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. |
| --- |
| SQL: create table LOGIN (username varchar(45), password varchar(45), email varchar(45));  PL/SQL :  drop PROCEDURE if exists addUser;  delimiter $  create PROCEDURE addUser(in username varchar(45), in password varchar(45), in email varchar(45))  BEGIN  INSERT INTO LOGIN (username, password, email) VALUES (username, password, email);  END $  delimiter ;  —------------------------------------------------------------------------------------------------  drop PROCEDURE if exists addUser;  delimiter $  create PROCEDURE addUser(in username varchar(45), in password varchar(45), in email varchar(45))  BEGIN  INSERT INTO LOGIN VALUES (username, password, email);  END $  delimiter ;  —-----------------------------------------------------------------------  drop PROCEDURE if EXISTS addUser;  delimiter $  CREATE PROCEDURE addUser()  BEGIN  INSERT into login values ('rishi', 'xxxxxx', 'rishi@rishi.com'), ('chitra', 'yyyyyyy', 'chitra@rishi.com');  END $  delimiter ;  SQL: call addUser('rishi', 'xxxxxxxx', '[rishi@rishi.com](mailto:rishi@rishi.com)');  call addUser('hrishi', 'YYYYYY', 'hrishi@rishi.com'); |
|  |
| 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. |
| SQL : CREATE table LOG ( id int primary key auto\_increment, curr\_date DATE NOT NUll, curr\_time DATE not null, message varchar(45) not null);  PL/SQL:  drop PROCEDURE if exists checkUser;  delimiter $  create PROCEDURE checkUser(in emailID varchar(45))  BEGIN  declare username varchar(45);  declare password varchar(45);  select username, password into username, password from LOGIN where email=emailId;    IF username IS NOT NULL AND password IS NOT NULL THEN  SELECT CONCAT('Username: ', username, ', Password: ', password) AS 'User Details';  ELSE  INSERT into LOG ( curr\_date, curr\_time, message) values (curdate(), curtime(), 'Email-ID not found in LOGIN table.');  END IF;  END $  delimiter ;  MAC 👍  CREATE table LOG ( id int primary key auto\_increment, curr\_date date, curr\_time DATE, message varchar(45));  PL/SQL 👍  drop procedure if exists checkUser;  delimiter $  create procedure checkUser(in emailID varchar(45))  BEGIN  declare x bool;  select true into x from LOGIN where emailID=email;  if x is true THEN  select \* from login where email=emailID;  else  insert into LOG (curr\_date, curr\_time, message) values (curdate(), curtime(), 'EmailID not present');  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  if studentID in (select id from student) THEN  select \* from student join student\_qualifications  where student.id= student\_qualifications.studentID and student.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 ;  drop procedure if exists p12;  delimiter $  create procedure p12(\_id int, \_namefirst varchar(45), \_namelast varchar(45), \_dob date, \_emailID varchar(45), \_id1 int, \_saID int, \_number varchar(45), \_isActive tinyint, \_id2 int, \_spid int, \_address varchar(45))  begin  insert into student values (\_id, \_namefirst, \_namelast, \_dob, \_emailID);  insert into student\_phone values (\_id1, \_saID, \_number, \_isActive);  insert into student\_address values (\_id2, \_spid, \_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( id int,\_id int, name varchar(20),college varchar(20),university varchar(50) ,marks int,year1 varchar(10))  BEGIN  declare x bool;  select true into x from student s where s.ID=\_id;  IF x then  insert into student\_qualifications(id,studentid,name,college,university,marks,year) VALUES(id,\_id,name,college,university,marks,year1);  select "Record inserted";  else  select "Student not found";  end if;  end $  delimiter ; |
|  |