SQL

1. Table Name: PERSONS

P_ID	LASTNAME	FIRSTNAME	ADDRESS	CITY
1	HANSEN	OLA	TIMOTEIVN 10	SANDNES
2	SVENDSON	TOVE	BORGVN23	SANDNES
3	PETTERSEN	KARI	STORGT20	STAVANGER

a. Write a SQL statement to create a table Persons and insert the values.

b. To select the content of the columns named LASTNAME AND FIRSTNAME from the PERSONS table.

c. To select the all columns from the PERSONS table.

d. To select only the persons living in the city “SANDNES” from the table PERSONS.

e. To select only PERSONS with first name “TOVE” And last name “SVENDSON”

f. To select only Persons with first name “TOVE” OR “OLA”.

2. Table Name: EMPLOYEE

E_ID	SALARY	BONUS
101	2000	200
102	3000	150
103	1000	100
104	2050	125
105	3500	200

a. To find total salary + Bonus from Employee table.

b. Using select statement Subtraction (-) operator used in SALARY & BONUS table.

3. Answer the a) & b) question on the basic of the following tables SHOPPE and ACCESSORIES.

TABLE: SHOPPE

ID	SNAME	AREA
S01	ABC COMPUTERONICS	CP
S02	ALL INFOTECH MEDIA	GK II
S03	TECH SHOPPE	CP
S04	GEEKS TECNO SOFT	Nehru Place
S05	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 & 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 ID IN(‘S02’, ‘S03’);

4. TABLE: ITEMS

CODE	INAME	QTY	PRICE	COMPANY	TCODE
1001	DIGITAL PAD 121	120	11000	XENITA	T01
1006	LED SCREEN 40	70	38000	SANTORA	T02
1004	CAR GPS SYSTEM	50	2150	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 and quantity of those items, which have quantity more than 150.

SQL

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

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

g) The expected output of these query should be:

i)

MAX(price)	MIN(price
38000	1200

ii)

AMOUNT
107500

iii)

TCODE
T01
T02
T03

iv)

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

5. **TABLE NAME: 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	ARLUNA	30000	F	A03	2012
2109	SHAKTI	35000	M	A04	2011
1101	KIRAT	25000	M	A01	2012

TABLE NAME: COURSES

C_ID	COURSE
A01	FASHION DESIGN
A02	NETWORKING
A03	HOTEL MANAGEMENT
A04	EVENT MANAGEMENT
A05	OFFICE MANAGEMENT

a) To display NAME, FEE, GENDER, JOINYEAR about the APPLICANTS, who have joined before 2010.

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

c) To display the names of all applicants in ascending order of their JOINYEAR.

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

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

f) To display the applicant's name with their respective courses name from the tables applicants and courses.

g) Give the SQL statements of the outputs:

i)

NAME	JOINYEAR
AVISHA	2009

ii)

MIN(JOINYEAR)
2009

iii)

AVG(FEE)
31666.666

iv)

SUM(FEE)	C_ID
55000	A01

NOTES:

The SQL DELETE Statement:

The DELETE statement is used to delete existing records in a table.

Syntax: DELETE FROM table_name WHERE condition;

The SQL WHERE Clause

The WHERE clause is used to filter records. The WHERE clause is used to extract only those records that fulfill a specified condition.

Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

The SQL SELECT DISTINCT Statement: The SELECT DISTINCT statement is used to return only distinct (different) values. Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

Syntax:

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

The SQL SELECT Statement: The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set.

SQL

Syntax: SELECT * FROM table_name;

6. **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	FEMALE	101	WEST
1006	R DUBEY	MALE	104	NORTH
1007	M AGARWAL	FEMALE	104	NORTH
1008	S DAS	FEMALE	103	SOUTH
1009	R K PATIL	MALE	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

a) To display CNAME, AREA of all female customers from CUSTOMER table.

b) To display the details of all the CUSTOMERS in ascending order of CNAME within SID.

c) To display the total number of customers for each AREA from CUSTOMER table.

d) To display CNAME and CORRESPONDING SHOP from CUSTOMER table and ONLINESHOP table.

e) Write the code to see the output:

COUNT(*)	GENDER
4	FEMALE
6	MALE

f) Write the code to see the output:

COUNT(*)
4

g) SELECT CNAME FROM CUSTOMER WHERE CNAME LIKE ‘L%’; [Write the output]

h) SELECT DISTINCT AREA FROM CUSTOMER; [Write the output]

NOTES

UPDATE: The UPDATE statement is used to modify the existing records in a table.

