Mini Project Demo - Documentation

Subject: Principles of Database Management System

Topic: “Query Statements for Online Store Database system”

Done By:

1. Abishek B – 18ME1002
2. Nadesh J – 18ME1044
3. Vijai A – 18ME1073

- The Schema of our Database consists of 6 Tables and They are,

  1.Customers

  2. order_items

  3. order_statuses

  4. orders

  5. Products

  6. Shippers

- **To Create Table "Customers"**:

```
CREATE TABLE `customers` (
  `customer_id` int NOT NULL AUTO_INCREMENT,
  `first_name` varchar(50) NOT NULL,
  `last_name` varchar(50) NOT NULL,
  `birth_date` date DEFAULT NULL,
  `phone` varchar(50) DEFAULT NULL,
  `address` varchar(50) NOT NULL,
  `city` varchar(50) NOT NULL,
  `state` char(2) NOT NULL,
  `points` int NOT NULL DEFAULT '0',
  PRIMARY KEY (`customer_id`)
  )
```

- **Inserting Values to the Customer Table**:

INSERT INTO `customers` VALUES (1,'Arun','Kumaran','1992-03-28','9003464371','7 Sage Street','Chennai','TN',2273);

INSERT INTO `customers` VALUES (2,'Mohan','Raj','1986-04-13','6443435102','40, Kalpana Villas, Marathahalli','Mysore','KA',947);

INSERT INTO `customers` VALUES (3,'Charlie','Pratap','1985-02-07','9892516822','251 Springs Junction','Trivandrum','KL',2967);

INSERT INTO `customers` VALUES (4,'Aishwarya','Chakrabarti','1979-04-14','7574382813','23, Sodala nagar','Coimbatore','TN',457);

INSERT INTO `customers` VALUES (5,'Rakhi','Bava','1995-11-07',NULL,'56, Hinjewadi',' Vishakhapattanam ','AP',3675);

INSERT INTO `customers` VALUES (6,'Ram','Barhia','1991-09-04','8940353247','44, Anjana Apartments, Yeshwanthpura','Mumbai','MH',3073);

INSERT INTO `customers` VALUES (7,'Fatima','Dowson','1964-08-30','9434459726','54, Kharadi street','Bengaluru','KA',1672);

INSERT INTO `customers` VALUES (8,'Veena ','Venkatesh','1993-07-17','9415273977','20, v.o.c street, Lawspet','Puducherry','PY',205);

INSERT INTO `customers` VALUES (9,'Naresh','Karnik','2001-05-23','8591813744','60, Aundh street',' Ahmedabad','GJ',1486);

INSERT INTO `customers` VALUES (10,'Govind','Sunder','1975-10-13','9042463370','77, Priyanka Nagar','Delhi','DL',796);

- **Output of "Customers" Table**:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Arun	Kumaran	1992-03-28	9003464371	7 Sage Street	Chennai	TN	2273
2	Mohan	Raj	1986-04-13	6443435102	40, Kalpana Villas, Marathahalli	Mysore	KA	947
3	Charlie	Pratap	1985-02-04	9892516822	251 Springs Junction	Trivandrum	KL	2967
4	Aishwarya	Chakrabarti	1979-04-22	7574382813	23, Sodala nagar	Coimbatore	TN	457
5	Rakhi	Sonam	1995-11-07	NULL	56, Hinjewadi	Vishakhapattanam	AP	3675
6	Ram	Barhia	1991-09-04	8940353247	44, Anjana Apartments, Yeshwanthpura	Mumbai	MH	3073
7	Fatima	Dowson	1964-08-30	9434459726	54, Kharadi street	Bengaluru	KA	1672
8	Veena	Venkatesh	1993-07-15	9415273977	20, v.o.c street, Lawspet	Puducherry	PY	205
9	Naresh	Karnik	2001-05-13	8591813744	60, Aundh street	Ahmedabad	GJ	1486
10	Govind	Sunder	1975-10-16	9042463370	77, Priyanka Nagar	Delhi	DL	796
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- Similarly, the other 5 Tables:
  - order_items:

order_id	product_id	quantity	unit_price
1	4	4	150
2	1	2	45
2	4	4	145
2	6	2	90
3	3	10	125
4	3	7	120
4	10	7	65
5	2	3	155
6	1	4	9
6	2	4	160
6	3	4	120
6	5	1	50
7	3	7	110
8	5	2	55
8	8	2	30
9	6	5	85
10	1	10	45
10	9	9	80
NULL	NULL	NULL	NULL

