

# E04 - Star Schema (SQL)

E04, Business Intelligence

**Oemer Furkan Coban**

*27.11.2025 22:06*

Business Intelligence  
Winter Term 2025/2026

University of Oldenburg

**Accepted:** 18 (95%)

**Total:** 19

**1.1** Find out how many customers this data warehouse has?

accepted

```
1 | SELECT COUNT(*) AS count
2 | FROM dim_customer;
```

**1.2** Find out how many products the company has currently?

accepted

```
1 | SELECT COUNT(*) AS count
2 | FROM dim_product
3 | WHERE end_date IS NULL;
```

**1.3** Find out how many facts the fact table contains?

accepted

```
1 | SELECT COUNT(*) FROM fact_internet_sales;
```

**1.4** What is the most sold product?

accepted

```
1 | SELECT fact_internet_sales.product_key, dim_product.english_product_name,
2 | SUM(fact_internet_sales.order_quantity) AS _cnt
3 | FROM fact_internet_sales
4 | JOIN dim_product ON fact_internet_sales.product_key = dim_product.product_key
5 | GROUP BY fact_internet_sales.product_key, dim_product.english_product_name
6 | ORDER BY _cnt DESC
7 | LIMIT 1;
```

**1.5** Which products are the top 3 bestsellers? Bestsellers means "most sold product".

accepted

```
1 | SELECT fact_internet_sales.product_key, dim_product.english_product_name,
2 | SUM(fact_internet_sales.order_quantity) AS _cnt
3 | FROM fact_internet_sales
4 | JOIN dim_product ON fact_internet_sales.product_key = dim_product.product_key
5 | GROUP BY fact_internet_sales.product_key, dim_product.english_product_name
6 | ORDER BY _cnt DESC
7 | LIMIT 3;
```

**1.6** On which day of the week the most products are usually sold?

accepted

```
1 | SELECT dim_date.english_day_name_of_week,
2 | SUM(fact_internet_sales.order_quantity) AS _cnt
3 | FROM fact_internet_sales
4 | JOIN dim_date ON fact_internet_sales.order_date_key = dim_date.date_key
5 | GROUP BY dim_date.english_day_name_of_week
6 | ORDER BY _cnt DESC;
```

**1.7** On which day of the week the most products are usually shipped?

accepted

```
1 | SELECT dim_date.english_day_name_of_week,
2 | SUM(fact_internet_sales.order_quantity) AS _cnt
3 | FROM fact_internet_sales
4 | JOIN dim_date ON fact_internet_sales.ship_date_key = dim_date.date_key
5 | GROUP BY dim_date.english_day_name_of_week
6 | ORDER BY _cnt DESC;
```

**1.8** Are there any delays in the process (see due\_date and ship\_date)?

accepted

```
1 | SELECT COUNT(*) AS before_delay
2 | FROM fact_internet_sales
3 | WHERE ship_date_key > due_date_key;
```

**1.9** Calculate revenue for each product (use all products). Use ROUND. Sort by revenue and show top 25.

accepted

```
1 | SELECT fact_internet_sales.product_key, dim_product.english_product_name,
2 | SUM(fact_internet_sales.order_quantity) AS _cnt,
3 | ROUND(SUM(fact_internet_sales.sales_amount)::numeric, 2) AS revenue,
4 | ROUND(SUM(fact_internet_sales.sales_amount)::numeric, 2) AS sales_amount
5 | FROM fact_internet_sales
6 | JOIN dim_product ON fact_internet_sales.product_key = dim_product.product_key
7 | GROUP BY fact_internet_sales.product_key, dim_product.english_product_name
8 | ORDER BY revenue DESC
9 | LIMIT 25;
```