Syntax: UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;

SQL

AND, OR and NOT Operators: The WHERE clause can be combined with AND, OR, and NOT operators.

The AND & OR operators are used to filter records based on more than one condition:
The AND operator displays a record if all the conditions separated by AND are TRUE.
The OR operator displays a record if any of the conditions separated by OR is TRUE.
The NOT operator displays a record if the condition(s) is NOT TRUE.

AND Syntax:

```
SELECT column1, column2, ... FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

OR Syntax:

```
SELECT column1, column2, ... FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

NOT Syntax:

```
SELECT column1, column2, ... FROM table_name
WHERE NOT condition;
```

7. **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	MERECEDES	RED	4	35

TABLE: CUSTOMER

CODE	CNAME	CCODE
1001	HAMANT SAHU	501
1002	RAJ LAL	509
1003	FEROZA SHAH	503
1004	KETAN DHAL	502

a)
i) To display the name of all the SILVER colored cars.
ii) To display name of Car, Make and sitting Capacity of cars in descending order of their sitting CAPACITY.
iii) To display the highest charges at which a vehicle can be hired from CARDEN.
iv) To display the CUSTOMER name and the corresponding name of the CARS hired by them.
b) Give the output of following SQL queries:
i) SELECT COUNT(DISTINCT MAKE) FROM CARDEN;
ii) SELECT MAX(CHARGES), MIN(CHARGES) FROM CARDEN;

iii) SELECT COUNT(*) MAKE FROM CARDEN;
iv) SELECT CARNAME FROM CARDEN WHERE CAPACITY=4;

SQL RENAME TABLE: SQL RENAME TABLE syntax is used to change the name of a table. Sometimes, we choose non-meaningful name for the table. So it is required to be changed.
Syntax: ALTER TABLE table_name RENAME TO new_table_name;

SQL COPY TABLE: If you want to copy a SQL table into another table in the same SQL server database, it is possible by using the select statement.
Syntax: SELECT * INTO <destination_table> FROM <source_table> ;

SQL ALTER TABLE DROP Column Syntax:
ALTER TABLE table_name DROP COLUMN column_name;

SQL ALTER TABLE Add Column Syntax:
ALTER TABLE table_name ADD column_name datatype;

SQL SELECT FIRST: The SQL first() function is used to return the first value of the selected column. Syntax: SELECT FIRST(column_name) FROM table_name;[The SELECT FIRST statement is only supported by MS Access]

SQL SELECT LAST: The last() function is used to return the last value of the specified column. Syntax: SELECT LAST (column_name) FROM table_name;[;[The SELECT LAST statement is only supported by MS Access]

SQL SELECT SUM Syntax: SELECT SUM (expression) FROM tables WHERE conditions;

8. Create a table to store information about weather observation stations. [No duplicate ID fields allowed]

```
CREATE TABLE STATION
(ID INTEGER PRIMARY KEY,
CITY CHAR(20),
STATE CHAR(2),
LAT_N REAL,
LONG_W REAL);
```

Populate the table STATION with a few rows:

```
INSERT INTO STATION VALUES (13, 'Phoenix', 'AZ', 33, 112);
INSERT INTO STATION VALUES (44, 'Denver', 'CO', 40, 105);
INSERT INTO STATION VALUES (66, 'Caribou', 'ME', 47, 68);
```

1.

i) Display the all stations.

ii) Query to select Northern stations (Northern latitude > 39.7).

iii) Query to select only ID, CITY, and STATE columns.

2. Create another table to store normalized temperature and precipitation data:

```
-- ID field must match some STATION table ID(so name and location will be known).
-- Allowable ranges will be enforced for other values.
-- No duplicate ID and MONTH combinations.
-- Temperature is in degrees Fahrenheit.
-- Rainfall is in inches.
CREATE TABLE STATS
(ID INTEGER REFERENCES STATION(ID),
MONTH INTEGER CHECK (MONTH BETWEEN 1 AND 12),
TEMP_F REAL CHECK (TEMP_F BETWEEN -80 AND 150),
RAIN_I REAL CHECK (RAIN_I BETWEEN 0 AND 100),
PRIMARY KEY (ID, MONTH));

INSERT INTO STATS VALUES (13, 1, 57.4, 0.31);
INSERT INTO STATS VALUES (13, 7, 91.7, 5.15);
INSERT INTO STATS VALUES (44, 1, 27.3, 0.18);
INSERT INTO STATS VALUES (44, 7, 74.8, 2.11);
INSERT INTO STATS VALUES (66, 1, 6.7, 2.10);
INSERT INTO STATS VALUES (66, 7, 65.8, 4.52);
```

i) Display the all values of STATS table.

