System Description

In the SHAMON University system we want to record information about a student’s enrolments. We want to know the id, name, address and phone-no of the student. We want to know what subjects they are enrolled in during a semester, and be able to record their result for their enrolments. We also want to know the code and name for a subject, and for each semester that it is offered we would like to know the location details, and the id, name and room no. of the lecturer taking the subject offering. An offering of a subject is only ever taken by one lecturer.

You have done the following as part of Assignment 1 – Functional Specification

**ENTITY RELATIONSHIP MODEL**

STUDENT

<table>
<thead>
<tr>
<th>STUDENT-ID</th>
<th>STUDENT-NAME</th>
<th>STUDENT-ADDRESS</th>
<th>STUDENT-PHONE-NO</th>
<th>SUBJECT-CODE</th>
<th>SUBJECT-NAME</th>
<th>SEMESTER</th>
<th>YEAR</th>
<th>RESULT</th>
</tr>
</thead>
</table>

SUBJECT-OFFERING

<table>
<thead>
<tr>
<th>SUBJECT-CODE</th>
<th>SUBJECT-NAME</th>
<th>SEMESTER</th>
<th>YEAR</th>
<th>SUBJECT-LOCATION</th>
<th>LECTURER-ID</th>
<th>LECTURER-NAME</th>
<th>STUDENT-ID</th>
<th>STUDENT-NAME</th>
<th>RESULT</th>
</tr>
</thead>
</table>

LECTURER

<table>
<thead>
<tr>
<th>LECTURER-ID</th>
<th>LECTURER-NAME</th>
<th>LECTURER-ROOM-NO</th>
<th>SUBJECT-CODE</th>
<th>SUBJECT-NAME</th>
<th>SEMESTER</th>
<th>YEAR</th>
<th>SUBJECT-LOCATION</th>
</tr>
</thead>
</table>

**DATA GATHERING FOR ENTITIES – UNNORMALISED RELATIONS**

*Using the system description and/or any process models (DFDs) which may exist for the system gather the attributes for the entities.*

For example, for the Student Entity you would look at what you needed to store about student to meet the specified business requirements, and you would also look to see what entities the Student entity had a relationship with in the ER model, to help you gather the appropriate attributes. In this case you are interested in the subject offering details for student as well as the student details.

STUDENT (Student-id, Student-name, Student-address, Student-phone-no, Subject-code, Subject-name, Semester, Year, Result)

SUBJECT-OFFERING (Subject-code, Subject-name, Semester, Year, Subject-location, Lecturer-id, Lecturer-name, Student-id, Student-name, Result)

LECTURER (Lecturer-id, Lecturer-name, Lecturer-room-no, Subject-code, Subject-name, Semester, Year, Subject-location)
DATA DICTIONARY – ENTITIES

STUDENT =
Student-id +
Student-name +
Student-address +
(Student-phone-no) +
({Subject-code +
Subject-name +
Semester +
Year +
(Result) })

() shows that the student phone no. is optional.

() shows that students details can exist without
any subject enrolments data, and the {} shows
that a student can enrol in many subject offerings.

Within subject details the result field is optional
as the student may not have received a result for the
subject as yet.

SUBJECT-OFFERING =
Subject-code +
Subject-name +
Semester +
Year +
(Subject-location )+
(Lecturer-id +
Lecturer-name )+
({Student-id +
Student-name +
(Result)})

LECTURER =
Lecturer-id +
Lecturer-name +
Lecturer-room-no +
({Subject-code +
Subject-name +
Semester +
Year +
(Subject-room-no )})
NORMALISATION PROCESS

For Assignment 2 your Normalisation needs to follow the format shown below:

Please note that any detail in italics is provided for information and should not be handed in as part of the assignment.

Unnormalised Relation – STUDENT

STUDENT (Student-id, Student-name, Student-address, Student-phone-no, Subject-code, Subject-name, Semester, Year, Result)

Identify primary key and repeating groups.

STUDENT (Student-id, Student-name, Student-address, Student-phone-no, (Subject-code, Subject-name, Semester, Year, Result))

1st Normal Form Relations

STUDENT (Student-id, Student-name, Student-address, Student-phone-no)

STUDENT-SUBJECT-OFFERING (Student-id, Subject-code, Semester, Year, Subject-name, Result)

2nd Normal Form Relations

STUDENT (Student-id, Student-name, Student-address, Student-phone-no)

STUDENT-SUBJECT-OFFERING (Student-id, Subject-code, Semester, Year, Result)

SUBJECT (Subject-code, Subject-name)

3rd Normal Form Relations

Relations in 2NF = 3NF
Unnormalised Relation – SUBJECT-OFFERING

SUBJECT-OFFERING (Subject-code, Subject-name, Semester, Year, Subject-location, Lecturer-id, Lecturer-name, Student-id, Student-name, Result)

Identify primary key and repeating groups.

