Assignment

Sept23/ DBT/126

Database Technologies

Diploma in Advance Computing

September 2023

**Procedure**

|  |
| --- |
| 1. Create a LOGIN table (username, password, and email). Write a procedure (named ***addUser***) to pass the username, password, and email-ID through the procedure and store the data in the LOGIN table. |
| DROP PROCEDURE if EXISTS adduser;  delimiter $  create PROCEDURE adduser(username varchar(45),password varchar(45),emailID varchar(128))  BEGIN  insert into login VALUES (username,password,emailID);  end $  delimiter ; |
|  |
| 1. Create a LOG table having following columns (id (auto\_increment), curr\_date, curr\_time, and message). Write a procedure (named ***checkUser***) to pass the email-ID as an input, check whether passed email-ID is available in LOGIN table or not available. If the email-ID is available then display the username and his password. If the email-ID is not available then, insert (curr\_date, curr\_time, and message) in LOG table. |
| drop PROCEDURE if EXISTS checkUser;  delimiter $  CREATE PROCEDURE checkUser(\_emailId VARCHAR(128))  BEGIN  DECLARE \_checkmail bool;  select true into \_checkmail from login where \_emailId=emailId;  if \_checkmail THEN  SELECT username,password from login WHERE \_emailId=emailID;  ELSE  INSERT into log (curr\_date,curr\_time,message)  VALUES(CURRENT\_DATE,CURRENT\_TIME,"email doesnt exist");  SELECT \* from log;      end if;  end $  delimiter ; |
|  |
| 1. Write a procedure(named getQualification) that takes studentID as a parameter. If studentID is present in the student table, then print his student details along with STUDENT\_QUALIFICATION details and if the studentID is not present display message “Student not found…” (Use: STUDENT, and STUDENT\_QUALIFICATION tables) |
| DROP PROCEDURE if EXISTS getQualification;  delimiter $  CREATE PROCEDURE getQualification(studentID int)  BEGIN  DECLARE \_checkid BOOL;  SELECT true into \_checkid from student WHERE studentID=id;  if \_checkid THEN    select \* from student s join student\_qualifications sq on s.id=sq.studentid where s.id=studentID;  ELSE  select "student not found";    end IF;  end $  delimiter ; |
|  |
| 1. Write a procedure (named addStudent) that inserts a new student with his phone number and his address into the STUDENT, PHONE, and ADDRESS table. |
| DROP PROCEDURE IF EXISTS addStudent;  delimiter $  create procedure addStudent(\_namefirst VARCHAR(30), \_namelast VARCHAR (30),\_DOB DATE , emailID varchar(30) ,\_number varchar(10),\_isActive bool, \_address varchar(50))  BEGIN  declare \_idStudent int;  declare \_idStudentPhone int;  declare \_idStudentAddress int;  select max(id)+1 into \_idStudent from student;  select max(id)+1 into \_idStudentPhone from student\_phone;  select max(id)+1 into \_idStudentAddress from student\_address;  Insert into student VALUES(\_idStudent,\_namefirst,\_namelast,\_DOB,\_emailID);  insert into student\_phone VALUES(\_idStudentPhone,\_idStudent,\_number,\_isActive);  Insert into student\_address VALUES(\_idStudentAddress,\_idStudent,\_address);  end $  delimiter ; |
|  |
| 1. Write a procedure (named addQualification) that takes studentID, and qualification details as a parameter. If studentID is present in the STUDENT table, then insert the qualification in STUDENT\_QUALIFICATION table and return a message “Record inserted” or else print ‘Student not found’. (hint: using OUT parameter) (Use: STUDENT, and STUDENT\_QUALIFICATION tables) |
| drop procedure if exists addQualification;  delimiter $  create procedure addQualification(\_studentID int ,\_name varchar(20),\_college varchar(20),  \_university varchar(20),\_marks int, \_YEAR year)  BEGIN  declare \_idStudentQualification int;  declare rec\_Present BOOLEAN;  select max(id)+1 into \_idStudentQualification from student\_qualifications;  select DISTINCT true into rec\_Present from student\_qualifications where  \_studentid= student\_qualifications.studentid;    if rec\_Present THEN  Insert into student\_qualifications VALUES  (\_idStudentQualification,\_studentID,\_name,\_college, \_university,\_marks,\_YEAR);  select "Record inserted";    ELSE  select "Student not found";  end if ;  end $  delimiter ; |
|  |