Create Table

```sql
---- create table----
CREATE TABLE simplicode (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    city VARCHAR(50),
    salary NUMERIC(10,2)
);
```

Insert Values

```sql
---insert values---

INSERT INTO simplicode (name, age, city, salary)
VALUES
('Amit Kumar', 28, 'Delhi', 45000),
('Priya Sharma', 32, 'Mumbai', 52000),
('Ravi Verma', 25, 'Bangalore', 38000),
('Sneha Patel', 29, 'Pune', 60000),
('Rahul Mehta', 35, 'Chennai', 75000),
('Amit Kumar', 28, 'Delhi', 45000),
('Priya Sharma', 32, 'Mumbai', 52000),
('Ravi Verma', 25, 'Bangalore', 38000),
('Sneha Patel', 29, 'Pune', 60000),
('Rahul Mehta', 35, 'Chennai', 75000),
('Anjali Singh', 27, 'Kolkata', 42000),
('Vikram Rao', 40, 'Hyderabad', 88000),
('Neha Joshi', 30, 'Jaipur', 54000),
('Karan Gupta', 26, 'Delhi', 39000),
('Meena Iyer', 33, 'Chennai', 61000),
('Arjun Kapoor', 31, 'Bangalore', 70000),
('Pooja Desai', 24, 'Ahmedabad', 36000),
('Manish Tiwari', 38, 'Lucknow', 82000),
('Ritika Jain', 29, 'Pune', 58000),
('Sanjay Reddy', 45, 'Hyderabad', 95000),
('Komal Bhatia', 28, 'Mumbai', 47000),
('Rohit Malhotra', 34, 'Delhi', 64000),
('Divya Nair', 26, 'Kochi', 40000),
('Aditya Das', 30, 'Kolkata', 55000),
('Sunita Mishra', 37, 'Varanasi', 73000);
```

Identify Duplicates Values

```
--identify duplicates value----

SELECT name, age, city, salary, COUNT(*)
FROM simplicode
GROUP BY name, age, city, salary
HAVING COUNT(*) > 1;
```

Delete The Duplicates

```
----delete the duplicates values -----

WITH cte AS (
    SELECT id,
           ROW_NUMBER() OVER (
               PARTITION BY name, age, city, salary
               ORDER BY id
           ) AS row_num
    FROM simplicode
)
DELETE FROM simplicode
WHERE id IN (
    SELECT id FROM cte WHERE row_num > 1
);
```

SQL Operator

```
----addition----
SELECT id , name, age, city , salary , salary + 10000  AS new_salary  FROM simplicode
SELECT * FROM simplicode

--subtraction----
SELECT salary - 10000 AS old_salary FROM simplicode
SELECT * FROM simplicode

--multiplication----
SELECT salary * 20000 as new_salary from simplicode
SELECT * FROM simplicode

---division---
SELECT salary / 2 as division_salary from simplicode
```

Comparision Operator

```
----equal to -----
SELECT * FROM simplicode WHERE salary = 52000

-----not equal to ----
SELECT * FROM simplicode WHERE SALARY != 34000

---greater than -------
SELECT * FROM simplicode WHERE id >25

---lesser than----
SELECT * FROM simplicode WHERE age < 40

--greater than equal to-----
SELECT * FROM simplicode WHERE salary >=60000

----lesser than eqaul to ---
SELECT * FROM simplicode WHERE salary <=45000
```

Logical Operator

```
--AND---
SELECT * FROM simplicode where salary >60000 AND city='Delhi'

---OR-----
SELECT * FROM simplicode WHERE SALARY > 40000 or city ='delhi'

---between----
SELECT * FROM simplicode WHERE salary BETWEEN 56000 and 66000

----not----
SELECT * FROM simplicode WHERE NOT salary=78000
```

SQL Expressions

```
----Boolean-----
SELECT * FROM simplicode where city='Pune'
SELECT * FROM simplicode where age = 25

---numeric expressions-----

SELECT * FROM simplicode WHERE age / 2 > 20

SELECT AVG(age) from simplicode
SELECT SUM(age) from simplicode

------DATE-------(IF GIVEN)
SELECT * FROM simplicode WHERE DOB > '1999-09-12'
SELECT current_timestamp
```

Drop Table

```
-----drop table----
DROP table employees
```

Delete Table

```
------delete table-----
delete from employees where id='3'
```

Truncate Table

```
-----truncate table------
truncate table employees
```

SQL Alter Table

```
---sql alter table----
ALTER TABLE employees ADD COLUMN department VARCHAR(50)
```

