

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Programme | : | **M.Tech. (Integrated) Software Engineering** | Semester | : | **Fall 20-21** |
| Course | : | **Advanced Database Management Systems** | Code | : | **SWE2014** |
| Faculty | : | **Dr.Bhuvaneswari A** | Slot | : | **L27 + L28** |
| Date | : | **14-07-2020** | Marks | : | **5 Points** |
|  |  |  |  |  |  |

**NAME: K. NAVEEN KUMAR**

**REG NO: 17MIS1056**

**Exercise 1: ADVANCED REFERENTIAL INTEGERITY CONSTRAINTS**

**Creating Integrity Constraints on Relational Database Schema**

The Project Manager of Project MIKO wants to automate the process of allocation of bug fixing

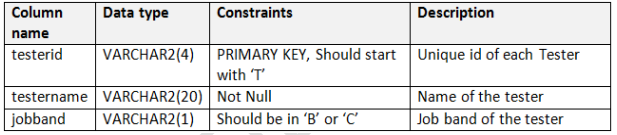
to the software engineers in the team. The testing team identifies the bugs and logs them. The

PM has to allocate these bugs to the development team. The development team has to fix the

bugs and close them before they get escalated. The escalation date is given no sooner a bug is

allocated to the developer.

**Table 1: tester**



**ANS:**

CREATE TABLE tester (

testerid VARCHAR2(4) PRIMARY KEY CHECK (testerid LIKE 'T%'),

testername VARCHAR2(20) NOT NULL,

jobband VARCHAR2(1) CHECK (jobband in ('B', 'C'))

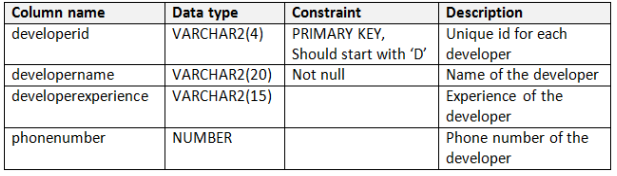
);



The developer table details the team of developers involved in fixing the bugs allocated to

them.

**Table 2: developer**



**ANS:**

CREATE TABLE developer (

developerid VARCHAR2(4) PRIMARY KEY CHECK (developerid LIKE 'D%'),

developername VARCHAR2(20) NOT NULL,

developerexperience VARCHAR2(15),

phonenumber NUMBER

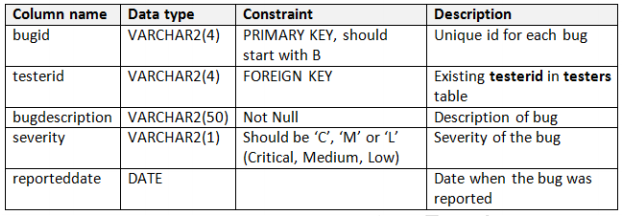
);



The bugreport table details the bug identified by testers, a brief description of the bug, its

severity and the date.

**Table 3: bugreport**



**ANS:**

CREATE TABLE bugreport (

bugid VARCHAR2(4) PRIMARY KEY CHECK (bugid LIKE 'B%'),

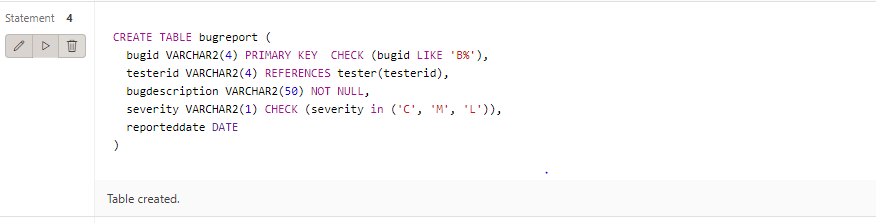
testerid VARCHAR2(4) REFERENCES tester(testerid),

bugdescription VARCHAR2(50) NOT NULL,

severity VARCHAR2(1) CHECK (severity in ('C', 'M', 'L')),

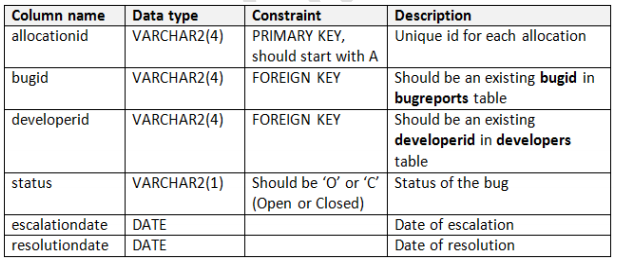
reporteddate DATE

);



The bugfixallocation table contains the details of allocation of bugs to developers and its corresponding status. The developer is expected to fix the bug within a specified timeframe. If a bug is not resolved within this timeframe then it will be escalated. The resolution date is the date when the developer has fixed the bug.

