31271 Database Fundamentals

Database Design Assignment – Spring 2023

Assignment Title: Designing a Database for an online delivery-based dietary company (e.g., Jenny Craig)

*NOTE: Please **DO NOT USE** "hello fresh"*

Assignment Due date: Refer to the subject outline or Ed Page for each part.

Table of Contents

I. Important Note

Vivas[1]* or other invigilated tasks may be used to verify student achievement of learning outcomes to ensure they have completed the work on their own and to assess their knowledge of the answers they have submitted.

Several random groups will be selected to have a face-to-face presentation meeting with their marker. The marker will ask them some questions regarding their work. Every member of the team is supposed to be aware of the content of the submitted assignment and needs to be able to answer the questions.

Students need to provide ethical referencing if they use any external sources, including ChatGPT.

[1] * Viva voce (derived from Medieval Latin) is defined as "an examination conducted by speech or assessment in which a student's response to the assessment task is verbal, in the sense of being expressed or conveyed by speech instead of writing" (Pearce & Lee 2009).

II. Introduction

The assignment involves the conceptual and logical design of a database to support customers signing up and ordering meals from an online delivery-based dietary company (e.g., Jenny Craig). The database may have stored data related to customers, plans, services, bills, receipts, etc.

Some case study examples can be:

- Marley Spoon
- Factore75
- etc,

WARNINGs:

- **Please be aware that you are not allowed to use Hello Fresh as it was used by some students in the previous semester as a case for another related topic.**
- **If the assignment does not match the topic, you will receive zero for the entire assignment.**
- **You need to register into a group on Canvas to be able to see assignment submission link on Canvas.**

The assignment will be marked to **45 marks in total** and is weighted **30%** of your final subject grade.

NOTE: Please be aware that we may remove Part C and replace it with a quiz. In addition, the normalization test will not be a bonus mark and will be part of your assignment mark. Therefore, the total mark 45 may be changed. However, your final assignment mark will be 30% of your final mark.

This is an individual or group assignment. You may submit it as an individual, in pairs, or in groups of a maximum of 3 students. The assignment will be marked according to the same criteria regardless of whether an individual or a group does work.

The data model can be based on an existing system or a fictional system, but in the latter case, we suggest that you base it primarily on a real system to ensure that you include all the necessary information. In your submission, include details about which website you used as a source for your ideas. The data model details should be as complex as your group can reasonably manage and model correctly. Generally, individuals will tend to have smaller, less complex data models than groups, which should be dealing with more entities and relationships.

Please ensure that you adhere to the generally accepted principles of privacy and confidentiality of data. In particular, if you are basing your assignment on an actual system to which you have access, do not use details such as names, contact details etc. of real customers or other people in your case study. You should change this data so that it cannot be linked to a real person or company.

Note: Please use the ***Assignment Template*** *that is available on Ed for your submissions (****except Part D*** *which needs to be submitted as a .txt or .sql file).*

III. Assignment Requirements

The submission includes the following deliverables:

➢ Part A - Database design case study

Give a short description of the application/website for which you will be designing the database (The overview of your own/chosen case study that is related to the assignment topic), outline the basic functions that your database needs to support, identify the data requirements, and business rules. Please include some sample documentation (forms, statements, webpage etc.).

- **The Structure of Part A Report:**

  Please see the marking criteria of Part A provided in Appendix A. You need to have four sections in your assignment each related to one of the first four criteria.

  Please have a **HEADING** for each four sections. The headings are: Overview, Business Functions, Data requirements and Business rules.

  A.1. Your case study **overview** should be similar to the New Oriental Hospital case study given for your data modelling tutorials (See Tutorials 2 and 3), but will probably be shorter, with 3 or 4 forms. Provide the overview in 2-3 pages (including the forms of your case study. The forms can be screen shots of the website). You **will** only **need 6-12 entities** in total when doing the conceptual data modelling in **Part B**, so bear this in mind when deciding the scope of your case study.

  A.2. The **Functions supported by the database** (business functions) are the services that the system (i.e. the targeted company's website) provides. These services should be related to the main goal of the system; however, their function altogether fulfils the final goal of the system. For example, to hire a car, a customer should "register", "make a quote", "pay", "collect", and "return" the vehicle. In regard to each function, data needs to be exchanged (passed) between the front-end user, and the system.

