

CONDITIONS IN SQL

(1) Demonstrate the SELECT statement with the following clauses giving an example of your choice:

- FROM :- The FROM clause is required in a SELECT statement to identify the tables and/or views that are being queried.

INPUT:-

```
CREATE TABLE REPORT_CARD(  
    Names VARCHAR(25),  
    Maths INT,  
    Hindi INT,  
    English INT,  
    Science INT,  
    EVS INT  
);  
INSERT INTO REPORT_CARD VALUES ('Jaanvi', '99', '98', '97', '99', '100');  
INSERT INTO REPORT_CARD VALUES ('Ria', '70', '45', '67', '00', '60');  
INSERT INTO REPORT_CARD VALUES ('Neha', '67', '54', '90', '90', '60');  
INSERT INTO REPORT_CARD VALUES ('Isha', '95', '76', '43', '23', '33');  
INSERT INTO REPORT_CARD VALUES ('Sanchi', '45', '90', '98', '78', '43');  
INSERT INTO REPORT_CARD VALUES ('Vaani', '89', '76', '87', '67', '45');  
INSERT INTO REPORT_CARD VALUES ('Ekta', '54', '98', '98', '76', '89');  
INSERT INTO REPORT_CARD VALUES ('Shivani', '56', '43', '47', '43', '67');  
INSERT INTO REPORT_CARD VALUES ('Khushboo', '55', '33', '96', '65', '65');  
INSERT INTO REPORT_CARD VALUES ('Mrinalini', '33', '45', '53', '98', '34');  
SELECT * FROM REPORT_CARD;
```

OUTPUT:-

	Names	Maths	Hindi	English	Science	EVS
1	Jaanvi	99	98	97	99	100
2	Ria	70	45	67	0	60
3	Neha	67	54	90	90	60
4	Isha	95	76	43	23	33
5	Sanchi	45	90	98	78	43
6	Vaani	89	76	87	67	45
7	Ekta	54	98	98	76	89
8	Shivani	56	43	47	43	67
9	Khushboo	55	33	96	65	65
10	Mrinalini	33	45	53	98	34

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

INPUT:-

```

CREATE TABLE REPORT_CARD(
    Names VARCHAR(25),
    Maths INT,
    Hindi INT,
    English INT,
    Science INT,
    EVS INT
);
INSERT INTO REPORT_CARD VALUES ('Jaanvi', '99', '98', '97', '99', '100');
INSERT INTO REPORT_CARD VALUES ('Ria', '70', '45', '67', '00', '60');
INSERT INTO REPORT_CARD VALUES ('Neha', '67', '54', '90', '90', '60');
INSERT INTO REPORT_CARD VALUES ('Isha', '95', '76', '43', '23', '33');
INSERT INTO REPORT_CARD VALUES ('Sanchi', '45', '90', '98', '78', '43');
INSERT INTO REPORT_CARD VALUES ('Vaani', '89', '76', '87', '67', '45');
INSERT INTO REPORT_CARD VALUES ('Ekta', '54', '98', '98', '76', '89');
INSERT INTO REPORT_CARD VALUES ('Shivani', '56', '43', '47', '43', '67');
INSERT INTO REPORT_CARD VALUES ('Khushboo', '55', '33', '96', '65', '65');
INSERT INTO REPORT_CARD VALUES ('Mrinalini', '33', '45', '53', '98', '34');
SELECT * FROM REPORT_CARD
WHERE Maths >= 70;

```

OUTPUT:-

	Names	Maths	Hindi	English	Science	EVS
1	Jaanvi	99	98	97	99	100
2	Ria	70	45	67	0	60
3	Isha	95	76	43	23	33
4	Vaani	89	76	87	67	45

- **GROUP BY:-** A **GROUP BY** statement in SQL specifies that a SQL **SELECT** statement partitions result rows into groups, based on their values in one or several columns. Typically, grouping is used to apply some sort of aggregate function for each group. The result of a query using a **GROUP BY** statement contains one row for each group.

INPUT:-

```
CREATE TABLE REPORT_CARD(
    Names VARCHAR(25),
    Address VARCHAR(25),
    Maths INT,
    Hindi INT,
    English INT,
    Science INT,
    EVS INT
);
INSERT INTO REPORT_CARD VALUES ('Jaanvi','Delhi','99','98','97','99','100');
INSERT INTO REPORT_CARD VALUES ('Ria','Noida','70','45','67','00','60');
INSERT INTO REPORT_CARD VALUES ('Neha','Delhi','67','54','90','90','60');
INSERT INTO REPORT_CARD VALUES ('Isha','Noida','95','76','43','23','33');
INSERT INTO REPORT_CARD VALUES ('Sanchi','Delhi','45','90','98','78','43');
INSERT INTO REPORT_CARD VALUES ('Vaani','Noida','89','76','87','67','45');
INSERT INTO REPORT_CARD VALUES ('Ekta','Delhi','54','98','98','76','89');
INSERT INTO REPORT_CARD VALUES ('Shivani','Noida','56','43','47','43','67');
INSERT INTO REPORT_CARD VALUES ('Khushboo','Delhi','55','33','96','65','65');
INSERT INTO REPORT_CARD VALUES ('Mrinalini','Noida','33','45','53','98','34');
SELECT COUNT(Names), Address
FROM REPORT_CARD
GROUP BY Address;
```

OUTPUT:-

	(No column name)	Address
1	5	Delhi
2	5	Noida

- **HAVING:-** A **HAVING** clause in SQL specifies that an SQL **SELECT** statement must only return rows where aggregate values meet the specified conditions.

