CS4.301: Data and Applications (Monsoon 2023)

Homework – 3

Submission Due: 8 Nov 2023
Instructor: Prof. Ponnurangam Kumaraguru (PK)

1. Integrity Constraints

Refer to the database state and constraints given below and answer the following question

1. Database state

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

2. Referential integrity constraints

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
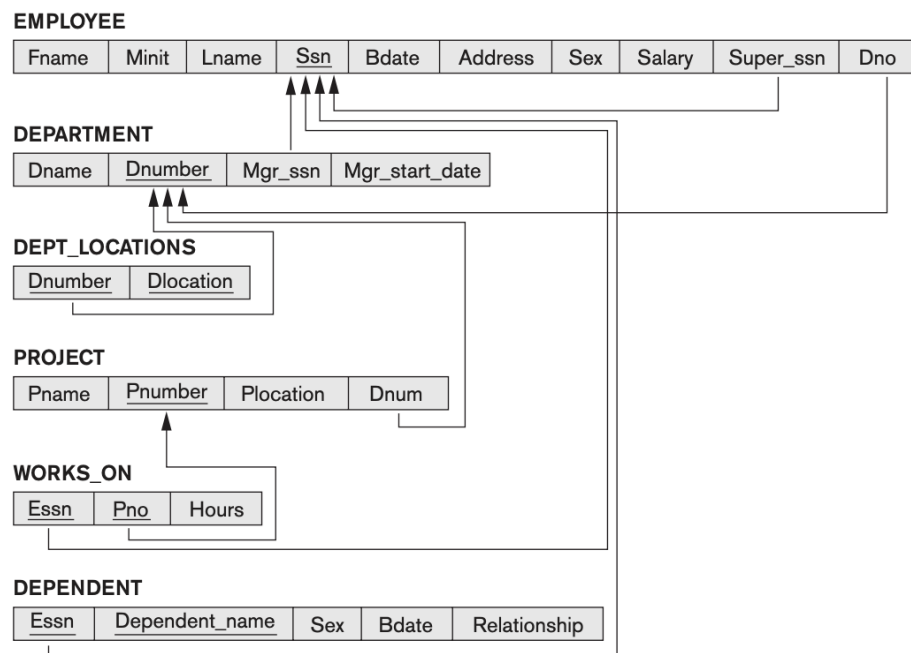
Suppose that each of the following Update operations is applied directly to the database independently. Mention **all integrity constraints** violated by each operation, if any, and discuss why they are violated.

1. Insert <'Robert', 'F', 'Scott', 99988777, '1972-06-21', '2365 Newcastle Rd, Bellaire, TX', M, 58000, '888665555', 1> into EMPLOYEE.

2. Insert <'ProductA', 4, 'Bellaire', 5> into PROJECT.

3. Insert <'Production', 4, '943775543', '2007-10-01'> into DEPARTMENT.

4. Insert <'999887777', NULL, NULL> into WORKS_ON.

5. Insert <'678678678', 'John', 'M', '1990-12-12', 'spouse'> into DEPENDENT.

6. Delete the WORKS_ON tuples with Pno = 1.

7. Delete the EMPLOYEE tuple with Ssn = '987654321'.

8. Delete the PROJECT tuple with Plocation = 'Stafford'

9. Modify the Mgr_ssn and Mgr_start_date of the DEPARTMENT tuple with Dnumber = 5 to '123456789' and '2007-13-01', respectively.

10. Modify the Super_ssn attribute of the EMPLOYEE tuple with Ssn = '999887777' to '333445555'.

11. Modify the Hours attribute of the WORKS_ON tuple with Essn = '999887777' and Pno = 4 to '5.0'.

2. SQL Queries

You are required to write 8 SQL queries on a relational movie database. The data in this database is from the IMDB website. Please read the instructions carefully as checking scripts are automated.

2.1 Database

The database consists of the following 5 tables (case-sensitive):
- Actors *(id, fname, lname, gender)*
- Movies *(id, name, year, ranking)*
- Directors *(id, fname, lname)*
- CAST *(pid, mid, role)*
- Movie_directors *(did, mid)*

The *id* column in ACTORS, MOVIES & DIRECTORS tables is a key for the respective table.
CAST.pid refers to ACTORS.id, CAST.mid refers to MOVIES.id
MOVIE_DIRECTORS.did refers to DIRECTORS.id and MOVIE_DIRECTORS.mid refers to MOVIES.id

A database file has been provided to you at this link which follows the above schema. The database is already populated with the required tuples. Please download and use the provided database to run your queries.

In MySQL, create a database named IMDB, then use the following command to import the data:
```
> mysql -u root -p IMDB < path/to/hw3.sql
```
Try with sudo if the above command fails.
```
> sudo mysql -u root -p IMDB < path/to/hw3.sql
```

2.2 Queries

1. List the first name and last name of actors who acted in at least one film between 1900 and 1950 (both inclusive)
2. List the first name and last name of the directors who directed a film in a leap year (remember: 1900 is *not* a leap year!)
3. List the names of the movies that have the same year as the movie 'Shrek (2001)', but a better ranking. (Note: bigger value of ranking implies a better ranking)
4. List first name and last name of all the actors who were cast members of the movie 'Officer 444 (1926)'
5. List director ids and number of films they have directed in descending order of the number of films they directed
6. List details of the movie with the largest cast.
7. List details of all the actors who have acted in at least 30 films.

8. List movie id, movie name and the number of actors for movies which have *at least 1* cast member whose last names start with "A". Order the list in ascending order by the number of actors fulfilling this criteria.
    Note: Movies may contain actors whose last names don't start with "A", the criteria is not that all cast members must have their last names starting with "A".

(**PS**: any obvious attempt to select the required rows **without attempting** the above queries will be given a **zero**)

2.3 Checker Scripts

A script has been provided that you can use to evaluate your queries: **checker.sh**

Pre-requisites
1. Make sure that your database is named **'IMDB'**.
2. Open the script and enter your mysql password for **root** in the indicated field.
3. Make sure that the **'answers/'** folder provided is placed in the same directory as the script.
    *** Please check this folder to make sure **you are outputting the correct column names. ***
4. Instructions are also provided inside the script so do go through them.

checker.sh: Use this script if you want to evaluate your query for answer **i**.

1. **USAGE**
    ```
    > chmod +x ./checker.sh # you only need to do this once
    > ./checker.sh <question_number> <query>
    ```

2. **Example**

For the checker script to work, Overall directory structure should look like this:

```
<dir>
    - answers/
    - checker.sh
    - <other files>
```

3. Submission Instructions

1. You are required to submit a zip file named *<teamnumber>.zip.* **If not named correctly, a penalty will be applied.**
2. You are required to submit a pdf for part 1, and 8 files for part 2. Please write down assumptions in the README.md if any. The zip file should unzip to following file structure for submission:

```
<team_number>
    - README.md
    - part1.pdf
    - queries
        - 1
        - 2
        - 3
        - 4
        - 5
        - 6
        - 7
        - 8
```
   Please follow the file structure strictly as automated checking will take place. A wrong file name will be given zero marks.
3. Only 1 submission per team is necessary.