SQL Reaname Table

```
-------- sql rename table------
alter table employees rename to members
select * from members
```

Select

```
----------------SELECT ----------
SELECT * FROM simplicode WHERE age=33
SELECT * FROM simplicode WHERE salary >= 60000
```

Distinct

```
-----distinct------
SELECT DISTINCT city from simplicode
```

Count

```
-----count-------
SELECT COUNT(*) FROM simplicode
SELECT COUNT(*) FROM simplicode WHERE  salary >78000
SELECT COUNT (DISTINCT city) FROM simplicode
```

Limit

```
------LIMIT----------
SELECT * FROM simplicode
limit 5
```

Order By

```
-----order by-------
SELECT * FROM simplicode order by salary desc
limit 5
```

Select Random

```
------------select random-----------------
SELECT * FROM simplicode ORDER BY random()
limit 5
SELECT * FROM simplicode WHERE id IN (1,2,17)
```

Sum

```
-------------sum----------------
SELECT sum(age) from simplicode
```

Select Null

```
----------select null-------------
SELECT * FROM simplicode where age is null
```

Where Clause

```
------------where clause----------------
SELECT * FROM simplicode where salary >=70000
SELECT * FROM simplicode where city='Chennai'
SELECT * FROM simplicode where salary between 45000 and 50000
```

Update

```
----------update----------
update simplicode set salary=90000 where id=1
select * from simplicode
select * from simplicode order by salary desc
```

AdityaTyagi

SQL

And

```
-------------and-------------
SELECT * FROM simplicode where salary > 45000 and city='Delhi'
```

Or

```
------------------or------------------
SELECT * FROM simplicode where id>5 or salary>89000
```

As

```
----------------as------------------
SELECT salary as total_salary  from simplicode
```

like

```
----------------like---------------
SELECT * FROM simplicode where name LIKE 'M%'
```

Order by asc

```
--------------order by asc --------------
SELECT * FROM  simplicode WHERE age > 23 order by  salary asc
```

Order by desc

```
--------------order by desc --------------
SELECT * FROM simplicode WHERE age <90 order by  salary desc
```

Order by/random

```
-------------------------order by/random-------------
SELECT * FROM simplicode order by random() limit 5
```

Order By Multiple

```
-------order by multiple-------------------
SELECT id, city, age, name
FROM simplicode
ORDER BY city DESC , age DESC, name ASC;
```

Delete

```
--------------------delete------------------------
DELETE FROM simplicode where id=3

SELECT COUNT(*) FROM simplicode
DELETE FROM simplicode where city='Delhi'
DELETE FROM simplicode where city='Chennai' and salary >70000
DELETE FROM simplicode where salary between 60000 and 90000
```

Add Column

```
------------------------add column-----------------
ALTER TABLE simplicode add column stipned varchar(20) not null

ALTER TABLE simplicode
ADD COLUMN phone_number VARCHAR(20),
ADD COLUMN is_active BOOLEAN DEFAULT TRUE;
```

Modify

```
-------------------------modify----------
ALTER TABLE simplicode
RENAME COLUMN salary TO total_salary;
```

Drop

```
---------------drop---------------------------
alter table simplicode drop column is_active
alter table simplicode drop column stipned
```

Default

```
-----------------------------DEFAULT----------------------

ALTER TABLE simplicode ALTER column stipned set default 15000
SELECT  * FROM simplicode
```

SQL JOIN

```sql
-- Table 1: Employees

CREATE TABLE Employees (
    EmpID INT PRIMARY KEY,
    EmpName VARCHAR(50),
    DeptID INT
);

-- Table 2: Departments

CREATE TABLE Departments (
    DeptID INT PRIMARY KEY,
    DeptName VARCHAR(50)
);

-- Insert sample data

INSERT INTO Employees (EmpID, EmpName, DeptID) VALUES
(1, 'Alice', 10),
(2, 'Bob', 20),
(3, 'Charlie', 30),
(4, 'David', NULL);

INSERT INTO Departments (DeptID, DeptName) VALUES
(10, 'HR'),
(20, 'Finance'),
(30, 'IT'),
(40, 'Marketing');

SELECT * FROM departments
SELECT * FROM employees
```

Inner Join

```
----------------INNER JOIN------------
SELECT e.EmpID,e.EmpName,d.DeptName
FROM Employees e
INNER JOIN Departments d
on e.DeptID = d.DeptID;
```