  - order_statuses:

order_status_id	name
1	Processed
2	Shipped
3	Delivered
NULL	NULL

- orders:

order_id	customer_id	order_date	status	comments	shipped_date	shipper_id
1	6	2021-01-03	1	NULL	NULL	NULL
2	7	2020-08-02	3	NULL	2018-08-03	4
3	8	2020-02-16	3	NULL	2020-02-18	5
4	2	2020-12-25	2	NULL	2020-12-27	NULL
5	5	2020-10-25	3	NULL	2020-10-26	3
6	10	2020-12-28	1	NULL	NULL	NULL
7	2	2020-12-19	2	NULL	2020-12-26	4
8	5	2020-12-30	1	NULL	NULL	NULL
9	3	2020-07-05	3	NULL	2020-07-08	1
10	1	2020-11-29	2	NULL	2020-12-10	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL

- Products:

product_id	name	quantity_in_stock	unit_price
1	Atkins Snack Bar	70	40
2	Organic Hibiscus Tea, 16 Tea Bags	49	160
3	Antiperspirant Deodorant Stick	38	120
4	Cetaphil Facial Moisturizer	90	170
5	Xyliwhite Toothpaste Gel	94	55
6	Sugar Free Coffee Flavoring Syrup	14	85
7	Cheddar cheese Pringles	98	115
8	Blueberry Raspberry Drink Mix	26	30
9	kitchen white pepper	67	80
10	Broom - Push	6	65
NULL	NULL	NULL	NULL

- Shippers:

shipper_id	name
1	Hettinger LLC
2	Schinner-Predovic
3	Satterfield LLC
4	Mraz, Renner and Nolan
5	Waters, Mayert and Prohaska
NULL	NULL

- **SQL Queries for the above database:**

1) WRITE A QUERY TO RETURN FIRST NAME, LAST NAME AND PHONE NO. OF ALL CUSTOMERS

SQL Query:

```
-- The SELECT statement is used to select data from a database


SELECT first_name, last_name, phone

FROM customers ;
```

Output:

first_name	last_name	phone
Arun	Kumaran	9003464371
Mohan	Raj	6443435102
Charlie	Pratap	9892516822
Aishwarya	Chakrabarti	7574382813
Rakhi	Sonam	NULL
Ram	Barhia	8940353247
Fatima	Dowson	9434459726
Veena	Venkatesh	9415273977
Naresh	Karnik	8591813744
Govind	Sunder	9042463370

2) GET ALL PRODUCTS PRICED ABOVE Rs.100

SQL Query:

```
SELECT * FROM products

WHERE unit_price > 100 ;
```

Output:

product_id	name	quantity_in_stock	unit_price
2	Organic Hibiscus Tea, 16 Tea Bags	49	160
3	Antiperspirant Deodorant Stick	38	120
4	Cetaphil Facial Moisturizer	90	170
7	Cheddar cheese Pringles	98	115
NULL	NULL	NULL	NULL

3) GET ALL CUSTOMERS BORN AFTER 1990 OR HAVE 2000+ STORE POINTS AND LIVE IN A SOUTH INDIAN STATE

-- The AND operator displays a record if all the conditions separated by AND are TRUE

-- The OR operator displays a record if any of the conditions separated by OR is TRUE.

-- The IN operator allows you to specify multiple values in a WHERE clause

SQL Query:

```
SELECT * FROM customers

WHERE (birth_date >'1990-01-01' OR points>= 2000)

AND state IN ('TN','KL', 'AP','PY','KA') ;
```

Output:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Arun	Kumaran	1992-03-28	9003464371	7 Sage Street	Chennai	TN	2273
3	Charlie	Pratap	1985-02-04	9892516822	251 Springs Junction	Trivandrum	KL	2967
5	Rakhi	Sonam	1995-11-07	NULL	56, Hinjewadi	Vishakhapattanam	AP	3675
8	Veena	Venkatesh	1993-07-15	9415273977	20, v.o.c street, Lawspet	Puducherry	PY	205
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4) GET ALL THE ORDERS PLACED BETWEEN JUNE - NOVEMBER,2020 AND LIST THEM FROM RECENT

