Step 3 : Project Proposal, Outline, ERD

Final version - Design HTML Interface + DML SQL

CS 340 Group 126

San Davran, Young-Joon Park

URL: https://web.engr.oregonstate.edu/~parky8/index.html

Feedback by the peer reviewers draft 3:

Notes left by team members are written in blue.
Changes to be applied have been changed to red.

- Review by Yash Sankanagouda

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

Yes, the UI utilizes a SELECT for every table in the schema (Employees, Customers, Movies, Actors, Order Items, Orders, Performances). The tables are clearly presented, one thing I would add is an add button which creates a new row where new information could be added instead of having empty rows saying None.

- We decided since we are migrating to flask where we will eventually populate the lists dynamically, to leave this be.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

Doesn't appear that the website has a SELECT utilizing a search/filter but there are two SELECTs using it in the DML.sql. Ensure that the website has the implementation as well!

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

Every table does not have an insert. I see that Employees and Customers have field with None where new information can be inserted.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

Yes, each INSERT seems to add the corresponding FK attributes, including at least one M:M relationship.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

Yes, the DELETE removes things from the M:M relationship (Movies and Actors). All the tables alos have a DELETE features and removes items from thte table. I noticed that your database outline doesn't talk about the M:M relationship but it is seen in the ERD.

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Yes, UPDATE is available for all the entities. It is an edit button used to update the information in the tables.

Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

In the ERD, I see that there are a couple optional relationships. However, on the website I don't see an example of it. The following entities are NULLable (optional): Customers and Orders, Employees and Orders, Movies and Orders/Order Items.

Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.

I think it is looking good so far. I would change the column headers to be more specific such as customer_id can be Customer ID or first can be First Name. I would also add a new row only when new information is provided rather than having a bunch of empty rows. Regardless, the databse is looking good, goodluck!

- # Review by Ismael Khalique

· ***Does the UI utilize a SELECT for every table in the schema?*** **In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.**

Yes, the UI displays data from each table in the schema using a SELECT.

· ***Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?***

Yes, there is a search filter in the Order_items DML.sql.

· ***Does the UI implement an INSERT for every table in the schema?*** **In other words, there should be UI input fields that correspond to each table and attribute in that table.**

Yes, the UI has a “New” field on the tables that implement INSERT.

· ***Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?*** **In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).**

Yes, each INSERT also adds corresponding FK attributes.

· ***Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?*** **In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.**

Yes, there is a DELETE for different tables including a DELETE for the M:M relationship between Movies, Actors.

· ***Is there at least one UPDATE for any one entity?*** **In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?**

Yes, there is at least one UPDATE or edit for multiple tables.

· ***Is at least one relationship NULLable?*** **In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.**

The Performances and Actors relationship appears to be Nullable.

· ***Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.***

I think you can make the "New" option to add a new value into the table on the UI standout a little more.

● Review by Ryan Kirkpatrick

- *Does the UI utilize a SELECT for every table in the schema?* In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.
    - Yes every table in the schema has a SELECT utilized by the UI
- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
    - Appears that all the SELECT do not utilize a search/filter, they just select all
- *Does the UI implement an INSERT for every table in the schema?* In other words, there should be UI input fields that correspond to each table and attribute in that table.
    - Yes, the UI implements an INSERT for every table in the schema
- *Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?* In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).
    - Yes, each INSERT adds corresponding FK attributes
- *Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the

corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
    - Yes, there is a DELETE and a DELETE can remove things from the M:M relationship.
- *Is there at least one UPDATE for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
    - Yes, there is an UPDATE
- *Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.
    - Yes, employes on an order is NULLable
- *Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.*
    - This is looking great! Keep up the good work. Just make sure to implement the select with the search/filter.

Review by Ashley Pak

Does the UI utilize a SELECT for every table in the schema? **In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.**

- Yes, every table in the schema is displayed.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

- On the website, a dynamic search/filter is not seen, but the DML sql does have drop down selects for two tables. So, technically, yes!

Does the UI implement an INSERT for every table in the schema? **In other words, there should be UI input fields that correspond to each table and attribute in that table.**

- The website does not have INSERT queries but the DML does.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? **In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).**

- The FKs and M:M relationship all seem to be handled correctly.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? **In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.**

- Yes, each entity has a DELETE and the M:M does as well! It also has the correct cascading sql to ensure no anomalies occur.

