

SQL Questions for Interview Preparation

Basic Questions - Difference Between Delete and Truncate ?
Difference Between Where and Having Clause ?
Difference Between Union & Union ALL
Difference Between Primary Key and Foreign Key ?
Rank , Dense Rank & Row Number ?
Why do we use Case When ?
How to Find Duplicates from a Table ?
Find Second Highest Salary?
Cumulative Sum using Window Functions ?
Cumulative sum using Self Join?
Date Functions Videos ?
Dense Rank Questions . Example Top 5 Employees via Salary in each department ?
Cartesian Product Questions on Inner Join

SQL Questions:

1. **Customer Segmentation Based on Premium: Question:** Write a query to segment customers into different categories based on their total premium amount. Categorize customers as 'High Value', 'Medium Value', and 'Low Value' based on premium ranges (e.g., > 10,000 for High, 5,000-10,000 for Medium, and < 5,000 for Low).
 - o **Explanation:** This question tests your ability to use conditional logic (CASE statement) and aggregate functions.

Example Solution:

```
SELECT customer_id,  
       SUM(premium_amount) AS total_premium,  
       CASE  
         WHEN SUM(premium_amount) > 10000 THEN 'High Value'  
         WHEN SUM(premium_amount) BETWEEN 5000 AND 10000 THEN 'Medium Value'  
         ELSE 'Low Value'  
       END AS customer_segment  
FROM policies  
GROUP BY customer_id;
```

2. **Calculating Policy Renewal Rate: Question:** How would you calculate the policy renewal rate for each customer? Assume the `policies` table contains a column `is_renewed` with values 'Yes' or 'No'.
 - o **Explanation:** This assesses your ability to calculate rates using aggregate functions and conditions.

Example Solution:

```
SELECT customer_id,  
       COUNT(CASE WHEN is_renewed = 'Yes' THEN 1 END) * 100.0 / COUNT(*) AS  
       renewal_rate  
FROM policies  
GROUP BY customer_id;
```

3. **Identifying Cross-Sell Opportunities: Question:** Write a query to identify customers who have purchased a car insurance policy but have not purchased a health insurance policy.
 - **Explanation:** This question evaluates your skills in filtering data and using **JOIN** operations.

Example Solution:

```
SELECT c.customer_id
FROM customers c
LEFT JOIN policies p_car ON c.customer_id = p_car.customer_id AND p_car.policy_type = 'Car Insurance'
LEFT JOIN policies p_health ON c.customer_id = p_health.customer_id AND p_health.policy_type = 'Health Insurance'
WHERE p_car.policy_id IS NOT NULL AND p_health.policy_id IS NULL;
```

4. **Tracking Customer Retention: Question:** How would you write a query to calculate the number of customers who renewed their policy within the last year?
 - **Explanation:** This question tests your ability to work with date functions and filtering data based on date ranges.

Example Solution:

```
SELECT COUNT(DISTINCT customer_id) AS retained_customers
FROM policies
WHERE is_renewed = 'Yes' AND renewal_date >= DATEADD(YEAR, -1, GETDATE());
```

5. **Finding the Most Popular Insurance Product: Question:** Write a query to determine the most popular insurance product based on the number of policies sold.
 - **Explanation:** This question examines your understanding of **GROUP BY** and **ORDER BY** clauses.

Example Solution:

```
sql
Copy code
SELECT policy_type, COUNT(*) AS policy_count
FROM policies
GROUP BY policy_type
ORDER BY policy_count DESC
LIMIT 1;
```

Theory Questions:

1. **Understanding Key Business Metrics: Question:** What are the key metrics a Data Analyst should focus on in the insurance domain, particularly in a company like Policy Bazaar?
 - **Explanation:** You should discuss metrics such as customer acquisition cost, policy renewal rate, average premium per customer, customer lifetime value (CLTV), and claim settlement ratio. The interviewer is assessing your domain knowledge and how you apply data analysis to business goals.
2. **Data Quality in Insurance: Question:** How do you ensure data quality when working with large datasets in the insurance industry?