**Table 4: bugfixallocation**



**ANS:**

create table bugfixallocation(

allocation varchar2(4) primary key,

bugid varchar2(4),

developerid varchar2(4),

status varchar2(1),

escalationdate date,

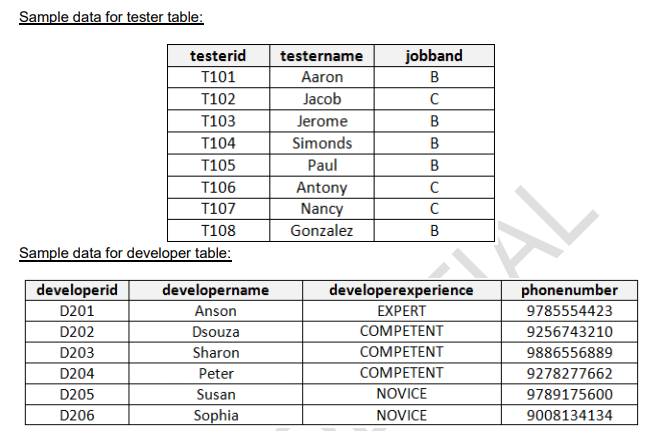
resolutiondate date,

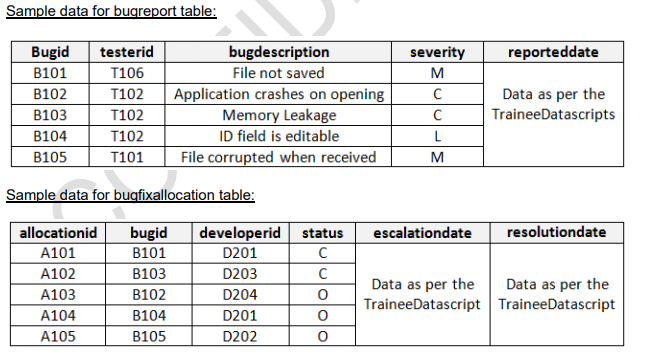
FOREIGN KEY (bugid) REFERENCES bugreport(bugid),

FOREIGN KEY (developerid) REFERENCES developer(developerid)

);







