

# **Project Report**

Vladislav Mazur

Vancouver Island University

CSCI 370 - Introduction to Database Systems

Huizhu Liu

April 14, 2024

## **Introduction:**

The application being developed is intended for hockey team staff, particularly the equipment managers, to keep track of the team's gear. In this application they will be able to view and order based on the offers provided by the manufacturers, view past transactions, view currently available gear, and make incidents reports for missing or broken gear which will directly affect the currently available gear once an incident is posted.

## **Data Description:**

The core data in our applications can be summarized by 5 tables, 'Brands' 'Offers', 'Transactions', 'Gear', and 'Incidents'. The 'Brands' will be related to 'transactions' through 'Offers' due to accepting an offer will automatically add that transaction to its transaction's log. The 'incidents' will be related to 'gear' since filing an incident will cause the gear to be deleted from the gear table since it is no longer available for use. The 'transactions' will also be related to 'gear' since when accepting a transaction the gear bought will automatically get added to the gear table.

## **Application Requirements:**

There will be four main functionalities for this application.

### *1. View and Order Gear*

Users should be able to browse offers given by the gear manufacturers and place orders for required gear. This purchasing feature will help with acquiring new gear needed by the team.

### *2. View Transaction History*

Users should be able to view all of their past transactional history to track their purchases. This tracking feature will help with better financial management.

### *3. Manage Inventory*

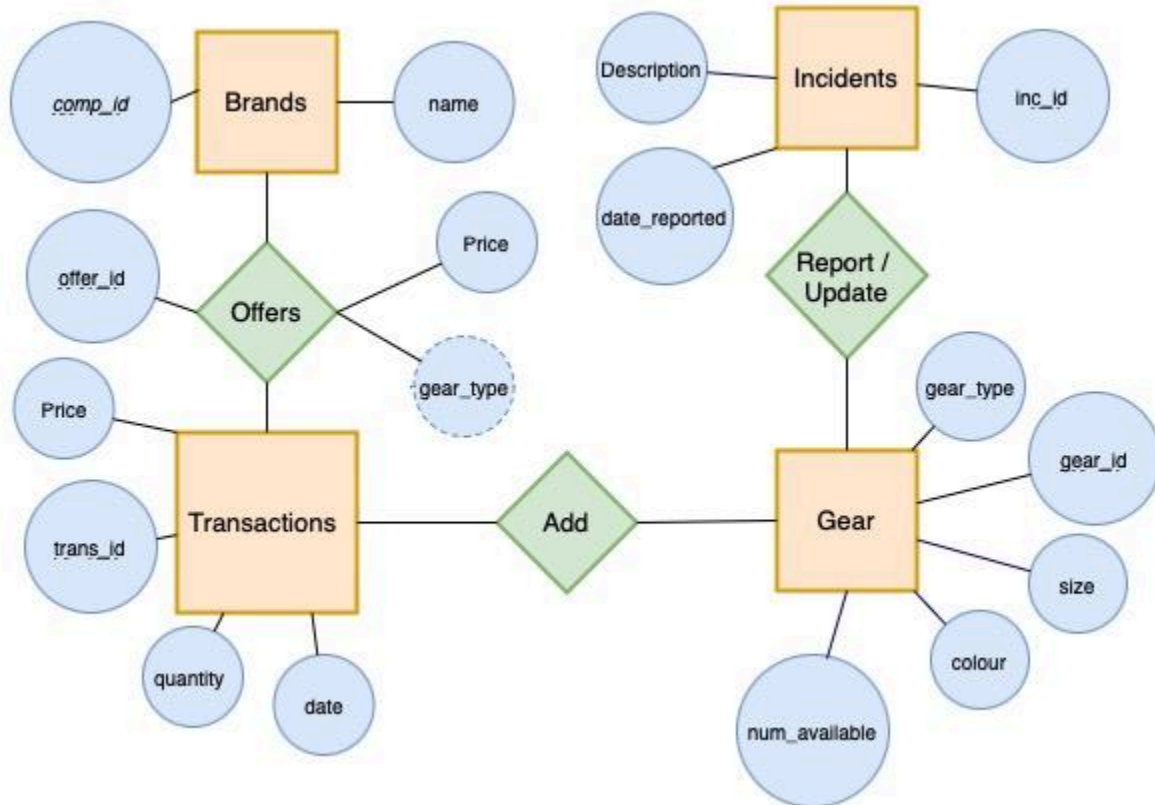
Users should be able to see the currently available inventory including quantities and specifications. This managing feature will help in making sure the team is ready for their upcoming events.

