Table of Contents

No.	Aggregate function
	LIMIT
	ORDER BY
	SELECT DISTINCT
	With
	WHERE
	UPDATE

No.	Questions
	Wildcard Characters/Like Characters?

Demo data for execute the query

```
CREATE TABLE employee (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(50),
    job_name VARCHAR(50),
    manager_id INT,
    hire_date DATE,
    salary DECIMAL(10, 2),
    commission DECIMAL(10, 2),
    dep_id INT
);

INSERT INTO employee (emp_id, emp_name, job_name, manager_id, hire_date, salary,
commission, dep_id)
VALUES
    (68319, 'KAYLING', 'PRESIDENT', NULL, '1991-11-18', 6000.00, NULL, 1001),
    (66928, 'BLAZE', 'MANAGER', 68319, '1991-05-01', 2750.00, NULL, 3001),
    (67832, 'CLARE', 'MANAGER', 68319, '1991-06-09', 2550.00, NULL, 1001),
    (65646, 'JONAS', 'MANAGER', 68319, '1991-04-02', 2957.00, NULL, 2001),
    (67858, 'SCARLET', 'ANALYST', 65646, '1997-04-19', 3100.00, NULL, 2001),
    (69062, 'FRANK', 'ANALYST', 65646, '1991-12-03', 3100.00, NULL, 2001),
    (63679, 'SANDRINE', 'CLERK', 69062, '1990-12-18', 900.00, NULL, 2001),
    (64989, 'ADELYN', 'SALESMAN', 66928, '1991-02-20', 1700.00, 400.00, 3001),
    (65271, 'WADE', 'SALESMAN', 66928, '1991-02-22', 1350.00, 600.00, 3001),
    (66564, 'MADDEN', 'SALESMAN', 66928, '1991-09-28', 1350.00, 1500.00, 3001),
    (68454, 'TUCKER', 'SALESMAN', 66928, '1991-09-08', 1600.00, 0.00, 3001),
    (68736, 'ADNRES', 'CLERK', 67858, '1997-05-23', 1200.00, NULL, 2001),
    (69000, 'JULIUS', 'CLERK', 66928, '1991-12-03', 1050.00, NULL, 3001),
    (69324, 'MARKER', 'CLERK', 67832, '1992-01-23', 1400.00, NULL, 1001);


+--------+----------+-----------+------------+------------+---------+------------
+--------+
| emp_id | emp_name | job_name  | manager_id | hire_date  | salary  | commission |
dep_id |
+--------+----------+-----------+------------+------------+---------+------------
+--------+
|  63679 | SANDRINE | CLERK     |      69062 | 1990-12-18 |  900.00 |       NULL |
2001 |
|  64989 | ADELYN   | SALESMAN  |      66928 | 1991-02-20 | 1700.00 |     400.00 |
3001 |
|  65271 | WADE     | SALESMAN  |      66928 | 1991-02-22 | 1350.00 |     600.00 |
3001 |
|  65646 | JONAS    | MANAGER   |      68319 | 1991-04-02 | 2957.00 |       NULL |
2001 |
|  66564 | MADDEN   | SALESMAN  |      66928 | 1991-09-28 | 1350.00 |    1500.00 |
3001 |
|  66928 | BLAZE    | MANAGER   |      68319 | 1991-05-01 | 2750.00 |       NULL |
3001 |
|  67832 | CLARE    | MANAGER   |      68319 | 1991-06-09 | 2550.00 |       NULL |
1001 |
```

```
|  67858 | SCARLET  | ANALYST   |      65646 | 1997-04-19 | 3100.00 |       NULL |
2001 |
|  68319 | KAYLING  | PRESIDENT |       NULL | 1991-11-18 | 6000.00 |       NULL |
1001 |
|  68454 | TUCKER   | SALESMAN  |      66928 | 1991-09-08 | 1600.00 |       0.00 |
3001 |
|  68736 | ADNRES   | CLERK     |      67858 | 1997-05-23 | 1200.00 |       NULL |
2001 |
|  69000 | JULIUS   | CLERK     |      66928 | 1991-12-03 | 1050.00 |       NULL |
3001 |
|  69062 | FRANK    | ANALYST   |      65646 | 1991-12-03 | 3100.00 |       NULL |
2001 |
|  69324 | MARKER   | CLERK     |      67832 | 1992-01-23 | 1400.00 |       NULL |
1001 |
+--------+----------+-----------+------------+------------+---------+------------
+--------+
```

