**Roll No. _____Name _____ Section ___**

**National University of Computer and Emerging Sciences, Lahore Campus**

| | | | |
|---|---|---|---|
| **Course:** | **Database Systems** | **Course Code:** | **CS203** |
| **Program:** | **BS(Computer Science)** | **Semester:** | **Spring 2019** |
| **Duration:** | **60 Minutes** | **Total Marks:** | **30** |
| **Paper Date:** | **Wed 27-Feb-2019** | **Weight** | **15%** |
| **Section:** | **ALL** | **Page(s):** | **5** |
| **Exam:** | **Midterm-I** | | |

**Instruction/Notes:** Scratch sheet can be used for rough work however, all the questions and steps are to be shown on question paper. ***No extra/rough sheets should be submitted with question paper***.
You will not get any credit if you do not show proper working, reasoning and steps as asked in question statements.

We want to design a website to help students grasp the Database concepts. We will call our site 'Web-For-DB-Dummies'. It will contain articles on the various topics of Database Systems. A registered user can be author, editor or just a reader. Any registered user can submit an article, after approval from an Editor (of our site) it will be uploaded. A user can rate an article on a scale of 1-5.

**NOTE:** Primary keys (PKs) are underlined and foreign keys (FKs) are in *italic* font. Assume referential integrity constraint(RIC) on FKs (*Topic.EditorID, Article.AuthorID* and *Article.TNo*) are On Delete Set Null and On Update Cascade, and RIC on FKs (*ArRating.ANo* and *ArRating.UID*) are On Delete/Update Cascade.

The schema and state of our website is as follows:

**USER**

| UID | UName | Gender |
|---|---|---|
| 1 | Sara | F |
| 2 | Zara | F |
| 5 | Ali | M |
| 3 | Ahmad | M |
| 9 | Aliya | F |
| 13 | Tania | F |
| 6 | Hamza | M |

**ARTICLE**

| ANo | Title | *AuthorID* | *TNo* |
|---|---|---|---|
| 1 | Why we need Relational Algebra? | 9 | 3 |
| 2 | All about Query Execution | 13 | 8 |
| 3 | What is Conceptual DB Model? | 6 | 6 |
| 4 | Nested SQL Queries | 9 | 7 |
| 5 | Transaction Processing | 9 | 9 |

**TOPIC**

| TNo | TName | *EditorID* |
|---|---|---|
| 1 | Intro to DB | 6 |
| 5 | Basic SQL | 5 |
| 7 | Advanced SQL | 9 |
| 9 | Transactions | 13 |
| 6 | ER Model | 1 |
| 8 | Query Execution | 13 |
| 3 | Relational Algebra | 9 |

**ArRATING**

| *ANo* | *UID* | Rating |
|---|---|---|
| 1 | 2 | 3 |
| 1 | 9 | 3 |
| 3 | 2 | 3 |
| 2 | 2 | 4 |
| 2 | 6 | 5 |
| 2 | 13 | 2 |
| 3 | 13 | 5 |
| 4 | 2 | 5 |
| 4 | 6 | 5 |

**Roll No. _____Name _____ Section ___**

**Roll No. _____Name _____ Section ___**

**Q1.** *(10 points)* Apply following operations on the above database. State if the operation would be carried out successfully or not. **Explain your answer briefly.** In case of successful operation indicate the changes that will be made to the above database. Please note that all operations are independent.

**a)** INSERT INTO article (ANo, TNo) VALUES (9, 9);

**Accept O** **Explain:**

Reject O

**b)** INSERT INTO arRating VALUES (3, 2, NULL);

Accept O **Explain:** PK-Unique constraint violation. Tuple# 3 with PK value (3, 2) already exist.

**Reject O**

**c)** DELETE FROM user WHERE name='Aliya';

**Accept O** **Explain:** Remove all matching tuples of parent relation user (i.e. t#5), child relation arRating (i.e. t#2), and also update EditorID & AuthorID attributes of all matching tuples (i.e. t# 3, 7 of Topic & t# 1, 4, 5 of Article) to NULL.

