**Faculty of Science and Engineering**

**University of Wolverhampton**

**FRONT COVER**

**­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**STUDENT ID: 1928477**

**STUDENT NAME: Amartey Pius**

**TUTOR NAME: Sherin Nassa**

**MODULE TITLE: Advanced Databases**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ASSIGNMENT TITLE: Database Design**

**SUBMISSION DATE: 01/04/2022**

**Time: 14:00**

**Scenario: Wolf University**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DATABASE DESIGN**

*submitted*

*by*

**2120687, 1928477**

*for*

**A DATABASE FOR WOLF UNIVERSITY**

**01/04/2022**

**Faculty of Science and Engineering**

**University of Wolverhampton**

Icon

Description automatically generated with medium confidence

**TABLE OF CONTENT**

**Business Rules………………………………………………………………………………….6**

**Entity Relation Diagram………………………………………………………………………7**

**Relational headings…………………………………………………………………………….8**

**SQL Scripts………………………………………………………………………………….8-11**

**Database Tables and Data…………………………………………………………………12-17**

**Queries……………………………………………………………………………................17-25**

**Testing Views……………………………………………………………………………….25-30**

**Audit Tables and Triggers…………………………………………………………………30-44**

**Data Warehouse…………………………………………………………………………….45-47**

**Wolf University Database**

**BUSINESS RULES**

* Student must register
* Student must hand in personal details which entails Full names, email, address and phone number
* Student must choose a Course
* Each course student must choose to study full time or part time
* When student is accepted student must be given a unique student number
* Each Student must have a unique course ID attached to their student number to identify what course they are studying
* Modules are either mandatory or optional depending on their course
* Students must have timetables relating to each module
* Students enrolling must be above 18
* Students must notify the university about any learning disabilities and if they would need any addition help or support during their studies.
* Students must take a test or do a coursework for each of their modules
* Overalmark should not be greater than 100
* Students can choose to defer their course
* Student initials must be in capital letters

**ER DIGRAM**

Diagram

Description automatically generated

**Relational Headings**

Student (studentno, studentname, studentinitials, gender,

age, email)

Module (moduleno, moduletitle, noofcredits, type)

Course (coursecode, coursetitle)

Registration (modes, status, startdate, courseresult,

*studentno, course\_coursecode*)

Result (*module\_moduleno, student\_studentno, academicyear,*

*Semester*, courseworkresults, courseworkweighting, examresults, examweighting, overalmark, status, comments)

1. **SQL SCRIPTS**

**Module Table**

CREATE TABLE module (

moduleno VARCHAR2(7) NOT NULL

CONSTRAINT PK\_module primary key,

moduletitle VARCHAR2(20) NOT NULL,

noofcredits NUMBER(5) NOT NULL,

type VARCHAR2(10) NOT NULL);

**Results Table**

CREATE TABLE results (

CREATE TABLE results (

academicyear VARCHAR2(7) NOT NULL,

semester VARCHAR2(5) NOT NULL,

CourseWorkresults NUMBER(5, 2) NOT NULL,

CourseWorkweighting NUMBER(5, 2) NOT NULL,

Examresults NUMBER(5, 2) NULL,

Examweighting NUMBER(5, 2) NULL,

overalmark NUMBER(5 ,2) NOT NULL CHECK (overalmark <= 100),

status VARCHAR2(5) NOT NULL,

comments VARCHAR2(16) NULL,

module\_moduleno VARCHAR2(7) NOT NULL

CONSTRAINT FK\_moduleno REFERENCES module(moduleno),

student\_studentno NUMBER(7) NOT NULL

CONSTRAINT FK\_studentno REFERENCES student(studentno),

CONSTRAINT PK\_results primary key (module\_moduleno, student\_studentno, academicYear, semester));

**Registration Table**

CREATE TABLE registration (

modes VARCHAR2(2) NOT NULL,

status VARCHAR2(2) NOT NULL,

startdate DATE NOT NULL,

courseresult VARCHAR2(5) NULL,

studentno NUMBER(7) NOT NULL

CONSTRAINT FK\_student\_studentno REFERENCES student(studentno),

course\_coursecode NUMBER(5) NOT NULL

CONSTRAINT FK\_coursecode REFERENCES course(coursecode));

