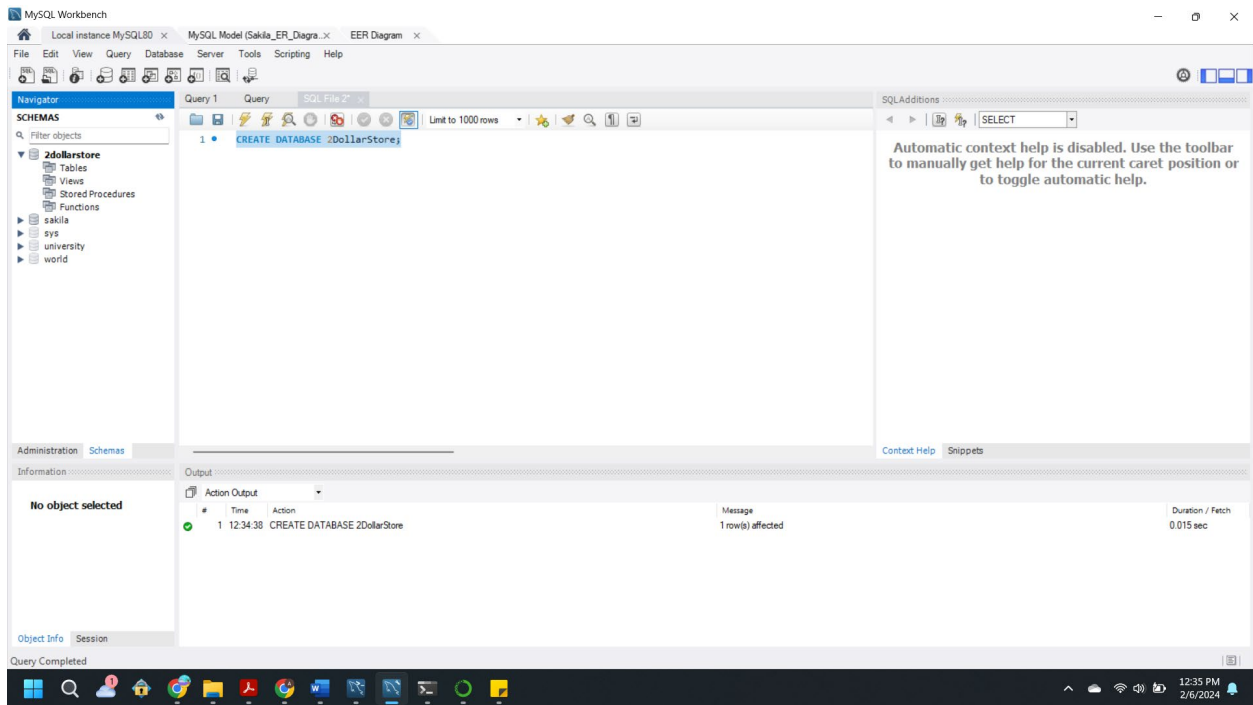# Homework_1_Aparna Bharathi Suresh

Question 1:

Create a new MySQL database named '2DollarStore'.

