Capitalise the full_name of the customers

First character of the first name and the last name should be in uppercase, all other characters should be in lowercase.

```
SELECT
 customer_id,
 customer_first_name,
 customer_last_name,
 CONCAT(UPPER(SUBSTR(customer_first_name, 1, 1)), LOWER(SUBSTR(customer_first_name, 2)), " ", UPPER(SUBSTR(customer_last_name, 1, 1)),
LOWER(SUBSTR(customer_last_name, 2))) AS capitalised_full_name
FROM `farmers_market.customer`;
```

```
SELECT
 customer_id,
 customer_first_name,
 customer_last_name,
 CONCAT(INITCAP(customer_first_name)," ", INITCAP(customer_last_name)) AS capitalised_full_name
FROM `farmers_market.customer`
```

Extract all the product names that belong to product category id 1.

WHERE clause

Syntax:

SELECT <cols>

FROM <tbl_name>

WHERE <conditional stmt>

ORDER BY

LIMIT

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

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 product with id 10

>3 and <8 → 4, 5, 6, 7, 10

1. Product_id > 3
2. Product_id < 8
3. Product_id = 10

Logical operators - AND / OR

```sql
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.

Whenever you have such range based questions, you can use the BETWEEN keyword.

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

Return a list of customers with the following last names: [Diaz, Edwards, Wilson]

IN keyword - whenever we have a list of values to filter from. Contains.

EDWards - edwards

WILSON - wilson

Diaz - diaz

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”.

LIKE - to work with partial strings

Wildcard characters :

1. % - stand-in for 0 or more characters.
2. _(underscore) - stand-in for one and only one character

LIKE

SELECT <>

FROM <>

WHERE fname LIKE “”

- “a%” - starts with “a”
- “%a” - ends with “a”
- “%or%” - that has “or” at any position
- “a_%” - any string that starts with “a” and is at least 2 chars long.

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