	What is the domain of an attribute? (a) The set of possible values that the attribute can take (b) The name of the attribute (c) The data type of the attribute (d) None of the above
3.	Which of the following is not a database constraint? a. CHECK b. DEFAULT c. UNIQUE d. NULL
4.	The data types CHAR (n) and VARCHAR (n) are used to create _____ and _____ types of string/text fields respectively in a database. a) Fixed, equal b) Equal, variable c) Fixed, variable d) Variable, equal
5.	Which of the following is a DDL command? A. UPDATE B. INSERT C. DELETE D. ALTER
6.	Which command is used to open the database "SCHOOL"? a. USE SCHOOL b. OPEN SCHOOL

	c. USE DATABASE SCHOOL d. SHOW SCHOOL
7.	In the given query which keyword has to be inserted? INSERT INTO employee_____(1002, "Kausar", 2000); a) Value b) Values c) Values into d) Into Values
8.	Which SQL statement is used to display all the data from PRODUCT table in the decreasing order of PRICE? a. SELECT * FROM PRODUCT ORDER PRICE BY DESC ; b. SELECT * FROM PRODUCT PRICE ORDER BY DESC; c. SELECT * FROM PRODUCT ORDER BY DESC PRICE; d. SELECT * FROM PRODUCT ORDER BY PRICE DESC;
9.	Which of the following function is used to FIND the largest value from the given data in MYSQL? a) MAX () b) MAXIMUM () c) LARGEST () d) BIG ()
10.	Which keyword is used for aliasing a table? a) ASC b) AS c) IS d) None of these
11.	_____ aggregate function does not ignore NULL values in a column. a) Min() b) Sum() c) Avg () d) Count ()
12.	SELECT name FROM class WHERE subject_____NULL; Which comparison operator may be used to fill the blank space in above query? a) = b) LIKE c) IS d) <>
13.	Which SQL statement is used to display all the data from ITEMS table where INAME is ending with 'L'? a. SELECT * FROM ITEMS WHERE INAME LIKE 'L%'; b. SELECT * FROM ITEMS WHERE INAME LIKE '%L'; c. SELECT * FROM ITEMS WHERE INAME LIKE '%L%'; d. SELECT * FROM ITEMS WHERE INAME LIKE '_L_';
14.	Which join combines each row from the first table with every row from the second table to make the result set? a. CROSS JOIN b. OUTER JOIN c. INNER JOIN d. EQUI JOIN
	State True / False for Q. No. 15 and 16
15.	MySQL statement to delete a table STUDENT from the database SCHOOL is DELETE TABLE STUDENT;
16.	Where and Having clauses can be used interchangeably in SQL queries.
	Q. No. 17 to 20 are ASSERTION (A) and REASONING (R) based questions. **Mark the correct choice as:** **a. Both A and R are true and R is the correct explanation for A.** **b. Both A and R are true and R is not correct explanation for A.** **c. A is true but R is false.** **d. A is false but R is true.**
17.	Assertion (A): A foreign key in the relational data model is a set of attributes in one relation that references the primary key of another relation. Reason (R): Foreign keys are used to establish relationships between tables.
18.	Assertion(A): DBMS is an application package which arranges the data in orderly manner in a tabular form. Reason(R): It is an interface between database and the user. It allows the users to access and perform various operations on stored data using some tools.
19.	Assertion(A): Aggregate function AVG() calculates the average of a set of values and produces a single value as result. Reason(R): The aggregate functions are used to perform some basic calculations

like sum, max, min, etc on a set of numbers.

20. Assertion(A): While inserting records in EMP table, value of DateOfBirth field must be enclosed withing quotes ‘ ‘.
Reasoning(R): Date is represented as char / varchar always.

Q. No. 21 to 30 are questions of 2 marks each

21. a) What do you mean by degree and cardinality of a table?
b) Consider the following table and find its degree and cardinality.

