FREE PDF · 2026

# SQL Queries for ETL Testing Interview Questions

20 queries with answers: basic to experienced level, including SCD Type 2, incremental loads and reconciliation.
By Asim Noaman Lodhi · online version

**Practise under interview conditions**
ETL Testing Interview Questions & Answers – Practice Tests: 500+ questions including SQL, scenario, Python and automation.
Start the practice tests on Udemy →

## Basic SQL Queries for ETL Testing Interviews

### 1. Compare record counts between source and target

```
SELECT (SELECT COUNT(*) FROM src.orders) AS src_count,
       (SELECT COUNT(*) FROM dw.fact_orders) AS tgt_count;
```

**What they check:** that you then mention filters and rejects — equal counts are only expected when the mapping has no filter.

### 2. Find rows in source that are missing in target

```
SELECT order_id FROM src.orders
EXCEPT
SELECT order_id FROM dw.fact_orders;
```

Use `MINUS` in Oracle. Reverse the order to find extra rows in the target. **What they check:** you test both directions.

### 3. Find missing rows with a LEFT JOIN

```
SELECT s.order_id
FROM src.orders s
LEFT JOIN dw.fact_orders t ON t.order_id = s.order_id
WHERE t.order_id IS NULL;
```

**What they check:** that you know the anti-join pattern works across databases and when set operators aren't available.

### 4. Find duplicate records in the target

```
SELECT order_id, COUNT(*) AS cnt
FROM dw.fact_orders
GROUP BY order_id
HAVING COUNT(*) > 1;
```

### 5. Check for NULLs in mandatory columns

```
SELECT COUNT(*) AS null_rows
FROM dw.fact_orders
WHERE customer_key IS NULL OR order_date IS NULL OR amount IS NULL;
```

### 6. Compare a measure total between source and target

```
SELECT (SELECT SUM(amount) FROM src.orders WHERE status <> 'CANCELLED') AS src_total,
       (SELECT SUM(amount) FROM dw.fact_orders) AS tgt_total;
```

**What they check:** you apply the same business filter on both sides.

### 7. Validate a transformation rule

```
SELECT s.order_id, s.city, t.city
FROM src.orders s
JOIN dw.fact_orders t ON t.order_id = s.order_id
WHERE UPPER(TRIM(s.city)) <> t.city;
```

Apply the rule from the mapping document to the source and compare it with the target. Expect zero rows.

### 8. Check referential integrity (orphan records)

```
SELECT f.order_id, f.customer_key
FROM dw.fact_orders f
LEFT JOIN dw.dim_customer d ON d.customer_key = f.customer_key
WHERE d.customer_key IS NULL;
```

### 9. Validate data length to catch truncation

```
SELECT order_id, LENGTH(customer_name) AS len
FROM src.orders
WHERE LENGTH(customer_name) > 100;
```

If the target column is `VARCHAR(100)`, these source rows will be truncated or rejected.

### 10. Find values outside an allowed list

```
SELECT status, COUNT(*) AS cnt
FROM dw.fact_orders
WHERE status NOT IN ('NEW', 'SHIPPED', 'DELIVERED', 'RETURNED')
GROUP BY status;
```

## SQL Queries for ETL Testing Interview Questions for Experienced

### 11. Find duplicates and keep the latest row (window function)

```
SELECT *
FROM (
  SELECT o.*, ROW_NUMBER() OVER (PARTITION BY order_id ORDER BY updated_at DESC) AS rn
  FROM src.orders o
) x
WHERE rn > 1;
```

**What they check:** window functions, and that you know which duplicate the ETL should keep.

### 12. Full column-by-column comparison

```
SELECT order_id, customer_id, order_date, amount FROM src.orders
EXCEPT
SELECT order_id, customer_id, order_date, amount FROM dw.fact_orders;
```

Run it in both directions. Any row returned has at least one column that differs.

### 13. Reconcile totals by period

```
SELECT s.order_month, s.src_total, t.tgt_total, s.src_total - t.tgt_total AS diff
FROM (SELECT DATE_TRUNC('month', order_date) AS order_month, SUM(amount) AS src_total
      FROM src.orders GROUP BY 1) s
JOIN (SELECT DATE_TRUNC('month', order_date) AS order_month, SUM(amount) AS tgt_total
      FROM dw.fact_orders GROUP BY 1) t ON t.order_month = s.order_month
WHERE s.src_total <> t.tgt_total;
```

**What they check:** that you can narrow a mismatch down to a period instead of reporting one big difference.

### 14. SCD Type 2: more than one current row per business key

```
SELECT customer_id, COUNT(*) AS current_rows
FROM dw.dim_customer
WHERE current_flag = 'Y'
GROUP BY customer_id
HAVING COUNT(*) <> 1;
```

### 15. SCD Type 2: overlapping date ranges

```
SELECT a.customer_id, a.customer_key, b.customer_key
FROM dw.dim_customer a
JOIN dw.dim_customer b
  ON a.customer_id = b.customer_id
 AND a.customer_key <> b.customer_key
 AND a.start_date < b.end_date
 AND b.start_date < a.end_date;
```

Any row returned means two versions were valid at the same time.

### 16. Incremental load: rows changed since the last run

```
SELECT COUNT(*) AS expected_delta
FROM src.orders
WHERE updated_at > (SELECT MAX(last_extract_ts) FROM etl.load_audit WHERE job_name =
'orders');
```

Compare this with the number of rows the job reports it inserted or updated.

### 17. Fact rows pointing to the wrong SCD2 version

```
SELECT f.order_id
FROM dw.fact_orders f
JOIN dw.dim_customer d ON d.customer_key = f.customer_key
WHERE f.order_date < d.start_date OR f.order_date >= d.end_date;
```

**What they check:** that you understand facts must join to the dimension version valid on the transaction date.

### 18. Detect a sudden drop in daily volume

```
SELECT load_date, row_count, avg_7d
FROM (
  SELECT load_date, COUNT(*) AS row_count,
         AVG(COUNT(*)) OVER (ORDER BY load_date ROWS BETWEEN 7 PRECEDING AND 1 PRECEDING)
AS avg_7d
  FROM dw.fact_orders
  GROUP BY load_date
) x
WHERE row_count < 0.5 * avg_7d;
```

### 19. Find the Nth highest value (classic warm-up)

```
SELECT amount
FROM (SELECT amount, DENSE_RANK() OVER (ORDER BY amount DESC) AS rnk FROM dw.fact_orders) x
WHERE rnk = 3;
```

### 20. Reconcile source + rejects + filtered = target

```
SELECT (SELECT COUNT(*) FROM src.orders) AS src_count,
       (SELECT COUNT(*) FROM dw.fact_orders) AS tgt_count,
       (SELECT COUNT(*) FROM etl.orders_reject) AS reject_count,
       (SELECT COUNT(*) FROM src.orders WHERE status = 'CANCELLED') AS filtered_count;
```

The load reconciles when target + rejects + filtered equals source. **What they check:** that you account for every row, not just compare two numbers.


> **Learn ETL testing hands-on**
> ETL Testing: From Beginners to Advanced – 79 lectures on ETL and data warehouse testing, data quality, Python, Power Query and AI tools.
> See the course on Udemy →

More free material: ETL Testing Interview Kit (PDF) · Start here