­­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(50),password varchar(50),emailID varchar(50))  BEGIN  INSERT INTO Login(username,password,emailID)values(username,password,emailID);  SELECT \* FROM Login;  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(50))  BEGIN  declare x boolean;  SELECT true into x FROM Login WHERE emailID=\_emailID;  IF(x=true)  THEN  SELECT username,password,emailID FROM Login WHERE emailID=\_emailID;  ELSE  INSERT INTO LOG(curr\_date,curr\_time,message)VALUES(curdate(),curtime(),"Email ID not Found");  TABLE 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(\_ID int)  BEGIN  declare x boolean;  SELECT true into x FROM student WHERE ID=\_ID;  IF(x=true)  THEN  SELECT \* FROM student s,student\_qualifications WHERE s.ID=studentID AND s.ID=\_ID;  ELSE  SELECT "Student Not found" Message;  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(10),namelast varchar(10),DOB date,emailID varchar(30),number varchar(10),isActive boolean,address varchar(50))  BEGIN  declare x,y,z int;  SELECT max(ID)+1 INTO x FROM student;  SELECT max(ID)+1 INTO y FROM student\_phone;  SELECT max(ID)+1 INTO z FROM student\_address;  INSERT INTO student(ID,namefirst,namelast,DOB,emailID)VALUES(x,namefirst,namelast,DOB,emailID);  INSERT INTO student\_phone(ID,studentID,number,isActive)VALUES(y,x,number,isActive);  INSERT INTO student\_address(ID,studentID,address)VALUES(z,x,address);  SELECT \* FROM student s JOIN student\_phone sp JOIN student\_address sa ON s.ID=sp.studentID AND s.ID=sa.studentID WHERE s.ID=x;  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(\_ID int,name varchar(20),college varchar(30),university varchar(30),marks varchar(30),year int)  BEGIN  declare x BOOLEAN;  declare y int;  SELECT true INTO x FROM student WHERE ID=\_ID;  SELECT max(ID)+1 INTO y FROM student\_qualifications;  IF(x=true)  THEN  INSERT INTO student\_qualifications (ID, studentID, name, college, university, marks, year) VALUES (y,\_ID,name,college,university,marks,year);  SELECT "Record Inserted" Message;  SELECT \* FROM student\_qualifications WHERE studentID=\_ID;  ELSE  SELECT "Student Not found " message;  END IF;  END $  delimiter ; |
|  |