Is there at least one UPDATE for any one entity? **In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?**

- Yes, there is more than one UPDATE so the requirement is filled!

Is at least one relationship NULLable? **In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.**

- I checked the outline and the sql files, but most attributes, excluding the FKs, are currently set to "not NULL". However, on the ERD, Employees/Orders, Movies/Orders, Customers/Orders all have an optional relationship.

Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.

- I think the team is off to great start! I think your website was the first one I saw with a photo on the homepage!
- Some minor design suggestions could be to organize the data in the tables so that they're in the center of each cell, rather than on the left/right side. Or choose one of the sides and stick with it for the rest of the tables.
    - For example, the Performances table, performance_id and actor_id, have the numbers adhered to the right while movie_id is adhered to the left.
- The Employees table on the UI could be more specific and have "first name" instead of just "first" - same for "last" as well.

Very interesting concept and good work! Good luck with the rest of the project!

Great work group 126! Remember to include a search function as part of the projects requirements

- TA, Katie Russell

Actions based on previous feedback:

1. Added short description to entities to clarify what they are

2. Changed entity names in the description section to plural

3. Change entity names in the ERD to plural

4. order_id and movie_id changed to FK in Order_item

5. Added more attribute candidates under Movies entity (will consider implementing as the project develops)

6. Added more backstory to clearly explain the goal of the database

7. Added more estimates and forecast for the growth of Gorak's business that the database must support

8. Dropped prefixes in attribute names

9. The composite key issue that was pointed out by the TA feedback has been worked out. Composite key kept

10. ERD diagram inconsistency pointed out by the TA corrected

11. Deleted ship_or_pickup attribute in Orders entity

12. Replaced screenshot of ERD

13. Added a detailed schema

14. Updated Movies.price from int to Decimal (16,2)

15. Updated ON DELETE CASCADE to select foreign keys (in Order_items, Performances)

16. Updated ERD to be consistent with outline

Actions and upgrades based on previous feedback:

1. Added bootstrap to code

2. Changed homepage layout

3. Added a SELECT that utilizes a search/filter to dynamically populate a dropdown on ORDERS new and edit

4. Started preparation to migrate to Flask

5. Added break lines to CRUD pages

6. Table CSS changed to make href stand out a little more

7. Updated foreign key not NULL to exclude Orders - employee_id

8. Update DDL.sql to reflect documentation updates

PROJECT OUTLINE

A) Overview

Gorak is an entrepreneurial Gorkian, an interstellar species faring from Gorkia Prime. In Gorkia, human movies from Earth have recently grown in popularity among Gorkian high society. With figures like Gelon Gusk collecting primitive blu-rays containing human cinema, Gorak has made a company to profit from the trend.

As part of the Gorkian policy of xeno-preservation, Gorkians are strictly prohibited when it comes to directly contacting humans in their original form. Therefore, specially trained Gorkian agents have infiltrated human society disguised in human form. The only direct contact permitted between Gorkians and humans is to be done in disguise until the Gorkian congress passes first contact protocols. However, the Gorkian senate is in political deadlock, and Gorak expects that passing the protocols will take a long time.

Thus, Gorak made the following business model:

Gorak's store wants to build a galaxy-website where a potential customer can look up the title, price, stock, and actors in a given human movie (the database currently has around 200 most popular movies, but is growing every week). If the customer wants to buy one or more blu-ray discs of one or more movies, she can order the products online by inputting her galactic-credit card number, products, and quantities. Once the order is received, one Gorkian infiltrator agent employee is assigned to that order. The order is then fulfilled by this agent and is shipped back to the address of a customer in Gorkia. Because of the expenses involved in transporting the original blu-ray back to Gorkia Prime, Gorak estimates that his customer base will number in the 10,000s.

<Problem to solve and numerical facts, entities noted in parentheses>

Gorak's first step is to design a database that can support his website and operations. Gorak envisions facilitating order (Orders and Order_items) fulfillment by using his database to let customers (Customers) pick an available movie (Movies) and look at the general information of the movie (Actors, Performances, Movies) from his site, submit an order, and assign that order to an employee (Employees, who is a Gorkian infiltrator). The database also has to support the growth of Gorak's business. Gorak expects that the website will support up to 10,000 movies and 100,000 customers by the end of the year.