-- The BETWEEN operator selects values within a given range

SQL Query:

```
SELECT* FROM orders

    WHERE order_date BETWEEN '2020-06-01' AND '2020-11-30'

    ORDER BY order_date DESC ;
```

Output:

order_id	customer_id	order_date	status	comments	shipped_date	shipper_id
10	1	2020-11-29	2	NULL	2020-12-10	2
5	5	2020-10-25	3	NULL	2020-10-26	3
2	7	2020-08-02	3	NULL	2018-08-03	4
9	3	2020-07-05	3	NULL	2020-07-08	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

5) [i] GET ALL CUSTOMERS WITH LETTER 'O' IN THEIR NAME

SQL Query:

```
SELECT* FROM customers
    WHERE first_name LIKE '%O%'
    OR last_name LIKE '%O%';
```

Output:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
2	Mohan	Raj	1986-04-13	6443435102	40, Kalpana Villas, Marathahalli	Mysore	KA	947
5	Rakhi	Sonam	1995-11-07	NULL	56, Hinjewadi	Vishakhapattanam	AP	3675
7	Fatima	Dowson	1964-08-30	9434459726	54, Kharadi street	Bengaluru	KA	1672
10	Govind	Sunder	1975-10-16	9042463370	77, Priyanka Nagar	Delhi	DL	796
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

[ii] GET ALL SHIPPERS WITH 'A' AS THE SECOND LETTER IN THEIR NAME

SQL Query:

```
SELECT* FROM shippers
    WHERE name LIKE '_A%' ;
```

Output:

shipper_id	name
3	Satterfield LLC
5	Waters, Mayert and Prohaska
NULL	NULL

[iii] GET ALL CUSTOMERS WHOSE PHONE NUMBER DOES NOT START WITH '9'

SQL Query:

```
SELECT * FROM customers
    WHERE phone NOT LIKE '9%' ;
```

Output:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
2	Mohan	Raj	1986-04-13	6443435102	40, Kalpana Villas, Marathahalli	Mysore	KA	947
4	Aishwarya	Chakrabarti	1979-04-22	7574382813	23, Sodala nagar	Coimbatore	TN	457
6	Ram	Barhia	1991-09-04	8940353247	44, Anjana Apartments, Yeshwanthpura	Mumbai	MH	3073
9	Naresh	Karnik	2001-05-13	8591813744	60, Aundh street	Ahmedabad	GJ	1486
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6) GET ALL CUSTOMERS WHOSE LAST NAME HAS 'ra','ia' or 'na'

-- A regular expression (REGEXP) is a more powerful way of specifying a pattern for a complex search

-- Its similar to LIKE

SQL Query:

```
SELECT* FROM customers
WHERE last_name REGEXP '[rin]a' ;
```

Output:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Arun	Kumaran	1992-03-28	9003464371	7 Sage Street	Chennai	TN	2273
2	Mohan	Raj	1986-04-13	6443435102	40, Kalpana Villas, Marathahalli	Mysore	KA	947
3	Charlie	Pratap	1985-02-04	9892516822	251 Springs Junction	Trivandrum	KL	2967
4	Aishwarya	Chakrabarti	1979-04-22	7574382813	23, Sodala nagar	Coimbatore	TN	457
5	Rakhi	Sonam	1995-11-07	NULL	56, Hinjewadi	Vishakhapattanam	AP	3675
6	Ram	Barhia	1991-09-04	8940353247	44, Anjana Apartments, Yeshwanthpura	Mumbai	MH	3073
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7) GET THE ORDERS THAT ARE NOT SHIPPED

SQL Query:

```
SELECT * FROM  orders
      WHERE shipped_date IS NULL ;
```

Output:

order_id	customer_id	order_date	status	comments	shipped_date	shipper_id
1	6	2021-01-03	1	NULL	NULL	NULL
6	10	2020-12-28	1	NULL	NULL	NULL
8	5	2020-12-30	1	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

