DATABASE SYSTEMS - ASSIGNMENT 2

UVIN ABEYSINGHE - 789931

/**********/

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 , NumberOfStudents

FROM Course INNER JOIN (SELECT course,  COUNT(Student.id) AS NumberOfStudents FROM Student GROUP BY course ) AS
TempTable ON   Course.id = TempTable.course;
```

CourseName	NumberOfStudents
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(area,  yearlevel, code) AS SubjectCode, count(DISTINCT student) AS NoOfFailures
FROM StudentTakesSubject
WHERE result<50
GROUP BY area,  yearlevel, code
HAVING COUNT(DISTINCT student)>1;
```

SubjectCode	NoOfFailures
INFO20003	2

1 ROWS

/**********/

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 PointsNeeded
FROM StudentTakesSubject NATURAL JOIN Subject
WHERE result>50 AND yearlevel!=9
GROUP BY student;
```

StudentID	PointsNeeded
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 GPATABLE.Student AS StudentID , lastname AS LastName, name AS CourseName, GPATABLE.GPA
FROM (Student INNER JOIN
(SELECT student, SUM(result*creditpoints)/SUM(creditpoints) AS GPA
FROMStudentTakesSubject NATURAL JOIN Subject
WHERE       yearlevel!=9
GROUP BY student
HAVING COUNT(*)>=4) AS GPATABLE ON Student.ID = GPATABLE.student) INNER JOIN Course ON course=Course.id;
```

StudentID	LastName	CourseName	GPA
123006	Belew	Bachelor of Science	75.33333
123010	Bruton	Bachelor of Science	64.50000
123018	Francia	Bachelor of Science	77.33333
123036	Ketterman	Bachelor of Science	75.50000
123055	Millner	Bachelor of Science	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(firstname," ",lastname) AS LecturerName,result AS Mark, CONCAT(area,yearlevel,code) AS SubjectCode
FROM (Subject NATURAL JOIN
(SELECT area,yearlevel,code,result
FROM StudentTakesSubject
WHERE result=
(SELECT MAX(result)
FROM StudentTakesSubject)) AS TempTable ) INNER JOIN Lecturer ON lecturer=id
```

LecturerName	Mark	SubjectCode
Mary Jackson	94	COMP10002

1 ROWS

/**/

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

```
SELECT CONCAT(firstname," ",lastname) AS Name, result AS Result, grade AS AcademicGrade
FROM Student INNER JOIN
(SELECT student,result,
 CASE
   WHEN  result>=80  THEN 'H1'
   WHEN result >=75 AND result<80  THEN 'H2A'
   WHEN result >=70 AND result<75  THEN 'H2B'
   WHEN result >=60 AND result<70  THEN 'H3'
```

```
    WHEN result >=50 AND result<60  THEN 'P'
    WHEN result >=49 AND result<=0  THEN 'N'
    ELSE 'NULL'
  END AS Grade
FROM StudentTakesSubject
WHERE area='COMP' AND yearlevel='1'AND code='0001')  AS ResultsGrades ON ResultsGrades.student = Student.id
```

	Name	Result	AcademicGrade
▶	Lon Belew	73	H2B
	Wai Bruton	77	H2A
	Roseline Francia	91	H1
	Rudolf Ketterman	71	H2B
	Shaunta Millner	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 INNER JOIN
(SELECT lecturer
FROM Subject
GROUP BY lecturer
HAVING (SUM(DISTINCT yearlevel)>9)) AS TempTable ON TempTable.lecturer=Lecturer.id;
```

	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 INNER JOIN
(SELECT lecturer
FROM Subject
GROUP BY lecturer
HAVING (COUNT(DISTINCT area)= (SELECT COUNT(DISTINCT area) FROM Subject)))AS TempTable ON
TempTable.lecturer=Lecturer.id;
```

	LecturerName
▶	Grace Hopper

1 ROWS

/**/

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

```
SELECT CONCAT(firstname, " ", lastname) AS Student_Name
FROM StudentTakesSubject INNER JOIN
(SELECT Student.id, firstname, lastname
FROM Student NATURAL JOIN Suburb INNER JOIN Course ON Course.id=course
WHERE Suburb.name='Gilberton'
AND  Course.name='Bachelor of Science') AS S1 ON S1.id=StudentTakesSubject.student
GROUP BY student
HAVING COUNT(DISTINCT CONCAT(area,yearlevel,code))!=COUNT( CONCAT(area,yearlevel,code)) ;
```

Student_Name
Fidelia Khang

1 ROWS

/**/

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 `StudentEvaluations`;
CREATE TABLE `StudentEvaluations` (
  `student` mediumint(8) unsigned NOT NULL,
  `area` char(4) NOT NULL,
  `yearlevel` tinyint(3) unsigned NOT NULL,
  `code` char(4) NOT NULL,
  `year` year(4) NOT NULL,
  `sem` enum('1','2') NOT NULL,
  `comment` varchar(100),
  PRIMARY KEY (`student`,`area`,`yearlevel`,`code`,`year`,`sem`),
  KEY `fk_StudentEval` (`student`, `area`, `yearlevel`, `code`, `year`,`sem`),
  CONSTRAINT `fk_StudentEval` FOREIGN KEY (`student`, `area`, `yearlevel`, `code`, `year`,`sem` )  REFERENCES
`StudentTakesSubject` (`student`, `area`, `yearlevel`, `code`, `year`,`sem` ) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Field	Type	Null	Key	Default
student	mediumint(8) unsigned	NO	PRI	NULL
area	char(4)	NO	PRI	NULL
yearlevel	tinyint(3) unsigned	NO	PRI	NULL
code	char(4)	NO	PRI	NULL
year	year(4)	NO	PRI	NULL
sem	enum('1','2')	NO	PRI	NULL
comment	varchar(100)	YES		NULL

7 ROWS

/**/