B) Database Outline:

Customers:

Description: a customer who put in an order to Gorak's business. The customer is located in Gorkia prime and is using Gorak's galactic-website to put in orders for Gorak's business to fulfill.

- Attributes:
    - id : int, auto_increment, not NULL, PK;
    - first : VARCHAR(50), not NULL;
    - last : VARCHAR(50), not NULL;
    - street : VARCHAR(50), not NULL;
    - city : VARCHAR(50), not NULL;
    - state : VARCHAR(50), not NULL;
    - phone : VARCHAR(15), not NULL;
- Relationships:
    - M : 1 relationship between Customer and Order (FK customer_id in Order)

Employees:

Description: an Gorkian infiltrator hired by Gorak to procure ordered blu-ray discs. One employee is assigned to one order from one customer and is in charge of filling out that order.

- Attributes:
    - id : int, auto_increment, not NULL, PK;
    - first : VARCHAR(50), not NULL;
    - last : VARCHAR(50), not NULL;
    - street : VARCHAR(50), not NULL;
    - city : VARCHAR(50), not NULL;
    - state : VARCHAR(50), not NULL;
    - phone : VARCHAR(50), not NULL;
- Relationships:
    - M : 1 relationship between Employee and Order (FK employee_id in Order)

Orders:

Description: an order that the customer has filled out. The junction table Order_items indicate what and how many items an order contains. Employee is nullable indicating that no employee has been assigned yet.

- Attributes:

        - order_id : int, auto_increment, not NULL, PK;
        - customer_id : int, not NULL, FK
        - employee_id : int, FK
        - order_date : DATETIME, not NULL;
        - credit_card_num : VARCHAR(16), not NULL;
        - order_fulfilled : BIT, not NULL; – this is a BOOL value
- Relationships:
    - 1 : M relationship between Order and Customer (FK customer_id in Order)
    - 1 : M relationship between Order and Employee (FK employee_id in Order)
    - M : 1 relationship between Order and Order_item (FK order_id in Order_item)

Order_items:

Description: linked to an order, an order item contains one movie and the quantity of that movie that has been ordered. It is linked to an order_id, allowing customers to buy multiple movies and multiple copies of the same movie. **Junction table between Movies and Orders**.

- Attributes:
    - order_id : int, auto_increment, not NULL, PK, FK;
        - Composite key
    - movie_id : int, auto_increment, not NULL, PK, FK;
        - Composite key
    - quantity : int;
- Relationships:
    - 1 : M relationship between Order_item and Order (FK order_id in Order_item)
    - 1 : M relationship between Order_item and Movie (FK movie_id in Order_item)
- Deletion of order_id and movie_id cascades to delete Order_items entry

Movies:

Description: a human movie that Gorak's business offers on the galactic-website for his store. Xeno-preservists in the Gorkian congress love the Avatar series.

- Attributes:
    - movie_id : int, auto_increment, not NULL, PK;
    - title : VARCHAR(100), not NULL;
    - stock : int, not NULL;
    - price : DECIMAL(16, 2), not NULL;

- Possible additional attributes:
    - main_genre: VARCHAR(32)
    - ratings: int;
- Relationships:
    - M : 1 relationship between Movie and Order_item (FK movie_id in Order_item)
    - M : 1 relationship between Movie and Performance (FK movie_id in Performance)

Performances:

Description: a junction table indicating a performance of an actor in a movie. The entity is there to get rid of the M:M relationship between Movies and Actors. **Junction table between Movies and Actors**.