  Note: *you may have more than one entity related to each business function (see the related slide in lecture one).*

  A.3. The **data requirements** are the data that need to be collected, stored and provided. They are usually the attributes of your entities.

  A.4. The last part of your report should be a list of at least 10 **Business Rules**.

  Note: *Please see the "Assignment Template" file uploaded on* ***Ed****.*

- **Please** See Section IV.4 (Instructions) before submission

➢ Part B - Conceptual data modelling

Produce the final ERD reflecting the data requirements given in the case study (using the **textbook/Lecture note notation**). **Justify your design decisions by documenting** all your assumptions for your conceptual data model, **provide a list of PK, and FK(s) for each entity**. Your conceptual data model will be marked according to your ability to identify appropriate entities, identify attributes, attributes, relationships and cardinalities, and to structure a data model using these elements.

- **The Structure of Part B Report:**

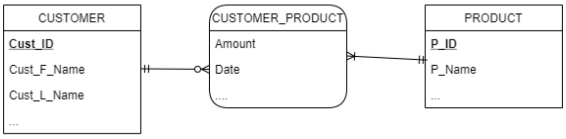
B. 1. Overview of the case study in at most five sentences (From part A)

B. 2. Revised Business rules and assumptions of part A

B. 3. ERD (Conceptual data modelling)

B. 4. Justifications of the ERD based on the business rules and assumptions.

You need to provide correct, detailed business rules describing entities, cardinality and relationships between entities (Justification of design). You may need to repeat a business rule several times. **For example,** you need to justify each **entity**, **relationship**, and **cardinalities** in the designed ERD below, and **list the PK and FK(s)** as follows:



- Business rules related to entity CUSTOMER:

BR1: Every customer needs to register to the system.

BR2: Every customer should provide his or her name, address and mobile number.

The PK of CUSTOMER is: Cust_ID

- Business rules related to the relationship between CUSTOMER and PRODUCT and its cardinalities:

BR4: Every customer can buy many products (cardinality 0 to many).

BR7: Every product needs to be purchased by at least one customer (cardinality 1 to many).

Note: As you can see, based on BR4 and BR7, there is a **many-to-many relationship** between CUSTOMER and PRODUTC that is converted to the associative entity (CUSTOMER_ PRODUTC).

The PK of the CUSTOMER_ PRODUTC is: Cust_ID, P_ID, Date

The FKs of the CUSTOMER_ PRODUTC are:

- Cust_ID from CUSTOMER
- P_ID from PRODUTC

- Business rules related to entity PRODUTC:

BR11: Each product has a unique ID and a name(description).

The PK of the PRODUTC is: P_ID

Note: *Please see the "Assignment Template" file uploaded on* ***Ed****.*

- **Please** See Section IV.4 (Instructions) before submission

➢ Part C - Logical design: Schema conversion and Normalisation

Please be aware that we will replace this part of the assignment with a quiz plus the normalization test.
I will make the quiz available soon.
The normalization test is already available for you to complete.

- ~~**The Structure of Part C Report:**~~

~~**C.1. Revised Business rules and assumptions (From part B)**~~

~~**C.2. Revised ERD of Part B**~~

~~**C.3. Relations** (Logical design / Schema conversion):~~

~~Convert your ERD (conceptual schema) into a set of relational tables. Use the following format to list each table, where the primary keys are underlined, and the foreign keys are marked with asterisk(s):~~

~~`TableName (Identifier, non-key attributes)`~~

~~OR (for those relations that have FKs):~~

~~`TableName (Identifier, non-key attributes, ForeignKey*)`~~
~~`ForeignKey references OtherTable`~~

~~Your relational model will be marked according to your correct application of the appropriate conversion rules.~~

~~**C.4. List of functional dependencies related to each business rules.**~~

~~**C.5. Normalisation (Logical design):**~~

~~a) Identify and list the functional dependencies (FDs), and specify each FD is determined based on which business rules and/or forms of your case study.~~
~~Please use the following format for your functional dependencies:~~
~~X → Y~~
~~i.e. Determinant Attribute(s) → Dependent Attribute(s)~~

