```
create database Colleges;



CREATE TABLE students (
   id SERIAL PRIMARY KEY,
   name VARCHAR(50),
   age INT,
   grade VARCHAR(5)
);




INSERT INTO students (name, age, grade)
VALUES
('Ankit', 20, 'A'),
('Amit', 22, 'B'),
('Vanshika', 21, 'A');



SELECT * FROM students;        -- all data

SELECT name, age FROM students; -- specific columns

SELECT * FROM students WHERE age > 20;
```

```
SELECT * FROM students WHERE grade = 'A';



SELECT COUNT(*) FROM students;          -- number of rows

SELECT COUNT(*) FROM students WHERE grade = 'A';

SELECT SUM(age) FROM students;



UPDATE students
SET grade = 'A+'
WHERE name = 'Vanshika';

DELETE FROM students WHERE name = 'Vanshika';

ALTER TABLE students ADD COLUMN email VARCHAR(100);

ALTER TABLE students DROP COLUMN email;

SELECT * FROM students ORDER BY age ASC;
SELECT * FROM students ORDER BY age DESC;
```

```
SELECT grade, COUNT(*)
FROM students
GROUP BY grade;
```

```
SELECT * FROM students
WHERE grade = 'A' AND age > 20;
```

```
SELECT * FROM students
WHERE grade = 'A' OR age < 22;
```

```
SELECT * FROM students
WHERE age BETWEEN 20 AND 22;
```

```
SELECT * FROM students
WHERE grade IN ('A', 'B');
```

```
SELECT * FROM students
WHERE name LIKE 'A%';   -- starts with A
SELECT * FROM students
WHERE name LIKE '%t';   -- ends with t
SELECT * FROM students
```

```
WHERE name LIKE '%An%'; -- contains "An"




CREATE TABLE departments (
   dept_id SERIAL PRIMARY KEY,
   dept_name VARCHAR(50)
);

CREATE TABLE employees (
   emp_id SERIAL PRIMARY KEY,
   emp_name VARCHAR(50),
   salary INT,
   dept_id INT REFERENCES departments(dept_id)
);



INSERT INTO departments (dept_name) VALUES
('HR'),      -- dept_id = 1
('IT'),      -- dept_id = 2
('Finance'),  -- dept_id = 3
('Marketing');-- dept_id = 4
INSERT INTO employees (emp_name, salary, dept_id) VALUES
('Brijesh', 50000, 1),   -- HR
('Anjali', 60000, 2),     -- IT
```

```
('Bijeta', 55000, 2), -- IT

('Ruchika', 70000, 3),   -- Finance

('Chandan', 85000, NULL);  -- No department


--INNER JOIN (only matching records)


SELECT e.emp_name, d.dept_name

FROM employees e

INNER JOIN departments d ON e.dept_id = d.dept_id;


--LEFT JOIN (all employees, even without department)

SELECT e.emp_name, d.dept_name

FROM employees e

LEFT JOIN departments d ON e.dept_id = d.dept_id;


--RIGHT JOIN (all departments, even without employees)


SELECT e.emp_name, d.dept_name

FROM employees e

RIGHT JOIN departments d ON e.dept_id = d.dept_id;


--FULL OUTER JOIN (everything, even no match on both sides)

SELECT e.emp_name, d.dept_name

FROM employees e

FULL OUTER JOIN departments d ON e.dept_id = d.dept_id;
```

```
--Aggregations with GROUP BY

SELECT d.dept_name, AVG(e.salary) AS avg_salary
FROM employees e
JOIN departments d ON e.dept_id = d.dept_id
GROUP BY d.dept_name;

--Count employees per department

SELECT d.dept_name, COUNT(e.emp_id) AS num_employees
FROM employees e
JOIN departments d ON e.dept_id = d.dept_id
GROUP BY d.dept_name;

--Filtering with HAVING
-- Departments with more than 1 employee
SELECT d.dept_name, COUNT(e.emp_id) AS num_employees
FROM employees e
JOIN departments d ON e.dept_id = d.dept_id
GROUP BY d.dept_name
HAVING COUNT(e.emp_id) > 1;
```

```
--ORDER BY Example

SELECT emp_name, salary
FROM employees
ORDER BY salary DESC;

--WHERE, IN, LIKE

SELECT emp_name, salary
FROM employees
WHERE salary BETWEEN 50000 AND 70000;

SELECT emp_name
FROM employees
WHERE dept_id IN (1, 2);

SELECT emp_name
FROM employees
WHERE emp_name LIKE 'A%'; -- starts with A

--Creating a table with a DATE column

CREATE TABLE orders (
    order_id SERIAL PRIMARY KEY,
```

```
    customer_name VARCHAR(50),
    order_date DATE,
    amount INT
);



INSERT INTO orders (customer_name, order_date, amount) VALUES
('Amit', '2025-01-10', 200),
('Shubham',   '2025-02-15', 350),
('Somyadeep', '2025-02-20', 150),
('Zayed', '2025-03-05', 500);

-- Orders in February 2025
SELECT * FROM orders
WHERE order_date BETWEEN '2025-02-01' AND '2025-02-28';

-- Orders after March 1, 2025
SELECT * FROM orders
WHERE order_date > '2025-03-01';



-- Extract year, month, day
SELECT order_id, customer_name,
       EXTRACT(YEAR FROM order_date) AS year,
       EXTRACT(MONTH FROM order_date) AS month,
```

```
    EXTRACT(DAY FROM order_date) AS day
FROM orders;
-- Current date
SELECT CURRENT_DATE;
-- Current timestamp (date + time)
SELECT NOW();nnnn
-- Orders from the last 30 days
SELECT * FROM orders
WHERE order_date >= CURRENT_DATE - INTERVAL '30 days';
```