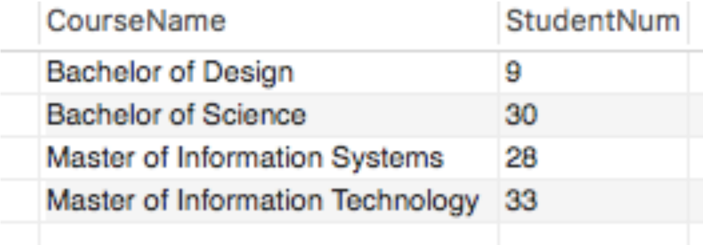
1. Count the number of students in each course at the University. Print the course name, as well as the number of students.

```
SELECT Course.name AS CourseName, COUNT(Student.id) AS
StudentNum
FROM Course JOIN Student ON Course.id = Student.course
GROUP BY Course.name;
```

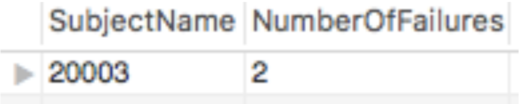


CourseName	StudentNum
Bachelor of Design	9
Bachelor of Science	30
Master of Information Systems	28
Master of Information Technology	33

4 ROWS

2. Is there any subject failed by more than one student? List the subject code as well as the number of failures.

```
SELECT CONCAT(yearlevel, code) AS SubjectName, COUNT(student)
AS NumberOfFailures FROM StudentTakesSubject
WHERE 50 > result GROUP BY yearlevel, area, code
HAVING COUNT(*) > 1;
```

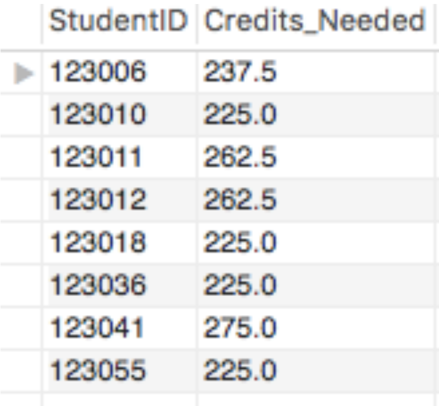


SubjectName	NumberOfFailures
20003	2

1 ROW

3. For the students who have completed at least one subject at undergraduate level, how many points does each student need to complete their degree?

```
SELECT student AS StudentID, 300 - SUM(creditpoints) AS
Credits_Needed FROM StudentTakesSubject
NATURAL JOIN Subject
WHERE result >= 50 AND student in
(SELECT Student.id FROM Student WHERE course like 'B%')
GROUP BY student;
```

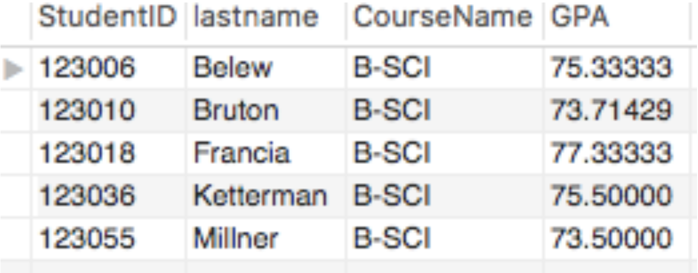


StudentID	Credits_Needed
123006	237.5
123010	225.0
123011	262.5
123012	262.5
123018	225.0
123036	225.0
123041	275.0
123055	225.0

8 ROWS

4. List the student number, lastname, course and GPA of students who have completed more than 4 subjects at undergraduate level? (To calculate *GPA you need to (1) multiply the student's result per subject by their credit points, (2) sum them up for all the subjects the student has taken and (3) divide it by the sum of the credit points these subjects are worth*)

```
SELECT student AS StudentID, lastname, course AS CourseName,
SUM(result * creditpoints) / SUM(creditpoints) AS GPA
FROM StudentTakesSubject
NATURAL JOIN Subject
JOIN Student ON Student.id = StudentTakesSubject.student
WHERE result IS NOT NULL AND course like 'B%'
GROUP BY id HAVING COUNT(result) > 4;
```

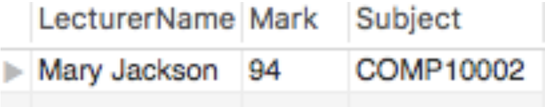


StudentID	lastname	CourseName	GPA
123006	Belew	B-SCI	75.33333
123010	Bruton	B-SCI	73.71429
123018	Francia	B-SCI	77.33333
123036	Ketterman	B-SCI	75.50000
123055	Millner	B-SCI	73.50000

5 ROWS

5. Which lecturer awarded the highest mark and what subject(s) was it (print the lecturer's full name, the mark and the entire subject code e.g. "*INFO20003*")?

```
SELECT CONCAT(Lecturer.firstname,' ', Lecturer.lastname) AS
LecturerName, result AS Mark,
CONCAT(area, yearlevel, code) AS Subject
FROM Lecturer JOIN Subject
NATURAL JOIN StudentTakesSubject
ON Lecturer.id = Subject.lecturer WHERE result IN (SELECT
MAX(result) FROM StudentTakesSubject);
```

