SQL Interview Practice Guide (30 Questions with Answers)

Q1. List all orders with customer name and order amount.

```
SELECT order_id, customer_name, amount FROM orders;
```

Q2. Show names of all delivery agents in Mumbai.

```
SELECT agent_name FROM deliveryagent WHERE city = 'Mumbai';
```

Q3. Find total number of restaurants in Delhi.

```
SELECT COUNT(*) AS total_restaurants FROM restaurant WHERE city =
'Delhi';
```

Q4. Show all unique restaurant cities (no duplicates).

```
SELECT DISTINCT city FROM restaurant;
```

Q5. Find the total sales amount handled by each delivery agent.

```
SELECT da.agent_name, SUM(o.amount) AS total_sales FROM orders o JOIN
deliveryagent da ON o.agent_id = da.agent_id GROUP BY da.agent_name;
```

Q6. Show orders where amount is greater than the average order amount.

```
SELECT * FROM orders WHERE amount > (SELECT AVG(amount) FROM orders);
```

Q7. Find the restaurant with the maximum number of orders.

```
SELECT r.restaurant_name, COUNT(*) AS total_orders FROM orders o JOIN
restaurant r ON o.restaurant_id = r.restaurant_id GROUP BY
r.restaurant_name ORDER BY total_orders DESC LIMIT 1;
```

Q8. Find delivery agents who delivered orders from more than one restaurant.

```
SELECT da.agent_name, COUNT(DISTINCT o.restaurant_id) AS
restaurants_handled FROM orders o JOIN deliveryagent da ON o.agent_id =
da.agent_id GROUP BY da.agent_name HAVING COUNT(DISTINCT o.restaurant_id)
> 1;
```

Q9. For each city, find the total sales done by agents in that city.

```
SELECT da.city, SUM(o.amount) AS total_sales FROM orders o JOIN
deliveryagent da ON o.agent_id = da.agent_id GROUP BY da.city;
```

Q10. Find the top restaurant per agent (highest sales).

```
SELECT da.agent_name, (SELECT r.restaurant_name FROM orders o2 JOIN
restaurant r ON o2.restaurant_id = r.restaurant_id WHERE o2.agent_id =
da.agent_id GROUP BY r.restaurant_name ORDER BY SUM(o2.amount) DESC LIMIT
```

```
1) AS top_restaurant FROM deliveryagent da;
```

Q11. Rank delivery agents by total sales.

```
SELECT da.agent_name, SUM(o.amount) AS total_sales, RANK() OVER (ORDER BY
SUM(o.amount) DESC) AS sales_rank FROM orders o JOIN deliveryagent da ON
o.agent_id = da.agent_id GROUP BY da.agent_name;
```

Q12. Find the top 3 restaurants by total sales.

```
SELECT r.restaurant_name, SUM(o.amount) AS total_sales FROM orders o JOIN
restaurant r ON o.restaurant_id = r.restaurant_id GROUP BY
r.restaurant_name ORDER BY total_sales DESC LIMIT 3;
```

Q13. Find agents who delivered orders from restaurants located in their own city.

```
SELECT DISTINCT da.agent_name FROM orders o JOIN deliveryagent da ON
o.agent_id = da.agent_id JOIN restaurant r ON o.restaurant_id =
r.restaurant_id WHERE da.city = r.city;
```

Q14. Find agents who have never delivered from 'KFC'.

```
SELECT da.agent_name FROM deliveryagent da WHERE NOT EXISTS (SELECT 1
FROM orders o JOIN restaurant r ON o.restaurant_id = r.restaurant_id
WHERE o.agent_id = da.agent_id AND r.restaurant_name = 'KFC');
```

Q15. Show the highest order amount per city (restaurant city).

```
SELECT r.city, MAX(o.amount) AS max_order FROM orders o JOIN restaurant r
ON o.restaurant_id = r.restaurant_id GROUP BY r.city;
```

Q16. Find the restaurant with the highest sales handled by each agent.

```
SELECT da.agent_name, (SELECT r.restaurant_name FROM orders o2 JOIN
restaurant r ON o2.restaurant_id = r.restaurant_id WHERE o2.agent_id =
da.agent_id GROUP BY r.restaurant_name ORDER BY SUM(o2.amount) DESC LIMIT
1) AS top_restaurant FROM deliveryagent da;
```