Customer_Details

Customer_id	Name	Address	Age
1	Billie	NY	22
2	Eilish	London	19
3	Ariana	Miami	18
4	Selena	New Jersey	32
5	Kety	Hawaii	42
6	Adele	Miami	29

22. What do you mean by referential integrity? Explain with suitable example.

23. Write MySQL statement to create a table named REMEDIAL based on the following specification:

Table: REMEDIAL

Attribute	Data type	Constraints
SNAME	VARCHAR(20)	NOT NULL
ROLL	INT	UNIQUE
FEES	FLOAT	
ADMN	INT	PRIMARY KEY

24. Write MySQL statements to do the following:
    a. Enter into the database OFFICE and display the name of all the tables.
    b. Display the structure of the table ‘EMPL’.

25. Consider the following table PLAYER:

PNO	PNAME	SPORTS	COUNTRY	SALARY
C001	VIRAT	CRICKET	INDIA	112
F003	RONALDO	FOOTBALL	PORTUGAL	1134
T007	ROGER	TENNIS	SWITZERLAND	795
B002	SINDHU	BADMINTON	INDIA	59

    a. Suggest the most suitable Primary Key. Justify your answer.
    b. Identify the alternate Keys.

26. Consider the table MOTOR having the following attributes:
VNO, MODEL, BRAND, PRICE

Write SQL statement for the following:
    a. Modify the existing column MODEL as varchar(30) and NOT NULL.
    b. Increase PRICE by 20% for the BRAND ‘TATA’

27.	Categorize the following commands as DDL or DML: INSERT, UPDATE, ALTER, DROP
28.	A MySQL table, sales have 10 rows. The following queries were executed on the sales table. SELECT COUNT(*) FROM sales; COUNT(*) 10 SELECT COUNT(discount) FROM sales; COUNT(discount) 6 Write a statement to explain as to why there is a difference in both the counts.
29.	Explain Group by clause with a suitable example.
30.	Distinguish between Natural Join and Equi Join.
	Q. No. 31 to 40 – 3 marks question
31.	Write output of the SQL queries based on the following table Employee: Table: Employee +-----+----------+------------+------------+--------+---------+ | EID | Name | DOB | DOJ | Salary | Project | +-----+----------+------------+------------+--------+---------+ | E01 | Ranjan | 1990-07-12 | 2015-01-21 | 150000 | P01 | | E02 | Akhtar | 1992-06-21 | 2015-02-01 | 125000 | P04 | | E03 | Muneera | 1996-11-15 | 2018-08-19 | 135000 | P01 | | E04 | Alex | 1991-10-25 | 2018-10-19 | 75000 | P02 | | E05 | Satyansh | 1993-12-16 | 2018-10-19 | 85000 | P04 | +-----+----------+------------+------------+--------+---------+ (i) select name, project from employee order by project; (ii) select name, salary from employee where doj like '2015%'; (iii) select min(doj), max(dob) from employee;
32.	Write output of the SQL queries based on the following tables Projects and Employee: Table: Projects +-----+----------------------+------------+------------+ | PID | PName | Startdate | Enddate | +-----+----------------------+------------+------------+ | P01 | Road 102 Carpeting | 2022-01-28 | 2022-02-26 | | P02 | Civil Lines Parking | 2022-01-30 | 2023-01-29 | | P03 | T-3 Renovation | 2022-03-16 | 2022-12-15 | | P04 | Footover Bridge K-13 | 2022-03-19 | 2023-02-01 | +-----+----------------------+------------+------------+ Table: Employee +-----+----------+------------+------------+--------+---------+ | EID | Name | DOB | DOJ | Salary | Project | +-----+----------+------------+------------+--------+---------+ | E01 | Ranjan | 1990-07-12 | 2015-01-21 | 150000 | P01 | | E02 | Akhtar | 1992-06-21 | 2015-02-01 | 125000 | P04 | | E03 | Muneera | 1996-11-15 | 2018-08-19 | 135000 | P01 | | E04 | Alex | 1991-10-25 | 2018-10-19 | 75000 | P02 | | E05 | Satyansh | 1993-12-16 | 2018-10-19 | 85000 | P04 | +-----+----------+------------+------------+--------+---------+ (i) select project, count(*) from employee group by project; (ii) select pid, pname, eid, name from projects p, employee e where p.pid=e.project; (iii) select avg(salary) from employee where doj between '2018-08-01' and '2018-08-31';