Ques. How to create a new MySQL user account in MySQL?

```
CREATE USER 'testuser' IDENTIFIED BY 'sample password';
```

- grant all **privileges** of the database for a newly created user

```
GRANT ALL PRIVILEGES ON * . * TO 'testuser'@'localhost';
```

- For changes to take effect immediately **flush** these privileges by typing in the command:

```
FLUSH PRIVILEGES;
```

- to **withdraw** all privileges for our non-root user we should use:

```
REVOKE ALL PRIVILEGES ON * . * FROM 'testuser'@'localhost';
```

- Also, replace the **PERMISSION_TYPE** value with the kind of access you want to grant to your new user account.
    - CREATE — enable users to create a database or table
    - SELECT — permit users to retrieve data
    - INSERT — let users add new entries in tables
    - UPDATE — allow users to modify existing entries in tables
    - DELETE — enable users to erase table entries
    - DROP — let users delete entire database tables

```
GRANT CREATE, SELECT ON * . * TO 'testuser'@'localhost';
```

- **Display** MySQL user **account privileges**

```
SHOW GRANTS FOR 'local_user'@'localhost';
```

- you can entirely **delete** an existing user account by using the following command:

```
DROP USER 'testuser'@'localhost';
```

- **Change** a MySQL user account **password**

```
ALTER USER 'testuser'@'localhost' IDENTIFIED BY 'new_password';
```

Ques. Check version of the sql?

```
select version()
```

Create Query

Create table

```
CREATE TABLE table_name (
  id int(11) NOT NULL,

  column_name data_type(2),
  .......
);
```

Create a table which is already exists?

```
CREATE TABLE IF NOT EXISTS table_name (
  column_name data_type(2),
  column_name data_type(2),
  .......
);
```

Creaet a table through another table/Duplicate table through another table.

```
CREATE TABLE IF NOT EXISTS new_table_name LIKE exsting_table_name;
```

Creaet a table through another table/Duplicate table through another table, with structure and data?

```
CREATE TABLE IF NOT EXISTS new_table_name AS SELECT * FROM exsting_table_name;
```

Create a table and check max_salary is not exceed the upper limit of 25000

```
CREATE TABLE IF NOT EXISTS jobs(
    JOB_ID varchar(10) NOT NULL,
    JOB_TITLE varchar(35) NOT NULL,
    MIN_SALARY decimal(6,0),
    MAX_SALARY decimal(6,0),
    CHECK(MAX_SALARY<=25000)
);
```

Aggregate function

```sql
-- Sum() :- The SUM() function returns the total sum of a numeric column.
SELECT SUM(column_name) FROM table_name;

-- AVG():- The AVG() function returns the average value of a numeric column.
SELECT AVG(column_name) FROM table_name;

-- MAX() :- The MAX() function returns the largest value of the selected column.
SELECT MAX(column_name) FROM table_name;

-- Min():- The MIN() function returns the smallest value of the selected column.
SELECT MIN(column_name) FROM table_name;

-- count():- The COUNT() function returns the number of rows that matches a
specified criterion.
SELECT COUNT(column_name) FROM table_name;
```

ROUND()

- The ROUND() function is used to round a numeric value to a specified number of decimal places.*
- syntex:- syntex:- ROUND(number, decimal_places)

```sql
SELECT ROUND(123.4567, 2);  -- Returns 123.46
SELECT ROUND(123.4567, 0);  -- Returns 123
SELECT ROUND(123.4567, -1); -- Returns 120 (rounds to the nearest 10)

-- example:-
SELECT ROUND(salary, 2) AS rounded_salary FROM employees;
```