8) [i] WRITE A QUERY TO GET A EVEN ROWS OF A TABLE

SQL Query:

```
SELECT * FROM products
        WHERE product_id IN
  (SELECT product_id
   FROM products
   WHERE product_id%2 = 0);
```

Output:

product_id	name	quantity_in_stock	unit_price
2	Organic Hibiscus Tea, 16 Tea Bags	49	160
4	Cetaphil Facial Moisturizer	90	170
6	Sugar Free Coffee Flavoring Syrup	14	85
8	Blueberry Raspberry Drink Mix	26	30
10	Broom - Push	6	65
NULL	NULL	NULL	NULL

8) [ii] TO GET ODD ROWS OF A TABLE

SQL Query:

```
SELECT * FROM products
        WHERE product_id IN
  (SELECT product_id
   FROM products
   WHERE product_id%2 <> 0);
```

Output:

	product_id	name	quantity_in_stock	unit_price
▶	1	Atkins Snack Bar	70	40
	3	Antiperspirant Deodorant Stick	38	120
	5	Xyliwhite Toothpaste Gel	94	55
	7	Cheddar cheese Pringles	98	115
	9	kitchen white pepper	67	80
*	NULL	NULL	NULL	NULL

9) WRITE A QUERY TO JOIN ORDER ITEMS AND PRODUCTS TO SHOW ORDER WITH PRODUCT NAME [JOINING TWO TABLES]

SQL Query:

```
SELECT order_id , od.product_id ,name , quantity, od.unit_price
      FROM order_items od
            INNER JOIN products p
            ON od.product_id = p.product_id
      ORDER BY order_id ;
```

Output:

order_id	product_id	name	quantity	unit_price
1	4	Cetaphil Facial Moisturizer	4	150
2	1	Atkins Snack Bar	2	45
2	4	Cetaphil Facial Moisturizer	4	145
2	6	Sugar Free Coffee Flavoring Syrup	2	90
3	3	Antiperspirant Deodorant Stick	10	125
4	3	Antiperspirant Deodorant Stick	7	120
4	10	Broom - Push	7	65
5	2	Organic Hibiscus Tea, 16 Tea Bags	3	155
6	1	Atkins Snack Bar	4	9
6	2	Organic Hibiscus Tea, 16 Tea Bags	4	160
6	3	Antiperspirant Deodorant Stick	4	120
6	5	Xyliwhite Toothpaste Gel	1	50
7	3	Antiperspirant Deodorant Stick	7	110
8	5	Xyliwhite Toothpaste Gel	2	55
8	8	Blueberry Raspberry Drink Mix	2	30
9	6	Sugar Free Coffee Flavoring Syrup	5	85
10	1	Atkins Snack Bar	10	45
10	9	kitchen white pepper	9	80

10) WRITE A QUERY TO SHOW CONSOLIDATED ORDER REPORT WITH ORDER DATE, CUSTOMER NAME, STATUS OF ORDER [JOINING MULTIPLE TABLES]

SQL Query:

```
SELECT  o.order_id,o.order_date,
        c.first_name,c.last_name,
        os.name AS status,o.shipped_date
    FROM orders o
        INNER JOIN customers c
    ON o.customer_id = c.customer_id
      INNER JOIN order_statuses os
    ON o.status = os.order_status_id
    ORDER BY order_id ;
```

Output:

order_id	order_date	first_name	last_name	status	shipped_date
1	2021-01-03	Ram	Barhia	Processed	NULL
2	2020-08-02	Fatima	Dowson	Delivered	2018-08-03
3	2020-02-16	Veena	Venkatesh	Delivered	2020-02-18
4	2020-12-25	Mohan	Raj	Shipped	2020-12-27
5	2020-10-25	Rakhi	Sonam	Delivered	2020-10-26
6	2020-12-28	Govind	Sunder	Processed	NULL
7	2020-12-19	Mohan	Raj	Shipped	2020-12-26
8	2020-12-30	Rakhi	Sonam	Processed	NULL
9	2020-07-05	Charlie	Pratap	Delivered	2020-07-08
10	2020-11-29	Arun	Kumaran	Shipped	2020-12-10