~~b) Use these FDs to determine the highest normal form for each table/relation defined in deliverable 3. Justify your decisions. If there are tables which are not fully normalised (i.e., not in 1NF, 2NF or 3NF), perform normalisation for the tables until all the tables are in 3NF. You must document in detail each part of the normalisation process.~~
~~Please use the following format for your **final set of relations**, where the primary keys are underlined, and the foreign keys are marked with asterisk(s):~~

~~`RelationName (Identifier, non-key attributes)`~~

~~OR (for those relations that have FKs):~~

~~`RelationName(Identifier, non-key attributes, ForeignKey*)`~~
~~`FK (`*`ForeignKey`*`) References `*`OtherRelation`*~~

~~***Note:*** *Please see the "Assignment Template" file uploaded on* ***Ed****.*~~

- ~~**Important notes for Part C:**~~
  - ~~*Your assignments will be marked according to your demonstration of knowledge, i.e., you must demonstrate that you understand each of the processes of database design, such as conceptual data modelling, logical design of a database and normalisation. You need to justify your design decisions, so do not simply show your final answers. Show your working and reasoning as much as possible.*~~
  - ~~*Remember that we are not determining FDs based on relations. If you extract FDs out of your ERD we will mark you down. The FDs created based on relations **worth 0 marks**.*~~
  - ~~*Any FD should be justified by at least by one BR.*~~
  - ~~*To do normalization, please follow "**Normalization Step-by-Step**" document (uploaded in assignment module on Ed). This document is a good sample for the normalization section of Part C.*~~
  - ~~*For the normalization part: If any of your relations are already in 3NF, that's totally fine. However, you need to write about how each relation/table is already in 3NF. To do this, please follow the structure that is provided in the last pages of "Normalization step-by-step" document.*~~
- ~~**Please** See Section IV.4 (Instructions) before submission~~

➢ Part D - Implementation

In Part D of the assignment, you need to create a corresponding database to your designed ERD using PostgreSQL.

Your database must satisfy the "minimal essential" conditions given below, otherwise, you will be penalized in the marking.

Please remember to click on the **mark** button; then your code will be submitted.

1) The Database

The database must be based on your previously created ERD that you designed and submitted for Assignment B (Or normalized ERD submitted as Part C) of this semester with Database Fundamentals.

Choose **THREE** tables from your ERD to implement. The implemented tables **should have RELATIONSHIPS (WITH EACH OTHER)**. These three entities should be **an associative entity accompanied with two neighbour entities**.

Please do NOT use Surrogate key.

2) The Data

Populate your database with suitable data for testing the SQL queries below (Section 3). Imaginary data for that web site is acceptable, but real data from that web site is preferred, where possible. You **must** provide the URL for the web site that inspired your project.

Each table should have at least **5** rows. Also, provide enough data so that the table rows demonstrate the relationships. For example, if there is a 1:M relationship between 2 tables, ensure that there are at least two records in the M-side table that are related to a respective record in the 1-side table.

3) Queries

Write the following queries for your database:

- Three SELECT * statements for three separate tables (see section 4.5).
- A query involving a “Group by”, perhaps also with a “HAVING” (see section 4.6)
- A query which uses "inner join" (see section 4.6).
- A query which uses a “sub query” (see section 4.6).

4) The Database Scripts

Build up your database scripts based on the following instruction.

4.1. Provide the Required information

The file containing your SQL should begin with a **comment header block** (i.e. lines beginning with two dashes, or using /* <comment goes here> */).

The first line of the header block should contain (“Database Fundamentals, Assignment Part D”), followed by lines providing your name and email address (but **not** your student number). The header block should then contain, in English, the nature of your database application. Do NOT use technical database language in this section. Write something simple. You **must** provide the URL for the web site that inspired your project in this hearer block.

Example: See the first lines in the provided example file “dbpizza_Revised.txt”

4.2. Drop Each Table

Start the SQL Scripts with “**DROP**” commands for each of your tables, so you can run your script more than once.

Example: See the “DROP TABLE IF EXISTS” statements in the provided example file “dbpizza_Revised.txt”

4.3. Create Each Table

Use **CREATE** statements to create your tables. Use “**Constraint**” to define primary and foreign keys.

Example: See the “create table” statements in the provided example file “dbpizza_Revised.txt”

4.4. Insert Data in Each Table

Use **INSERT** statements to insert data into your tables as required (see section 2).