**1.10** Which product make the highest revenue?

accepted

```
1 | SELECT fact_internet_sales.product_key, dim_product.english_product_name,
2 | SUM(fact_internet_sales.order_quantity) AS _cnt,
3 | SUM(fact_internet_sales.sales_amount) AS revenue,
4 | SUM(fact_internet_sales.sales_amount) AS sales_amount
5 | FROM fact_internet_sales
6 | JOIN dim_product ON fact_internet_sales.product_key = dim_product.product_key
7 | GROUP BY fact_internet_sales.product_key, dim_product.english_product_name
8 | ORDER BY revenue DESC
9 | LIMIT 1;
```

**2.1** Who is the best customer (the one that buys at most)? Show customer\_key, first\_name + last\_name, number of sales and revenue.

accepted

```
1 | SELECT fact_internet_sales.customer_key,
2 | CONCAT(dim_customer.first_name, ' ', dim_customer.last_name) AS full_name,
3 | COUNT(*) AS number_of_sales,
4 | SUM(fact_internet_sales.sales_amount) AS revenue
5 | FROM fact_internet_sales
6 | JOIN dim_customer ON fact_internet_sales.customer_key = dim_customer.customer_key
```

```

7 | GROUP BY fact_internet_sales.customer_key, dim_customer.first_name, dim_customer.last_name
8 | ORDER BY revenue DESC
9 | LIMIT 1;

```

**2.2** Identify the top 10 customers who could be categorized as "not so good ones" based on their average revenue per order. A "not a good customer" is defined as having a less favorable ratio between the total items bought and revenue. The analysis should only include customers who have placed more than 20 orders. Apply the ROUND function to fields with decimal values in the final results.

accepted

```

1 | SELECT fact_internet_sales.customer_key, CONCAT(dim_customer.first_name, ' ', 
2 |           dim_customer.last_name) AS customer_name,
3 | COUNT(*) AS _cnt,
4 | ROUND(CAST(SUM(fact_internet_sales.sales_amount) AS numeric), 0) AS revenue,
5 | ROUND(CAST(SUM(fact_internet_sales.sales_amount) / SUM(fact_internet_sales.order_quantity) AS
6 |           numeric), 0) AS ratio
7 | FROM fact_internet_sales
8 | JOIN dim_customer ON fact_internet_sales.customer_key = dim_customer.customer_key
9 | GROUP BY fact_internet_sales.customer_key, dim_customer.first_name, dim_customer.last_name
10 | HAVING COUNT(*) > 20
11 | ORDER BY ratio ASC
12 | LIMIT 10;

```

**2.3** Compare drop or growth of the sales of each products for the a time period between 01.01.2012 and 31.12.2012. Show top 25 entries, sort by date and select proper columns (product\_key, order\_date, ship\_date, due\_date, customer\_key, unit\_price, english\_product\_name).

accepted

```

1 | SELECT fact_internet_sales.product_key,
2 | fact_internet_sales.order_date,
3 | fact_internet_sales.ship_date,
4 | fact_internet_sales.due_date,
5 | fact_internet_sales.customer_key,
6 | fact_internet_sales.unit_price,
7 | dim_product.english_product_name
8 | FROM fact_internet_sales
9 | JOIN dim_product ON fact_internet_sales.product_key = dim_product.product_key
10 | WHERE CAST(fact_internet_sales.order_date AS date)
11 |       BETWEEN '2012-01-01' AND '2012-12-31'
12 | ORDER BY fact_internet_sales.order_date DESC
13 | LIMIT 25;

```

**2.4** Are there any customers, who uses multiple locations / stores to buy their products (are enough data provided to answer this question at all)?

accepted

```

1 | SELECT dim_customer.customer_key,
2 | COUNT(DISTINCT dim_customer.geography_key) AS _cnt
3 | FROM dim_customer
4 | GROUP BY dim_customer.customer_key
5 | HAVING COUNT(DISTINCT dim_customer.geography_key) > 1;

```

**2.5** Are there any product(s), which were sold a lot (e.g., significant ↓ -6) in a particular day/week and then experienced a significant sales drop? Use "ship\_date\_key" to make sure, that a particular sale has been processed. Hint: you may want to use CTE and LAG to in your SQL query and sort by "product\_key", "fiscal\_year" and "week\_number\_of\_year".