11) WRITE A QUERY TO JOIN CUSTOMER AND ORDER TABLE WITH OUTER JOIN TO SHOW CUSTOMER NAME,ID AND ORDER ID

SQL Query:

```
SELECT c.customer_id, c.first_name,c.last_name, o.order_id
        FROM customers c
        LEFT OUTER JOIN orders o
        USING(customer_id) ;
 -- 'using' command can also be used , instead of 'ON c.customer_id = o.customer_id'
```

Output:

customer_id	first_name	last_name	order_id
1	Arun	Kumaran	10
2	Mohan	Raj	4
2	Mohan	Raj	7
3	Charlie	Pratap	9
4	Aishwarya	Chakrabarti	NULL
5	Rakhi	Sonam	5
5	Rakhi	Sonam	8
6	Ram	Barhia	1
7	Fatima	Dowson	2
8	Veena	Venkatesh	3
9	Naresh	Karnik	NULL
10	Govind	Sunder	6

12) WRITE A QUERY TO DO A CROSS JOIN BETWEEN CUSTOMER AND PRODUCT TABLE

SQL Query:

```
SELECT c.first_name AS CUSTOMER, p.name AS PRODUCT
        FROM customers c
        CROSS JOIN products p
                ORDER BY c.first_name ;
--(Output too long to be shown in an image)
```

13) WRITE A QUERY TO CLASSIFY CUSTOMER DEPENDING UPON THEIR STORE POINTS

SQL Query:

```
-- UNION IS USED TO JOIN ROWS OF SAME TABLE OR DIFFERENT TABLES


       SELECT customer_id,first_name,last_name,points,'Silver' AS
Member_Type
       FROM customers
        WHERE points <= 1000
    UNION
        SELECT customer_id,first_name,last_name,points,'Gold' AS
Member_Type
        FROM customers
        WHERE points BETWEEN 1000 AND 2500
    UNION
        SELECT customer_id,first_name,last_name,points,'Platinum' AS
Member_Type
        FROM customers
        WHERE points> 2500
```

ORDER BY customer_id ;

Output:

customer_id	first_name	last_name	points	Member_Type
1	Arun	Kumaran	2273	Gold
2	Mohan	Raj	947	Silver
3	Charlie	Pratap	2967	Platinum
4	Aishwarya	Chakrabarti	457	Silver
5	Rakhi	Sonam	3675	Platinum
6	Ram	Barhia	3073	Platinum
7	Fatima	Dowson	1672	Gold
8	Veena	Venkatesh	205	Silver
9	Naresh	Karnik	1486	Gold
10	Govind	Sunder	796	Silver

14) WRITE A QUERY TO ADD A NEW CUSTOMER INTO THE DATABASE

SQL Query:

-- The INSERT INTO statement is used to insert new records in a table.

INSERT INTO customers

VALUES (11, 'NEW','CUSTOMER','2000-01-01','123456789','c.address','c.city','ST',DEFAULT) ;

Output:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Arun	Kumaran	1992-03-28	9003464371	7 Sage Street	Chennai	TN	2273
2	Mohan	Raj	1986-04-13	6443435102	40, Kalpana Villas, Marathahalli	Mysore	KA	947
3	Charlie	Pratap	1985-02-04	9892516822	251 Springs Junction	Trivandrum	KL	2967
4	Aishwarya	Chakrabarti	1979-04-22	7574382813	23, Sodala nagar	Coimbatore	TN	457
5	Rakhi	Sonam	1995-11-07	NULL	56, Hinjewadi	Vishakhapattanam	AP	3675
6	Ram	Barhia	1991-09-04	8940353247	44, Anjana Apartments, Yeshwanthpura	Mumbai	MH	3073
7	Fatima	Dowson	1964-08-30	9434459726	54, Kharadi street	Bengaluru	KA	1672
8	Veena	Venkatesh	1993-07-15	9415273977	20, v.o.c street, Lawspet	Puducherry	PY	205
9	Naresh	Karnik	2001-05-13	8591813744	60, Aundh street	Ahmedabad	GJ	1486
10	Govind	Sunder	1975-10-16	9042463370	77, Priyanka Nagar	Delhi	DL	796
11	NEW	CUSTOMER	2000-01-01	123456789	c.address	c.city	ST	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