### *4. Report and View Incidents*

Users should be able to report and view incidents such as missing or damaged gear. This reporting feature will help in addressing gear issues in a timely fashion.

## Database Design:

ER Diagram:



Schema:

```
CREATE TABLE Brands (  
    comp_id CHAR(2) PRIMARY KEY,  
    b_name VARCHAR(255)  
);
```

```
CREATE TABLE Offer (  
    offer_id CHAR(2) PRIMARY KEY,  
    comp_id CHAR(2),  
    gear_type VARCHAR(255),  
    price DECIMAL(10, 2),
```

```
FOREIGN KEY (comp_id) REFERENCES Brands(comp_id)
);
```

```
CREATE TABLE Transaction (
    trans_id CHAR(2) PRIMARY KEY,
    offer_id CHAR(2),
    quantity INT,
    price DECIMAL(10, 2),
    purchase_date DATE,
    FOREIGN KEY (offer_id) REFERENCES Offer(offer_id)
);
```

```
CREATE TABLE Gear (
    gear_id CHAR(2) PRIMARY KEY,
    gear_type VARCHAR(255),
    gear_size VARCHAR(4),
    colour VARCHAR(50),
    num_available INT
);
```

```
CREATE TABLE Incident (
    inc_id CHAR(2) PRIMARY KEY,
    gear_id CHAR(2),
    i_description VARCHAR(255),
    date_reported DATE,
    FOREIGN KEY (gear_id) REFERENCES Gear(gear_id)
);
```

## **Implementation:**

The basic implementation of all of the above requirements have been completed. The user is able to view their available offers and choose to make one (buy it) or to return back to the main menu. If an offer is bought then it will be added to the transaction history and add the quantity bought to the available gear quantity. Viewing the history of the transaction will show the newly accepted offer. The user is able to view their inventory to check on all their gear, its attributes, and the quantity. Reporting an incident will take the id of the gear and decrement it by one in the Gear table, the user is also able to describe why the incident happened. The Incident will then be documented in the incident history and can be viewed by the user at any time.

The testing was run off of the data below as well as:

### Make Offer:

Gave the function a trans ID of 92, choose an offer by the ID 14 and decided to buy 2.

Transaction ID: 92

Offer ID: 14

Quantity: 2

Price: \$89.99

Purchase Date: 14-APR-24

Note that Price is based off of offer ID; purchase date was taken from SYSDATE.

### Report Incident:

Gave the function an incident ID of 65, chose the Gear by ID 32 and decided to say it was “wrong size”.

Incident ID	Gear ID	Description	Date Reported
65	32	wrong size	14-APR-24

Note purchase date was taken from SYSDATE.

## **Testing:**

### *Sample Data:*

#### Brands:

```
INSERT INTO Brands (comp_id, b_name) VALUES ('01', 'Bauer');
INSERT INTO Brands (comp_id, b_name) VALUES ('02', 'CCM');
INSERT INTO Brands (comp_id, b_name) VALUES ('03', 'True');
INSERT INTO Brands (comp_id, b_name) VALUES ('04', 'Sherwood');
```

#### Offers:

```
INSERT INTO Offer (offer_id, comp_id, gear_type, price) VALUES ('11', '01', 'Hockey Stick',
129.99);
INSERT INTO Offer (offer_id, comp_id, gear_type, price) VALUES ('12', '02', 'Goalie Mask',
299.99);
INSERT INTO Offer (offer_id, comp_id, gear_type, price) VALUES ('13', '03', 'Hockey Skates',
199.99);
INSERT INTO Offer (offer_id, comp_id, gear_type, price) VALUES ('14', '04', 'Hockey Gloves',
89.99);
```

#### Transactions:

```
INSERT INTO Transactions (trans_id, offer_id, quantity, price, purchase_date) VALUES ('21',
'11', 5, 129.99, TO_DATE('2024-04-13', 'YYYY-MM-DD'));
INSERT INTO Transactions (trans_id, offer_id, quantity, price, purchase_date) VALUES ('22',
'12', 2, 299.99, TO_DATE('2024-04-12', 'YYYY-MM-DD'));
INSERT INTO Transactions (trans_id, offer_id, quantity, price, purchase_date) VALUES ('23',
'13', 3, 199.99, TO_DATE('2024-04-11', 'YYYY-MM-DD'));
INSERT INTO Transactions (trans_id, offer_id, quantity, price, purchase_date) VALUES ('24',
'14', 4, 89.99, TO_DATE('2024-04-10', 'YYYY-MM-DD'));
```