**Student Table**

CREATE TABLE student (

studentno NUMBER(7) NOT NULL

CONSTRAINT PK\_student primary key,

studentname VARCHAR2(20) NOT NULL,

studentinitials VARCHAR2(2) NOT NULL

CONSTRAINT UC\_studentinitials CHECK (studentinitials=UPPER(studentinitials)),

gender VARCHAR2(2) NOT NULL,

age NUMBER(2) NOT NULL CHECK (age > 18),

email VARCHAR2(20) NOT NULL);

**Course Table**

CREATE TABLE course (

coursecode NUMBER(5) NOT NULL

CONSTRAINT PK\_course primary key,

coursetitle VARCHAR2(40) NOT NULL);

1. **DATABASE TABLES AND DATA**

**SELECT \* FROM STUDENT**;

STUDENTNO STUDENTNAME ST GE AGE EMAIL

---------- -------------------- -- -- ---------- --------------------

1254657 Chris Wood CW M 21 chriswood@gmail.com

4765392 Anthony Marshal AM M 20 anthonym@gmail.com

6743823 Angela Korman AK F 23 angelakor@gmail.com

7007541 Elanga Junior EJ M 19 elangaj@gmail.com

2307653 Ashley Smith AS F 21 ashleys@gmail.com

9691208 Shiembo Mambo SM F 19 shiembom@gmail.com

1888754 Louisa Shina LS F 25 louisas@gmail.com

4489026 Paul Pogba PP M 30 paulp@gmail.com

7649062 Maison Mount MM M 34 maisonm@gmail.com

7563789 Vanessa Reece VR F 25 vanessar@gmail.com

8734102 Thomas Patel TP M 29 thomasp@gmail.com

3509742 Joel Matip JM M 22 joelm@gmail.com

9835411 Mira Kay MK F 24 mirak@gmail.com

3121622 Louis Jay LJ M 20 louisj@gmail.com

7289332 Sarah May SM F 24 sarahm@gmail.com

2295789 Karen Bibi KB F 20 Karenbm@gmail.com

8884521 Bruce Breezy BB M 21 bruceb@gmail.com

**select \* from results;**

ACADEMI SEMES COURSEWORKRESULTS COURSEWORKWEIGHTING EXAMRESULTS EXAMWEIGHTING OVERALMARK STATU COMMENTS MODULE\_ STUDENT\_STUDENTNO

------- ----- ----------------- ------------------- ----------- ------------- ---------- ----- ---------------- ------- -----------------