15) WRITE A QUERY TO COPY DELIVERED ORDERS INTO A NEW TABLE

SQL Query:

```
CREATE TABLE orders_delivered AS
        SELECT order_id,customer_id, order_date,shipped_date,sh.name AS
Shipper_Name
        FROM orders o
        JOIN shippers sh
        USING(shipper_id)
        WHERE o.status= 3
        ORDER BY order_id ;
```

Output:

	order_id	customer_id	order_date	shipped_date	Shipper_Name
▶	2	7	2020-08-02	2018-08-03	Mraz, Renner and Nolan
	3	8	2020-02-16	2020-02-18	Waters, Mayert and Prohaska
	5	5	2020-10-25	2020-10-26	Satterfield LLC
	9	3	2020-07-05	2020-07-08	Hettinger LLC

16) WRITE A QUERY TO ADD COMMENT ON ORDERS OF CUSTOMERS HAVING MORE THAN 2500 POINTS

SQL Query:

```
-- The UPDATE statement is used to modify the existing records in a table.


        UPDATE orders
        SET comments = 'Prime Customer'
        WHERE customer_id IN
                (SELECT customer_id
                 FROM customers
```

WHERE points >2500) ;

Output:

order_id	customer_id	order_date	status	comments	shipped_date	shipper_id
1	6	2021-01-03	1	Prime Customer	NULL	NULL
2	7	2020-08-02	3	NULL	2018-08-03	4
3	8	2020-02-16	3	NULL	2020-02-18	5
4	2	2020-12-25	2	NULL	2020-12-27	NULL
5	5	2020-10-25	3	Prime Customer	2020-10-26	3
6	10	2020-12-28	1	NULL	NULL	NULL
7	2	2020-12-19	2	NULL	2020-12-26	4
8	5	2020-12-30	1	Prime Customer	NULL	NULL
9	3	2020-07-05	3	Prime Customer	2020-07-08	1
10	1	2020-11-29	2	NULL	2020-12-10	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL

17) WRITE A QUERY TO DELETE A ROW FROM CUSTOMER TABLE

SQL Query:

-- The DELETE statement is used to delete existing records in a table.

```
DELETE FROM customers
WHERE customer_id = 11 ;
```

Output:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Arun	Kumaran	1992-03-28	9003464371	7 Sage Street	Chennai	TN	2273
2	Mohan	Raj	1986-04-13	6443435102	40, Kalpana Villas, Marathahalli	Mysore	KA	947
3	Charlie	Pratap	1985-02-04	9892516822	251 Springs Junction	Trivandrum	KL	2967
4	Aishwarya	Chakrabarti	1979-04-22	7574382813	23, Sodala nagar	Coimbatore	TN	457
5	Rakhi	Sonam	1995-11-07	NULL	56, Hinjewadi	Vishakhapattanam	AP	3675
6	Ram	Barhia	1991-09-04	8940353247	44, Anjana Apartments, Yeshwanthpura	Mumbai	MH	3073
7	Fatima	Dowson	1964-08-30	9434459726	54, Kharadi street	Bengaluru	KA	1672
8	Veena	Venkatesh	1993-07-15	9415273977	20, v.o.c street, Lawspet	Puducherry	PY	205
9	Naresh	Karnik	2001-05-13	8591813744	60, Aundh street	Ahmedabad	GJ	1486
10	Govind	Sunder	1975-10-16	9042463370	77, Priyanka Nagar	Delhi	DL	796
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

18) WRITE a QUERY TO ADD A NEW COLUMN 'Email_id' TO THE CUSTOMER TABLE

SQL Query:

-- The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

```
ALTER TABLE customers
ADD Email_id VARCHAR(50) ;
```

