### SQL-03 | JOINS

Lecture Queries

Question: Get the report of all the vendors who primarily sell fresh produce and who don't and mark it in front of their names.

Question: Get the report of all the vendors who primarily sell fresh produce and who don't and mark it in front of their names.

```
Vendor_id,
vendor_name,
vendor_type,
CASE
WHEN LOWER(vendor_type) LIKE '%fresh%'
THEN 'Fresh Produce'
ELSE 'Other'
END AS vendor_type_condensed
FROM farmers market.vendor
```

### Question: Put the total cost to customer purchases into bins of

- under \$5.00,
- \$5.00-\$9.99,
- \$10.00–\$19.99, or
- \$20.00 and over.

#### Question: Put the total cost to customer purchases into bins of

- under \$5.00,
- \$5.00-\$9.99,
- \$10.00–\$19.99, or
- \$20.00 and over.

```
SELECT
       market_date,
       customer id.
       vendor id,
       ROUND(quantity * cost_to_customer_per_qty, 2) AS price,
       CASE
         WHEN quantity * cost to customer per qty < 5.00
           THEN 'Under $5'
         WHEN quantity * cost to customer per gty < 10.00
           THEN '$5-$9.99'
         WHEN quantity * cost to customer per qty < 20.00
           THEN '$10-$19.99'
         WHEN quantity * cost_to_customer_per_qty >= 20.00
            THEN '$20 and Up'
         END AS price bin
     FROM farmers market.customer purchases
     LIMIT 10
```

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

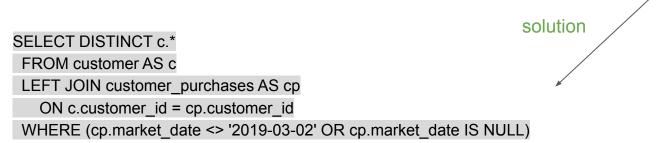
Question: Let's say we want to write a query that returns a list of all customers who did not make a purchase on March 2, 2019.

## 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 booths, as well as every vendor booth assignment 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
```