Report on Database Project

1. Introduction

The purpose of this project is to design and implement a database system for a bike store. The database manages customers, staffs, stores, orders, and products. The design is based on an ERD (Entity Relationship Diagram) that shows the relationship between tables in the **Sales** and **Production** modules.

Database Setup

1. Database Creation

First, we created a new database named **BikeStore**:

```
CREATE DATABASE BikeStore;

USE BikeStore;
```

---

2. Table Creation

We created tables for both **Sales** and **Production** modules.

Example: Customers Table

```
CREATE TABLE customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    phone VARCHAR(20),
    email VARCHAR(150),
    street VARCHAR(255),
    city VARCHAR(100),
    state VARCHAR(100),
    zip_code VARCHAR(20)
);
```

Example: Products Table

```
CREATE TABLE products (
```

```
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    product_name VARCHAR(255),
    brand_id INT,
    category_id INT,
    model_year YEAR,
    list_price DECIMAL(10,2),
    FOREIGN KEY (brand_id) REFERENCES brands(brand_id),
    FOREIGN KEY (category_id) REFERENCES categories(category_id)
);
```

We repeated this process for **all tables** (customers, staffs, stores, orders, order_items, categories, brands, products, stocks).

---

3. Relationships

After creating tables, we added **primary keys** to uniquely identify each record and **foreign keys** to connect tables.

- Customers are linked to Orders.
- Orders are linked to Order Items.
- Order Items are linked to Products.
- Products are linked to Brands and Categories.
- Stocks link Products with Stores.
- Staffs are linked to Stores and Orders.

Example:

```
ALTER TABLE orders
ADD FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
ADD FOREIGN KEY (store_id) REFERENCES stores(store_id),
ADD FOREIGN KEY (staff_id) REFERENCES staffs(staff_id);
```

Import Data into Tables

After creating the database and tables, the next step was to insert the data.
I used two different methods:

2.1 Import Data Using CSV Files

For most tables (such as categories, brands, products, customers, stores, orders, and order_items), I used **MySQL Workbench Table Data Import Wizard**.

Steps:

1. Right-click the table.
2. Select **Table Data Import Wizard**.
3. Choose the CSV file from the project folder.
4. Map the columns with the table structure.
5. Click **Finish** to insert the data.

This method successfully imported all data from CSV files into the respective tables.

2.2 Insert Staff Data Manually

For the staffs table, the import wizard did not work correctly.
Therefore, I inserted the staff data manually using an **INSERT statement**

This ensured that all staff records were inserted correctly into the database.

Queries_Report

1) Which bike is most expensive? What could be the motive behind pricing this bike at the high price?

```
SELECT product_name ,list_price
FROM products
where list_price = (SELECT max(list_price ) FROM
products);
```

```
5
6   SELECT product_name ,list_price
7   FROM products
8   where list_price = (SELECT max(list_price ) FROM products);
9
10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_name	list_price
Trek Domane SLR 9 Disc - 2018	11999.99

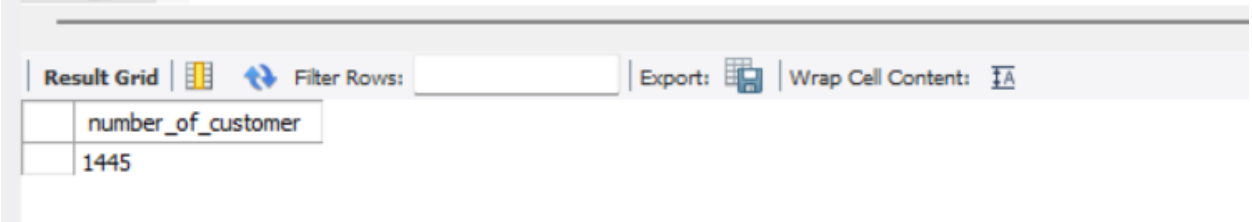
```
select br.brand_name , pro.model_year,pro.product_name ,pro.list_price

from products pro

join brands br

 on br.brand_id = pro.brand_id

 where pro.product_name = 'Trek Domane SLR 9 Disc - 2018';
```

brand_name	model_year	product_name	list_price
Trek	2018	Trek Domane SLR 9 Disc - 2018	11999.99

The most expensive bike is *Trek Domane SLR 9 Disc - 2018* (Brand: Trek, Year: 2018) with a price of 11,999.99. The high price can be justified by the strong Trek brand reputation, use of advanced technology (carbon frame, disc brakes), and the fact that it targets professional cyclists rather than casual customers.

2) How many total customers does BikeStore have? Would you consider people with order status 3 as customers substantiate your answer?