- **Explanation:** You should discuss data validation techniques, handling missing or inconsistent data, maintaining data integrity, and the importance of accurate and complete data for analysis. Mention specific tools or processes you've used to maintain data quality.
- 3. **Predictive Analytics in Insurance: Question:** How would you use predictive analytics to help a company like Policy Bazaar identify potential customers who are likely to purchase a new policy?
 - **Explanation:** Discuss how you would use historical data, feature engineering, and machine learning models to predict customer behavior. Explain the importance of segmentation, customer profiling, and the types of models (e.g., logistic regression, decision trees) that could be used.
- 4. **Regulatory Compliance: Question:** What are the key regulatory considerations a Data Analyst should be aware of when handling insurance data?
 - **Explanation:** Highlight your understanding of data privacy regulations such as GDPR, IRDAI guidelines, and how these regulations impact data collection, storage, and analysis in the insurance industry. Discuss the importance of ensuring data security and compliance in your work.
- 5. **Use of Dashboards for Decision Making: Question:** How would you design a dashboard for the senior management of Policy Bazaar to track the performance of their various insurance products?
 - **Explanation:** Describe the key performance indicators (KPIs) you would include, such as sales trends, customer acquisition and retention rates, claims ratios, and profitability. Explain the importance of real-time data, user-friendly visualizations, and how you would ensure the dashboard provides actionable insights.

These questions are designed to assess both your technical SQL skills and your understanding of the insurance industry, which are crucial for a Data Analyst working at a company like Policy Bazaar.

Table A:

id

1

2

3

2

Table B:

id

2

2

3

4

Result of the INNER JOIN:

When performing an **INNER JOIN** on **id**:

sql

Copy code

```
SELECT A.id, B.id
FROM A
INNER JOIN B
ON A.id = B.id;
```

The result of this `INNER JOIN` will be:

A.id	B.id
2	2
2	2
2	2
2	2
3	3

Explanation:

- The value `1` from `A` does not appear in `B`, so it is excluded.
- The value `2` in `A` is matched with every occurrence of `2` in `B`. Since `2` appears twice in both `A` and `B`, the join results in four combinations of `2`.
- The value `3` in both tables has a single match, so it appears once in the result.
- The value `4` in `B` does not appear in `A`, so it is excluded.

This result reflects all possible matches between `A` and `B` based on their shared `id` values.

4o

You said:

i want all left , right , inner and full in same code snippet

ChatGPT said:

ChatGPT

Sure! Below is an example of how to perform `LEFT JOIN`, `RIGHT JOIN`, `INNER JOIN`, and `FULL OUTER JOIN` on the same tables `A` and `B`, using SQL.

Tables Definition:

Table A:

id

1

2

3

2

Table B:

id

2

2

3

4

SQL Code Snippet:

sql

Copy code

-- INNER JOIN: Returns only matching rows from both tables

```
SELECT A.id AS A_id, B.id AS B_id
```

```
FROM A
```

```
INNER JOIN B ON A.id = B.id;
```

-- LEFT JOIN: Returns all rows from A, and the matching rows from B

-- Non-matching rows from B will have NULL values

```
SELECT A.id AS A_id, B.id AS B_id
```

```
FROM A
```

```
LEFT JOIN B ON A.id = B.id;
```

-- RIGHT JOIN: Returns all rows from B, and the matching rows from A

-- Non-matching rows from A will have NULL values

```
SELECT A.id AS A_id, B.id AS B_id
```

```
FROM A
```

```
RIGHT JOIN B ON A.id = B.id;
```

-- FULL OUTER JOIN: Returns all rows when there is a match in either table

-- Non-matching rows will have NULL values for the columns of the non-matching table

```
SELECT A.id AS A_id, B.id AS B_id
```

```
FROM A
```

```
FULL OUTER JOIN B ON A.id = B.id;
```

Results Explanation:

1. **INNER JOIN:**
 - **Output:**

A_id	B_id
2	2
2	2
2	2
2	2
3	3

Explanation: Only matching rows from both tables are returned.

2. **LEFT JOIN:**

Output:

A_id	B_id
1	NULL
2	2
2	2
2	2
2	2
3	3

Explanation: All rows from **A** are returned. Non-matching rows from **B** are represented with **NULL**.

3. **RIGHT JOIN:**

Output:

A_id	B_id
2	2
2	2
2	2
2	2
3	3
NULL	4

Explanation: All rows from **B** are returned. Non-matching rows from **A** are represented with **NULL**.

4. FULL OUTER JOIN:

Output:

A_id	B_id
1	NULL
2	2
2	2
2	2
2	2
3	3
NULL	4

○

Explanation: All rows from both **A** and **B** are returned. Non-matching rows are represented with **NULL** for the columns of the non-matching table.

This SQL snippet demonstrates how to apply different types of joins (**INNER**, **LEFT**, **RIGHT**, and **FULL OUTER**) to the same tables to see how each join operation affects the result set.