

# **END-TO-END E-COMMERCE DATABASE ANALYSIS USING SQL**

## **PROJECT REPORT**

**AUTHOR**

**(DHARSAN R S)**

## DDL COMMANDS

### 1. Create a table and name it categories.

```
CREATE TABLE customers (  
    customer_id INT PRIMARY KEY,  
    name VARCHAR(100),  
    email VARCHAR(100),  
    phone VARCHAR(15),  
    city VARCHAR(50),  
    created_at DATE  
);
```

### 2. Create a table and name it customer

```
create table customers  
(  
    customer_id int          not null  
        primary key,  
    name          varchar(100) null,  
    email         varchar(100) null,  
    phone         varchar(15)  null,  
    city          varchar(50)  null,  
    created_at   date          null  
);
```

### 3. Create a table and name it order\_items

```
create table order_items  
(  
    order_item_id int not null  
        primary key,  
    order_id      int null,  
    product_id    int null,  
    quantity      int null,  
    constraint order_items_ibfk_1  
        foreign key (order_id) references orders (order_id),
```

```
        constraint order_items_ibfk_2
            foreign key (product_id) references products (product_id)
    );
```

#### **4. Create a table and name it orders**

create table orders

```
(
    order_id      int            not null
        primary key,
    customer_id   int            null,
    order_date    date            null,
    status        varchar(20)    null,
    constraint orders_ibfk_1
        foreign key (customer_id) references customers (customer_id)
);
```

#### **5. Create a table and name it payments**

create table payments

```
(
    payment_id     int            not null
        primary key,
    order_id       int            null,
    payment_method  varchar(30)    null,
    amount         decimal(10, 2) null,
    payment_status  varchar(20)    null,
    payment_date   date            null,
    constraint payments_ibfk_1
        foreign key (order_id) references orders (order_id)
);
```

#### **6. Create a table and name it product**

create table products

```
(
    product_id     int            not null
```

```

        primary key,
product_name varchar(100)    null,
category_id  int             null,
price        decimal(10, 2) null,
stock        int             null,
constraint products_ibfk_1
        foreign key (category_id) references categories (category_id)
);

```

## 7. Insert records into table customer

```

INSERT INTO customers (customer_id, name, email, phone, city, created_at)
VALUES
(1, 'Amit Sharma', 'amit.sharma@gmail.com', '9876543201', 'Delhi', '2024-01-01'),
(2, 'Priya Singh', 'priya.singh@gmail.com', '9876543202', 'Mumbai', '2024-01-02'),
(3, 'Rahul Verma', 'rahul.verma@gmail.com', '9876543203', 'Bangalore', '2024-01-03'),
(4, 'Neha Patel', 'neha.patel@gmail.com', '9876543204', 'Ahmedabad', '2024-01-04'),
(5, 'Arjun Reddy', 'arjun.reddy@gmail.com', '9876543205', 'Hyderabad', '2024-01-05'),
(6, 'Sneha Iyer', 'sneha.iyer@gmail.com', '9876543206', 'Chennai', '2024-01-06'),
(7, 'Karan Mehta', 'karan.mehta@gmail.com', '9876543207', 'Pune', '2024-01-07'),
(8, 'Pooja Nair', 'pooja.nair@gmail.com', '9876543208', 'Kochi', '2024-01-08'),
(9, 'Vikas Gupta', 'vikas.gupta@gmail.com', '9876543209', 'Delhi', '2024-01-09'),
(10, 'Anjali Kapoor', 'anjali.kapoor@gmail.com', '9876543210', 'Noida', '2024-01-10'),
(11, 'Rohit Malhotra', 'rohit.m@gmail.com', '9876543211', 'Gurgaon', '2024-01-11'),
(12, 'Shweta Joshi', 'shweta.j@gmail.com', '9876543212', 'Nagpur', '2024-01-12'),
(13, 'Suresh Kumar', 'suresh.k@gmail.com', '9876543213', 'Coimbatore', '2024-01-13'),
(14, 'Divya Agarwal', 'divya.a@gmail.com', '9876543214', 'Jaipur', '2024-01-14'),
(15, 'Manish Pandey', 'manish.p@gmail.com', '9876543215', 'Patna', '2024-01-15'),
(16, 'Ritu Saxena', 'ritu.s@gmail.com', '9876543216', 'Indore', '2024-01-16'),
(17, 'Aakash Jain', 'aakash.j@gmail.com', '9876543217', 'Bhopal', '2024-01-17'),

