Database Report

Authors: Samsu Miah, Hussein Ahmed Tejan and Sanjeevan Paruthirajan

Index

Relational Schema

- Some primary key: primary key
- (FK) keyname: foreign key

- Student (Student ID: Integer, Name: String, Degree: String)

- Exam (Exam ID: Integer, (FK) Module Number: Integer, Semester: Char, Year: Integer, Lecturer: String, Attempt: Integer, Weighting: double, Outcome: String, Marks: String, (FK) QuestionsID: Integer)

- Subject (Subject ID: Integer, Subject Name: String,)

- Module (Module Number: Integer, Module Name: String)

- Coursework (Coursework ID: Integer, (FK) Module Number: Integer, Semester: Char, Year: Integer, Lecturer: String, Mark: String, Weighting: double, Outcome: String)

Relational Schema

Staff (Staff ID: Integer, Staff Name: String, Department: String)

Grade Report ((FK) Student ID: Integer, (FK) Subject ID: Integer, (FK) Module Number: String, Module Grade: String)

Lecturer (Lecturer ID: String, Subject Name: String, Module Name: String)

Questions (QuestionsID: Integer, Marks: String)

Relational Schema

Student

Student ID	Name	Degree	Subject ID

Exam

Exam ID	Module Number	Semester	Year	Author	Attempt	Weighting	Outcome	Mark	QuestionsID

Subject

Subject ID	Subject Name

Module

Module Number	Module Name

Coursework

Coursework ID	Module Number	Semester	Year	LecturerID	Weighting	Outcome	Mark

Staff

Staff ID	Staff Name	Department

Grade Report

Student ID	Subject ID	Overall Mark	Grade	Module Number

Lecturer

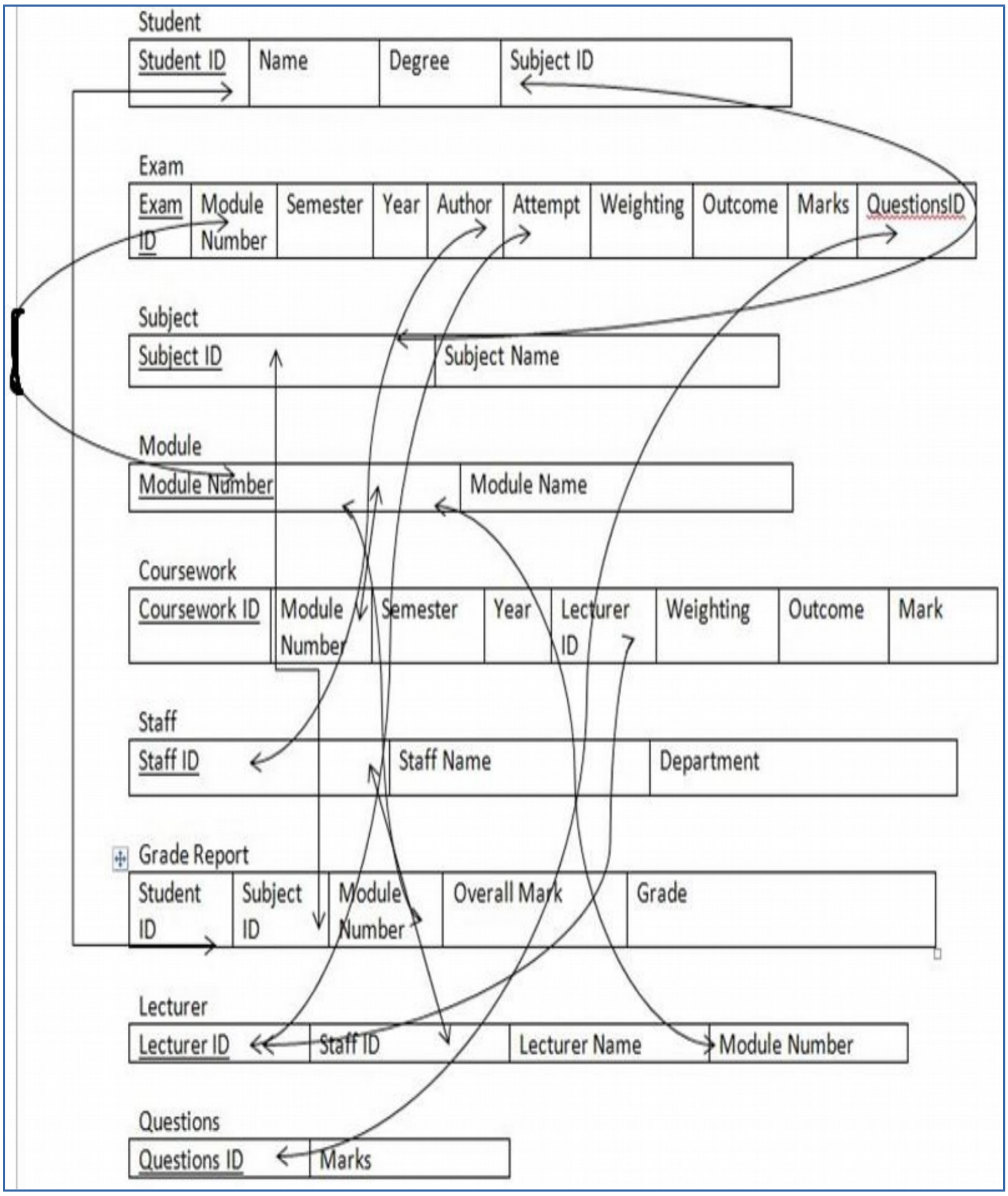
Lecturer ID	Staff ID	Lecturer Name	Module Number

Questions

QuestionsID	Marks

Relational Schema

The below diagram was explained in notational form in the above table as the arrows from one table's column to another table's column denotes a primary key from a single table being used as a foreign key in another table, for example StudentID (from Student table) is present in other tables such as Grade Report to allow data to be collectively matched just like in a real University System.



Student

Student ID	Name	Degree	Subject ID

Exam

Exam ID	Module Number	Semester	Year	Author	Attempt	Weighting	Outcome	Marks	QuestionsID

Subject

Subject ID	Subject Name

Module

Module Number	Module Name

Coursework

Coursework ID	Module Number	Semester	Year	Lecturer ID	Weighting	Outcome	Mark

Staff

Staff ID	Staff Name	Department

Grade Report

Student ID	Subject ID	Module Number	Overall Mark	Grade

Lecturer

Lecturer ID	Staff ID	Lecturer Name	Module Number

Questions

Questions ID	Marks

Relational Schema

Student {Student ID, Name, Degree, Subject ID} – first normal form

Exam {Exam Id, Module number, Semester, Year, Author, Attempt, Weighting, Outcome, Marks, Question ID,} – first normal form

Subject {Subject ID, Subject name}-first normal form

Module {Module number, Module name} – first normal form

Coursework {Coursework ID, Module number, Semester, Year, Lecturer ID, Weighting, Outcome, Mark}- First Normal form

Staff {Staff Id, Staff name, Department}-first normal

Grade Report {Student ID, Subject ID, Module name, Overall mark, Grade}

Lecturer {Lecturer ID, Staff Id, Lecturer Name, Module Number}

Questions {Question Id, Marks}

Second Normal Formal

Student {Student ID, Name, Degree id, Subject ID}- second normal form

Degree _class {Degree id, Degree classification}- second normal form

Exam {Exam Id, Module number, SyID, Author, Attempt, Weighting, Outcome, Marks, Question ID,} – second normal form

Sem_year {Syid, Semester, yid}- second normal form
Year {yid, Year}-second normal form

Subject {Subject ID, Subject name}-second normal form

Module {Module number, Module name} – second normal form

Relational Schema

Coursework {Coursework Id, Module number, Syid, Lecturer ID, Weighting, Outcome, Mark}-Second Normal form

