LONDON
METROPOLITAN
UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

CC5051NI Databases

100% Individual Coursework

Autumn 2024

Credit: 15 Semester Long Module

Student Name: Pooja Kumari Yadav

London Met ID: 23056290

Assignment Submission Date: 10 December

Word Count:

I confirm that I understand my coursework needs to be submitted online via My Second Teacher Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of contents

Introduction 5

Initial Entity relationship Model (ERD) 7

Identification of the entities and attributes 7

Initial Entity Relationship Diagram 10

1NF 11

2NF 12

3NF 14

Data dictionary 16

Result module 20

Final ERD 20

Implementation 21

Queries 29

Information queries 29

Transaction query 34

Dump file 37

Conclusion 38

References 38

Table of figures

Figure 1: Initial ERD 10
Figure 2 20
Figure 3 21
Figure 4 21
Figure 5 22
Figure 6 22
Figure 7 23
Figure 8 23
Figure 9 23
Figure 10 24
Figure 11 24
Figure 12 24
Figure 13 25
Figure 14 25
Figure 15 26
Figure 16 26
Figure 17 26
Figure 18 26
Figure 19 27
Figure 20 28
Figure 21 28
Figure 22 29
Figure 23 29
Figure 24 29
Figure 25 30
Figure 26 30
Figure 27 31
Figure 28 32
Figure 29 33
Figure 30 34
Figure 31 34
Figure 32 35
Figure 33 36
Figure 34 36
Figure 35 37

Table of tables

Table 1 16
Table 2 16
Table 3 17
Table 4 17
Table 5 17
Table 6 18
Table 7 18
Table 8 18
Table 9 19
Table 10 19
Table 11 19
Table 12 20

Introduction

Miss Mary is a founder and principal of Islington college which is established in 1996, which is a leading institution committed to providing high quality education in computing, networking, Artificial Intelligence, Multimedia and BBA. It helps to give the students ideas to prepare for global challenges which are used to point up innovation, flexibility and academic excellence. Miss Mary is the visionary founder and principal of the Islington College which is dedicated to enhancing the students and teacher experience throughout the innovation solution. It recognizes the demand of online education. Miss Mary proposes the development of the E-classroom Platform. The digital platform aims to design an efficient and structured educational environment which is used to integrate the students, teachers and academic resources.

Islington college used to operate a variety of academic programs such as BIT in computing, BIT in Networking, BIT in Multimedia, BIT in AI and Bachelor in BBA. These programs consist of multiple modules to each course which is carefully designed to deliver core knowledge and practical skills. Students are enrolled in one program at a time and each program's modules provide access to the resources, assignments and assessments. The teacher plays a vital role in delivering contents, evaluating the student's performance and maintaining academic integrity. Each module has their several assessments which are linked to the student's evaluations. The college also facilitates access to structured resources to assure progressive learning. The Teacher also used to enhance to make the module specific announcements to enhance communication. Miss Mary requirements for the E-classroom Platform to include some database system that will manage the students, teachers and their respective activities, also organize programs, modules and assessments effectively and also maintain the resources, academic records, assignments and announcements within their modules.

Business rules

- Each program consists of multiple compulsory modules.
- Modules consist of multiple programs.
- Students can register in only one program at a time.
- Program consist of multiple students at a time.
- Each module can be taught by one or as well as more teachers.
- Each module must include one or more assessments.
- Assessments attributes include ID, title and deadline, which is uniquely identified and associated with a single module.
- Each module contains multiple resources.
- Teachers can post announcements for specific modules.
- Announcements are uniquely identified and contain details such as ID, title and content.

Assumptions

- A student cannot be enrolled in more than one program at any given time.
- Each program consists solely of compulsory modules (no optional modules specified).
- Assessments must be explicitly tied to only one module.
- Resources within a module have unique attributes to differentiate them.
- Teachers can post multiple announcements per module, and announcements are tied to a single module.
- Each module requires at least one teacher but can have more if needed.
- The relationship between modules and resources is one-to-many (one module can have multiple resources).
- Announcements are module-specific and cannot be generalized across modules.

Initial Entity relationship Model (ERD)

Identification of the entities and attributes

Entities	**Attributes**
Student	Student_ID(PK), Student_Name, StudentDateofBirth, StudentEmail, StudentAddress
Program	Program_ID(PK), Program_Name, ProgramDescription

Module	Module_ID(PK), Module_Name, ModuleDuration, ModuleDescription, Teacher_ID, Teacher_Name, TeacherEmail, TeacherAddress, Resource_ID, Resource_Name, ResourceDescription, ResourceStatus, Announcement_ID, Announcement_Name, AnnouncementContent, Assessment_ID(PK), Assessment_Name, Assessment_status, AssessmentDeadline, AssessmentDuration, Result_ID, Result_status, Fullmarks, PassMarks ObtainedMarks
Teacher	Teacher_ID(PK), Teacher_Name, TeacherEmail,

	TeacherAddress
Resources	Resources_ID(PK), Resources_Name, ResourcesDescription, ResourceStatus
Announcement	Announcement_ID(PK), Announcement_Name, Announcementcontent
Assessment	Assessment_ID(PK), Assessment_Name, AssessmentDeadline, AssessmentDuration, Result_ID, FullMarks, PassMarks, ObtainedMarks

Initial Entity Relationship Diagram

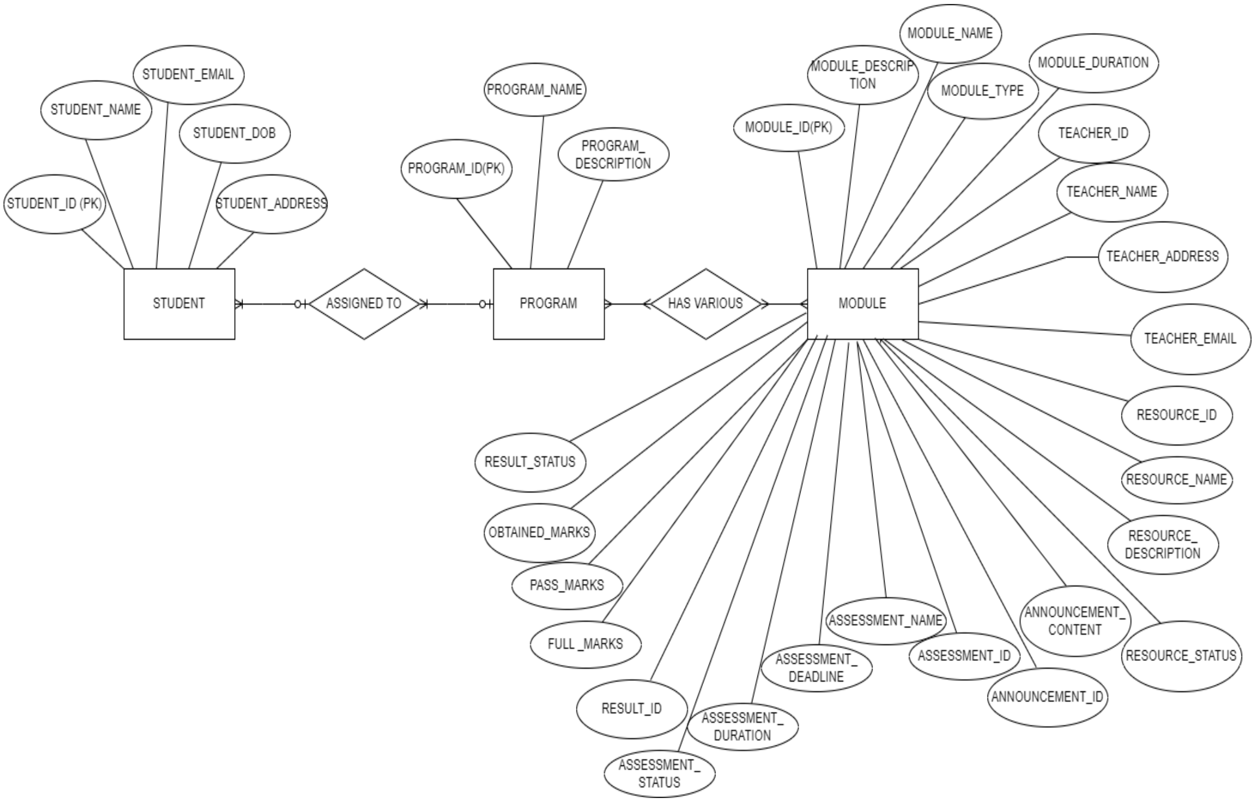


Figure 1: Initial ERD

Normalization

UNF

(Student_ID, StudentName, StudentAddress, StudentEmail, StudentDateOfBirth, Program_ID, Program name, ProgramDescription, {Module_ID, ModuleName, ModuleDescription, ModuleDuration,{Teacher_ID, TreacherName, TeacherAddress, TeacherEmail{Announcement_ID, Announcementcontent}}, {Resource_ID, ResourceName, ResourceDescription}, {Assessment_ID, AssessmentName, AssessmentDeadline, AssessmentDuration, PassMarks, FullMarks, Result_ID, ObtainedMarks,Result_status}})

1NF

Student-1 (Student_ID, StudentName, StudentEmail, StudentDateofBirth, Program_ID, ProgramName, ProgramDescription)

Module-1 (Module_ID, Student_ID*, ModuleName, ModuleType, ModuleDescription, ModuleDuration)

Teacher-1 (Teacher_ID , Module_ID*,Student_ID*, TeacherName, TeacherAddress, TeacherEmail)

Resource-1 (Resource_ID, Module_ID*, Student_ID*, ResourceName, ResourceDescription)

Announcement-1 (Announcement_ID, Module_ID*,Teacher_ID*, Student_ID*, AnnouncementContent)

Assessment-1 (Assessment_ID, AssessmentName, AssessmentDeadline, AssessmentDuration, Result_ID, ObtainedMarks, PassMarks, FullMarks, Result_Status,Student_ID*, Module_ID*)

2NF

Student Module → X

Student-2 (Student_ID, StudentName, StudentEmail, StudentDateofBirth, Program_ID, ProgramName, ProgramDescription)

Module_ID –> ModuleName

Module_ID Student_ID → X

Student_ID → X

Module -2 ((Module_ID, ModuleName, ModuleType, ModuleDescription, ModuleDuration)

Teacher_ID Module_ID → X

Teacher_ID →TeacherName, TeacherAddress, TeacherEmail

Module_ID → X

Teacher-2(Teacher_ID, TeacherName , TeacherAddress, TeacherEmail)

Module Teacher-2(Teacher ID*, Module_ID*, Student_ID*)

Module_ID*, Resource_ID, Student_ID * → X

Resource _ID* –> ResourceName, ResourceDescription, ResourceStatus

Module_ID* → X

Student_ID * → X

Resource_ID Module_ID –> ResourceName

Resource-2 (Resource_ID, ResourceName, ResourceDescription)

ResourceModule-2 (Resource_ID*, Module_ID*, Student_ID*, ResourceStatus)

Announcement_ID, Module_ID*, Student_ID* → X

Announcement_ID → AnnouncementName, AnnouncementContent

Module_ID* → X

Student_ID * → X

Announcement-2 (Announcement_ID, AnnouncementContent)

Announcement Module-2 (Announcement_ID*, Module_ID*, Student_ID*)

Assessment-2 (Assessment_ID, AssessmentName, AssessmentDeadline, AssessmentDuration, Result_ID, ObtainedMarks, PassMarks, FullMarks, Student_ID*, Module_ID*)

Final 2NF

Student-2 (Student_ID, StudentName, StudentEmail, StudentDateofBirth, Program_ID, ProgramName, ProgramDescription)

Module -2 ((Module_ID, ModuleName, ModuleType, ModuleDescription, ModuleDuration)

Student_module-2 (Student_ID*, Module_ID*)

Teacher-2(Teacher_ID, TeacherName , TeacherAddress, TeacherEmail)

Module Teacher-2(Teacher ID*, Module_ID*, Student_ID*)

Resource-2 (Resource_ID, ResourceName, ResourceDescription)

ResourceModule-2 (Resource_ID*, Module_ID*, Student_ID*, ResourceStatus)

Announcement-2 (Announcement_ID, AnnouncementContent)

Announcement Module-2 (Announcement_ID*, Module_ID*, Student_ID*, teacher_ID*)

Assessment-2 (Assessment_ID, AssessmentName, AssessmentDeadline, AssessmentDuration, Result_ID, ObtainedMarks, PassMarks, FullMarks, Student_ID*, Module_ID*)

3NF

Student_ID -> Student_ID, StudentName, StudentEmail, StudentDateofBirth

Student_ID -> Program_ID, ProgramName, ProgramDescription

Student-3 (Student_ID, StudentName, StudentEmail, StudentDateofBirth, Program_ID, ProgramName, ProgramDescription)

Program-3(Program_ID, ProgramName, ProgramDescription)

Module_ID -> Module_ID, ModuleName, ModuleType, ModuleDescription, ModuleDuration

Module-3 (Module_ID, ModuleName, ModuleType, ModuleDescription, ModuleDuration)

Teacher -> Teacher_ID, TeacherName, TeacherAddress, TeacherEmail

Teacher Module -> Module_ID*, Teacher_ID*

Teacher-3 (Teacher_ID, TeacherName, TeacherAddress, TeacherEmail)

Teacher Module-3 (Module_ID*, Teacher_ID*)

Resource -> Resource_ID, ResourceName, ResourceDescription, ResourceStatus

ResourceModule -> Resource_ID*, Module_ID*, Student_ID*

Resources (Resource_ID, ResourceName, ResourceDescription, ResourceStatus)

ResourceModule (Resource_ID*, Module_ID*, Student_ID*)

Announcement -> Announcement_ID, AnnouncementContent

Announcement Module -> Announcement_ID*, Module_ID*, Student_ID*

Announcement-3 (Announcement_ID, Module_ID*, Student_ID*, AnnouncementContent)

Announcement Module-3(Announcement_ID*, Module_ID*, Student_ID*)

Assessment -> Assessment_ID, AssessmentName, AssessmentDeadline, AssessmentDuration, Module_ID*

Result -> Result_ID, Assessment_ID*, Student_ID*, ObtainedMarks, PassMarks, FullMarks

Result_Module -> Result_ID, Student_ID

Assessment-3 (Assessment_ID, AssessmentName, AssessmentDeadline, AssessmentDuration, Module_ID*, Resource_ID*, Teacher_ID*)

Result-3 (Result_ID, Assessment_ID*, Student_ID*, ObtainedMarks, PassMarks, FullMarks)

Result_Module -3 (Result_ID*, Student_ID*)

Final 3NF

Student-3 (Student_ID, StudentName, StudentEmail, StudentDateofBirth, Program_ID*)

Program-3(Program_ID, ProgramName, ProgramDescription)

Module-3 (Module_ID, ModuleName, ModuleType, ModuleDescription, ModuleDuration)

Module Student-3(Module_ID*, Student_ID*)

Teacher-3 (Teacher_ID, TeacherName, TeacherAddress, TeacherEmail)

Teacher module-3 (Module_ID*, Teacher_ID*,Student_ID*)