```

(18, 'Kavya Rao', 'kavya.rao@gmail.com', '9876543218', 'Bangalore', '2024-01-18'),  
(19, 'Nitin Arora', 'nitin.a@gmail.com', '9876543219', 'Amritsar', '2024-01-19'),  
(20, 'Meena Chawla', 'meena.c@gmail.com', '9876543220', 'Chandigarh', '2024-01-20'),  
(21, 'Harsh Vardhan', 'harsh.v@gmail.com', '9876543221', 'Kanpur', '2024-01-21'),  
(22, 'Pallavi Kulkarni', 'pallavi.k@gmail.com', '9876543222', 'Nashik', '2024-01-22'),  
(23, 'Sanjay Mishra', 'sanjay.m@gmail.com', '9876543223', 'Varanasi', '2024-01-23'),  
(24, 'Nidhi Bansal', 'nidhi.b@gmail.com', '9876543224', 'Meerut', '2024-01-24'),  
(25, 'Rajesh Khanna', 'rajesh.k@gmail.com', '9876543225', 'Lucknow', '2024-01-25'),  
(26, 'Simran Kaur', 'simran.k@gmail.com', '9876543226', 'Ludhiana', '2024-01-26'),  
(27, 'Deepak Yadav', 'deepak.y@gmail.com', '9876543227', 'Rewari', '2024-01-27'),  
(28, 'Isha Mallick', 'isha.m@gmail.com', '9876543228', 'Kolkata', '2024-01-28'),  
(29, 'Tarun Goyal', 'tarun.g@gmail.com', '9876543229', 'Agra', '2024-01-29'),  
(30, 'Pankaj Tripathi', 'pankaj.t@gmail.com', '9876543230', 'Prayagraj', '2024-01-30'),  
(31, 'Rakesh Sinha', 'rakesh.s@gmail.com', '9876543231', 'Ranchi', '2024-02-01'),  
(32, 'Monika Sehgal', 'monika.s@gmail.com', '9876543232', 'Panipat', '2024-02-02'),  
(33, 'Ankit Tiwari', 'ankit.t@gmail.com', '9876543233', 'Rewa', '2024-02-03'),  
(34, 'Payal Deshmukh', 'payal.d@gmail.com', '9876543234', 'Aurangabad', '2024-02-04'),  
(35, 'Gaurav Bhatia', 'gaurav.b@gmail.com', '9876543235', 'Faridabad', '2024-02-05'),  
(36, 'Saloni Mittal', 'saloni.m@gmail.com', '9876543236', 'Hisar', '2024-02-06'),  
(37, 'Yogesh Patil', 'yogesh.p@gmail.com', '9876543237', 'Jalgaon', '2024-02-07'),  
(38, 'Namrata Kulkarni', 'namrata.k@gmail.com', '9876543238', 'Kolhapur', '2024-02-08'),  
(39, 'Mohit Chauhan', 'mohit.c@gmail.com', '9876543239', 'Dehradun', '2024-02-09'),  
(40, 'Rina Dutta', 'rina.d@gmail.com', '9876543240', 'Siliguri', '2024-02-10'),  
(41, 'Abhishek Roy', 'abhishek.r@gmail.com', '9876543241', 'Durgapur', '2024-02-11'),

```
(42, 'Kirti Desai', 'kirti.d@gmail.com', '9876543242', 'Surat', '2024-02-12'),
(43, 'Ashish Tomar', 'ashish.t@gmail.com', '9876543243', 'Mathura', '2024-02-13'),
(44, 'Preeti Malviya', 'preeti.m@gmail.com', '9876543244', 'Ujjain', '2024-02-14'),
(45, 'Naveen Chandra', 'naveen.c@gmail.com', '9876543245', 'Ballia', '2024-02-15'),
(46, 'Priyanka Soni', 'priyanka.s@gmail.com', '9876543246', 'Sagar', '2024-02-16'),
(47, 'Akash Solanki', 'akash.s@gmail.com', '9876543247', 'Dewas', '2024-02-17'),
(48, 'Rashmi Tandon', 'rashmi.t@gmail.com', '9876543248', 'Bareilly', '2024-02-18'),
(49, 'Vivek Oberoi', 'vivek.o@gmail.com', '9876543249', 'Alwar', '2024-02-19'),
(50, 'Sonal Trivedi', 'sonal.t@gmail.com', '9876543250', 'Rajkot', '2024-02-20');
```