2021/22 SEM1 87 50 75 50 81 Pass Excellent 5CA3333 1254657

2021/22 SEM1 95 50 60 50 77.5 Pass Excellent 5CA3333 7007541

2021/22 SEM1 95 50 78 50 86.5 Pass Excellent 5CA3333 1888754

2021/22 SEM1 56 50 80 50 68 Pass Good 6DE7683 4765392

2021/22 SEM1 73 50 55 50 64 Pass Good 6DE7683 2307653

2021/22 SEM1 54 50 63 50 58.5 Pass Good 6DE7683 4489026

2021/22 SEM2 12 50 60 50 36 Defer Resit Coursework 2SE4444 9691208

2021/22 SEM2 43 50 34 50 38.5 Defer Resit Exam 2SE4444 6743823

2021/22 SEM2 78 50 88 50 83 Pass Excellent 2SE4444 7563789

2021/22 SEM2 63 100 0 0 63 Pass Good 7TY7362 8734102

2021/22 SEM2 13 100 0 0 13 Defer Resit coursework 7TY7362 7649062

2021/22 SEM2 43 100 0 0 43 Pass Fair 7TY7362 3509742

2021/22 SEM2 89 100 0 0 89 Pass Excellent 8BH2911 9835411

2021/22 SEM2 76 100 0 0 76 Pass Excellent 8BH2911 3121622

2021/22 SEM2 91.5 100 0 0 91.5 Pass Ecellent 8BH2911 7289332

2021/22 SEM2 78 100 0 0 78 Pass Ecellent 9JK6272 2295789

2021/22 SEM2 51 100 0 0 51 Pass Good 1BN8354 8884521

**SELECT \* FROM COURSE**

COURSECODE COURSETITLE

---------- ----------------------------------------

95056 BSc Computer Science

76098 BSc Cyber Security

45632 BSc Software Engineering

64526 BSc Electrical Engineering

90047 BSc Mechanical Engineering

37774 Nursing

11112 Sciences

**SELECT \* FROM MODULE;**

MODULEN MODULETITLE NOOFCREDITS TYPE

------- -------------------- ----------- ----------

5CA3333 Advanced Databases 100 Mandatory

6DE7683 Web Development 100 Mandatory

2SE4444 Computer Networking 100 Mandatory

7TY7362 Robotics 100 Mandatory

8BH2911 Analysis of circuits 100 Mandatory

9JK6272 Future Nurse 100 Optional

1BN8354 Physics 100 Optional

**SELECT \* FROM REGISTRATION.**

MO ST STARTDATE COURS STUDENTNO COURSE\_COURSECODE

-- -- --------- ----- ---------- -----------------

FT H 29-SEP-21 1 1254657 95056

PT H 29-SEP-21 1 7007541 95056

PT H 29-SEP-21 1 1888754 95056

FT O 15-SEP-21 2.1 4765392 45632

FT H 15-OCT-21 2.1 2307653 45632

FT H 15-OCT-21 2.2 4489026 45632

FT O 15-JAN-22 Pass 9691208 76098

PT O 15-JAN-22 Pass 6743823 76098

PT O 15-JAN-22 1 7563789 76098

FT H 26-JAN-22 2.2 8734102 90047

FT H 26-JAN-22 Pass 7649062 90047

FT O 26-JAN-22 3 3509742 90047

PT O 15-FEB-22 1 9835411 64526

PT H 15-FEB-22 1 3121622 64526

FT H 15-FEB-22 1 7289332 64526

FT H 20-FEB-22 1 2295789 37774

FT O 25-FEB-22 3 8884521 11112

3**) QUERIES**

a) SELECT s.studentName ,s.gender ,s.email

from Student s

where s.studentno in ( SELECT r.student\_studentno

from Results r

where r.overalMark = ( SELECT max(overalmark)

from Results ));

STUDENTNAME GE EMAIL

-------------------- -- --------------------

Sarah May F [sarahm@gmail.com](mailto:sarahm@gmail.com)

b) SELECT COUNT(student.studentno)

FROM student

INNER JOIN results ON student.studentno = results.student\_studentno

GROUP BY results.academicyear;

COUNT(STUDENT.STUDENTNO)

------------------------

17

c)

SELECT course.coursecode, course.coursetitle, student.studentNo, student.studentinitials , registration.status,

CASE registration.courseresult

WHEN '1'

THEN 'First'

WHEN '2.1'

THEN 'Upper Second'

WHEN '3'

THEN 'Third'

ELSE registration.courseresult

END courseResult

FROM student

JOIN registration

ON student.studentNo = registration.studentNo

JOIN Course

ON course.coursecode = registration.course\_coursecode;

COURSECODE COURSETITLE STUDENTNO ST ST COURSERESULT

---------- ---------------------------------------- ---------- -- -- ------------

95056 BSc Computer Science 1254657 CW H First

95056 BSc Computer Science 7007541 EJ H First

95056 BSc Computer Science 1888754 LS H First

45632 BSc Software Engineering 4765392 AM O Upper Second

45632 BSc Software Engineering 2307653 AS H Upper Second

45632 BSc Software Engineering 4489026 PP H 2.2

76098 BSc Cyber Security 9691208 SM O Pass

76098 BSc Cyber Security 6743823 AK O Pass

76098 BSc Cyber Security 7563789 VR O First

90047 BSc Mechanical Engineering 8734102 TP H 2.2

90047 BSc Mechanical Engineering 7649062 MM H Pass

90047 BSc Mechanical Engineering 3509742 JM O Third

64526 BSc Electrical Engineering 9835411 MK O First

64526 BSc Electrical Engineering 3121622 LJ H First

64526 BSc Electrical Engineering 7289332 SM H First

37774 Nursing 2295789 KB H First

11112 Sciences 8884521 BB O Third

d)SELECT \* FROM course

WHERE coursecode

IN (SELECT course\_coursecode

FROM registration

GROUP BY(course\_coursecode)

HAVING COUNT(studentno) < 2);