Resources (Resource_ID, ResourceName, ResourceDescription, ResourceStatus)

ResourceModule (Resource_ID*, Module_ID*, Student_ID*)

Announcement-3 (Announcement_ID, Module_ID*, Student_ID*, Teacher_ID*, AnnouncementContent)

Announcement Module-3(Announcement_ID*, Module_ID*, Student_ID*, Teacher_ID*)

Assessment-3 (Assessment_ID, AssessmentName, AssessmentDeadline, AssessmentDuration, Module_ID*, Resource_ID*, Teacher_ID*)

Result-3 (Result_ID, Assessment_ID*, Student_ID*, ObtainedMarks, PassMarks, FullMarks, Result_Status)

Result_module -3 (Result_ID*, Module_ID*);

Data dictionary

Student

Table 1

S.N	Attribute name	Data Type	Size	Constraints	Composite constraints
1.	Student_ID	Number	10	Primary key, not null, unique	-
2.	StudentName	Character	30	Not null	-
3.	StudentEmail	Character	30	Not null, unique	-
4.	StudentAddress	Character	50	Not null	-
5.	StudentDateOfBirth	Date	-	Not null	
6.	Program_ID	Number	10	Foreign key references Program(Program_ID), not null	Primary key

Program

Table 2

S.N	Attribute name	Data Type	Size	constraints
1	Program_ID	Number	10	Primary key, not null
2	ProgramName	Character	30	Not null
3	ProgramDescription	Character	30	Null

Module

Table 3

S.N	Attribute Name	Data Type	Size	Constraint
1	Module_ID	Number	10	Primary key, not null
2	ModuleName	Character	20	Not null
3	ModuleType	Character	50	Not null
4	ModuleDescription	Character	50	Null
5	ModuleDuration	Number	4	Not null

Teacher

Table 4

S.N	Attribute name	Data Type	Size	Constraints
1	Teacher_ID	Number	10	Primary key
2	TeacherName	Character	10	Not null
3	TeacherAddress	Character	15	Null
4	TeacherEmail	Character	50	Unique, not null

Teacher Module

Table 5

S.N	Attribute name	Data type	Size	Constraints	Composite constraints
1	Teacher_ID	Number	10	Foreign key, notnull	Primary key
2	Module_ID	Number	10	Foreign key, notnull	Primary key

Resources

Table 6

S.N	Attribute Name	Data Type	Size	Constraints
1	Resource_ID	Number	10	Primary key, not null
2	ResourceName	Character	30	Not null
3	ResourceDescription	Character	-	Null
4	ResourceStatus	Character	20	Not null

Resource Module

Table 7

S.N	Attribute name	Data Type	Size	Constraints	Composite constraints
1	Resource_ID	Number	10	Foreign key, not null	Primary key
2	Module_ID	Number	10	Foreign key, not null	Primary key
3	Student_ID	Number	10	Foreign key, not null	Primary key

Announcement

Table 8

S.N	Attribute Name	Data Type	Size	Constraints
1	Announcement_ID	Number	10	Primary key, no null, unique
2	AnnouncementContent	Character	-	Not null

Announcement Module

Table 9

S.N	Attribute name	Data Type	Size	Constraints	Composite constraints
1	Announecement_ID	Number	10	Foreign key, not null	Primary key
2	Module_ID	Number	10	Foreign key, not null	Primary key
3	Student_ID	Number	10	Foreign key , not null	Primary key

Assessment

Table 10

S.N	Attribute Name	Data type	Size	Constraints
1	Assessment_ID	Number	10	Primary key, not null
2	AssessmentName	Character	30	Not null
3	AssessmentDeadline	Number	-	Not null
4	AssessmentDuration	Number	4	Not null(in hours)

Result

Table 11

S.N	Attribute Name	Data type	Size	Constraints	Composite constraints
1	Result_ID	Number	10	Primary key, not null	Primary key
2	Assessment_ID	Number	10	Foreign key, not null	Primary key
3	Student_ID	Number	10	Foreign key, not null	Primary key
4	FullMarks	Number	4	Not null	
5	PassMarks	Number	4	Not null	
6	ObtainedMarks	Number	4	Not Null	

Result module

Table 12

SN	Attribute Name	Data type	Size	constraints	Composite constraints
1	Result_ID	Number	10	Foreign key, not null	Primary key
2	Module_ID	number	10	Foreign key,not null	Primary key

Final ERD

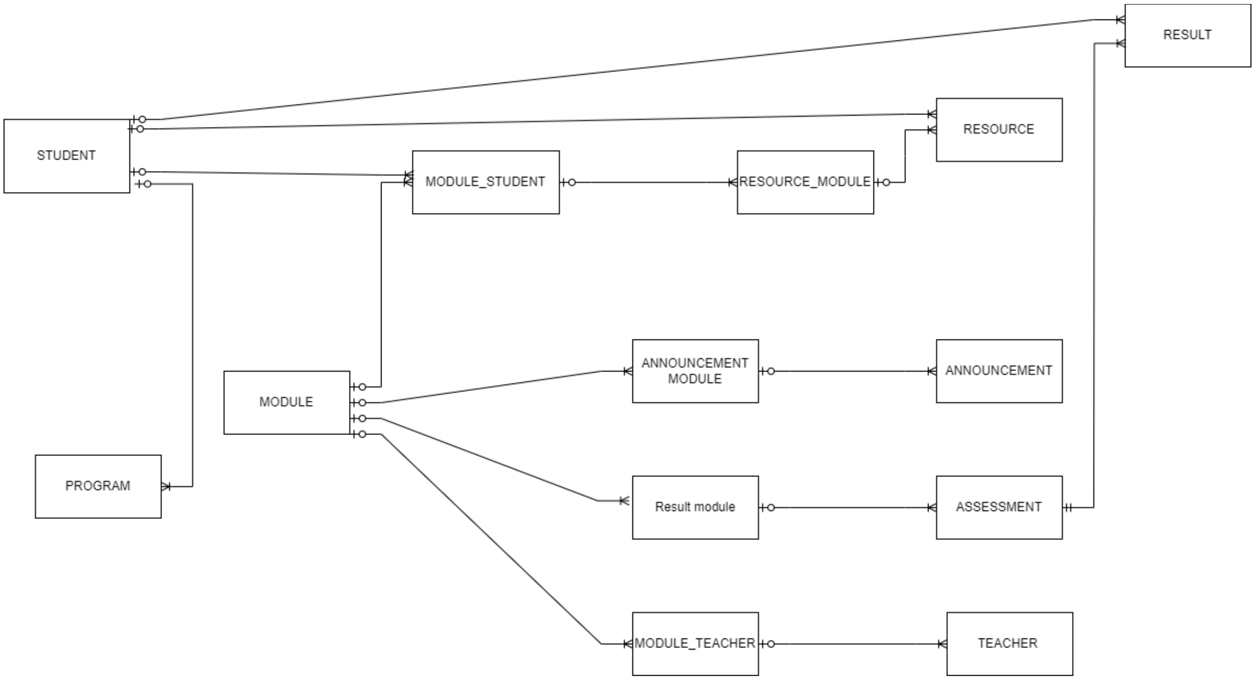
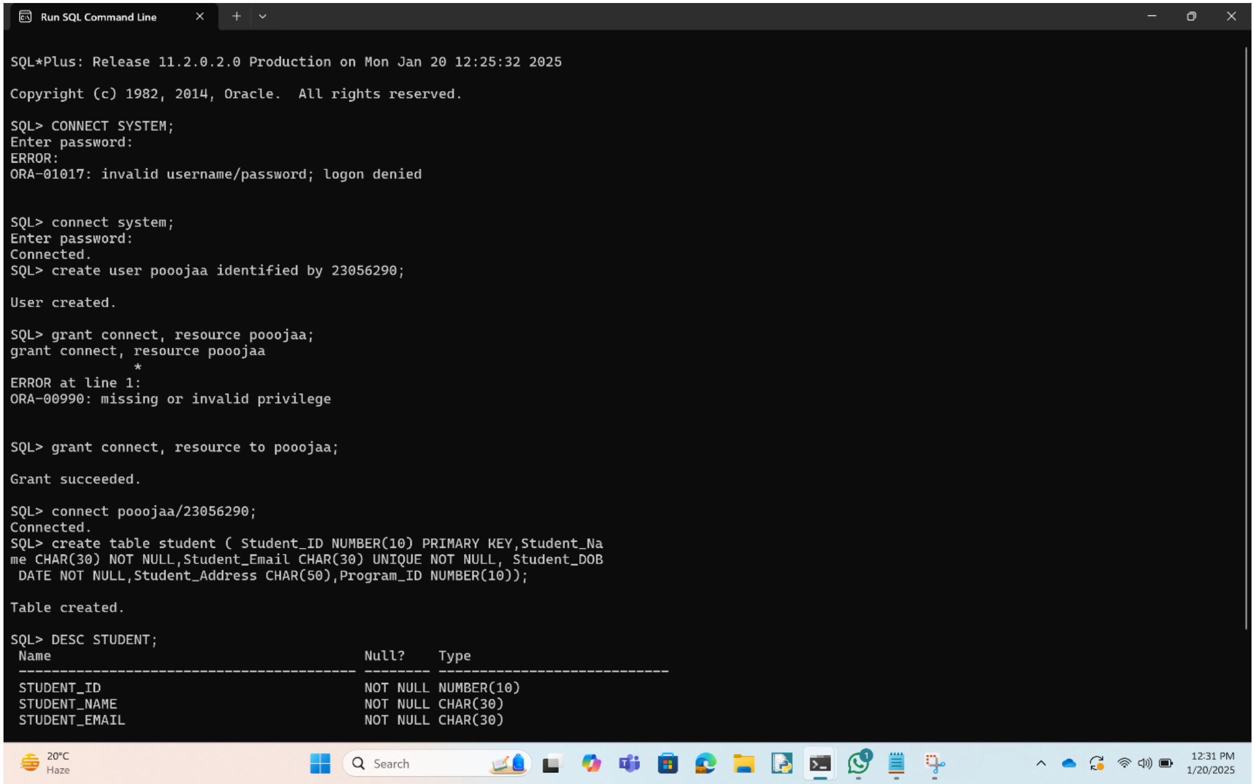


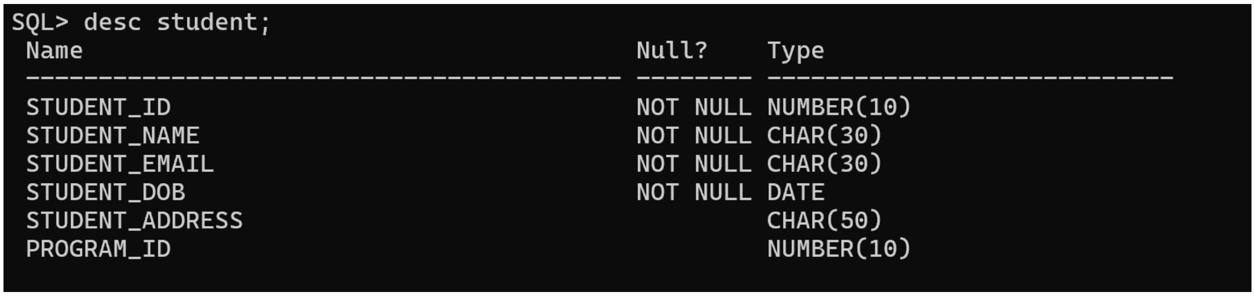
Figure 2

Implementation



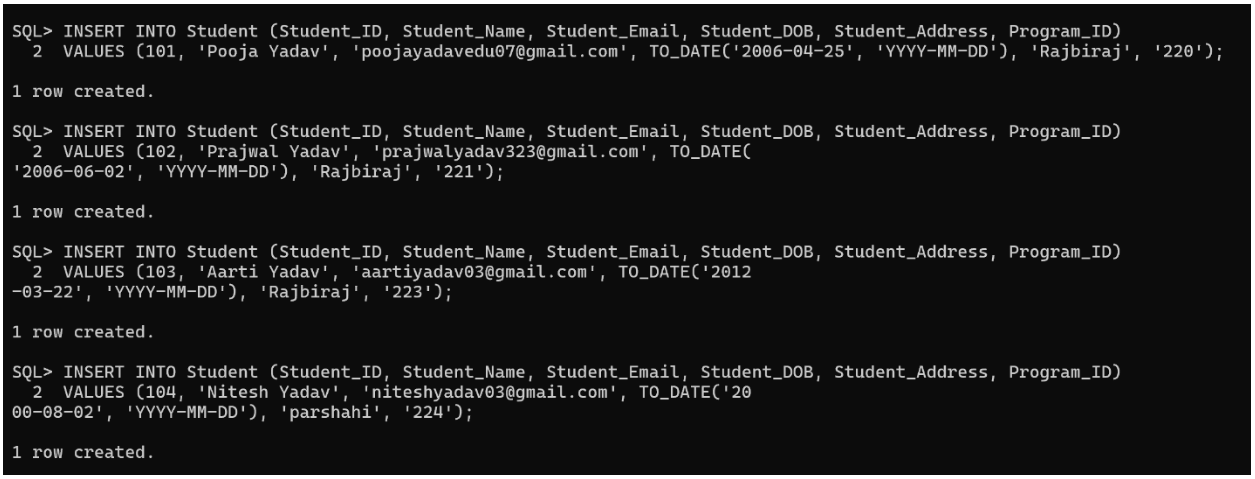
```
SQL*Plus: Release 11.2.0.2.0 Production on Mon Jan 20 12:25:32 2025

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> CONNECT SYSTEM;
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied


SQL> connect system;
Enter password:
Connected.
SQL> create user pooojaa identified by 23056290;

User created.

SQL> grant connect, resource pooojaa;
grant connect, resource pooojaa
                        *
ERROR at line 1:
ORA-00990: missing or invalid privilege


SQL> grant connect, resource to pooojaa;

Grant succeeded.

SQL> connect pooojaa/23056290;
Connected.
SQL> create table student ( Student_ID NUMBER(10) PRIMARY KEY,Student_Na
me CHAR(30) NOT NULL,Student_Email CHAR(30) UNIQUE NOT NULL, Student_DOB
 DATE NOT NULL,Student_Address CHAR(50),Program_ID NUMBER(10));

Table created.

SQL> DESC STUDENT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STUDENT_ID                                NOT NULL NUMBER(10)
 STUDENT_NAME                              NOT NULL CHAR(30)
 STUDENT_EMAIL                             NOT NULL CHAR(30)
```

