# National University of Computer and Emerging Sciences, Lahore Campus

| | | | |
|---|---|---|---|
| Course: | Database Systems (Lab) | Course Code: | CL203 |
| Program: | BS (Computer Science) | Semester: | Fall 2019 |
| Duration: | 2.5 Hours | Total Marks: | 100 |
| Paper Date: | 30-Nov-2019 | Weight | 40% |
| Section: | All | Page(s): | 3 |
| Exam: | Lab Final Exam | Reg. No | |

**EXAM_NO_1**

**Important Instructions (Please read them before attempting the exam):**

- Submit **ONLY .sql File** in this format (All parts of a question in one SQL File **named** with your **Roll Number** e.g., L13-4152). Do not zip your file.
- Write question 1's solution in one file and questions 2's solution in another file.
- Copy the schema for question 2 from the following path \\cactus\xeon\Ishaq\Database Systems- spring 2019\Lab Material\final_schema.sql. Put the sql file in your D drive, and then **unplug the ethernet cable.**
- **Plagiarism** will result in **F grade** in lab.
- No cell phones are allowed. Sharing of **USBs** or any other items is **not allowed.**
- Submission path will be announced soon.
- **Write your roll number on this paper, and submit this paper to invigilator before leaving the lab.**

**SQL Server Login Details:**

- **Server: localhost**
- **username: sa**
- **password: fstky2e4mdt (OR) 12345678**

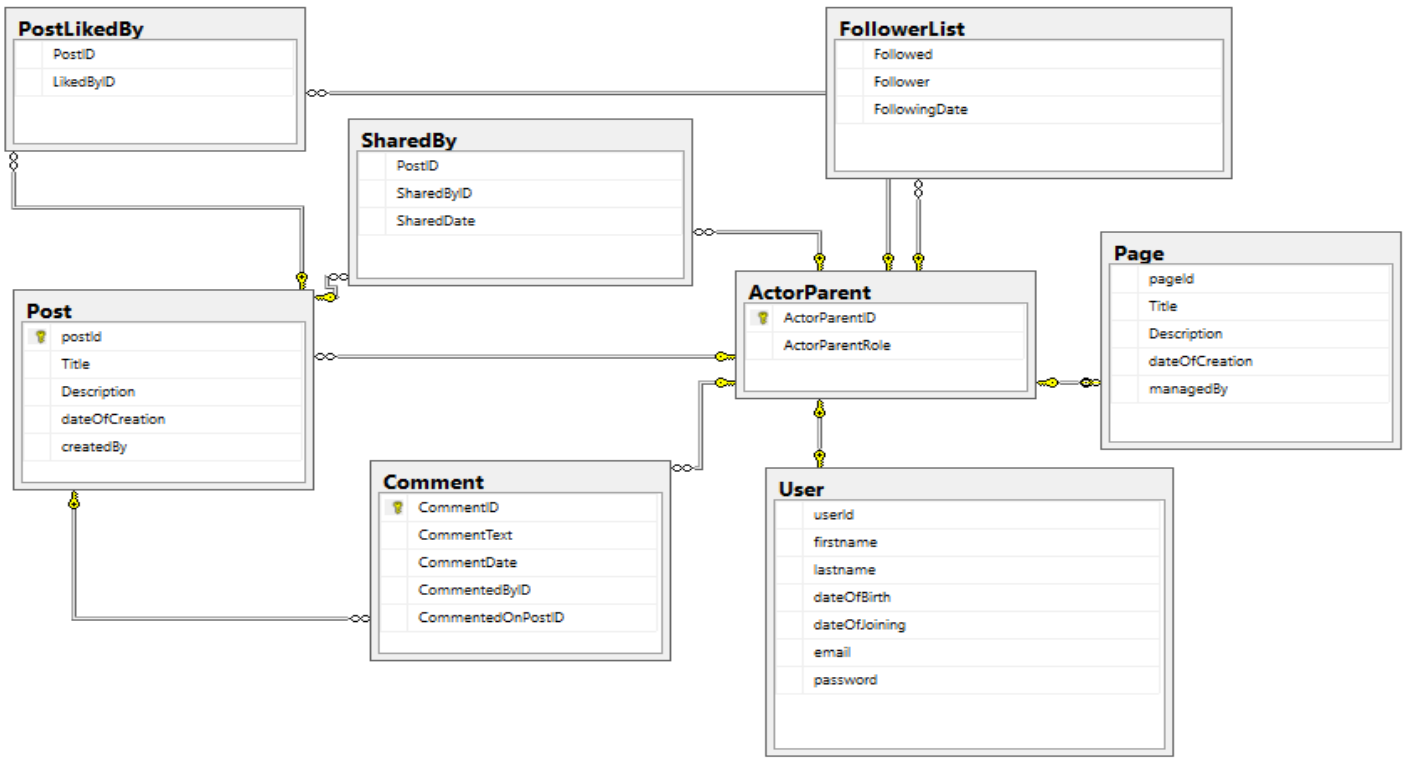
**Question #1 Write the SQL solution of the following questions. Clearly mention the part number above each SQL solution in comments.**