ii) Query to look at table STATS, picking up location information by joining with table STATION on the ID column.

iii) Query to look at the table STATS, ordered by month and greatest rainfall, with columns rearranged.

iv) Query to look at temperatures for July from table STATS, lowest temperatures first, picking up city name and latitude by joining with table STATION on the ID column.

v) Query to show MAX and MIN temperatures as well as average rainfall for each station.

vi) Query (with sub query) to show stations with year-round average temperature above 50 degrees.

vii) Create a view (derived table or persistent query) to convert Fahrenheit to Celsius and inches to centimeters.

viii) Query to look at table STATS in a metric light (through the new view).

ix) Another metric query restricted to January below-freezing (0 Celsius) data, sorted on rainfall.

x) Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches low.

Create a database using the CREATE DATABASE command:

```
CREATE DATABASE database_name;
```

Insert data multiple rows, use a comma to separate each row, like this:

```
INSERT INTO table_name
VALUES
(value_1, value_2, value_3),
(value_1, value_2, value_3),
(value_1, value_2, value_3),
(value_1, value_2, value_3);
```

SQL PRIMARY KEY: Primary keys must contain UNIQUE values, and cannot contain NULL values.

SQL PRIMARY KEY on CREATE TABLE EXMAPLE:

MySQL:	**SQL Server / Oracle / MS Access**
CREATE TABLE Persons (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int, PRIMARY KEY (ID));	CREATE TABLE Persons (ID int NOT NULL PRIMARY KEY, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int);

9. Create table Worker.

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
001	NIHARIKA	ARORA	20000	2013-02-25 09:00:00	HR
002	AYUSHI	GURONDIA	5000	2015-02-10 09:00:00	ADMIN
003	PRIYANSHA	CHOUKSEY	25000	2014-05-16 09:00:00	HR
004	APARNA	DESHPANDE	8000	2016-12-20 09:00:00	ADMIN
005	SHAFALI	JAIN	21000	2015-08-29 09:00:00	ADMIN
006	SUCHITA	JOSHI	20000	2017-02-12 09:00:00	ACCOUNT

007	SHUBHI	MISHRA	15000	2018-03-23 09:00:00	ADMIN
008	DEVYANI	PATIDAR	18000	2014-05-02 09:00:00	ACCOUNT

1. i) Query to create the Table

```
CREATE TABLE Worker (
WORKER_ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
FIRST_NAME CHAR(25),
LAST_NAME CHAR(25),
SALARY INT(15),
JOINING_DATE DATETIME,
DEPARTMENT CHAR(25));
```

ii) Query to insert values into the Table Worker.

iii) Create table **Bonus**

WORKER_REF_ID	BONUS_DATE	BONUS_AMOUNT
1	2015-04-20 00:00:00	5000
2	2015-08-11 00:00:00	3000
3	2015-04-20 00:00:00	4000
1	2015-04-20 00:00:00	4500
2	2015-08-11 00:00:00	3500

```
CREATE TABLE Bonus(
WORKER_REF_ID INT,
BONUS_DATE DATETIME,
BONUS_AMONT INT(10),
FOREIGN KEY (WORKER_REF_ID)
REFERENCES WORKER(WORKER_ID)
ON DELETE CASCADE);
```

iv) Query to insert values into table Bonus.

v) Create table **Title**

WORKER_REF_ID	WORKER_TITLE	AFFECTED_FROM
1	Manager	2016-02-20 00:00:00
2	Executive	2016-06-11 00:00:00
8	Executive	2016-06-11 00:00:00
5	Manager	2016-06-11 00:00:00
4	Asst. Manager	2016-06-11 00:00:00
7	Executive	2016-06-11 00:00:00
6	Lead	2016-06-11 00:00:00
3	Lead	2016-06-11 00:00:00

```
CREATE TABLE Title (
WORKER_REF_ID INT,
WORKER_TITLE CHAR(25),
AFFECTED_FROM DATETIME,
FOREIGN KEY (WORKER_REF_ID)
REFERENCES Worker(WORKER_ID)
ON DELETE CASCADE);
```

vi) Query to insert values into table Title.

