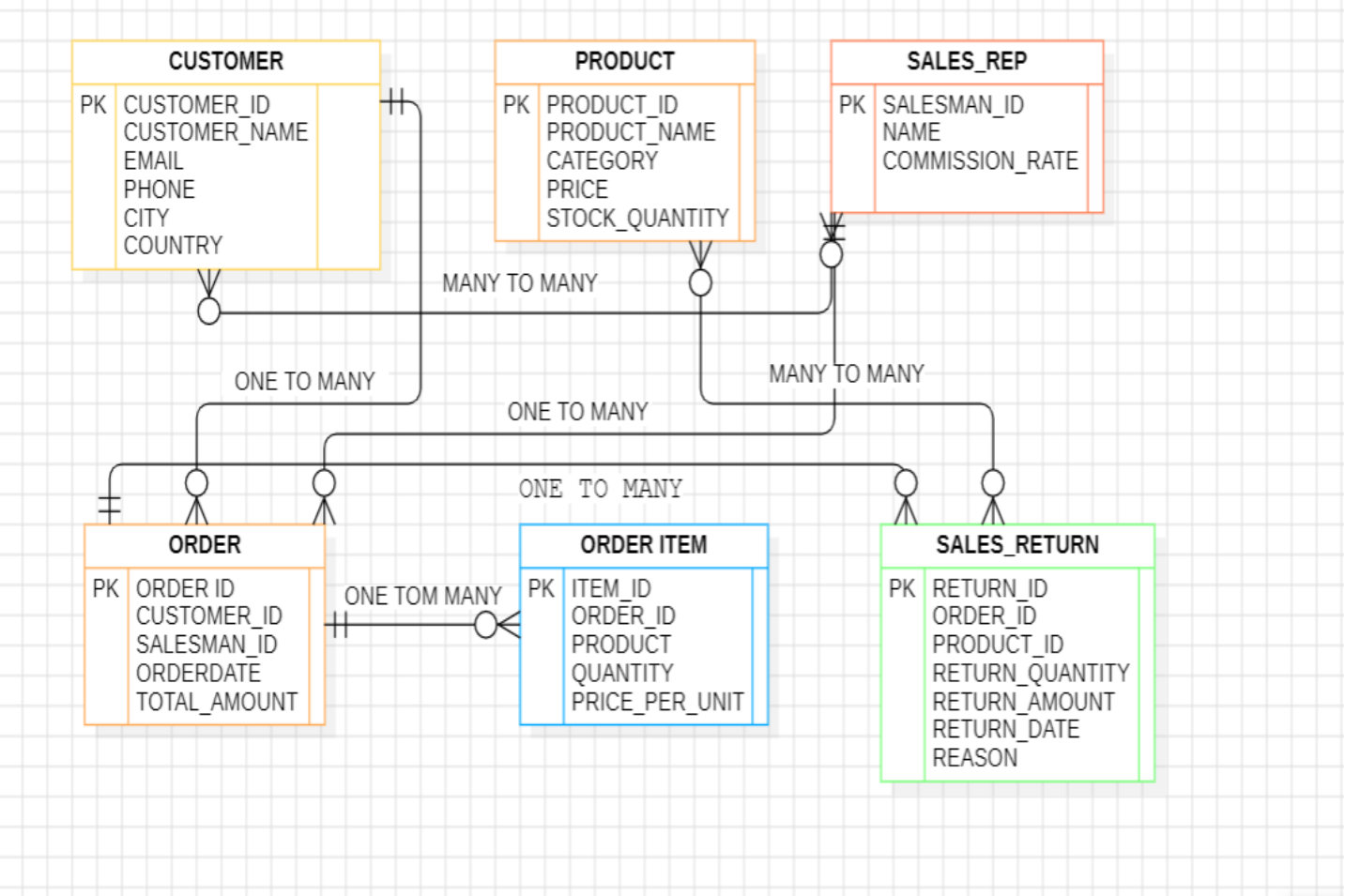
SALES DATASET

Introduction

The SQL project around a comprehensive sales management system designed to facilitate efficient handling of customers, products, orders, and sales transactions. At its core, the database comprises five key tables: Customers, which stores essential details such as customer ID, name, email, phone number, city, and country; Products, which catalogues various items available for sale, including product ID, name, category, price, and stock quantity; and Sales Representatives, detailing the sales personnel along with their commission rates. The Orders table links customers to their purchases, capturing order ID, customer ID, sales representative ID, order date, and total amount spent. Complementing this, the Order Items table allows for multiple products to be associated with each order, detailing the individual item ID, order ID, product ID, quantity, and price per unit. Finally, the Sales Returns table tracks any returned items, including return IDs, order IDs, product IDs, quantities returned, return amounts, return dates, and reasons for the returns. This structured approach not only streamlines sales operations but also enables insightful analysis of customer spending patterns, product performance, and sales representative effectiveness, ultimately aiding in informed business decisions and improved customer satisfaction.

ER DIAGRAM


CUSTOMER
PK
CUSTOMER_ID
CUSTOMER_NAME
EMAIL
PHONE
CITY
COUNTRY
PRODUCT
PK
PRODUCT_ID
PRODUCT_NAME
CATEGORY
PRICE
STOCK_QUANTITY
SALES_REP
PK
SALESMAN_ID
NAME
COMMISSION_RATE
MANY TO MANY
ONE TO MANY
MANY TO MANY
ONE TO MANY
ONE TO MANY
ORDER
PK
ORDER ID
CUSTOMER_ID
SALESMAN_ID
ORDERDATE
TOTAL_AMOUNT
ONE TOM MANY
ORDER ITEM
PK
ITEM_ID
ORDER_ID
PRODUCT
QUANTITY
PRICE_PER_UNIT
SALES_RETURN
PK
RETURN_ID
ORDER_ID
PRODUCT_ID
RETURN_QUANTITY
RETURN_AMOUNT
