­­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(in userName varchar(30),in pass varchar(30),in email varchar(30))  begin  declare exit handler for 1050 select "Table with same name already exists" as 'Error';  create table login(userName varchar(30),pass varchar(30),email varchar(30));  insert into login values(username,pass,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 addUser;**  **delimiter $**  **create procedure addUser(in \_email varchar(30),in \_msg varchar(3000))**  **begin**    **declare flag bool;**    **select true into flag from login where \_email=login.email;**      **if flag THEN**  **select userName,pass from login where \_email=login.email;**  **ELSE**  **insert into log(curr\_date,curr\_time,msg) values(CURRENT\_DATE(),CURRENT\_TIME(),\_msg);**  **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 flag bool;    select true into flag from student where \_studentid=student.id;      if flag THEN  select \* from student join student\_qualifications on \_studentid=student.id=student\_qualifications.studentid;  ELSE  select "Student not found!!" as "Error";  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(nfirst varchar(45),nlast varchar(45),\_number varchar(45),\_address varchar(128))**  **begin**  **declare flag bool;**  **select true into flag from student where nfirst=student.namefirst and nlast=student.namelast;**      **if flag THEN**  **select "student with same namefirst and namelast already existed" as "Duplication";**    **ELSE**    **insert into s(namefirst,namelast) values(nfirst,nlast);**  **insert into student\_phone(number) values(\_number);**  **insert into student\_address(address) values(\_address);**    **end if;**    **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,\_sId int,\_name varchar(128),\_college varchar(128),\_university varchar(128), \_marks varchar(45),\_year int)  begin  declare flag bool;  select true into flag from student where \_id=student.id;  if flag THEN  insert into student\_qualifications values(\_sid ,id ,\_name,\_college ,\_university , \_marks ,\_year);  ELSE  select "Student with this id is not found" as "Result";  end if;  end $  delimiter ; |
|  |