

INFO6002: Database Management 2
Trimester 2, 2019 - Callaghan

Assignment 2 – Database Implementation

Due: 10:00 am, Wednesday 10th July, 2019

WORTH 25% of final course mark.
This is an INDIVIDUAL Assignment.

Assignment Progress Monitoring

This assignment has four sections to it. To monitor your progress, you will be asked to present sections of your work to your tutor during upcoming lab classes. This progress monitoring is not formally marked but is used as a means to monitor how you are progressing through the assignment and provide feedback. Also, this encourages students to achieve milestones as they work on their assignment and not leave it to the last moment. The entire assignment will be marked after the due date.

Following are the sections and when they are due for progress monitoring:

Section	Description	Due Date	Type of submission
Section 1	Revised EER diagram, Relational Mapping and Normalisation	June 24 th	Progress Monitoring
Section 2 & 3	Database scripts with all constraints and sample data. Stored Procedures and Test Scripts	July 3 rd	Progress Monitoring
Sections 1 - 4	Entire Assignment	July 10 th 10am	Final marking

Section 1: Conceptual DB Design, EER to Relational Model Mapping & Normalisation
(5 marks)

In this section, you will use the Assignment 1 feedback to revise the data requirements, transaction requirements, business rules and EER model.

Next, the EER model needs to be mapped to a relational schema and normalised.

The relational model needs to be documented in DBDL format. Sample DBDL format is given below:

ISBN (id, number, itemNo)
Primary Key id

Alternate Key number

Foreign Key itemNo **references** Book(itemNo)

ON UPDATE CASCADE, ON DELETE CASCADE

DBDL format is provided in your text – Database Systems : A Practical Approach to Design, Implementation and Management – 5th Edition.

Section 2: Implementation – Database Script (5 marks)

Create a T-SQL script for the database design in section 1. You will create a database with all the necessary tables and constraints: primary key, foreign key, not null, unique and check constraints. The database must be populated with sufficient and meaningful records for evaluation purposes.

Section 3: Stored Procedure (10 marks)

Implement the following stored procedure. Ensure that the stored procedure is tested with appropriate sample data. Test cases should be saved in a separate test script.

(1) Create an order	
Procedure name	usp_createCustomerOrder
Description	This stored procedure creates a new customer order. The sales tax is 10% of order amount.
Input Parameters	Customer id – Id of customer Items – A TVP (table valued parameter) of items (item number, quantity) Discount code – Discount code (A discount code is NULL if there is no discount) Type – Type of order (phone or in-store or App) CallStartTime* – Date and time of call start (if phone order). CallEndTime* – Date and time of call end (if phone order) isHoax* – True/False (if phone order) DeliveryMode - Delivery or pickup DeliveryAddress** – Delivery address if it is a delivery order * This field is NULL for in store and App orders. ** Delivery address is NULL for pickup orders.
Output Parameter	Order number of the newly created order
Functionality	Creates a new order with the provided input parameters. After each order, the ingredients used for the order are deducted from the current stock levels of the ingredients. Returns the newly created order number. If there is any error, an appropriate error message is raised.
SQL script	create_esp_createCustomerOrder.sql
Test script	test_esp_createCustomerOrder.sql

Section 4: Business Rule (5 marks)

Business Rule: Order Satisfiability

Before an order can be taken, it is important to verify that the order can be satisfied with the available ingredients in the store. If the ingredients available are insufficient to fulfill the order, an appropriate error message needs to be generated and the order cancelled.

Ensure that the above business rule is enforced in the database. You need to generate appropriate error messages if an attempt to violate the constraint is attempted.

Notes: It is especially important for sections 2-4 that the T-SQL scripts are documented - commented and indented for clear understanding by the marker. Where complex pieces of code are written, documentation in pseudo-code is encouraged.

The test scripts also need to be well documented. e.g. a heading comment firstly to state whether the expected result is a *Success* or *Fail* test case and a description. Please ensure that you test appropriately with different inputs (valid and invalid).

Ensure that appropriate sample data is used to test your database comprehensively. Also ensure that appropriate error messages are generated.

Submission Requirements

Both a softcopy and hardcopy must be submitted for the assignment.