**8. Display all the details from the table customers.**

```
SELECT * FROM customers;
```

**9. Insert records into table categories.**

```
INSERT INTO categories (category_id, category_name) VALUES
(1, 'Electronics'),
(2, 'Electronics'),
(3, 'Electronics'),
(4, 'Electronics'),
(5, 'Electronics'),
(6, 'Fashion'),
(7, 'Fashion'),
(8, 'Fashion'),
(9, 'Fashion'),
(10, 'Fashion'),
(11, 'Home Appliances'),
(12, 'Home Appliances'),
(13, 'Home Appliances'),
(14, 'Home Appliances'),
(15, 'Home Appliances');
```

(16, 'Books'),  
(17, 'Books'),  
(18, 'Books'),  
(19, 'Books'),  
(20, 'Books'),  
(21, 'Sports'),  
(22, 'Sports'),  
(23, 'Sports'),  
(24, 'Sports'),  
(25, 'Sports'),  
(26, 'Electronics Accessories'),  
(27, 'Electronics Accessories'),  
(28, 'Electronics Accessories'),  
(29, 'Electronics Accessories'),  
(30, 'Electronics Accessories'),  
(31, 'Fashion Accessories'),  
(32, 'Fashion Accessories'),  
(33, 'Fashion Accessories'),  
(34, 'Fashion Accessories'),  
(35, 'Fashion Accessories'),  
(36, 'Kitchen Appliances'),  
(37, 'Kitchen Appliances'),  
(38, 'Kitchen Appliances'),  
(39, 'Kitchen Appliances'),  
(40, 'Kitchen Appliances'),  
(41, 'Health & Fitness'),  
(42, 'Health & Fitness'),  
(43, 'Health & Fitness'),  
(44, 'Health & Fitness'),  
(45, 'Health & Fitness'),  
(46, 'Stationery'),

```
(47, 'Stationery'),  
(48, 'Stationery'),  
(49, 'Stationery'),  
(50, 'Stationery');
```