**INSERTING VALUES IN TABLE:**

**Tester table:**

**insert into tester values('T101','Aaron','B');**

**insert into tester values('T102','Jacob','B');**

**insert into tester values('T103','Jerome','B');**

**insert into tester values('T104','Simonds','B');**

**insert into tester values('T105','Paul','B');**

**insert into tester values('T106','Antony','B');**

**insert into tester values('T107','Nancy','B');**

**insert into tester values('T108','Gonzalez','B');**



**Developer table:**

**insert into developer values('D201','Anson','EXPERT',9785554423);**

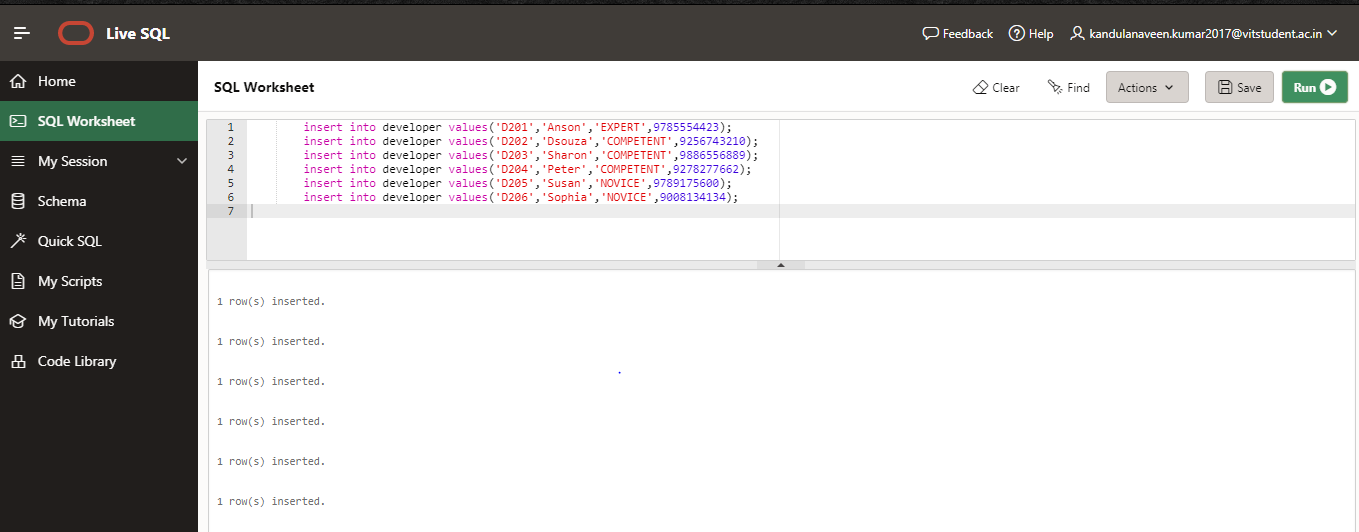
**insert into developer values('D202','Dsouza','COMPETENT',9256743210);**

**insert into developer values('D203','Sharon','COMPETENT',9886556889);**

**insert into developer values('D204','Peter','COMPETENT',9278277662);**

**insert into developer values('D205','Susan','NOVICE',9789175600);**

**insert into developer values('D206','Sophia','NOVICE',9008134134);**



**Bugreport table:**

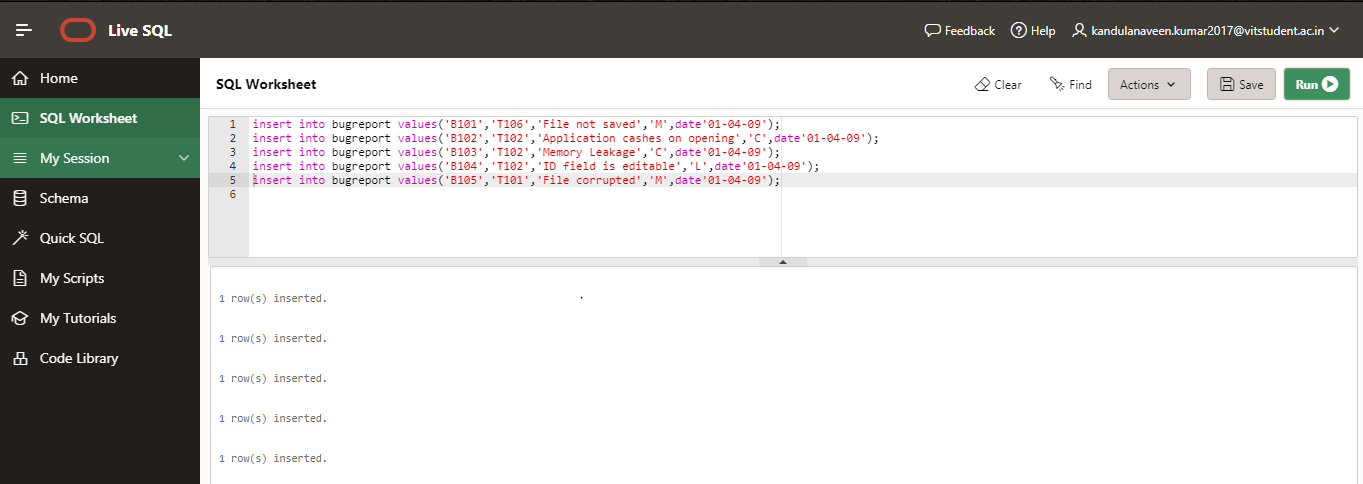
**insert into bugreport values('B101','T106','File not saved','M',date'01-04-09');**

