Assignment

March23/ DBT/126

Database Technologies

Diploma in Advance Computing

March 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 p1;  delimiter $  create procedure p1(username varchar(20),password varchar(20),email varchar(20))  begin  insert into login values(username,password,email);  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 pr2;  delimiter $  create procedure pr2 (e varchar(20))  begin  declare y varchar(20);  declare z varchar(20);  declare \_email varchar(20);  set \_email=e;  select username,password into y,z from login where email=\_email;    if(y is not null) THEN  select username,password from login;  else  insert into log values(curr\_date,curr\_time,message);  select curr\_date,curr\_time,message 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 getqualifications;  delimiter $  create procedure getqualifications(studentID int)  begin  declare R varchar(20);  select namefirst into R from student where ID=studentID;  if R is not null then  select \* from student inner join student\_qualifications ON student.id=student\_qualifications.id;  else  select '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 addQualification;  delimiter $  create procedure addQualification(name varchar(20), number int, address varchar(20))  begin  declare x int default 0;  declare y int default 0;  select max(id) + 1 from student into x ;  insert into student values (x,name);  declare y int default 0;  select max(id) + 1 from student\_phone into y;  insert into student\_ phone values(x,y,number);  select max(id) + 1 from student\_address into z;  insert into student\_address values(x,y,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 (id1 int, degree varchar(20),out msg varchar(20))  begin  declare flag bool default false;  declare x int default 0;  select true into flag from student where student.id= id1;  if flag then  select max(id)+1 into x from student\_qualifications ;  insert into student\_qualifications (id,studentid,name) values (x,id1,degree);  select "Record inserted" into msg;  else  select "Student not found" into msg;  end if;  end $  delimiter ; |