**10. Insert records into table categories.**

```
SELECT * FROM categories;
```

**11. Display all the details from the table products.**

```
INSERT INTO products (product_id, product_name, category_id, price, stock)  
VALUES
```

```
(101, 'Wireless Mouse', 1, 799, 50),  
(102, 'Keyboard', 1, 1299, 40),  
(103, 'Smartphone', 1, 15999, 25),  
(104, 'Laptop', 1, 55999, 15),  
(105, 'Bluetooth Headphones', 1, 2499, 35),  
(106, 'Smart Watch', 1, 4999, 30),  
(107, 'USB Cable', 26, 299, 100),  
(108, 'Power Bank', 26, 1999, 45),  
(109, 'Tablet', 1, 22999, 20),  
(110, 'Webcam', 26, 1799, 25),  
(111, 'T-Shirt', 6, 499, 120),  
(112, 'Jeans', 6, 1499, 80),  
(113, 'Formal Shirt', 6, 1299, 60),  
(114, 'Women''s Top', 6, 899, 70),  
(115, 'Jacket', 6, 2999, 40),  
(116, 'Sneakers', 6, 2599, 50),  
(117, 'Running Shoes', 21, 2999, 45),  
(118, 'Sandals', 31, 999, 65),  
(119, 'Cap', 31, 399, 90),  
(120, 'Socks Pack', 31, 299, 150),  
(121, 'Refrigerator', 11, 24999, 10),  
(122, 'Washing Machine', 11, 18999, 12),
```



```

(123, 'Microwave Oven', 36, 9999, 18),
(124, 'Air Conditioner', 11, 34999, 8),
(125, 'Vacuum Cleaner', 36, 6999, 20),
(126, 'Electric Kettle', 36, 1999, 30),
(127, 'Mixer Grinder', 36, 3999, 25),
(128, 'Water Purifier', 36, 12999, 15),
(129, 'Ceiling Fan', 11, 2499, 35),
(130, 'Room Heater', 36, 2999, 22),
(131, 'Fiction Book', 16, 499, 100),
(132, 'Non-Fiction Book', 16, 699, 90),
(133, 'Exam Preparation Book', 16, 899, 70),
(134, 'Children Story Book', 16, 399, 120),
(135, 'Notebook Pack', 46, 299, 200),
(136, 'Pen Set', 46, 199, 300),
(137, 'Office Diary', 46, 249, 150),
(138, 'Drawing Kit', 46, 599, 80),
(139, 'Magazine Subscription', 16, 999, 60),
(140, 'Comic Book', 16, 349, 110),
(141, 'Cricket Bat', 21, 2499, 30),
(142, 'Football', 21, 999, 50),
(143, 'Badminton Racket', 21, 1799, 40),
(144, 'Tennis Ball Pack', 21, 499, 70),
(145, 'Yoga Mat', 41, 899, 60),
(146, 'Dumbbell Set', 41, 2999, 25),
(147, 'Skipping Rope', 41, 299, 100),
(148, 'Gym Gloves', 41, 499, 90),
(149, 'Cycle Helmet', 21, 1999, 35),
(150, 'Sports Water Bottle', 21, 349, 120);

```

**11. Insert records into table products.**

```
SELECT * FROM products;
```

## 12. Display all the details from the table orders.

```
INSERT INTO orders (order_id, customer_id, order_date, status) VALUES
(1001,1,'2024-03-01','Delivered'),
(1002,2,'2024-03-02','Delivered'),
(1003,3,'2024-03-03','Cancelled'),
(1004,4,'2024-03-04','Shipped'),
(1005,5,'2024-03-05','Delivered'),
(1006,6,'2024-03-06','Delivered'),
(1007,7,'2024-03-07','Pending'),
(1008,8,'2024-03-08','Shipped'),
(1009,9,'2024-03-09','Delivered'),
(1010,10,'2024-03-10','Cancelled'),
(1011,11,'2024-03-11','Delivered'),
(1012,12,'2024-03-12','Pending'),
(1013,13,'2024-03-13','Shipped'),
(1014,14,'2024-03-14','Delivered'),
(1015,15,'2024-03-15','Delivered'),
(1016,16,'2024-03-16','Cancelled'),
(1017,17,'2024-03-17','Shipped'),
(1018,18,'2024-03-18','Delivered'),
(1019,19,'2024-03-19','Pending'),
(1020,20,'2024-03-20','Delivered'),
(1021,21,'2024-03-21','Delivered'),
(1022,22,'2024-03-22','Shipped'),
(1023,23,'2024-03-23','Delivered'),
(1024,24,'2024-03-24','Cancelled'),
(1025,25,'2024-03-25','Delivered'),
(1026,26,'2024-03-26','Delivered'),
(1027,27,'2024-03-27','Pending'),
(1028,28,'2024-03-28','Delivered'),
(1029,29,'2024-03-29','Shipped'),
```

```
(1030,30,'2024-03-30','Delivered'),
(1031,31,'2024-04-01','Cancelled'),
(1032,32,'2024-04-02','Delivered'),
(1033,33,'2024-04-03','Shipped'),
(1034,34,'2024-04-04','Delivered'),
(1035,35,'2024-04-05','Delivered'),
(1036,36,'2024-04-06','Pending'),
(1037,37,'2024-04-07','Shipped'),
(1038,38,'2024-04-08','Delivered'),
(1039,39,'2024-04-09','Cancelled'),
(1040,40,'2024-04-10','Delivered'),
(1041,41,'2024-04-11','Delivered'),
(1042,42,'2024-04-12','Shipped'),
(1043,43,'2024-04-13','Delivered'),
(1044,44,'2024-04-14','Delivered'),
(1045,45,'2024-04-15','Pending'),
(1046,46,'2024-04-16','Delivered'),
(1047,47,'2024-04-17','Shipped'),
(1048,48,'2024-04-18','Delivered'),
(1049,49,'2024-04-19','Delivered'),
(1050,50,'2024-04-20','Delivered');
```