RETURN_DATE
REASON

DATABASE DESIGN

DATABASE: SALES

TABLES:

A) CUSTOMER

B) PRODUCT

C) SALES_REP

D) ORDER

E) ORDER ITEM

F) SALES_RETURN

CREATING TABLE

A) CUSTOMER:

```
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY,
    customer_name VARCHAR(100),
    email VARCHAR(100),
    phone VARCHAR(15),
    city VARCHAR(50),
    country VARCHAR(50)
);
```

B) PRODUCT:

```
CREATE TABLE Products (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    category VARCHAR(50),
    price DECIMAL(10, 2),
    stock_quantity INT
);
```

C) SALES_REP:

```sql
CREATE TABLE Sales_Reps (
    salesman_id INT PRIMARY KEY,
    name VARCHAR(100),
    commission_rate DECIMAL(5, 2)
);
```

D) ORDER:

```sql
CREATE TABLE Orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    salesman_id INT,
    order_date DATE,
    total_amount DECIMAL(10, 2),
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id),
    FOREIGN KEY (salesman_id) REFERENCES Sales_Reps(salesman_id)
);
```

E) ORDER ITEM:

```sql
CREATE TABLE Order_Items (
    item_id INT PRIMARY KEY,
    order_id INT,
    product_id INT,
    quantity INT,
    price_per_unit DECIMAL(10, 2),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
```

F) SALES_RETURN:

```sql
CREATE TABLE Sales_Returns (
    return_id INT PRIMARY KEY,
    order_id INT,
    product_id INT,
    return_quantity INT,
    return_amount DECIMAL(10, 2),
    return_date DATE,
    reason VARCHAR(100),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
```

TABLE IN DATABASE:

```
+---------------------------------+
| Tables_in_sandbox_db |
+---------------------------------+
| customers                       |
| order_items                     |
| orders                          |
| products                        |
| sales_reps                      |
| sales_returns                   |
+---------------------------------+
```

