Bicycle Part Management System
Milestone 3

CNIT 37200

Justin Allange

Jaden Soroka

Evan Spadafora

Jordan Wozniak

Project Background

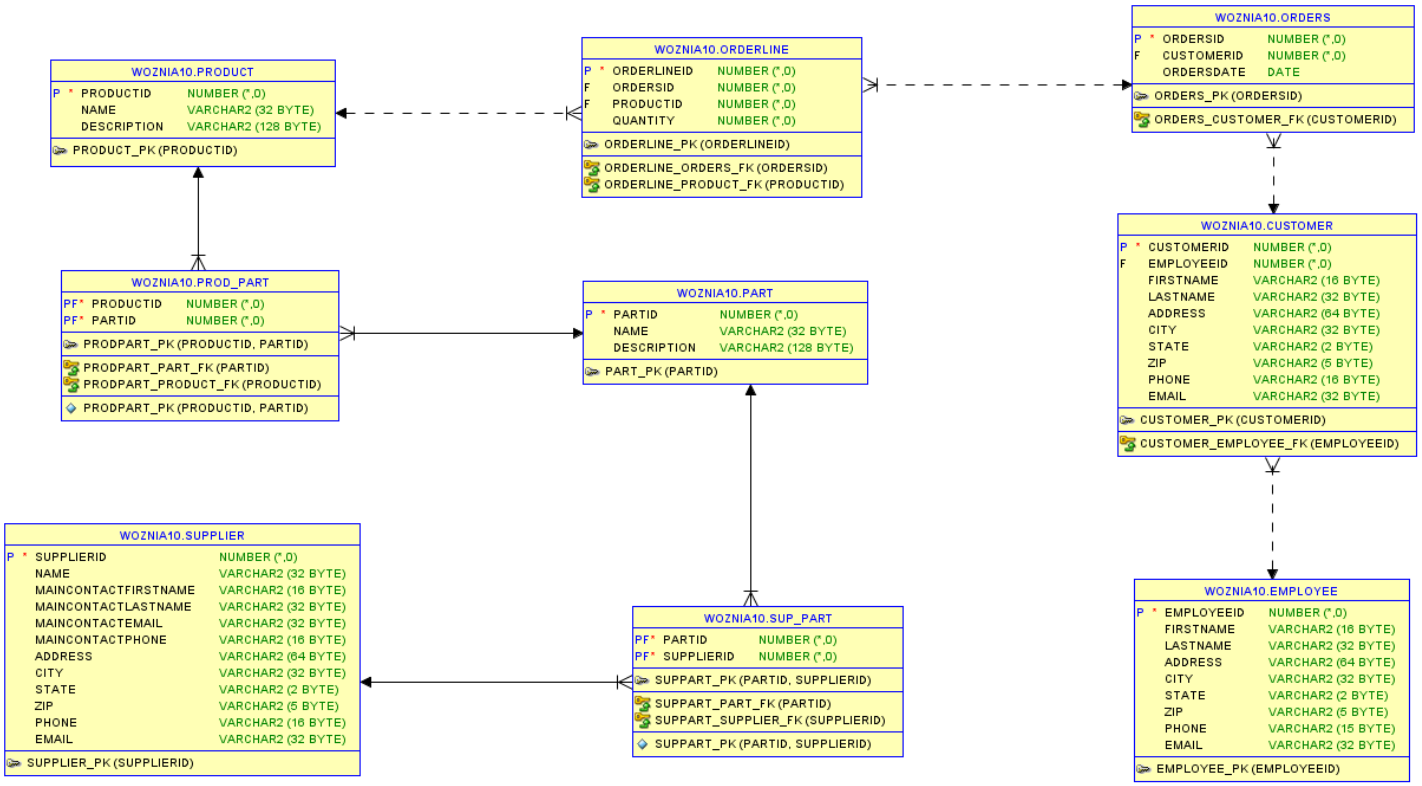
The bicycle shop industry is highly competitive, and managing inventory is a crucial aspect of running a successful business. The efficient management of inventory can lead to cost savings, increased revenue, and improved customer satisfaction. However, manual inventory management can be time-consuming and error-prone, leading to inaccurate stock levels, missed sales opportunities, and increased costs. To address these challenges, this project aims to develop a PL/SQL database that will allow a bicycle shop to manage its inventory more efficiently.