**insert into bugreport values('B102','T102','Application cashes on opening','C',date'01-04-09');**

**insert into bugreport values('B103','T102','Memory Leakage','C',date'01-04-09');**

**insert into bugreport values('B104','T102','ID field is editable','L',date'01-04-09');**

**insert into bugreport values('B105','T101','File corrupted','M',date'01-04-09');**



**Bugfixallocation table:**

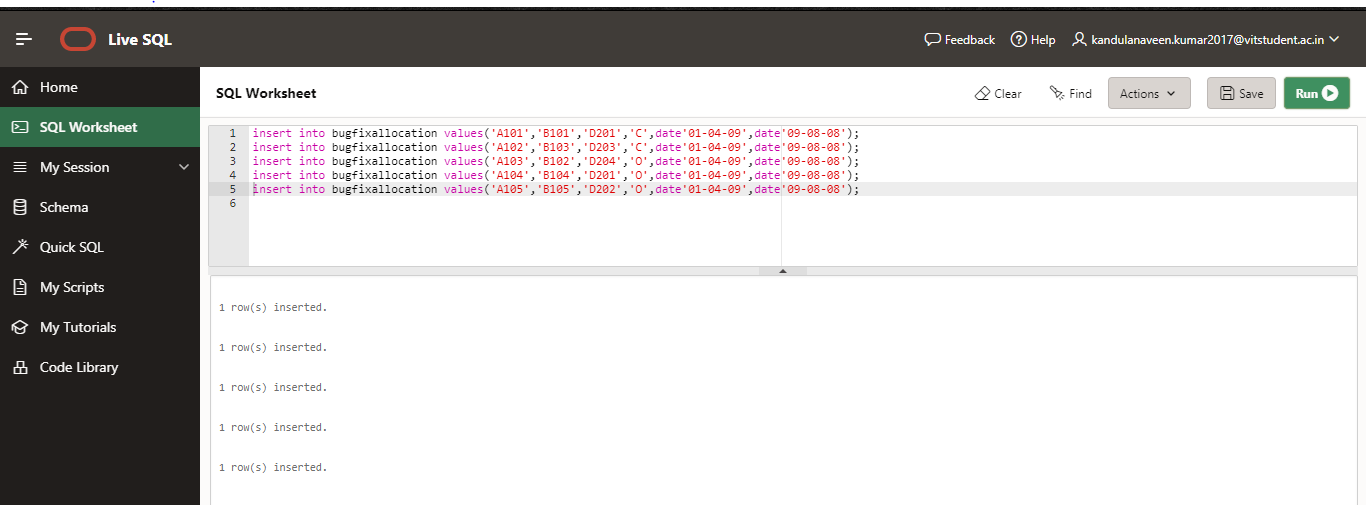
insert into bugfixallocation values('A101','B101','D201','C',date'01-04-09',date'09-08-08');

insert into bugfixallocation values('A102','B103','D203','C',date'01-04-09',date'09-08-08');

insert into bugfixallocation values('A103','B102','D204','O',date'01-04-09',date'09-08-08');

insert into bugfixallocation values('A104','B104','D201','O',date'01-04-09',date'09-08-08');

insert into bugfixallocation values('A105','B105','D202','O',date'01-04-09',date'09-08-08');

