

**-- Show columns & small sample from each table**

```
SELECT 'CUSTOMERS SAMPLE' as table_name, * FROM stg_customers_raw LIMIT 5;
SELECT 'ORDERS SAMPLE' as table_name, * FROM stg_orders_raw LIMIT 5;
SELECT 'SHIPPING SAMPLE' as table_name, * FROM stg_shipping_raw LIMIT 5;
```

**-- Row counts across all tables**

```
SELECT 'orders' AS table_name, COUNT(*) as record_count FROM stg_orders_raw
UNION ALL
SELECT 'shipping', COUNT(*) FROM stg_shipping_raw
UNION ALL
SELECT 'customers', COUNT(*) FROM stg_customers_raw;
```

**-- Orders referencing customers not present in customers table**

```
WITH cust AS (
  SELECT DISTINCT CAST(trim(Customer_ID) AS BIGINT) AS customer_id
  FROM stg_customers_raw
  WHERE COALESCE(trim(Customer_ID), "") <> ""
),
ord_cust AS (
  SELECT DISTINCT CAST(trim(Customer_ID) AS BIGINT) AS customer_id
  FROM stg_orders_raw
  WHERE COALESCE(trim(Customer_ID), "") <> ""
)
SELECT o.customer_id
FROM ord_cust o
LEFT JOIN cust c ON o.customer_id = c.customer_id
WHERE c.customer_id IS NULL
LIMIT 100;
```

**-- Shipping referencing customers not present in customers table**

```
WITH cust AS (
  SELECT DISTINCT CAST(trim(Customer_ID) AS BIGINT) AS customer_id
  FROM stg_customers_raw
),
ship_cust AS (
  SELECT DISTINCT CAST(trim(Customer_ID) AS BIGINT) AS customer_id
  FROM stg_shipping_raw
)
SELECT s.customer_id
FROM ship_cust s
LEFT JOIN cust c ON s.customer_id = c.customer_id
WHERE c.customer_id IS NULL
LIMIT 100;
```

**-- Check for NULL values in key fields**

```
SELECT
    'customers' as table_name,
    SUM(CASE WHEN customer_id IS NULL OR customer_id = '' THEN 1 ELSE 0 END) as
null_customer_ids,
    SUM(CASE WHEN first IS NULL OR first = '' THEN 1 ELSE 0 END) as null_first_names,
    SUM(CASE WHEN last IS NULL OR last = '' THEN 1 ELSE 0 END) as null_last_names,
    SUM(CASE WHEN age IS NULL OR age = '' THEN 1 ELSE 0 END) as null_ages,
    SUM(CASE WHEN country IS NULL OR country = '' THEN 1 ELSE 0 END) as
null_countries
FROM stg_customers_raw
```

UNION ALL

```
SELECT
    'orders' as table_name,
    SUM(CASE WHEN order_id IS NULL OR order_id = '' THEN 1 ELSE 0 END) as
null_order_ids,
    SUM(CASE WHEN item IS NULL OR item = '' THEN 1 ELSE 0 END) as null_items,
    SUM(CASE WHEN amount IS NULL OR amount = '' THEN 1 ELSE 0 END) as
null_amounts,
    SUM(CASE WHEN customer_id IS NULL OR customer_id = '' THEN 1 ELSE 0 END) as
null_customer_ids,
    0 as null_countries
FROM stg_orders_raw
```

UNION ALL

```
SELECT
    'shipping' as table_name,
    SUM(CASE WHEN shipping_id IS NULL THEN 1 ELSE 0 END) as null_shipping_ids,
    SUM(CASE WHEN status IS NULL OR status = '' THEN 1 ELSE 0 END) as null_statuses,
    SUM(CASE WHEN customer_id IS NULL THEN 1 ELSE 0 END) as null_customer_ids,
    0 as null_ages,
    0 as null_countries
FROM stg_shipping_raw;
```

**-- Check for duplicate records**

```
SELECT
    'customers' as table_name,
    COUNT(*) as total_records,
    COUNT(DISTINCT customer_id) as unique_customer_ids,
    COUNT(*) - COUNT(DISTINCT customer_id) as duplicate_customer_ids
FROM stg_customers_raw
```

UNION ALL

```

SELECT
  'orders' as table_name,
  COUNT(*) as total_records,
  COUNT(DISTINCT order_id) as unique_order_ids,
  COUNT(*) - COUNT(DISTINCT order_id) as duplicate_order_ids
FROM stg_orders_raw

```

```

UNION ALL

```

```

SELECT
  'shipping' as table_name,
  COUNT(*) as total_records,
  COUNT(DISTINCT shipping_id) as unique_shipping_ids,
  COUNT(*) - COUNT(DISTINCT shipping_id) as duplicate_shipping_ids
FROM stg_shipping_raw;

```

#### **-- Validate data type conversions**

```

SELECT
  'age_conversion_test' as test_name,
  COUNT(*) as total_records,
  SUM(CASE WHEN REGEXP_LIKE(age, '[0-9]+$') THEN 1 ELSE 0 END) as valid_ages,
  SUM(CASE WHEN NOT REGEXP_LIKE(age, '[0-9]+$') AND age IS NOT NULL AND
age != " THEN 1 ELSE 0 END) as invalid_ages
FROM stg_customers_raw

```

```

UNION ALL

```

```

SELECT
  'amount_conversion_test' as test_name,
  COUNT(*) as total_records,
  SUM(CASE WHEN REGEXP_LIKE(amount, '[0-9]+\.[0-9]*$') THEN 1 ELSE 0 END) as
valid_amounts,
  SUM(CASE WHEN NOT REGEXP_LIKE(amount, '[0-9]+\.[0-9]*$') AND amount IS NOT
NULL AND amount != " THEN 1 ELSE 0 END) as invalid_amounts
FROM stg_orders_raw;

```

#### **-- Check for data consistency issues**

```

SELECT
  'customer_id_consistency' as check_type,
  COUNT(DISTINCT customer_id) as unique_customer_ids,
  COUNT(*) as total_records,
  COUNT(*) - COUNT(DISTINCT customer_id) as potential_duplicates
FROM stg_customers_raw
WHERE customer_id IS NOT NULL AND customer_id != "";

```

#### **-- Overall data summary**

```

SELECT

```

```
    'DATA_SUMMARY' as metric_type,  
    'Total Customers' as metric_name,  
    CAST(COUNT(*) AS STRING) as metric_value  
FROM stg_customers_raw  
WHERE customer_id IS NOT NULL AND customer_id != "
```

UNION ALL

```
SELECT  
    'DATA_SUMMARY',  
    'Total Orders',  
    CAST(COUNT(*) AS STRING)  
FROM stg_orders_raw  
WHERE order_id IS NOT NULL AND order_id != "
```

UNION ALL

```
SELECT  
    'DATA_SUMMARY',  
    'Total Shipping Records',  
    CAST(COUNT(*) AS STRING)  
FROM stg_shipping_raw  
WHERE shipping_id IS NOT NULL
```

UNION ALL

```
SELECT  
    'DATA_SUMMARY',  
    'Total Revenue',  
    CAST(ROUND(SUM(CAST(amount AS DECIMAL(15,2))), 2) AS STRING)  
FROM stg_orders_raw  
WHERE amount IS NOT NULL AND amount != " AND REGEXP_LIKE(amount,  
'^[0-9]+\.[0-9]*$');
```