Example: See the “insert” statements in the provided example file “dbpizza_Revised.txt”

4.5. Show the Inserted Data using Select * Statement

We need to check that the data that are inserted into your tables. To do this, you need to include “select * from TableName” statement in your database script with an English language description of what the SQL query does. You need to comment out the English language description of the query **with** '-- ' so you do not get a syntax error when running your script!

Note: *Please write the “select * from TableName” statement* ***in one line****.*

All designed SELECT statements, and the corresponding SELECT statements should be appended to your script right after the **last** INSERT statement of your script.

Example: Follow the structure provided in 2.b.1, 2.b.2 and 2.b.3 examples provided in the example file “dbpizza_Revised.txt”. You need to specify the question, and the select statement.

4.6. Provide the Queries Using “Group by” “Inner Join” and “Sub Query”

An English language description of what the SQL query does. Place each question **right before** the corresponding SELECT statement. Comment out the questions (with '-- ' as it is described in Section 4.5) so you do not get a syntax error when running your script!

Example: Follow the structure provided in 3.a, 3.b, and 3.c examples provided in the example file “dbpizza_Revised.txt”. You need to specify the question, and the select statement.

5) The SQL .txt File: Minimum Essential Conditions

Your database must satisfy the following “minimal essential” conditions, otherwise you will be penalized in the marking, and perhaps receive zero marks.

1. Implement your database design in PostgreSQL.

1. We have created the "Part D (Working/Test Environment)" under the Part D module.

   Please use this environment to create your work and test that your DDL compiles correctly.

2. Submit the work on **two** locations:

   - **Every member of the team** needs to submit the work on **Ed**: You need to submit your code on "Part D Submission" located in “Part D” lesson page. You also need to upload your ERD on Ed.

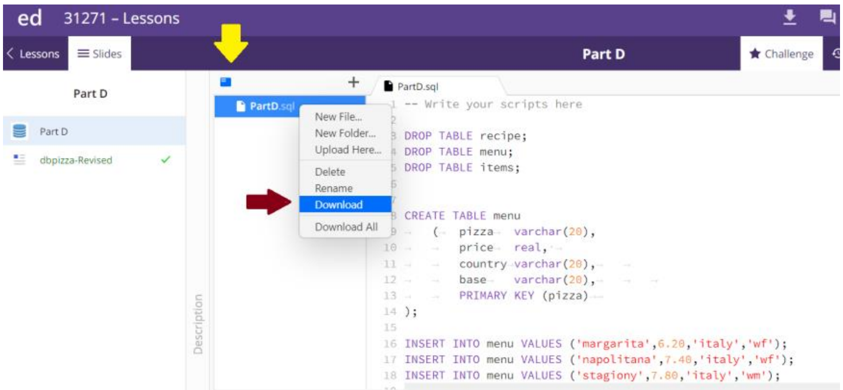
     The submission guideline and the submission environment (Part D Submission page) are available there. Please be aware that your submitted version on Ed will be marked. You need to click on the **MARK** button in the Part D Submission page**, then** your code will be submitted and marked.

   - **Just one member of the team** need to submit the dbName.sql on **Canvas** (as explained in the assignment specification) to keep a copy of your code on Canvas as well.

3. Your database scripts in the .txt (or .sql) file must successfully build a database in PostgreSQL. A submission that produces an error when run within Ed environment will

receive a **zero** mark for Part D.

4. Your SQL should be laid out so it is easy to read. When writing your SQL code, you should approximately follow the indentation style used in the file provided (the dbpizza_Revised.txt file).
5. Please do not change the name of PartD.sql in the Part D assessment. Your written codes in PartD.sql file will be compiled and marked.
6. All your **CREATE statements should precede** all your INSERT statements.
7. When submitting the assignment, provide all your scripts in ONLY AND ONLY one file, which is your designed script.
8. You should download the dbName.sql file as follows:



```
-- Write your scripts here

DROP TABLE recipe;
DROP TABLE menu;
DROP TABLE items;


CREATE TABLE menu
        (   pizza   varchar(20),
            price   real,
            country varchar(20),
            base    varchar(20),
            PRIMARY KEY (pizza)
);

INSERT INTO menu VALUES ('margarita',6.20,'italy','wf');
INSERT INTO menu VALUES ('napolitana',7.40,'italy','wf');
INSERT INTO menu VALUES ('stagiony',7.80,'italy','wm');
```

