

Project Group #1

Grant Wu and Jessica Ramirez

Project Step 1 - Database Design

1. The Oaklaura Bike Cooperative

The Oaklaura Bike Cooperative is a non-profit organization that accepts donations of old or broken bicycles, refurbishes them, and then sells them at an affordable price. Due to limited funding, the co-op operates with a small team of employees and relies heavily on volunteers to assist with bicycle repairs. The co-op's limited funding also requires they use a basic POS system that can only process one bike sale at a time, setting a limit of one bike per sales order.

Store Personnel consist of both volunteers and employees. Most volunteers are not experienced bike mechanics, so they often can't fully repair a bike during the few hours the co-op is open for volunteer work. To maintain continuity and organization, volunteers are expected to repair what they can during their shift and document their progress in a report. This allows the next volunteer and/or employee to review the logs and continue the work where the previous one left off. Once a volunteer believes a bike is fully repaired, a trained employee inspects it to ensure it meets safety standards before placing it on the sales floor.

Historically, the co-op tracked repair progress and sales using handwritten notecards stored in a filing cabinet. However, as the organization grows, this system has become increasingly difficult to manage. Implementing a database would be an ideal solution for organizing and sharing information between volunteers and employees about the status of each bicycle.

2. Database Schema

Bikes Table

Contains details on a particular bike within the co-op

- **bikeID [PK]:** int, not NULL, auto_increment
- **color:** enum('Black', 'White', 'Red', 'Blue', 'Green', 'Pink', 'Purple', 'Yellow', 'Orange', 'Silver', 'Other'), not NULL
- **style:** enum('Mountain', 'Road', 'Fat', 'Hybrid', 'Enduro', 'BMX', 'Cruiser', 'Kids', 'Electric'), not NULL
- **brand:** varchar(45)
- **status:** enum('In Repair', 'Employee Review', 'For Sale', 'Sold'), not NULL
- **dateReceived:** date, not NULL
- **completed:** tinyint(default 0=false), not NULL

Relationships:

- M:M relationship between Bikes and StorePersonnel is implemented with bikeID and personnelID as FK's within both RepairReports and within SalesReports.
- 1:1 relationship between Bikes and SalesReports is implemented by bikeID as a FK within SalesReports. *Note that due to our outdated POS system, only one bike can be sold at a time (i.e. only one Bike instance per SalesReport instance).*
- 1:M relationship between Bikes and RepairReports is implemented with bikeID as a FK within RepairReports.
- 1:M relationship between Bikes and SalesReports is implemented with bikeID as a FK within SalesReports.

StorePersonnel Table

Holds information on store employees and volunteers

- **personnelID [PK]:** int, not NULL, auto_increment
- **firstName:** varchar(45) not NULL
- **lastName:** varchar(45) not NULL
- **phone:** varchar(20)
- **email:** varchar(100), not NULL, unique
- **role:** enum('Employee', 'Volunteer'), not NULL

Relationships:

- M:M relationship between StorePersonnel and Bikes is implemented with bikeID and personnelID as FK's within both RepairReports and within SalesReports.
- 1:M relationship between StorePersonnel and RepairReports is implemented with personnelID as a FK within RepairReports.
- 1:M relationship between StorePersonnel and SalesReports is implemented with personnelID as a FK within SalesReports.

Customers Table

Holds customer information

- **customerID [PK]:** int, not NULL, auto_increment
- **firstName:** varchar(45) not NULL
- **lastName:** varchar(45) not NULL
- **phone:** varchar(20)
- **email:** varchar(100), not NULL, UNIQUE

Relationships:

- 1:M relationship between Customers and SalesReports is implemented with customerID as a FK inside of SalesReports.

RepairReports Table

Holds repair information performed on a particular bikes (Bikes_StorePersonnel Intersection Table that includes additional repair information)

- **repairID [PK]:** int, not NULL, auto_increment
- **personnelID [FK - StorePersonnel]:** int, not NULL
- **bikeID [FK - Bikes]:** int, not NULL
- **dateRepaired:** datetime, not NULL
- **hoursSpent:** decimal(4,2), not NULL
- **description:** varchar(255)

Relationships:

- 1:M relationship between RepairReports and StorePersonnel is implemented with personnelID as a FK inside RepairReports.
- 1:M relationship between RepairReports and Bikes is implemented with bikeID as a FK inside RepairReports.