COURSECODE COURSETITLE

---------- ----------------------------------------

11112 Sciences

37774 Nursing

e)SELECT studentname

FROM student

WHERE studentno

IN (SELECT student\_studentno

FROM results

WHERE overalmark > 60 AND semester = 'SEM1'

)

AND gender = 'F';

STUDENTNAME

--------------------

Ashley Smith

Louisa Shina

**FOUR ADDITIONAL QUERIES**

F) This query finds students with the overall mark between 40 and 60. It produces list of student name, student number, academic year, semester, and the module name

SELECT s.studentno, s.studentname, r.academicyear, r.semester, m.moduletitle

FROM student s

INNER JOIN results r ON s.studentno = r.student\_studentno

INNER JOIN module m ON r.module\_moduleno = m.moduleno

WHERE overalmark BETWEEN 40 AND 60

ORDER BY s.studentno ASC;

STUDENTNO STUDENTNAME ACADEMI SEMES MODULETITLE

---------- -------------------- ------- ----- --------------------

3509742 Joel Matip 2021/22 SEM2 Robotics

4489026 Paul Pogba 2021/22 SEM1 Web Development

8884521 Bruce Breezy 2021/22 SEM2 Physics

G) This query shows studentname, studentno, academic year, semester and moduletitle who have character letters in their names with ‘m’ and ‘s’ letters ,

select studentname

from student

where studentname LIKE '%m%' OR studentname LIKE '%s%';

STUDENTNAME

--------------------

Chris Wood

Anthony Marshal

Angela Korman

Ashley Smith

Shiembo Mambo

Louisa Shina

Maison Mount

Vanessa Reece

Thomas Patel

Louis Jay

H) This query divides students into groups by course, then counts the number of students in each group and calculates each group’s average mark.

select course.coursetitle ,

count(registration.studentno) As Number\_of\_Students,

avg(results.overalmark) As Average\_overalMark

from registration

join course

on course.coursecode=registration.course\_coursecode

join results

on results.student\_studentno=registration.studentno

group by course.coursetitle;

COURSETITLE NUMBER\_OF\_STUDENTS AVERAGE\_OVERALMARK

---------------------------------------- ------------------ ------------------

BSc Mechanical Engineering 3 39.6666667

Nursing 1 78

Sciences 1 51

BSc Software Engineering 3 63.5

BSc Computer Science 3 81.6666667

BSc Cyber Security 3 52.5

BSc Electrical Engineering 3 85.5

1. This query shows the students who have an average total mark below 80 in their coursework results

select student.\*

from results

join student

on results.student\_studentno =student.studentno

where results.CourseWorkresults in (select avg(CourseWorkresults) AS AverageCWR from results group by student\_studentno having avg(CourseWorkresults) >80);

STUDENTNO STUDENTNAME ST GE AGE EMAIL

---------- -------------------- -- -- ---------- --------------------

1254657 Chris Wood CW M 21 chriswood@gmail.com

7007541 Elanga Junior EJ M 19 elangaj@gmail.com

1888754 Louisa Shina LS F 25 louisas@gmail.com

9835411 Mira Kay MK F 24 mirak@gmail.com

7289332 Sarah May SM F 24 sarahm@gmail.com

4) **TESTING VIEWS**

a)CREATE VIEW studentdetail AS

SELECT studentno, studentname, studentinitials, gender, age, email,

module\_moduleno , academicyear ,

semester , overalmark ,status

FROM student a, results b

WHERE a. studentno =b. student\_studentno ;

STUDENTNO STUDENTNAME ST GE AGE EMAIL MODULE\_ ACADEMI SEMES OVERALMARK STATU

---------- -------------------- -- -- ---------- -------------------- ------- ------- ----- ---------- -----

