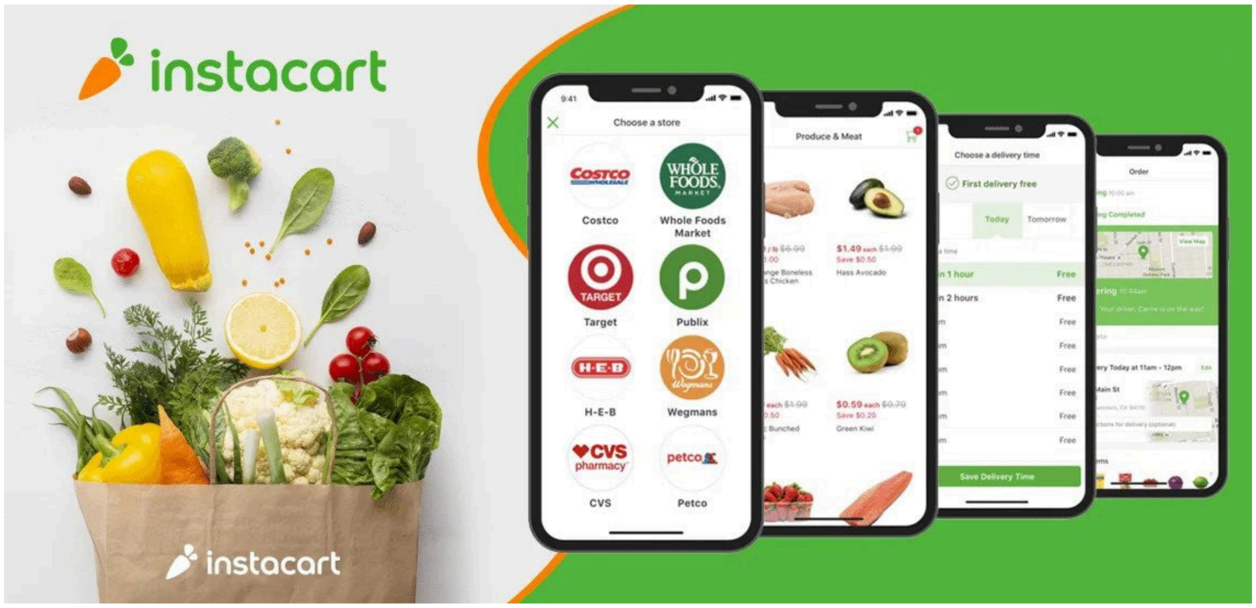
Instacart SQL Case Study

Case Study Background: About Instacart

Instacart is a grocery delivery and pickup service. Users can select items from local grocery stores through the Instacart app or website, and then either have them delivered to their doorstep by a personal shopper or prepared for pickup at the store.



For our non-North American friends working on this case study, Instacart is similar to India's Blinkit, Swiggy Instamart, or Dunzo app. In Europe, the comparable app is Getir and Gorrilas. In Latin America, Rappi serves a similar use case.

Case Study Background: The Data Analysis Task

You're a data analyst at Instacart. Aside from the discounted groceries, you also get the benefit of solving interesting data problems.

In your 1-1 call this morning, your manager tells you that leadership wants to analyze the Instacart market data over time, to understand how the business is changing or staying the same.

Unfortunately, data engineering found some logging errors in the pipeline, and there are currently no date fields in the market data tables 🙃.

Your manager has a call with engineering tomorrow to work on fixing this so the team can track changes more closely in the future. But you're already mid-way through Q3 and the data pipeline can't be refreshed again until Q4. So for now, you're stuck with the data you have.

The Task: find a way to understand how Instacart's business changed over time…without using explicit dates!

Case Study Background: Instacart Grocery Orders Data

Here are the schemas for all 5 tables in the Instacart market data. You'll decide which ones are relevant and how to best use them throughout this case study.

ic_order_products_curr**: this table specifies which products were purchased in each Instacart order.**

Column Name	**Type**
order_id	**integer**
product_id	**integer**
add_to_cart_order	**integer**
reordered	**integer boolean (1 or 0)**

Notes:

- **The 'reordered' field indicates that the customer has a previous order that contains the product.**
- **Some orders will have no reordered items**
- **None of these fields are unique to this table, but the combination of** order_id **and** product_id **is unique!**

ic_order_products_prior: this table contains *previous* order contents for all customers. This data was collected in Q2, verus ic_order_products_currwhich is data from the current quarter (Q3).

Column Name	Type
order_id	integer
product_id	integer
add_to_cart_order	integer
reordered	integer boolean (1 or 0)

Note: the table ic_order_products_prior has the same exact schema as ic_order_products_curr. You'll likely want to compare these two tables!

ic_products: info about each item in the Instacart product catalog

Column Name	Type
product_id	integer
product_name	string
aisle_id	integer
department_id	integer

ic_departments: info about each department

Column Name	Type

department_id	**integer**
department	**string**

ic_aisles**: info about each aisle in a grocery store**

Column Name	**Type**
aisle_id	**integer**
aisle	**string**

Our Solution

Here's how we'd approach this ambiguous data case study. Feel free to follow this approach, or adapt it to derive your own insights!