Softcopy

The following items need to be submitted to Blackboard via *Assessment / ASSIGNMENT 2 / Assignment 2 Submission* (will be checked by Turnitin). Include all files in a single zip folder identified by 4 sections: A2, your first name, your surname and your student number, e.g. A3SimonLee1234567.zip

1. Database documentation – Data Requirements, EER Model, Data Dictionary, Normalised Relational Schema. Note that you need to specify to which normal form that the schema is normalised. If you found any functional dependencies that violated the normal form, illustrate the normalization process.
2. SQL Script files

The **SQL script** files to be included are listed below.

Section	SQL scripts	Description
Database	createDB.sql	The createDB script contains the script that creates the database along with all constraints.
	insertSampleData.sql	This script inserts sample data to the database
Stored Procedures	create_usp_createCustomerOrder.sql	This script creates the stored procedure to create a new Customer order
	test_usp_createCustomerOrder.sql	This script contains the test cases for createCustomerOrder stored procedure
Business Rule	create_enforceBusinessRule.sql	This script considers any necessary code to enforce the business rule
	test_enforceBusinessRule.sql	This script contains any code that tests the business rule

Hardcopy

Hardcopy is to be submitted to your lecturer in person in lab class. The following items will be required for submission:

- A signed Individual Assessment cover sheet
- Printout of database documentation (Requirements, Revised EER Model, Normalised Relational Model)
- Printout of all SQL Scripts

Demonstration

Each student must demonstrate his or her assignment. Assignment 2 demonstrations will be held during the lab session on the due date. Failure to attend the demonstration can result in a zero mark for the assignment.

The assessment RUBRIC is given on the following page:

	Excellent	Good	Satisfactory	Poor	Fail
Revised EER, Normalised Relational Model (5)	(5)	(4)	(3)	(1-2)	(0)
	The requirements are clearly and correctly documented. Any issues/errors in EER model are revised and corrected. All requirements modelled accurately without any errors. The EER model is converted to the relational model and normalised without errors.	The requirements are clearly and correctly documented. The EER model has minor errors. The EER model is converted to the relational model and normalised without errors.	The requirements are clearly documented. Errors and minor misinterpretations in requirements. EER model has errors. The EER model is converted to relational model and normalized.	Poor understanding of requirements and poor documentation. Errors in requirements reflected with major errors in EER. The converted relational model has errors and is not normalised.	No submission/demonstration
Database Script (5)	(5)	(4)	(3)	(1-2)	(0)
	The T-SQL database scripts map to the section 1 design and are documented. All necessary tables and constraints are clearly shown. The database is populated with sufficient and meaningful records for evaluation purposes.	The T-SQL database scripts map to the section 1 design. Necessary tables and constraints are shown and commented. The database is populated but not sufficiently.	The T-SQL database scripts have constraints missing and is poorly commented. The tables are partially populated.	The T-SQL database scripts have tables but are missing constraints. Scripts are not commented. Tables are not populated.	No submission/demonstration
Stored Procedure (10)	(9-10)	(7-8)	(4-6)	(1-3)	(0)
	The functionality is correctly implemented. The code is well documented. All appropriate warnings and errors are generated. The code is tested for all relevant cases.	The functionality is correctly documented. The code is partially documented and/or tested. The code is tested for major cases.	The functionality has minor errors in implementation. The code is partially documented. The code runs and has at least 1 test case.	The functionality has major errors. The code is not documented. The code does not run and has minimal testing.	No submission/demonstration
Business Rule (5)	(5)	(4)	(3)	(1-2)	(0)
	The business rules are correctly implemented. The code is well commented. All appropriate error messages, warnings etc. are generated. The test script ensures all appropriate test cases are tested and code verified.	The business rules are correctly implemented. The code is partially commented. Appropriate error messages are generated. The test script ensures major test cases are verified.	Minor errors in implementing business rules. The code is partially commented. Error messages are generated. The testing is partial.	Major errors in enforcing rules. Poor comments. Code does not run or has errors. Warning/errors are not appropriately raised. Few/no testing.	No submission/demonstration

* The assignment needs to be demonstrated and the tutor will determine the expertise and authenticity of the assignment.