SalesReports Table

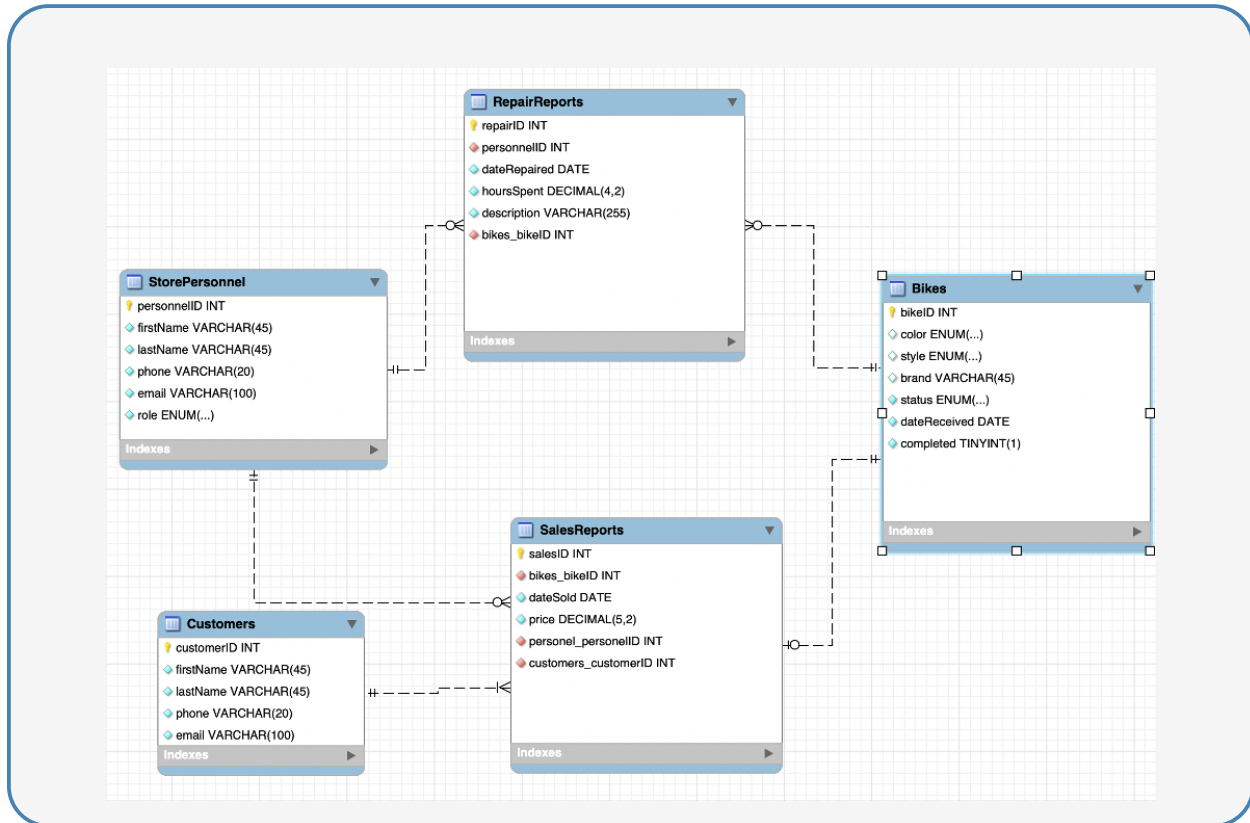
Holds information pertaining to the sale of a particular bike (Bikes_StorePersonnel Intersection Table that includes additional sale information)

- **salesID [PK]:** int, not NULL, auto_increment
- **personnelID [FK - StorePersonnel]:** int, not NULL
- **bikeID [FK - Bikes]:** int, not NULL
- **customerID [FK - Customers]:** int, not NULL
- **dateSold:** date, not NULL
- **price:** decimal(5,2), not NULL

Relationships:

- 1:1 relationship between Bikes and SalesReports is implemented by bikeID as a FK within SalesReports. *Note that due to our outdated POS system, only one bike can be sold at a time (i.e. only one Bike instance per SalesReport instance).*
- 1:M relationship between SalesReports and StorePersonnel is implemented with personnelID as a FK within SalesReports.
- 1:M relationship between SalesReports and Customers is implemented with customerID as a FK within SalesReports.

3. Entity-Relationship Diagram



4. Citations

- Inspiration for the Bike Co-Op came from The Recyclery, a non-profit bike shop based out of Chicago, IL (last retrieved on 4/9/2025): <https://www.therecyclery.org/>
- MySQL workbench was used to create the ERD diagram shown above.
- The L^AT_EX template used here was adapted from the Cleese-Assignment template v.2.0 (retrieved on 4/2/2025): <https://latextemplates.com/template/cleese-assignment>
- TeXShop was used for all L^AT_EX related compilations.
- All database and design related work is original.