COMP 3311: Database Management Systems

Task 2: Table Creation and SQL Queries—Grading Scheme and Example Solution

PART 1: TABLE CREATION

The main requirement for this part is to specify all applicable constraints and to create the tables in the correct order so that the referential integrity constraints can be validated by the DBMS.

Grading Scheme for Part 1

Tables	Create order (marks)	Primary key (marks)	Referential integrity (number)	Constraints			Total Marks
				Not Null (number)	Default (number)	Check/Unique (number)	
AcademyAward	0.3	0.1	0	1	0	2	3.5
AwardWin	0.3	0.2	3	0	0	0	3.5
CastMemberAppearsIn	0.3	0.2	2	1	0	0	2.6
DirectorOf	0.3	0.2	2	0	0	0	2.5
Movie	0.3	0.1	1	6	1	3	6.6
MovieGenre	0.3	0.2	1	1	0	0	1.6
MoviePerson	0.3	0.1	0	3	0	1	2.2
ReelflicsMember	0.3	0.1	0	15	0	11	18.4
Review	0.3	0.2	2	4	0	1	4.4
WatchHistory	0.3	0.2	2	1	0	0	2.6
Watchlist	0.3	0.2	2	0	0	0	2.5
						Max marks	50

For create order, *marks are allocated for each table* as indicated in the grading scheme. The create order is: AcademyAward before {AwardWin, Movie}; Movie before {AwardWin, CastMemberAppearsIn, DirectorOf, MovieGenre, Review, WatchHistory, Watchlist}; MoviePerson before {AwardWin, CastMemberAppearsIn, DirectorOf}; ReelflicsMember before {Review, WatchHistory, Watchlist}.

For the primary key mark, the grading scheme lists *the marks for the key* in each table.

For the referential integrity mark, the grading scheme lists *the number of referential integrity constraints* in each table. Each referential integrity constraint is worth 1 mark.

For the constraints mark, the grading scheme lists *the number of constraints* of each type in each table. Each Not Null constraint is worth 0.1 marks, each Default constraint is worth 0.1 marks, and each Check/Unique constraint is worth 1.5 marks.

PART 2: SQL QUERIES

For each query, an example SQL statement and the expected result when executed on the sample database for Task 3 is given. Different SQL statements that obtain the same result are also acceptable solutions. The Task3DB.sql script file can be used to test your own queries to see if they get the expected result.

1. Find the genre name and the frequency of occurrence of the five most frequently occurring genres for only those movies that members have watched. Order the result by frequency descending. **[6 marks]**

```
select genre, count(genre)
from Movie natural join (select distinct movieId – Note: should only count each movie's genres once
                         from WatchHistory)
group by genre
order by count(genre) desc
fetch first 5 rows with ties;
```

Expected Result

genre	count(genre)
drama	98
adventure	26
action	25
crime	24
comedy	19

2. Find the cast member name, the role he/she played, the title and the release year of the movie in which the cast member both appeared in and directed the movie, and the movie won the Best Picture Academy award. Order the result by name ascending. **[6 marks]**

```
select name, role, title, releaseYear
from Movie natural join CastMemberAppearsIn natural join DirectorOf natural join MoviePerson
where bestPictureAwardId is not null
order by name;
```

Expected Result

name	role	title	releaseYear
Ben Affleck	Tony Mendez	Argo	2012
Clint Eastwood	Frankie Dunn	Million Dollar Baby	2004
Clint Eastwood	Bill Munny	Unforgiven	1992
Mel Gibson	William Wallace	Braveheart	1995

3. Find the title, release year and MPAA rating of the movies that won the Best Picture academy award and whose director won the Best Director academy award for the same movie, but the movie has not been watched by any member. Order the result by title ascending. **[10 marks]**

```
select title, releaseYear, MPAARating
from Movie
where bestPictureAwardId is not null
        and movieId not in (select movieId
                                from WatchHistory)
        and movieId in (select movieId
                        from AwardWin natural join AcademyAward
                        where awardName='Best Director')
order by title;
```

Expected Result

title	releaseYear	MPAARating
Kramer vs. Kramer	1979	PG
Parasite	2019	R

4. Find the title, release year, running time, MPAA rating and the number of times watched for the movies that have been watched the greatest number of times. Order the result by title ascending. **[12 marks]**

```
with WatchCount as
(select movieId, count(*) as timesWatched
  from Movie natural join WatchHistory
  group by movieId)
select title, releaseYear, runningTime, MPAARating, timesWatched
from Movie natural join WatchCount
where timesWatched=(select max(timesWatched)
                        from WatchCount)
order by title;
```

Expected Result

title	releaseYear	runningTime	MPAARating	timesWatched
Aliens	1986	197	R	7
Guardians of the Galaxy	2014	121	PG-13	7
Hara-Kiri	1962	133	Not Rated	7

5. Find the title, MPAA rating, IMDB rating, Reelflics rating, number of times watched by different female members and number of times watched by different male members for the movies that won the Best Picture academy award and that have been watched more times by different female members than by different male members. Order the result first by the number of times watched by different female members descending, then by the number of times watched by different male members descending, then by title ascending. If a movie has not been watched, then the number of times watched should be shown as zero not as null or as a blank. **[16 marks]**

```
-- Number of times each movie is watched by a different male member
with MaleWatched as
(select movieId, numMaleWatched
 from Movie natural left outer join
        (select movieId, count(distinct username) as numMaleWatched
         from Movie natural left outer join
                (select *
                 from ReelflicsMember natural join WatchHistory
                 where gender='M')
         group by movieId)),

-- Number of times each movie is watched by a different female member
FemaleWatched as
(select movieId, numFemaleWatched
 from Movie natural left outer join
        (select movieId, count(distinct username) as numFemaleWatched
         from Movie natural left outer join
                (select *
                 from ReelflicsMember natural join WatchHistory
                 where gender='F')
         group by movieId)),

-- Calculate Reelflics rating
ReelflicsRating as
(select movieId, reelflicsRating
 from Movie natural left outer join
        (select movieId, avg(rating) reelflicsRating
         from Review
         group by movieId))

select title, MPAARating, IMDBRating, reelflicsRating, numFemaleWatched, numMaleWatched
from Movie natural join MaleWatched natural join FemaleWatched natural join ReelflicsRating
where numFemaleWatched>numMaleWatched
        and bestPictureAwardId is not null
order by numFemaleWatched desc, numMaleWatched desc, title;
```

Expected Result

title	MPAA Rating	IMDB Rating	Reelflics Rating	numFemaleWatched	numMaleWatched
Spotlight	R	8.1		3	0
No Country for Old Men	R	8.1		2	1
One Flew Over the Cuckoo's Nest	R	8.7		2	1
The King's Speech	R	8		2	1
Unforgiven	R	8.2		2	1
Casablanca	PG	8.5		2	0
Shakespeare in Love	R	7.1		2	0
Moonlight	R	7.4		1	0
The Artist	PG-13	7.9		1	0
Titanic	PG-13	7.8	7	1	0