9. Submit your file in **txt or sql format**. DO **NOT** submit a Microsoft word document. The only accepted file extension is **txt**, or **sql**.
10. Your database **must not** resemble closely any database you found in the textbook, or any other book, and it must not resemble closely any database built by a student in a previous semester of this subject or any other subjects.
11. Students are warned that a token effort (i.e. seriously breaches the “minimal essential” conditions) of assignment will attract zero marks.
12. Students are free to produce a database that goes well beyond the requirements of the minimum essential conditions, if they wish to do so for their own satisfaction. However, students are warned that databases that go well beyond the minimum essential conditions do not attract extra marks.
13. **The database file must be runnable with no errors.**

6) Sample File

To get an idea on how to give structure to your database script file, have a look at the revised version of Pizza database script (the dbpizza_Revised.txt file), available for download one the same page as you found the current document on Ed. This file is also provided in pdf format to use if the structure of the text file is changed in different OS.

7) Files to Submit:

- The SQL .txt or .sql File

- See also Section IV.4 (Instructions)

IV. Marking Scheme

In marking your submissions, the total mark of 45 is split among the deliverables as follows:

Assignment Part	Deliverable	Points	Marks out of 30%
Part A	Case Study	5	3.3
Part B	Data Modelling	15	10
Part C	Schema Conversion and Normalization	20	13.3
Part D	Database	5	3.4

Note: The normalization test bonus marks (5 marks) will be added to this mark. Total assignment mark will be at most 30.

V. Instructions

1. **Each group should have a code** which is your group code on Canvas.
2. **Group Code and Assignment Marks:** You can find your "group code" and "Assignment marks" on Canvas.
3. You are expected to submit a professional presentation in a **soft copy format**, prepared using a suitable word processor and ER diagramming tool (for Part B). A hand-written submission is not acceptable and will not be marked.
4. The following jobs need to be completed for each part of the assignment by **11.59 pm** of the due date:
   - ✓ **For each assignment part**, **one of the team members** should submit the assignment soft copy on **Canvas once,** and **Part D** should be submitted by **all members** on **Ed as well**.
   - ✓ **The marking criteria for the Part due need to be checked.**
     For each deliverable, you must double check the related marking criteria to make sure that you have provided the required information in your work. Marking criteria, which the teaching team will also use when marking student assignments, are in Appendix A of this document.
     - **Note*:*** We may change the marking Criteria. If the subject coordinator makes any changes on the marking criteria, they will update you prior to the assignment due.
5. For pairs and groups, students **can** assess the contribution of their peers to the assignment. To do this we will using **SparkPlus.**

a. This peer assessment will be used to determine whether all group members receive the same mark for the assignment. If discrepancies in contribution are noted, individual marks will be scaled in accordance with the peer feedback.

b. If you are having trouble with the operation of your group, ask the Subject Coordinator for advice (preferably ask as a group) well before the next due date. If part of the group feels that other member(s) are not contributing, the coordinator should be informed, and a group meeting held to produce a solution. It is your responsibility to manage your groups, your group work and to ensure that any problems are identified and dealt with promptly.

- **Note:** The subject coordinate will inform you if it is required to complete SparkPlus. Otherwise just send an email to her **IF** you had any discrepancies in your group.

6. There may be errors and ambiguities in the assignment specification. If so, corrections/clarifications will be posted on Canvas or Ed. You are expected to incorporate these changes into your submission.
7. A thread for the assignment will be on the discussion page on Ed. Any general questions about the specification and answers from the subject co-ordinator will be put into this thread. You can only post questions related to the specifications (not the solutions). Please check this thread regularly for updates, or subscribe to it so that you are notified of new posts.
8. The assignment specification (with corrections and/or clarifications) will be frozen one week prior to the submission dates. Do not submit the assignment more than one week earlier than the submission dates, and make sure you have checked the **Assessment module** and discussion forum before you submit your assignment.
9. Please take note of the due dates, and work to those dates. **Late submissions will incur heavy penalties**. See the subject Outline for more information.
10. **Special consideration**, for late submission due to illness or misadventure, of up to one week must be arranged **before the due date** with the subject co-ordinator, unless extenuating circumstances make this impossible, in which case the student must contact the subject coordinator as soon as possible. More than one week's extension must be arranged via a formal, online Special Consideration application, **by the due date**.