Figure 3



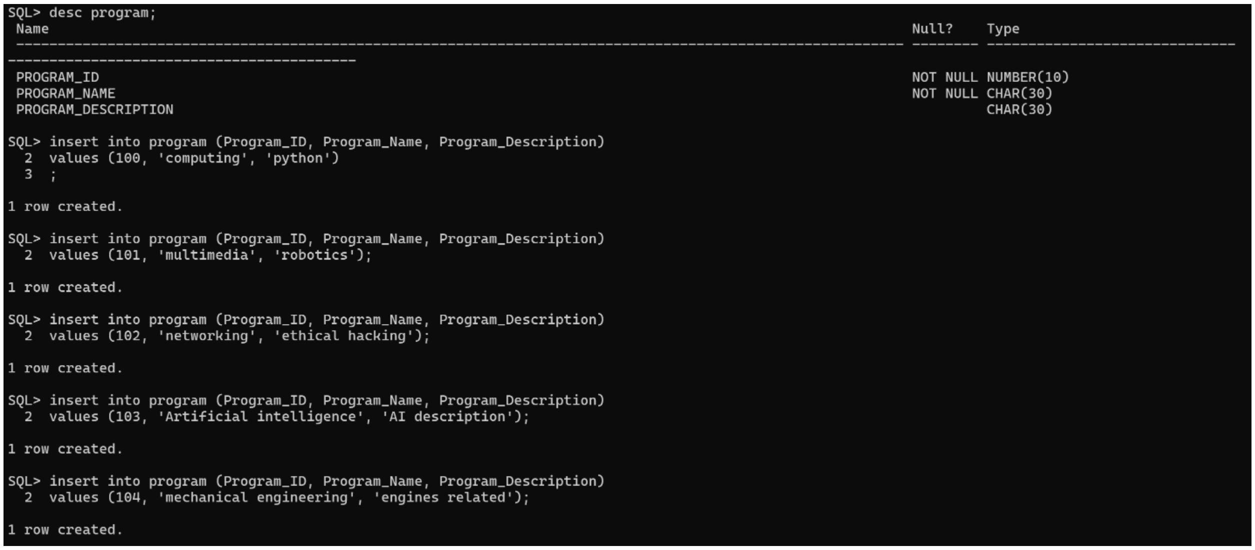
```
SQL> desc student;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STUDENT_ID                                NOT NULL NUMBER(10)
 STUDENT_NAME                              NOT NULL CHAR(30)
 STUDENT_EMAIL                             NOT NULL CHAR(30)
 STUDENT_DOB                               NOT NULL DATE
 STUDENT_ADDRESS                                    CHAR(50)
 PROGRAM_ID                                         NUMBER(10)
```

Figure 4



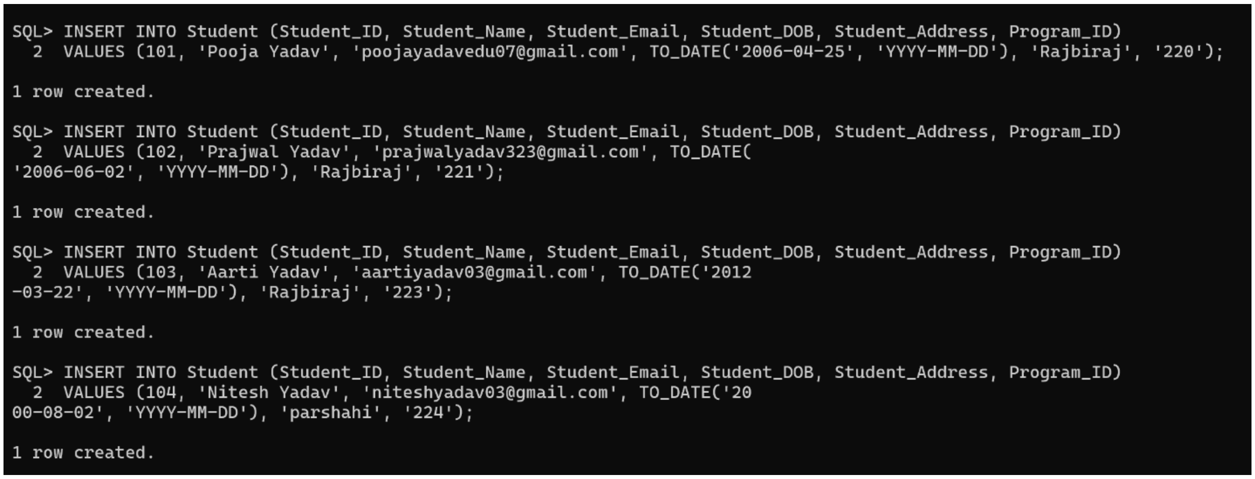
```
SQL> INSERT INTO Student (Student_ID, Student_Name, Student_Email, Student_DOB, Student_Address, Program_ID)
  2  VALUES (101, 'Pooja Yadav', 'poojayadavedu07@gmail.com', TO_DATE('2006-04-25', 'YYYY-MM-DD'), 'Rajbiraj', '220');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Student_Email, Student_DOB, Student_Address, Program_ID)
  2  VALUES (102, 'Prajwal Yadav', 'prajwalyadav323@gmail.com', TO_DATE(
'2006-06-02', 'YYYY-MM-DD'), 'Rajbiraj', '221');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Student_Email, Student_DOB, Student_Address, Program_ID)
  2  VALUES (103, 'Aarti Yadav', 'aartiyadav03@gmail.com', TO_DATE('2012
-03-22', 'YYYY-MM-DD'), 'Rajbiraj', '223');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Student_Email, Student_DOB, Student_Address, Program_ID)
  2  VALUES (104, 'Nitesh Yadav', 'niteshyadav03@gmail.com', TO_DATE('20
00-08-02', 'YYYY-MM-DD'), 'parshahi', '224');

1 row created.
```

Figure 5



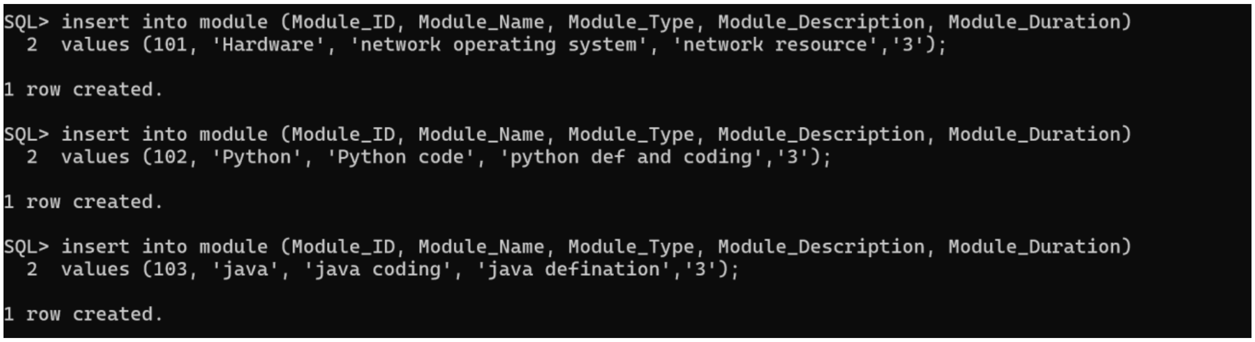
```
SQL> desc program;
 Name                                                                                                    Null?    Type
 ------------------------------------------------------------------------------------------------------- -------- -----------------------------
-----------------------------------------
 PROGRAM_ID                                                                                              NOT NULL NUMBER(10)
 PROGRAM_NAME                                                                                            NOT NULL CHAR(30)
 PROGRAM_DESCRIPTION                                                                                              CHAR(30)

SQL> insert into program (Program_ID, Program_Name, Program_Description)
  2  values (100, 'computing', 'python')
  3  ;

1 row created.

SQL> insert into program (Program_ID, Program_Name, Program_Description)
  2  values (101, 'multimedia', 'robotics');

1 row created.

SQL> insert into program (Program_ID, Program_Name, Program_Description)
  2  values (102, 'networking', 'ethical hacking');

1 row created.

SQL> insert into program (Program_ID, Program_Name, Program_Description)
  2  values (103, 'Artificial intelligence', 'AI description');

1 row created.

SQL> insert into program (Program_ID, Program_Name, Program_Description)
  2  values (104, 'mechanical engineering', 'engines related');

1 row created.
```

Figure 6



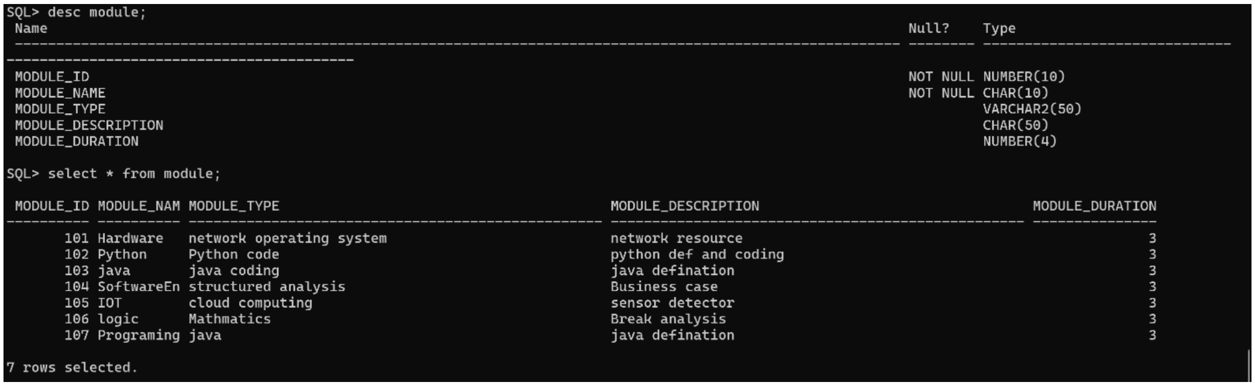
```
SQL> INSERT INTO Student (Student_ID, Student_Name, Student_Email, Student_DOB, Student_Address, Program_ID)
  2  VALUES (101, 'Pooja Yadav', 'poojayadavedu07@gmail.com', TO_DATE('2006-04-25', 'YYYY-MM-DD'), 'Rajbiraj', '220');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Student_Email, Student_DOB, Student_Address, Program_ID)
  2  VALUES (102, 'Prajwal Yadav', 'prajwalyadav323@gmail.com', TO_DATE(
'2006-06-02', 'YYYY-MM-DD'), 'Rajbiraj', '221');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Student_Email, Student_DOB, Student_Address, Program_ID)
  2  VALUES (103, 'Aarti Yadav', 'aartiyadav03@gmail.com', TO_DATE('2012
-03-22', 'YYYY-MM-DD'), 'Rajbiraj', '223');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Student_Email, Student_DOB, Student_Address, Program_ID)
  2  VALUES (104, 'Nitesh Yadav', 'niteshyadav03@gmail.com', TO_DATE('20
00-08-02', 'YYYY-MM-DD'), 'parshahi', '224');

1 row created.
```

Figure 7



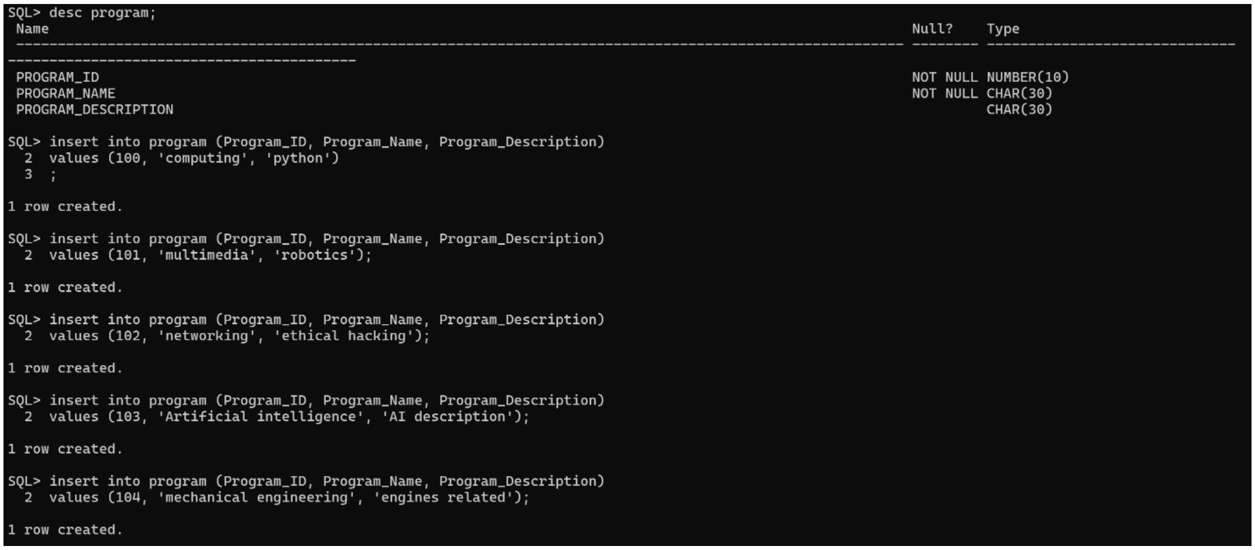
```
SQL> insert into module (Module_ID, Module_Name, Module_Type, Module_Description, Module_Duration)
  2  values (101, 'Hardware', 'network operating system', 'network resource','3');

1 row created.

SQL> insert into module (Module_ID, Module_Name, Module_Type, Module_Description, Module_Duration)
  2  values (102, 'Python', 'Python code', 'python def and coding','3');

1 row created.

SQL> insert into module (Module_ID, Module_Name, Module_Type, Module_Description, Module_Duration)
  2  values (103, 'java', 'java coding', 'java defination','3');

1 row created.
```

Figure 8



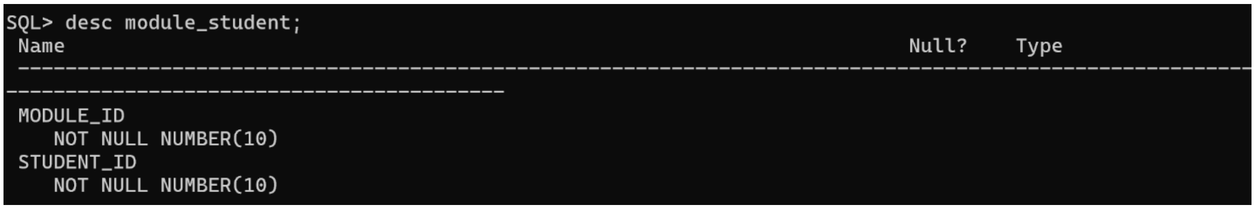
```
SQL> desc module;
 Name                                                                                                          Null?    Type
 ------------------------------------------------------------------------------------------------------------ -------- ------------------------------
------------------------------------------
 MODULE_ID                                                                                                     NOT NULL NUMBER(10)
 MODULE_NAME                                                                                                   NOT NULL CHAR(10)
 MODULE_TYPE                                                                                                            VARCHAR2(50)
 MODULE_DESCRIPTION                                                                                                     CHAR(50)
 MODULE_DURATION                                                                                                        NUMBER(4)

SQL> select * from module;

 MODULE_ID MODULE_NAM MODULE_TYPE                                        MODULE_DESCRIPTION                                 MODULE_DURATION
---------- ---------- -------------------------------------------------- -------------------------------------------------- ---------------
       101 Hardware   network operating system                           network resource                                                 3
       102 Python     Python code                                        python def and coding                                            3
       103 java       java coding                                        java defination                                                  3
       104 SoftwareEn structured analysis                                Business case                                                    3
       105 IOT        cloud computing                                    sensor detector                                                  3
       106 logic      Mathmatics                                         Break analysis                                                   3
       107 Programing java                                               java defination                                                  3

7 rows selected.
```