```
select count(customer_id) as number_of_customer from customers;
```

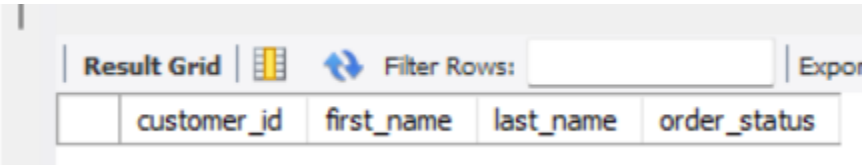


SELECT DISTINCT c.customer_id, c.first_name, c.last_name, o.order_status

FROM customers c

JOIN orders o ON c.customer_id = o.customer_id

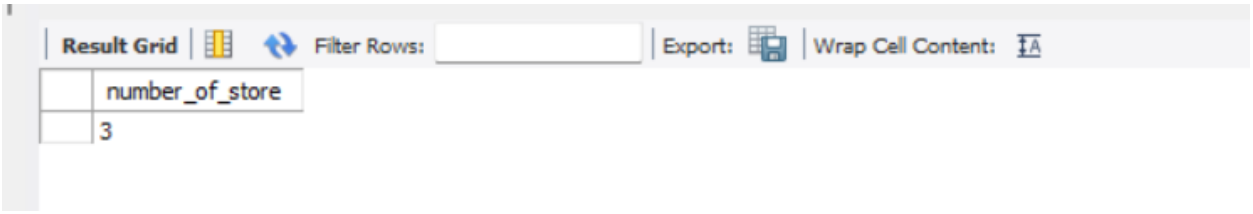
WHERE o.order_status = 3 ;



BikeStore has **1445 registered customers**.
When checking for order status = 3, there were no results. This suggests either that status 3 is not used in the current dataset or that there are no customers with orders in this status. Therefore, all 1445 customers are registered, but the meaning of status 3 should be clarified with the business context.

3) How many stores does BikeStore have?

select count(*) as number_of_store from stores;



3) What is the total price spent per order?
select order_id, quantity, list_price, discount,
(list_price *quantity*(1-discount)) as total_price
from order_items;

order_id	quantity	list_price	discount	total_price
1	1	599.99	0.20	479.9920
1	2	1799.99	0.07	3347.9814
1	2	1549.00	0.05	2943.1000
1	2	599.99	0.05	1139.9810
1	1	2899.99	0.20	2319.9920
2	1	599.99	0.07	557.9907
2	2	599.99	0.05	1139.9810
3	1	999.99	0.05	949.9905
3	1	599.99	0.05	569.9905
4	2	749.99	0.10	1349.9820
5	2	1549.00	0.05	2943.1000

Result 20 ×

4) What's the sales/revenue per store? Hint: Sales revenue = ([list_price] *[quantity]*(1-[discount]))

```
select order_id, quantity, list_price, discount,
(list_price *quantity*(1-discount)) as Sales_revenue
 from order_items;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

order_id	quantity	list_price	discount	Sales_revenue
1	1	599.99	0.20	479.9920
1	2	1799.99	0.07	3347.9814
1	2	1549.00	0.05	2943.1000
1	2	599.99	0.05	1139.9810
1	1	2899.99	0.20	2319.9920
2	1	599.99	0.07	557.9907
2	2	599.99	0.05	1139.9810
3	1	999.99	0.05	949.9905
3	1	599.99	0.05	569.9905
4	2	749.99	0.10	1349.9820
5	2	1549.00	0.05	2943.1000

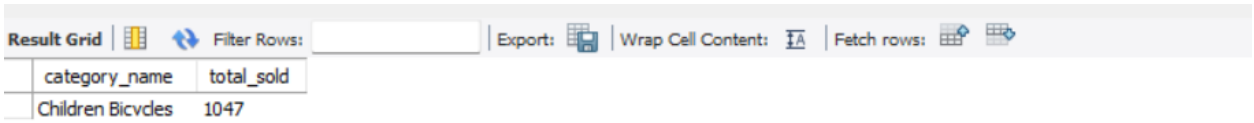
Result 21 ×

5) Which category is most sold?

```
select ca.category_name , sum( ord.quantity) as total_sold

from categories ca

join products pr on ca.category_id = pr.category_id

join order_items ord on ord.product_id = pr.product_id
```