**13. Display all the details from the table orders.**

```
SELECT * FROM orders;
```

**14.. Insert records into table orders\_items.**

```
INSERT INTO order_items (order_item_id, order_id, product_id, quantity)
VALUES
```

```
(1,1001,101,2),
(2,1001,111,1),
(3,1002,103,1),
(4,1002,112,2),
(5,1003,104,1),
```

(6,1003,113,1),  
(7,1004,105,1),  
(8,1004,114,2),  
(9,1005,106,1),  
(10,1005,115,1),  
(11,1006,107,3),  
(12,1006,116,1),  
(13,1007,108,1),  
(14,1007,117,2),  
(15,1008,109,1),  
(16,1008,118,1),  
(17,1009,110,2),  
(18,1009,119,1),  
(19,1010,101,1),  
(20,1010,120,2),  
(21,1011,102,1),  
(22,1011,121,1),  
(23,1012,103,2),  
(24,1012,122,1),  
(25,1013,104,1),  
(26,1013,123,1),  
(27,1014,105,1),  
(28,1014,124,1),  
(29,1015,106,2),  
(30,1015,125,1),  
(31,1016,107,1),  
(32,1016,126,2),  
(33,1017,108,1),  
(34,1017,127,1),  
(35,1018,109,1),  
(36,1018,128,1),

```
(37,1019,110,1),
(38,1019,129,2),
(39,1020,111,1),
(40,1020,130,1),
(41,1021,112,1),
(42,1021,131,2),
(43,1022,113,1),
(44,1022,132,1),
(45,1023,114,1),
(46,1023,133,1),
(47,1024,115,1),
(48,1024,134,2),
(49,1025,116,1),
(50,1025,135,3);
```

**15. Display all the details from the table orders\_items.**

```
SELECT * FROM order_items;
```

**16. Insert records into table payments.**

```
INSERT INTO payments (payment_id, order_id, payment_date, amount,
payment_method, payment_status) VALUES
```

```
(5001,1001,'2024-03-01',1598,'Credit Card','Completed'),
(5002,1002,'2024-03-02',3198,'Debit Card','Completed'),
(5003,1003,'2024-03-03',0,'UPI','Failed'),
(5004,1004,'2024-03-04',4498,'Credit Card','Completed'),
(5005,1005,'2024-03-05',7998,'UPI','Completed'),
(5006,1006,'2024-03-06',4198,'Debit Card','Completed'),
(5007,1007,'2024-03-07',4999,'Credit Card','Pending'),
(5008,1008,'2024-03-08',1998,'UPI','Completed'),
(5009,1009,'2024-03-09',3598,'Debit Card','Completed'),
(5010,1010,'2024-03-10',0,'Credit Card','Failed'),
(5011,1011,'2024-03-11',1299,'UPI','Completed'),
(5012,1012,'2024-03-12',2598,'Debit Card','Pending'),
```