Staff {Staff Id, Staff name, DId}-Second normal form

Dept {Did, Department name} – second normal form

Staff {Staff Id, Staff name, Did}-first normal

Grade Report {Student ID, Subject ID, Module id, Overall mark, Grade} – second normal form

Lecturer {Lecturer ID, Staff Id, Lecturer Name, Module Number} -second normal form

Questions {Question Id, Marks}-second normal form

Student {Student ID, Name, Degree id, Subject ID}- third normal form

Degree _class {Degree id, Degree classification}- third normal form

Exam {Exam Id, Module number, SyID} – third normal form

Exam_det {Exam id, Author, Attempt, Weighting, Outcome, Marks, Question iD} – third normal form

Sem_year {Syid, Semester, yid}-third normal form Year {yid, Year}-third normal form

Subject {Subject ID, Subject name}-third normal form

Module {Module number, Module name} –third normal form

Coursework {Coursework Id, Module number, Syid, Lecturer Id}-Third Normal form

Coursework_marks {Coursework id, Weighting, Outcome, Mark}

Relational Schema

Staff {Staff Id, Staff name, DId}-Third normal form

Dept {Did, Department name} – Third normal form

Staff {Staff Id, Staff name, Did}-Third normal

Grade Report {Student ID, Subject ID} – third normal form

Sub_in_modules {Subject id, Module id, Overall mark, Grade} – third normal form

Lecturer {Lecturer ID, Lecturer Name, Module Number} -Third normal form

Questions {Question Id, Marks}-third normal form

Student {Student ID, Name, Degree id, Subject ID}- third normal form

Relation 1: Student ID, Name – Relation 2: Degree id, Subject id

Degree _class {Degree id, Degree classification}- third normal form

Exam {Exam Id, Module number, SyID} – third normal form

Exam_det {Exam id, Author, Attempt, Weighting, Outcome, Marks, Question iD} – third normal form

Sem_year {Syid, Semester, yid}-third normal form Year {yid, Year}-third normal form.

Subject {Subject ID, Subject name}-third normal form

Module {Module number, Module name} –third normal form

Coursework {Coursework Id, Module number, Syid, Lecturer Id}-Third Normal form

Coursework_marks {Coursework id, Weighting, Outcome, Mark}

Staff {Staff Id, Staff name, DId}-Third normal form Dept {Did,

Relational Schema

Department name} – Third normal form

Staff {Staff Id, Staff name, Did}-Third normal

Grade Report {Student ID, Subject ID} – third normal form

Sub_in_modules {Subject id, Module id, Overall mark, Grade} – third normal form

Lecturer {Lecturer ID, Lecturer Name, Module Number} -Third normal form

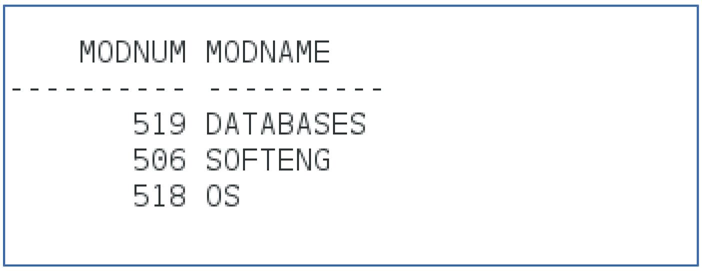
Questions {Question Id, Marks}-third normal form

Database table commands

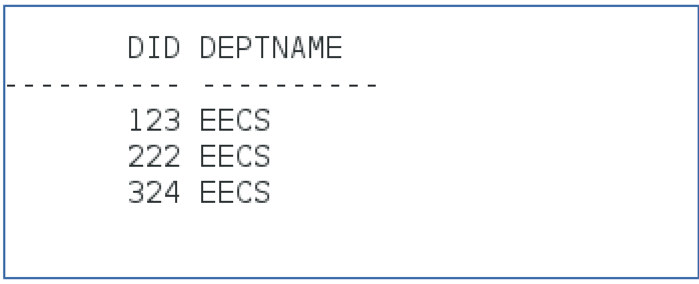
```
DROP TABLE GRADEREPORT;
    DROP TABLE STUDENT;
    DROP TABLE DEGREE_CLASS ;
    DROP TABLE COURSEWORK_MARKS;
    DROP TABLE COURSEWORK;

    DROP TABLE LECTURER;
    DROP TABLE EXAM_DET;
    DROP TABLE EXAM;

    DROP TABLE SUBINMOD;
    DROP TABLE MODULE;
    DROP TABLE SUBJECT;
    DROP TABLE SEM_YEAR;
    DROP TABLE YEAR;



    DROP TABLE QUESTIONS;



    DROP TABLE STAFF;
    DROP TABLE DEPT;
    DROP VIEW SOMECRAP;
    DROP VIEW CRAP;
    DROP VIEW LNAME_MODNAME;


    SET TERMOUT ON





         CREATE TABLE DEGREE_CLASS(
         DEGREEID NUMBER (10) NOT NULL PRIMARY KEY,
         DEGREECLASS VARCHAR (10) NOT NULL
         );

         CREATE TABLE MODULE(
         MODNUM NUMBER (10) NOT NULL PRIMARY KEY ,
         MODNAME VARCHAR (10) NOT NULL
         );



         CREATE TABLE SUBJECT(
         SUBJECTID NUMBER (10) NOT NULL PRIMARY KEY,
         SUBJECTNAME VARCHAR (500)NOT NULL
         );



         CREATE TABLE LECTURER(
         LID NUMBER (10) NOT NULL PRIMARY KEY,
         LNAME VARCHAR (10) NOT NULL,
         MODNUM NUMBER(10) NOT NULL,
     FOREIGN KEY (MODNUM) REFERENCES MODULE(MODNUM)
         );

         CREATE TABLE YEAR(
         YID NUMBER (10) NOT NULL PRIMARY KEY ,
         YEAR VARCHAR (10) NOT NULL
         );


         CREATE TABLE SEM_YEAR(
         SYID NUMBER (10) NOT NULL PRIMARY KEY ,
         SEMESTER VARCHAR (10) NOT NULL,
         YID NUMBER (10) NOT NULL,
         FOREIGN KEY (YID) REFERENCES YEAR(YID)
         );


         CREATE TABLE STUDENT(
         STUDENTID NUMBER (10) NOT NULL PRIMARY KEY,
         STUDENTNAME VARCHAR (10) NOT NULL,
         DEGREEID NUMBER (10) ,
         SUBJECTID NUMBER (10),
         FOREIGN KEY (DEGREEID) REFERENCES DEGREE_CLASS(DEGREEID),
         FOREIGN KEY (SUBJECTID) REFERENCES SUBJECT(SUBJECTID)
         );
         CREATE TABLE EXAM(
         EXAMID NUMBER (10) NOT NULL PRIMARY KEY,
         MODNUM NUMBER (10) NOT NULL,
         SYID NUMBER (10) NOT NULL,
         FOREIGN KEY (MODNUM) REFERENCES MODULE(MODNUM),
         FOREIGN KEY (SYID) REFERENCES SEM_YEAR(SYID)
         );

         CREATE TABLE QUESTIONS(
         QID NUMBER (10) NOT NULL PRIMARY KEY,
         MARKS NUMBER(10) NOT NULL
         );

         CREATE TABLE EXAM_DET(
         EXAMID NUMBER (10) NOT NULL,
         AUTHOR VARCHAR (10)  NOT NULL,
         ATTEMPT NUMBER (10) NOT NULL,
         WEIGHTING VARCHAR (10) NOT NULL,
         OUTCOME VARCHAR (10) NOT NULL,
         MARKS NUMBER (10)NOT NULL,
         QID NUMBER (10) NOT NULL,
         FOREIGN KEY (EXAMID) REFERENCES EXAM(EXAMID),
         FOREIGN KEY (QID) REFERENCES QUESTIONS(QID)
         );

         CREATE TABLE COURSEWORK(
         CWID      NUMBER (10) NOT NULL PRIMARY KEY,
         MODNUM     NUMBER (10) NOT NULL,
         SYID       NUMBER (10) NOT NULL,
         LID       NUMBER (10) NOT NULL,
     FOREIGN KEY (MODNUM) REFERENCES MODULE(MODNUM),
         FOREIGN KEY (SYID) REFERENCES SEM_YEAR(SYID),
     FOREIGN KEY (LID) REFERENCES LECTURER(LID)
         );



         CREATE TABLE COURSEWORK_MARKS(
         CWID      NUMBER (10) NOT NULL ,
         WEIGHTING VARCHAR (10) NOT NULL,
         YID       NUMBER (10) NOT NULL,
         OUTCOME   VARCHAR (10)  NOT NULL,
         FOREIGN KEY (CWID) REFERENCES COURSEWORK(CWID),
         FOREIGN KEY (YID) REFERENCES YEAR(YID)
         );
```

```
CREATE TABLE DEPT(
DID NUMBER (10) NOT NULL PRIMARY KEY,
DEPTNAME VARCHAR (10) NOT NULL
);


CREATE TABLE STAFF(
STAFFID NUMBER (10) NOT NULL PRIMARY KEY,
STAFFNAME VARCHAR (10) NOT NULL,
DID NUMBER (10) NOT NULL,
FOREIGN KEY (DID) REFERENCES DEPT(DID)
);

CREATE TABLE GRADEREPORT(
STUDENTID NUMBER (10) NOT NULL,
SUBJECTID NUMBER (10)  NOT NULL,
FOREIGN KEY (STUDENTID) REFERENCES STUDENT(STUDENTID),
FOREIGN KEY (SUBJECTID) REFERENCES SUBJECT(SUBJECTID)
);

CREATE TABLE SUBINMOD(
SUBJECTID NUMBER (10) NOT NULL,
MODNUM NUMBER (10) NOT NULL,
OVERALLMARK NUMBER(10) NOT NULL,
FOREIGN KEY (SUBJECTID) REFERENCES SUBJECT(SUBJECTID),
FOREIGN KEY (MODNUM) REFERENCES MODULE(MODNUM)
);
```