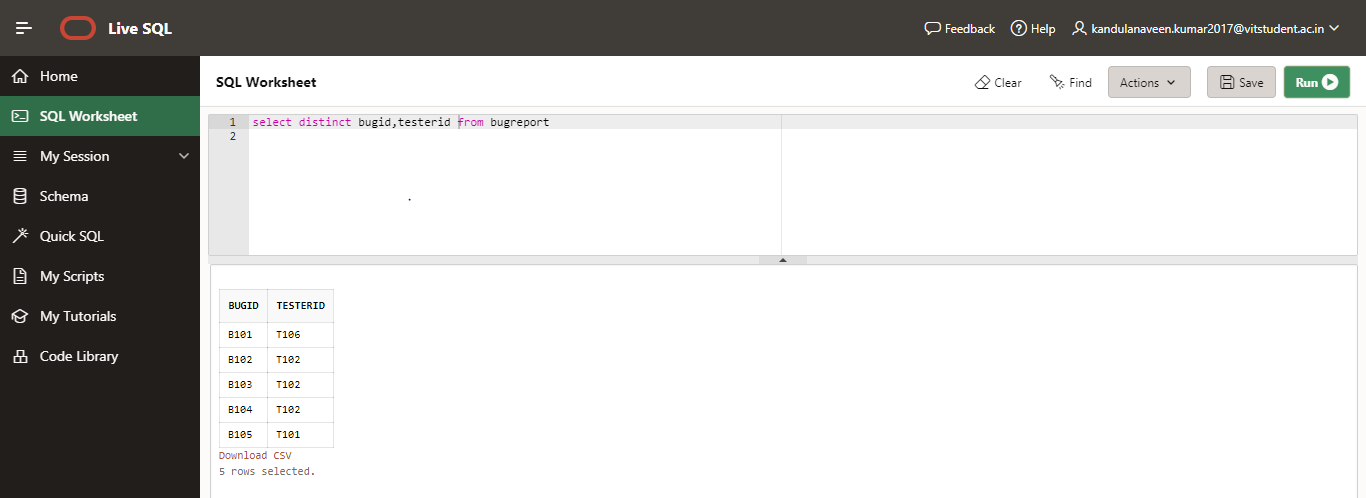


**WRITE QUERY for the following:**

1. For all the testers, display testerid and total number of bugs reported by them.

ANS:

select distinct bugid,testerid from bugreport

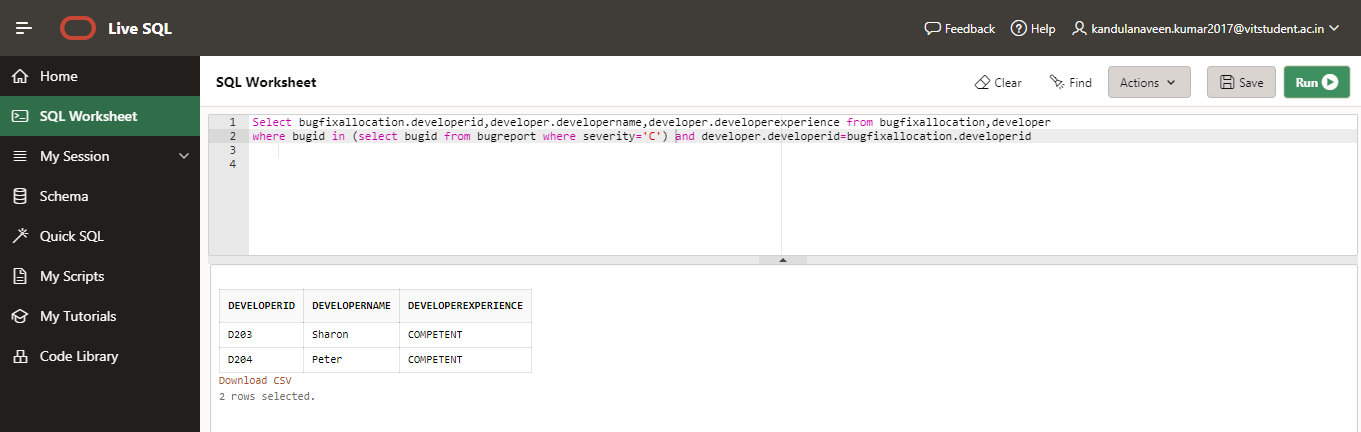


1. Display developerid, developername and developerexperience of developers who have been allocated bugs of severity ‘C’

ANS:

Select bugfixallocation.developerid,developer.developername,developer.developerexperience from bugfixallocation,developer

where bugid in (select bugid from bugreport where severity='C') and developer.developerid=bugfixallocation.developerid