DATA DEFINATION LANGUAGE(DDL)

A) CREATING TABLE:

1.CUSTOMER:

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
customer_name	varchar(100)	YES		NULL	
email	varchar(100)	YES		NULL	
phone	varchar(15)	YES		NULL	
city	varchar(50)	YES		NULL	
country	varchar(50)	YES		NULL	

2.PRODUCT:

Field	Type	Null	Key	Default	Extra
product_id	int	NO	PRI	NULL	
product_name	varchar(100)	YES		NULL	
category	varchar(50)	YES		NULL	
price	decimal(10,2)	YES		NULL	
stock_quantity	int	YES		NULL	

3. SALES_REPS:

Field	Type	Null	Key	Default	Extra
salesman_id	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
commission_rate	decimal(5,2)	YES		NULL	

4. ORDER:

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	
customer_id	int	YES	MUL	NULL	
salesman_id	int	YES	MUL	NULL	
order_date	date	YES		NULL	
total_amount	decimal(10,2)	YES		NULL	

5. ORDER_ITEM:

Field	Type	Null	Key	Default	Extra
item_id	int	NO	PRI	NULL	
order_id	int	YES	MUL	NULL	
product_id	int	YES	MUL	NULL	
quantity	int	YES		NULL	
price_per_unit	decimal(10,2)	YES		NULL	

5. SALES_RETURN:

Field	Type	Null	Key	Default	Extra
return_id	int	NO	PRI	NULL	
order_id	int	YES	MUL	NULL	
product_id	int	YES	MUL	NULL	
return_quantity	int	YES		NULL	
return_amount	decimal(10,2)	YES		NULL	
return_date	date	YES		NULL	
reason	varchar(100)	YES		NULL	

B) ALTER:

1.ALTER TABLE ADD COLUMN

```
alter table Customers add last_name VARCHAR(100) after customer_name;
```

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
customer_name	varchar(100)	YES		NULL	
last_name	varchar(100)	YES		NULL	
email	varchar(100)	YES		NULL	
phone	varchar(15)	YES		NULL	
city	varchar(50)	YES		NULL	
country	varchar(50)	YES		NULL	

2.ALTER TABLE ADD COLUMN

```
alter table products modify category varchar(100);
```

```
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| product_id    | int           | NO   | PRI | NULL    |       |
| product_name  | varchar(100)  | YES  |     | NULL    |       |
| category      | varchar(100)  | YES  |     | NULL    |       |
| price         | decimal(10,2) | YES  |     | NULL    |       |
| stock_quantity | int          | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
```

3.ALTER TABLE RENAME

```
alter table products change column price cost int;
```

```
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| product_id    | int          | NO   | PRI | NULL    |       |
| product_name  | varchar(100) | YES  |     | NULL    |       |
| category      | varchar(100) | YES  |     | NULL    |       |
| cost          | int          | YES  |     | NULL    |       |
| stock_quantity | int         | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
```

4.ALTER TABLE DROP COLUMN:

```
alter table customers drop city;
```

C) RENAME TABLE:

```
alter table Sales_Reps rename Sales;
```

```
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| salesman_id     | int          | NO   | PRI | NULL    |       |
| name            | varchar(100) | YES  |     | NULL    |       |
| commission_rate | decimal(5,2) | YES  |     | NULL    |       |
+-----------------+--------------+------+-----+---------+-------+
```

D) TRUNCATE:

```
Truncate employee;
```

E) DROP TABLE:

```
DROP table EMPLOYEE;
```

DATA MANUPILATION LANGUAGE(DML)