Reject O

**d)** UPDATE arRating SET ANo = 4 WHERE rating=4;

Accept O **Explain:** PK-Unique constraint violation. Tuple# 8 with PK value (4, 2) already exist.

**Reject O**

**e)** DELETE FROM article WHERE TNo=7;

**Accept O** **Explain:** Remove all matching tuples of parent relation Article (i.e. t#4) and child relation arRating (i.e. t#8, 9 with ANo=4).

Reject O

**Roll No. _____Name _____ Section ___**

**Q2.** *(10 points)* Write the result of the following queries for the Database State given above and explain in one sentence what these queries are doing.

**a.** $R1 \leftarrow \Pi_{UID, TNo, Tname}(\sigma_{AuthorID = EditorID \wedge Gender = `F`} USER) \bowtie_{UID=AuthorID} ARTICLE) \bowtie_{TNo=TNo} TOPIC))$
$R2 \leftarrow {}_{UID}\mathcal{F}_{count(*)} R1$

**b.** SELECT ano, editorid
FROM article AS A JOIN (SELECT ano, AVG(rating) FROM arrating GROUP BY ano HAVING AVG(rating)>=4) AS highAR ON A.ano=highAR.ano JOIN topic AS T ON A.tno=T.tno;

**Ans:**

**a)**

**R1**

| UID | TNo | Tname |
|---|---|---|
| 9 | 3 | Relational Algebra |
| 9 | 7 | Advanced SQL |
| 13 | 8 | Query Execution |

**R2**

| UID | COUNT(*) |
|---|---|
| 9 | 2 |
| 13 | 1 |

**b)**

**HighAR**

| ANo | AVG(Rating) |
|---|---|
| ~~1~~ | ~~3~~ |
| ~~2~~ | ~~3.6~~ |
| 3 | 4 |
| 4 | 5 |

| ANo | Title | AuthorID | TNo | ANo | AVG(Rating) | TNo | Tname | EditorID |
|---|---|---|---|---|---|---|---|---|
| ~~3~~ | ~~What is Conceptual DB Model?~~ | ~~6~~ | ~~6~~ | ~~3~~ | ~~4~~ | ~~6~~ | ~~ER Model~~ | ~~1~~ |
| ~~4~~ | ~~Nested SQL Queries~~ | ~~9~~ | ~~7~~ | ~~4~~ | ~~5~~ | ~~7~~ | ~~Advanced SQL~~ | ~~9~~ |

**Final Result:**

| ANo | EditorID |
|---|---|
| 3 | 1 |
| 4 | 9 |

**Roll No. _____Name _____ Section ___**

**Q3.** *(10 points)*

**a.** We wish to find users who gives very high rating. Write a **SQL Query** to list down the names of the USERs who always gives a rating of 4 or 5. For example, in the above database state only user with UID = 6 (name Hamza) has given high rating on every article he rated.

**b.** Write a **Relational Algebra Query** to list down the IDs and name of the authors who have written articles on the Topic named "Intro to DB" and also on the Topic named "Basic SQL".

**Ans:**

**a)**

**Option1:**

```
SELECT UName FROM user JOIN arRating ON user.UID=arrating.UID WHERE rating>=4
EXCEPT
SELECT UName FROM user JOIN arRating ON user.UID=arrating.UID WHERE rating<4;
```

**Option2:**

```
SELECT UName
FROM user
WHERE UID IN      (SELECT   UID   FROM   arRating
                EXCEPT
                SELECT   UID   FROM   arRating   WHERE   rating<4);
```

**b)**

$$R1 \leftarrow \Pi_{AuthorID,\ UName}(USER \bowtie_{UID=AuthorID} ¿_{TName='Intro\ to\ DB'} TOPIC \bowtie_{TNo=TNo} ARTICLE))$$

$$R2 \leftarrow \Pi_{AuthorID,\ UName}(USER \bowtie_{UID=AuthorID} ¿_{TName='Basic\ SQL'} TOPIC \bowtie_{TNo=TNo} ARTICLE))$$

$$Result \leftarrow R1 \cap R2$$

**Roll No. _____Name _____ Section ___**