2. i) Write an SQL query for fetching “FIRST_NAME” from the WORKER table using <WORKER_NAME> as alias.

ii) What is an SQL Query for fetching the “FIRST_NAME” from WORKER table in upper case?

iii) What is an SQL query for fetching the unique values of the column DEPARTMENT from the WORKER table?

iv) Write an SQL query for printing the first three characters of the column FIRST_NAME.

v) What is an SQL query for finding the position of the alphabet (‘A’) in the FIRST_NAME column of Ayushi.

vi) What is an SQL Query for printing the FIRST_NAME from Worker Table after the removal of white spaces from right side.

vii) Write an SQL Query for printing the DEPARTMENT from Worker Table after the removal of white spaces from the left side.

viii) What is an SQL query for fetching the unique values from the DEPARTMENT column and thus printing is the length?

ix) Write an SQL query for printing the FIRST_NAME after replacing ‘A’ with ‘a’.

x) What is an SQL query for printing the FIRST_NAME and LAST_NAME into a column named COMPLETE_NAME? (A space char should be used)

xi) What is an SQL query for printing all details of the worker table which ordered by FIRST_NAME ascending?

xii) Write an SQL query for printing the all details of the worker table which ordered by FIRST_NAME ascending and the DEPARTMENT in descending

xiii) What is an SQL query to print the details of the workers ‘NIHARIKA’ and ‘PRIYANSHA’.

xiv) What is an SQL query printing all details of workers excluding the first names of ‘NIHARIKA’ and ‘PRIYANSHA’

xv) Write an SQL query for printing the details of DEPARTMENT name as “Admin”.

xvi) What is an SQL query for printing the details of workers whose FIRST_NAME Contains ‘A’?

xvii) What is an SQL Query for printing the FIRST_NAME of workers whose name ends with ‘A’?

xviii) What is an SQL Query for printing the details of the workers whose FIRST_NAME ends with ‘H’ and contains six alphabets?

xix) Write an SQL Query for printing the details of workers whose SALARY lies between 10000 and 20000.

xx) Write an SQL Query for printing the details of workers who joined inFeb’2014

SQL

xxi) Write an SQL Query for fetching the count of workers in DEPARTMENT with ‘Admin’.
xxii) Write an SQL Query for fetching the details of workers with Salaries >= 5000 and <= 10000.
xxiii) What is an SQL Query for fetching the no. of workers in each department in descending order?
xxiv) What is an SQL Query for printing the details of workers who are also managers?
xxv) Write an SQL Query for fetching the details of duplicate records in some fields.
xxvi) What is an SQL Query for only showing odd rows?
xxvii) What is an SQL Query for only showing even rows?
xxviii) Write an SQL Query for cloning a new table from another table.
xxix) Write an SQL Query for fetching the intersecting details of two tables.
xxx) What is an SQL Query for showing the details of one table that another doesn’t have.

Drop Column Syntax:

MYSQL	Oracle and SQL Server
ALTER TABLE "table_name" DROP "column_name";	ALTER TABLE "table_name" DROP COLUMN "column_name";

Rename Column Syntax:

MySQL	Oracle
ALTER TABLE "table_name" Change "column 1" "column 2" ["Data Type"];	ALTER TABLE "table_name" RENAME COLUMN "column 1" TO "column 2";

MySQL Numeric Functions:

ABS() - SELECT ABS(-243.5);	CEIL() - SELECT CEIL(35.75);
ACOS() - SELECT ACOS(0.5);	CEILING() - SELECT CEILING(25.75);
ATAN() - SELECT ATAN(.25);	COS() - SELECT COS(1);
ATAN2()- SELECT ATAN2(0.4, 2);	COT() - SELECT COT(6);
DEGREES() - SELECT DEGREES(2.5);	DIV() - SELECT 200 DIV 2;
EXP() - SELECT EXP(3);	FLOOR() - SELECT FLOOR(45.95);
GREATEST()- SELECT GREATEST(30, 120, 20, 81, 205);	LEAST()-SELECT LEAST(8, 102, 74, 81, 275);
LN() (logarithm) -SELECT LN(10);	LOG()-SELECT LOG(4, 10);
LOG10() - SELECT LOG10(2);	MOD()-SELECT MOD(28, 3);
PI() - SELECT PI();	POW()-SELECT POW(5, 3);
POWER()-SELECT POWER(7, 3);	RADIANS() - SELECT RADIANS(180);
RAND()-SELECT RAND();	ROUND()-SELECT ROUND(135.375, 2);