1254657 Chris Wood CW M 21 chriswood@gmail.com 5CA3333 2021/22 SEM1 81 Pass

1888754 Louisa Shina LS F 25 louisas@gmail.com 5CA3333 2021/22 SEM1 86.5 Pass

2295789 Karen Bibi KB F 20 Karenbm@gmail.com 9JK6272 2021/22 SEM2 78 Pass

2307653 Ashley Smith AS F 21 ashleys@gmail.com 6DE7683 2021/22 SEM1 64 Pass

3121622 Louis Jay LJ M 20 louisj@gmail.com 8BH2911 2021/22 SEM2 76 Pass

3509742 Joel Matip JM M 22 joelm@gmail.com 7TY7362 2021/22 SEM2 43 Pass

4489026 Paul Pogba PP M 30 paulp@gmail.com 6DE7683 2021/22 SEM1 58.5 Pass

4765392 Anthony Marshal AM M 20 anthonym@gmail.com 6DE7683 2021/22 SEM1 68 Pass

6743823 Angela Korman AK F 23 angelakor@gmail.com 2SE4444 2021/22 SEM2 38.5 Defer

7007541 Elanga Junior EJ M 19 elangaj@gmail.com 5CA3333 2021/22 SEM1 77.5 Pass

7289332 Sarah May SM F 24 sarahm@gmail.com 8BH2911 2021/22 SEM2 91.5 Pass

7563789 Vanessa Reece VR F 25 vanessar@gmail.com 2SE4444 2021/22 SEM2 83 Pass

7649062 Maison Mount MM M 34 maisonm@gmail.com 7TY7362 2021/22 SEM2 13 Defer

8734102 Thomas Patel TP M 29 thomasp@gmail.com 7TY7362 2021/22 SEM2 63 Pass

8884521 Bruce Breezy BB M 21 bruceb@gmail.com 1BN8354 2021/22 SEM2 51 Pass

9691208 Shiembo Mambo SM F 19 shiembom@gmail.com 2SE4444 2021/22 SEM2 36 Defer

9835411 Mira Kay MK F 24 mirak@gmail.com 8BH2911 2021/22 SEM2 89 Pass

b) **INSERT**

1) INSERT INTO studentdetail

VALUES(3535343, 'Pius Mein', 'PM', 'M', 25, 'pius@gmail.com', '8BH2911', '2021/22', 'SEM1', 78, 'pass');

A screen shot of a computer

Description automatically generated with low confidence

INSERT INTO studentdetail

VALUES(7464532, 'Carlton Banks', 'CB', 'F', 50, 'carl@gmail.com', '7TY7362', '2021/22', 'SEM2', 69, 'pass');

A screen shot of a computer

Description automatically generated with low confidence

INSERT INTO studentdetail

VALUES(5555242, 'Elvira Mani', 'EM', 'F', 40, 'Elvira@gmail.com', '2SE4444', '2021/22', 'SEM2', 80, 'Defer');

Text

Description automatically generated

2)**UPDATE**

update studentdetail

set email = 'chris@outlook.com', gender = 'F', STATUS = 'FAIL'

WHERE studentname = 'chris wood';

Text

Description automatically generated

update studentdetail

set email = 'vanessa@outlook.com', studentinitials = 'RV', STATUS = 'Defer'

WHERE studentname = 'Vanessa Reece';

Text

Description automatically generated

update studentdetail

set age = 30, studentinitials = 'YU',overalmark = 98

WHERE studentname = 'Shiembo Mambo';

Text

Description automatically generated

1. **DELETE**

DELETE FROM STUDENTDETAIL

WHERE studentNAME ='Chris Wood'

AND studentno = 1254657;

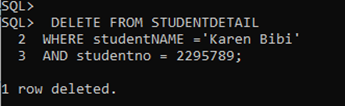
Text

Description automatically generated

DELETE FROM STUDENTDETAIL

WHERE studentNAME ='Karen Bibi'

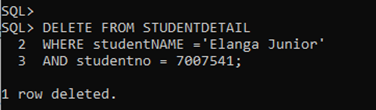
AND studentno = 2295789;



DELETE FROM STUDENTDETAIL

WHERE studentNAME ='Elanga Junior'

AND studentno = 7007541;



This error means that in the result of the subquery, the content of the updated data source (studentdetail)

is not unique, causing one row in the updated object to correspond to multiple rows in the data source.

Which means that all the rows in the studentdetail table are duplicated because they can be found in the main tables from which they were taken from to form the view table.