    Please see http://www.gsu.uts.edu.au/rules/student/section-8.html#r8.3 for university policy on Special Considerations.
11. Please refer to "Academic integrity, plagiarism, and cheating" section in the faculty student guide with regard to academic standards:

    http://www.uts.edu.au/sites/default/files/FEIT%20Student%20Guide.pdf

    Participants are also reminded of the principles laid down in the "Statement of Good Practice and Ethics in Informal Assessment" in the Faculty Handbook.

Appendix A: Marking Criteria

Each student/pair/group is required to do self-assessment using marking sheets of each assignment part. Students need to assess their own work according to the given criteria, and make sure that they have provided the required information, before they submit their work for assessment.

Making the marking criteria available to students before they attempt the assignment clarifies what knowledge and skills they are expected to master through constructing a solution for the assignment.

Furthermore, part of an effective learning experience is learning how to make accurate judgments about the quality of one’s own work, an essential skill for both students and professionals. Self-assessment enables students to judge whether or not they have achieved each of the outcomes (criteria) throughout the construction of their assignment solution. This encourages reflection, with revision where necessary, and an active learning experience.

The teaching team will use exactly the same marking sheets and criteria as the students when they mark the assignments. The marking criteria for each assignment part are given on the following pages.

Note 1: The Self-assessment DOES NOT need to be submitted as part of the assignment solution as it will raises the similarity rate of your work.

Note 2: Please be aware that the marking criteria for each part of the assignment may be changed. In this case, you will be notified about this one week prior to your submission due.

Part A Marking Criteria

Part A: Case Study (5 marks)

1. Overview of the case study with the inclusion of forms	1	
2. Outline functions that the database supports	1	
3. Description of data requirements	1	
4. Description of at least 10 business rules	2	
	Total	

Part B Marking Criteria

Part B: ERD (15 marks)

Revised Business rules	**1**	
1. Correct list of business rules.	1	
Overall ERD: Identification of appropriate data elements and relationship between them (Note: Necessary entities and relationships should be included, and there should not be any extra).	**5.5**	
2. Consistent notation for relationships and entities (lecture note notation) for regular, weak, and associative entities, and subtype/supertype(s).	0.5	
3. Identification of appropriate entities (at least 6 entities where subtypes and super-type are considered as one entity).	0.5	
4. Identification of appropriate relationships between entities.	0.5	
5. No foreign keys shown on the diagram.	0.5	
6. Provide at least one correct optional cardinality and refer to the related business rule.	0.25	
7. Provide at least one correct mandatory cardinality and refer to the related business rule.	0.25	
8. Provide at least one correct use of one-to-many relationship (not part of associative entity's relationship).	0.25	
9. Correct relevant attributes included in all entities.	1	
10. Correct use of self-referencing relationship.	1	
Regular entity criteria (at least 3 regular entities).	**2**	
11. Correct identifier attribute(s) in all regular entities **Note:** PK and FK(s) should be listed, and PK in the entity should be underlined.	0.75	
12. Correct detailed business rules describing entities, cardinality, and relationships between entities (Justification of design for each regular entity).	1.5	
Associative entity criteria (M:N relationship that are converted to associative entity)	**3**	
13. Correct identifier, and relevant attributers to the relationship (if needed) for associative entity. **Note:** PK and FK(s) should be listed, and PK in the entity should be underlined.	0.75	
14. Correct conversion of M:N relationship into associative entity.	1	
15. Correct detailed business rules describing entities, cardinality, and relationships between entities (Justification of design).	1.5	
Weak entity criteria	**2**	
16. Correct identifier attributes for the weak entity. **Note:** PK and FK(s) should be listed, and PK in the entity should be underlined.	0.75	
17. Correct use of identifying relationship to strong entity (choice and notation).	0.5	
18. Correct detailed business rules describing entities, cardinality, and relationships between entities (Justification of design).	1	

Super and sub type criteria (Note: each supertype should have at least two subtypes)	**1.5**	
19. Correct identifiers and chosen attributes (a supertype should have at least two subtypes). **Note:** PK and FK(s) should be listed, and PK in the entity should be underlined.	0.5	
20. Correct use of Completeness (total or partial) and Disjointedness (overlapping or disjoint) constraints as well as subtype discriminator.	0.5	
21. Correct description of business rules for super-type and subtypes related to attributes, completeness, disjointedness (Justification of design).	0.5	
	Total	

Notes:

1- Please check and complete the self-assessment (Part B marking criteria) and do not submit it.

