Assignment – 0084-0025

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(20), password varchar(20), email\_id varchar(20))  BEGIN  create table LOGIN ( username varchar(20), password varchar(20), email\_id varchar(20));  insert into login values( username,password,email\_id);  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 log1;  delimiter $  create procedure log1()  BEGIN  create table log1(id int primary key auto\_increment,insertdate date,inserttime time,message varchar(200));  end $  delimiter ;  drop procedure if exists checkuser;  delimiter $  create procedure checkuser(email\_id1 varchar(20))  BEGIN  declare x bool;  select true into x from login where email\_id=email\_id1;  IF x=true THEN  select username,password from login;  else  insert into log1 (insertdate,inserttime,message) values(curdate(),curtime(),'message hi');  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(studentid1 int)  BEGIN  declare x bool;  select true into x from student s where s.id=studentid1;  IF x=true THEN  select s. \*, sq. \* from student s inner join student\_qualifications sq where s.id=sq.studentid and s.id=studentid1;  ELSE  select "Student not found!, in student table";  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,Firstname varchar(20),Lastname varchar(20),DOB1 date, email\_id varchar(100), spid int, Phone varchar(20), isActive bool, aID int, address1 varchar(100))  BEGIN  insert into student (id,namefirst,namelast,DOB,emailid) values(id,firstname,lastname,DOB1,email\_id);  insert into student\_phone (id,studentid,number,isActive) values (spid,id,phone,isActive);  insert into student\_address (id,studentid,address) values (aid,id,address1);  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,studentID1 int, name1 varchar (20), college1 varchar(20), university1 varchar(20), mark1 int, passoutyear1 varchar(20))  BEGIN  declare x bool;  select true into x from student s where s.id=studentid1;    if x=true THEN  insert into student\_qualifications (id,studentid,name,college,university,marks,year) values (id,studentid1,name1,college1,university1,mark1,passoutyear1);  select 'Record Inserted' R1;    ELSE    Select 'Student Not Found !' R2;    END IF;  END $  delimiter ; |
| mysql> call addqualification(86,30,10,'NYU','New york','87','2015'); |