1) Write a stored procedure for insertUser which takes First Name, Last Name, Email, Password and DateOfBirth as parameter. This procedure checks whether any user with same email address exists already in User table or not. If already exists return -1. If User with same email address does not already exist, insert a tuple in ActorParent. The id of this tuple is the maximum value of Already exiting IDs plus one. And the role of this tuple is 1. Now insert the user in users table with id of this inserted tuple. And Date of Joining should be current Date. If the data is inserted successfully return 1. Otherwise return 2.
2) Write a view 'popActors' which lists the POPULAR ACTORS along with their all attributes. 'popular Actors' are the users who have created more than 2 posts or have shared more than 2 posts or have at least 5 comments. Actors term is used for both page and user.
3) Write a user defined function which uses view 'popActors' and print the complete timeline of each popular Actor. You will print following things for each tuple in popActors:
   i. Print Profile: Profile of ActorParent consists of, it's id, role (1 is for User and 2 is for Page). If actor is User, you need to print the attributes of User and if role is Page, then you need to print the attributes of Page.
   ii. Print Posts (created/SharedBy): Posts which are created by this actor and posts shared by this actor sorted in descending order.
4) Write a user define function named as 'bestFriends'. This function finds the pair of actors which follow each other. You need to print these things:
   For every pair of best friends print these:
   i. Print First name of both friends.
   ii. The count of posts on which both friends have commented.
   iii. The count of likes (for how many times they liked the common posts).
   iv. You need to print the friendship age (consider the latest followingDate for friendship age).

5) Write a user defined function which finds the celebrity using FollowerList.
   For an ActorParent C there are two types of followers:
   i. Explicit Followers: Who follow C directly.
   ii. Implicit Followers: Who are actually the followers of followers of C.
   An ActorParent C is celebrity if the sum of explicit and implicit followers is maximum. You need to print the name/title of celebrity and the sum of implicit and explicit followers.

6) As you know there is no cascade on update of Post. Write a trigger on update of Post. Whenever a Post is going to be updated. You need to update the tuples from each child table. Then update the Post. There must not be any runtime error.
7) Whenever a post is shared by some ActorParent. This post is replicated automatically. So, write a trigger which replicates the post in Post table. This trigger finds the id of current post, sharedByID and sharedDate from inserted table. Now inserts a new tuple in Post table with previous title and description. But new postId, new creationDate and createdBy. This createdBy is actually the person/page who is sharing the post.
   Hint: New Post ID = max(postId) + 1.
8) Whenever a page is updated you need to check whether given information is valid.
   i. PageID must exist in ActorParent table.
   ii. Title should have length at least 5 characters.
   iii. Description should have 10 characters.
   iv. ManagedBy must be in ActorParent table and must have a role 1. (ManagedBy must be user).
   v. DateOfCreation must not be beyond currentDate and must not behind the DateOfJoining the user who is managing the page.
   You need to write an after trigger, which checks the values of inserted tuples. If a tuple is valid, it allows to update and if tuple violates any condition, do not allow this tuple to be updated.
9) Write a trigger on insertion and update of FollowerList. Neither the follower and followed be same nor the tuple should be existing already. As well as the following date should not be beyond the current date and not even following date should be behind the joining date of actor Parents.
10) Write a procedure to suggest the friends to those two ActorParents who have joined most recently. Suggest those ActorParents who are Popular Actors. Use the view 'popActors' to suggest the friends.

**Schema:** This schema is about a social website where **users** and **pages** can create, like and share the **posts**. **ActorParent** is parent of **user** and **page**. **ActorParent** can **comment** on a specific **post**. **Users** and **pages** can follow each other. To resolve the issue of many to many relationship **SharedBy** and **PostLikedBy** tables are used.



**Database values:**

| | ActorParentID | ActorParentRole |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 1 |
| 4 | 4 | 1 |
| 5 | 5 | 1 |
| 6 | 6 | 2 |
| 7 | 7 | 2 |
| 8 | 8 | 1 |
| 9 | 9 | 2 |

| | PostID | SharedByID | SharedDate |
|---|---|---|---|
| 1 | 8 | 6 | 2013-01-01 |
| 2 | 8 | 4 | 2013-01-15 |
| 3 | 5 | 1 | 2011-03-04 |
| 4 | 5 | 3 | 2011-03-14 |
| 5 | 5 | 6 | 2011-04-04 |
| 6 | 9 | 4 | 2012-11-04 |
| 7 | 9 | 3 | 2012-11-10 |
| 8 | 9 | 5 | 2012-11-14 |
| 9 | 2 | 8 | 2011-01-04 |
| 10 | 3 | 3 | 2011-01-23 |