Figure 9



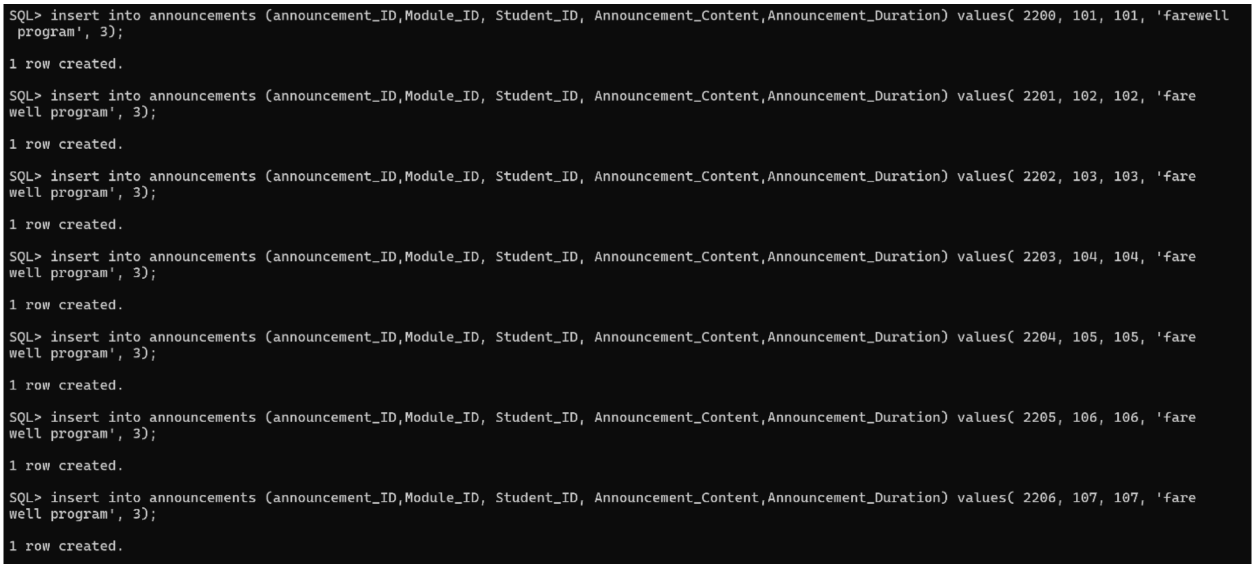
```
SQL> desc program;
 Name                                                                                                          Null?    Type
 ------------------------------------------------------------------------------------------------------------- -------- ------------------------------
------------------------------------------
 PROGRAM_ID                                                                                                    NOT NULL NUMBER(10)
 PROGRAM_NAME                                                                                                  NOT NULL CHAR(30)
 PROGRAM_DESCRIPTION                                                                                                    CHAR(30)

SQL> insert into program (Program_ID, Program_Name, Program_Description)
  2  values (100, 'computing', 'python')
  3  ;

1 row created.

SQL> insert into program (Program_ID, Program_Name, Program_Description)
  2  values (101, 'multimedia', 'robotics');

1 row created.

SQL> insert into program (Program_ID, Program_Name, Program_Description)
  2  values (102, 'networking', 'ethical hacking');

1 row created.

SQL> insert into program (Program_ID, Program_Name, Program_Description)
  2  values (103, 'Artificial intelligence', 'AI description');

1 row created.

SQL> insert into program (Program_ID, Program_Name, Program_Description)
  2  values (104, 'mechanical engineering', 'engines related');

1 row created.
```

Figure 10



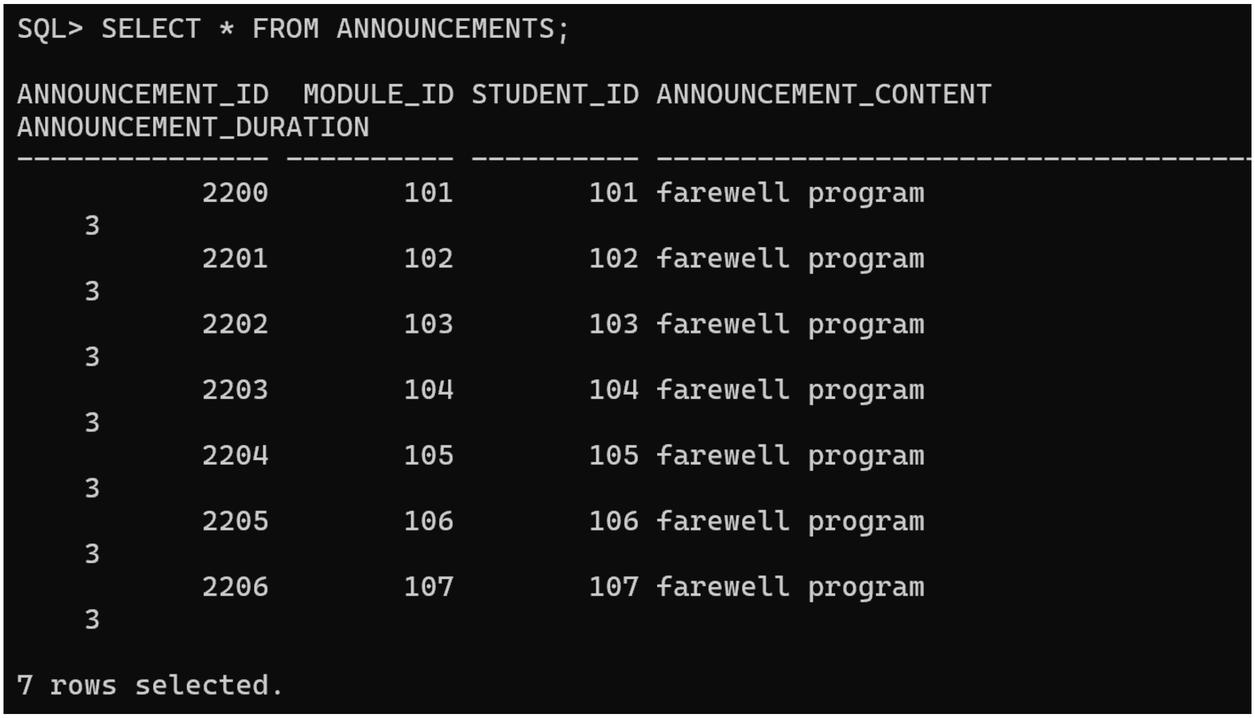
```
SQL> desc module_student;
 Name                                                                          Null?    Type
 -----------------------------------------------------------------------------------------------
------------------------------------------
 MODULE_ID
    NOT NULL NUMBER(10)
 STUDENT_ID
    NOT NULL NUMBER(10)
```

Figure 11



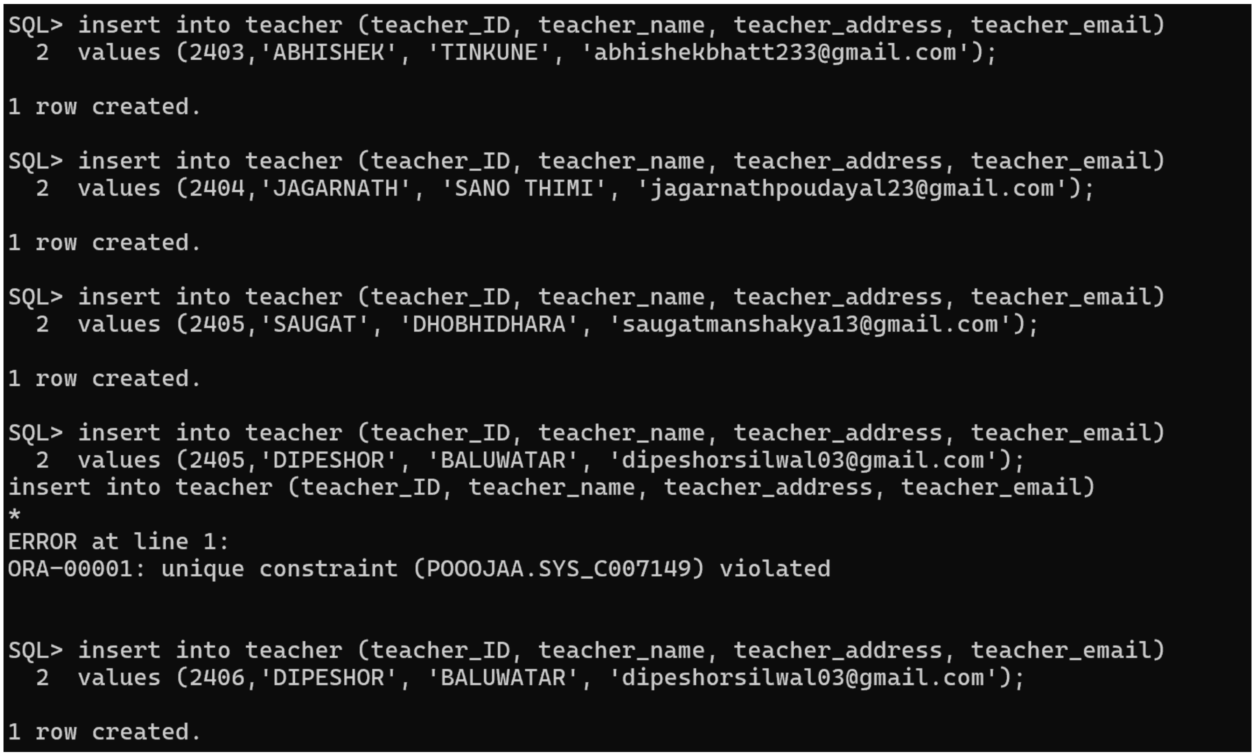
```
SQL> insert into announcements (announcement_ID,Module_ID, Student_ID, Announcement_Content,Announcement_Duration) values( 2200, 101, 101, 'farewell
 program', 3);

1 row created.

SQL> insert into announcements (announcement_ID,Module_ID, Student_ID, Announcement_Content,Announcement_Duration) values( 2201, 102, 102, 'fare
well program', 3);

1 row created.

SQL> insert into announcements (announcement_ID,Module_ID, Student_ID, Announcement_Content,Announcement_Duration) values( 2202, 103, 103, 'fare
well program', 3);

1 row created.

SQL> insert into announcements (announcement_ID,Module_ID, Student_ID, Announcement_Content,Announcement_Duration) values( 2203, 104, 104, 'fare
well program', 3);

1 row created.

SQL> insert into announcements (announcement_ID,Module_ID, Student_ID, Announcement_Content,Announcement_Duration) values( 2204, 105, 105, 'fare
well program', 3);

1 row created.

SQL> insert into announcements (announcement_ID,Module_ID, Student_ID, Announcement_Content,Announcement_Duration) values( 2205, 106, 106, 'fare
well program', 3);

1 row created.

SQL> insert into announcements (announcement_ID,Module_ID, Student_ID, Announcement_Content,Announcement_Duration) values( 2206, 107, 107, 'fare
well program', 3);

1 row created.
```

Figure 12



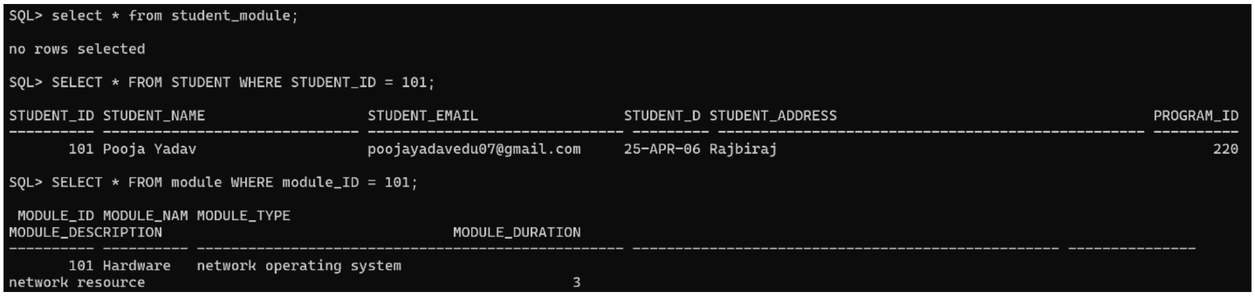
```
SQL> SELECT * FROM ANNOUNCEMENTS;

ANNOUNCEMENT_ID  MODULE_ID STUDENT_ID ANNOUNCEMENT_CONTENT
ANNOUNCEMENT_DURATION
--------------- ---------- ---------- -------------------------------------
           2200        101        101 farewell program
    3
           2201        102        102 farewell program
    3
           2202        103        103 farewell program
    3
           2203        104        104 farewell program
    3
           2204        105        105 farewell program
    3
           2205        106        106 farewell program
    3
           2206        107        107 farewell program
    3

7 rows selected.
```

Figure 13



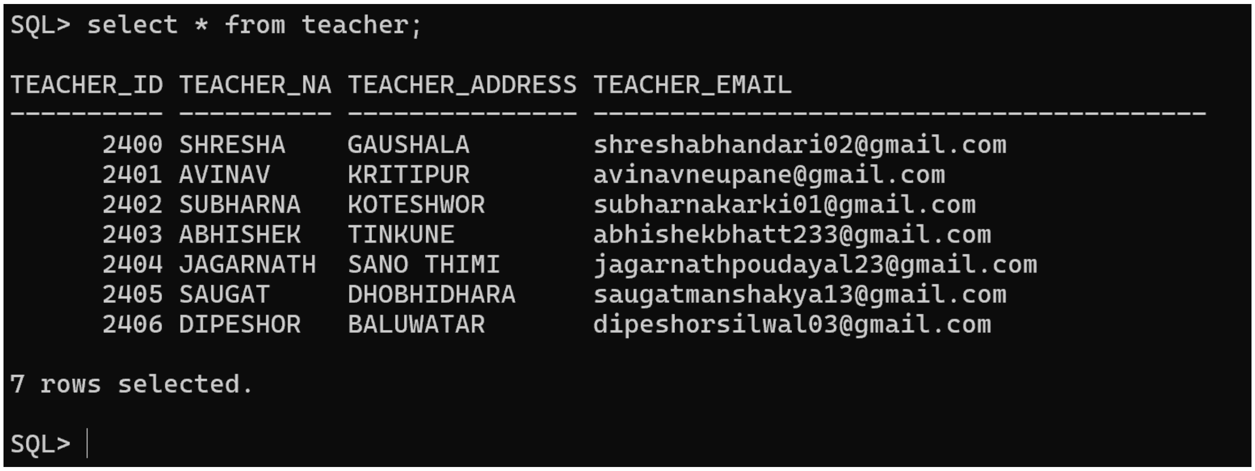
```
SQL> insert into teacher (teacher_ID, teacher_name, teacher_address, teacher_email)
  2  values (2403,'ABHISHEK', 'TINKUNE', 'abhishekbhatt233@gmail.com');

1 row created.

SQL> insert into teacher (teacher_ID, teacher_name, teacher_address, teacher_email)
  2  values (2404,'JAGARNATH', 'SANO THIMI', 'jagarnathpoudayal23@gmail.com');

1 row created.

SQL> insert into teacher (teacher_ID, teacher_name, teacher_address, teacher_email)
  2  values (2405,'SAUGAT', 'DHOBHIDHARA', 'saugatmanshakya13@gmail.com');

1 row created.

SQL> insert into teacher (teacher_ID, teacher_name, teacher_address, teacher_email)
  2  values (2405,'DIPESHOR', 'BALUWATAR', 'dipeshorsilwal03@gmail.com');
insert into teacher (teacher_ID, teacher_name, teacher_address, teacher_email)
*
ERROR at line 1:
ORA-00001: unique constraint (POOOJAA.SYS_C007149) violated


SQL> insert into teacher (teacher_ID, teacher_name, teacher_address, teacher_email)
  2  values (2406,'DIPESHOR', 'BALUWATAR', 'dipeshorsilwal03@gmail.com');

1 row created.
```