SUBJECT-OFFERING (Subject-code, Semester, Year, Subject-name, Subject-location, Lecturer-id, Lecturer-name, (Student-id, Student-name, Result))

1st Normal Form Relations

SUBJECT-OFFERING

(Subject-code, Semester, Year, Subject-name, Subject-location, Lecturer-id, Lecturer-name)

SUBJECT-OFFERING-STUDENT (Subject-code, Semester, Year, Student-id, Student-name, Result)

2nd Normal Form Relations

SUBJECT-OFFERING (Subject-code, Semester, Year, Subject-location, Lecturer-id, Lecturer-name)

SUBJECT (Subject-code, Subject-name)

SUBJECT-OFFERING-STUDENT (Subject-code, Semester, Year, Student-id, Result)

STUDENT (Student-id, Student-name)

3rd Normal Form Relations

SUBJECT-OFFERING (Subject-code, Semester, Year, Subject-location, Lecturer-id)

LECTURER (Lecturer-id, Lecturer-name)

SUBJECT (Subject-code, Subject-name)

SUBJECT-OFFERING-STUDENT (Subject-code, Semester, Year, Student-id, Result)

STUDENT (Student-id, Student-name)
Unnormalised Relation – LECTURER

LECTURER (Lecturer-id, Lecturer-name, Lecturer-room-no, Subject-code, Subject-name, Semester, Year, Subject-location)

Identify primary key and repeating groups.

LECTURER (Lecturer-id, Lecturer-name, Lecturer-room-no, (Subject-code, Subject-name, Semester, Year, Subject-location))

1st Normal Form Relations

LECTURER (Lecturer-id, Lecturer-name, Lecturer-room-no)

LECTURER-SUBJECT-OFFERING
(Lecturer-id, Subject-code, Semester, Year, Subject-name, Subject-location)

Note: These arrows show the following partial dependencies:
For each Subject code there is only ever one subject name.
For each ‘Subject code, Semester, Year’ there is only ever one Subject-location.
For each ‘Subject code, Semester, Year’ there is only ever one Lecturer-id (this is a special case of a partial dependency within the key)

2nd Normal Form Relations

LECTURER (Lecturer-id, Lecturer-name, Lecturer-room-no)

SUBJECT-OFFERING (Subject-code, Semester, Year, Subject-location, Lecturer-id)

SUBJECT (Subject-code, Subject-name)

3rd Normal Form Relations

Relations in 2NF = 3NF
Review all the 3NF relations to remove any duplicates (relations with the same key and same attributes) and to join any relations with the same key but different attributes. If the name of the relations with the same key should differ, choose one name that most suits the joint relation.

1. **STUDENT** (Student-id, Student-name, Student-address, Student-phone-no)

2. **STUDENT-SUBJECT-OFFERING** (Student-id, Subject-code, Semester, Year, Result)

3. **SUBJECT** (Subject-code, Subject-name)

4. **SUBJECT-OFFERING** (Subject-code, Semester, Year, Subject-location, Lecturer-id)

5. **LECTURER** (Lecturer-id, Lecturer-name)
   Same key and data as 9, so should be deleted.

6. **SUBJECT** (Subject-code, Subject-name)
   Same key and data as 3, so should be deleted

7. **SUBJECT-OFFERING-STUDENT** (Subject-code, Semester, Year, Student-id, Result)
   Same key and data as 2, even though the order of the key attributes, and the relation names are different, so should be deleted.

8. **STUDENT** (Student-id, Student-name)
   Same key and data as 1, so should be deleted.

9. **LECTURER** (Lecturer-id, Lecturer-name, Lecturer-room-no)

10. **SUBJECT-OFFERING** (Subject-code, Semester, Year, Subject-location, Lecturer-id)
    Same key and data as 5, so should be deleted.

11. **SUBJECT** (Subject-code, Subject-name)
    Same key and data as 3, so should be deleted
FINAL LIST OF 3NF RELATIONS (after review)

STUDENT (Student-id, Student-name, Student-address, Student-phone-no)

STUDENT-SUBJECT-OFFERING (Student-id, Subject-code, Semester, Year, Result)

SUBJECT (Subject-code, Subject-name)

SUBJECT-OFFERING (Subject-code, Semester, Year, Subject-location, Lecturer-id)

LECTURER (Lecturer-id, Lecturer-name, Lecturer-room-no)

Use this final list to construct your Data Structure Diagram after identifying the foreign keys. The Data Structure Diagram should not have any many-to-many relationships.

DATA STRUCTURE DIAGRAM