| | userId | firstname | lastname | dateOfBirth | dateOfJoining | email | password |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Muhammad | Ali | 1992-11-11 | 2009-11-11 | muhammad_ali@hotmail.com | ali123 |
| 2 | 3 | Soban | Ali | 1995-06-23 | 2010-05-10 | Soban_ali@live.com | 659854 |
| 3 | 4 | Saad | Ali | 2001-10-08 | 2010-07-07 | Saad_ali@outlook.com | AllahisO... |
| 4 | 5 | Adeel | Ahmed | 1996-05-13 | 2012-04-29 | adeel_ahmed@gmail.com | dumma... |
| 5 | 8 | Abdul Wa... | Akram | 1990-06-12 | 2015-05-31 | wahab_akram@gmail.com | 987654... |

| | Followed | Follower | FollowingDate |
|---|---|---|---|
| 1 | 8 | 6 | 2013-01-01 |
| 2 | 8 | 4 | 2013-01-15 |
| 3 | 5 | 1 | 2011-03-04 |
| 4 | 5 | 3 | 2011-03-14 |
| 5 | 4 | 8 | 2011-04-04 |
| 6 | 1 | 4 | 2012-11-04 |
| 7 | 2 | 3 | 2012-11-10 |
| 8 | 2 | 5 | 2012-11-14 |
| 9 | 2 | 8 | 2011-01-04 |
| 10 | 1 | 5 | 2011-01-23 |

| | pageId | Title | Description | dateOfCreation | managedBy |
|---|---|---|---|---|---|
| 1 | 2 | Mualam Study Anything | The mission of page is to teach needy. | 2010-12-31 | 1 |
| 2 | 6 | Maria B | The updates of Brand. | 2015-06-16 | 4 |
| 3 | 7 | Prof. A. Rafique Akhtar | Allah or Insaan ka rishta | 2012-10-08 | 1 |
| 4 | 9 | DummyPage | This page is used for temporary purp... | 2018-02-28 | 3 |

| | postId | Title | Description | dateOfCreation | createdBy |
|---|---|---|---|---|---|
| 1 | 1 | Lets create a page | thinking to create a page for database systems. | 2010-12-26 | 1 |
| 2 | 2 | My page #mualamofficial | Checkout the page and study from anywhere | 2010-12-31 | 1 |
| 3 | 3 | Database Connectivity | Checkout the first lecture on database conne... | 2011-01-01 | 2 |
| 4 | 4 | DML exercises | Checkout the Second lecture on database D... | 2011-01-11 | 2 |
| 5 | 5 | DB Project Demo | Checkout the Third lecture on database Proje... | 2011-02-22 | 2 |
| 6 | 6 | #MariaB | thinking to create a page for MariaB. | 2015-06-29 | 4 |
| 7 | 7 | My page #MariaB | Checkout the latest trends for the brand | 2010-07-31 | 6 |
| 8 | 8 | #Tasawuf | Some people say the noor of Allah is blue | 2012-10-05 | 1 |
| 9 | 9 | 1st Lecture by Prof. Sb | Professor Sb has given a beautifull lecture | 2012-10-03 | 7 |

| | PostID | LikedByID |
|---|---|---|
| 1 | 5 | 1 |
| 2 | 2 | 2 |
| 3 | 4 | 6 |
| 4 | 4 | 4 |
| 5 | 3 | 3 |
| 6 | 1 | 5 |
| 7 | 2 | 8 |
| 8 | 3 | 1 |
| 9 | 3 | 2 |
| 10 | 5 | 6 |
| 11 | 5 | 4 |
| 12 | 2 | 3 |
| 13 | 3 | 5 |
| 14 | 2 | 8 |
| 15 | 1 | 1 |
| 16 | 9 | 2 |
| 17 | 4 | 6 |
| 18 | 7 | 7 |
| 19 | 5 | 3 |
| 20 | 7 | 5 |
| 21 | 3 | 8 |

| | CommentID | CommentText | CommentDate | CommentedByID | CommentedOnPostID |
|---|---|---|---|---|---|
| 1 | 1 | Its is great news | 2010-12-26 | 4 | 1 |
| 2 | 2 | Happy to see the post | 2010-12-31 | 4 | 1 |
| 3 | 4 | I will be loving this | 2011-01-01 | 5 | 2 |
| 4 | 5 | Hurray a good portal | 2011-01-01 | 8 | 2 |
| 5 | 6 | I will be loving this | 2011-01-01 | 4 | 2 |
| 6 | 7 | a good lecture | 2011-01-11 | 8 | 2 |
| 7 | 8 | Loved this | 2011-01-11 | 4 | 2 |
| 8 | 9 | Really, a beautifull l... | 2012-10-03 | 1 | 9 |