#### Gear:

```
INSERT INTO Gear (gear_id, gear_type, gear_size, colour, num_available) VALUES ('31',
'Hockey Stick', 'XL', 'Black', 20);
INSERT INTO Gear (gear_id, gear_type, gear_size, colour, num_available) VALUES ('32',
'Goalie Mask', 'L', 'White', 15);
INSERT INTO Gear (gear_id, gear_type, gear_size, colour, num_available) VALUES ('33',
'Hockey Skates', '10', 'Black', 30);
INSERT INTO Gear (gear_id, gear_type, gear_size, colour, num_available) VALUES ('34',
'Hockey Gloves', 'XXL', 'Black', 25);
```

#### Incidents:

```
INSERT INTO Incident (inc_id, gear_id, i_description, date_reported) VALUES ('41', '31',
'Cracked blade', TO_DATE('2024-04-13', 'YYYY-MM-DD'));
INSERT INTO Incident (inc_id, gear_id, i_description, date_reported) VALUES ('42', '32', 'Strap
broken', TO_DATE('2024-04-12', 'YYYY-MM-DD'));
INSERT INTO Incident (inc_id, gear_id, i_description, date_reported) VALUES ('43', '33',
'Blade chipped', TO_DATE('2024-04-11', 'YYYY-MM-DD'));
INSERT INTO Incident (inc_id, gear_id, i_description, date_reported) VALUES ('44', '34', 'Torn
palm', TO_DATE('2024-04-10', 'YYYY-MM-DD'));
```

### Results:

After running the above inserts and the test mentioned previously the database now looks like the following:

Available Offers:

-----

Offer ID: 11

Gear Type: Hockey Stick

Price: \$129.99

-----

Offer ID: 12

Gear Type: Goalie Mask

Price: \$299.99

-----

Offer ID: 13

Gear Type: Hockey Skates

Price: \$199.99

-----

Offer ID: 14

Gear Type: Hockey Gloves

Price: \$89.99

-----

-----  
Transaction ID: 22

Offer ID: 12

Quantity: 2

Price: \$299.99

Purchase Date: 12-APR-24  
-----

Transaction ID: 23

Offer ID: 13

Quantity: 3

Price: \$199.99

Purchase Date: 11-APR-24  
-----

Transaction ID: 24

Offer ID: 14

Quantity: 4

Price: \$89.99

Purchase Date: 10-APR-24  
-----

Transaction ID: 90

Offer ID: 12

Quantity: 3

Price: \$299.99

Purchase Date: 14-APR-24  
-----

Transaction ID: 92

Offer ID: 14



Quantity: 2

Price: \$89.99

Purchase Date: 14-APR-24

-----

-----

Incident ID	Gear ID	Description	Date Reported
-------------	---------	-------------	---------------

-----

41	31	Cracked blade	13-APR-24
----	----	---------------	-----------

-----

42	32	Strap broken	12-APR-24
----	----	--------------	-----------

-----

43	33	Blade chipped	11-APR-24
----	----	---------------	-----------

-----

44	34	Torn palm	10-APR-24
----	----	-----------	-----------

-----

65	32	wrong size	14-APR-24
----	----	------------	-----------

-----

-----

Brand ID	Name
----------	------

01	Bauer
----	-------

02	CCM
----	-----

03	True
----	------

04	Sherwood
----	----------

Note that there is more information in the database, the preceding information comes from what the user sees.

Brands are also not seen or used by the user, these could be integrated in later version of the product.

## **Conclusion and Future Work:**

Throughout this project I got very comfortable with prepared statements and updating tables in SQL. I plan to continue working on this project a little further once handed in to make sure the user cannot enter an offer or transaction ID that already exists, I wanted to do this through running another statement that would check if said ID is already in the table and if so asking the user to input a different one. Eventually this would change into an incrementing ID however that is out of the scope for this specific project. I would also make a way for new gear with different attributes to be able to be bought or changed, currently only the 4 pieces of gear in the database are able to be removed or added. This wouldn't be difficult, just needs more data. To bring this to large scale all the ID's would have to have a remake as well, incrementing would be best but to keep up with oracle database I would change the ID chars to something larger, since my test data will not be larger than two digits worth of samples, I kept it low. Another addition would be making the managed inventory be able to show when a specific equipment is running low so that the user is able to make sure they are always stocked up. I am happy with how this project turned out but would've hoped to get the tools to start the implementation sooner.