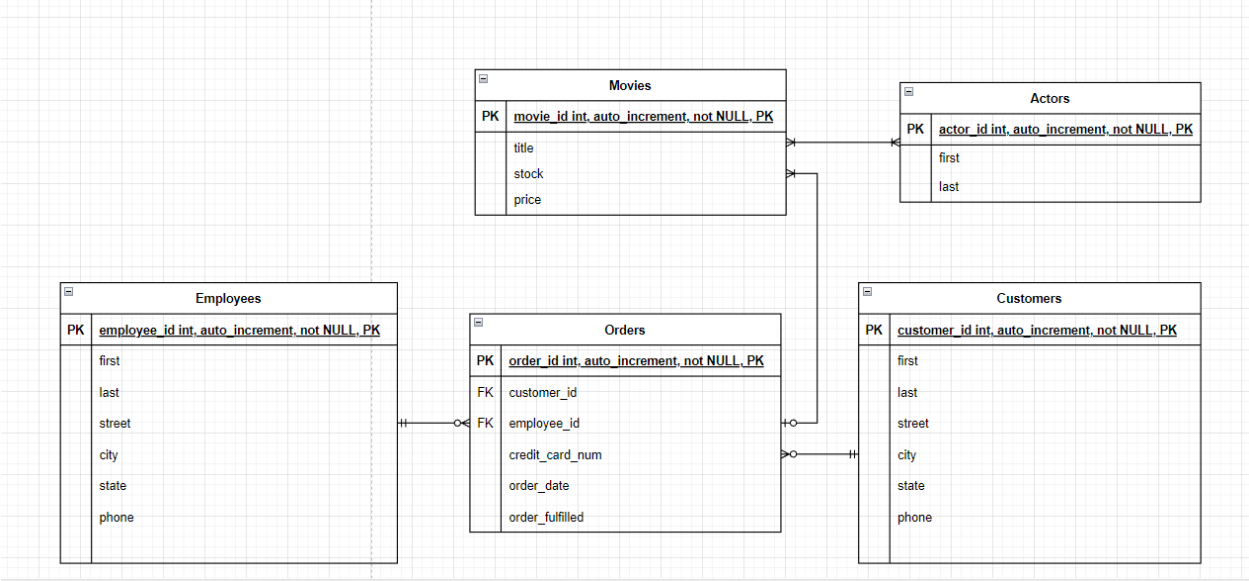
- Attributes:
    - performance_id : int, auto_increment, not NULL, PK;
    - movie_id : int, not NULL, FK
    - actor_id : int, not NULL, FK
- Relationships:
    - 1 : M relationship between Performance and Movie (FK movie_id in Performance)
    - 1 : M relationship between Performance and Actor (FK actor_id in Performance)
- Deletion of movie_id and actor_id cascades to delete Performances entry

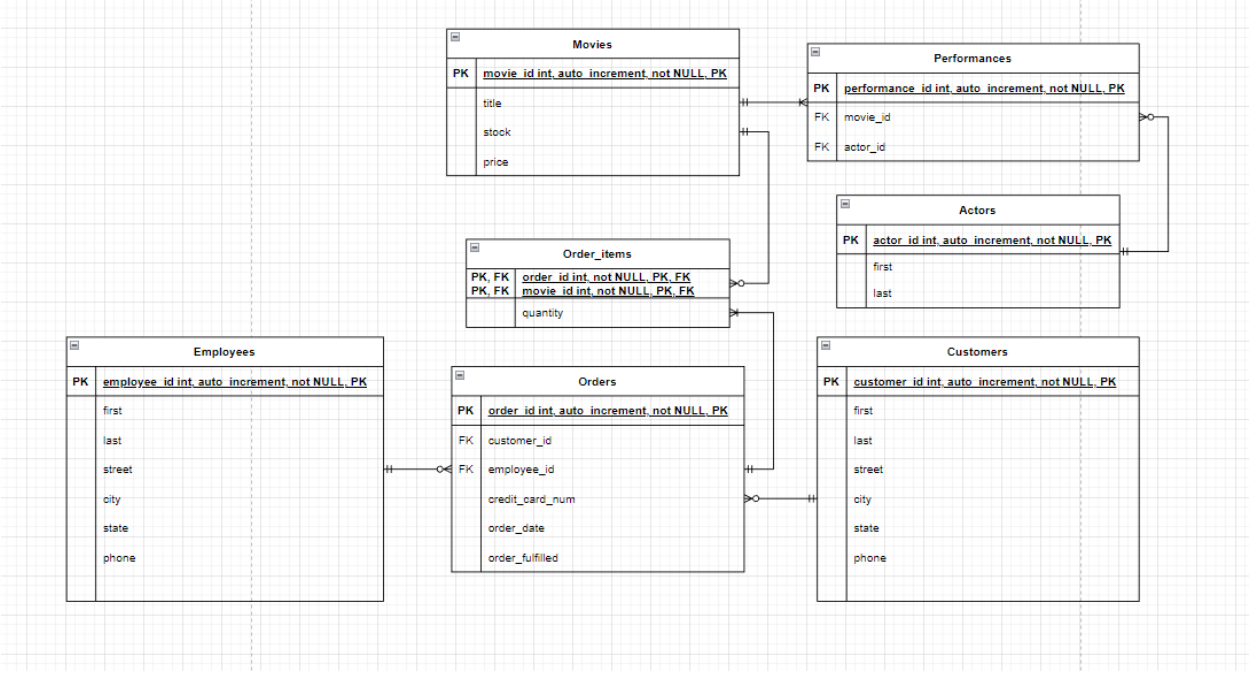
Actors:

Description: an Earther actor who performs in a movie. There are rumors that a famous Hollywood actor is a Gorkian agent.

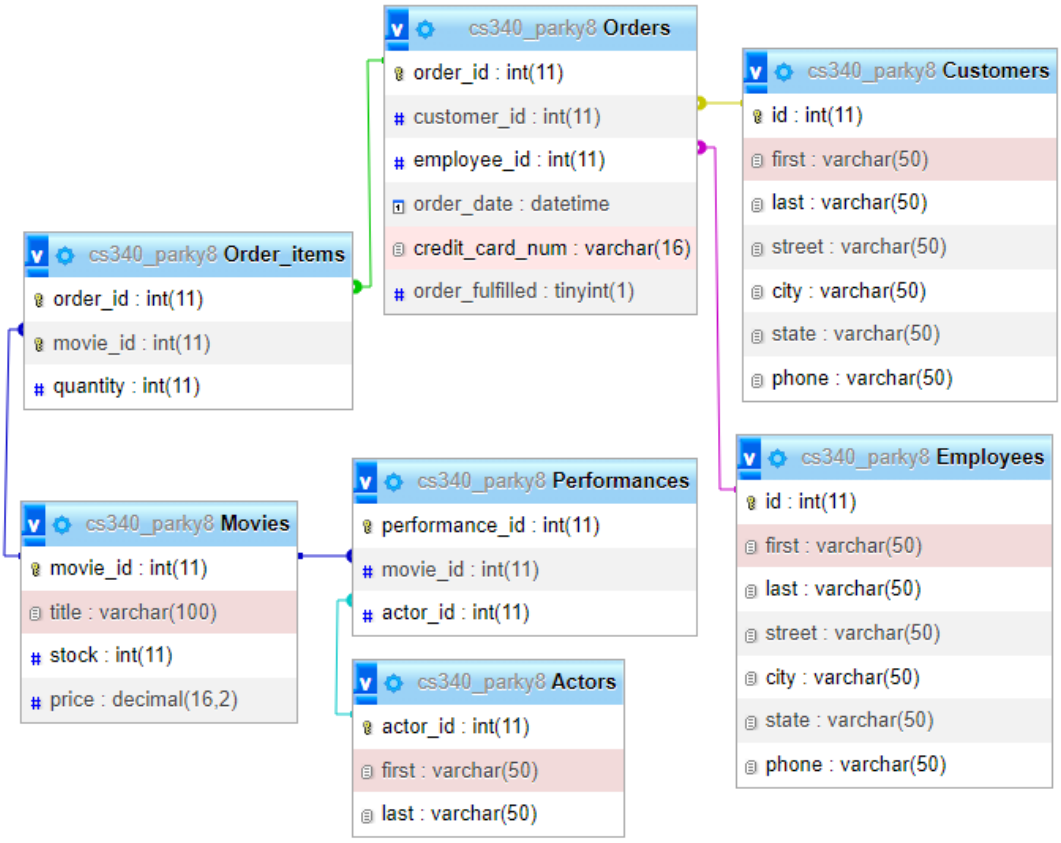
- Attributes:
    - actor_id : int, auto_increment, not NULL, PK;
    - first : VARCHAR(50), not NULL;
    - last : VARCHAR(50), not NULL;
- Relationships:
    - M : 1 relationship between Actor and Performance (FK actor_id in Performance)