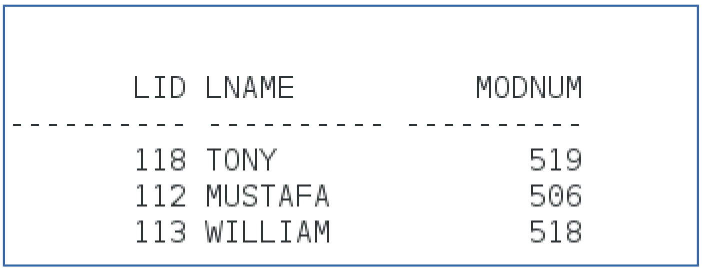
Database test data



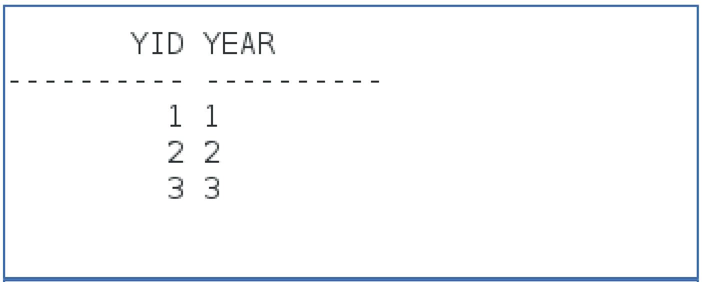
```
    MODNUM MODNAME
---------- ----------
       519 DATABASES
       506 SOFTENG
       518 OS
```



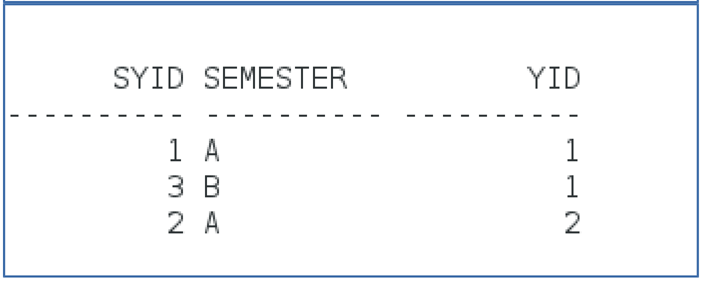
```
       DID DEPTNAME
---------- ----------
       123 EECS
       222 EECS
       324 EECS
```



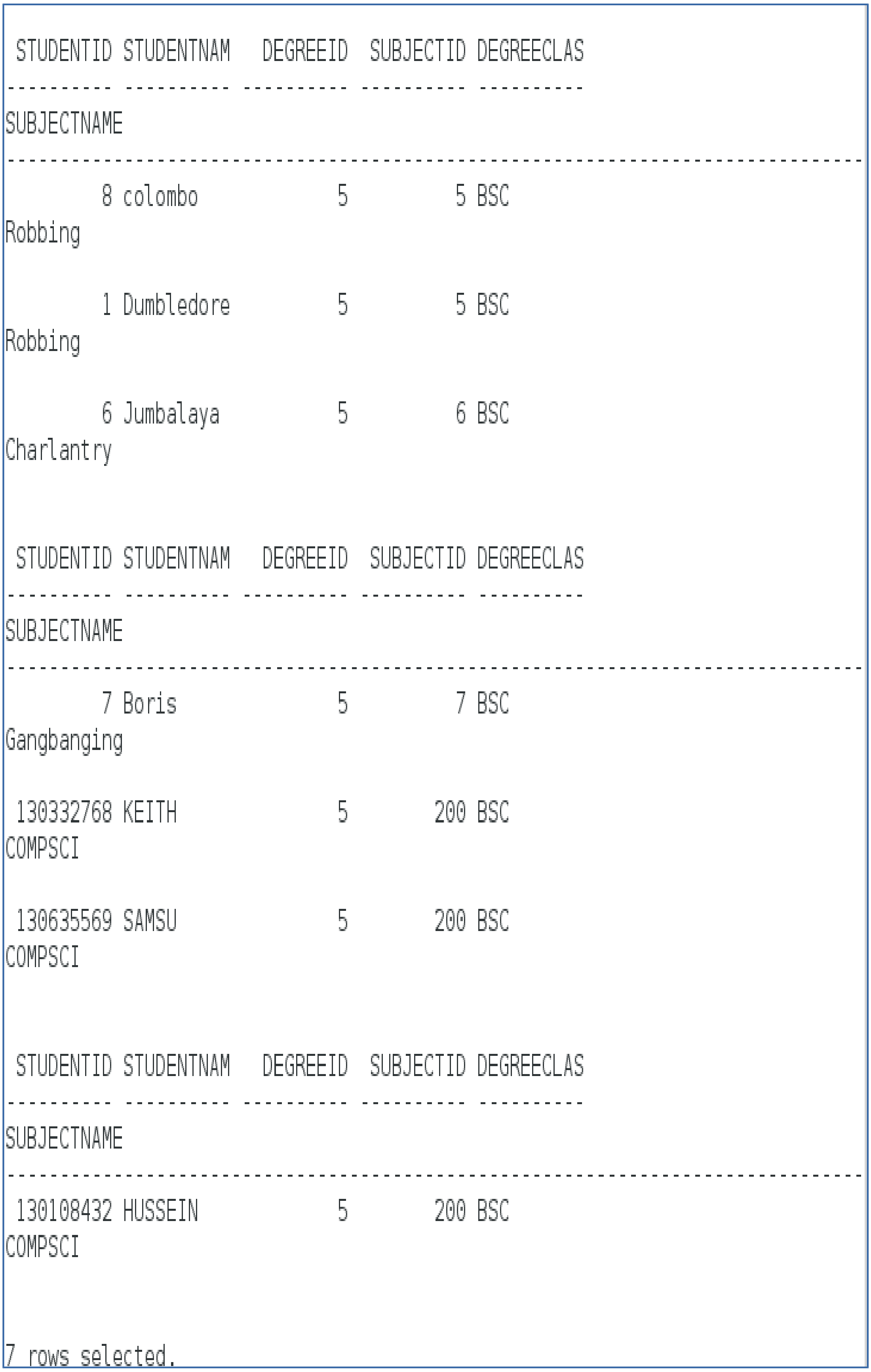
```
       LID LNAME          MODNUM
---------- ---------- ----------
       118 TONY              519
       112 MUSTAFA           506
       113 WILLIAM           518
```