33. Write SQL statements to do the following:

COMPANY

CID	NAME	CITY	PRODUCTNAME
111	SONY	DELHI	TV
222	NOKIA	MUMBAI	MOBILE
333	ONIDA	DELHI	TV
444	SONY	MUMBAI	MOBILE
555	BLACKBERRY	MADRAS	MOBILE
666	DELL	DELHI	LAPTOP

CUSTOMER

CUSTID	NAME	PRICE	QTY	CID
101	ROHAN SHARMA	70,000	20	222
102	DEEPAK KUMAR	50,000	10	666
103	MOHAN KUMAR	30,000	5	111
104	SAHIL BANSAL	35,000	3	333
105	NEHA SONI	25,000	7	444
106	SONAL AGGARWAL	20,000	5	333
107	ARUN SINGH	50,000	15	666

i. Define CID in CUSTOMER table as Foreign Key that refers to CID i.e. Primary Key of COMPANY table.

ii. Display the 'CUSTOMER NAME', 'PRODUCT NAME' who have purchased any product from the 'COMPANY NAME' 'SONY'.

iii. Increase the QTY by 15 for the products with PRICE below 40,000.

34. Consider the GAMES table and answer the following questions:

Table: GAMES

GCode	GameName	Number	PrizeMoney	ScheduleDate
101	CaromBoard	2	5000	23-Jan-2004
102	Badminton	2	12000	12-Dec-2003
103	TableTennis	4	8000	14-Feb-2004
105	Chess	2	9000	01-Jan-2004
108	LawnTennis	4	25000	19-Mar-2004

i. Identify the possible Candidate Keys in the above table.
ii. Suggest the most suitable column for Primary key of the above table. Give reason behind your answer.
iii. Write down the Alternate Keys for the above table.

35. Consider a table STORE having attributes as following:

ItemNo –numeric
ItemName – character of size 20
Scode – numeric
Quantity – numeric

Abhay wants to do the following operations on the STORE table. Please help him to do by writing appropriate SQL statements.

i. Insert the following record in the STORE table:
(2010, Notebook, 23, NULL)
ii. Add a new column price with data type as decimal.

iii. Remove STORE table from the database.

36. What do you mean b y CHECK constraint and DEFAULT constraint? Explain with suitable example.

37. Consider the following tables and answer the questions below:

Table : PRODUCT

P_ID	ProductName	Manufacturer	Price
TP01	Talcom Powder	LAK	40
FW05	Face Wash	ABC	45
BS01	Bath Soap	ABC	55
SH06	Shampoo	XYZ	120
FW12	Face Wash	XYZ	95

Table : CLIENT

C_ID	ClientName	City	P_ID
01	Cosmetic Shop	Delhi	FW05
06	Total Health	Mumbai	BS01
12	Live Life	Delhi	SH06
15	Pretty Woman	Delhi	FW12
16	Dreams	Banglore	TP01

i. What will be the degree and cardinality of the resultant table after performing Cartesian Product between PRODUCT and CLIENT?
ii. What will be the degree and cardinality of the resultant table after performing NATURAL JOIN between PRODUCT and CLIENT?
iii. Are these values same? What can be the reason for this?

38. i. Write down the purpose of using aggregate functions in MySql.
ii. Give example of any two aggregate functions and their purposes.
iii. Can we use aggregate functions without GROUP BY clause? Justify.

39. i. What is the significance of NULL value in database?
ii. Name the operator that can check for NULL value in a column.
iii. Name the SQL command to permanently save the changes caused by DML statements in the database.

40. i. Name the aggregate functions valid on a column of DATE data type.
ii. Suggest a keyword for renaming an attribute or a table in MySql.
iii. Write down the syntax of representing the common column CODE while performing Equi Join between two tables GAME and USER.