1. Display the bug id, reported date and bug description of bugs status ‘Open’

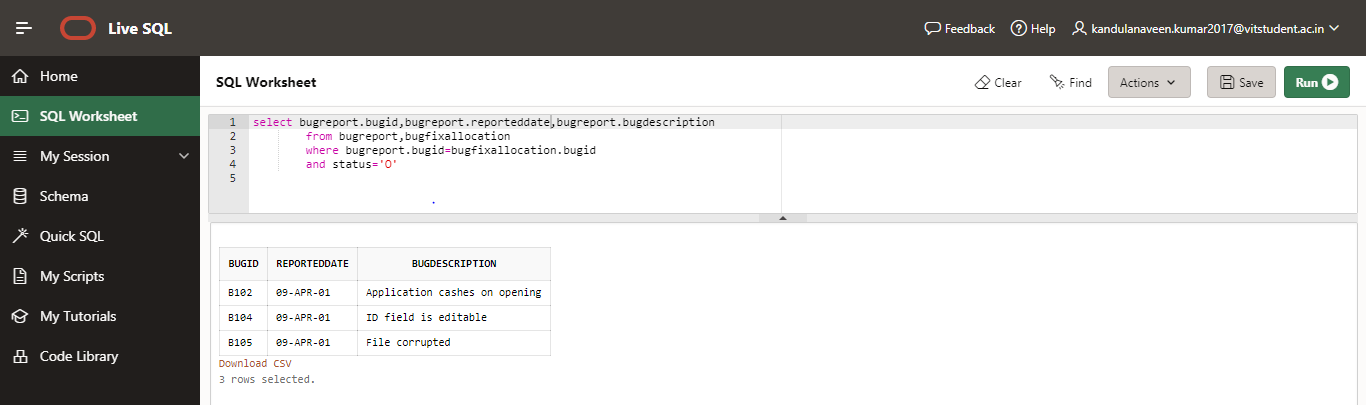
ANS:

select bugreport.bugid,bugreport.reporteddate,bugreport.bugdescription

from bugreport,bugfixallocation

where bugreport.bugid=bugfixallocation.bugid

and status='O'



1. Display the bugid, developerid, developername and identified priority of all ‘Closed’ bugs.

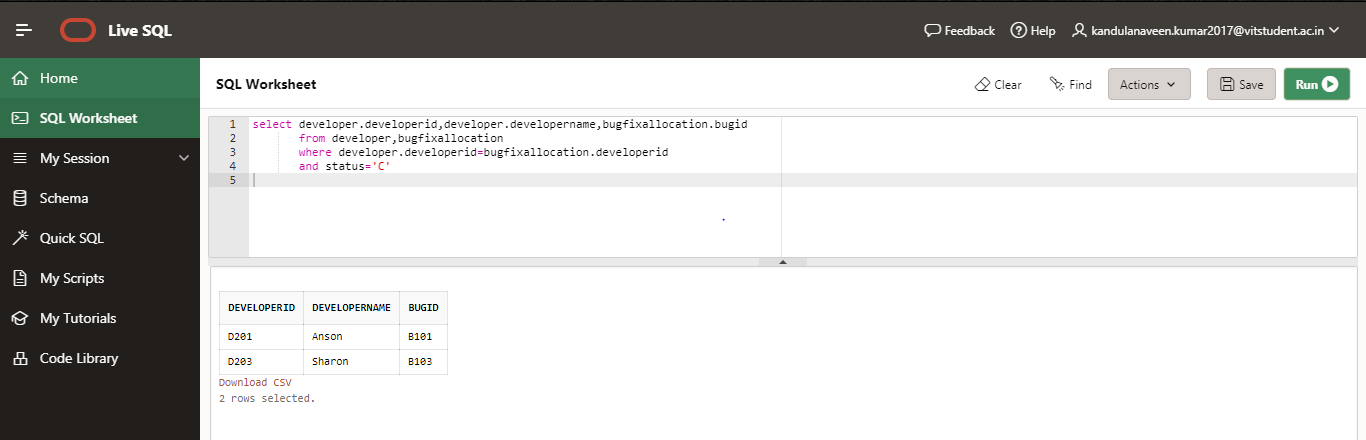
ANS:

select developer.developerid,developer.developername,bugfixallocation.bugid

from developer,bugfixallocation

where developer.developerid=bugfixallocation.developerid

and status='C'



1. Display the developer name, phone number, bug description of ‘Closed’ bugs

ANS:

select developer.developername,developer.phonenumber,bugreport.bugdescription from developer,bugreport

where bugreport.bugid in (select bugid from bugfixallocation where status='C' and developer.developerid=bugfixallocation.developerid)