Left Join

```
----------left join-----------------

SELECT e.EmpID, e.EmpName, d.DeptName
FROM Employees e
LEFT JOIN Departments d
    ON e.DeptID = d.DeptID;
```

Right Join

```
-------------right join---------------
SELECT e.EmpID,e.EmpName,d.DeptName
from Employees e
right join Departments d
on e.DeptID=d.DeptID;
```

Full Outer Join

```
------------full outer join---------
SELECT e.EmpID,e.EmpName,d.DeptName
from Employees e
full outer join Departments d
on e.DeptID=d.DeptID;
```

Cross Join

```
-------------CROSS JOIN----------
SELECT e.EmpName, d.DeptName
FROM Employees e
CROSS JOIN Departments d;
```

AdityaTyagi

SQL

Create Table

```
-------------table create----------

CREATE TABLE employees2 (
    EmpID VARCHAR(10) PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Department VARCHAR(50),
    Phone VARCHAR(15),
    Email VARCHAR(100),
    Salary INT
);
```

Insert Values

```
INSERT INTO employees2 (EmpID, FirstName, LastName, Department, Phone, Email, Salary) VALUES
('E001','Diya','Bose','Finance','+918793896299','diya.bose64@company.com',127018),
('E002','Ira','Singh','Sales','+919383528137','ira.singh40@yahoo.com',136959),
('E003','Atharv','Reddy','Finance','+918245168936','atharv.reddy89@company.com',42087),
('E004','Aadhya','Mehta','IT','+919291314017','aadhya.mehta38@outlook.com',89888),
('E005','Vivaan','Pillai','Finance','+919374424356','vivaan.pillai43@yahoo.com',47201),
('E006','Myra','Patel','Marketing','+919278625729','myra.patel14@company.com',95676),
('E007','Atharv','Gill','HR','+919441228388','atharv.gill55@yahoo.com',43205),
('E008','Ananya','Nair','Finance','+917051325915','ananya.nair83@yahoo.com',115702),
('E009','Krishna','Verma','Sales','+917706534054','krishna.verma24@company.com',45902),
('E010','Navya','Naidu','Marketing','+919707417275','navya.naidu47@company.com',96714),
('E011','Anika','Naidu','Finance','+917708226882','anika.naidu45@gmail.com',129572),
('E012','Sai','Mishra','Finance','+919068882991','sai.mishra2@yahoo.com',138847),
('E013','Atharv','Patel','Marketing','+917713240322','atharv.patel37@outlook.com',80945),
('E014','Vivaan','Pillai','Finance','+917352406879','vivaan.pillai18@company.com',122117),
('E015','Ananya','Yadav','Sales','+917222429155','ananya.yadav9@yahoo.com',131627),
('E016','Myra','Gill','Sales','+919099330050','myra.gill16@outlook.com',119057),
('E017','Aditya','Naidu','Finance','+919222335099','aditya.naidu32@outlook.com',96167),
('E018','Atharv','Naidu','Marketing','+917933657913','atharv.naidu36@yahoo.com',50976),
('E019','Arjun','Mehta','Finance','+919474214079','arjun.mehta46@company.com',130221),
('E020','Aarohi','Yadav','Finance','+919807294790','aarohi.yadav6@company.com',59108),
('E021','Sai','Tiwari','Marketing','+919361254705','sai.tiwari58@outlook.com',139672),
('E037','Krishna','Rastogi','Finance','+919651509203','krishna.rastogi61@company.com',86972),
('E038','Ananya','Sharma','HR','+917739406915','ananya.sharma31@company.com',116024),
('E039','Navya','Kapoor','Finance','+918207191721','navya.kapoor77@company.com',146202),
('E040','Aditya','Kohli','Sales','+917200108265','aditya.kohli53@yahoo.com',92589),
('E041','Krishna','Bose','Sales','+918201283783','krishna.bose33@outlook.com',44355),
('E042','Arjun','Mehta','Finance','+918280591288','arjun.mehta14@company.com',134059),
('E043','Vivaan','Iyer','HR','+917006597105','vivaan.iyer41@outlook.com',62237),
('E044','Aadhya','Bose','Marketing','+917674277627','aadhya.bose95@outlook.com',101832),
('E045','Krishna','Kohli','Finance','+919892517087','krishna.kohli52@yahoo.com',45549),
('E046','Sai','Nair','Finance','+919446418345','sai.nair97@company.com',139987),
('E047','Reyansh','Kohli','Finance','+919002510105','reyansh.kohli26@outlook.com',125997),
('E048','Vivaan','Bose','Finance','+917802440764','vivaan.bose9@company.com',65516),
('E049','Aditya','Naidu','Marketing','+919395295326','aditya.naidu10@outlook.com',63348),
('E050','Sai','Kohli','Finance','+918891970535','sai.kohli52@outlook.com',90235);
```