5**) AUDIT TABLE WHICH WILL CONTAIN THE DATA GENERATED BY THE AUDIT TRIGGERS**

CREATE TABLE audit\_results (

Audit\_type CHAR(10),

Username VARCHAR2(20),

Date\_of\_audit DATE,

academicyear VARCHAR2(7) NOT NULL,

semester VARCHAR2(5) NOT NULL,

CourseWorkresults NUMBER(5, 2) NOT NULL,

CourseWorkweighting NUMBER(5, 2) NOT NULL,

Examresults NUMBER(5, 2) NULL,

Examweighting NUMBER(5, 2) NULL,

overalmark NUMBER(5 ,2) NOT NULL,

status VARCHAR2(5) NOT NULL,

comments VARCHAR2(16) NULL,

module\_moduleno VARCHAR2(7) NOT NULL,

student\_studentno NUMBER(7) NOT NULL

);

**Triggers**

CREATE OR REPLACE TRIGGER audit\_result\_trg

BEFORE INSERT OR DELETE OR UPDATE ON results

FOR EACH ROW

DECLARE

-- Variable to store change type insert or update or delete

AuditType CHAR(10);

BEGIN

-- store the change in the variabe

IF INSERTING THEN AuditType := 'Insertion';

ELSIF UPDATING THEN AuditType := 'Updation';

ELSE AuditType := 'Deletion';

END IF;

-- username is current user who makes changes in the table

-- sysdate is current date

if updating or inserting then

INSERT INTO audit\_results

(

Audit\_type ,

Username ,

Date\_of\_audit ,

academicyear ,

semester,

CourseWorkresults,

CourseWorkweighting ,

Examresults,

Examweighting ,

overalmark,

status,

comments,

module\_moduleno ,

student\_studentno

)

VALUES (

AuditType,

USER,

SYSDATE,

:new.academicyear ,

:new.semester,

:new.CourseWorkresults,

:new.CourseWorkweighting ,

:new.Examresults,

:new.Examweighting ,

:new.overalmark,

:new.status,

:new.comments,

:new.module\_moduleno ,

:new.student\_studentno

);

else

INSERT INTO audit\_results

(

Audit\_type ,

Username ,

Date\_of\_audit ,

academicyear ,

semester,

CourseWorkresults,

CourseWorkweighting ,

Examresults,

Examweighting ,

overalmark,

status,

comments,

module\_moduleno ,

student\_studentno

)

VALUES (

AuditType,

USER,

SYSDATE,

:old.academicyear ,

:old.semester,

:old.CourseWorkresults,

:old.CourseWorkweighting ,

:old.Examresults,

:old.Examweighting ,

:old.overalmark,

:old.status,

:old.comments,

:old.module\_moduleno ,

:old.student\_studentno

);

end if;

END;

/

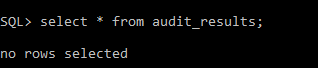
**Testing - Inserting record**

**Displaying records from results table before changes**

**A picture containing graphical user interface

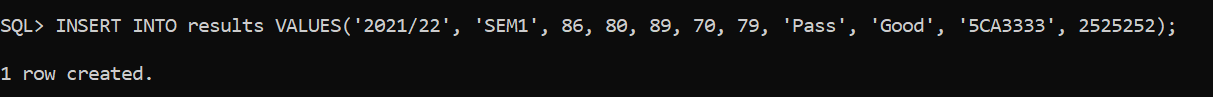
Description automatically generated**

**Displaying records from audit\_results table before changes**



**Insert one record into results table.**

INSERT INTO results VALUES('2021/22', 'SEM1', 86, 80, 89, 70, 79, 'Pass', 'Good', '5CA3333', 2525252);

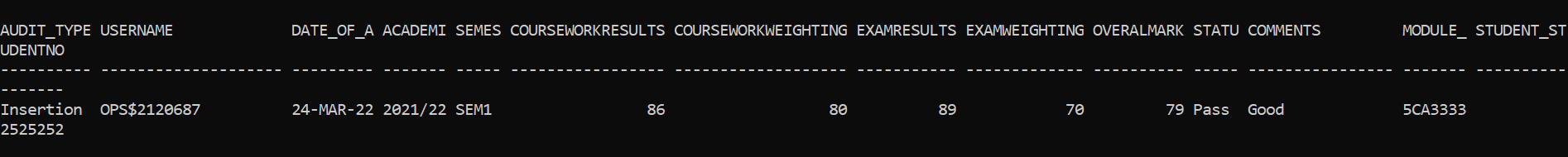
****

**Displaying records from results table after insert**

A computer screen capture

Description automatically generated with medium confidence

**Displaying records from audit\_results table after insert**



**Testing - updating record**

update results set CourseWorkresults = 67 where module\_moduleno ='5CA3333' and

student\_studentno = 2525252;

