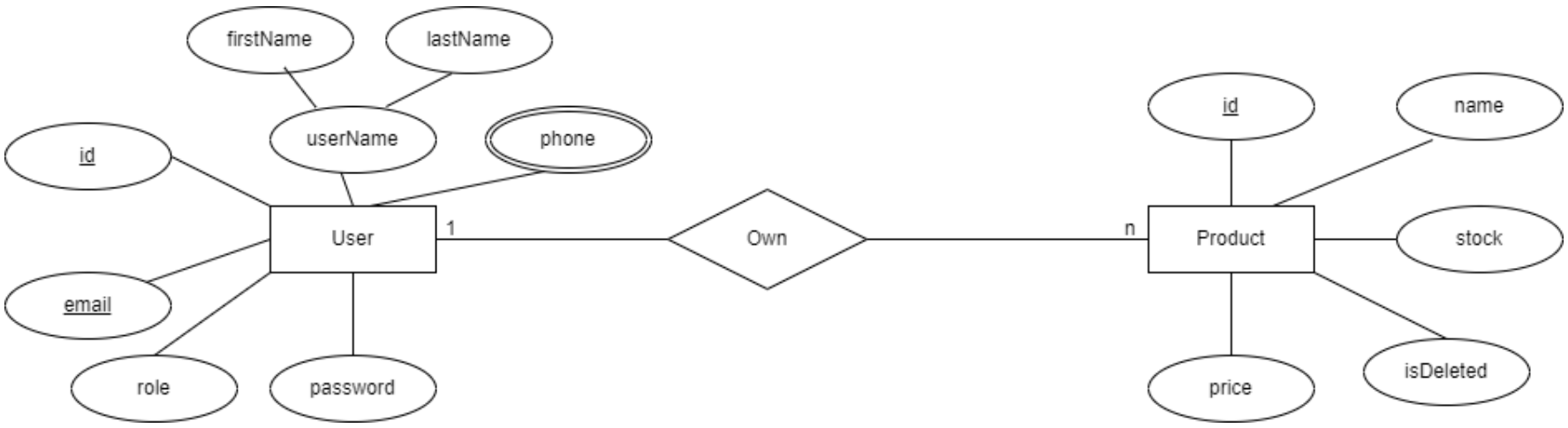
Assignment5

Part 1: ERD Diagram (1 Grade)

Musicana records have decided to store information on musicians who perform on their albums in a database. The company has wisely chosen to hire you as a database designer.

- Each musician that is recorded at Musicana has an ID number,a name, an address (street, city) and a phone number.
- Each instrument that is used in songs recorded at Musicana has a unique name and a musical key (e.g., C, B-flat, E-flat).
- Each album that is recorded at the Musicana label has a unique title, a copyright date, and an album identifier.
- Each song recorded at Musicana has a unique title and an author.
- Each musician may play several instruments, and a given instrument may be played by several musicians.
- Each album has a number of songs on it, but no song may appear on more than one album.
- Each song is performed by one or more musicians, and a musician may perform a number of songs.
- Each album has exactly one musician who acts as its producer.
- A producer may produce several albums.

Part2: Design a schema (Mapping) for the following ERD. (Use any design tool you want) (1 Grade)



Part 3: (Using Node.js and MySQL) Answer the Questions below based on the given Scenario

The small retail store needs a database to manage information about its products, suppliers, and sales.

Database Requirements

1. **Products Table:**
   - **ProductID**: Unique identifier for each product (integer, primary key, auto-increment).
   - **ProductName**: Name of the product (**text**).
   - **Price**: Price of the product (**decimal**).
   - **StockQuantity**: Quantity of the product in stock (**integer**).
   - **SupplierID**: ID of the supplier providing the product (**integer, foreign key referencing Suppliers**).
2. **Suppliers Table:**
   - **SupplierID:** Unique identifier for each product (integer, primary key, auto-increment).
   - **SupplierName**: Name of the supplier (**text**).

Assignment5

- **ContactNumber:** Supplier's contact number (**text**).

3. **Sales Table:**
   - **SaleID:** Unique identifier for each product (integer, primary key, auto-increment).
   - **ProductID:** Reference to the product sold (**integer, foreign key referencing Products**).
   - **QuantitySold:** Quantity of the product sold (**integer**).
   - **SaleDate:** Date of sale (**date**).

(Using Node.js and MySQL) generate queries that perform the following tasks (8 Grades):

1- Create the required tables for the retail store database based on the tables structure and relationships. (0.5 Grade)

2- Add a column "**Category"** to the Products table. (0.5 Grade)

3- Remove the **"Category"** column from Products. (0.5 Grade)

4- Change **"ContactNumber"** column in Suppliers to **VARCHAR** (15). (0.5 Grade)

5- Add a **NOT NULL** constraint to ProductName. (0.5 Grade)

6- **Perform Basic Inserts:** (0.5 Grade)
   a. Add a supplier with the name **'FreshFoods'** and contact number **'01001234567'.**
   b. Insert the following three products, all provided by **'FreshFoods'**:
      i. **'Milk'** with a price of **15.00** and stock quantity of **50.**
      ii. **'Bread'** with a price of **10.00** and stock quantity of **30.**
      iii. **'Eggs'** with a price of **20.00** and stock quantity of **40.**
   c. Add a record for the sale of **2** units of **'Milk'** made on **'2025-05-20'**.

7- Update the price of **'Bread'** to **25.00**. (0.5 Grade)

8- Delete the product **'Eggs'.** (0.5 Grade)

9- Retrieve the total quantity sold for each product. (0.5 Grade)

10- Get the product with the highest stock. (0.5 Grade)

11- Find suppliers with names starting with 'F'. (0.5 Grade)

12- Show all products that have never been sold. (0.5 Grade)

13- Get all sales along with product name and sale date. (0.5 Grade)

14- Create a user **"store_manager"** and give them **SELECT, INSERT, and UPDATE** permissions on **all tables**. (0.5 Grade)

15- Revoke **UPDATE** permission from **"store_manager"**. (0.5 Grade)

16- Grant **DELETE** permission to **"store_manager"** only on the Sales table. (0.5 Grade)

Bonus (2 Grades)

How to deliver the bonus?

1- Solve the problem Customer Who Visited but Did Not Make Any Transactions on **LeetCode**
2- Inside your assignment folder, create a **SEPARATE FILE** and name it "bonus.txt"
3- Copy the code that you have submitted on the website inside "bonus.txt" file