C) Entity-Relationship Diagram, Resolution of M to N Relationship:





D) Schema:



E) Sample Data:

```
MariaDB [cs340_parky8]> select * from Customers;
+----+-------+--------+------------------+---------------+-------+------------+
| id | first | last   | street           | city          | state | phone      |
+----+-------+--------+------------------+---------------+-------+------------+
|  1 | YJ    | Park   | 2415 Chestnut St | Philadelphia  | PA    | 1234567890 |
|  2 | San   | Davran | 1231 Maple St    | New York City | NY    | 0987654321 |
|  3 | Tupac | Shakur | 9807 Oak St      | Los Angeles   | CA    | 3216540987 |
+----+-------+--------+------------------+---------------+-------+------------+
3 rows in set (0.000 sec)
```

Customers						
id	first	last	street	city	state	phone
1	YJ	Park	2415 Chestnut	Philadelphia	PA	1234567890

			St			
2	San	Davran	1231 Maple St	New York City	NY	0987654321
3	Tupac	Shakur	9807 Oak St	Los Angeles	CA	3216540987

```
MariaDB [cs340_parky8]> select * from Employees;
+----+--------+---------+------------------+------------+-------+------------+
| id | first  | last    | street           | city       | state | phone      |
+----+--------+---------+------------------+------------+-------+------------+
|  1 | Erling | Haaland | 3211 Oakview St. | Manchester | MA    | 1111111111 |
|  2 | Harry  | Kane    | 3453 Carnaby St  | London     | TN    | 2222222222 |
|  3 | Darwin | Nunez   | 1315 Bristol St  | Liverpool  | UK    | 3333333333 |
+----+--------+---------+------------------+------------+-------+------------+
3 rows in set (0.000 sec)
```

Employees						
id	first	last	street	city	state	phone
1	Erling	Haaland	3211 Oakview St	Manchester	MA	1111111111
2	Harry	Kane	3453 Carnaby St	London	TN	2222222222
3	Darwin	Nunez	9807 Oak St	Los Angeles	CA	3333333333

```
MariaDB [cs340_parky8]> select * from Orders;
+----------+-------------+-------------+---------------------+------------------+-----------------+
| order_id | customer_id | employee_id | order_date          | credit_card_num  | order_fulfilled |
+----------+-------------+-------------+---------------------+------------------+-----------------+
|        1 |           1 |           2 | 2022-01-11 00:00:00 | 1111222233334444 |               0 |
|        2 |           2 |           1 | 2022-01-12 00:00:00 | 2222333344445555 |               1 |
|        3 |           3 |           2 | 2022-01-01 00:00:00 | 3333444455556666 |               1 |
|        4 |           2 |           3 | 2022-01-04 00:00:00 | 4444555566667777 |               0 |
|        5 |           1 |           2 | 2022-01-04 00:00:00 | 5555666677778888 |               0 |
+----------+-------------+-------------+---------------------+------------------+-----------------+
5 rows in set (0.001 sec)
```

Orders					
order_id	customer_id	employee_id	order_date	credit_card_num	order_fulfilled
1	1	2	2022-01-11	1111222233334444	0
2	1	1	2022-01-12	2222333344445555	1
3	2	2	2022-01-01	3333444455556666	1

4	3	3	2022-01-04	4444555566667777	0
5	2	2	2022-01-04	5555666677778888	0

```
MariaDB [cs340_parky8]> select * from Order_items;
+----------+----------+----------+
| order_id | movie_id | quantity |
+----------+----------+----------+
|        1 |        2 |        2 |
|        1 |        3 |        4 |
|        2 |        1 |        8 |
|        3 |        2 |        5 |
|        4 |        1 |        1 |
|        5 |        2 |        3 |
+----------+----------+----------+
6 rows in set (0.000 sec)
```