BETWEEN()

- The BETWEEN operator is used to filter the result set within a certain range. The values can be numbers, text or dates.

```sql
SELECT column_name(s) FROM table_name
WHERE column_name BETWEEN value_1 AND value_2;
```

AND

- AND is an operator that combines two conditions. Both conditions must be true for the row to be included in the result set.
- The MySQL AND Condition (also called the AND Operator) is used to test two or more conditions in a SELECT, INSERT, UPDATE, or DELETE statement.
- AND condition allows you to test 2 or more conditions.

```
SELECT column_name(s)
FROM table_name
WHERE column_1 = value_1
  AND column_2 = value_2;

SELECT * FROM contacts
WHERE state = 'California'
AND contact_id > 3000;
```

OR

- OR is an operator that filters the result set to only include rows where either condition is true.

```
SELECT column_name
FROM table_name
WHERE column_name = value_1
   OR column_name = value_2;
```

Case

- CASE statements are used to create different outputs (usually in the SELECT statement). It is SQL's way of handling if-then logic.

```
SELECT column_name,
  CASE
    WHEN condition THEN 'Result_1'
    WHEN condition THEN 'Result_2'
    ELSE 'Result_3'
  END
FROM table_name;
```

Aliases

- AS is a keyword in SQL that allows you to rename a column or table using an alias.
- Aliases are used to give a table, or a column in a table, a temporary name.
- An alias is created with the **AS** keyword.

```
SELECT column_name AS alias_name FROM table_name;
```

IS NULL / IS NOT NULL

- IS NULL and IS NOT NULL are operators used with the WHERE clause to test for empty values.

```
SELECT column_name(s)
FROM table_name
WHERE column_name IS NULL;
```

GROUP BY

- GROUP BY is a clause in SQL that is only used with aggregate functions. It is used in collaboration with the SELECT statement to arrange identical data into groups.

```
SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name;
```

HAVING

- HAVING was added to SQL because the WHERE keyword could not be used with aggregate functions.

```
SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name
HAVING COUNT(*) > value;
```

LIMIT

- LIMIT is a clause that lets you specify the maximum number of rows the result set will have.

```
SELECT column_name(s)
FROM table_name
LIMIT number;
```

ORDER BY

- ORDER BY is a clause that indicates you want to sort the result set by a particular column either alphabetically or numerically.

```
SELECT column_name
FROM table_name
ORDER BY column_name ASC | DESC;
```

SELECT DISTINCT

- SELECT DISTINCT specifies that the statement is going to be a query that returns unique values in the specified column(s).

```
SELECT DISTINCT column_name
FROM table_name;
```

With

- WITH clause lets you store the result of a query in a temporary table using an alias. You can also define multiple temporary tables using a comma and with one instance of the WITH keyword.
- The WITH clause is also known as common table expression (CTE) and subquery factoring.

```
WITH temporary_name AS (
   SELECT *
   FROM table_name)
SELECT *
FROM temporary_name
WHERE column_name operator value;
```

WHERE

- WHERE is a clause that indicates you want to filter the result set to include only rows where the following condition is true.

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator value;
```

UPDATE

- UPDATE statements allow you to edit rows in a table.

```
UPDATE table_name SET some_column = some_value
WHERE some_column = some_value;

Update customer set name="mohit" where id =1;
```

Ques. How to copy a table in another table?

```
CREATE TABLE EMP1 AS (SELECT * FROM EMP); //constraint will not copied.
```

Ques. How to copy structure of a table but not data?

```
CREATE TABLE STD AS (SELECT * FROM EMP WHERE EMPNO=-1);
```

DELETE TABLE?

- The DELETE statement is used to delete rows from a table. If you want to remove a **specific row** from a table you should use WHERE condition.

