***Group: (Najib Abdulkadir, Gurminder Singh Badwal)***

*Section: 1 (Introduction)*

The Database structure is tailored to efficiently manage and track critical information within the realm of a bicycle repair store. It encompasses tables dedicated to customers, bicycles, mechanics, invoices, and problems, integral components of this enterprise. Each table serves a distinct purpose and establishes relationships with others, ensuring organized data retrieval and management.

1. Customers:
   * Holds vital information about customers, including their cusID, first name, last name, phone number, and country.
   * Primary key: cusID
2. Bicycle:
   * Contains comprehensive details about bicycles, such as their VIN, make, model, price, and size.
   * Primary key: VIN
3. Mechanic:
   * Compiles pertinent data about mechanics, including their mechID, first name, last name, age, and country.
   * Primary key: mechID
4. Invoice:
   * Serves as a repository for invoices and encompasses invoiceID, order date in, order date out, cusID, and VIN.
   * Primary key: invoiceID
   * Foreign keys: cusID, VIN
5. Problem:
   * Catalogs specific problems with corresponding information including proID, problem description, price, invoiceID, and mechID.
   * Primary key: proID
   * Foreign keys: invoiceID, mechID

*Explanation of the Data (Bicycle Repair Store):*

This database system is meticulously designed to cater specifically to the operations of a bicycle repair store. It centralizes information related to customers, bicycles, mechanics, invoices, and problem records. For instance, the 'Customers' table retains details of clientele, facilitating streamlined communication and service provision. The 'Bicycle' table encompasses critical data about the store's inventory, including unique identifiers (VIN), make, model, pricing, and sizing information. The 'Mechanic' table keeps track of the skilled workforce, recording their identification, personal details, and relevant demographics. The 'Invoice' table acts as a hub for transactional records, detailing customer interactions, order dates, and associated bicycle information. Lastly, the 'Problem' table captures specific issues encountered during repairs, along with cost implications, tying them back to the original invoice and the responsible mechanic.

*Relationships between the tables in the database:*

1. Customers and Invoice:
   * Many to one relationship
   * Multiple invoices can be associated with one customer, but one customer can have multiple invoices.
2. Bicycle and Invoice:
   * Many to one relationship
   * Multiple invoices can involve one bicycle, but one bicycle can be in multiple invoices.
3. Mechanic and Problem:
   * Many to one relationship
   * Multiple problems can be associated with one mechanic, but one mechanic can solve multiple problems.

*Section 2: Group Membership and Tasks*-Najib Abdulkadir: [Abdu0451@algonquinlive.com](mailto:Abdu0451@algonquinlive.com)

-Gurminder Singh Badwal: [badw0006@algonquinlive.com](mailto:badw0006@algonquinlive.com)

*Section 3: Historical Data Framework and Multi-valued Fields*

In our database system, we plan to introduce multi-valued historical data fields in the following tables and relationships:

1. Customers - Contains (relationship)
2. Invoice - Is-related-to (relationship)
3. Problem - Is-a (relationship)

For each of these fields, we will create a set of triggers and stored procedures to handle INSERTS, UPDATES, and DELETES, and to keep a history of changes using timestamps. This will allow us to track changes and updates in our database system over time.

**Database Schema**

**Entity Descriptions**

* **Customers**: Stores customer information.
* **Bicycle**: Contains details about bicycles.
* **Mechanic**: Holds information about mechanics.
* **Invoice**: Tracks invoices for services rendered.
* **Problem**: Records specific problems addressed in each repair job.
* **Customers\_History**, **Invoice\_History**, **Problem\_History**: Tables for tracking historical data changes.

Steps:  
  
  
  
First, Creating the Tables: A screenshot of a computer program

Description automatically generated

Creating the Table history:

A screenshot of a computer

Description automatically generated

And the Second History: A screenshot of a computer program

Description automatically generated

"First, Creating the Trigger:

