

PostGres Sample data

1. store_db

customers

```
CREATE TABLE customers (  
  cust_id SERIAL PRIMARY KEY,  
  cust_name VARCHAR(100) NOT NULL  
);
```

```
INSERT INTO customers (cust_name)  
VALUES  
('Raju'), ('Sham'), ('Paul'), ('Alex');
```

Orders

```
CREATE TABLE orders (  
  ord_id SERIAL PRIMARY KEY,  
  ord_date DATE NOT NULL,  
  price NUMERIC NOT NULL,  
  cust_id INTEGER NOT NULL,  
  FOREIGN KEY (cust_id) REFERENCES  
  customers (cust_id)  
);
```

```
INSERT INTO orders (ord_date, cust_id, price)  
VALUES  
('2024-01-01', 1, 250.00),  
('2024-01-15', 1, 300.00),  
('2024-02-01', 2, 150.00),  
('2024-03-01', 3, 450.00),  
('2024-04-04', 2, 550.00);
```

Institute

Table Creation

- **courses**

- **Create Table**

```
CREATE TABLE courses (  
  c_id SERIAL PRIMARY KEY,  
  name VARCHAR(100) NOT NULL,  
  fee NUMERIC NOT NULL  
);
```

- **Data**

```
INSERT INTO courses (name, fee)  
VALUES  
( 'Mathematics', 500.00),  
( 'Physics', 600.00),  
( 'Chemistry', 700.00);
```

- **students**

- **Create Table**

```
CREATE TABLE students (  
  s_id SERIAL PRIMARY KEY,  
  name VARCHAR(100) NOT NULL  
);
```

- **Data**

```
INSERT INTO Students (name) VALUES  
( 'Raju'),  
( 'Sham'),  
( 'Alex');
```

- **enrollment**

- **Create Table**

```
CREATE TABLE enrollment (  
  enrollment_id SERIAL PRIMARY KEY,
```

```
s_id INT NOT NULL,  
c_id INT NOT NULL,  
enrollment_date DATE NOT NULL,  
FOREIGN KEY (s_id) REFERENCES students(s_id),  
FOREIGN KEY (c_id) REFERENCES courses(c_id)  
);
```

- **Data**

```
INSERT INTO enrollment (s_id, c_id, enrollment_date)  
VALUES  
(1, 1, '2024-01-01'), -- Raju enrolled in Mathematics  
(1, 2, '2024-01-15'), -- Raju enrolled in Physics  
(2, 1, '2024-02-01'), -- Sham enrolled in Mathematics  
(2, 3, '2024-02-15'), -- Sham enrolled in Chemistry  
(3, 3, '2024-03-25'); -- Alex enrolled in Chemistry
```

SHOW DATA

```
SELECT  
    e.enrollment_id,  
    s.name AS student_name,  
    c.name AS course_name,  
    c.fee,  
    e.enrollment_date  
FROM  
    enrollment e  
JOIN  
    students s ON e.s_id = s.s_id  
JOIN  
    courses c ON e.c_id = c.c_id;
```

=====

TASK StoreDB

- **customers**

```
CREATE TABLE customers (  
    cust_id SERIAL PRIMARY KEY,  
    cust_name VARCHAR(100) NOT NULL  
);
```

```
INSERT INTO customers (cust_name)  
VALUES  
    ('Raju'), ('Sham'), ('Paul'), ('Alex');
```

- **orders**

```
CREATE TABLE orders (  
    ord_id SERIAL PRIMARY KEY,  
    ord_date DATE NOT NULL,  
    cust_id INTEGER NOT NULL,  
    FOREIGN KEY (cust_id) REFERENCES customers(cust_id)  
);
```

```
INSERT INTO orders (ord_date, cust_id)  
VALUES  
    ('2024-01-01', 1), -- Raju first order  
    ('2024-02-01', 2), -- Sham first order  
    ('2024-03-01', 3), -- Paul first order  
    ('2024-04-04', 2); -- Sham second order
```

- **order_items**

```
CREATE TABLE order_items (  
    item_id SERIAL PRIMARY KEY,  
    ord_id INTEGER NOT NULL,  
    p_id INTEGER NOT NULL,  
    quantity INTEGER NOT NULL,  
    FOREIGN KEY (ord_id) REFERENCES orders(ord_id),  
    FOREIGN KEY (p_id) REFERENCES products(p_id)  
);
```

```
INSERT INTO order_items (ord_id, p_id, quantity)
```

VALUES

(1, 1, 1), -- Raju ordered 1 Laptop
(1, 4, 2), -- Raju ordered 2 Cables
(2, 1, 1), -- Sham ordered 1 Laptop
(3, 2, 1), -- Paul ordered 1 Mouse
(3, 4, 5), -- Paul ordered 5 Cables
(4, 3, 1); -- Sham ordered 1 Keyboard

• products

```
CREATE TABLE products (  
  p_id SERIAL PRIMARY KEY,  
  p_name VARCHAR(100) NOT NULL,  
  price NUMERIC NOT NULL  
);
```

```
INSERT INTO products (p_name, price)
```

VALUES

```
  ('Laptop', 55000.00),  
  ('Mouse', 500),  
  ('Keyboard', 800.00),  
  ('Cable', 250.00)  
;
```

=====

To see overall report

	cust_name character varying (100) 🔒	ord_date date 🔒	p_name character varying (100) 🔒	price numeric 🔒	quantity integer 🔒	total_price numeric 🔒
1	Raju	2024-01-01	Laptop	55000.00	1	55000.00
2	Raju	2024-01-01	Cable	250.00	2	500.00
3	Sham	2024-02-01	Laptop	55000.00	1	55000.00
4	Paul	2024-03-01	Mouse	500	1	500
5	Paul	2024-03-01	Cable	250.00	5	1250.00
6	Sham	2024-04-04	Keyboard	800.00	1	800.00

SELECT

```
c.cust_name,  
o.ord_date,  
p.p_name,  
p.price,  
oi.quantity,  
(oi.quantity*p.price) AS total_price  
FROM order_items oi  
  JOIN  
    products p ON oi.p_id=p.p_id  
  JOIN  
    orders o ON o.ord_id=oi.ord_id  
  JOIN  
    customers c ON o.cust_id=c.cust_id;
```