INPUT:-

```

CREATE TABLE REPORT_CARD(
    Names VARCHAR(25),
    Address VARCHAR(25),
    Maths INT,
    Hindi INT,
    English INT,
    Science INT,
    EVS INT
);
INSERT INTO REPORT_CARD VALUES ('Jaanvi','Delhi','99','98','97','99','100');
INSERT INTO REPORT_CARD VALUES ('Ria','Noida','70','45','67','00','60');
INSERT INTO REPORT_CARD VALUES ('Neha','Delhi','67','54','90','90','60');
INSERT INTO REPORT_CARD VALUES ('Isha','Noida','95','76','43','23','33');
INSERT INTO REPORT_CARD VALUES ('Sanchi','Delhi','45','90','98','78','43');
INSERT INTO REPORT_CARD VALUES ('Vaani','Noida','89','76','87','67','45');
INSERT INTO REPORT_CARD VALUES ('Ekta','Delhi','54','98','98','76','89');
INSERT INTO REPORT_CARD VALUES ('Shivani','Noida','56','43','47','43','67');
INSERT INTO REPORT_CARD VALUES ('Khushboo','Delhi','55','33','96','65','65');
INSERT INTO REPORT_CARD VALUES ('Mrinalini','Noida','33','45','53','98','34')
SELECT COUNT(Names), Maths
FROM REPORT_CARD
GROUP BY Maths
HAVING Maths>70;

```

OUTPUT:-

	(No column name)	Maths
1	1	89
2	1	95
3	1	99

- **ORDER BY:-** The ORDER BY statement in SQL is used to sort the fetched data in either ascending or descending according to one or more columns. By default ORDER BY sorts the data in ascending order. We can use the keyword DESC to sort the data in descending order and the keyword ASC to sort in ascending order.

INPUT:-

```

CREATE TABLE REPORT_CARD(
    Names VARCHAR(25),
    Address VARCHAR(25),
    Maths INT,
    Hindi INT,
    English INT,
    Science INT,
    EVS INT
);
INSERT INTO REPORT_CARD VALUES ('Jaanvi','Delhi','99','98','97','99','100');
INSERT INTO REPORT_CARD VALUES ('Ria','Noida','70','45','67','00','60');
INSERT INTO REPORT_CARD VALUES ('Neha','Delhi','67','54','90','90','60');
INSERT INTO REPORT_CARD VALUES ('Isha','Noida','95','76','43','23','33');
INSERT INTO REPORT_CARD VALUES ('Sanchi','Delhi','45','90','98','78','43');
INSERT INTO REPORT_CARD VALUES ('Vaani','Noida','89','76','87','67','45');
INSERT INTO REPORT_CARD VALUES ('Ekta','Delhi','54','98','98','76','89');
INSERT INTO REPORT_CARD VALUES ('Shivani','Noida','56','43','47','43','67');
INSERT INTO REPORT_CARD VALUES ('Khushboo','Delhi','55','33','96','65','65');
INSERT INTO REPORT_CARD VALUES ('Mrinalini','Noida','33','45','53','98','34')
SELECT * FROM REPORT_CARD
ORDER BY Names, Address;

```

OUTPUT:-

	Names	Address	Maths	Hindi	English	Science	EVS
1	Ekta	Delhi	54	98	98	76	89
2	Isha	Noida	95	76	43	23	33
3	Jaanvi	Delhi	99	98	97	99	100
4	Khushboo	Delhi	55	33	96	65	65
5	Mrinalini	Noida	33	45	53	98	34
6	Neha	Delhi	67	54	90	90	60
7	Ria	Noida	70	45	67	0	60
8	Sanchi	Delhi	45	90	98	78	43
9	Shivani	Noida	56	43	47	43	67
10	Vaani	Noida	89	76	87	67	45

(2) With an example scenario of your choice, demonstrate the implementation of SQL Subqueries.

A Subquery or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause.

A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

Subqueries can be used with the SELECT, INSERT, UPDATE, and DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN, etc.

INPUT:-

```
CREATE TABLE customervisits (
    visit_id INT,
    first_name VARCHAR (50) NOT NULL,
    last_name VARCHAR (50) NOT NULL,
    Age INT,
    phone VARCHAR(20),
    store_id INT NOT NULL
);
INSERT INTO customervisits VALUES('1','Jaanvi','Gour','21','98#####00','102');
INSERT INTO customervisits VALUES('2','Ria','Babbar','22','98#####11','102');
INSERT INTO customervisits VALUES('3','Neha','Puri','25','98#####22','052');
INSERT INTO customervisits VALUES('4','Mansi','Tanwar','24','98#####33','202');
INSERT INTO customervisits VALUES('4','Ashley','vats','13','98#####44','902');
INSERT INTO customervisits VALUES('5','samarth','Gour','16','98#####55','892');
INSERT INTO customervisits VALUES('6','Naman','Babbar','16','98#####66','402');
INSERT INTO customervisits VALUES('7','Ajitesh','Singh','24','98#####77','102');
INSERT INTO customervisits VALUES('8','Arjun','Puri','31','98#####88','302');
SELECT * FROM customervisits;
SELECT first_name,last_name,Age
FROM customervisits
WHERE Age>
(SELECT Age
FROM customervisits
WHERE store_id=052
);
```

OUTPUT:-

	visit_id	first_name	last_name	Age	phone	store_id
1	1	Jaanvi	Gour	21	98#####00	102
2	2	Ria	Babbar	22	98#####11	102
3	3	Neha	Puri	25	98#####22	52
4	4	Mansi	Tanwar	24	98#####33	202
5	4	Ashley	vats	13	98#####44	902
6	5	samarth	Gour	16	98#####55	892
7	6	Naman	Babbar	16	98#####66	402
8	7	Ajitesh	Singh	24	98#####77	102

	first_name	last_name	Age
1	Arjun	Puri	31