```
DELETE FROM table_name [WHERE condition];
```

- But if you do not specify the WHERE condition it will remove **all the rows** from the table.

```
DELETE FROM table_name;
```

Delete Duplicate Records?

```
CREATE TABLE employee (
    id INT,
    customer_name VARCHAR(255),
    email VARCHAR(255)
);
INSERT INTO employee (id, customer_name, email)
VALUES
    (1, 'John Doe', 'john.doe@example.com'),
    (2, 'Jane Doe', 'jane.doe@example.com'),
    (3, 'Muzamil Amin', 'Muzamilaminitoo@gmail.com'),
    (1, 'John Doe', 'john.doe@example.com'),
    (4, 'Alice Johnson', 'alice.johnson@example.com'),
    (2, 'Jane Doe', 'jane.doe@example.com');

mysql> DELETE FROM employee WHERE id IN ( SELECT id FROM employee GROUP BY id
HAVING COUNT(*) > 1 )
```

Add foreign key?

```
ALTER TABLE `bookings` ADD CONSTRAINT `advance_bookings_user_id_foreign` FOREIGN
KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
```

Highest Salary Department wise

```
Create table If Not Exists Employee (id int, name varchar(255), salary int,
departmentId int);
Create table If Not Exists Department (id int, name varchar(255));
Truncate table Employee;
insert into Employee (id, name, salary, departmentId) values ('1', 'Joe', '85000',
'1');
insert into Employee (id, name, salary, departmentId) values ('2', 'Henry',
'80000', '2');
insert into Employee (id, name, salary, departmentId) values ('3', 'Sam', '60000',
'2');
insert into Employee (id, name, salary, departmentId) values ('4', 'Max', '90000',
'1');
insert into Employee (id, name, salary, departmentId) values ('5', 'Janet',
'69000', '1');
insert into Employee (id, name, salary, departmentId) values ('6', 'Randy',
'85000', '1');
insert into Employee (id, name, salary, departmentId) values ('7', 'Will',
'70000', '1');
insert into Employee (id, name, salary, departmentId) values ('8', 'Mohit',
'90000', '1');
Truncate table Department;
insert into Department (id, name) values ('1', 'IT');
insert into Department (id, name) values ('2', 'Sales');
```

```
Employee table:
+----+-------+--------+--------------+
| id | name  | salary | departmentId |
+----+-------+--------+--------------+
| 1  | Joe   | 85000  | 1            |
| 2  | Henry | 80000  | 2            |
| 3  | Sam   | 60000  | 2            |
| 4  | Max   | 90000  | 1            |
| 5  | Janet | 69000  | 1            |
| 6  | Randy | 85000  | 1            |
| 7  | Will  | 70000  | 1            |
| 8  | Mohit | 90000  | 1            |
+----+-------+--------+--------------+
Department table:
+----+-------+
| id | name  |
+----+-------+
| 1  | IT    |
| 2  | Sales |
+----+-------+
```

Ques. Find the Highest Salary of Each Department?

```
# for single table
SELECT dep_id, max(salary) FROM `employee` GROUP BY dep_id;
+--------+-------------+
| dep_id | max(salary) |
+--------+-------------+
|   1001 |     6000.00 |
|   2001 |     3100.00 |
|   2244 |     6000.00 |
|   3001 |     2750.00 |
+--------+-------------+

# for two table
select max(employee.salary) AS salery, department.name from employee JOIN
department WHERE employee.departmentId = department.id GROUP BY department.name;
+--------+-------+
| salery | name  |
+--------+-------+
| 90000  | IT    |
| 80000  | Sales |
+--------+-------+
```

Ques. Find the Highest Salary of Each Department with name?