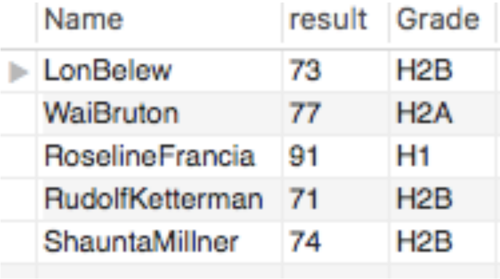


LecturerName	Mark	Subject
Mary Jackson	94	COMP10002

1 ROW

6. For each student who has completed COMP10001 print their name, result and their academic grade (*H1*,*H2A* etc).

```
SELECT CONCAT(Student.firstname, Student.lastname) AS Name,
result,
        CASE
               WHEN 50 > result THEN 'N'
      WHEN 65 > result AND result >= 50 THEN 'P'
      WHEN 70 > result AND result >= 65 THEN 'H3'
      WHEN 75 > result AND result >= 70 THEN 'H2B'
      WHEN 80 > result AND result >= 75 THEN 'H2A'
      WHEN 100 > result AND result >= 80 THEN 'H1'
        END AS Grade
      FROM Student JOIN StudentTakesSubject ON
      Student.id = StudentTakesSubject.student
               WHERE CONCAT(area, yearlevel, code) = 'COMP10001';
```



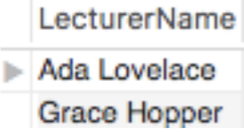
Name	result	Grade
LonBelew	73	H2B
WaiBruton	77	H2A
RoselineFrancia	91	H1
RudolfKetterman	71	H2B
ShauntaMillner	74	H2B

5 ROWS

7. Find the names of lecturers who teach at both undergraduate and postgraduate level.

```
SELECT CONCAT(firstname, ' ', lastname) AS LecturerName
FROM Lecturer WHERE Lecturer.id IN
```

```
(SELECT DISTINCT lecturer FROM Subject WHERE yearlevel = 9)
AND Lecturer.id IN
(SELECT DISTINCT lecturer FROM Subject WHERE yearlevel != 9);
```

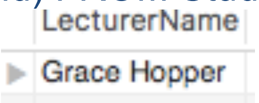


LecturerName
Ada Lovelace
Grace Hopper

2 ROWS

8. List the lecturers who teach across all study areas.

```
SELECT CONCAT(firstname, ' ', lastname) AS LecturerName
FROM Lecturer JOIN Subject ON Lecturer.id = Subject.lecturer
GROUP BY lecturer
HAVING COUNT(DISTINCT area) = (SELECT COUNT(DISTINCT
id) FROM StudyArea);
```

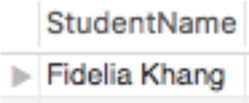


LecturerName
Grace Hopper

1 ROW

9. Have any students from Gilberton suburb enrolled into Bachelor of Science course repeated a subject at undergraduate level?

```
SELECT CONCAT(firstname, ' ', lastname) AS StudentName
FROM Student NATURAL JOIN Suburb
JOIN StudentTakesSubject ON Student.id = StudentTakesSubject.student
WHERE Suburb.name = 'Gilberton'
AND Student.id in
(SELECT student FROM StudentTakesSubject WHERE 50 > result)
GROUP BY student
HAVING COUNT(*) > 1;
```



StudentName
Fidelia Khang

1 ROW

10. The Dean has asked you to design a table that will record the student evaluations for each lecturer for each subject he has taught in each academic semester. You are to write the DDL to create the table including all suitable attributes and write the references to the Foreign Keys.

```
DROP TABLE IF EXISTS `StudentEvaluation`;

CREATE TABLE `StudentEvaluation` (
  `id` MEDIUMINT(8) NOT NULL,
  `evaluation` VARCHAR(50) NULL,
  `Lecturer_id` MEDIUMINT(8) UNSIGNED NOT NULL,
  `Subject_area` CHAR(4) NOT NULL,
  `Subject_yearlevel` TINYINT(3) UNSIGNED NOT NULL,
  `Subject_code` CHAR(4) NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_StudentEvaluation_Lecturer1_idx` (`Lecturer_id` ASC),
  INDEX `fk_StudentEvaluation_Subject1_idx` (`Subject_area` ASC,
`Subject_yearlevel` ASC, `Subject_code` ASC),
  CONSTRAINT `fk_StudentEvaluation_Lecturer1`
    FOREIGN KEY (`Lecturer_id`)
    REFERENCES `Lecturer` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_StudentEvaluation_Subject1`
    FOREIGN KEY (`Subject_area` , `Subject_yearlevel` , `Subject_code`)
    REFERENCES `Subject` (`area` , `yearlevel` , `code`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```