(5013,1013,'2024-03-13',5599,'Credit Card','Completed'),  
(5014,1014,'2024-03-14',1798,'UPI','Completed'),  
(5015,1015,'2024-03-15',7998,'Credit Card','Completed'),  
(5016,1016,'2024-03-16',0,'Debit Card','Failed'),  
(5017,1017,'2024-03-17',6998,'UPI','Completed'),  
(5018,1018,'2024-03-18',2298,'Credit Card','Completed'),  
(5019,1019,'2024-03-19',0,'Debit Card','Pending'),  
(5020,1020,'2024-03-20',999,'UPI','Completed'),  
(5021,1021,'2024-03-21',1499,'Credit Card','Completed'),  
(5022,1022,'2024-03-22',899,'Debit Card','Completed'),  
(5023,1023,'2024-03-23',1798,'UPI','Completed'),  
(5024,1024,'2024-03-24',0,'Credit Card','Failed'),  
(5025,1025,'2024-03-25',2599,'UPI','Completed'),  
(5026,1026,'2024-03-26',1999,'Debit Card','Completed'),  
(5027,1027,'2024-03-27',0,'Credit Card','Pending'),  
(5028,1028,'2024-03-28',4999,'UPI','Completed'),  
(5029,1029,'2024-03-29',3499,'Debit Card','Completed'),  
(5030,1030,'2024-03-30',1299,'Credit Card','Completed'),  
(5031,1031,'2024-04-01',0,'UPI','Failed'),  
(5032,1032,'2024-04-02',1799,'Debit Card','Completed'),  
(5033,1033,'2024-04-03',3999,'Credit Card','Completed'),  
(5034,1034,'2024-04-04',899,'UPI','Completed'),  
(5035,1035,'2024-04-05',2999,'Debit Card','Completed'),  
(5036,1036,'2024-04-06',0,'Credit Card','Pending'),  
(5037,1037,'2024-04-07',1999,'UPI','Completed'),  
(5038,1038,'2024-04-08',2499,'Debit Card','Completed'),  
(5039,1039,'2024-04-09',0,'Credit Card','Failed'),  
(5040,1040,'2024-04-10',1299,'UPI','Completed'),  
(5041,1041,'2024-04-11',1599,'Debit Card','Completed'),  
(5042,1042,'2024-04-12',899,'Credit Card','Completed'),  
(5043,1043,'2024-04-13',1799,'UPI','Completed'),

```
(5044,1044,'2024-04-14',999,'Debit Card','Completed'),  
(5045,1045,'2024-04-15',0,'Credit Card','Pending'),  
(5046,1046,'2024-04-16',1999,'UPI','Completed'),  
(5047,1047,'2024-04-17',2499,'Debit Card','Completed'),  
(5048,1048,'2024-04-18',1299,'Credit Card','Completed'),  
(5049,1049,'2024-04-19',1599,'UPI','Completed'),  
(5050,1050,'2024-04-20',1999,'Debit Card','Completed');
```

**17.Display all the details from the table orders.**

```
SELECT * FROM payments;
```

**- BASIC SELECT AND FILTERING--**

**-- 1. Write a Query to Display all the products with price and stock.**  
SELECT PRODUCT\_NAME, PRICE, STOCK FROM products;

**-- 2. Write a Query to Find products costing more than RS.5000**  
SELECT \* FROM products  
WHERE price>5000;

**-- 3. Write a Query to Display ORDERS that are delivered.**  
SELECT \* FROM orders  
WHERE status="Delivered";

**-- 4. Write a Query to Display ORDERS that are Cancelled.**  
SELECT \* FROM orders  
WHERE status="Cancelled";

**-- 5. Write a Query to Display Orders placed after 1st April 2024**  
SELECT \* FROM orders  
WHERE order\_date='2024-04-01';

**-- 6. Write a Query to Display distinct product categories**  
SELECT DISTINCT category\_name  
FROM categories;

**-- 7. Write a Query to Display Products priced between 1000 and 5000**  
SELECT product\_name  
FROM products  
WHERE price BETWEEN 1000 AND 5000;