Group By /Order By

```
---GROUP BY------ORDER BY -------
SELECT count(empid)as total , firstname
from employees2
GROUP by firstname
order by firstname
```

Between

```
---------BETWEEN----------------
SELECT salary ,empid
from employees2
WHERE salary BETWEEN  45000 and 100000
order by empid
```

Where

```
--------------where ------------
select * from employees2
where salary between 25000 and 100000
```

Having/Group by

```
-----having-----  group by-------
select empid,avg(salary),count(employees2)
from employees2
group by empid
having avg(salary)>25000
```

Delete

```
------------delete-----------
delete  from employees2
where empid='E004'
select  * from employees2
```

Subqueries

Select

```
-------select---------
SELECT * FROM employees2
SELECT *FROM employees2
where salary < (select avg(salary) from employees2)
```

Insert

```
----------insert------------
CREATE TABLE employees1 (
    EmpID VARCHAR(10) PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Department VARCHAR(50),
    Phone VARCHAR(15),
    Email VARCHAR(100),
    Salary INT
);
```

Correct PostgreSQL Example (User-Defined Function)

```
-- Create a sample table for testing
CREATE TABLE employees3 (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50),
    salary NUMERIC
);
```

Insert

```
-- Insert some sample data
INSERT INTO employees3 (name, salary) VALUES
('Aditya', 40000),
('Rohit', 30000),
('Neha', 50000),
('kiran',20000);
```

Function to check bonus

```
-- Function to check bonus eligibility
CREATE OR REPLACE FUNCTION bonusstatus(salary NUMERIC)
RETURNS VARCHAR AS $$
BEGIN
    IF salary > 35000 THEN
        RETURN 'Eligible for Bonus';
    ELSE
        RETURN 'Not Eligible';
    END IF;
END;
$$ LANGUAGE plpgsql;
```

Function In Query

```
-- Use the function in a query
SELECT name, salary, bonusstatus(salary) AS bonus_status
FROM employees3;

select *from employees1
```

SQL Cast & Convert Function

```
-----------sql cast and convert functions---------------
SELECT CAST('12,09,2004' as date ) as converted_date ----date---
SELECT CAST('100' AS INTEGER) AS converted_int;   -----string to integer-----
SELECT CAST(123 AS VARCHAR) AS converted_text;  ----integer to text---------
SELECT CAST(45.67 AS INTEGER) AS converted_int; ----------decimal to integer-
```

Autocommit

```
----------autocommit----------
CREATE TABLE test_commit(id INT);
INSERT INTO test_commit VALUES (1);
commit;

select * from test_commit
```

Update

```
------update---------------

update employees2
set salary =90000
where empid='E001';

select * from employees2
```

Rollback

```
-------------rollback -----------------
rollback;

delete from employees2 where department='sales';
rollback;
select * from employees2
```

SQL LIKE AND WILDCARDS CHARACTERS

Percent

```
----percent

select * from employees2 where firstname LIKE 'A%'
 select * from employees2 where firstname LIKE 'A%v'
   select * from employees2 where firstname LIKE 'K%' and salary between 25000 and 80000
```

Underscore

```
-----------------------underscore---------------------
select * from employees2 where firstname LIKE '___'
select * from employees2 where firstname LIKE 'A____'
```

how to find the nth highest salary in sql

```
---------how to find the nth highest salary in sql--
SELECT max(salary) FROM employees2 where salary  <
(SELECT MAX(salary) from employees2)

select * from employees2 order by salary desc
```

Dense Rank

```
------------desnse rank------------------
SELECT empid,firstname,lastname,department,phone ,salary from
(SELECT empid,firstname,lastname,department,phone ,salary,dense_rank() over (order by salary desc) as rank_salary from employees2) as E ------main syntax--
where rank_salary=12
```

View as

```
------------veiew as-----------
-----ex1
SELECT * FROM employees3

CREATE VIEW  employees3_view as
SELECT * FROM employees3

SELECT * FROM employees3_view

 ----ex2
 SELECT * FROM employees2

CREATE VIEW employees2_view as
SELECT * FROM employees2

SELECT * FROM employees2_view
```

how to delete duplicates rows in the sql

```
-----how to delete duplicates rows in the sql ----
SELECT * FROM employees2

SELECT firstname,lastname,count(*) from employees2
group by firstname,lastname
having count(*)>1
```