A) INSERT INTO TABLE

```
insert into customers VALUES(5,'Samule davis','samule44@gmail.com',9874563210,'Chennai','india');
```

```
+-------------+---------------+--------------------+------------+---------+---------+
| customer_id | customer_name | email              | phone      | city    | country |
+-------------+---------------+--------------------+------------+---------+---------+
|           5 | Samule davis  | samule44@gmail.com | 9874563210 | Chennai | india   |
+-------------+---------------+--------------------+------------+---------+---------+
```

B)UPDATE INTO TABLE

```
UPDATE customers set city='thane' where customer_id=5;
```

```
+-------------+---------------+--------------------+------------+-------+---------+
| customer_id | customer_name | email              | phone      | city  | country |
+-------------+---------------+--------------------+------------+-------+---------+
|           5 | Samule davis  | samule44@gmail.com | 9874563210 | thane | india   |
+-------------+---------------+--------------------+------------+-------+---------+
```

C)DELETE INTO TABLE

```
delete FROM Customers WHERE customer_id=5;
```

DATA QUERY LANGUAGE (DQL)

A) SELECT QUERY

```
select *  FROM Customers;
```

customer_id	customer_name	email	phone	city	country
1	Alice Johnson	alice@example.com	9876543210	Mumbai	India
2	Bob Brown	bob@example.com	8765432109	Delhi	India
3	Charlie Davis	charlie@example.com	7654321098	Bangalore	India
4	Diana Evans	diana@example.com	6543210987	Chennai	India
5	Ethan Ford	ethan@example.com	5432109876	Kolkata	India

B)QUERY ORDER BY ASC

```
SELECT * FROM Customers ORDER BY customer_id ASC;
```

customer_id	customer_name	email	phone	city	country
1	Alice Johnson	alice@example.com	9876543210	Mumbai	India
2	Bob Brown	bob@example.com	8765432109	Delhi	India
3	Charlie Davis	charlie@example.com	7654321098	Bangalore	India
4	Diana Evans	diana@example.com	6543210987	Chennai	India
5	Ethan Ford	ethan@example.com	5432109876	Kolkata	India

C)QUERY ORDER BY DSC:

```
SELECT * FROM Customers ORDER BY customer_id DESC;
```

customer_id	customer_name	email	phone	city	country
5	Ethan Ford	ethan@example.com	5432109876	Kolkata	India
4	Diana Evans	diana@example.com	6543210987	Chennai	India
3	Charlie Davis	charlie@example.com	7654321098	Bangalore	India
2	Bob Brown	bob@example.com	8765432109	Delhi	India
1	Alice Johnson	alice@example.com	9876543210	Mumbai	India

D)QUERY ORDER BY COLUMN:

```
SELECT * FROM Products ORDER BY PRODUCT_NAME;
```

product_id	product_name	category	price	stock_quantity
104	Chair	Furniture	150.00	60
103	Desk	Furniture	250.00	75
105	Headphones	Electronics	100.00	200
101	Laptop	Electronics	1200.00	50
102	Smartphone	Electronics	800.00	100

E) LIMIT QUERY:

```
SELECT * FROM Products LIMIT 3;
```

product_id	product_name	category	price	stock_quantity
101	Laptop	Electronics	1200.00	50
102	Smartphone	Electronics	800.00	100
103	Desk	Furniture	250.00	75

F) SELECT QUERY FROM SPECIFIC COLUMN:

```
select customer_name, PHONE FROM Customers;
```

```
+---------------+------------+
| customer_name | PHONE      |
+---------------+------------+
| Alice Johnson | 9876543210 |
| Bob Brown     | 8765432109 |
| Charlie Davis | 7654321098 |
| Diana Evans   | 6543210987 |
| Ethan Ford    | 5432109876 |
+---------------+------------+
```

G) SELECT QUERY WITH COLUMN NAME CHANGE:

```
select customer_name AS NAME,  PHONE AS CONTACT FROM Customers;
```

```
+---------------+------------+
| NAME          | CONTACT    |
+---------------+------------+
| Alice Johnson | 9876543210 |
| Bob Brown     | 8765432109 |
| Charlie Davis | 7654321098 |
| Diana Evans   | 6543210987 |
| Ethan Ford    | 5432109876 |
+---------------+------------+
```