wrong answer

```
1 WITH weekly_sales AS (
2     SELECT fact_internet_sales.product_key,
3         dim_date.fiscal_year,
4         dim_date.week_number_of_year,
5         COUNT(*) AS _cnt
6     FROM fact_internet_sales
7     JOIN dim_date ON fact_internet_sales.ship_date_key = dim_date.date_key
8     GROUP BY fact_internet_sales.product_key, dim_date.fiscal_year,
9             dim_date.week_number_of_year
10),
11 weekly_with_lag AS (
12     SELECT product_key, fiscal_year, week_number_of_year, _cnt,
13     LAG(_cnt) OVER (
14         PARTITION BY product_key
15         ORDER BY fiscal_year, week_number_of_year
16     ) AS previous_cnt
17     FROM weekly_sales
18 )
19 SELECT product_key, fiscal_year, week_number_of_year, _cnt, previous_cnt,
20 (_cnt - previous_cnt) AS diff
21 FROM weekly_with_lag
22 WHERE previous_cnt IS NOT NULL AND (_cnt - previous_cnt) < -6
23 ORDER BY product_key, fiscal_year, week_number_of_year;
```

**3.1** What is the average amount of products / goods / units sold each day over all years? Apply ROUND function to fields with decimal values.

accepted

```
1 WITH daily_sales AS (
2     SELECT fact_internet_sales.order_date,
3         SUM(fact_internet_sales.order_quantity) AS daily_cnt
4     FROM fact_internet_sales
5     GROUP BY fact_internet_sales.order_date
6 )
7 SELECT ROUND(AVG(daily_cnt), 0) AS avg_day
8 FROM daily_sales;
```

**3.2** What is the average amount of products / goods / units sold each week over all years? Apply ROUND function to fields with decimal values.

accepted

```
1 WITH weekly_sales AS (
2     SELECT dim_date.fiscal_year,
3         dim_date.week_number_of_year,
4         SUM(fact_internet_sales.order_quantity) AS weekly_cnt
5     FROM fact_internet_sales
6     JOIN dim_date ON fact_internet_sales.order_date_key = dim_date.date_key
7     GROUP BY dim_date.fiscal_year, dim_date.week_number_of_year
8 )
9 SELECT ROUND(AVG(weekly_cnt), 0) AS avg_week
10 FROM weekly_sales;
```

**3.3** What is the average amount of products / goods / units sold each month over all years? Apply ROUND function to fields with decimal values.

accepted

```

1 WITH monthly_sales AS (
2     SELECT dim_date.calendar_year,
3         dim_date.month_number_of_year,
4         SUM(fact_internet_sales.order_quantity) AS monthly_cnt
5     FROM fact_internet_sales
6     JOIN dim_date ON fact_internet_sales.order_date_key = dim_date.date_key
7     GROUP BY dim_date.calendar_year, dim_date.month_number_of_year
8 )
9 SELECT ROUND(AVG(monthly_cnt), 0) AS avg_month
10 FROM monthly_sales;

```

**3.4** Show sales distribution by months where month name written in French. Order by month\_number\_of\_year and apply ROUND function to fields with decimal values.

accepted

```

1 SELECT dim_date.french_month_name,
2     SUM(fact_internet_sales.order_quantity) AS _cnt,
3     dim_date.month_number_of_year,
4     ROUND(CAST(SUM(fact_internet_sales.extended_amount) AS numeric), 0) AS _extended_amount,
5     ROUND(CAST(SUM(fact_internet_sales.sales_amount) AS numeric), 0) AS _sales_amount
6     FROM fact_internet_sales
7     JOIN dim_date ON fact_internet_sales.order_date_key = dim_date.date_key
8     GROUP BY dim_date.french_month_name, dim_date.month_number_of_year
9     ORDER BY dim_date.month_number_of_year;

```