Figure 14



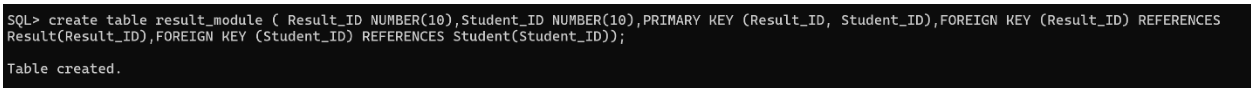
```
SQL> select * from student_module;

no rows selected

SQL> SELECT * FROM STUDENT WHERE STUDENT_ID = 101;

STUDENT_ID STUDENT_NAME                   STUDENT_EMAIL                  STUDENT_D STUDENT_ADDRESS                                    PROGRAM_ID
---------- ------------------------------ ------------------------------ --------- -------------------------------------------------- ----------
       101 Pooja Yadav                    poojayadavedu07@gmail.com      25-APR-06 Rajbiraj                                                  220

SQL> SELECT * FROM module WHERE module_ID = 101;

 MODULE_ID MODULE_NAM MODULE_TYPE
MODULE_DESCRIPTION                                 MODULE_DURATION
---------- ---------- -------------------------------------------------- -------------------------------------------------- ---------------
       101 Hardware   network operating system
network resource                                                 3
```

Figure 15



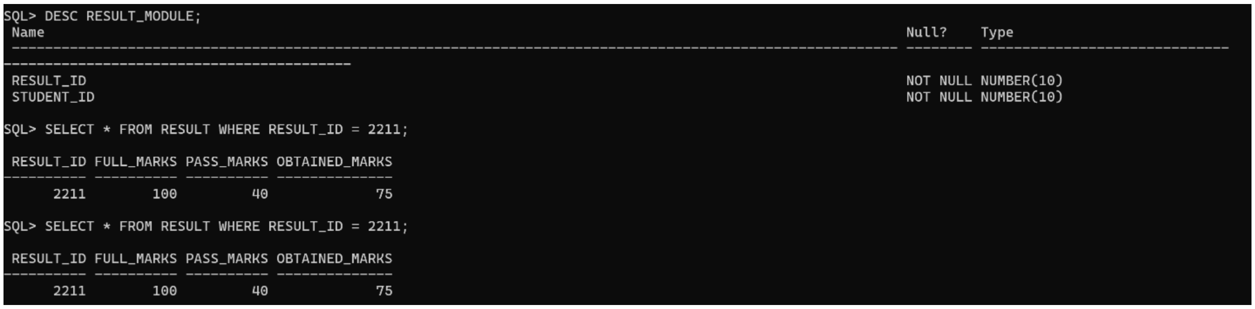
```
SQL> select * from teacher;

TEACHER_ID TEACHER_NA TEACHER_ADDRESS TEACHER_EMAIL
---------- ---------- --------------- ----------------------------------------
      2400 SHRESHA    GAUSHALA        shreshabhandari02@gmail.com
      2401 AVINAV     KRITIPUR        avinavneupane@gmail.com
      2402 SUBHARNA   KOTESHWOR       subharnakarki01@gmail.com
      2403 ABHISHEK   TINKUNE         abhishekbhatt233@gmail.com
      2404 JAGARNATH  SANO THIMI      jagarnathpoudayal23@gmail.com
      2405 SAUGAT     DHOBHIDHARA     saugatmanshakya13@gmail.com
      2406 DIPESHOR   BALUWATAR       dipeshorsilwal03@gmail.com

7 rows selected.

SQL>
```

Figure 16



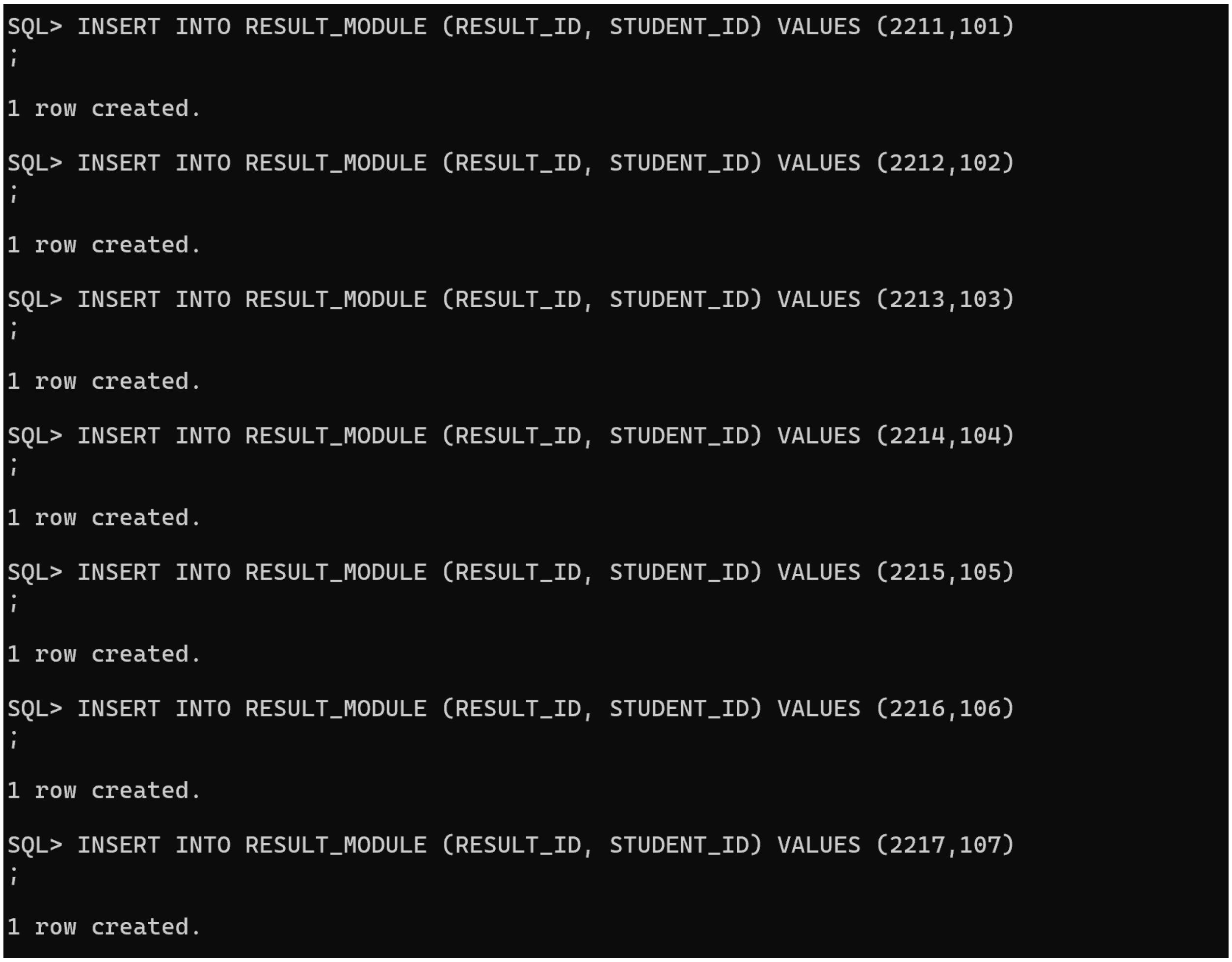
```
SQL> create table result_module ( Result_ID NUMBER(10),Student_ID NUMBER(10),PRIMARY KEY (Result_ID, Student_ID),FOREIGN KEY (Result_ID) REFERENCES
Result(Result_ID),FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID));

Table created.
```

Figure 17



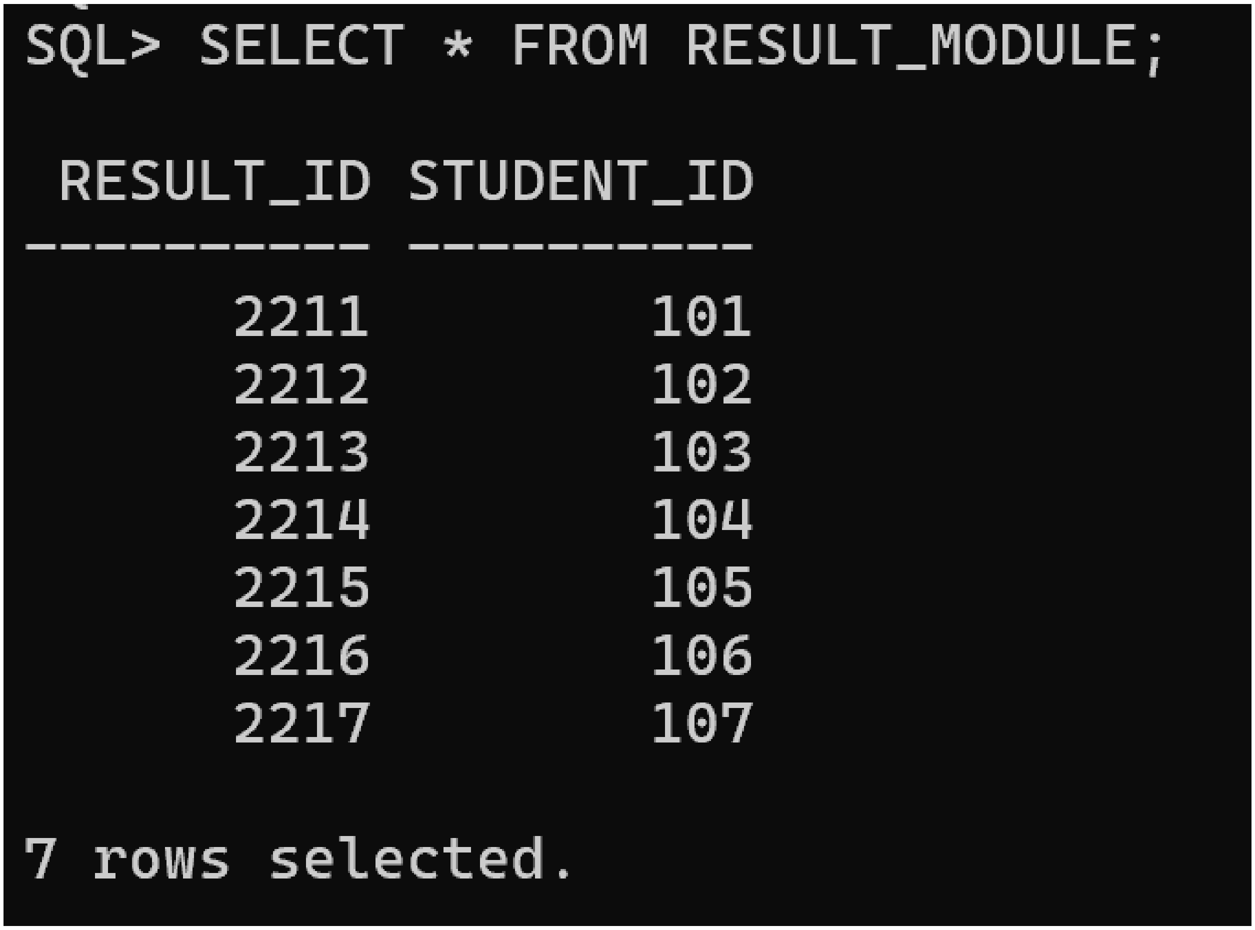
```
SQL> DESC RESULT_MODULE;
 Name                                                                                                              Null?    Type
 ----------------------------------------------------------------------------------------------------------------- -------- ------------------------------
----------------------------------------
 RESULT_ID                                                                                                         NOT NULL NUMBER(10)
 STUDENT_ID                                                                                                        NOT NULL NUMBER(10)

SQL> SELECT * FROM RESULT WHERE RESULT_ID = 2211;

 RESULT_ID FULL_MARKS PASS_MARKS OBTAINED_MARKS
---------- ---------- ---------- --------------
      2211        100         40             75

SQL> SELECT * FROM RESULT WHERE RESULT_ID = 2211;

 RESULT_ID FULL_MARKS PASS_MARKS OBTAINED_MARKS
---------- ---------- ---------- --------------
      2211        100         40             75
```

Figure 18



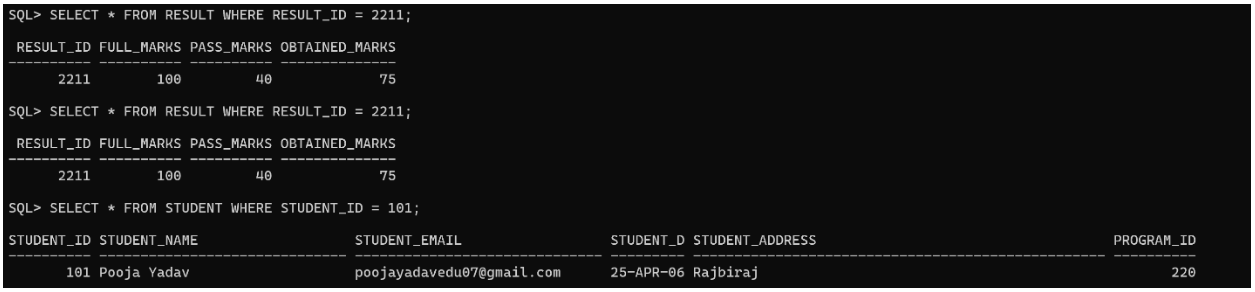
```
SQL> INSERT INTO RESULT_MODULE (RESULT_ID, STUDENT_ID) VALUES (2211,101)
;

1 row created.

SQL> INSERT INTO RESULT_MODULE (RESULT_ID, STUDENT_ID) VALUES (2212,102)
;

1 row created.

SQL> INSERT INTO RESULT_MODULE (RESULT_ID, STUDENT_ID) VALUES (2213,103)
;

1 row created.

SQL> INSERT INTO RESULT_MODULE (RESULT_ID, STUDENT_ID) VALUES (2214,104)
;

1 row created.

SQL> INSERT INTO RESULT_MODULE (RESULT_ID, STUDENT_ID) VALUES (2215,105)
;

1 row created.

SQL> INSERT INTO RESULT_MODULE (RESULT_ID, STUDENT_ID) VALUES (2216,106)
;

1 row created.

SQL> INSERT INTO RESULT_MODULE (RESULT_ID, STUDENT_ID) VALUES (2217,107)
;

1 row created.
```