H) DISTINCT QUERY

```
select DISTINCT(email) FROM Customers;
```

```
+---------------------+
| email               |
+---------------------+
| alice@example.com   |
| bob@example.com     |
| charlie@example.com |
| diana@example.com   |
| ethan@example.com   |
+---------------------+
```

USING WHERE CLAUSE

COMPARISION OPERTATOR

```
select * FROM Customers WHERE customer_id=3 ;
```

```
+-------------+---------------+---------------------+------------+-----------+---------+
| customer_id | customer_name | email               | phone      | city      | country |
+-------------+---------------+---------------------+------------+-----------+---------+
|           3 | Charlie Davis | charlie@example.com | 7654321098 | Bangalore | India   |
+-------------+---------------+---------------------+------------+-----------+---------+
```

```
select * FROM Customers WHERE CITY="MUMBAI" ;
```

```
+-------------+---------------+-------------------+------------+--------+---------+
| customer_id | customer_name | email             | phone      | city   | country |
+-------------+---------------+-------------------+------------+--------+---------+
|           1 | Alice Johnson | alice@example.com | 9876543210 | Mumbai | India   |
+-------------+---------------+-------------------+------------+--------+---------+
```

```
SELECT * FROM Products WHERE PRICE>= 100;
```

```
+------------+--------------+-------------+---------+----------------+
| product_id | product_name | category    | price   | stock_quantity |
+------------+--------------+-------------+---------+----------------+
|        101 | Laptop       | Electronics | 1200.00 |             50 |
|        102 | Smartphone   | Electronics |  800.00 |            100 |
|        103 | Desk         | Furniture   |  250.00 |             75 |
+------------+--------------+-------------+---------+----------------+
```

USING LOGICAL OPERATER

A) USING AND OPERTATOR

```
SELECT * FROM Order_Items WHERE quantity= 1 AND ORDER_id=1002;
```

```
+---------+----------+------------+----------+----------------+
| item_id | order_id | product_id | quantity | price_per_unit |
+---------+----------+------------+----------+----------------+
|       3 |     1002 |        102 |        1 |         800.00 |
|       4 |     1002 |        103 |        1 |         250.00 |
+---------+----------+------------+----------+----------------+
```

B) USING AND/OR OPERTATOR

```
SELECT * FROM PRODUCTS WHERE(category='Electronics' AND stock_quantity =50 ) OR PRICE <100;
```

```
+------------+--------------+-------------+---------+----------------+
| product_id | product_name | category    | price   | stock_quantity |
+------------+--------------+-------------+---------+----------------+
|        101 | Laptop       | Electronics | 1200.00 |             50 |
+------------+--------------+-------------+---------+----------------+
```

C)USING BETWEEN OPERTATOR

```
SELECT * FROM ORDERS WHERE total_amount BETWEEN 100 AND 1000;
```

```
+----------+-------------+-------------+------------+--------------+
| order_id | customer_id | salesman_id | order_date | total_amount |
+----------+-------------+-------------+------------+--------------+
|     1003 |           3 |         203 | 2023-02-10 |       400.00 |
|     1004 |           4 |         201 | 2023-02-15 |       150.00 |
|     1005 |           5 |         202 | 2023-03-05 |       500.00 |
+----------+-------------+-------------+------------+--------------+
```

D) USING IN OPERTATOR

```
SELECT * FROM PRODUCTS WHERE product_name IN('LAPTOP','CHAIR','DESK');
```

```
+------------+--------------+-------------+---------+----------------+
| product_id | product_name | category    | price   | stock_quantity |
+------------+--------------+-------------+---------+----------------+
|        101 | Laptop       | Electronics | 1200.00 |             50 |
|        103 | Desk         | Furniture   |  250.00 |             75 |
|        104 | Chair        | Furniture   |  150.00 |             60 |
+------------+--------------+-------------+---------+----------------+
```