Text

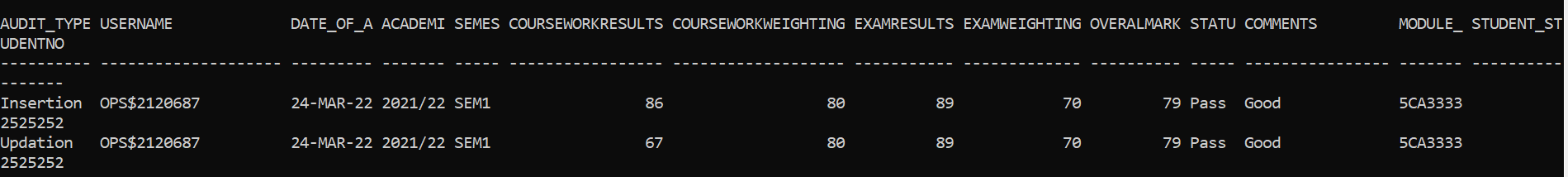
Description automatically generated

**Displaying records from results table after update**

A computer screen capture

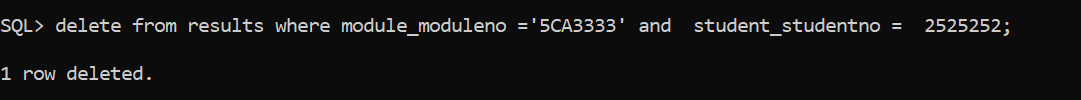
Description automatically generated with medium confidence

**Displaying records from audit\_results table after update**



**Testing - deleting record**

delete from results where module\_moduleno ='5CA3333' and student\_studentno = 2525252;

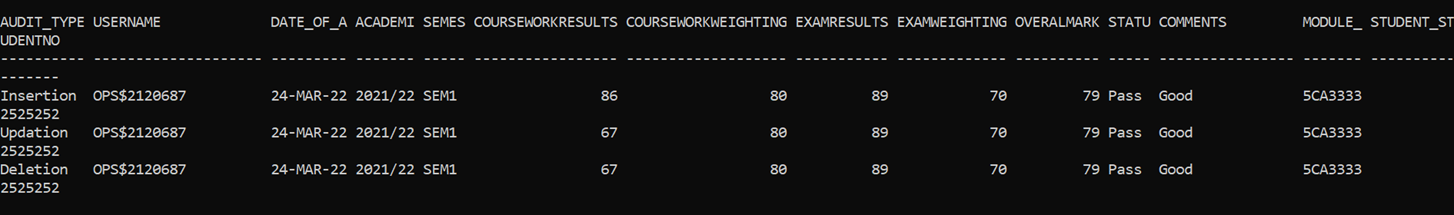


**Displaying records from results table after delete**

**Graphical user interface

Description automatically generated with medium confidence**

**Displaying records from audit\_results table after delete**

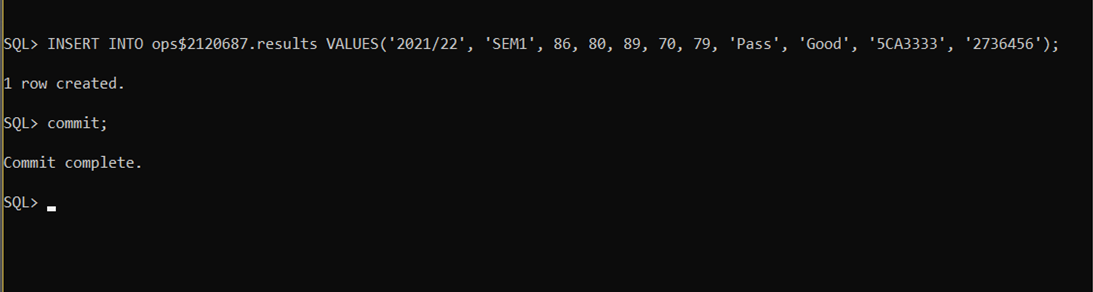


**Testing the trigger with the non-owner of the table.**

INSERT INTO ops$2120687.results VALUES (‘2012/22’, ‘SEM1’, 86, 80, 89, 70, 79,  ‘Pass’,  ‘Good’,  ‘5CA3333’, ‘2736456’);

