# SQL-03 | Joins and Aggregation

Lecture Queries

Question: Let's say we wanted to list each product name along with its product category name.

Question: Let's say we wanted to list each product name along with its product category name.

```
SELECT * FROM

product

LEFT JOIN product_category

ON product_product_category_id = product_category.product_category_id

With table aliasing:
```

```
p.product_id,
p.product_name,
pc.product_category_id,
pc.product_category_name
FROM product AS p

LEFT JOIN product_category_id = pc.product_category_id
```

Question: Get all the Customers who have not purchased anything from the market yet.

Question: Get all the Customers who have not purchased anything from the market yet.

SELECT c.\* # select columns from customer table only
FROM customer AS c

LEFT JOIN customer\_purchases AS cp

ON c.customer\_id = cp.customer\_id

WHERE cp.customer\_id IS NULL

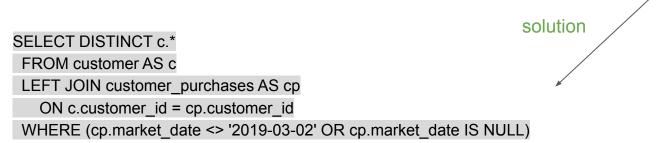
Question: Let's say we want to write a query that returns a list of all customers who did not make a purchase on March 2, 2019.

#### Question: Let's say we want to write a query that returns a list of all customers who did not make a purchase on March 2, 2019.

## SELECT c.\*, cp.market\_date FROM customer AS c LEFT JOIN customer\_purchases AS cp ON c.customer\_id = cp.customer\_id WHERE cp.market date <> '2019-03-02'

#### Two problems with the output:

- 1. Some rows/ customers are missing because the market date is NULL.
- 2. We are getting multiple rows for each customer which is not required.



Question: Let's say we want details about all farmer's market booths, as well as every vendor booth assignment for every market date along with the vendor details.

Question: Let's say we want details about all farmer's market booths, as well as every vendor booth assignment for every market date.

```
SELECT
b.booth_number,
b.booth_type,
vba.market_date,
v.vendor_id,
v.vendor_name,
v.vendor_type
FROM booth AS b
LEFT JOIN vendor_booth_assignments AS vba ON b.booth_number = vba.
booth_number
LEFT JOIN vendor AS v ON v.vendor_id = vba.vendor_id
ORDER BY b.booth_number, vba.market_date
```

Question: Get a list of customer IDs of customers who made purchases on each market date.

Question: Get a list of customer IDs of customers who made purchases on each market date.

# SELECT market\_date, customer\_id FROM farmers\_market.customer\_purchases GROUP BY market\_date, customer\_id ORDER BY market\_date, customer\_id

Question: Count the number of purchases each customer made per market date.

### Question: Count the number of purchases each customer made per market date.

```
SELECT

market_date,
customer_id,
COUNT(*) AS num_purchases
FROM farmers_market.customer_purchases
GROUP BY market_date, customer_id
ORDER BY market_date, customer_id
LIMIT 10
```

### Question: Calculate the total quantity that each customer bought per market date.

### Question: Calculate the total quantity that each customer bought per market date.

#### **SELECT**

market\_date,

customer\_id,

SUM(quantity) AS total\_qty\_purchased

FROM farmers\_market.customer\_purchases

GROUP BY market\_date, customer\_id

ORDER BY market\_date, customer\_id

Question: how many different kinds of products were purchased by each customer on each market date:

#### Question: how many different kinds of products were purchased by each customer on each market date:

```
SELECT

market_date,
customer_id,
COUNT(DISTINCT product_id) AS different_products_purchased
FROM farmers_market.customer_purchases
GROUP BY market_date, customer_id
ORDER BY market_date, customer_id
```

Question: Calculate the total price paid by customer\_id 3 per market\_date.

Question: Calculate the total price paid by customer\_id 3 per market\_date.

```
customer_id,
market_date,
SUM(quantity * cost_to_customer_per_qty) AS total_spent
FROM farmers_market.customer_purchases
WHERE
customer_id = 3
GROUP BY market_date
ORDER BY market_date
```

Question: Let's add some customer details and vendor details to these results.

Customer details are in the customer table and vendor details are in the vendor table.

Question: Let's add some customer details and vendor details to these results. Customer details are in the customer table and vendor details are in the vendor table.

```
SELECT
  c.customer first name,
  c.customer last name,
  cp.customer id,
  v.vendor id,
  v.vendor name,
  SUM(quantity * cost to customer per qty) AS total price
FROM customer AS c
LEFT JOIN customer purchases AS cp
  ON c.customer id = cp.customer id
LEFT JOIN vendor AS v
  ON cp.vendor id = v.vendor id
GROUP BY c.customer first name,
  c.customer last name,
  cp.customer id,
  v.vendor id,
  v.vendor name
```