AGGREGATE FUNCTION

A) COUNT FUNCTION

```
SELECT count(product_id) FROM PRODUCTS;
```

```
+-------------------+
| count(product_id) |
+-------------------+
|                 5 |
+-------------------+
```

B) AVERAGE FUNCTION

```
SELECT AVG(PRICE) FROM PRODUCTS;
```

```
+------------+
| AVG(PRICE) |
+------------+
| 500.000000 |
+------------+
```

C) SUM FUNCTION

```
SELECT SUM(PRICE) FROM PRODUCTS;
```

```
+------------+
| SUM(PRICE) |
+------------+
|    2500.00 |
+------------+
```

GROUP BY

```
SELECT COUNT(category) category FROM PRODUCTS group BY category;
```

```
+----------+
| category |
+----------+
|        3 |
|        2 |
+----------+
```

```
SELECT COUNT(order_id) ID FROM ORDERS group BY salesman_id;
```

```
+----+
| ID |
+----+
|  2 |
|  2 |
|  1 |
+----+
```

LIKE OPERATOR

```
SELECT * FROM CUSTOMERS WHERE customer_name LIKE 'A%';
```

```
+-------------+---------------+-------------------+------------+--------+---------+
| customer_id | customer_name | email             | phone      | city   | country |
+-------------+---------------+-------------------+------------+--------+---------+
|           1 | Alice Johnson | alice@example.com | 9876543210 | Mumbai | India   |
+-------------+---------------+-------------------+------------+--------+---------+
```

```
SELECT * FROM CUSTOMERS WHERE customer_name LIKE '%A%';
```

```
+-------------+---------------+---------------------+------------+-----------+---------+
| customer_id | customer_name | email               | phone      | city      | country |
+-------------+---------------+---------------------+------------+-----------+---------+
|           1 | Alice Johnson | alice@example.com   | 9876543210 | Mumbai    | India   |
|           3 | Charlie Davis | charlie@example.com | 7654321098 | Bangalore | India   |
|           4 | Diana Evans   | diana@example.com   | 6543210987 | Chennai   | India   |
|           5 | Ethan Ford    | ethan@example.com   | 5432109876 | Kolkata   | India   |
+-------------+---------------+---------------------+------------+-----------+---------+
```

```
SELECT * FROM CUSTOMERS WHERE customer_name LIKE '%D';
```

```
+-------------+---------------+-------------------+------------+---------+---------+
| customer_id | customer_name | email             | phone      | city    | country |
+-------------+---------------+-------------------+------------+---------+---------+
|           5 | Ethan Ford    | ethan@example.com | 5432109876 | Kolkata | India   |
+-------------+---------------+-------------------+------------+---------+---------+
```

UNION

```sql
SELECT customer_name AS name FROM Customers
UNION
SELECT name FROM Sales_Reps;
```

```
+---------------+
| name          |
+---------------+
| Alice Johnson |
| Bob Brown     |
| Charlie Davis |
| Diana Evans   |
| Ethan Ford    |
| John Doe      |
| Jane Smith    |
| Sam Wilson    |
+---------------+
```

```sql
SELECT product_name AS name FROM Products
UNION
SELECT customer_name FROM Customers;
```

```
+---------------+
| name          |
+---------------+
| Laptop        |
| Smartphone    |
| Desk          |
| Chair         |
| Headphones    |
| Alice Johnson |
| Bob Brown     |
| Charlie Davis |
| Diana Evans   |
| Ethan Ford    |
+---------------+
```