Figure 19



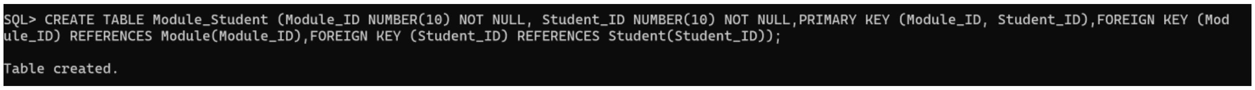
```
SQL> SELECT * FROM RESULT_MODULE;

 RESULT_ID STUDENT_ID
---------- ----------
      2211        101
      2212        102
      2213        103
      2214        104
      2215        105
      2216        106
      2217        107

7 rows selected.
```

Figure 20



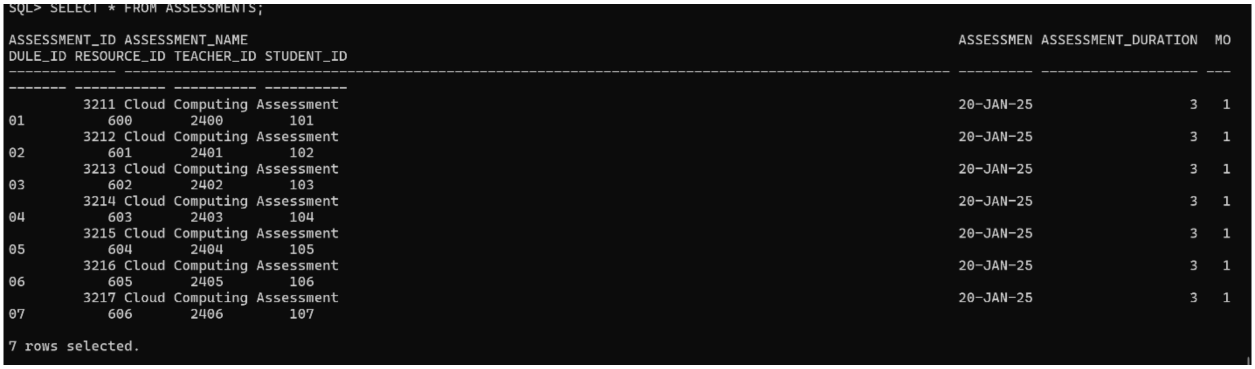
```
SQL> SELECT * FROM RESULT WHERE RESULT_ID = 2211;

 RESULT_ID FULL_MARKS PASS_MARKS OBTAINED_MARKS
---------- ---------- ---------- --------------
      2211        100         40             75

SQL> SELECT * FROM RESULT WHERE RESULT_ID = 2211;

 RESULT_ID FULL_MARKS PASS_MARKS OBTAINED_MARKS
---------- ---------- ---------- --------------
      2211        100         40             75

SQL> SELECT * FROM STUDENT WHERE STUDENT_ID = 101;

STUDENT_ID STUDENT_NAME                   STUDENT_EMAIL                  STUDENT_D STUDENT_ADDRESS                                    PROGRAM_ID
---------- ------------------------------ ------------------------------ --------- -------------------------------------------------- ----------
       101 Pooja Yadav                    poojayadavedu07@gmail.com      25-APR-06 Rajbiraj                                                  220
```

Figure 21



```
SQL> CREATE TABLE Module_Student (Module_ID NUMBER(10) NOT NULL, Student_ID NUMBER(10) NOT NULL,PRIMARY KEY (Module_ID, Student_ID),FOREIGN KEY (Mod
ule_ID) REFERENCES Module(Module_ID),FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID));

Table created.
```

Figure 22

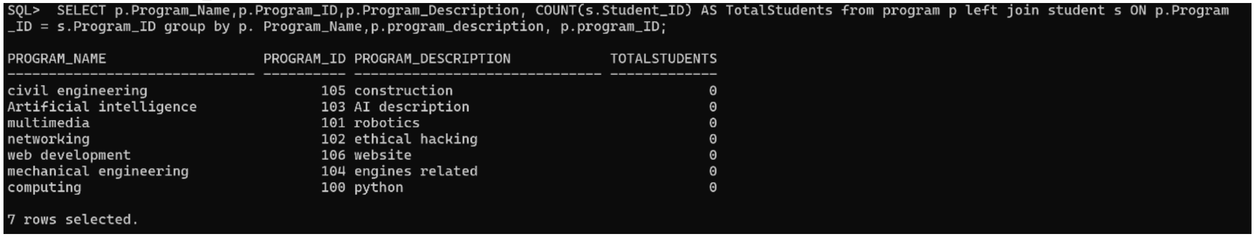


```
SQL> SELECT * FROM ASSESSMENTS;

ASSESSMENT_ID ASSESSMENT_NAME                                                                                    ASSESSMEN ASSESSMENT_DURATION  MO
DULE_ID RESOURCE_ID TEACHER_ID STUDENT_ID
------------- ---------------------------------------------------------------------------------------------------- --------- ------------------- ---
------- ----------- ---------- ----------
         3211 Cloud Computing Assessment                                                                           20-JAN-25                   3   1
01          600       2400        101
         3212 Cloud Computing Assessment                                                                           20-JAN-25                   3   1
02          601       2401        102
         3213 Cloud Computing Assessment                                                                           20-JAN-25                   3   1
03          602       2402        103
         3214 Cloud Computing Assessment                                                                           20-JAN-25                   3   1
04          603       2403        104
         3215 Cloud Computing Assessment                                                                           20-JAN-25                   3   1
05          604       2404        105
         3216 Cloud Computing Assessment                                                                           20-JAN-25                   3   1
06          605       2405        106
         3217 Cloud Computing Assessment                                                                           20-JAN-25                   3   1
07          606       2406        107

7 rows selected.
```

Figure 23

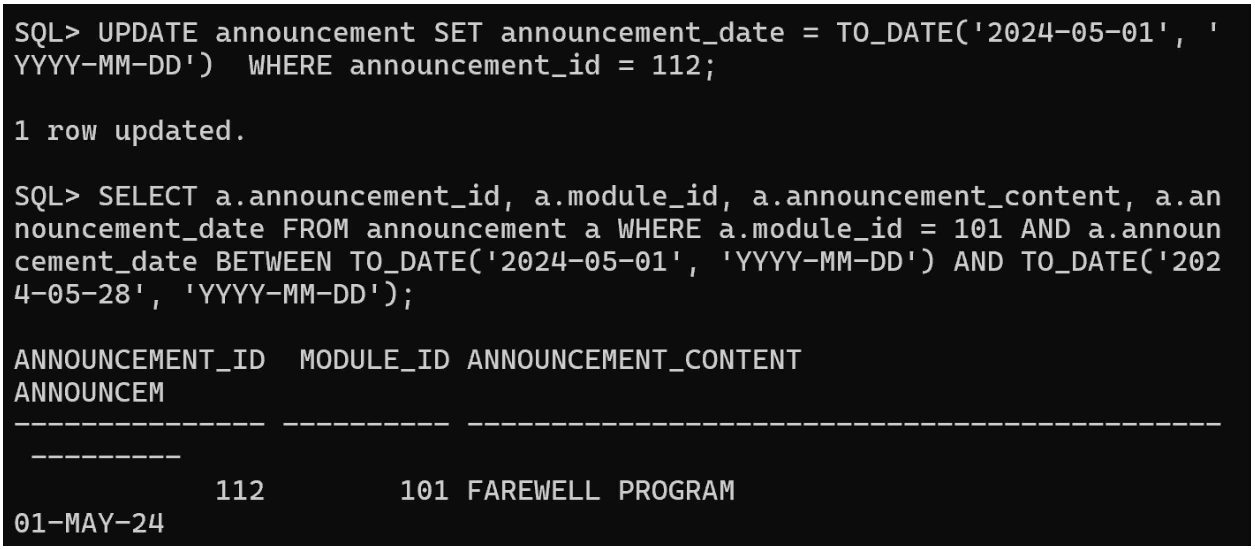
Queries

Information queries



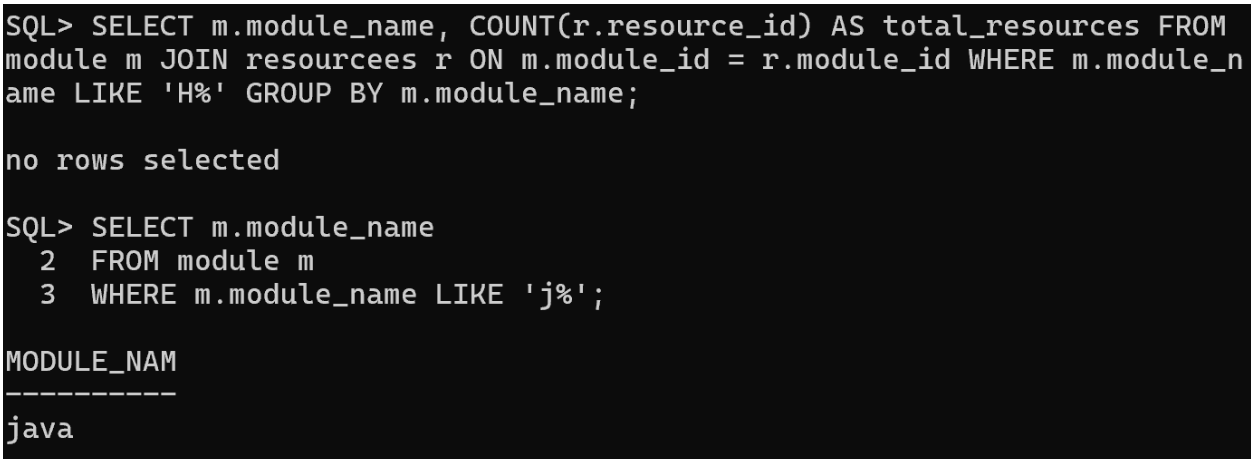
```
SQL>  SELECT p.Program_Name,p.Program_ID,p.Program_Description, COUNT(s.Student_ID) AS TotalStudents from program p left join student s ON p.Program
_ID = s.Program_ID group by p. Program_Name,p.program_description, p.program_ID;

PROGRAM_NAME                   PROGRAM_ID PROGRAM_DESCRIPTION            TOTALSTUDENTS
------------------------------ ---------- ------------------------------ -------------
civil engineering                     105 construction                               0
Artificial intelligence               103 AI description                             0
multimedia                            101 robotics                                   0
networking                            102 ethical hacking                            0
web development                       106 website                                    0
mechanical engineering                104 engines related                            0
computing                             100 python                                     0

7 rows selected.
```

Figure 24



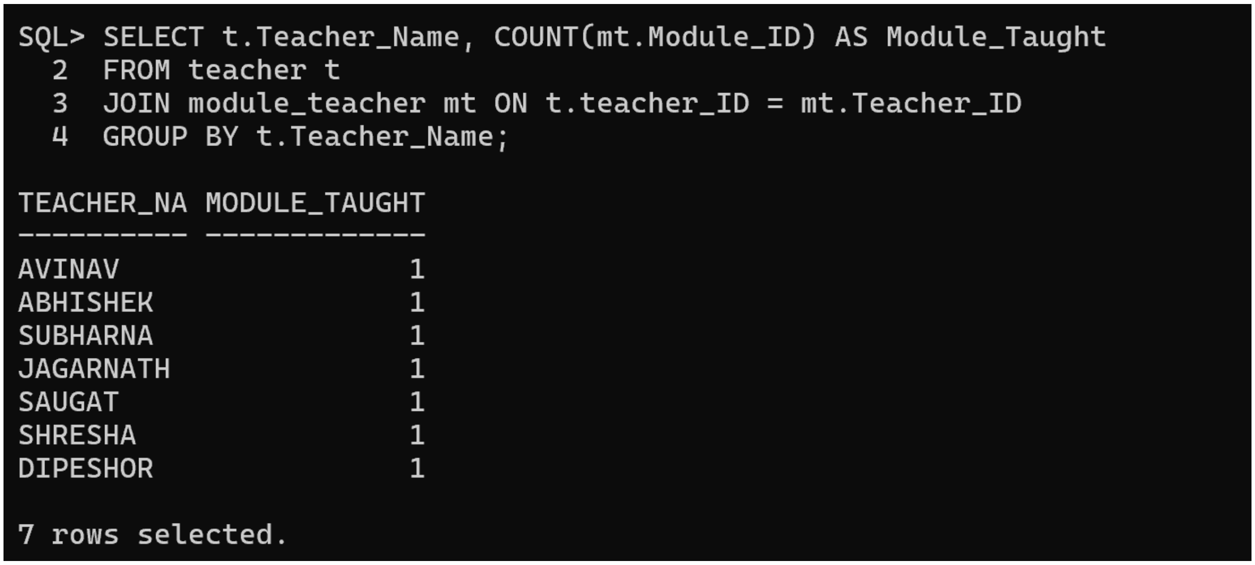
```
SQL> UPDATE announcement SET announcement_date = TO_DATE('2024-05-01', '
YYYY-MM-DD')  WHERE announcement_id = 112;

1 row updated.

SQL> SELECT a.announcement_id, a.module_id, a.announcement_content, a.an
nouncement_date FROM announcement a WHERE a.module_id = 101 AND a.announ
cement_date BETWEEN TO_DATE('2024-05-01', 'YYYY-MM-DD') AND TO_DATE('202
4-05-28', 'YYYY-MM-DD');

ANNOUNCEMENT_ID  MODULE_ID ANNOUNCEMENT_CONTENT
ANNOUNCEM
--------------- ---------- ---------------------------------------------
 ---------
            112        101 FAREWELL PROGRAM
01-MAY-24
```