Order_items		
order_id	movie_id	quantity
1	3	4
1	2	2
2	1	8
3	2	5
4	1	1
5	2	3

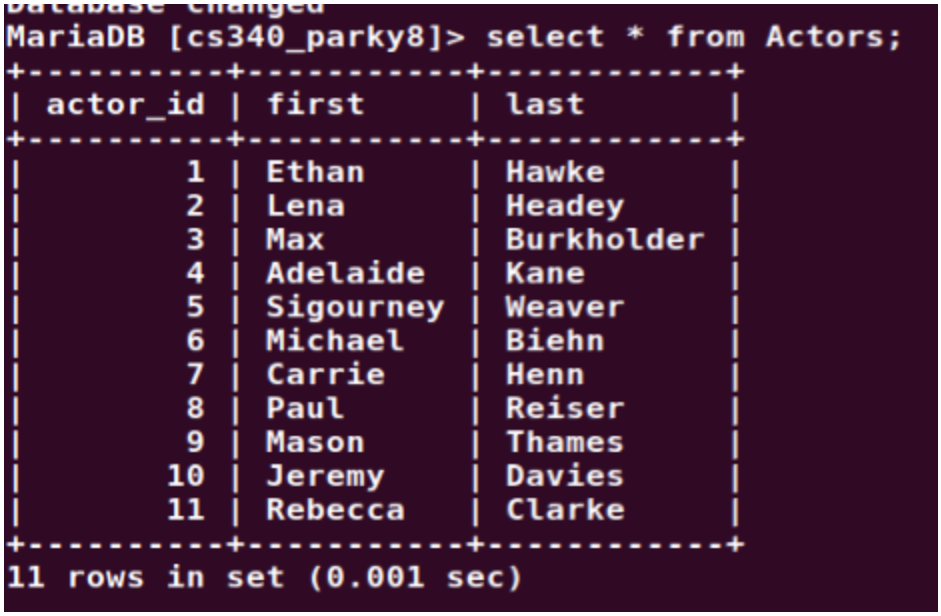
```
MariaDB [cs340_parky8]> select * from Movies;
+----------+-----------------+-------+-------+
| movie_id | title           | stock | price |
+----------+-----------------+-------+-------+
|        1 | Aliens          |     3 | 12.99 |
|        2 | The Purge       |    13 |  2.99 |
|        3 | The Black Phone |    21 |  5.99 |
+----------+-----------------+-------+-------+
3 rows in set (0.000 sec)
```

Movies			
movie_id	title	stock	price
1	Aliens	3	12.99

2	The Purge	13	2.99
3	The Black Phone	21	5.99

```
MariaDB [cs340_parky8]> select * from Performances;
+----------------+----------+----------+
| performance_id | movie_id | actor_id |
+----------------+----------+----------+
|              1 |        2 |        1 |
|              2 |        2 |        2 |
|              3 |        2 |        3 |
|              4 |        2 |        4 |
|              5 |        1 |        5 |
|              6 |        1 |        6 |
|              7 |        1 |        7 |
|              8 |        1 |        8 |
|              9 |        3 |        9 |
|             10 |        3 |       10 |
|             11 |        3 |       11 |
+----------------+----------+----------+
11 rows in set (0.000 sec)
```

Performances		
performance_id	movie_id	actor_id
1	2	1
2	2	2
3	2	3
4	2	4
5	1	5
6	1	6
7	1	7
8	1	8
9	3	9
10	3	10
11	3	11
12	3	1



```
Database changed
MariaDB [cs340_parky8]> select * from Actors;
+----------+-----------+------------+
| actor_id | first     | last       |
+----------+-----------+------------+
|        1 | Ethan     | Hawke      |
|        2 | Lena      | Headey     |
|        3 | Max       | Burkholder |
|        4 | Adelaide  | Kane       |
|        5 | Sigourney | Weaver     |
|        6 | Michael   | Biehn      |
|        7 | Carrie    | Henn       |
|        8 | Paul      | Reiser     |
|        9 | Mason     | Thames     |
|       10 | Jeremy    | Davies     |
|       11 | Rebecca   | Clarke     |
+----------+-----------+------------+
11 rows in set (0.001 sec)
```

Actors		
actor_id	first	last
1	Ethan	Hawke
2	Lena	Headey
3	Max	Burkholder
4	Adelaide	Kane
5	Sigourney	Weaver
6	Michael	Biehn
7	Carrie	Henn
8	Paul	Reiser
9	Mason	Thames
10	Jeremy	Davies
11	Rebecca	Clarke