Update ops$2120687.results set CourseWorkresults = 67 where module\_moduleno = ‘5CA3333’ and Student\_studentno = 2736456 ;

Delete from ops$2120687.results where module\_moduleno = ’5CA3333’ and syudent\_studentno = 2736456;



Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**Proof of audit by a non-owner**

Text

Description automatically generated

**The problems that can arise in a live system**

**ISSUES THAT CAN ARISE**

* **The above audit\_result table will not save old values for update operation. So table can be redesigned to save old values in all columns.**

CREATE TABLE audit\_results (

Audit\_type CHAR(10),

Username VARCHAR2(20),

Date\_of\_audit DATE,

academicyear VARCHAR2(7) NOT NULL,

semester VARCHAR2(5) NOT NULL,

CourseWorkresults NUMBER(5, 2) NOT NULL,

CourseWorkweighting NUMBER(5, 2) NOT NULL,

Examresults NUMBER(5, 2) NULL,

Examweighting NUMBER(5, 2) NULL,

overalmark NUMBER(5 ,2) NOT NULL,

status VARCHAR2(5) NOT NULL,

comments VARCHAR2(16) NULL,

module\_moduleno VARCHAR2(7) NOT NULL,

student\_studentno NUMBER(7) NOT NULL,

old\_academicyear VARCHAR2(7) NOT NULL,

old\_semester VARCHAR2(5) NOT NULL,

old\_CourseWorkresults NUMBER(5, 2) NOT NULL,

old\_CourseWorkweighting NUMBER(5, 2) NOT NULL,

old\_Examresults NUMBER(5, 2) NULL,

old\_Examweighting NUMBER(5, 2) NULL,

old\_overalmark NUMBER(5 ,2) NOT NULL,

old\_status VARCHAR2(5) NOT NULL,

old\_comments VARCHAR2(16) NULL,

old\_module\_moduleno VARCHAR2(7) NOT NULL,

sold\_student\_studentno NUMBER(7) NOT NULL

);

* Another issue with using triggers is that they often have permissions that users don't, which can cause problems when a user needs to bypass the audit triggers in a database.
* malicious hackers can use SQL Server audit triggers to their advantage. If hackers know a system is equipped with triggers, they may repeatedly attack it to create a diversion or to see how quickly the DBA responds and what security measures he takes.

**OTHER METHODS USED FOR AUDITING PURPOSES**

* Database backup can be used for auditing purposes as an alternative method.
* SQL server management studio can also be used. It is a database engine that stores data and allows you to query it. SQL Server Management Studio Express is a management tool that provides a graphic interface for working with SQL Server database servers.

6) `**DATA WAREHOUSE**

CREATE TABLE data\_warehouse(

category\_marker VARCHAR2(26) NOT NULL,

category VARCHAR2(26) NOT NULL,

first\_year\_marker VARCHAR2(26) NOT NULL,

level\_of\_study VARCHAR2(26) NOT NULL,

mode\_of\_study VARCHAR2(26) NOT NULL,

country\_of\_he\_provider VARCHAR2(26) NOT NULL,

academic\_year VARCHAR2(26) NOT NULL,

student\_id Number(38) NOT NULL,

percentage NUMBER(38,2);

Graphical user interface, text

Description automatically generated

b) Used SQL developer to import the data into my database.

C) In Online Transaction Processing (OLTP) system, the queries which are used are INSERT, UPDATE and DELETE. This is because the OLTP is characterized by many short online transactions The focus for OLTP systems is very fast processing of requests, ensuring data integrity in multiple access environments and efficiency, measured by the number of transactions per second whereas in an Online Analytical Processing (OLAP) the queries are SELECT, WHERE, WITH and FROM. OLAP deals with historical data or archival data. OLAP query processing, response time is a measure of performance, and its applications are widely used in data mining.

D)SELECT \* FROM DATA\_WAREHOUSE

WHERE mode\_of\_study = 'Part-time'

AND CATEGORY = 'Black';

A picture containing text, computer, electronics

Description automatically generated

SELECT Category, student\_id, percentage FROM data\_warehouse

WHERE category = 'Male';

A picture containing table

Description automatically generated