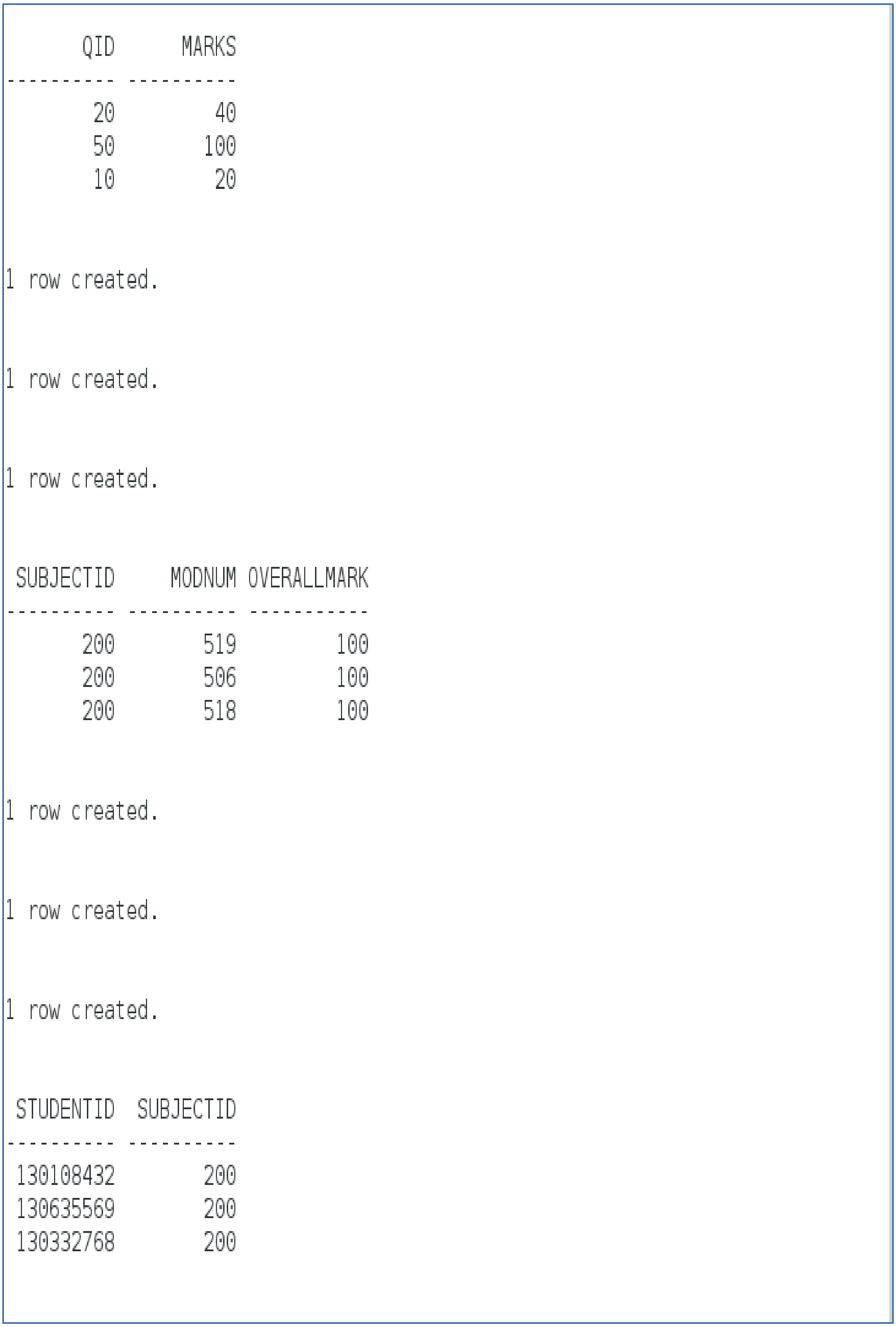
```
       YID YEAR
---------- ----------
         1 1
         2 2
         3 3
```



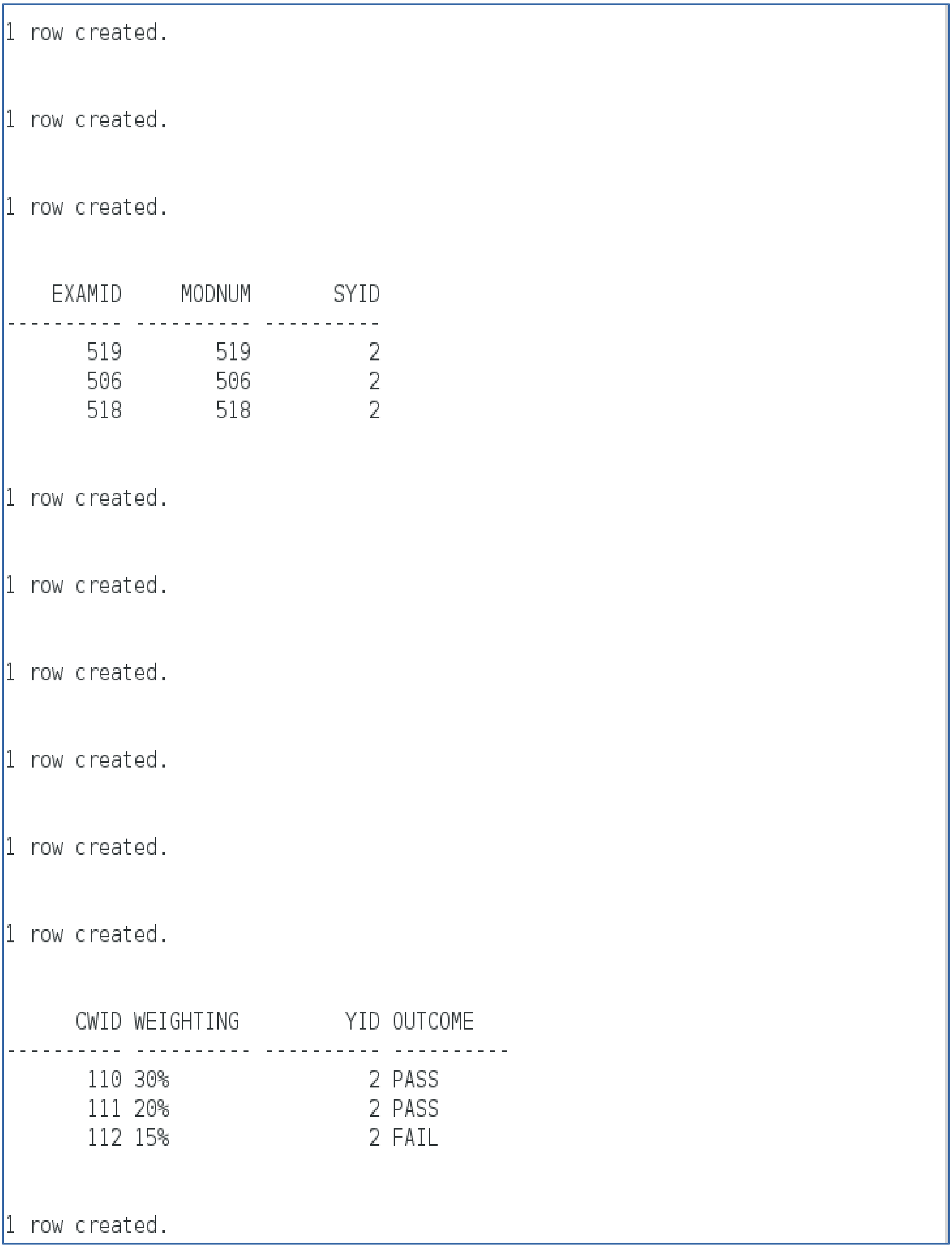
```
      SYID SEMESTER          YID
---------- ---------- ----------
         1 A                   1
         3 B                   1
         2 A                   2
```