Figure 25



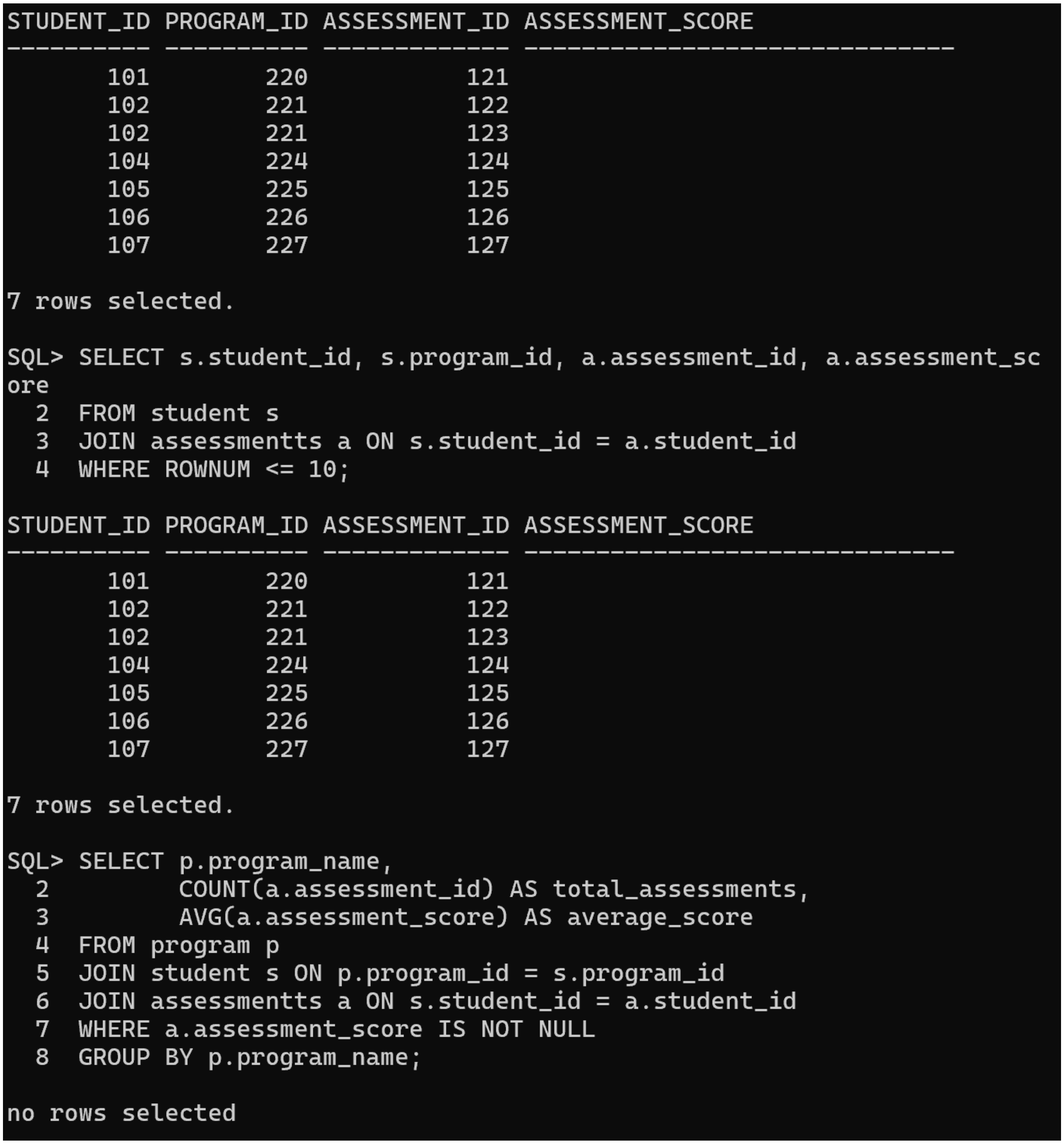
```
SQL> SELECT m.module_name, COUNT(r.resource_id) AS total_resources FROM
module m JOIN resourcees r ON m.module_id = r.module_id WHERE m.module_n
ame LIKE 'H%' GROUP BY m.module_name;

no rows selected

SQL> SELECT m.module_name
  2  FROM module m
  3  WHERE m.module_name LIKE 'j%';

MODULE_NAM
----------
java
```

Figure 26



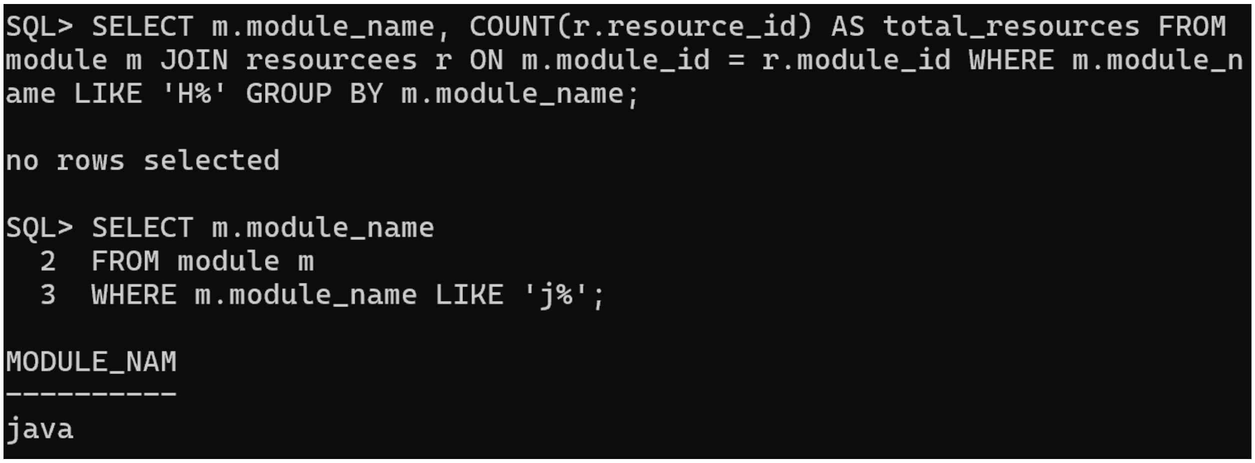
```
SQL> SELECT t.Teacher_Name, COUNT(mt.Module_ID) AS Module_Taught
     FROM teacher t
     JOIN module_teacher mt ON t.teacher_ID = mt.Teacher_ID
     GROUP BY t.Teacher_Name;

TEACHER_NA MODULE_TAUGHT
---------- -------------
AVINAV                 1
ABHISHEK               1
SUBHARNA               1
JAGARNATH              1
SAUGAT                 1
SHRESHA                1
DIPESHOR               1

7 rows selected.
```

Figure 27



```
STUDENT_ID PROGRAM_ID ASSESSMENT_ID ASSESSMENT_SCORE
---------- ---------- ------------- ------------------------------
       101        220           121
       102        221           122
       102        221           123
       104        224           124
       105        225           125
       106        226           126
       107        227           127

7 rows selected.

SQL> SELECT s.student_id, s.program_id, a.assessment_id, a.assessment_sc
ore
  2  FROM student s
  3  JOIN assessmentts a ON s.student_id = a.student_id
  4  WHERE ROWNUM <= 10;

STUDENT_ID PROGRAM_ID ASSESSMENT_ID ASSESSMENT_SCORE
---------- ---------- ------------- ------------------------------
       101        220           121
       102        221           122
       102        221           123
       104        224           124
       105        225           125
       106        226           126
       107        227           127

7 rows selected.

SQL> SELECT p.program_name,
  2         COUNT(a.assessment_id) AS total_assessments,
  3         AVG(a.assessment_score) AS average_score
  4  FROM program p
  5  JOIN student s ON p.program_id = s.program_id
  6  JOIN assessmentts a ON s.student_id = a.student_id
  7  WHERE a.assessment_score IS NOT NULL
  8  GROUP BY p.program_name;

no rows selected
```

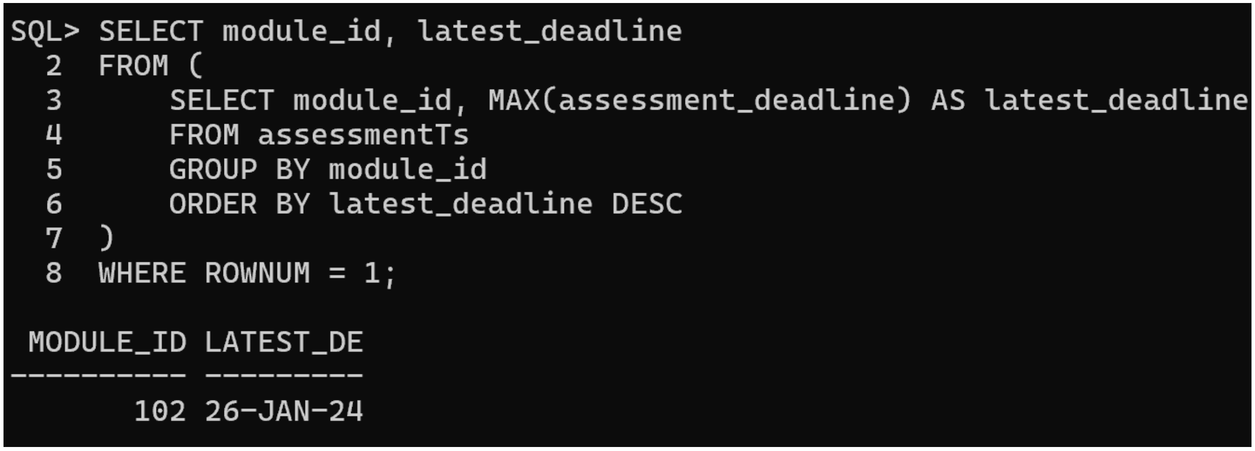
Figure 28



```
SQL> SELECT m.module_name, COUNT(r.resource_id) AS total_resources FROM
module m JOIN resourcees r ON m.module_id = r.module_id WHERE m.module_n
ame LIKE 'H%' GROUP BY m.module_name;

no rows selected

SQL> SELECT m.module_name
  2  FROM module m
  3  WHERE m.module_name LIKE 'j%';

MODULE_NAM
----------
java
```

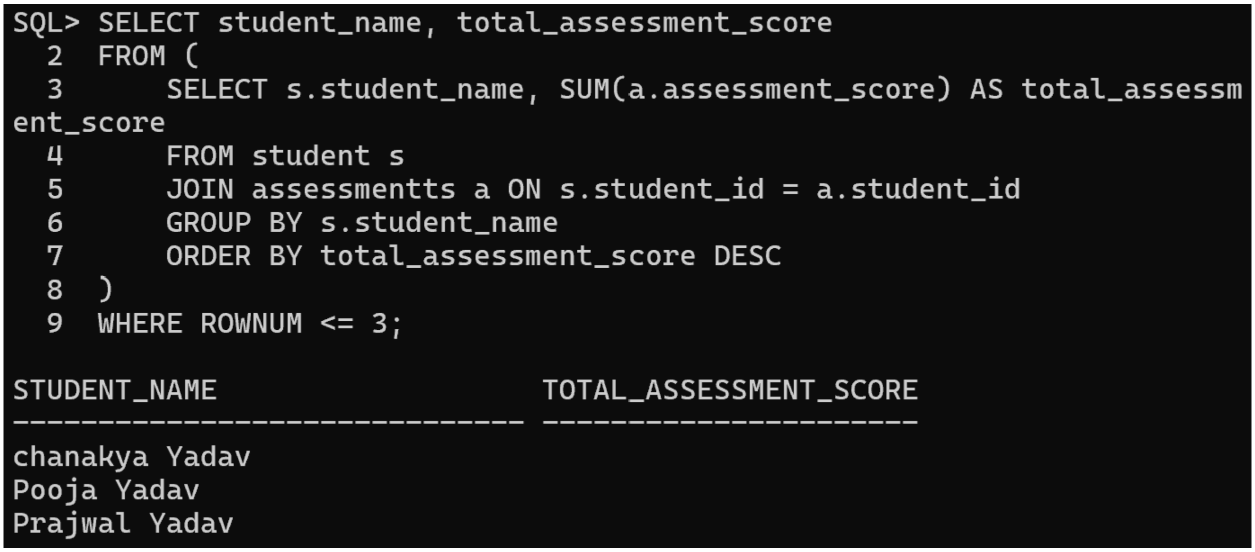
Figure 29

Transaction query



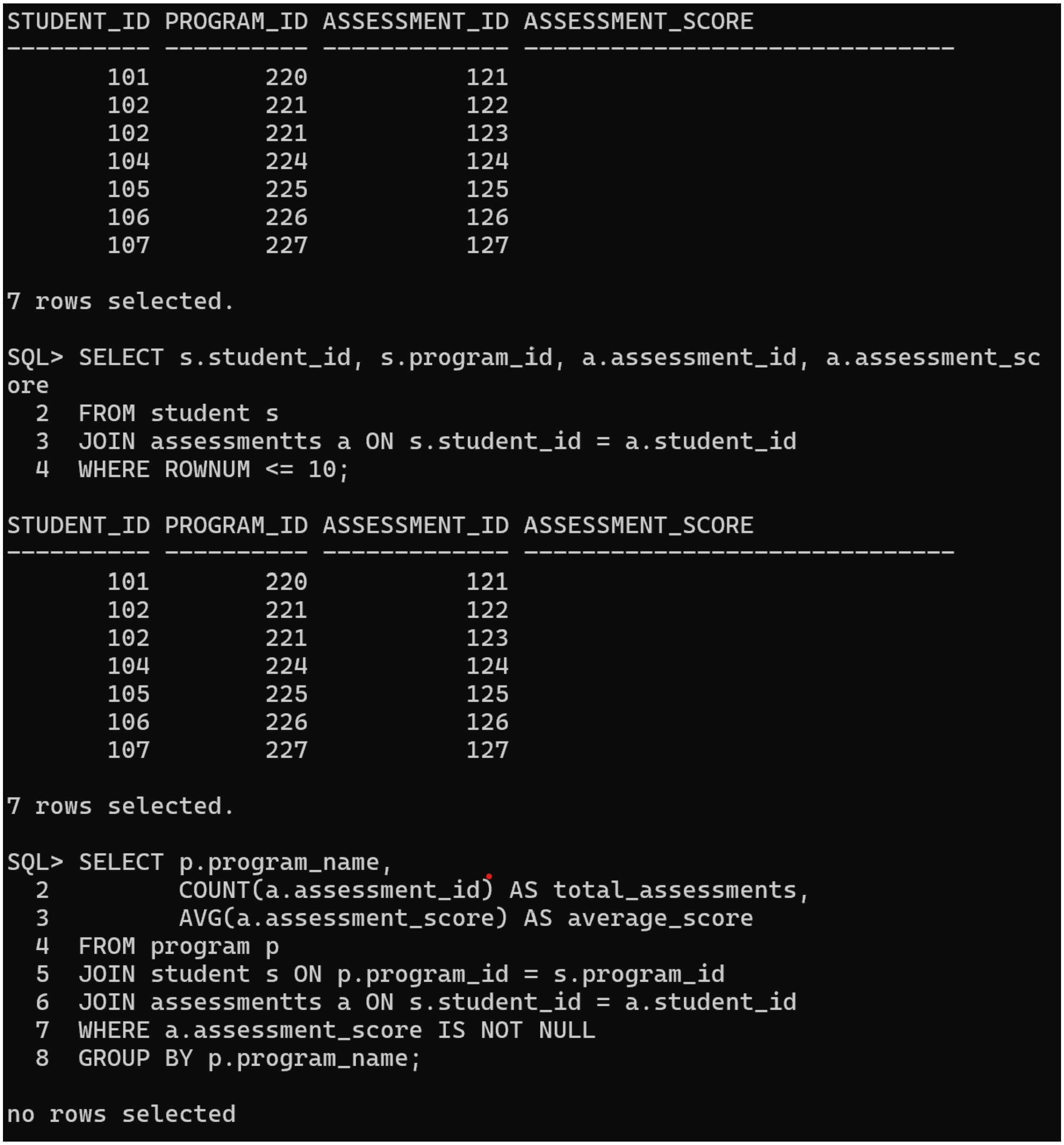
```
SQL> SELECT module_id, latest_deadline
     FROM (
         SELECT module_id, MAX(assessment_deadline) AS latest_deadline
         FROM assessmentTs
         GROUP BY module_id
         ORDER BY latest_deadline DESC
     )
     WHERE ROWNUM = 1;

 MODULE_ID LATEST_DE
---------- ---------
       102 26-JAN-24
```