The project's motivation is to help the bicycle shop maintain accurate inventory information while also providing insights to drive the business's activities. The developed PL/SQL database will interact with a frontend GUI, providing an easy-to-use interface to manipulate the data stored in the database. With the help of PL/SQL, the bicycle shop will be able to generate reports to identify which parts are selling the most during a given time period, predict when stock levels on certain parts should be higher to meet customer demand, and determine which parts are not being ordered as frequently. This project's success will allow the bicycle shop to make data-driven decisions, reduce user error, and provide meaningful data for the operation of its daily business.

GitHub Repository

Database Description

Below is an Entity Relationship Diagram of our database schema.



Questions and Solutions

Below are the questions, their business value, and the PL/SQL solutions to them.

1: What are the current inventory levels for each bike part?

This helps the business visualize which parts are in high demand and which parts need to be restocked more frequently.

Solution:

```
SELECT p.NAME AS PART_NAME, COUNT(*) AS INVENTORY_LEVEL
FROM PART p
JOIN SUP_PART sp ON p.PARTID = sp.PARTID
JOIN SUPPLIER s ON sp.SUPPLIERID = s.SUPPLIERID
JOIN PRODUCT pr ON p.PARTID = pr.PRODUCTID
JOIN ORDERLINE ol ON pr.PRODUCTID = ol.PRODUCTID
GROUP BY p.NAME;
```

Results:

```
PART_NAME                      INVENTORY_LEVEL
------------------------------ ---------------
Tube                                        12
Chain                                       10
Seatpost                                    10
Handlebar Tape                               2
Brake Cable                                  9
Shift Cable                                  1
Tire                                        14
Cassette                                    16
Brake Pad                                    6
Bottom Bracket                               4
Chainring                                    2

PART_NAME                      INVENTORY_LEVEL
------------------------------ ---------------
Derailleur Hanger                            3
Brake Lever                                  1
Headset                                      3
```

2: What is the best-selling product?

This helps the business owner to determine their most popular product so a larger on-hand inventory could potentially be purchased to maintain availability of the most popular product. The best-selling product is defined as the product that appears on the most orders.

Solution:

```
Select count(ol.ProductID), p.ProductID, p.name
From OrderLine ol inner join Product p
On ol.productID = p.productID
Group by p.productID, p.name
Order by count(ol.ProductID) desc
Fetch first row only;
```

Results:

```
COUNT(OL.PRODUCTID)  PRODUCTID NAME
-------------------- ---------- --------------------------------
                   8          5 City Bike
```

3: How many orders did customers place in the last quarter?

This information can be used to track productivity to know which times of the year bicycle sales are higher and which are lower.

Solution:

```
Select count(*) AS LAST_QUARTER_ORDERS
FROM ORDERS
Where ORDERSDATE > SYSDATE - 90;
```

Results:

```
LAST_QUARTER_ORDERS
-------------------
                 50
```

4: On which day of the week are the most orders placed?

This would be important for both staffing and inventory reasons. Knowing what days the most staff are typically needed as well as what days the inventory would need to be highest is beneficial to the management.

Solution:

```
SELECT TO_CHAR(ORDERSDATE, 'DAY') AS DAY_OF_WEEK, COUNT(*) AS NUM_ORDERS
FROM ORDERS
GROUP BY TO_CHAR(ORDERSDATE, 'DAY')
ORDER BY COUNT(*) DESC;
```

Results:

```
DAY_OF_WE NUM_ORDERS
--------- ----------
WEDNESDAY          8
THURSDAY           8
SATURDAY           7
SUNDAY             7
TUESDAY            7
MONDAY             7
FRIDAY             6
```

5: How many bicycles are currently available for sale?

It is important for a shop owner to know the amount of inventory available so that he or she can be ready to order more when supplies run low, but before they run out.

Solution:

```
SELECT COUNT(*) AS Bike_Count
FROM PRODUCT
WHERE NAME LIKE '%Bike';
```

Results:

```
BIKE_COUNT
----------
        11
```

6: Which bike parts are selling the most?

This data can be used to determine what parts are the most popular. This can help when deciding what parts need to be continually purchased and what parts can be phased out of the store's orders.

