

DATABASE PROJECT SPECIFICATION

Due Date : 3 weeks

I. Introduction

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 work is done by an individual or a group.

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.

II. 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**.

➤ **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, identifying 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

- 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

➤ **Part C - Logical design: Schema conversion and Normalisation**

• **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):

- 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

- **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.*

➤ **Part D - Implementation**

In Part D of the assignment, you need to create correspond 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.

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).

PART E : Create PHP-based Database Application

Create PHP-based Database Application based on the database that you already created in PART D. The application must have the ability to update, modify, and insert the data, to query the content of the database, and so on.

III. 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

Part E	PHP-based application	55	16.5
--------	-----------------------	----	------

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 (Textbook 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.	0.5	
7. Provide at least one correct mandatory cardinality and refer to the related business rule.	0.5	
8. Provide at least one correct use of one-to-many relationship (not part of associative entity's relationship).	0.5	
9. Correct relevant attributes included in all entities → add	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	0.5	
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.	0.5	
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 weak entity.	0.5	
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 attributes chosen (a supertype should have at least two subtypes).	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		

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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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. $X \rightarrow Y$	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)	6	
Note: You need to specify that each FD is related to which BR to get a mark.		
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		

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 <ul style="list-style-type: none"> • 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		0.5
<ul style="list-style-type: none"> • An English language description of what the SQL query does. • The select statement query 		
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		0.5
<ul style="list-style-type: none"> • An English language description of what the SQL query does. • The select statement query 		
3.c. There is one SELECT statement which uses SUBQUERIES including		1
<ul style="list-style-type: none"> • An English language description of what the SQL query does. • The select statement query 		
Total		

PART E MARKING CRITERIA

1. The correctness of the PHP-based application	27,5	
2. The complexity of the PHP-based application	27,5	
Total		