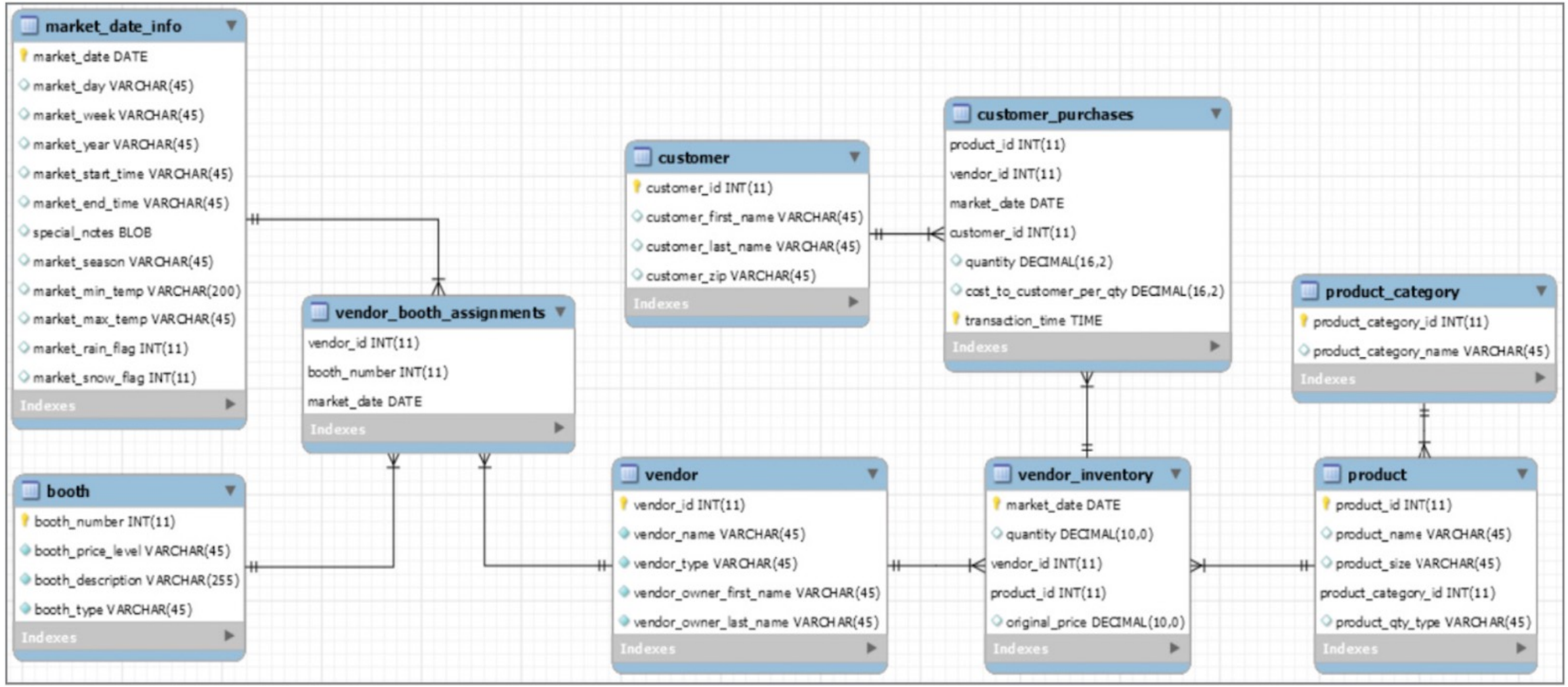
SQL-02 Filtering
Lecture 3


market_date_info
market_date DATE
market_day VARCHAR(45)
market_week VARCHAR(45)
market_year VARCHAR(45)
market_start_time VARCHAR(45)
market_end_time VARCHAR(45)
special_notes BLOB
market_season VARCHAR(45)
market_min_temp VARCHAR(200)
market_max_temp VARCHAR(45)
market_rain_flag INT(11)
market_snow_flag INT(11)
Indexes
vendor_booth_assignments
vendor_id INT(11)
booth_number INT(11)
market_date DATE
Indexes
customer
customer_id INT(11)
customer_first_name VARCHAR(45)
customer_last_name VARCHAR(45)
customer_zip VARCHAR(45)
Indexes
customer_purchases
product_id INT(11)
vendor_id INT(11)
market_date DATE
customer_id INT(11)
quantity DECIMAL(16,2)
cost_to_customer_per_qty DECIMAL(16,2)
transaction_time TIME
Indexes
product_category
product_category_id INT(11)
product_category_name VARCHAR(45)
Indexes
booth
booth_number INT(11)
booth_price_level VARCHAR(45)
booth_description VARCHAR(255)
booth_type VARCHAR(45)
Indexes
vendor
vendor_id INT(11)
vendor_name VARCHAR(45)
vendor_type VARCHAR(45)
vendor_owner_first_name VARCHAR(45)
vendor_owner_last_name VARCHAR(45)
Indexes
vendor_inventory
market_date DATE
quantity DECIMAL(10,0)
vendor_id INT(11)
product_id INT(11)
original_price DECIMAL(10,0)
Indexes
product
product_id INT(11)
product_name VARCHAR(45)
product_size VARCHAR(45)
product_category_id INT(11)
product_qty_type VARCHAR(45)
Indexes

Quiz

- We can assign alias names to only inline calculated columns?
  - True
  - False – correct
- Which of these are right? (N, M could be any number)