Screenshot 1:
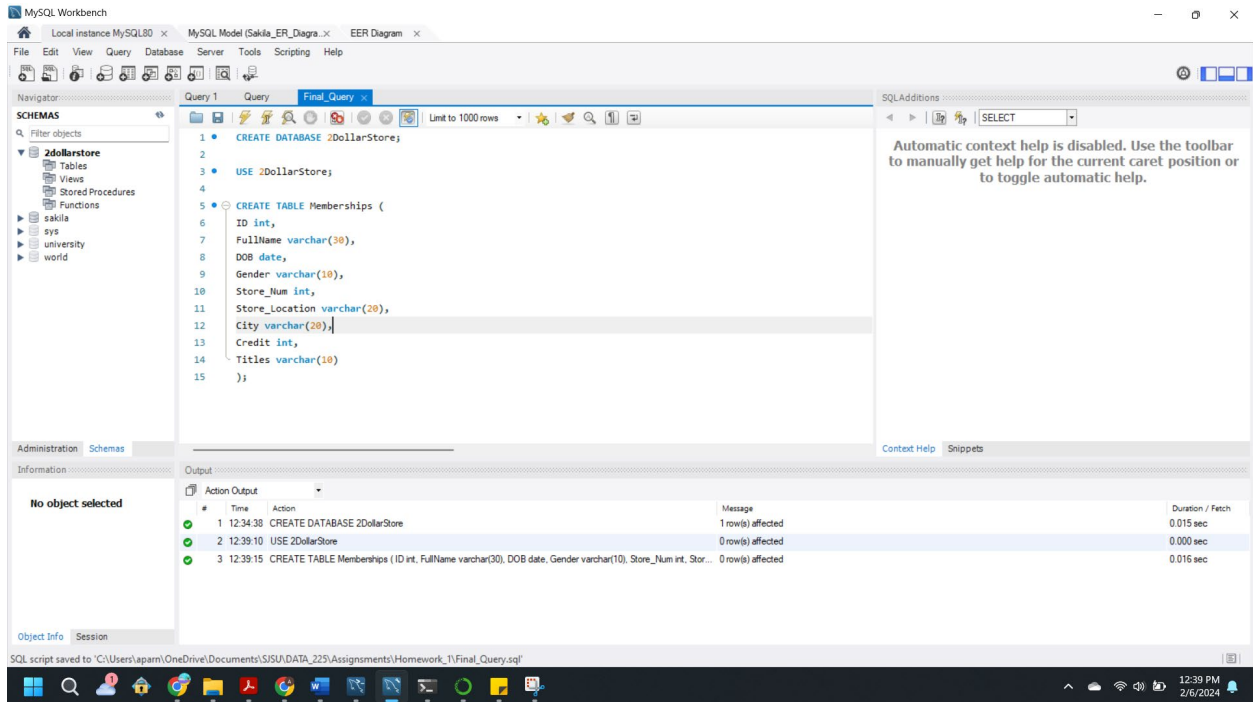

Code 1:

CREATE DATABASE 2DollarStore;

Question 2:

Create table memberships with schema (id, name, DOB, gender, store number, store location, city, credit, titles).

Screenshot 2:

Code 2:

```
USE 2DollarStore;

CREATE TABLE Memberships (
ID int,
FullName varchar(30),
DOB date,
Gender varchar(10),
Store_Num int,
Store_Location varchar(20),
City varchar(20),
Credit int,
Titles varchar(10)
);
```

Justification:

1. ID - I used <u>int</u> data type for id, as id can have up to 10 digits.
2. FullName – I used <u>varchar(30)</u> because name is a string and varchar stores a string of variable length.
3. DOB – I used <u>date</u> data type for the date of birth because its format is YYYY-MM-DD and date of birth doesn't require any other additional information like time, so date is the perfect datatype matching the DOB.
4. Gender - I used <u>varchar(10)</u> for gender because it has values like "Male", "Female" and "Other", as it has strings with variable length I used varchar.
5. Store_Num – I used <u>int</u> for store num as it has integer values.
6. Store_Location – I used <u>varchar</u>(20) for store location as it has strings of variable length.
7. City – I used <u>varchar</u>(20) as city names are strings with variable length.
8. Credit – I used <u>int</u> for credit as the credit point will be a whole number.
9. Titles – I used <u>varchar</u>(10) for title as it has values like "Gold", "Silver" and "Platinum".

Question 3:

Insert 12 records into the memberships table.

Screenshot 3:

Code 3:

Row_1:

```sql
INSERT INTO Memberships VALUES (30351,"ANA ROSE","1994-05-12","Female",100,"DUBLIN_BLVD","DUBLIN",103,"GOLD");
```

Row_2:

```sql
INSERT INTO Memberships VALUES (30352,"James William","1984-08-22","Male",130,"Independence Drive","Livermore",18,"Silver");
```

Row_3:

```sql
INSERT INTO Memberships VALUES (30353,"Ammy Jane","1974-11-12","Female",110,"Tassajara","DUBLIN",54,"Platinum");
```

Row_4:

```sql
INSERT INTO Memberships VALUES (30354,"Jeffercy Kim","2000-05-06","Female",110,"Tassajara","DUBLIN",107,"GOLD");
```

Row_5:

```sql
INSERT INTO Memberships VALUES (30355,"Aparna Suresh","1998-05-31","Female",100,"DUBLIN_BLVD","DUBLIN",25,"Silver");
```

Row_6:

```sql
INSERT INTO Memberships VALUES (30356,"Sowmya Vignesh","1991-09-27","Female",130,"Independence Drive","Livermore",70,"GOLD");
```