Q. No. 41 to 45 – 5 marks question

41. i. Differentiate between 'WHERE' clause and 'HAVING' clause in MySQL with appropriate example.

ii. Consider the following table and find the output of the following queries:

TEACHER

TCODE	TNAME	SUBJECT	SEX	SALARY
5467	Narendra Kumar	Computer Science	M	70000
6754	Jay Prakash	Accountancy	M	Null
8976	Ajay Kumar	Chemistry	M	65000
5674	Jhuma Nath	English	F	55000
8756	Divya Bothra	Computer Science	F	75000
6574	Priyam Kundu	Physics	M	Null
3425	Dinesh Verma	Economics	M	71000

a. select SEX, avg(SALARY) from TEACHER group by SEX;

b. select SUBJECT, count(*) from TEACHER group by SUBJECT having count(*)>1;

c. select SUBJECT, min(SALARY) from TEACHER

	where TNAME not like '%Kumar' group by SUBJECT;
42.	i. Differentiate between DELETE and DROP in MySQL. Cite suitable examples. ii. Consider the following tables – Bank_Account and Branch: BANK_ACCOUNT E_CODE | NAME E01 | ASHISH E02 | SURESH BRANCH E_CODE | LOCATION E05 | MUMBAI What will be the output of the following statement? SELECT * FROM Bank_Account, Branch; iii. Choose the correct option: The above SQL query represents _____ operation. a. Outer join b. Natural join c. Equi join d. Cross join
43.	Fill in the blanks with appropriate keywords in order to complete the following SQL queries: CAR **CID** | **BRAND** | **MODEL** | **TYPE** | **COLOUR** | **PRICE** CO1 | MARUTI | SWIFT | HATCHBACK | WHITE | 5,00,000 D32 | HYUNDAI | VERNA | SEDAN | BLACK | 16,00,000 F32 | TATA | NEXON | SUV | RED | 9,00,000 C29 | KIA | SELTOS | SUV | BLUE | 17,00,000 i. Display all the databases present in MySQL of a system. show _____ ; ii. Display the values in TYPE column of the table CAR after removing the duplicate values from the output. select _____ TYPE from CAR; iii. Display MODEL, PRICE, COLOUR from CAR whose COLOUR is neither RED nor BLUE. select MODEL, PRICE, COLOUR from CAR where COLOUR _____ ('RED', 'BLUE'); iv. Display the total number of records present in CAR table. select _____(CID) from CAR; v. Display Discount on each CAR where Discount is 5% of the PRICE. select MODEL, PRICE*0.05 _____ 'DISCOUNT' from CAR;
44.	i. Write two advantages of using database. ii. Distinguish between CHAR and VARCHAR data type. Which one is preferable in general and why? iii. Write down the significance of the data type NUMERIC(7,3).
45.	i. What do you mean by Self Join. Give an example. ii. Fill in the blanks with appropriate keywords for creating the table DRESS with the following specifications: • Default COLOR is 'BLACK'. • PRICE between 0 and 8000.

CREATE _____(a)_____ DRESS
(
DCODE INT PRIMARY KEY,
DNAME VARCHAR(15),
COLOR VARCHAR(10) ____(b)____ 'BLACK',
PRICE DECIMAL(6, 2) _____(c)____ PRICE BETWEEN 0 and 8000
);

Q. No. 46 to 50 – Case Study based questions of 4 marks each

46. Consider the following tables ITEM and CUSTOMER and find the outputs of the following queries:

Table: ITEM

ID	Item_Name	Manufacturer	Price
PC01	Personal Computer	ABC	35000
LC05	Laptop	ABC	55000
PC03	Personal Computer	XYZ	32000
PC06	Personal Computer	COMP	37000
LC03	Laptop	PQR	57000

Table: CUSTOMER

C_ID	CName	City	ID
01	N Roy	Delhi	LC03
06	R Singh	Mumbai	PC03
12	R Pandey	Delhi	PC06
15	C Sharma	Delhi	LC03
16	K Agarwal	Bangalore	PC01

i) SELECT ITEM_NAME, MAX(PRICE), COUNT(*) FROM ITEM GROUP BY ITEM_NAME;

ii) SELECT CNAME, MANUFACTURER FROM ITEM, CUSTOMER WHERE ITEM.ID=CUSTOMER.ID;

iii) SELECT ITEM_NAME, PRICE*100 FROM ITEM WHERE MANUFACTURER="ABC";

iv) SELECT DISTINCT CITY FROM CUSTOMER;