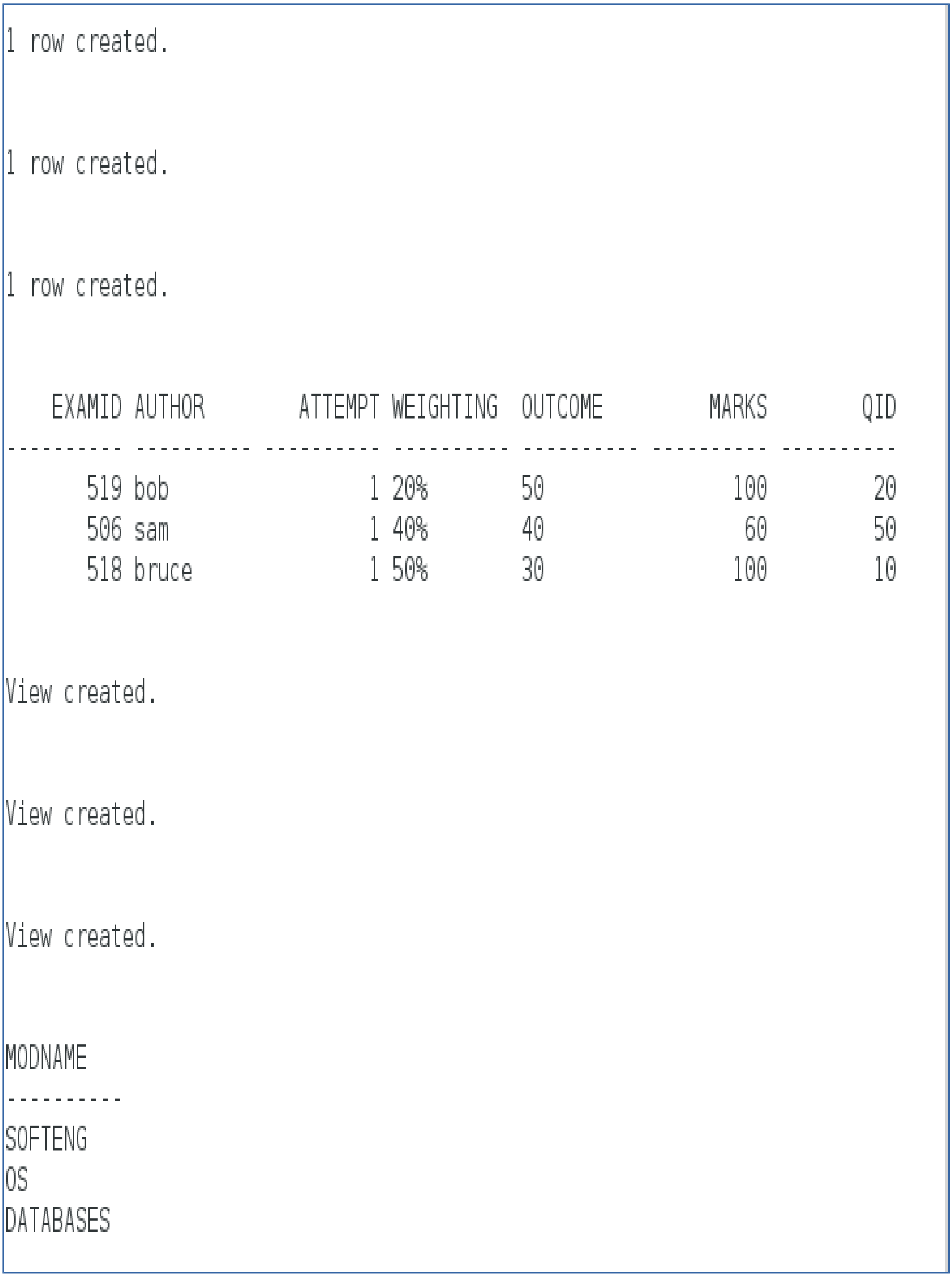
```
 STUDENTID STUDENTNAM   DEGREEID  SUBJECTID DEGREECLAS
---------- ---------- ---------- ---------- ----------
SUBJECTNAME
--------------------------------------------------------------------------------
         8 colombo             5          5 BSC
Robbing

         1 Dumbledore          5          5 BSC
Robbing

         6 Jumbalaya           5          6 BSC
Charlantry


 STUDENTID STUDENTNAM   DEGREEID  SUBJECTID DEGREECLAS
---------- ---------- ---------- ---------- ----------
SUBJECTNAME
--------------------------------------------------------------------------------
         7 Boris               5          7 BSC
Gangbanging

 130332768 KEITH               5        200 BSC
COMPSCI

 130635569 SAMSU               5        200 BSC
COMPSCI


 STUDENTID STUDENTNAM   DEGREEID  SUBJECTID DEGREECLAS
---------- ---------- ---------- ---------- ----------
SUBJECTNAME
--------------------------------------------------------------------------------
 130108432 HUSSEIN             5        200 BSC
COMPSCI



7 rows selected.
```



```
       QID      MARKS
---------- ----------
        20         40
        50        100
        10         20


1 row created.


1 row created.


1 row created.


 SUBJECTID     MODNUM OVERALLMARK
---------- ---------- -----------
       200        519         100
       200        506         100
       200        518         100


1 row created.


1 row created.


1 row created.


 STUDENTID  SUBJECTID
---------- ----------
 130108432        200
 130635569        200
 130332768        200
```