  1) Offset skips top ‘N’ rows
  2) Limit skips top ‘N’ rows
  3) Offset decides ‘M’ rows to display
  4) Limit decides ‘M’ rows to display
  - 1 and 2
  - 2 and 3
  - 1 and 4 - correct
  - 3 and 4
- You have to mandatorily specify ‘desc’ when sorting the data in descending order?
  - True -Correct
  - False
- Pick the wrong query from this
  - Select * from FM.customer_tbl order by customer_name desc
  - Select * from FM.customer_tbl limit 10 order by customer_name - correct
  - Select customer_name from FM.customer_tbl
  - Select * from FM.customer_tbl order by customer_name limit 5 offset 10
- When writing a Query, OFFSET is a optional clause or a mandatory clause?
  - Optional - correct
  - Mandatory
- What is the output of ceil(12.01), round(14.63, 1), floor(6.56)?
  - 12, 14.6, 6
  - 13, 14.6, 6 - correct
  - 13, 14.7, 6
  - 12, 14.6, 7
- What is the output of substr(customer_name, -4, 2) on a customer column having the following names “Jessy”, ‘Bryan”,”Walker”?
  - es, ry, lk - correct
  - ss, yan, lk
  - Jes, Br, Wa
  - sy, an, er

Extract all the product names that are part of product category 1

```
SELECT
product_id, product_name, product_category_id
FROM farmers_market.product
WHERE product_category_id = 1
LIMIT 5
```

Print a report of everything the customer_id 4 has ever purchased at the market, sorted by date. Add total_amt column as well for each purchase.

```
SELECT
customer_id, market_date, quantity,
cost_to_customer_per_qty,
ROUND(quantity * cost_to_customer_per_qty, 2) AS total_amt
FROM `farmers_market.customer_purchases`
WHERE customer_id = 4
ORDER BY market_date ASC
```

Get all the product info for products with id between 3 and 8 (not inclusive) and of products with id 10.

```
SELECT *
FROM `farmers_market.product`
WHERE (product_id > 3 AND product_id < 8) OR product_id =10
```

Find the booth assignments for vendor_id 7 for all dates between April 3, 2019 and May 16, 2019, including the 2 dates.

```
SELECT *
FROM `farmers_market.vendor_booth_assignments`
WHERE vendor_id = 7 AND
(market_date >= "2019-04-03" AND market_date <= "2019-05-18")
```

```
SELECT *
FROM `farmers_market.vendor_booth_assignments`
WHERE vendor_id = 7 AND
(market_date BETWEEN "2019-04-03" AND "2019-05-18")
```

Find the customer detail with the first name of “Carlos” or the last name of “Diaz,”:

```
SELECT
customer_id,
customer_first_name,
customer_last_name
FROM farmers_market.customer
WHERE
customer_last_name = 'Diaz'
OR customer_last_name = 'Edwards'
OR customer_last_name = 'Wilson'
```

Using IN

```
SELECT
customer_id,
customer_first_name,
customer_last_name
FROM farmers_market.customer
WHERE
customer_last_name IN ('Diaz' , 'Edwards', 'Wilson')
```

```
SELECT
customer_id,
customer_first_name,
customer_last_name
FROM farmers_market.customer
WHERE LOWER(customer_last_name) IN ('diaz' , 'edwards', 'Wilson')
```

You can also use

```
UPPER(customer_last_name) IN ('DIAZ' , 'EDWARDS', 'WILSON')
```

Question: You want to get data about a customer you knew as Jerry but you are not sure if they are listed as Jeremy or Jeremiah or Jerry. Get all customers whose name starts with “jer”.

```
SELECT *
FROM `farmers_market.customer`
WHERE lower(customer_first_name) LIKE "jer%"
```

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that starts with "a"
WHERE CustomerName LIKE '%a'	Finds any values that ends with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a__%'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that starts with "a" and ends with "o"

What if we want a customers with minimum 5 characters and first 3 chars as “jer”

```
SELECT *
FROM `farmers_market.customer`
WHERE lower(customer_first_name) LIKE "jer__%"
```

Find all of the products from the product table which don’t have sizes mentioned.

```
SELECT *
FROM farmers_market.product
WHERE
product_size IS NULL
OR TRIM(product_size) = “”
```

: 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,
quantity, cost_to_customer_per_qty,
CASE
        WHEN quantity * cost_to_customer_per_qty < 5 THEN "Under $5"
        WHEN quantity * cost_to_customer_per_qty BETWEEN 5 AND 9.99 THEN "$5 - $9.99"
        WHEN quantity * cost_to_customer_per_qty BETWEEN 10 AND 19.99 THEN "$10 - $19.99"
ELSE "Above $20"
END AS price_bin
FROM `farmers_market.customer_purchases`
```

Find out which vendors primarily sell fresh products and which don’t

```
SELECT
vendor_id,
vendor_name,
vendor_type,
CASE
        WHEN LOWER(vendor_type) LIKE '%fresh%'
THEN 'Fresh Produce'
ELSE 'Other'
        END AS category
FROM farmers_market.vendor
```

What if we want to add 1 for vendors who sell fresh products and 0 for those who don’t?

```
SELECT vendor_id, vendor_name, vendor_type,
CASE
        WHEN LOWER(vendor_type) LIKE "%fresh%" THEN 1
ELSE 0
END AS category
FROM `farmers_market.vendor`
```

We can also do the same using an IF statement -

```
SELECT *,
IF(LOWER(vendor_type) LIKE "%fresh%", "Fresh", "Not Fresh") AS type
FROM `farmers_market.vendor`
```

```
SELECT *,
IF(LOWER(vendor_type) LIKE "%fresh%", 1, 0) AS type FROM
`farmers_market.vendor`
```