```
# for one table
 SELECT dep_id, emp_name, salary FROM employee WHERE (dep_id,salary) IN (SELECT
dep_id, MAX(salary) FROM employee GROUP BY dep_id);
+--------+----------+---------+
| dep_id | emp_name | salary  |
+--------+----------+---------+
|   2244 | Mohit    | 6000.00 |
|   3001 | BLAZE    | 2750.00 |
|   2001 | SCARLET  | 3100.00 |
|   1001 | KAYLING  | 6000.00 |
|   2001 | FRANK    | 3100.00 |
+--------+----------+---------+

# for two table
SELECT department.name AS dep_name, employee.name AS emp_name, employee.salary AS
salary
FROM employee JOIN department
ON employee.departmentId = department.id
JOIN(
SELECT departmentId, max(salary) AS max_salary
from employee GROUP BY departmentId
) AS dept_max ON employee.departmentId = dept_max.departmentId AND employee.salary
= dept_max.max_salary

+----------+----------+--------+
| dep_name | emp_name | salary |
+----------+----------+--------+
```

```
| Sales     | Henry    | 80000  |
| IT        | Max      | 90000  |
| IT        | Mohit    | 90000  |
+----------+----------+--------+

* Using SubQuery
SELECT department.name AS dep_name, employee.name AS emp_name, employee.salary AS
salary
FROM employee
JOIN department ON employee.departmentId = department.id
WHERE (employee.departmentId, employee.salary) IN (
    SELECT departmentId, MAX(salary)
    FROM employee
    GROUP BY departmentId
);

+----------+----------+--------+
| dep_name | emp_name | salary |
+----------+----------+--------+
| Sales    | Henry    | 80000  |
| IT       | Max      | 90000  |
| IT       | Mohit    | 90000  |
+----------+----------+--------+
```

Ques. How to find Nth highest salary from a table?

Using the LIMIT Clause

- Syntex:-

```
Select DISTINCT Salary from table_name order by Salary DESC limit n-1,1;
SELECT DISTINCT salary FROM employees ORDER BY salary DESC LIMIT 1 OFFSET N-1;
```

- The limit clause has two components, the **First component** is to skip a number of rows from the top and the **second component** is to display the number of rows we want.
- To find the **4th** Highest salary query will be

```
Select DISTINCT emp_name, salary from Employee order by salary DESC limit 3,1;
(OR)
Select DISTINCT Salary from employees order by Salary DESC limit 1 OFFSET 3;

+----------+---------+
| emp_name | salary  |
+----------+---------+
| JONAS    | 2957.00 |
+----------+---------+
```

using sub Query

```
# 3rd higest salery
SELECT MAX(salary) AS ThirdHighestSalary FROM Employee WHERE salary < (SELECT
MAX(salary) FROM Employee WHERE salary < (SELECT MAX(salary) FROM Employee));
+-------------+
| MAX(salary) |
+-------------+
|     2957.00 |
+-------------+
```

Ques. Top 5 Salery?

- Using limit

```
SELECT salary FROM employee ORDER BY salary DESC LIMIT 4
+----------+---------+
| emp_name | salary  |
+----------+---------+
| KAYLING  | 6000.00 |
| FRANK    | 3100.00 |
| SCARLET  | 3100.00 |
| JONAS    | 2957.00 |
| BLAZE    | 2750.00 |
+----------+---------+
```

```
SELECT SAL FROM(SELECT DISTINCT SAL FROM EMP WHERE SAL IS NOT NULL  ORDER BY SAL
DESC)WHERE ROWNUM <6; (in oracle)
```

- Using sub Query

```

```

Ques. Top Salery?

```
select emp_name, salary from employee order by Salary DESC;
+----------+---------+
| emp_name | salary  |
+----------+---------+
| Mohit    | 6000.00 |
| KAYLING  | 6000.00 |
| FRANK    | 3100.00 |
| SCARLET  | 3100.00 |
```

```
| JONAS    | 2957.00 |
| BLAZE    | 2750.00 |
| CLARE    | 2550.00 |
| ADELYN   | 1700.00 |
| TUCKER   | 1600.00 |
| MARKER   | 1400.00 |
| MADDEN   | 1350.00 |
| WADE     | 1350.00 |
| ADNRES   | 1200.00 |
| JULIUS   | 1050.00 |
| SANDRINE |  900.00 |
+----------+---------+
```

- List of duplicate data