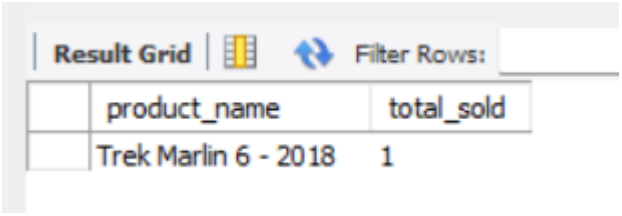
group by(pr.category_id)

limit 1;



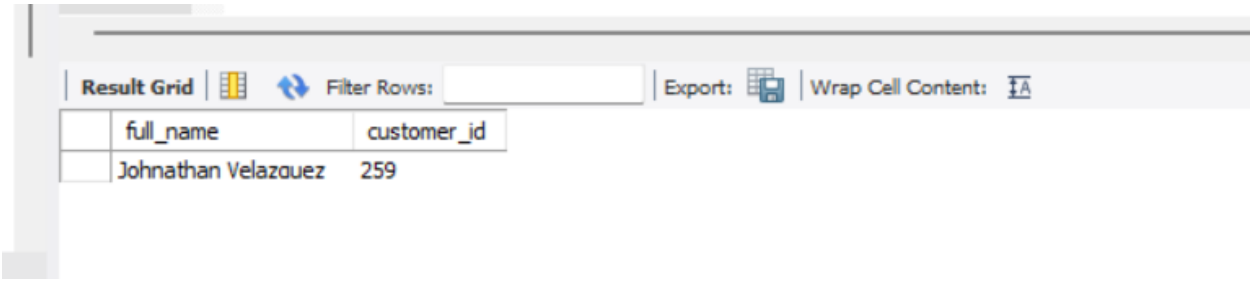
6) Which category rejected more orders?

```
SELECT
   p.product_name,
   SUM(oi.quantity) AS total_sold
FROM order_items oi
JOIN products p
   ON oi.product_id = p.product_id
GROUP BY p.product_name
ORDER BY total_sold ASC
LIMIT 1;
```



9) What's the full name of a customer with ID 259?

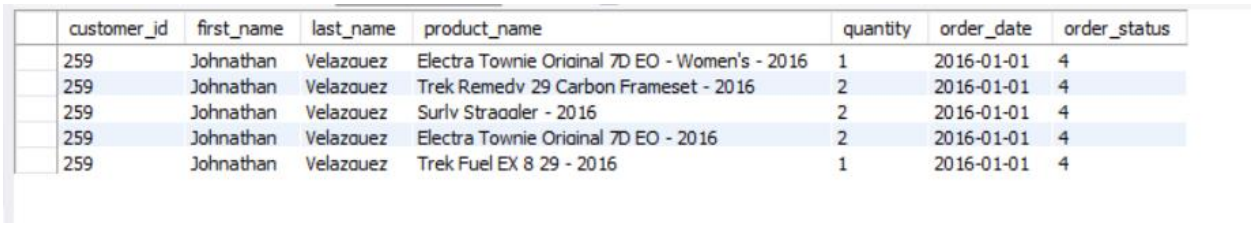
```
select concat(first_name,' ',last_name) as full_name ,customer_id

from customers

where customer_id = 259 ;
```



10) What did the customer on question 9 buy and when? What's the

status of this order?

```
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    p.product_name,
    oi.quantity,
    o.order_date,
    o.order_status
FROM customers c
JOIN orders o
    ON c.customer_id = o.customer_id
JOIN order_items oi
    ON o.order_id = oi.order_id
JOIN products p
    ON oi.product_id = p.product_id
WHERE c.customer_id = 259;
```



customer_id	first_name	last_name	product_name	quantity	order_date	order_status
259	Johnathan	Velazquez	Electra Townie Original 7D EQ - Women's - 2016	1	2016-01-01	4
259	Johnathan	Velazquez	Trek Remedy 29 Carbon Frameset - 2016	2	2016-01-01	4
259	Johnathan	Velazquez	Surly Straggler - 2016	2	2016-01-01	4
259	Johnathan	Velazquez	Electra Townie Original 7D EQ - 2016	2	2016-01-01	4
259	Johnathan	Velazquez	Trek Fuel EX 8 29 - 2016	1	2016-01-01	4