2- To do the **justification of design**, please have a look at the example that is provided in Part B of the assignment specification.

3- The subject coordinator may change this marking criteria. In this case they will inform you one week prior to the assignment due.

Part C Marking Criteria

Part C: Schema Conversion and Normalization (20 marks)

Revised Business rules and ERD	**1.5**	
1. Correct list of business rules and assumptions (From part B)	0.5	
2. Revised ERD of Part B	1	
Question 3: ERD Conversion to the Relations	**5**	
3. Following specified format for writing relations	0.5	
4. Correct conversion of every regular entity into a relation (at least 3 regular entities) with correct identification of: • Primary keys (Underling primary keys) • Foreign keys (Marking foreign keys with asterisks)	1.5	
5. Correct conversion of M:N relationships (associative entity) into new relations with correct identification of: • Primary keys (Underling primary keys) • Foreign keys (Marking foreign keys with asterisks)	1	
6. Correct conversion of strong and weak entities into new relations with correct identification of: • Primary keys (Underling primary keys) • Foreign keys (Marking foreign keys with asterisks)	1	
7. Correct conversion of supertype and subtypes into new relations with correct identification of: • Primary keys (Underling primary keys) • Foreign keys (Marking foreign keys with asterisks)	0.5	

8. Correct conversion of self-referencing relationship into new relation with correct identification of: • Primary keys (Underling primary keys) • Foreign keys (Marking foreign keys with asterisks)	0.5	
Question 4-a: Normalization: Identification and Listing of Functional Dependencies	**6.5**	
9. Following specified format for writing functional dependencies, i.e. X→Y	0.5	
10. Correct identification of functional dependencies related to each business rules (at least ten BRs and at least one BR related to each FD. See the sample in "Normalization step by step" document) **Note:** You need to specify that each FD is related to which BR to get a mark.	6	
Question 4-b: Normalization: Determination of normal forms	**7**	
11. Following specified format and order for writing normalization process (for each relation, list related FDs, see "Normalization step by step")	1	
12. Correct definition of 1NF and identification of violation causes	1	
13. Correct definition of 2NF and identification of FDs causing 2NF violations	1	
14. Correct definition of 3NF and identification of FDs causing 3NF violations	1	
15. Correct identification of attributes in new normalized relations	1	
16. Correct identification of primary keys in new normalized relations	1	
17. Correct identification of foreign keys in new normalized relations	1	
Note: If your relations are normalized, use the format provided at the last page of "Normalization step by step" document to describe the process.		
Total		

Notes:

1- Please check and complete the self-assessment (Part C marking criteria) and do not submit it.

2- Please have a look at the examples that is provided in "Normalization step by step" document (You can find this document in "Assignment" or "Lesson 6" Pages on Ed).

3- The subject coordinator may change this marking criteria. In this case they will inform you one week prior to the assignment due.

Part D Marking Criteria

Part D: Implementation

1. The database is loadable without any error (2 marks)		
1.a. All tables (at least three tables) are created successfully.	0.75	
1.b. The (minimum three) generated tables have proper primary keys and foreign keys.	0.75	
1.c. There is at least one associative relationship implemented in the database.	0.5	
2. Sample data are inserted (1 mark)		
2.a. There are enough (at least 15) INSERT statements, with at least five INSERT statements presented for each table (at least 3).	0.5	
2.b. There are at least three SELECT statement in the form of "SELECT * FROM Table;" including ● An English language description of what the SQL query does. ● The select statement query	0.5	
3. Correct queries (2 mark)		
3.a. There is one SELECT statement which uses GROUP BY including ● An English language description of what the SQL query does. ● The select statement query	0.5	
3.b. There is one SELECT statement in the form of "SELECT * FROM TableA INNER JOIN TableB;" (Where TableA and TableB are two tables implemented in your database) including ● An English language description of what the SQL query does. ● The select statement query	0.5	
3.c. There is one SELECT statement which uses SUBQUERIES including ● An English language description of what the SQL query does. ● The select statement query	1	
Total		

1- The subject coordinator may change this marking criteria. In this case they will inform you one week prior to the assignment due.