Row_7:

```sql
INSERT INTO Memberships VALUES (30357,"Chang Lee","1984-06-15","Male",110,"Tassajara","DUBLIN",40,"Platinum");
```

Row_8:

```sql
INSERT INTO Memberships VALUES (30358,"Suma Blaji","1964-01-22","Female",160,"ElCamino","Sunnyvale",70,"GOLD");
```

Row_9:

```sql
INSERT INTO Memberships VALUES (30359,"Raji Saravanan","2001-02-12","Female",120,"Santa_Rita","Pleasanton",251,"Platinum");
```

Row_10:

```sql
INSERT INTO Memberships VALUES (30360,"Anu Rama","1995-08-02","Female",100,"DUBLIN_BLVD","DUBLIN",10,"Silver");
```

Row_11:

```sql
INSERT INTO Memberships VALUES (30361,"Suja K","1999-09-12","Female",110,"Tassajara","DUBLIN",35,"Silver");
```
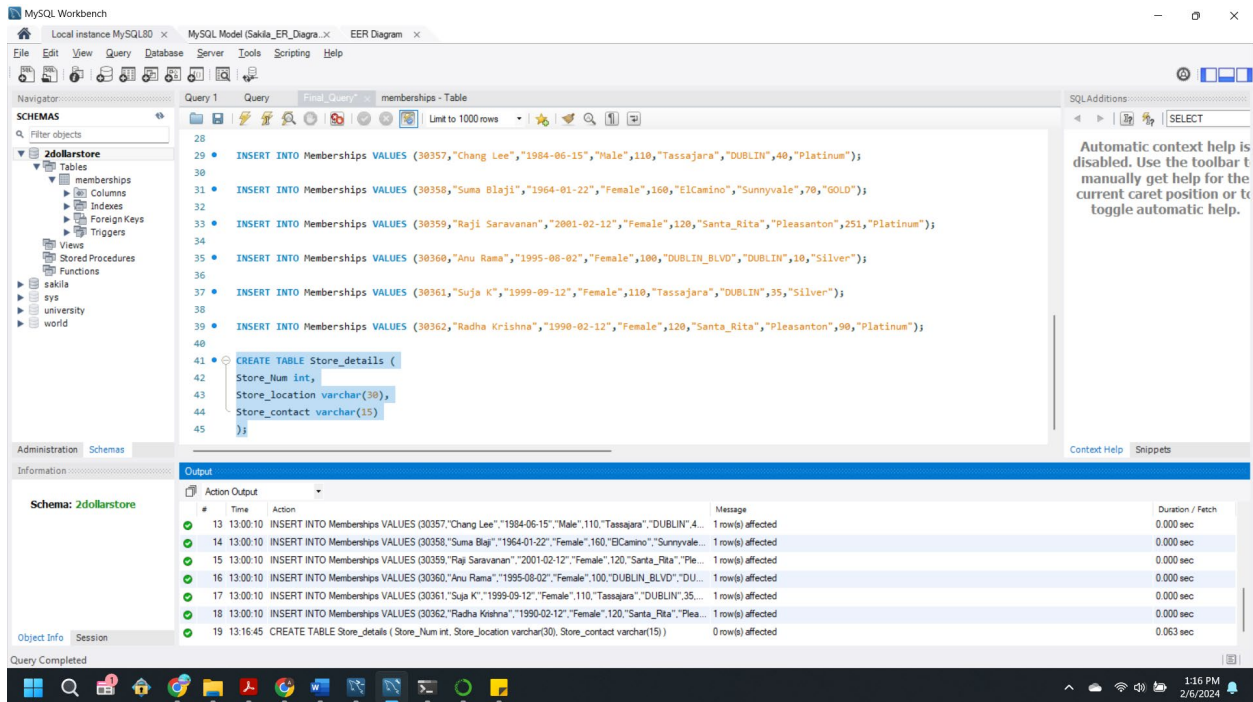
Row_12:

```sql
INSERT INTO Memberships VALUES (30362,"Radha Krishna","1990-02-12","Female",120,"Santa_Rita","Pleasanton",90,"Platinum");
```

Question 4:

Create another table Store_details to save store information with Schema (store_no, store location, store contact)

Screenshot 4:

Code 4:

```
CREATE TABLE Store_details (
Store_Num int,
Store_location varchar(20),
Store_contact varchar(15)
);
```
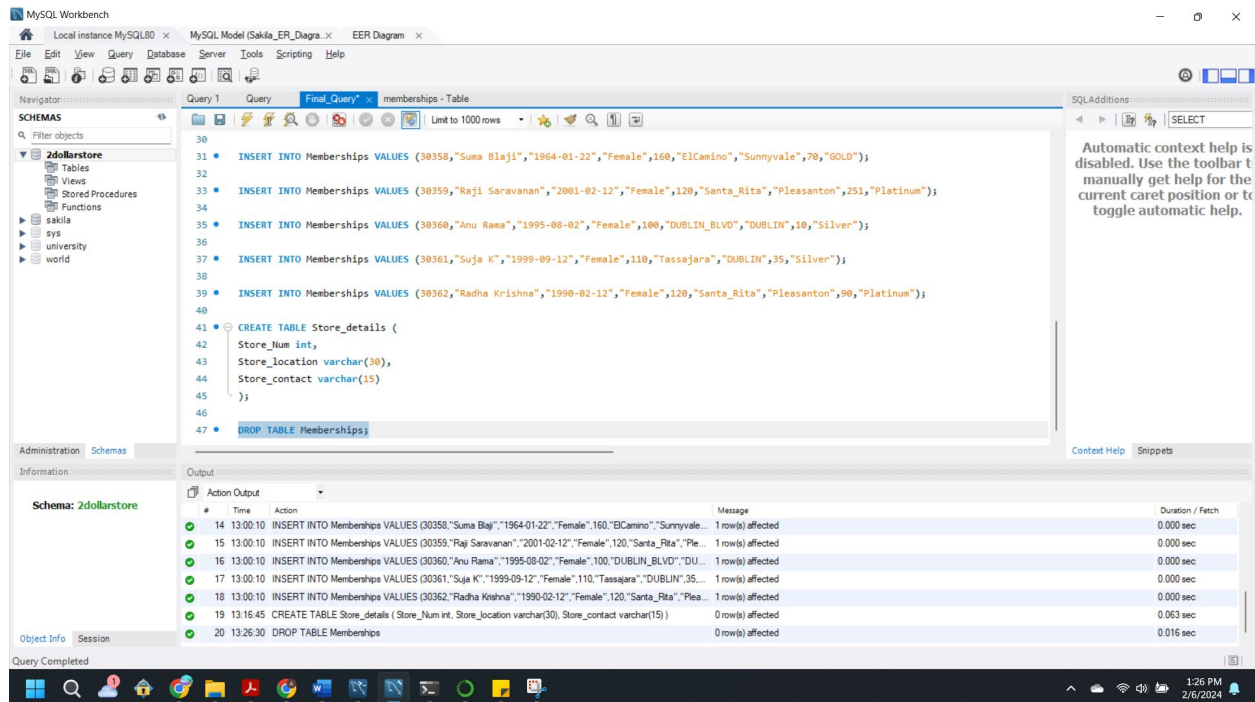
Justification:

1. Store_Num – Store number will generally have whole number, so I chose int data type for it.
2. Store_location- Store_location will have string values of different length, so I chose varchar.
3. Store_contact- Contact number may have numbers, country code and hyphens, so I chose varchar.

Question 5:

Drop table memberships table.

Screenshot 5:

Code 5:

DROP TABLE Memberships;

Question 6:

Which one's better approach?
-> Data Deletion by dropping the relation or Data Archival by renaming the relation.
Provide your justification for each of the above specified approaches.

Answer 6:

Both Data Deletion and Data Archival have their pros and cons depending on the situation.

It is better to archive the data instead of deletion when:

- We need to recover the data when we require it in the future.
- We need to use the stored historical data for analytics or reporting purposes.
- We want a flexible option to either delete or analyze the archived data in the future, but data deletion is irreversible.

Data Deletion is better than Data Archival when:

- We need more storage space, because dropping the table will immediately release the storage space.
- We need less data maintenance, as it becomes simple to maintain and manage less data.
- We need not protect the sensitive data anymore as it will be gone forever when it's deleted, and it reduces the security risks.
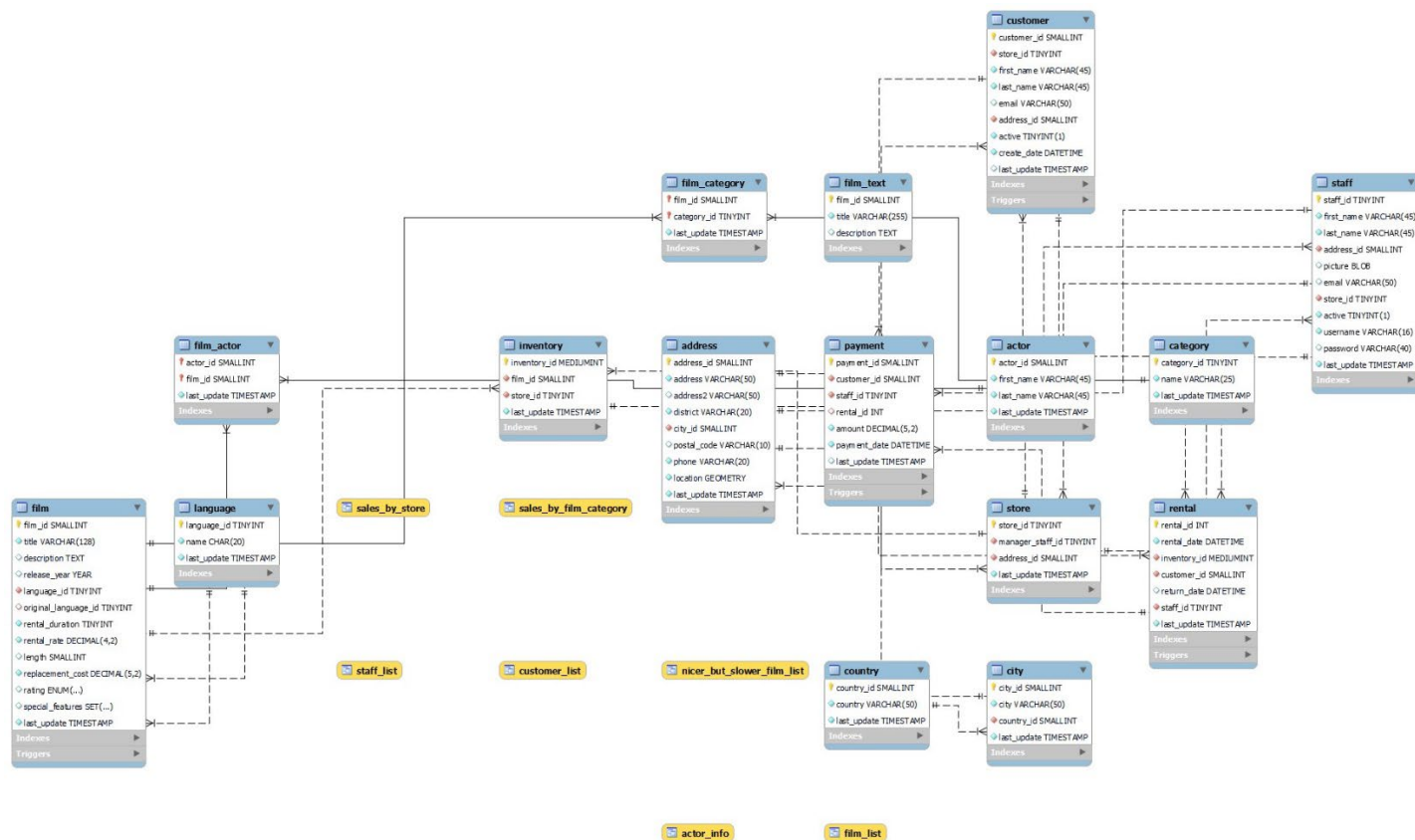
Question 7:

Load the 'sakila' database. Explore the workbench. Paste the ER diagram. (Reverse Engineering)

Screenshot 7:

ER Diagram (Crow's foot notation):

The Sakila database has 16 tables and 7 views.

Question 8:

Describe the potential ethical and privacy considerations when working with employee data in a database. How would you ensure that the database complies with data protection regulations like GDPR or HIPAA?

Answer 8:

- Protect data from unauthorized access. Review the audit logs regularly to monitor unauthorized activities.
- Collect only the necessary data from the employees.
- Employees should be informed about the data collected, how it will be used and who will use it.
- Sensitive data should be encrypted.
- Review and update data protection policies regularly.
- Do not store the data for a long period if it's not necessary.
- Perform a thorough risk assessment to identify potential risks to the security and privacy of health information (covered by HIPAA) and personal information (covered by GDPR).
- Train the employees regularly in data protection policies and regulations.