Output:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points	Email_id
1	Arun	Kumaran	1992-03-28	9003464371	7 Sage Street	Chennai	TN	2273	NULL
2	Mohan	Raj	1986-04-13	6443435102	40, Kalpana Villas, Marathahalli	Mysore	KA	947	NULL
3	Charlie	Pratap	1985-02-04	9892516822	251 Springs Junction	Trivandrum	KL	2967	NULL
4	Aishwarya	Chakrabarti	1979-04-22	7574382813	23, Sodala nagar	Coimbatore	TN	457	NULL
5	Rakhi	Sonam	1995-11-07	NULL	56, Hinjewadi	Vishakhapattanam	AP	3675	NULL
6	Ram	Barhia	1991-09-04	8940353247	44, Anjana Apartments, Yeshwanthpura	Mumbai	MH	3073	NULL
7	Fatima	Dowson	1964-08-30	9434459726	54, Kharadi street	Bengaluru	KA	1672	NULL
8	Veena	Venkatesh	1993-07-15	9415273977	20, v.o.c street, Lawspet	Puducherry	PY	205	NULL
9	Naresh	Karnik	2001-05-13	8591813744	60, Aundh street	Ahmedabad	GJ	1486	NULL
10	Govind	Sunder	1975-10-16	9042463370	77, Priyanka Nagar	Delhi	DL	796	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- TO DROP A COLUMN

```
ALTER TABLE customers
    DROP COLUMN Email_id;
```

19) GET THE LIST OF TOP ORDERED ITEMS

SQL Query:

-- The GROUP BY statement groups rows that have the same values into summary rows

-- The COUNT() function returns the number of rows that matches a specified criterion.

```
SELECT p.name AS 'Product', COUNT(p.product_id) AS 'No. of times ordered'
FROM order_items od
    INNER JOIN products p
    ON od.product_id = p.product_id
```

```
GROUP BY p.name
ORDER BY COUNT(p.product_id) DESC ;
```

Output:

Product	No. of times ordered
Antiperspirant Deodorant Stick	4
Atkins Snack Bar	3
Organic Hibiscus Tea, 16 Tea Bags	2
Cetaphil Facial Moisturizer	2
Xyliwhite Toothpaste Gel	2
Sugar Free Coffee Flavoring Syrup	2
Blueberry Raspberry Drink Mix	1
kitchen white pepper	1
Broom - Push	1

20) GET THE CUSTOMER WITH 2ND HIGHEST AND 2ND LOWEST STORE POINTS

SQL Query:

```
-- The MIN() function returns the smallest value of the selected column.
 -- The MAX() function returns the largest value of the selected column.
    -- 2ND HIGHEST :
SELECT  first_name,last_name,points AS 'Store Points'
FROM Customers
WHERE points =
        (SELECT  MAX(points)
                                FROM customers
         WHERE points <  (SELECT MAX(points)
                        FROM customers));
-- 2ND LOWEST :
SELECT  first_name,last_name,points AS 'Store Points'
```

```
FROM Customers
WHERE points =
        (SELECT  MIN(points)
         FROM customers
                                WHERE points >   (SELECT MIN(points)
                                        FROM customers));
```

Output:

- 2nd Highest

	first_name	last_name	Store Points
▶	Ram	Barhia	3073

- 2nd Lowest

	first_name	last_name	Store Points
▶	Aishwarya	Chakrabarti	457

21) WRITE A QUERY TO CREATE VIEW (TEMP.TABLE) OF PRODUCTS WITH ABOVE AVERAGE PRICE

SQL Query:

```
-- A view is a virtual table based on the result-set,
    -- A view contains rows and columns, just like a real table.

    CREATE VIEW prdcts_above_avg_price AS
    SELECT *
    FROM products
    WHERE unit_price > ( SELECT avg(unit_price)
```

FROM products) ;

Output:

product_id	name	quantity_in_stock	unit_price
2	Organic Hibiscus Tea, 16 Tea Bags	49	160
3	Antiperspirant Deodorant Stick	38	120
4	Cetaphil Facial Moisturizer	90	170
7	Cheddar cheese Pringles	98	115

22) SQL CHECK CONSTRAIN

SQL Query:

-- The CHECK constraint is used to limit the value range that can be placed in a column.

```
    CREATE TABLE Persons (
ID int NOT NULL,
Name varchar(50) NOT NULL,
 Age int,
 CHECK (Age>=18) );


    INSERT INTO persons
    VALUES ( 01, 'Person1',16) ;
```

Output:

It is an ERROR as Value entered doesn't come under the constrain

Error Code: 3819. Check constraint 'persons_chk_1' is violated.

But, if Age = 19 is given table is created and values entered

ID	Name	Age
1	Person1	19