Madhupurna Dutta

102190136

**Database - Build and Query**

Question 1

1. Convert the provided ERD to a Relational Schema.
2. Follow the design as provided. Do NOT make design changes of any kind.

Submit your Relational Schema below

|  |
| --- |
| **Subject** (SubjCode, Description)  PK (SubjCode)  **Teacher** (StaffID, Surname, GivenName)  PK (StaffID)  **Student** (StudentID, Surname, GivenName, Gender)  PK (StudentID)  **SubjectOffering** (SubjCode, Year, Semester, Fee, StaffID)  PK (Year, Semester)  FK (SubjCode, StaffID)  Enrolment (StudentID, SubjCode, Year, Semester, DateEnrolled, Grade)  PK (StudentID, SubjCode, Year, Semester)  FK (StudentID, SubjCode, Year, Semester)  **Enrolment** (StudentID, SubjCode, Year, Semester, DateEnrolled, Grade)  PK (StudentID, SubjCode, Year, Semester)  FK (StudentID, SubjCode, Year, Semester) |

Question 2

Based directly on your Relational Schema from task 1and the provided Data Dictionary, write and execute the DDL to create your database.

Using an SQL Query (not the GUI) verify that all tables have been successfully created.

***Capture screenshot/s of this query and its result set & add submit below.***

Commit your work in your Git repo with the commit message “Task 2 Complete” & push it to origin.

|  |
| --- |
| RESULT: |

Question 3

Write and execute the DML to add the test data provided to your database

Additional Data:You MUSTalso add yourself as a student. Use your name &student id, invent other data.

In a query editor on the cloud service your database is deployed on & run the query:

*Select \* from student*

***Submit a screenshot of the result set from the above query below:***

|  |
| --- |
|  |

Question 4

Write a query that shows the student first name and surname, the subject code and description, the subject offering year, semester & fee and the given name and surname of the teacher for that subject offering.

***Screenshot the query and result set and submit below***

|  |
| --- |
|  |

Question 5

Write a query which shows the number of enrolments, for each year and semester in the following example format. For example:

|  |  |  |
| --- | --- | --- |
| Year | Semester | Num Enrollments |
| 2018 | 2 | 1 |
| 2019 | 1 | 7 |
| 2019 | 2 | 4 |
| 2020 | 1 | 5 |

*(The actual results will vary. This demonstrates format only)*

***Screenshot the query and result set and submit below:***

|  |
| --- |
|  |

Question 6

Query 3:

Write a query which lists all enrolments which for the subject offering which has the highest fee. (This query must use a sub-query.)

***Screenshot the query and result set and submit below:***

|  |
| --- |
|  |

Commit your work in your Git repo with the commit message “Queries Complete” & push it to origin.

Question 7

Create a View based on Question/Task 3

***Submit the SQL below***

|  |
| --- |
| DROP VIEW if exists SubqueryTask3;  CREATE VIEW SubqueryTask3 AS  (SELECT  \*  FROM  Enrolment  WHERE  (SubjCode , Year, Semester) IN (SELECT  SubjCode, Year, Semester  FROM  SubjectOffering  WHERE  EventFee = (SELECT  MAX(EventFee)  FROM  SubjectOffering))); |

Commit your work in your Git repo with the commit message “View Complete”& push it to origin.

Question 8

Write queries to prove your responses to questions/tasks 3 - 6 are returning the correct/sensibleresults.

E.g. to test that select \* from studentis returning the correct number of rows you could use select count(\*) from student and check that the number in the count query is the same as the number of rows returned by the select \* query.

***Provide a (short) written explanation of how each of your ‘test’ queries verifies the original query:***

|  |
| --- |
|  |