Solution:

```
SELECT p.NAME AS PART_NAME, COUNT(ol.QUANTITY) AS QUANTITY_SOLD
FROM PART p
JOIN PROD_PART pp ON p.PARTID = pp.PARTID
JOIN PRODUCT pr ON pp.PRODUCTID = pr.PRODUCTID
JOIN ORDERLINE ol ON pr.PRODUCTID = ol.PRODUCTID
GROUP BY p.NAME
ORDER BY QUANTITY_SOLD DESC;
```

Results:

```
PART_NAME                       QUANTITY_SOLD
------------------------------- -------------
Brake Lever                                 8
Stem                                        8
Shift Cable                                 8
Tube                                        7
Bottom Bracket                              7
Cassette                                    7
Seatpost                                    6
Handlebar Tape                              6
Chainring                                   6
Chain                                       5
Cable Housing                               5

PART_NAME                       QUANTITY_SOLD
------------------------------- -------------
Tire                                        5
Brake Pad                                   5
Pedal                                       5
Brake Caliper                               5
Brake Cable                                 4
Headset                                     4
Derailleur Hanger                           4
Brake Hose                                  3
Derailleur Cable                            3
```

```
Brake Rotor                                3
Water Bottle Cage                          3

PART_NAME                         QUANTITY_SOLD
--------------------------------- -------------
Grip Tape                                  3
Computer Mount                             3
Power Meter                                2
Front Derailleur                           2
Rear Derailleur                            2
Chain Tool                                 2
Saddle                                     2
Chain Lubricant                            2
Wheelset                                   2
Battery                                    2
Bar End Plug                               2

PART_NAME                         QUANTITY_SOLD
--------------------------------- -------------
Seat Clamp                                 1
Wheel Skewer                               1
Pedal Cleat                                1
Bar End Shifter                            1
Chain Catcher                              1
Chainstay Protector                        1
Headset Spacer                             1
Fork                                       1
Derailleur Pulley                          1
```

7: Who was the customer that placed the most orders and how many orders did they place?

This data could be useful to the business owner to give a special discount to the customer(s) who place frequent orders with the bicycle shop.

Solution:

```
select count(o.customerid), o.customerID, c.firstName, c.lastName
from Orders o inner join Customer c
On o.customerID = c.customerID
Group by o.customerid, c.firstName, c.lastName
Order by count(o.customerid) desc
Fetch first row only;
```

Results:

```
COUNT(O.CUSTOMERID) CUSTOMERID FIRSTNAME        LASTNAME
------------------- ---------- ---------------- -------------
                  1         22 Jacob            Smith
```

8: When are large quantities of a product ordered?

This information is useful for the business because they will likely need to order more stock after a large quantity is purchased

Solution:

```
CREATE OR REPLACE TRIGGER LOW_STOCK_NOTIFICATION
AFTER INSERT ON ORDERLINE
FOR EACH ROW
WHEN (NEW.QUANTITY >= 10)
BEGIN
      DBMS_OUTPUT.PUT_LINE('10 OR MORE OF PRODUCT ' || :NEW.PRODUCTID || '
WERE JUST ADDED TO AN ORDER.');
END;
/
```

Results:

```
>INSERT INTO ORDERLINE(ORDERLINEID, ORDERSID, PRODUCTID, QUANTITY)
>VALUES (53, 1, 3, 10);

10 OR MORE OF PRODUCT 3 WERE JUST ADDED TO AN ORDER.


1 row inserted.
```

9: What is the average quantity of items included in an order?

This information could be valuable to determine productivity of employees at pushing sales. Comparing this over time could identify trends amongst employees or times of the year.

Solution:

```
SELECT AVG(QUANTITY) AS AVG_ITEMS_PER_ORDER
FROM ORDERLINE;
```

Results:

```
AVG_ITEMS_PER_ORDER
-------------------
               1.46
```

10: Which bicycle model is purchased most by customers?

This is an important piece of information for inventory reasons. If this bike is the most commonly purchased bike it would be logical for the company to have the highest inventory of this bike to support the demand.

Solution:

```
SELECT P.NAME, COUNT(*) AS NUM_PURCHASES
FROM PRODUCT P
JOIN ORDERLINE OL ON P.PRODUCTID = OL.PRODUCTID
GROUP BY P.NAME
ORDER BY COUNT(*) DESC
FETCH FIRST 1 ROWS ONLY;
```

Results:

```
NAME                             NUM_PURCHASES
-------------------------------- -------------
City Bike                                   8
```

Team Contribution Chart

Below is a chart of team member contributions for both Milestone 3 and the final presentation, given the significant overlap between the two.

Team Member	**Contribution**
Justin	-Make database load script -Create/format documents -Questions 8
Jaden	-Questions 3, 5, 9 -Work on front-end GUI
Evan	-Create Github Repo -Questions 1, 4, 6, 10
Jordan	-Write Project Background -Created database description ERD -Questions 2,7 -Troubleshoot errors in database load script