47. Consider the tables ITEM and CUSTOMER and write the queries:

TABLE: SCHOOL

CODE	TEACHERNAME	SUBJECT	DOJ	PERIODS	EXPERIENCE
1001	RAVI SHANKAR	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

(i) Display the total PERIODS for each SUBJECT from SCHOOL table.

(ii) Display TEACHERNAME, GENDER from the tables SCHOOL and ADMIN whose DESIGNATION is 'COORDINATOR'.

(iii) Display TEACHERNAME and DOJ in the descending order of CODE.

(iv) Display TEACHERNAME whose DOJ is in the year 2000.

48. Modern Public School is maintaining fees records of students. The database administrator Aman decided that-

- Name of the database -School
- Name of the table – Fees
- The attributes of Fees are as follows:
  - Rollno – numeric
  - Name – character of size 20
  - Class - character of size 20
  - Fee – Numeric
  - PayDate – Date
  - Primary Key – (Rollno, Class)

Answer the following questions:

(i) Write the DDL statement to create database School.
(ii) Write the SQL statement to create Fees table in School database with the above-mentioned specifications.
(iii) Write SQL statement to display all the table names in School database.

49. Consider the table Fees mentioned in Q. No. 48 and answer the following questions:

i. Insert the following record into the table
Rollno-1201, Name-Akshay, Class-12th, Fee-350, PayDate-24 JUNE 2019

ii. Increase the second quarter fee of class 12th students by 50.
iii. Delete the record of student with Rollno-1212
iv. Aman wants to display the schema (structure) of Fees table. Which command will he use from the following:

a) CREATE b) ALTER c) SHOW d) DESCRIBE

50. Sagar, a cloth merchant creates a table CLIENT with a set of records to maintain the client's order volume in Qtr1, Qtr2, Qtr3 and their total. After creation of the table, he has entered data of 7 clients in the table.

CLIENT

ClientName	Client_ID	Qtr1	Qtr2	Qtr3	Total
Suraj	C120	200	300	400	900
Radha	C650	190	356	220	766
Estha	C430	200	100	400	700
Karuna	C790	130	540	380	1050
Naresh	C660	200	400	800	1400
Varun	C233	400	300	220	920
Kritika	C540	500	100	400	1000

Based on table CLIENT, write SQL statements for the following:

i. Write the statements to Update a record present in the table with data for Qtr2 = 200, Qtr3 = 600 , total = sum of all Qtrs where the Client_ID is C660.
ii. Delete all records where total is between 500 to 900.

	iii. Make changes in ClientName with data type varchar(20) and not null constraint. iv. Remove the column Total from the CLIENT table.

SOLUTIONS

1. B	2. a	3. d	4. c	5. d
6. a	7. b	8. d	9. a	10. b
11. d	12. c	13. b	14. a	15. False
16. False	17. a	18. a	19. b	20. c

21. a) Degree - no. of attributes in a table, Cardinality – no. of records in a table.
b) Degree – 4, cardinality – 6

22. Foreign key of one table refers to the Primary key of another table.

23.
```
CREATE TABLE REMEDIAL
(
        SNAME VARCHAR(20) NOT NULL,
        ROLL INT(5) UNIQUE,
        FEES FLOAT(7,2),
        ADMN INT(5) PRIMARY KEY
);
```

24. a. USE OFFICE;
b. DESC EMPL; or DESCRIBE EMPL;

25. a. PNO as unique throughout table and not null.
b. PNAME, SPORTS, SALARY.

26. a. ALTER TABLE MOTOR MODIFY MODEL VARCHAR(30) NOT NULL;
b. UPDATE MOTOR SET PRICE = PRICE * 1.20 WHERE BRAND = 'TATA';

27. DML - INSERT, UPDATE
DDL - ALTER, DROP

28. Count(*) will return the number of records in the table sales. Count(discount) will return the number of records having not null values in the discount field of sales table.

