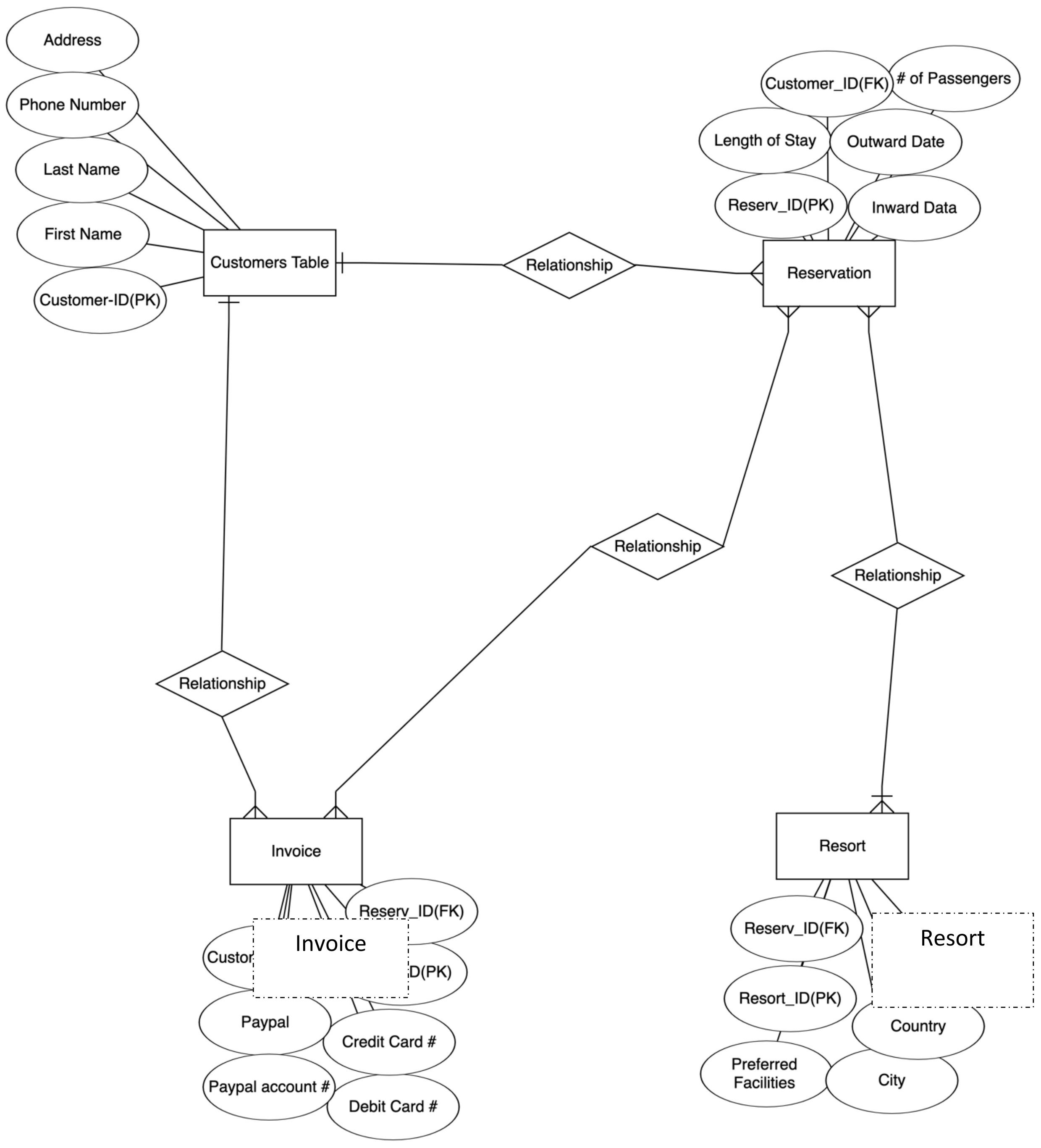
Amin Baabol
INFS 762
Assignment 2

Task 1



Task 2

Strong Entities

1. TVSeries (TVSeriesID (PK), Name, Network)
2. Actor (ActorID (PK), Name, Email, Phone, Address)
3. Character (ID(PK), Names, TVSeriesID (FK), ActorID (PK))
4. Writer (WriterID, Name, DOB, Agency)

Weak Entities

1. Season (SeasonCode (PK), TVSeriesID (FK), StartDate, EndDate)
2. Episode (EpisodeNum (PK), SeasonCode (FK), TVSeriesID (FK), WriterID (FK), ID(FK), Title, Length)

Task 3

For the following queries, you just need to submit your SQL statements, and don't need to submit the outputs.