**-- 8. Write a Query to Display Orders sorted by latest date**  
SELECT \* FROM orders  
ORDER BY order\_date DESC ;



**-- AGGREGATE FUNCTION ---**

**-- 9. Write a Query to Display total number of orders**

```
SELECT COUNT(*) AS TOTAL_NUMBER_OF_ORDERS  
FROM orders;
```

**-- 10. Write a Query to display Total revenue from completed payments**

```
SELECT SUM(payments.amount) AS TOTAL_REVENUE  
FROM payments  
WHERE payment_status='COMPLETED';
```

**-- 11. Write a Query to display Average product price**

```
SELECT AVG(products.price) AS AVG_PRODUCT_PRICE  
FROM products;
```

**-- 12. Write a Query to display Highest product price**

```
SELECT MAX(products.price) AS HIGHEST_PRODUCT_PRICE  
FROM products;
```

**-- 13. Write a Query to display Lowest product price**

```
SELECT MIN(products.price) AS LOWEST_PRODUCT_PRICE  
FROM products;
```

**-- 14. Write a Query to display Total quantity sold**

```
SELECT SUM(order_items.quantity) AS TOTAL_QTY  
FROM order_items;
```

## -- GROUPBY CLAUSE AND HAVING CLAUSE

-- 15. Write a Query to display Total revenue per payment method

```
SELECT payment_method, SUM(amount) AS total_revenue
FROM payments
WHERE payment_status = 'COMPLETED'
GROUP BY payment_method;
```

-- 16. Write a Query to display Categories having more than 5 products

```
SELECT COUNT(*), category_id
FROM products
GROUP BY category_id
HAVING COUNT(*) > 5;
```

-- 17. Write a Query to display Customers with more than 1 order

```
SELECT customer_id, COUNT(*) AS total_orders
FROM orders
GROUP BY customer_id
HAVING COUNT(*) > 1;
```

## **JOINS(INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN, CROSS JOIN, SELF JOIN)**

**-- 18. Write a Query to Display all orders along with customer names**

```
SELECT orders.order_id, orders.customer_id, customer.name, orders.order_date
FROM orders INNER JOIN customer
ON orders.customer_id = customer.customer_id;
```

**-- 19. Write a Query to Find all customers and their orders (including customers with no orders)**

```
SELECT customer.name, orders.customer_id, orders.order_id
FROM customer LEFT JOIN orders
ON orders.customer_id = customer.customer_id;
```

**-- 20. Write a Query to display all orders and customer details (even if customer data is missing.)**

```
SELECT
customer.name, customer.email, customer.phone, customer.city, orders.customer_id,
order_id, order_date
FROM customer RIGHT JOIN orders
ON orders.customer_id = customer.customer_id;
```

**-- 21. Write a Query to display Get all customers and all orders (matched or unmatched)**

```
SELECT orders.customer_id, customer.name, orders.order_id, orders.order_date
FROM customer
LEFT JOIN orders
ON customer.customer_id = orders.customer_id
```

UNION

```
SELECT orders.customer_id, customer.name, orders.order_id, orders.order_date
FROM customer
RIGHT JOIN orders
ON customer.customer_id = orders.customer_id;
```

**-- 22. Write a Query to Generate all possible combinations of products and categories**

```
SELECT products.product_name, categories.category_name
FROM products CROSS JOIN categories;
```

**-- 23. Write a Query to Find customers who share the same city**

```
SELECT C1.customer_id, C1.name, C1.city
FROM customer C1
JOIN customer C2
ON C1.city = C2.city
AND C1.customer_id <> C2.customer_id;
```

#### -- SUBQUERIES

-- 24. Write a Query to Identify all products whose price is higher than the average price of all products.

```
SELECT products.product_id,products.product_name,products.price
FROM products
WHERE price>( SELECT AVG(price)
              FROM products);
```

-- 25. Write a Query to Find customers who have placed at least one order.

```
SELECT customer.name,customer_id
FROM customer
WHERE customer_id IN (SELECT DISTINCT customer_id
                     FROM orders);
```