Our Solution: High-Level Overview

1. Identify the **two specific tables** you should focus on to understand broad trends over time.
2. Consider a phenomenon in the data that may have changed over time. Choose something practical that you can highlight to your manager.
3. Define in plain ol' English how you plan to investigate the data to solve the data analysis task.
4. Express that approach in SQL.
5. If your observed phenomenon **has** changed over time, develop 2 or more hypotheses to explain potential causes of that change. If your observed phenomenon **has not** changed over time, develop 2 or more hypotheses to explain why this phenomenon has remained stagnant.
6. Consider other relevant factors aside from just the Instacart data, such as food trends, seasonality, supply chain, etc.

BONUS: Based on your hypotheses, write a recommendation to leadership explaining how they should either:

- Support this phenomenon if it’s **helpful** to Instacart business, or

- Combat this phenomenon if it’s **harmful** to Instacart business.

If you want to check your work against our solution, scroll down to check the answer key below.

Remember, a successful case study simply means you developed a coherent process with a data-driven conclusion and defended your method! It doesn’t have to be the same as ours; it just needs to be similarly rigorous!

Our Solution: Analyzing Prior vs. Current Products.

The two tables we want to focus on are ic_order_products_prior and ic_order_products_curr.

As a data analyst, one surprising event you might want to investigate using SQL could be a sudden and significant change in the reordering behavior of certain products in the current orders compared to their behavior in the prior orders.

Specifically, you could look for products that were not frequently reordered in the past (based on data from ic_order_products_prior) but are now being reordered more frequently in the current orders (from ic_order_products_curr).

We can formulate the following query to investigate:

```
SELECT
   prod.product_id,
   prod.product_name,
   prod.aisle_id,
   prod.department_id,
   dept.department,
   aisles.aisle,
   SUM(op_prior.reordered) AS prior_reorders,
   SUM(op_curr.reordered) AS current_reorders
 FROM
   ic_products AS prod
 JOIN
   Ic_order_products_prior AS op_prior
   ON prod.product_id = op_prior.product_id
 JOIN
   ic_order_products_curr AS op_curr
   ON prod.product_id = op_curr.product_id
 JOIN
   ic_departments AS dept
   ON prod.department_id = dept.department_id
 JOIN
```

```
    ic_aisles AS aisles
    ON prod.aisle_id = aisles.aisle_id
GROUP BY
    prod.product_id,
    prod.product_name,
    prod.aisle_id,
    prod.department_id,
    dept.department,
    aisles.aisle
HAVING
    SUM(op_prior.reordered) < 10
    AND SUM(op_curr.reordered) >= 10
ORDER BY
    current_reorders DESC;
```

product_id	product_name	aisle_id	department_id	department	aisle	prior_reorders	current_reorders
8174	Organic Navel Orange	24	4	produce	fresh fruits	0	30
7948	Organic Unsalted Butter	36	16	dairy eggs	butter	0	24
8193	Russet Potato	83	4	produce	fresh vegetables	0	24

The first few rows of results should look like this:

Our Solution: Analyzing Changes in Reorders By Department, Aisle

Looking at this granular view won't give us a broad picture of changes over time, so let's do some summarizing. We can wrap the above query in a CTE (Common Table Expression) called increased_orders and do some quick aggregation. We can aggregate by department, which shows that the majority of products with increased reorders fall under Produce.

```
SELECT department,
COUNT(*) AS num_products
FROM increased_reorders
GROUP BY department
ORDER BY num_products DESC;
```

We can aggregate by aisle to see if that data tells the same story:

```
SELECT aisle,
COUNT(*) AS num_products
FROM increased_reorders
GROUP BY aisle
ORDER BY num_products DESC;
```

According to the results, most of these products come from aisles that are related to produce (fresh vegetables, fresh fruit, and packaged produce).

Our Solution: Hypothesis Behind Re-order Change

Now it’s time for the actionable part of our work: figuring out what this data means for the company, and what we'll do because of the data.

We’ll start by developing some hypotheses to explain potential reasons for the increase in reorders of produce items.

First of all, we mentioned that it’s currently **Q3**, which means we’re somewhere between June and August in this scenario. Fruits and vegetables might become more popular in the summer months due to their freshness and nutritional value. So we’ll call hypothesis #1 seasonality.

It’s possible that limited-time discounts, bundle deals, or loyalty programs could drive higher reorders. We should check in with the marketing department and see if there were any deals or discounts related to produce recently. We’ll call hypothesis #2 deals.

We could also take a more skeptical approach to things and see if these products were even available before, or if a recent increase in supply chain activity has allowed for more reorders. When we see reorders go from 0 to 30 for the Organic Navel Orange, we have to ask if the conditions around those orders have changed at all. We’ll call hypothesis #3 availability.

We could keep going, but with the 3 hypotheses below, we’ve covered step 5 of the strategy framework:

1. Seasonality: Produce is more popular in the summer due to its freshness and nutritional value.
2. Deals: Discounts, bundle deals, or loyalty programs around produce could drive higher reorders.
3. Availability: Recent shifts in supply chain and availability may have influenced consumers’ ability to buy produce.

Our Solution: Our Business Recommendation

Here is the bonus recommendation informed by our hypotheses:

After analyzing our current order data compared to previous order data, we discovered that the reorder rates for produce have significantly increased across both departments and aisles. Increased produce sales are beneficial for the company, so leadership should capitalize on customers’ higher propensity to reorder produce during the summer months.

If not already implemented, marketing should consider promoting bundle deals on produce to incentivize new buyers, who may then become repeat customers for those products. Additionally, team members working with suppliers and grocers should ensure the consistent availability of popular produce items, including Organic Navel Oranges, Russet Potatoes, and Cantaloupes, in order to maintain high reorder rates.