1. Write the SQL command to change the movie year for movie number 1245 to 2008.

```
UPDATE MOVIE
SET MOVIE_YEAR = 2008
WHERE MOVIE_NUM = 1245;
```

2. Write a single SQL command to increase all price rental fee values by $0.50.

```
UPDATE PRICE
set PRICE_RENTFEE     = PRICE_RENTFEE       + 0.50
```

3. Write a query to display the movie year, movie title, and movie cost sorted by movie year in descending order

```
SELECT MOVIE_YEAR, MOVIE_TITLE, MOVIE_COST
FROM MOVIE
ORDER BY MOVIE_COST DESC;
```

4. Write a query to display the movie title, movie year, and movie genre for all movies

```
SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_GENRE
FROM MOVIE
```

5. Write a query to display the movie title, movie year, and movie genre for all movies sorted by movie genre in ascending order, then sorted by movie year in descending order within genre

```
SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_GENRE
FROM MOVIE
ORDER BY MOVIE_GENRE, MOVIE_YEAR DESC;
```

6. Write a query to display the movie number, movie title, and price code for all movies with a title that starts with the letter "R"

```
SELECT MOVIE_NUM, MOVIE_TITLE, PRICE_CODE
FROM MOVIE
WHERE MOVIE_TITLE like 'R%';
```

7. Write a query to display the movie number, movie title, and movie cost for all movies with a cost greater than $40

```
SELECT MOVIE_NUM, MOVIE_TITLE, MOVIE_COST
FROM MOVIE
WHERE MOVIE_COST > 40;
```

8. Write a query to display the movie number, movie title, movie cost, and movie genre for all movies that are either action or comedy movies and have a cost that is less than $50. Sort the results in ascending order by genre. (*Reminder*: because the default order of operations for logical connectors is to evaluate all of the ANDs, then evaluate all of the ORs, it is necessary to either use the IN operator or use parentheses to have the OR evaluated first).

```
SELECT MOVIE_NUM, MOVIE_TITLE, MOVIE_COST, MOVIE_GENRE
FROM MOVIE
WHERE MOVIE_GENRE IN ('ACTION', 'COMEDY') AND MOVIE_COST < 50
ORDER BY MOVIE_GENRE ASC;
```

9. Write a query to display the movie genre and the number of movies in each genre.

```
SELECT MOVIE_GENRE, COUNT (*) AS "Number of Movies"
FROM MOVIE
GROUP BY MOVIE_GENRE;
```

10. Write a query to display the movie title, movie genre, price description, and price rental fee for all movies with a price code

```
SELECT MOVIE_TITLE, MOVIE_GENRE, PRICE_DESCRIPTION, PRICE_RENTFEE
FROM MOVIE, PRICE
WHERE MOVIE.PRICE_CODE = PRICE.PRICE_CODE;
```

11. Write a query to display the movie genre and average cost of movies in each genre

```
SELECT MOVIE_GENRE, AVG(MOVIE_COST) AS "Average Cost"
FROM MOVIE
GROUP BY MOVIE_GENRE;
```

12. Write a query to display the movie genre and average price rental fee for movies in each genre that have a price

```
SELECT MOVIE_GENRE, AVG (PRICE_RENTFEE) AS " AVERAGE PRICE"
FROM MOVIE, PRICE
WHERE MOVIE.PRICE_CODE = PRICE.PRICE_CODE
```

13. Write a query to display the movie title, movie year, and the movie cost divided by the price

rental fee for each movie that has a price to determine the number of rentals it will take to break even on the purchase of the movie

```
SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_COST/PRICE_RENTFEE
FROM MOVIE, PRICE
WHERE MOVIE.PRICE_CODE=PRICE.PRICE_CODE
```

14. Write a query to display the movie title, movie year, and movie cost for all movies that have a cost between $44.99 and $49.99

```
SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_COST
FROM MOVIE
WHERE MOVIE_COST BETWEEN 49.99 AND 49.99
```

15. Write a query to display the membership number, first name, last name, and balance of members that have a rental

```
SELECT MEM_NUM, MEM_FNAME, MEM_LNAME, MEM_BALANCE
FROM MEMBERSHIP
WHERE MEM_NUM IN (SELECT MEM_NUM FROM RENTAL);
```

16. Write a query to display the minimum balance, maximum balance, and average balance for members that have a rental

```
SELECT MIN(MEM_BALANCE) AS "MINIMUM”, MAX(MEM_BALANCE) AS "MAXIMUM”,
AVG(MEM_BALANCE) AS "AVERAGE"
FROM MEMBERSHIP
WHERE MEM_NUM IN (SELECT MEM_NUM FROM RENTAL);
```

17. Write a query to display the movie title, movie year, price description, and price rental fee for all movies that are in the genres Family, Comedy, or Drama

```
SELECT MOVIE_TITLE, MOVIE_YEAR, PRICE_DESCRIPTION, PRICE_RENTFEE,
MOVIE_GENRE
FROM MOVIE, PRICE
WHERE MOVIE.PRICE_CODE = PRICE.PRICE_CODE
AND MOVIE_GENRE IN ('FAMILY', 'COMEDY', 'DRAMA');
```

18. Write a query to display the membership number, last name, and total rental fees earned from each member. The total rental fee is the sum of all the detail fees (without the late fees) from all movies that the member has rented.

```
SELECT MEMBERSHIP.MEM_NUM, MEM_LNAME, MEM_FNAME,
SUM (DETAILRENTAL.DETAIL_FEE) AS "TOTAL RENTAL FEE"
FROM MEMBERSHIP, RENTAL, DETAILRENTAL
WHERE MEMBERSHIP.MEM_NUM = RENTAL.MEM_NUM
AND RENTAL.RENT_NUM = DETAILRENTAL.RENT_NUM
GROUP BY MEMBERSHIP.MEM_NUM, MEM_LNAME, MEM_FNAME;
```