```
SELECT a.id, a.name, a.email
FROM sample_table a
INNER JOIN sample_table b ON a.name = b.name AND a.email = b.email
WHERE a.id != b.id;
```

```
select emp_name, salary from employee where salary = (select max(salary) from
employee);
+----------+---------+
| emp_name | salary  |
+----------+---------+
| Mohit    | 6000.00 |
| KAYLING  | 6000.00 |
+----------+---------+
```

How to Find Duplicate values in a Table?

```
# where, groupby, having
extra:- SELECT phone, count(phone) as total_phone FROM `users` WHERE role_id = 4
group by phone having count(phone) > 1;
```

```
+----+---------+
| Id | Email   |
+----+---------+
| 1  | a@b.com |
| 2  | c@d.com |
| 3  | a@b.com |
+----+---------+

select Email, count(Email) as num from Person group by Email;
+---------+-----+
```

```
| Email   | num |
+ ------- + --- +
| a@b.com | 2   |
| c@d.com | 1   |
+---------+-----+

select Email, count(Email) as num from Person group by Email HAVING COUNT(Email) >
1;
+---------+-----+
| Email   | num |
+ ------- + --- +
| a@b.com | 2   |
+---
```

How many employees under the manager

```
 emp_id | emp_name | job_name  | manager_id | hire_date  | salary  | commission |
dep_id
--------+----------+-----------+------------+------------+---------+------------+-
-------
  68319 | KAYLING  | PRESIDENT |            | 1991-11-18 | 6000.00 |            |
1001
  66928 | BLAZE    | MANAGER   |      68319 | 1991-05-01 | 2750.00 |            |
3001
  67832 | CLARE    | MANAGER   |      68319 | 1991-06-09 | 2550.00 |            |
1001
  65646 | JONAS    | MANAGER   |      68319 | 1991-04-02 | 2957.00 |            |
2001
  67858 | SCARLET  | ANALYST   |      65646 | 1997-04-19 | 3100.00 |            |
2001
  69062 | FRANK    | ANALYST   |      65646 | 1991-12-03 | 3100.00 |            |
2001
  63679 | SANDRINE | CLERK     |      69062 | 1990-12-18 |  900.00 |            |
2001
  64989 | ADELYN   | SALESMAN  |      66928 | 1991-02-20 | 1700.00 |     400.00 |
3001
  65271 | WADE     | SALESMAN  |      66928 | 1991-02-22 | 1350.00 |     600.00 |
3001
  66564 | MADDEN   | SALESMAN  |      66928 | 1991-09-28 | 1350.00 |    1500.00 |
3001
  68454 | TUCKER   | SALESMAN  |      66928 | 1991-09-08 | 1600.00 |       0.00 |
3001
  68736 | ADNRES   | CLERK     |      67858 | 1997-05-23 | 1200.00 |            |
2001
  69000 | JULIUS   | CLERK     |      66928 | 1991-12-03 | 1050.00 |            |
3001
  69324 | MARKER   | CLERK     |      67832 | 1992-01-23 | 1400.00 |            |
1001
```

How many employee under the manager

```
SELECT w.manager_id,
       count(*)
FROM employees w,
     employees m
WHERE w.manager_id = m.emp_id
GROUP BY w.manager_id
ORDER BY w.manager_id ASC;

Output:-
  manager_id | count
-------------+-------
       65646 |     2
       66928 |     5
       67832 |     1
       67858 |     1
       68319 |     3
       69062 |     1
```

and count higest emp under the manager.