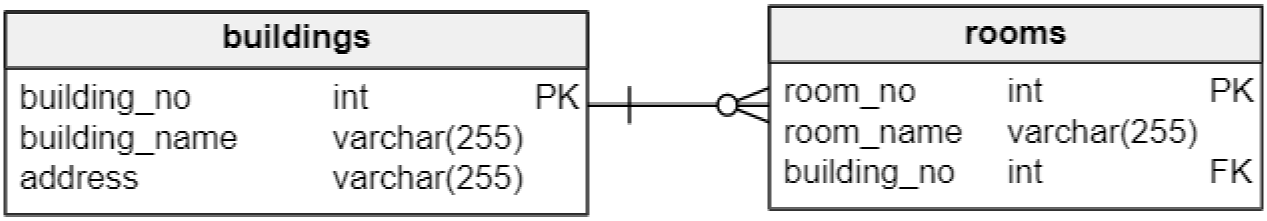
SIGN() - SELECT SIGN(-78);	SQRT()-SELECT SQRT(25);
TAN() - SELECT TAN(90);	TRUNCATE()- SELECT TRUNCATE(7235.37589, 2);

10. **Table: Student**

ID	STD_ID	NAME	MARKS
1	3	ABHI	99
2	5	GEETHASRI	89
3	6	RAHIM	49
4	9	RAM	69
5	1	RAHUL	87
6	1	RAHUL	96
7	1	RAHUL	96
8	9	RAM	96
9	9	RAM	96

1. Query To Find Second Highest Marks Of A Student?
2. Query To Find Duplicate Rows In Table?
3. What Is The Query To Fetch First Record From Student Table?
4. What Is The Query To Fetch Last Record From The Student Table?
5. What Is Query to Display First 4 Records from Student Table?
6. What Is Query to Display Last 3 Records from Student Table?
7. What Is Query To Display Nth Record From Student Table?
8. How to Get 3 Highest Marks from Student Table?
9. How to Display Odd Rows in Student Table?
10. How to Display Even Rows in Student Table?
11. How Can I Create Table With Same Structure Of Student Table?
12. Select All Records from Student Table Whose Name Is ‘abhi’ and ‘geethasri’.

MySQL ON DELETE CASCADE: Deleting Data from Multiple Related Tables:



buildings		
building_no	int	PK
building_name	varchar(255)	
address	varchar(255)	

rooms		
room_no	int	PK
room_name	varchar(255)	
building_no	int	FK

1· Create the buildings table:

CREATE TABLE buildings (

```
    building_no INT PRIMARY KEY AUTO_INCREMENT,
    building_name VARCHAR(255) NOT NULL,
    address VARCHAR(255) NOT NULL
);
```

2. Create the rooms table:

```
CREATE TABLE rooms (
    room_no INT PRIMARY KEY AUTO_INCREMENT,
    room_name VARCHAR(255) NOT NULL,
    building_no INT NOT NULL,
    FOREIGN KEY (building_no)
    REFERENCES buildings (building_no)
    ON DELETE CASCADE
);
```

3. Insert data into the buildings table:

```
INSERT INTO buildings(building_name,address) VALUES('ACME Headquaters','3950 North 1st Street CA 95134'), ('ACME Sales','5000 North 1st Street CA 95134');
```

4. SELECT * FROM buildings;

5. Insert data into the rooms table:

```
INSERT INTO rooms(room_name,building_no) VALUES('Amazon',1),('War Room',1), ('Office of CEO',1),('Marketing',2),('Showroom',2);
```

6. SELECT * FROM rooms;

Delete the building with building no. 2:

```
DELETE FROM buildings WHERE building_no = 2;

SELECT * FROM rooms;
```

Note: ON DELETE CASCADE means that if the parent record is deleted, any child records are also deleted.

Primary Key: A primary key is a field in a table which uniquely identifies each row/record in a database table. Primary keys must contain unique values. A primary key column cannot have NULL values.

FOREIGN KEY: A FOREIGN KEY is a key used to link two tables together.

SQL

A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.

The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

11. **TABLE: EMPLOYEE**

ECODE	NAME	DESIGN	SGRADE	DOJ	DOB
101	ABDUL AHMED	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

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

a) i) To display the details of all the employee in descending order of DOJ.

ii) To display NAME and DESIGN of those Employees, whose SGRADE is either S02 or S03 .

iii) To display the content of all the EMPLOYEEs, whose DOJ is between ’09-FEB-2006’ and ’08-AUG-2009’.

iv) To add a new row in the EMPLOYEE table with the following:

109, ‘HARISH ROY’, ‘HEAD-IT’, ‘S02’, ‘09-SEP-2007’, ‘21-APR-1983’.