Delete

```
-----------------delete-------------

delete  FROM employees2 where empid not in(
select max(empid) from employees2 group by firstname,lastname);

SELECT * FROM employees2
```

Row Number

```
-------------row number-------------
SELECT empid,firstname,lastname,department, row_number()
over(partition by firstname order by firstname) as row_num
from employees2;
```

Delete Duplicates

```
---------delete duplicates----------

DELETE FROM employees2
WHERE empid IN (
    SELECT empid
    FROM (
        SELECT empid,
               ROW_NUMBER() OVER (PARTITION BY firstname ORDER BY empid) AS row_num
        FROM employees2
    ) t
    WHERE row_num > 1
);
```

pattern matching in sql-

```
------like %
SELECT * from employees2
where firstname like 'A%'
order by firstname
```

Like %

Like last%

```
----like last%
SELECT DISTINCT lastname from employees2 WHERE lastname like 'Ya%'
SELECT * FROM employees2 where lastname like '%a'
SELECT DISTINCT firstname from employees2 where firstname like '%i'
SELECT * from employees2 where firstname like '%vy%'
SELECT * FROM employees2 where firstname like 'N%%a'
```

Underscore

```
--------underscore
SELECT * FROM employees2 where firstname like 'A____a'
SELECT * FROM employees2 where firstname like'_____i'
SELECT * FROM employees2 where department not like 'finance'
```

SQL TRIGGERS

Trigger Function

```
-- Trigger function ----------before insert ----------
CREATE OR REPLACE FUNCTION check_empid_before_insert()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.empid < 'E004' THEN
        NEW.empid := 0;    -- modify value before insert
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Trigger

```
-- Trigger
CREATE TRIGGER before_insert_empid
BEFORE INSERT ON employees2
FOR EACH ROW
EXECUTE FUNCTION check_empid_before_insert();
```

After Insert Create log table

```
-- Create log table
CREATE TABLE employees_log (
    log_id SERIAL PRIMARY KEY,
    emp_id VARCHAR(10),
    action VARCHAR(50),
    log_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Trigger Function

```
-- Trigger function
CREATE OR REPLACE FUNCTION log_employee_insert()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO employees_log(emp_id, action)
    VALUES (NEW.empid, 'Inserted');
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Before Update

```sql
-- Staff table
CREATE TABLE staff_before_update (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50),
    salary NUMERIC(10,2)
);
```

Trigger Function

```sql
-- Trigger function
CREATE OR REPLACE FUNCTION prevent_salary_decrease()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.salary < OLD.salary THEN
        RAISE EXCEPTION '❌ Salary cannot be decreased!';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```sql
-- Trigger
CREATE TRIGGER trg_before_update_staff
BEFORE UPDATE ON staff_before_update
FOR EACH ROW
EXECUTE FUNCTION prevent_salary_decrease();


INSERT INTO staff_before_update (name, salary) VALUES ('Aditya', 50000);

UPDATE staff_before_update SET salary = 60000 WHERE name = 'Aditya';  -- ✅ Works
UPDATE staff_before_update SET salary = 40000 WHERE name = 'Aditya';  -- ❌ Error


SELECT * FROM staff_before_update;
SELECT * FROM employees2;
```

AdityaTyagi

SQL

Serial

```
-----SERIAL------------auto_increment (only in mysql not in pgamdin)--

CREATE TABLE products(
product_id  int primary key,
custid int not null,
product_name varchar(20) not null,
qunatity int,
price varchar(20)
);
```

how to create index in sql

```
------ ◆  1. Create a Simple Index---------

CREATE INDEX idx_employee_name
ON employees2 (firstname);

SELECT * FROM employees2 WHERE firstname = 'Aditya';

-------------- ◆  2. Unique Index

CREATE UNIQUE INDEX idx_unique_department
ON employees2 (department);

SELECT * FROM employees2 where department='Finance'

----------------drop -------
DROP INDEX idx_employee_name;