29. GROUP BY clause is used if statistical records of a table are to be displayed based on a field. Groups are formed based on the number of different values in the GROUP BY column present in the table.

30. In Natural Join the common attribute between two tables appears only once. Where as in Equi Join the common attribute appears as it is i.e. twice. Hence these common attributes are accessed as table_name.attribute in the query to resolve the conflict.

31. i.

Name	**Project**
Ranjan	P01
Muneera	P01
Alex	P02
Akhtar	P04
Satyansh	P04

ii.

Name	**Salary**
Ranjan	150000
Akhtar	125000

iii.

min(DOJ)	**max(DOB)**
2015-01-21	1996-11-15

<table>
<tr><td></td><td></td></tr>
<tr><td>32.</td><td>i.
<table><tr><th>Project</th><th>count(*)</th></tr><tr><td>P01</td><td>2</td></tr><tr><td>P04</td><td>2</td></tr><tr><td>P02</td><td>1</td></tr></table></td></tr>
<tr><td></td><td>ii.
<table><tr><th>PID</th><th>PName</th><th>EID</th></tr><tr><td>P01</td><td>Road 102 Carpenting</td><td>E01</td></tr><tr><td>P04</td><td>Footover Bridge K-13</td><td>E02</td></tr><tr><td>P01</td><td>Road 102 Carpenting</td><td>E03</td></tr><tr><td>P02</td><td>Civil Lines Parking</td><td>E04</td></tr><tr><td>P04</td><td>Footover Bridge K-13</td><td>E05</td></tr></table>
iii.
<table><tr><td>avg(Salary)</td></tr><tr><td>135000</td></tr></table></td></tr>
<tr><td>33.</td><td>i. ALTER TABLE CUSTOMER
ADD FOREIGN KEY(CID) REFERENCES COMPANY(CID);

ii. SELECT CU.NAME, CO.PRODUCTNAME
FROM COMPANY CO, CUSTOMER CU
WHERE CU.CID = CO.CID AND CO.NAME = ‘SONY’;

iii. UPDATE CUSTOMER SET QTY = QTY*1.15 where PRICE<40000;</td></tr>
<tr><td>34.</td><td>i. Candidate keys - GCode, GameName, PrizeMoney, ScheduleDate
ii. Primary key - GCode as not null and unique for each game.
iii. Alternate key - GameName, PrizeMoney, ScheduleDate</td></tr>
<tr><td>35.</td><td>i. INSERT INTO STORE VALUES (2010, ‘Notebook’, 23, NULL);
ii. ALTER TABLE STORE ADD PRICE DECIMAL(10,2);
iii. DROP TABLE STORE;</td></tr>
<tr><td>36.</td><td>CHECK – Ensure that the attribute contains only permissible set of values.
DEFAULT – Ensure the default value is inserted if no value is mentioned.
e.g. -
CREATE TABLE STOCK
(
SNO INT PRIMARY KEY,
SNAME VARCHAR(20),
LOCATION VARCHAR(15) DEFAULT ‘BANGALORE’,
PRICE FLOAT(7,2) CHECK (PRICE BETWEEN 0.00 AND 10000.00)
)</td></tr>
<tr><td>37.</td><td>i. After Cartesian product, Degree = 8, Cardinality = 25
ii. After natural join, Degree = 7, Cardinality = 5
iii. No, because cartesian product is the all-possible combination of tuples between two tables. Where as Natural join selects only those tuples for whom the values of the common attributes are same.</td></tr>
<tr><td>38.</td><td>i. Aggregate functions perform calculation on a set of values, and returns a single</td></tr>
</table>