Figure 30



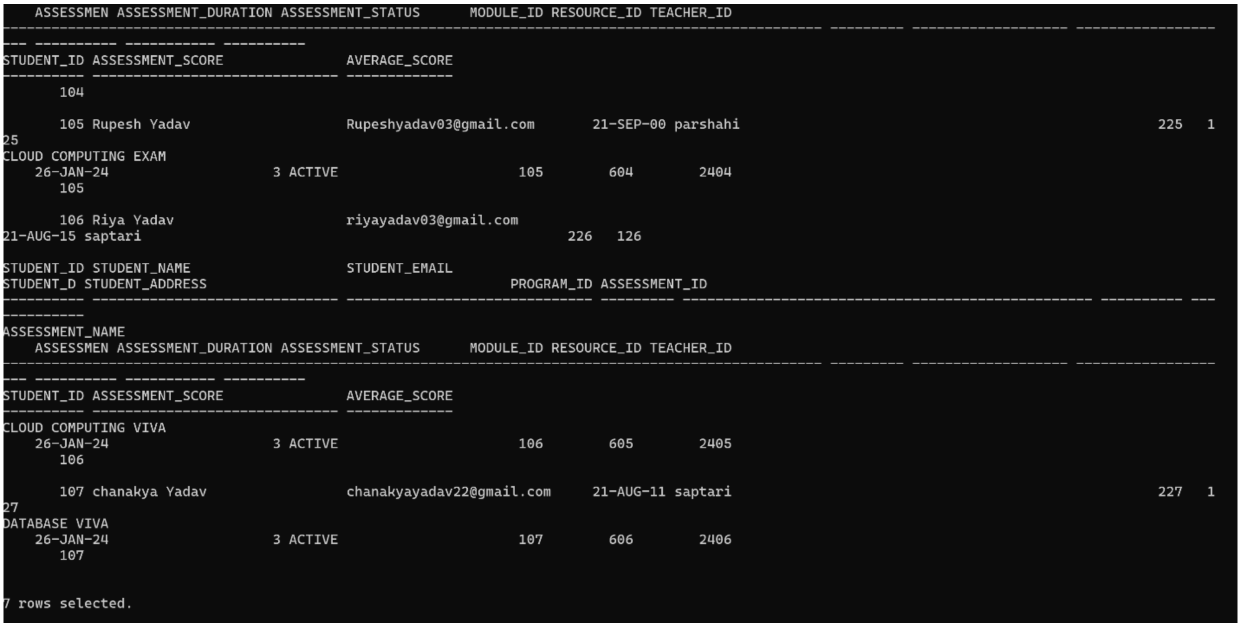
```
SQL> SELECT student_name, total_assessment_score
     FROM (
         SELECT s.student_name, SUM(a.assessment_score) AS total_assessm
ent_score
         FROM student s
         JOIN assessmentts a ON s.student_id = a.student_id
         GROUP BY s.student_name
         ORDER BY total_assessment_score DESC
     )
     WHERE ROWNUM <= 3;

STUDENT_NAME                   TOTAL_ASSESSMENT_SCORE
------------------------------ ----------------------
chanakya Yadav
Pooja Yadav
Prajwal Yadav
```

Figure 31



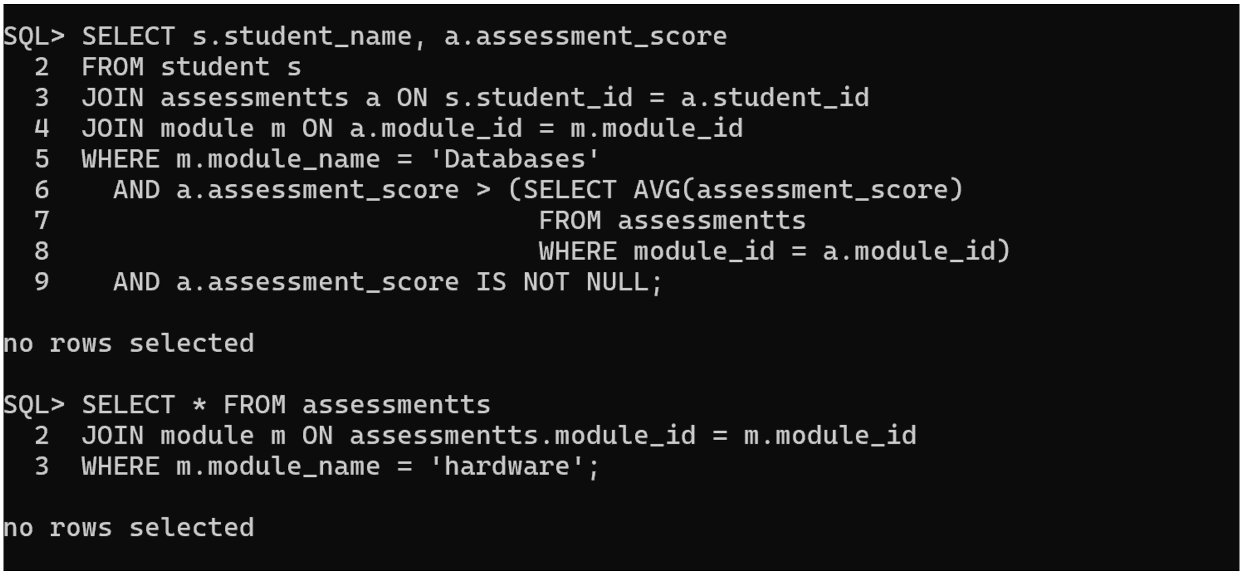
```
STUDENT_ID PROGRAM_ID ASSESSMENT_ID ASSESSMENT_SCORE
---------- ---------- ------------- ------------------------------
       101        220           121
       102        221           122
       102        221           123
       104        224           124
       105        225           125
       106        226           126
       107        227           127

7 rows selected.

SQL> SELECT s.student_id, s.program_id, a.assessment_id, a.assessment_sc
ore
  2  FROM student s
  3  JOIN assessmentts a ON s.student_id = a.student_id
  4  WHERE ROWNUM <= 10;

STUDENT_ID PROGRAM_ID ASSESSMENT_ID ASSESSMENT_SCORE
---------- ---------- ------------- ------------------------------
       101        220           121
       102        221           122
       102        221           123
       104        224           124
       105        225           125
       106        226           126
       107        227           127

7 rows selected.

SQL> SELECT p.program_name,
  2         COUNT(a.assessment_id) AS total_assessments,
  3         AVG(a.assessment_score) AS average_score
  4  FROM program p
  5  JOIN student s ON p.program_id = s.program_id
  6  JOIN assessmentts a ON s.student_id = a.student_id
  7  WHERE a.assessment_score IS NOT NULL
  8  GROUP BY p.program_name;

no rows selected
```

Figure 32



```
    ASSESSMEN ASSESSMENT_DURATION ASSESSMENT_STATUS      MODULE_ID RESOURCE_ID TEACHER_ID
---------------------------------------------------------------------------------------------------- --------- ------------------- ------------------
--- ---------- ----------- ----------
STUDENT_ID ASSESSMENT_SCORE               AVERAGE_SCORE
---------- ------------------------------ -------------
       104

       105 Rupesh Yadav                   Rupeshyadav03@gmail.com       21-SEP-00 parshahi                                                   225   1
25
CLOUD COMPUTING EXAM
    26-JAN-24                    3 ACTIVE                      105        604        2404
       105

       106 Riya Yadav                     riyayadav03@gmail.com
21-AUG-15 saptari                                                   226   126

STUDENT_ID STUDENT_NAME                   STUDENT_EMAIL
STUDENT_D STUDENT_ADDRESS                                   PROGRAM_ID ASSESSMENT_ID
---------- ------------------------------ ------------------------------ --------- -------------------------------------------------- ---------- ---
----------
ASSESSMENT_NAME
    ASSESSMEN ASSESSMENT_DURATION ASSESSMENT_STATUS      MODULE_ID RESOURCE_ID TEACHER_ID
---------------------------------------------------------------------------------------------------- --------- ------------------- ------------------
--- ---------- ----------- ----------
STUDENT_ID ASSESSMENT_SCORE               AVERAGE_SCORE
---------- ------------------------------ -------------
CLOUD COMPUTING VIVA
    26-JAN-24                    3 ACTIVE                      106        605        2405
       106

       107 chanakya Yadav                 chanakyayadav22@gmail.com     21-AUG-11 saptari                                                    227   1
27
DATABASE VIVA
    26-JAN-24                    3 ACTIVE                      107        606        2406
       107


7 rows selected.
```

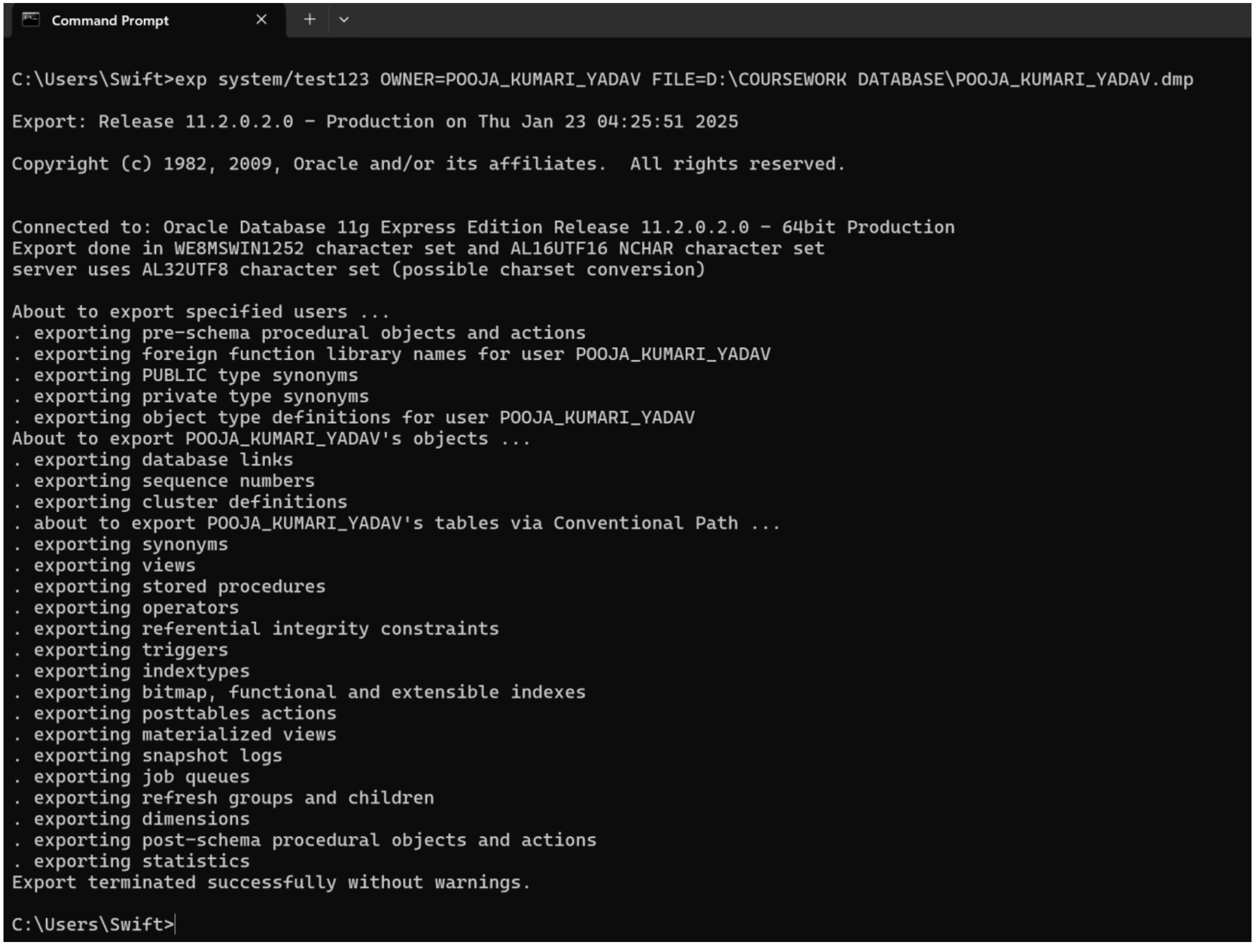
Figure 33



```
SQL> SELECT s.student_name, a.assessment_score
     FROM student s
     JOIN assessmentts a ON s.student_id = a.student_id
     JOIN module m ON a.module_id = m.module_id
     WHERE m.module_name = 'Databases'
       AND a.assessment_score > (SELECT AVG(assessment_score)
                                  FROM assessmentts
                                  WHERE module_id = a.module_id)
       AND a.assessment_score IS NOT NULL;

no rows selected

SQL> SELECT * FROM assessmentts
     JOIN module m ON assessmentts.module_id = m.module_id
     WHERE m.module_name = 'hardware';

no rows selected
```

Figure 34

Dump file



```
C:\Users\Swift>exp system/test123 OWNER=POOJA_KUMARI_YADAV FILE=D:\COURSEWORK DATABASE\POOJA_KUMARI_YADAV.dmp

Export: Release 11.2.0.2.0 - Production on Thu Jan 23 04:25:51 2025

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All rights reserved.


Connected to: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
server uses AL32UTF8 character set (possible charset conversion)

About to export specified users ...
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user POOJA_KUMARI_YADAV
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user POOJA_KUMARI_YADAV
About to export POOJA_KUMARI_YADAV's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export POOJA_KUMARI_YADAV's tables via Conventional Path ...
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting statistics
Export terminated successfully without warnings.

C:\Users\Swift>
```

Figure 35

Conclusion

The module has been focuses on concepts like database management, programming and system which are foundational in fields like software development, business analytics and backend engineering. In real world, skills are directly applicable in industries such as development. This module links to subjects like computer science, mathematics, AI, and business studies. The coursework depends upon how much they have experience and knowledge about it. The challenges which has faces many debugging issues like name error, parent not found error and managing multiple components eg. GUI, database etc can be challenging when learning how they interact during queries. The coursework effectively ties together therotical knowledge with practical application . we can improve debugging by enhancing error handling and validation. And for modularization improvement we should break down code into smaller functions or classes for debugging. It include detailed comments and more unit tests to improve maintainability and reliability. Throughout the coursework , it was so knowledgeable, it makes our skills much better to enhance our learning thoughout the semester.

References

(normal forms inDBMS, 2025) (Simplilearn, 2024)

(geeksforgeeks, 2022)

(yasar, 2022)