Assignment

Feb19/ DBT/126

Database Technologies

Diploma in Advance Computing

February 2019

**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. |
| create table login (username varchar(20), password varchar(20), email varchar(20));  drop procedure if exists addUser;  delimiter $$  create procedure addUser (userName varchar(20), password varchar(20), email varchar(20))  begin  insert into login values(userName, password, email);  Select "Record inserted..." as Message;  end$$  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. |
| create table log (id int, curr\_date date, curr\_time time, message varchar(20));  drop procedure if exists checkUser;  delimiter $$  create procedure checkUser (\_email varchar(20))  begin  declare \_username varchar(20);  declare \_password varchar(20);  select username, password into \_username, \_password from login where email = \_email;    if \_username is not null and \_password is not null then  select \_username, \_password as Message;  else  Insert into log (curr\_date, curr\_time, message) values ((select current\_date), (select current\_time), "Record not found...");  end if;  end$$  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 $$  create procedure getQualification (\_id int)  begin  declare id int;  select distinct s.id into id from student s,student\_qualifications q where s.id = q.studentid and s.id = \_id;    if id is not null then  select \* from student s, student\_qualifications q where s.id = q.studentid and s.id = \_id;  else  select "Student not found…" as "Message";  end if;  end$$  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 $$  create procedure addStudent (\_id int, \_nameFirst varchar(45), \_nameLast varchar(45), \_DOB date, \_emailID varchar(128), \_number varchar(10), \_address varchar(128))  begin  insert into student values(\_id, \_nameFirst, \_nameLast, \_DOB, \_emailID);    insert into student\_phone values(\_id, \_id, \_number);    insert into student\_address values(\_id, \_id, \_address);  end$$  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 $$  create procedure addQualification (in \_ID int, \_name varchar(128), \_college varchar(128), \_university varchar(128), \_marks varchar(45), \_year int, out \_message varchar(100))  begin  declare x int;  select student.id into x from student where student.id = \_ID;  if x is not null then  insert into student\_qualifications values (\_ID, \_ID, \_name, \_college, \_university, \_marks, \_year);  set \_message := "Record inserted...";  else  set \_message := "Employee not found";  end if;  end$$  delimiter ; |
|  |