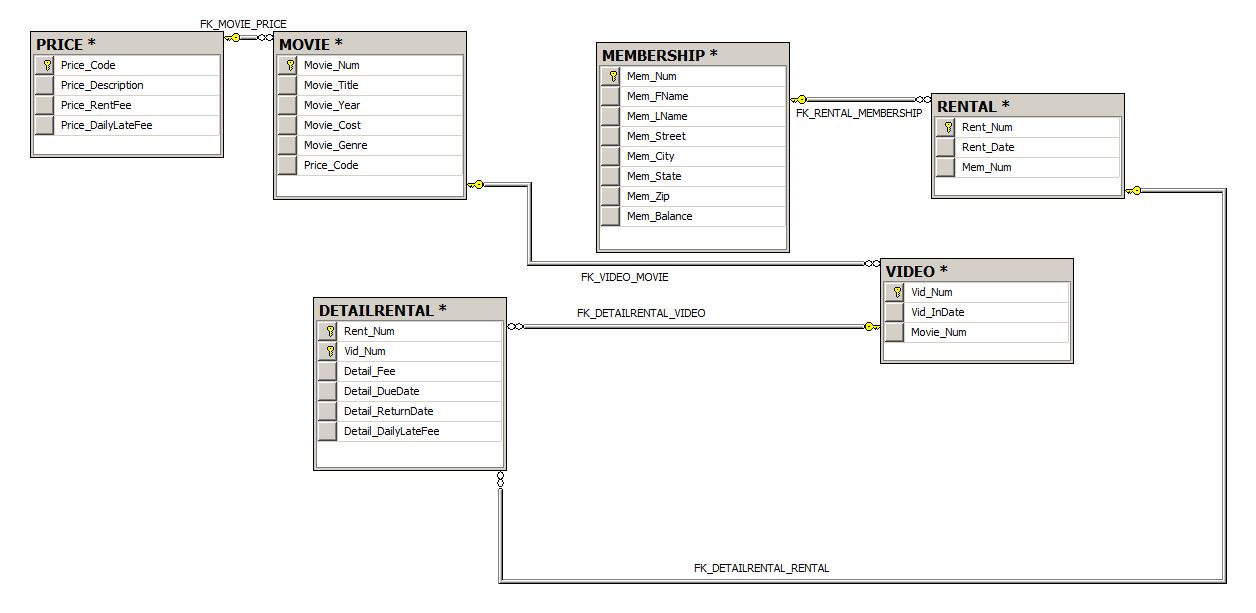
Part 1: Upload the given Access database into your SQL Server account. Create all the PKs and the relationships. Screen capture the ERD in SQL Server and save it as A7.jpeg.

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