ALTER TABLE employees2 DROP INDEX idx_employee_name;
```

Regex

```
-------------regexp-----------------------

* `~` → regex match (case-sensitive)
* `~*` → regex match (case-insensitive)
* `!~` → does not match (case-sensitive)
* `!~*` → does not match (case-insensitive)
```

SQL CTE

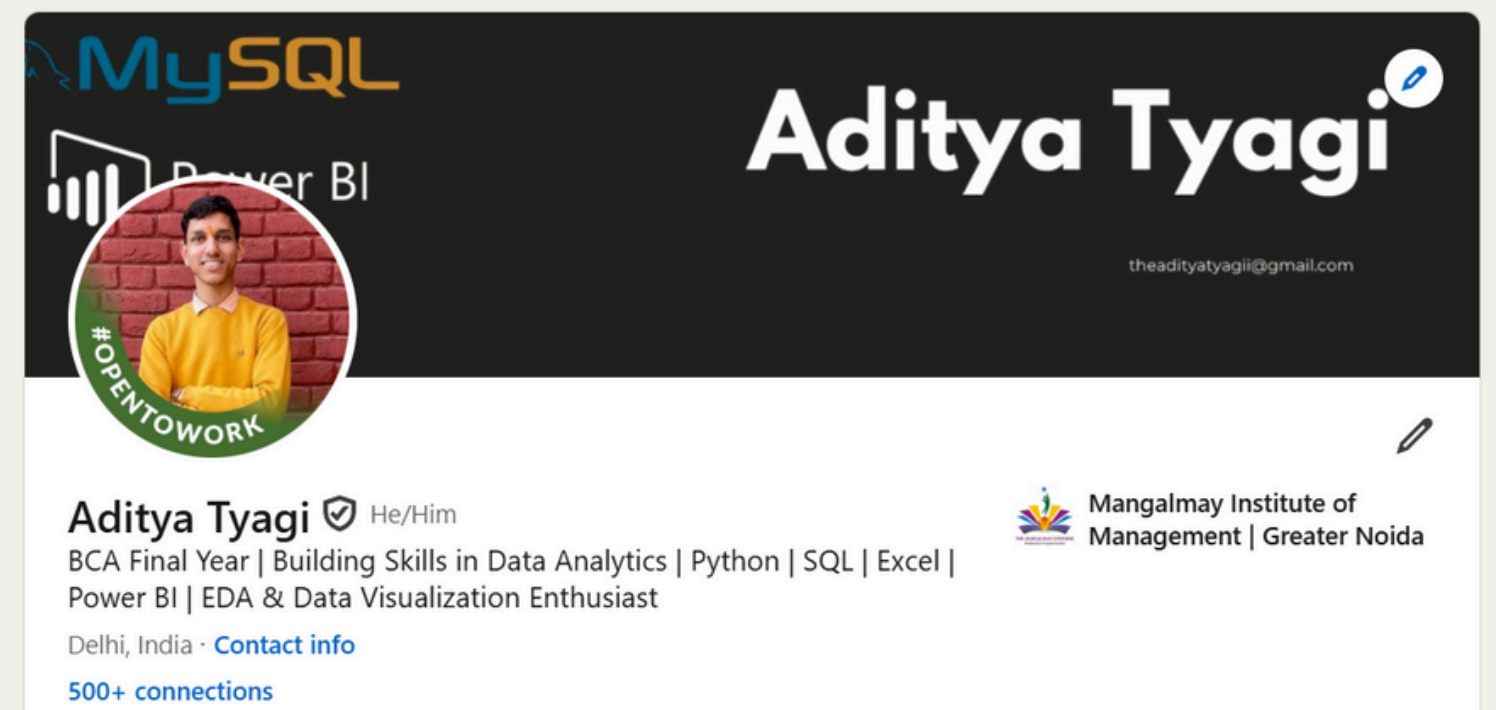
```sql
--------------------------SQL CTE---------------
---ex1
SELECT * FROM employees2


WITH avg_salary_cte AS (
    SELECT department, AVG(salary) AS avg_sal
    FROM employees2
    GROUP BY department
)
SELECT *
FROM avg_salary_cte
WHERE avg_sal > 50000;

---ex2


-- Suppose we have employees2 table
-- emp_id, name, department, salary

-- Step 1: Create a CTE to filter by department
WITH dept_cte AS (
    SELECT empid, firstname, department, salary
    FROM employees2
    WHERE department = 'Finance'
)
-- Step 2: Use it in a main query
SELECT *
FROM dept_cte
WHERE salary > 50000;
```


MySQL
Power BI
Aditya Tyagi
theadityatyagii@gmail.com
#OPENTOWORK
Aditya Tyagi He/Him
BCA Final Year | Building Skills in Data Analytics | Python | SQL | Excel | Power BI | EDA & Data Visualization Enthusiast
Delhi, India · Contact info
500+ connections
Mangalmay Institute of Management | Greater Noida


Thank you