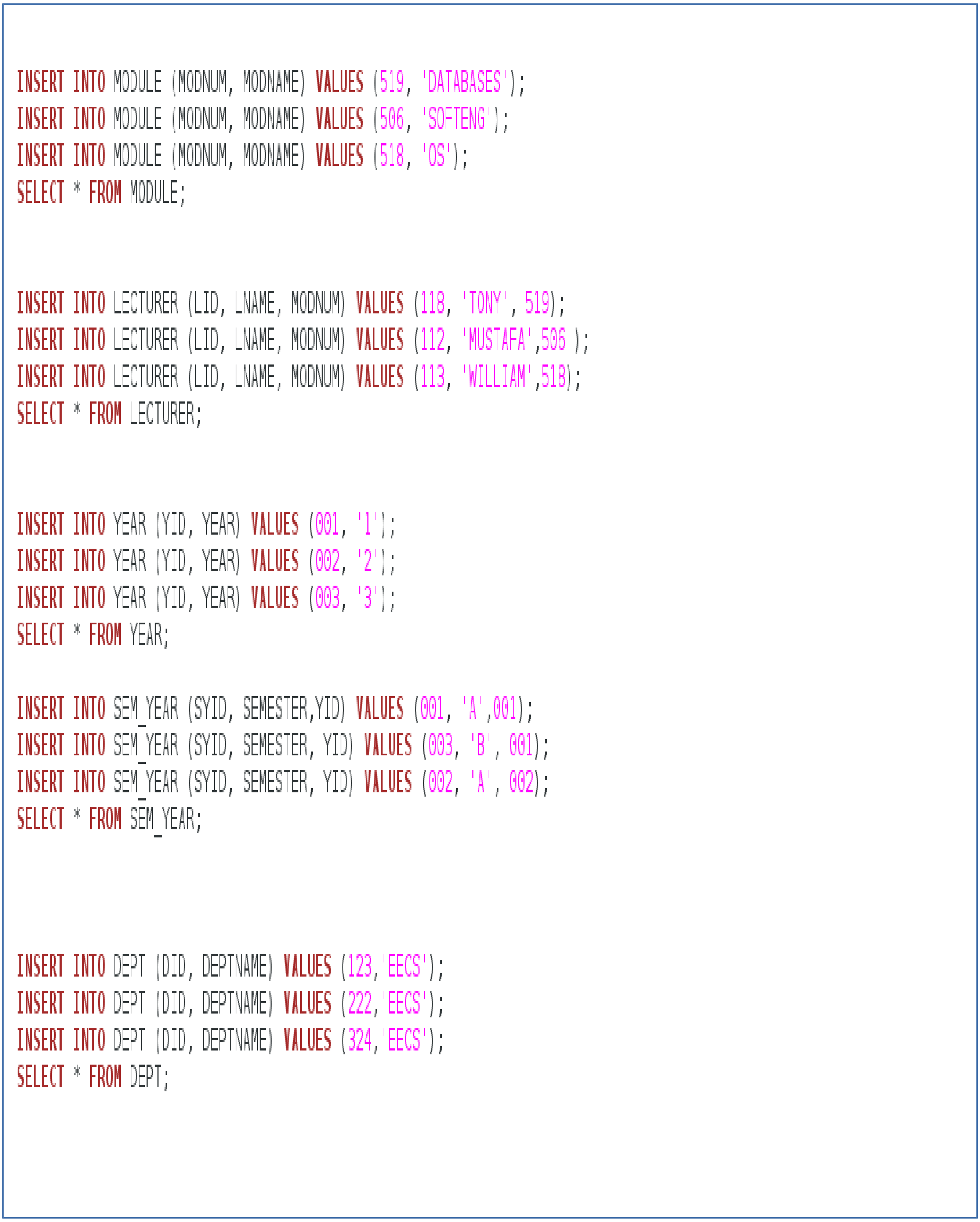


```
1 row created.


1 row created.


1 row created.


    EXAMID     MODNUM       SYID
---------- ---------- ----------
       519        519          2
       506        506          2
       518        518          2


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


      CWID WEIGHTING         YID OUTCOME
---------- ---------- ---------- ----------
       110 30%                 2 PASS
       111 20%                 2 PASS
       112 15%                 2 FAIL


1 row created.
```



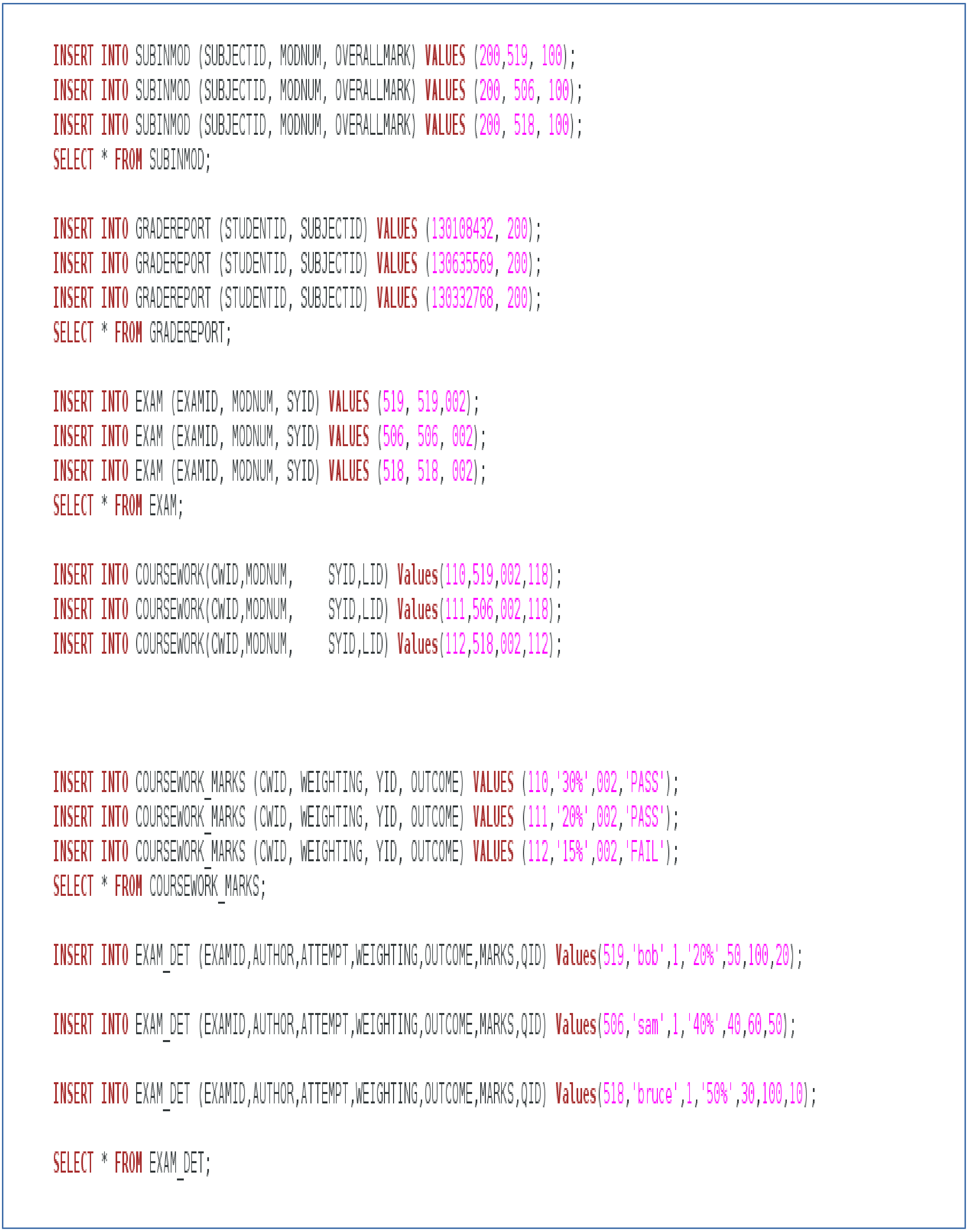
```
1 row created.


1 row created.


1 row created.


    EXAMID AUTHOR        ATTEMPT WEIGHTING  OUTCOME         MARKS        QID
---------- ---------- ---------- ---------- ---------- ---------- ----------
       519 bob                 1 20%        50                100         20
       506 sam                 1 40%        40                 60         50
       518 bruce               1 50%        30                100         10


View created.


View created.


View created.


MODNAME
----------
SOFTENG
OS
DATABASES
```