JOINS

A) INNER JOIN

```sql
SELECT
    o.order_id,
    c.customer_name,
    sr.name AS sales_rep,
    o.order_date,
    o.total_amount
FROM Orders o
INNER JOIN Customers c ON o.customer_id = c.customer_id
INNER JOIN Sales_Reps sr ON o.salesman_id = sr.salesman_id;
```

order_id	customer_name	sales_rep	order_date	total_amount
1001	Alice Johnson	John Doe	2023-01-15	2000.00
1004	Diana Evans	John Doe	2023-02-15	150.00
1002	Bob Brown	Jane Smith	2023-01-20	1250.00
1005	Ethan Ford	Jane Smith	2023-03-05	500.00
1003	Charlie Davis	Sam Wilson	2023-02-10	400.00

B) LEFT JOIN

```sql
SELECT
    c.customer_name,
    o.order_id,
    o.order_date,
    o.total_amount
FROM Customers c
LEFT JOIN Orders o ON c.customer_id = o.customer_id;
```

customer_name	order_id	order_date	total_amount
Alice Johnson	1001	2023-01-15	2000.00
Bob Brown	1002	2023-01-20	1250.00
Charlie Davis	1003	2023-02-10	400.00
Diana Evans	1004	2023-02-15	150.00
Ethan Ford	1005	2023-03-05	500.00

C) RIGHT JOIN

```sql
SELECT
    sr.name AS sales_rep,
    o.order_id,
    o.total_amount
FROM Sales_Reps sr
right JOIN Orders o ON sr.salesman_id = o.salesman_id;
```

sales_rep	order_id	total_amount
John Doe	1001	2000.00
Jane Smith	1002	1250.00
Sam Wilson	1003	400.00
John Doe	1004	150.00
Jane Smith	1005	500.00

D)CROSS JOIN

```sql
SELECT
    p.product_name,
    sr.name AS sales_rep
FROM Products p
CROSS JOIN Sales_Reps sr;
```

product_name	sales_rep
Laptop	Sam Wilson
Laptop	Jane Smith
Laptop	John Doe
Smartphone	Sam Wilson
Smartphone	Jane Smith
Smartphone	John Doe
Desk	Sam Wilson
Desk	Jane Smith
Desk	John Doe
Chair	Sam Wilson
Chair	Jane Smith
Chair	John Doe
Headphones	Sam Wilson
Headphones	Jane Smith
Headphones	John Doe

E) MULTIPLE JOIN

```sql
SELECT
    c1.customer_name AS customer1,
    c2.customer_name AS customer2,
    c1.city
FROM Customers c1
JOIN Customers c2
    ON c1.city = c2.city;
```

customer1	customer2	city
Alice Johnson	Alice Johnson	Mumbai
Bob Brown	Bob Brown	Delhi
Charlie Davis	Charlie Davis	Bangalore
Diana Evans	Diana Evans	Chennai
Ethan Ford	Ethan Ford	Kolkata

SUBQUERYS

```sql
SELECT
    product_name,
    category,
    price
FROM Products p
WHERE price = (
    SELECT MAX(price)
    FROM Products
    WHERE category = p.category
);
```

```
+--------------+-------------+---------+
| product_name | category    | price   |
+--------------+-------------+---------+
| Laptop       | Electronics | 1200.00 |
| Desk         | Furniture   |  250.00 |
+--------------+-------------+---------+
```

```sql
SELECT
    customer_name
FROM Customers c
WHERE c.customer_id IN (
    SELECT customer_id
    FROM Orders
    GROUP BY customer_id
    HAVING SUM(total_amount) > (
        SELECT AVG(total_amount) FROM Orders
    )
);
```

```
+---------------+
| customer_name |
+---------------+
| Alice Johnson |
| Bob Brown     |
+---------------+
```

```sql
SELECT name
FROM Sales_Reps sr
WHERE sr.salesman_id = (
    SELECT salesman_id
    FROM Orders
    GROUP BY salesman_id
    ORDER BY SUM(total_amount) DESC
    LIMIT 1
);
```

```
+----------+
| name     |
+----------+
| John Doe |
+----------+
```