SQL Quiz

Duration: 1.5 Hours
Database: Customer Orders (CO)

Q1. Display all customer names and email addresses from the CUSTOMERS table.

```
1   SELECT FULL_NAME, EMAIL_ADDRESS
2   FROM CO.CUSTOMERS;
3
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.008 seconds

	FULL_NAME	EMAIL_ADDRESS
1	Tammy Bryant	tammy.bryant@inter
2	Roy White	roy.white@internalm
3	Gary Jenkins	gary.jenkins@interna
4	Victor Morris	victor.morris@intern
5	Beverly Hughes	beverly.hughes@inte
6	Evelyn Torres	evelyn.torres@intern
7	Carl Lee	carl.lee@internalmail
8	Douglas Flores	douglas.flores@inter

Q2. List all products whose UNIT_PRICE is greater than 30. Show PRODUCT_NAME and UNIT_PRICE.

```
1   SELECT PRODUCT_NAME, UNIT_PRICE
2   FROM CO.PRODUCTS
3   WHERE UNIT_PRICE > 30;
4
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.005 seconds

	PRODUCT_NAME	UNIT_PRICE
1	Boy's Trousers (Blue)	34.06
2	Boy's Trousers (Black	39.32
3	Boy's Sweater (Green	44.17
4	Boy's Trousers (White	43.71
5	Girl's Shorts (Red)	38.28
6	Women's Skirt (Red)	30.69
7	Women's Socks (Grey	39.89
8	Girl's Shorts (Green)	38.34

Q3. Find all orders with ORDER_STATUS = 'COMPLETE'. Display ORDER_ID, CUSTOMER_ID, and ORDER_TMS.

```
SELECT ORDER_ID, CUSTOMER_ID, ORDER_TMS
FROM CO.ORDERS
WHERE ORDER_STATUS = 'COMPLETE';

```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.012 seconds

	ORDER_ID	CUSTOMER_ID	ORDER_TMS
1	792	271	2021-09-13T04:00:0
2	793	189	2021-09-13T13:35:
3	794	326	2021-09-13T14:43:0
4	795	33	2021-09-13T21:54:
5	796	124	2021-09-14T03:49:0
6	797	71	2021-09-14T12:34:
7	798	263	2021-09-14T20:57:
8	799	332	2021-09-15T02:14:

Q4. Show all stores located in a specific city (use any city name). Display STORE_NAME and PHYSICAL_ADDRESS.

```
SELECT STORE_NAME, PHYSICAL_ADDRESS
FROM CO.STORES
WHERE STORE_NAME = 'New York City';

```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.004 seconds

	STORE_NAME	PHYSICAL_ADDRES
1	New York City	205 Lexington Ave 7

Q5. Find all products whose PRODUCT_NAME contains the word 'Jeans. Show PRODUCT_ID and PRODUCT_NAME.

```
1  SELECT PRODUCT_ID, PRODUCT_NAME
2  FROM CO.PRODUCTS
3  WHERE PRODUCT_NAME LIKE '%Jeans%';
4
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.006 seconds

	PRODUCT_ID	PRODUCT_NAME
	37	Boy's Jeans (Blue)
	42	Boy's Jeans (Black)
	9	Women's Jeans (Brov
	25	Girl's Jeans (Grey)
	34	Women's Jeans (Red
	45	Men's Jeans (Grey)

Q6. List all orders placed in 2022. Display ORDER_ID, ORDER_TMS, and ORDER_STATUS

```
1  SELECT ORDER_ID, ORDER_TMS, ORDER_STATUS
2  FROM CO.ORDERS
3  WHERE EXTRACT(YEAR FROM ORDER_TMS) = 2022;
4
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.008 seconds

	ORDER_ID	ORDER_TMS	ORDER_STATUS
1	1392	2022-01-01T02:41:	COMPLETE
2	1393	2022-01-01T07:14:5	COMPLETE
3	1394	2022-01-01T15:29:4	COMPLETE
4	1395	2022-01-01T18:51:5	COMPLETE
5	1396	2022-01-01T20:05:4	COMPLETE
6	1397	2022-01-01T21:01:5	COMPLETE
7	1398	2022-01-01T23:53:0	COMPLETE
8	1399	2022-01-02T02:57:0	COMPLETE

Q7. Calculate the total number of orders placed by each customer. Show CUSTOMER_ID and total count.

```
SELECT CUSTOMER_ID, COUNT(*) AS TOTAL_ORDERS
FROM CO.ORDERS
GROUP BY CUSTOMER_ID;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.006 seconds

	CUSTOMER_ID	TOTAL_ORDERS
1	1	5
2	2	2
3	3	10
4	4	7
5	5	3
6	6	6
7	7	5
8	8	8

Q8. Find the average UNIT_PRICE of all products.