<table>
<tr><td></td><td>value. If used with GROUP BY clause, it returns one value for each group.
SUM() - returns the total sum of a numerical column
MAX() - returns the largest value within the selected column

ii. Yes. Then it returns a single value for the selected attribute by considering all the records in that table.</td></tr>
<tr><td>39.</td><td>i. NULL is said to be absence of any value in an attribute. NULL cannot participate in any operation.
ii. IS
iii. COMMIT</td></tr>
<tr><td>40.</td><td>i. MAX(), MIN(), COUNT()
ii. AS
iii. SELECT * FROM GAME G, USER U WHERE G.CODE=U,CODE;</td></tr>
<tr><td>41.</td><td>i. WHERE clause allows to filter data from individual rows of a table based on certain conditions. In contrast, the HAVING clause allows to filter data from a group of rows in a query based on conditions involving aggregate functions.
ii.
a)
<table><tr><th>SEX</th><th>AVG(SALARY)</th></tr><tr><td>M</td><td>68666</td></tr><tr><td>F</td><td>65000</td></tr></table>
b)
<table><tr><th>SUBJECT</th><th>COUNT(*)</th></tr><tr><td>Computer Science</td><td>2</td></tr></table>
c)
<table><tr><th>SUBJECT</th><th>MIN(SALARY)</th></tr><tr><td>Computer Science</td><td>75000</td></tr><tr><td>English</td><td>55000</td></tr><tr><td>Economics</td><td>71000</td></tr></table></td></tr>
<tr><td>42.</td><td>i. DELETE is used for deleting records from a table. DROP is used to delete the entire schema of any database object like table.
e.g. –
DELETE FROM STUDENT WHERE ROLL = 5;
DROP TABLE STUDENT;
ii.
<table><tr><td>E_CODE</td><td>NAME</td><td>E_CODE</td><td>LOCATION</td></tr><tr><td>E01</td><td>ASHISH</td><td>E05</td><td>MUMBAI</td></tr><tr><td>E02</td><td>SURESH</td><td>E05</td><td>MUMBAI</td></tr></table>iii. d. Cross join</td></tr>
<tr><td>43.</td><td>i. databases
ii. distinct
iii. not in
iv. count
v. as</td></tr>
<tr><td>44.</td><td>i. Data integrity, data security
ii. Char data type stores data of fixed length, whereas the Varchar data type stores variable length data. Varchar is preferable as it is more flexible for data of any size.
iii. It can represent 7 digit real number with 3 digits in the right of decimal point.</td></tr>
<tr><td>45.</td><td>i. A self-join is a regular join, but the table is joined with itself.</td></tr>
</table>

SELECT * FROM EMP A, EMP B where A.ID = B.ID;

ii.

(a) TABLE
(b) DEFAULT
(c) CHECK

46. i.

ITEM_NAME	MAX(PRICE)	COUNT(*)
Personal Computer	37000	3
Laptop	57000	2

ii.

CNAME	MANUFACTURER
N Roy	PQR
R Singh	XYZ
R Pandey	COMP
C Sharma	PQR
K Agarwal	ABC

iii.

ITEM_NAME	PRICE*100
Personal Computer	3500000
Laptop	5500000

iv.

City
Delhi
Mumbai
Bangalore

47. (i) SELECT SUM (PERIODS), SUBJECT FROM SCHOOL
GROUP BY SUBJECT;

(ii) SELECT TEACHERNAME, GENDER FROM SCHOOL, ADMIN WHERE DESIGNATION = 'COORDINATOR' AND SCHOOL.CODE=ADMIN.CODE;

(iii) SELECT TEACHERNAME, DOJ FROM SCHOOL ORDER BY CODE DESC;

(iv) SELECT TEACHERNAME FROM SCHOOL WHERE DOJ LIKE '%2000';

48. i. CREATE DATABASE SCHOOL;
ii. USE SCHOOL

```
create table Fees
(
        Rollno numeric(5),
        Name varchar(20),
        Class varchar(20),
        Fee Numeric(7,2),
        PayDate Date,
        Primary Key(Rollno, Class)
);
```

iii. SHOW TABLES

49. i. INSERT INTO FEES VALUES(1201, 'Akshay', '12th', 350, '2019-06-24');
ii. UPDATE TABLE FEES SET FEE = FEE+50 WHERE CLASS='12th';