Insert Statements



```
INSERT INTO MODULE (MODNUM, MODNAME) VALUES (519, 'DATABASES');
INSERT INTO MODULE (MODNUM, MODNAME) VALUES (506, 'SOFTENG');
INSERT INTO MODULE (MODNUM, MODNAME) VALUES (518, 'OS');
SELECT * FROM MODULE;



INSERT INTO LECTURER (LID, LNAME, MODNUM) VALUES (118, 'TONY', 519);
INSERT INTO LECTURER (LID, LNAME, MODNUM) VALUES (112, 'MUSTAFA',506 );
INSERT INTO LECTURER (LID, LNAME, MODNUM) VALUES (113, 'WILLIAM',518);
SELECT * FROM LECTURER;



INSERT INTO YEAR (YID, YEAR) VALUES (001, '1');
INSERT INTO YEAR (YID, YEAR) VALUES (002, '2');
INSERT INTO YEAR (YID, YEAR) VALUES (003, '3');
SELECT * FROM YEAR;

INSERT INTO SEM_YEAR (SYID, SEMESTER,YID) VALUES (001, 'A',001);
INSERT INTO SEM_YEAR (SYID, SEMESTER, YID) VALUES (003, 'B', 001);
INSERT INTO SEM_YEAR (SYID, SEMESTER, YID) VALUES (002, 'A', 002);
SELECT * FROM SEM_YEAR;




INSERT INTO DEPT (DID, DEPTNAME) VALUES (123,'EECS');
INSERT INTO DEPT (DID, DEPTNAME) VALUES (222,'EECS');
INSERT INTO DEPT (DID, DEPTNAME) VALUES (324,'EECS');
SELECT * FROM DEPT;
```

```sql
INSERT INTO QUESTIONS (QID,MARKS) Values (20,40);
INSERT INTO QUESTIONS (QID,MARKS) Values (50,100);
INSERT INTO QUESTIONS (QID,MARKS) Values (10,20);









INSERT INTO DEGREE_CLASS (DEGREEID,DEGREECLASS) VALUES (5,'BSC');
INSERT INTO DEGREE_CLASS (DEGREEID,DEGREECLASS) VALUES (6,'MSC');
INSERT INTO DEGREE_CLASS (DEGREEID,DEGREECLASS) VALUES (7,'PHD');
INSERT INTO SUBJECT (SUBJECTID,SUBJECTNAME) VALUES (5,'Robbing');
INSERT INTO SUBJECT (SUBJECTID,SUBJECTNAME) VALUES (6,'Charlantry');
INSERT INTO SUBJECT (SUBJECTID,SUBJECTNAME) VALUES (7,'Gangbanging');
INSERT INTO SUBJECT (SUBJECTID, SUBJECTNAME) VALUES (200, 'COMPSCI');
INSERT INTO SUBJECT (SUBJECTID, SUBJECTNAME) VALUES (300, 'ENGLISH');
INSERT INTO SUBJECT (SUBJECTID, SUBJECTNAME) VALUES (528, 'MATHS');



INSERT INTO STUDENT (STUDENTID,STUDENTNAME,DEGREEID,SUBJECTID) VALUES (1,'Dumbledore',5,5);
INSERT INTO STUDENT (STUDENTID,STUDENTNAME,DEGREEID,SUBJECTID) VALUES (6,'Jumbalaya',5,6);
INSERT INTO STUDENT (STUDENTID,STUDENTNAME,DEGREEID,SUBJECTID) VALUES (7,'Boris',5,7);
INSERT INTO STUDENT (STUDENTID,STUDENTNAME,DEGREEID,SUBJECTID) VALUES (8,'colombo',5,5);
INSERT INTO STUDENT (STUDENTID, STUDENTNAME, DEGREEID, SUBJECTID) VALUES (130108432, 'HUSSEIN', 5, 200);
INSERT INTO STUDENT (STUDENTID, STUDENTNAME, DEGREEID, SUBJECTID) VALUES (130635569, 'SAMSU', 5, 200);
INSERT INTO STUDENT (STUDENTID, STUDENTNAME, DEGREEID, SUBJECTID) VALUES (130332768, 'KEITH', 5, 200);
```



```sql
INSERT INTO SUBINMOD (SUBJECTID, MODNUM, OVERALLMARK) VALUES (200,519, 100);
INSERT INTO SUBINMOD (SUBJECTID, MODNUM, OVERALLMARK) VALUES (200, 506, 100);
INSERT INTO SUBINMOD (SUBJECTID, MODNUM, OVERALLMARK) VALUES (200, 518, 100);
SELECT * FROM SUBINMOD;

INSERT INTO GRADEREPORT (STUDENTID, SUBJECTID) VALUES (130108432, 200);
INSERT INTO GRADEREPORT (STUDENTID, SUBJECTID) VALUES (130635569, 200);
INSERT INTO GRADEREPORT (STUDENTID, SUBJECTID) VALUES (130332768, 200);
SELECT * FROM GRADEREPORT;

INSERT INTO EXAM (EXAMID, MODNUM, SYID) VALUES (519, 519,002);
INSERT INTO EXAM (EXAMID, MODNUM, SYID) VALUES (506, 506, 002);
INSERT INTO EXAM (EXAMID, MODNUM, SYID) VALUES (518, 518, 002);
SELECT * FROM EXAM;

INSERT INTO COURSEWORK(CWID,MODNUM,    SYID,LID) Values(110,519,002,118);
INSERT INTO COURSEWORK(CWID,MODNUM,    SYID,LID) Values(111,506,002,118);
INSERT INTO COURSEWORK(CWID,MODNUM,    SYID,LID) Values(112,518,002,112);




INSERT INTO COURSEWORK_MARKS (CWID, WEIGHTING, YID, OUTCOME) VALUES (110,'30%',002,'PASS');
INSERT INTO COURSEWORK_MARKS (CWID, WEIGHTING, YID, OUTCOME) VALUES (111,'20%',002,'PASS');
INSERT INTO COURSEWORK_MARKS (CWID, WEIGHTING, YID, OUTCOME) VALUES (112,'15%',002,'FAIL');
SELECT * FROM COURSEWORK_MARKS;

INSERT INTO EXAM_DET (EXAMID,AUTHOR,ATTEMPT,WEIGHTING,OUTCOME,MARKS,QID) Values(519,'bob',1,'20%',50,100,20);

INSERT INTO EXAM_DET (EXAMID,AUTHOR,ATTEMPT,WEIGHTING,OUTCOME,MARKS,QID) Values(506,'sam',1,'40%',40,60,50);

INSERT INTO EXAM_DET (EXAMID,AUTHOR,ATTEMPT,WEIGHTING,OUTCOME,MARKS,QID) Values(518,'bruce',1,'50%',30,100,10);

SELECT * FROM EXAM_DET;
```