```
SELECT AVG(UNIT_PRICE) AS AVERAGE_PRICE
FROM CO.PRODUCTS;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.003 seconds

	AVERAGE_PRICE
1	6.19326086956522

Q9. Count how many products are available in each store. Display STORE_ID and product count from INVENTORY table.

```
SELECT STORE_ID, COUNT(PRODUCT_ID) AS PRODUCT_COUNT
FROM CO.INVENTORY
GROUP BY STORE_ID;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.004 seconds

	STORE_ID	PRODUCT_COUNT
1	1	46
2	2	23
3	3	22
4	4	40
5	5	23
6	6	1
7	7	25
8	8	23

Q10. Display ORDER_ID, ORDER_TMS, and CUSTOMER (FULL_NAME) for all orders. (Join ORDERS and CUSTOMERS)

```
SELECT O.ORDER_ID, O.ORDER_TMS, C.FULL_NAME
FROM CO.ORDERS O
JOIN CO.CUSTOMERS C
ON O.CUSTOMER_ID = C.CUSTOMER_ID;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.011 seconds

	ORDER_ID	ORDER_TMS	FULL_NAME
1	792	2021-09-13T04:00:(	Dorsey Arking
2	793	2021-09-13T13:35:'	Norman Lobregat
3	794	2021-09-13T14:43:(	Carmella Avalos
4	795	2021-09-13T21:54:'	Carolyn Wood
5	796	2021-09-14T03:49:(	Pete Chevis
6	797	2021-09-14T12:34:4	Eduardo Flignia
7	798	2021-09-14T20:57:'	Jerrell Kereluk

Q11. Show PRODUCT_NAME and its corresponding STORE_NAME where inventory exists. (Join PRODUCTS, INVENTORY, and STORES)

```
SELECT P.PRODUCT_NAME, S.STORE_NAME
FROM CO.PRODUCTS P
JOIN CO.INVENTORY I ON P.PRODUCT_ID = I.PRODUCT_ID
JOIN CO.STORES S ON I.STORE_ID = S.STORE_ID;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.017 seconds

	PRODUCT_NAME	STORE_NAME
1	Boy's Shirt (White)	Online
2	Women's Shirt (Gree	Online
3	Boy's Sweater (Green	Online
4	Boy's Trousers (Whit	Online
5	Girl's Shorts (Red)	Online
6	Boy's Socks (Grey)	Online
7	Boy's Socks (Black)	Online

Q12. List all ORDER_ITEMS with PRODUCT_NAME and UNIT_PRICE. (Join ORDER_ITEMS and PRODUCTS)

```
SELECT OI.ORDER_ID, P.PRODUCT_NAME, P.UNIT_PRICE
FROM CO.ORDER_ITEMS OI
JOIN CO.PRODUCTS P
ON OI.PRODUCT_ID = P.PRODUCT_ID;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.015 seconds

	ORDER_ID	PRODUCT_NAME	UNIT_PRICE
1	42	Women's Shirt (Gree	16.67
2	54	Women's Shirt (Gree	16.67
3	56	Women's Shirt (Gree	16.67
4	74	Women's Shirt (Gree	16.67
5	87	Women's Shirt (Gree	16.67
6	118	Women's Shirt (Gree	16.67
7	119	Women's Shirt (	16.67

Q13. Display SHIPMENT_ID, CUSTOMER (FULL_NAME), and DELIVERY_ADDRESS. (Join SHIPMENTS and CUSTOMERS)

```
SELECT S.SHIPMENT_ID, C.FULL_NAME, S.DELIVERY_ADDRESS
FROM CO.SHIPMENTS S
JOIN CO.CUSTOMERS C
ON S.CUSTOMER_ID = C.CUSTOMER_ID;

```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.015 seconds

	SHIPMENT_ID	FULL_NAME	DELIVERY_ADDRES
1	628	Douglas Flores	Boston, MA 02116 U
2	629	Kenny Campobasso	Milanville, PA 18443
3	630	Adam Miller	East Bernard, TX 774
4	631	Adam Miller	East Bernard, TX 774
5	632	Andrew Schieferstein	Fort Lauderdale, FL 3
6	633	Andrew Schieferstein	Fort Lauderdale, FL 3
7	634	Jonathan Coleman	Bowdoin, ME 04287

Q14. Find all products whose UNIT_PRICE is greater than the average UNIT_PRICE of all products. Display PRODUCT_NAME and UNIT_PRICE.

```
SELECT PRODUCT_NAME, UNIT_PRICE
FROM CO.PRODUCTS
WHERE UNIT_PRICE > (SELECT AVG(UNIT_PRICE) FROM CO.PRODUCTS);

```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.005 seconds

	PRODUCT_NAME	UNIT_PRICE
1	Women's Trousers (B	29.51
2	Boy's Trousers (Blue)	34.06
3	Boy's Trousers (Black	39.32
4	Boy's Shirt (White)	29.55
5	Boy's Sweater (Green	44.17
6	Boy's Trousers (White	43.71
7	Girl's Shorts (Red)	38.28

Q15. Find the top 3 customers who have placed the most orders. Display FULL_NAME and total number of orders.

```
SELECT C.FULL_NAME, COUNT(O.ORDER_ID) AS TOTAL_ORDERS
FROM CO.CUSTOMERS C
JOIN CO.ORDERS O
ON C.CUSTOMER_ID = O.CUSTOMER_ID
GROUP BY C.FULL_NAME
ORDER BY TOTAL_ORDERS DESC
FETCH FIRST 3 ROWS ONLY;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.01 seconds

	FULL_NAME	TOTAL_ORDERS
1	Kristina Livshits	11
2	Shamira Jones	11
3	Gary Jenkins	10

Instructions:

1. Write all queries in proper SQL syntax
2. Use appropriate table aliases
3. Test your queries before submission

Good Luck! 🎯