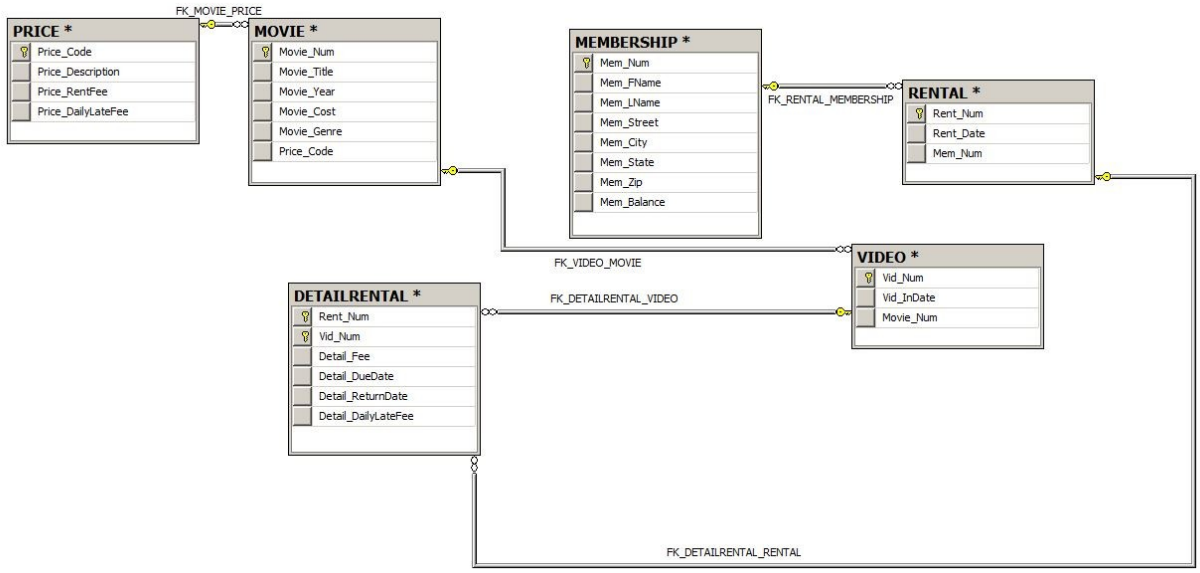
Part 1: Upload the given Access database into your SQL Server account. Create all the PKs and the relationships. Screen capture the ERD in SQL Server and save it as A7.jpeg.

Finish Exercises 103 through 122 on pages 334-338. Save the SQL statements in the file A7.sql.

You must submit the following: A7.jpeg and A7.sql before the due date and time on Blackboard. Also submit a stapled printed copy in class on the due date. A late assignment will be assessed a penalty of 10% of the assigned points per calendar day up to 7 days. After 7 days no late assignment will be accepted.



SQL Statements:

--Query 103

SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_GENRE

FROM MOVIE

--Query 104

SELECT MOVIE_YEAR, MOVIE_TITLE, MOVIE_COST

FROM MOVIE

ORDER BY -MOVIE_YEAR

```
--Query 105
SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_GENRE
FROM MOVIE
ORDER BY MOVIE_GENRE, -MOVIE_YEAR

--Query 106
SELECT MOVIE_NUM, MOVIE_TITLE, PRICE_CODE
FROM MOVIE
WHERE MOVIE_TITLE LIKE 'R%'

--Query 107
SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_COST
FROM MOVIE
WHERE MOVIE_TITLE LIKE '%HOPE%'

--Query 108
SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_GENRE
FROM MOVIE
WHERE MOVIE_GENRE = 'ACTION'

--Query 109
SELECT MOVIE_NUM, MOVIE_TITLE, MOVIE_COST
FROM MOVIE
WHERE MOVIE_COST > 40

--Query 110
SELECT MOVIE_NUM, MOVIE_TITLE, MOVIE_COST, MOVIE_GENRE
FROM MOVIE
WHERE MOVIE_GENRE IN ('ACTION','COMEDY') AND MOVIE_COST < 50
--Query 111
```

```
SELECT MEM_NUM, MEM_FNAME, MEM_LNAME, MEM_STREET, MEM_BALANCE
FROM MEMBERSHIP
WHERE MEM_BALANCE < 5 AND MEM_STREET LIKE '%AVENUE'

--Query 112
SELECT MOVIE_GENRE, COUNT(MOVIE_GENRE)
FROM MOVIE
GROUP BY MOVIE_GENRE

--Query 113
SELECT AVG(MOVIE_COST)
FROM MOVIE

--Query 114
SELECT MOVIE_GENRE, AVG(MOVIE_COST) AS 'AVG PRICE'
FROM MOVIE
GROUP BY MOVIE_GENRE

--Query 115
SELECT M.MOVIE_TITLE, M.MOVIE_GENRE, P.PRICE_DESCRIPTION, P.PRICE_RENTFEE
FROM MOVIE M INNER JOIN PRICE P
ON M.PRICE_CODE = P.PRICE_CODE

--Query 116
SELECT M.MOVIE_GENRE, AVG(P.PRICE_RENTFEE) AS 'AVG RENT FEE'
FROM MOVIE M INNER JOIN PRICE P
ON M.PRICE_CODE = P.PRICE_CODE
GROUP BY MOVIE_GENRE
--Query 117
SELECT M.MOVIE_TITLE, M.MOVIE_COST/P.PRICE_RENTFEE AS 'BREAKEVEN RENTALS'
```

```
FROM MOVIE M INNER JOIN PRICE P
ON M.PRICE_CODE = P.PRICE_CODE

--Query 118
SELECT MOVIE_TITLE, MOVIE_YEAR
FROM MOVIE
WHERE PRICE_CODE IS NOT NULL

--Query 119
SELECT MOVIE_TITLE, MOVIE_GENRE, MOVIE_COST
FROM MOVIE
WHERE MOVIE_COST BETWEEN 44.99 AND 49.99

--Query 120
SELECT M.MOVIE_TITLE, P.PRICE_DESCRIPTION, P.PRICE_RENTFEE, M.MOVIE_GENRE
FROM MOVIE M INNER JOIN PRICE P
ON M.PRICE_CODE = P.PRICE_CODE
WHERE M.MOVIE_GENRE IN ('FAMILY','COMEDY','DRAMA')

--Query 121
SELECT M.MEM_NUM, M.MEM_FNAME,M.MEM_LNAME, M.MEM_BALANCE
FROM MEMBERSHIP M INNER JOIN RENTAL R
ON M.MEM_NUM = R.MEM_NUM




--Query 122 (LAST ONE)
SELECT MIN(M.MEM_BALANCE) AS 'MINIMUM BALANCE', MAX(M.MEM_BALANCE) AS
'MAXIMUM BALANCE', AVG(M.MEM_BALANCE) AS 'AVERAGE BALANCE'
```

```
FROM MEMBERSHIP M INNER JOIN RENTAL R
ON M.MEM_NUM = R.MEM_NUM
```