Q17. Find agents whose total sales are above the overall average sales of all agents.

```
SELECT da.agent_name, SUM(o.amount) AS total_sales FROM orders o JOIN
deliveryagent da ON o.agent_id = da.agent_id GROUP BY da.agent_name
HAVING SUM(o.amount) > (SELECT AVG(agent_sales) FROM (SELECT SUM(amount)
AS agent_sales FROM orders GROUP BY agent_id) t);
```

Q18. Find the customer(s) who placed the highest single order.

```
SELECT customer_name, amount FROM orders WHERE amount = (SELECT
MAX(amount) FROM orders);
```

Q19. Show daily agent ranking by number of deliveries.

```
SELECT o.order_date, da.agent_name, COUNT(o.order_id) AS
total_deliveries, RANK() OVER (PARTITION BY o.order_date ORDER BY
COUNT(o.order_id) DESC) AS daily_rank FROM orders o JOIN deliveryagent da
```

```
ON o.agent_id = da.agent_id GROUP BY o.order_date, da.agent_name;
```

Q20. Find the restaurant(s) that have never received an order.

```
SELECT r.restaurant_name FROM restaurant r WHERE NOT EXISTS (SELECT 1
FROM orders o WHERE o.restaurant_id = r.restaurant_id);
```

Q21. Find agents who delivered orders on at least 3 different days.

```
SELECT da.agent_name, COUNT(DISTINCT o.order_date) AS days_worked FROM
orders o JOIN deliveryagent da ON o.agent_id = da.agent_id GROUP BY
da.agent_name HAVING COUNT(DISTINCT o.order_date) >= 3;
```

Q22. Find the city that generated the maximum sales overall.

```
SELECT r.city, SUM(o.amount) AS total_sales FROM orders o JOIN restaurant
r ON o.restaurant_id = r.restaurant_id GROUP BY r.city ORDER BY
total_sales DESC LIMIT 1;
```

Q23. Show the second highest order amount (without using LIMIT).

```
SELECT MAX(amount) AS second_highest FROM orders WHERE amount < (SELECT
MAX(amount) FROM orders);
```

Q24. Find all orders where the agent delivered in a different city from their own city.

```
SELECT o.order_id, da.agent_name, da.city AS agent_city, r.city AS
restaurant_city FROM orders o JOIN deliveryagent da ON o.agent_id =
da.agent_id JOIN restaurant r ON o.restaurant_id = r.restaurant_id WHERE
da.city <> r.city;
```

Q25. Find the top customer (by total amount spent).

```
SELECT customer_name, SUM(amount) AS total_spent FROM orders GROUP BY
customer_name ORDER BY total_spent DESC LIMIT 1;
```

Q26. Find restaurants that earned more than the average earnings of all restaurants.

```
SELECT r.restaurant_name, SUM(o.amount) AS total_sales FROM orders o JOIN
restaurant r ON o.restaurant_id = r.restaurant_id GROUP BY
r.restaurant_name HAVING SUM(o.amount) > (SELECT AVG(total_amount) FROM
(SELECT SUM(amount) AS total_amount FROM orders GROUP BY restaurant_id)
t);
```

Q27. Show the running total of sales by order_date.

```
SELECT order_date, SUM(amount) AS daily_sales, SUM(SUM(amount)) OVER
(ORDER BY order_date) AS running_total FROM orders GROUP BY order_date
ORDER BY order_date;
```

Q28. Find the restaurant with the highest single-day sales.

```
SELECT r.restaurant_name, o.order_date, SUM(o.amount) AS day_sales FROM
orders o JOIN restaurant r ON o.restaurant_id = r.restaurant_id GROUP BY
r.restaurant_name, o.order_date ORDER BY day_sales DESC LIMIT 1;
```

Q29. Find agents who delivered all orders for a given restaurant ('Dominos').

```
SELECT da.agent_name FROM deliveryagent da WHERE NOT EXISTS (SELECT 1
FROM orders o JOIN restaurant r ON o.restaurant_id = r.restaurant_id
WHERE r.restaurant_name = 'Dominos' AND o.agent_id <> da.agent_id);
```

Q30. Find agents who delivered orders for at least one restaurant in every city.

```
SELECT da.agent_name FROM deliveryagent da WHERE NOT EXISTS (SELECT city
FROM restaurant r EXCEPT SELECT DISTINCT r2.city FROM orders o JOIN
restaurant r2 ON o.restaurant_id = r2.restaurant_id WHERE o.agent_id =
da.agent_id);
```