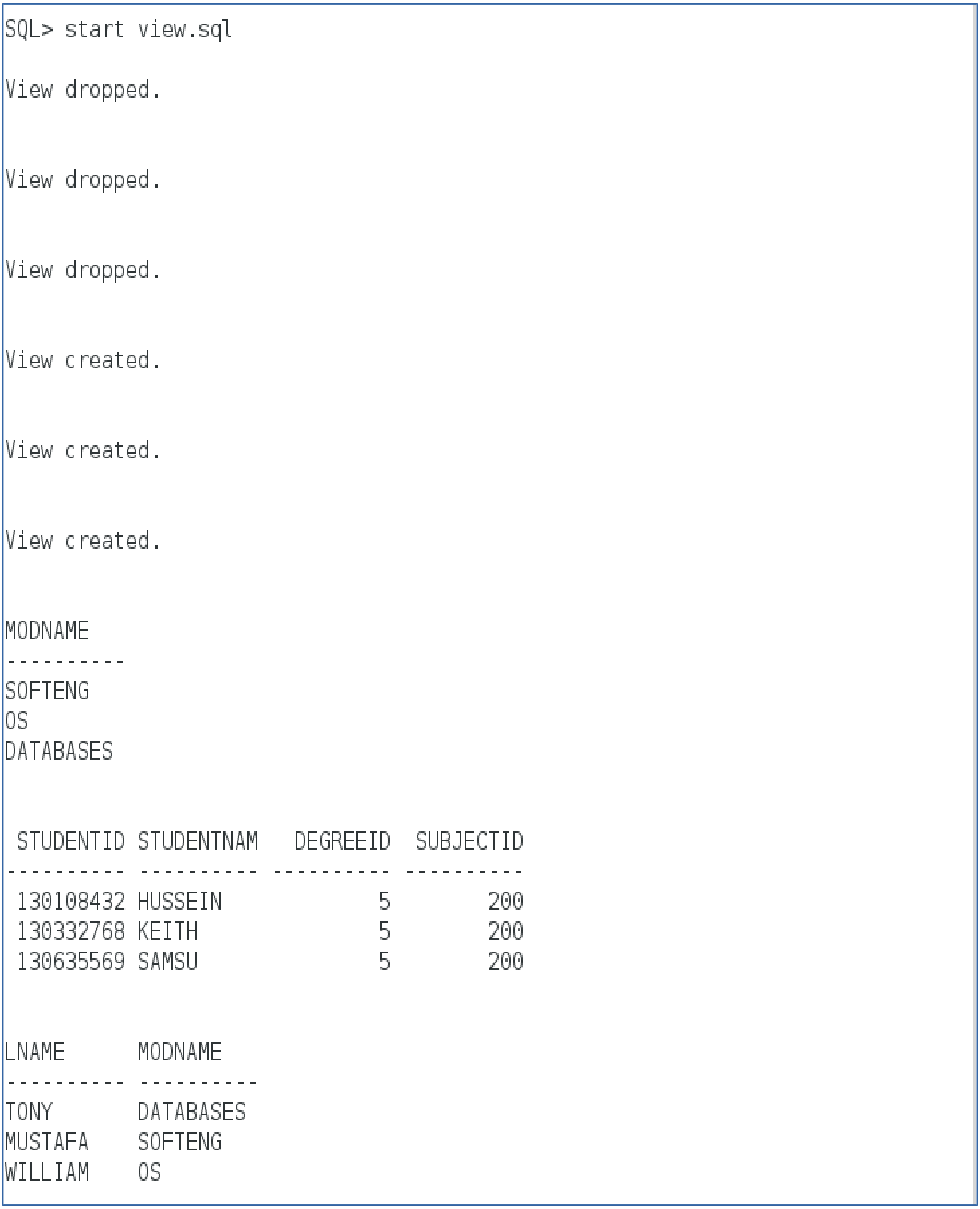
Create view Commands

- The first view was created to display a section of the module table where the module number from the SUBINMOD table has an overall mark of 100

- The second view was created to display the students with their studentID from the grade report

- The third view was created to pair information related to the lecturer with the module name, to ensure each module's lecturer was identified.

```
 DROP VIEW SOMECRAP;
  DROP VIEW CRAP;
  DROP VIEW LNAME_MODNAME;

        CREATE VIEW crap AS
       SELECT *
       FROM STUDENT
       WHERE STUDENTID IN (SELECT STUDENTID FROM GRADEREPORT)
   WITH CHECK OPTION;


        CREATE VIEW somecrap AS
        SELECT MODNAME
        FROM MODULE
        WHERE MODNUM IN (SELECT MODNUM FROM SUBINMOD WHERE
OVERALLMARK=100);


        CREATE VIEW LNAME_MODNAME (LNAME, MODNAME) AS
                        SELECT LNAME, MODNAME
                        FROM LECTURER, MODULE
                        WHERE LECTURER.MODNUM = MODULE.MODNUM;


        SELECT * FROM somecrap;

        SELECT * FROM crap;

        SELECT * FROM LNAME_MODNAME ;
```



```
SQL> start view.sql

View dropped.


View dropped.


View dropped.


View created.


View created.


View created.


MODNAME
----------
SOFTENG
OS
DATABASES


 STUDENTID STUDENTNAM   DEGREEID  SUBJECTID
---------- ---------- ---------- ----------
 130108432 HUSSEIN             5        200
 130332768 KEITH               5        200
 130635569 SAMSU               5        200


LNAME      MODNAME
---------- ----------
TONY       DATABASES
MUSTAFA    SOFTENG
WILLIAM    OS
```

SQL queries

```
    SELECT STUDENT.*,DEGREECLASS,SUBJECTNAME FROM STUDENT ,
DEGREE_CLASS,SUBJECT   WHERE STUDENT.DEGREEID=DEGREE_CLASS.DEGREEID
AND STUDENT.SUBJECTID=SUBJECT.SUBJECTID;
```

(shows students and what degree (subject) they are studying)

```
      SELECT * FROM QUESTIONS;
```

(shows all that is included in table Questions)

```
    SELECT SUBJECTNAME "SUBJECTS in UNIVERSITY" FROM SUBJECT;
```

(lists all the degree titles (subject titles) in University)

```
    SELECT SUBINMOD.SUBJECTID, STUDENT.STUDENTID, STUDENT.STUDENTNAME,
SUBINMOD.MODNUM, EXAM_DET.OUTCOME, EXAM_DET.MARKS FROM STUDENT
INNER JOIN SUBJECT ON         STUDENT.SUBJECTID = SUBJECT.SUBJECTID
    INNER JOIN SUBINMOD ON SUBINMOD.SUBJECTID = SUBJECT.SUBJECTID
    INNER JOIN MODULE ON MODULE.MODNUM = SUBINMOD.MODNUM
    INNER JOIN EXAM ON MODULE.MODNUM = EXAM.MODNUM
    INNER JOIN EXAM_DET ON EXAM.EXAMID = EXAM_DET.EXAMID
    WHERE STUDENT.STUDENTID = 130635569;
```

(lists the subject ID and student ID as well as the student names, along with the module number, exam outcome and exam marks merged with the student)

```
    SELECT COUNT(*) AS LNAME FROM LECTURER;
```

(counts the number of lecturers in the Lecturer table)

```
 STUDENTID STUDENTNAM DEGREEID   SUBJECTID DEGREECLAS
---------- ---------- ---------- ---------- ----------
SUBJECTNAME
--------------------------------------------------------------
------------------
     8 colombo          5   5 BSC
Robbing

     1 Dumbledore           5   5 BSC
Robbing

     6 Jumbalaya        5   6 BSC
Charlantry


 STUDENTID STUDENTNAMDEGREEID   SUBJECTID DEGREECLAS
---------- ---------- ---------- ---------- ----------
SUBJECTNAME
```

```
--------------------------------------------------------------
------------------
     7 Boris        5   7 BSC
Gangbanging

 130332768 KEITH        5 200 BSC
COMPSCI

 130635569 SAMSU        5 200 BSC
COMPSCI


 STUDENTID STUDENTNAMDEGREEID  SUBJECTID DEGREECLAS
---------- ---------- ---------- ---------- ----------
SUBJECTNAME
--------------------------------------------------------------
------------------
 130108432 HUSSEIN          5 200 BSC
COMPSCI


7 rows selected.


       QID   MARKS
---------- ----------
    20     40
    50    100
    10     20


SUBJECTS in UNIVERSITY
--------------------------------------------------------------
------------------
Robbing
Charlantry
Gangbanging
COMPSCI
ENGLISH
MATHS

6 rows selected.


 SUBJECTID  STUDENTID STUDENTNAM     MODNUM OUTCOME
MARKS
---------- ---------- ---------- ---------- ----------
----------
       200  130635569 SAMSU       519 50             100
       200  130635569 SAMSU       506 40              60
       200  130635569 SAMSU       518 30             100


     LNAME
```

[3]