```
SELECT m.emp_name,
       count(*)
FROM employees w,
     employees m
WHERE w.manager_id = m.emp_id
GROUP BY m.emp_name
HAVING count(*) =
  (SELECT MAX (mycount)
   FROM
     (SELECT COUNT(*) mycount
      FROM employees
      GROUP BY manager_id) a);

Output:-
 emp_name | count
----------+-------
 BLAZE    |     5
```

Replace a Column Values from 'male' to 'female' and 'female' to 'male'

```
CREATE TABLE EMPDATA
(
EMPNAME VARCHAR(25),
GENDER VARCHAR(6),
DEPT VARCHAR(20),
CONTACTNO BIGINT NOT NULL,
CITY VARCHAR(15)
```

```
);

INSERT INTO EMPDATA
VALUES ('VISHAL','MALE','SALES',9193458625,'GHAZIABAD'),
('DIVYA','FEMALE','MANAGER',7352158944,'BAREILLY'),
('REKHA','FEMALE','IT',7830246946,'KOLKATA'),
('RAHUL','MALE','MARKETING',9635688441,'MEERUT'),
('SANJAY','MALE','SALES',9149335694,'MORADABAD'),
('ROHAN','MALE','MANAGER',7352158944,'BENGALURU'),
('RAJSHREE','FEMALE','SALES',9193458625,'VODODARA'),
('AMAN','MALE','IT',78359941265,'RAMPUR'),
('RAKESH','MALE','MARKETING',9645956441,'BOKARO'),
('MOHINI','FEMALE','SALES',9147844694,'Delhi')

select * from empdata;
+----------+--------+-----------+-------------+-----------+
| EMPNAME  | GENDER | DEPT      | CONTACTNO   | CITY      |
+----------+--------+-----------+-------------+-----------+
| VISHAL   | FEMALE | SALES     |  9193458625 | GHAZIABAD |
| DIVYA    | MALE   | MANAGER   |  7352158944 | BAREILLY  |
| REKHA    | MALE   | IT        |  7830246946 | KOLKATA   |
| RAHUL    | FEMALE | MARKETING |  9635688441 | MEERUT    |
| SANJAY   | FEMALE | SALES     |  9149335694 | MORADABAD |
| ROHAN    | FEMALE | MANAGER   |  7352158944 | BENGALURU |
| RAJSHREE | MALE   | SALES     |  9193458625 | VODODARA  |
| AMAN     | FEMALE | IT        | 78359941265 | RAMPUR    |
| RAKESH   | FEMALE | MARKETING |  9645956441 | BOKARO    |
| MOHINI   | MALE   | SALES     |  9147844694 | Delhi     |
+----------+--------+-----------+-------------+-----------+
```

```
UPDATE empdata
SET GENDER = CASE
    WHEN GENDER='male' THEN 'female'
    WHEN GENDER='female' THEN 'male'
    END;
(OR)
UPDATE EMPDATA
SET gender = CASE
    gender WHEN 'male' THEN 'female'
            WHEN 'female' THEN 'male'
    ELSE gender
END;


+----------+--------+-----------+-------------+-----------+
| EMPNAME  | GENDER | DEPT      | CONTACTNO   | CITY      |
+----------+--------+-----------+-------------+-----------+
| VISHAL   | male   | SALES     |  9193458625 | GHAZIABAD |
| DIVYA    | female | MANAGER   |  7352158944 | BAREILLY  |
| REKHA    | female | IT        |  7830246946 | KOLKATA   |
| RAHUL    | male   | MARKETING |  9635688441 | MEERUT    |
```

```
| SANJAY   | male   | SALES     |  9149335694 | MORADABAD |
| ROHAN    | male   | MANAGER   |  7352158944 | BENGALURU |
| RAJSHREE | female | SALES     |  9193458625 | VODODARA  |
| AMAN     | male   | IT        | 78359941265 | RAMPUR    |
| RAKESH   | male   | MARKETING |  9645956441 | BOKARO    |
| MOHINI   | female | SALES     |  9147844694 | Delhi     |
+----------+--------+-----------+-------------+-----------+
```

Find Names of students whose age is greater than 21?

```sql
Select field_name1, field_name2 from table_name where student_age < 21;
```