A screenshot of a computer

Description automatically generated

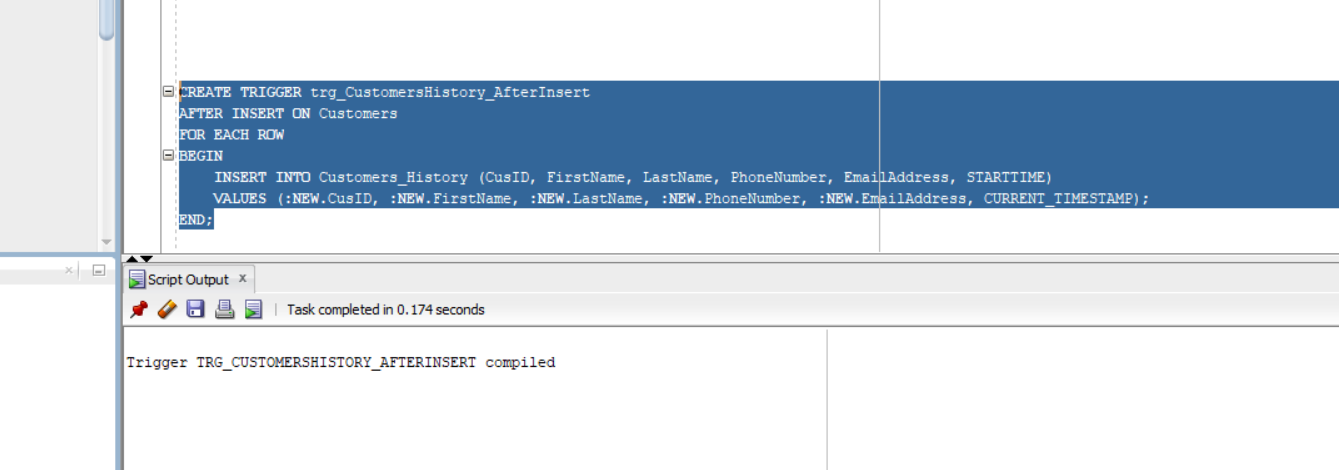
A screenshot of a computer

Description automatically generated

Create the Second Trigger:

A screenshot of a computer

Description automatically generated



**Triggers:**

* **trg\_CustomersHistory\_AfterInsert**: Captures new customer data insertions.
* **trg\_CustomersHistory\_AfterUpdate**: Records updates made to customer data.
* **trg\_CustomersHistory\_AfterDelete**: Records Delete made to customer data.
* **trg\_InvoiceHistory\_AfterUpdate**: Records updates made to Invoice data.
* **trg\_InvoiceHistory\_AfterInsert**: Captures new Invoice data insertions.
* **trg\_InvoiceHistory\_AfterDelete**: Records Delete made to Invoice data.
* **trg\_ProblemHistory\_AfterUpdate**: Records updates made to Problem data.
* **trg\_ProblemHistory\_AfterInsert**: Captures new Problem data insertions.
* **trg\_ProblemHistory\_AfterDelete**: Records Delete made to Problem data.
* **check\_dates\_before\_insert: OrderDateOut cannot be earlier than Orderdatein**
* validate\_customer\_data: First name cannot be null  
    
    
  A screenshot of a computer

  Description automatically generated

Start Inserting into Customers Table: A screenshot of a computer

Description automatically generated

Update the Customer table :

A screenshot of a computer error

Description automatically generated

Delete the customer table: A screenshot of a computer

Description automatically generated

Inserting to Table Invoice:

A screenshot of a computer

Description automatically generated

*Update the Invoice table:*

A screenshot of a computer

Description automatically generated

Delete from the Invoice Table: A screenshot of a computer

Description automatically generated

Insert to Problem Table:

A screenshot of a computer

Description automatically generated

*Update table problem:*

A screenshot of a computer

Description automatically generated

*Delete from table problem:*

A screenshot of a computer

Description automatically generated

***The result from the table Customer\_history:***

A screenshot of a computer

Description automatically generated

***The result from the table Invoice\_history:***

A screenshot of a computer

Description automatically generated

***The result from the table Problem \_history:***

A screenshot of a computer

Description automatically generated

***Views:***

***CustomerInvoices****: Combines customer data with their respective invoices.  
the* ***RepairDetails*** *view would allow users to easily access comprehensive details about each repair job without needing to understand the underlying complex SQL query.*A screenshot of a computer

Description automatically generated *Link to Access :*

***I created OBDC(64bit/32bit) it is called Assignment2\_last***

A screenshot of a computer error

Description automatically generated

***Now create new Access and open it***A screenshot of a computer

Description automatically generated

***Go to External Data***

A screenshot of a computer

Description automatically generated

***Navigate to” New Data Source”***A screenshot of a computer

Description automatically generated

***Click on ”From Other Sources” => And then “ODBC Database”***

A screenshot of a computer

Description automatically generated

***Click Ok***

***It is going to show you this page :***A screenshot of a computer

Description automatically generated

***Go to the second bar “Machine Data Source”***A screenshot of a computer

Description automatically generated

***Click on “Assignmnet2\_last” and then ok.***

A screenshot of a computer

Description automatically generated

***Choose the tables that you made it in oracle.***

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

***Finally press OK. You have imported successfully  
  
A screenshot of a computer

Description automatically generated***

***Access:   
Relationship:***A screenshot of a computer

Description automatically generated