

COMP3350

Assignment 1 – Database Design & Implementation

Semester 1, 2021

Due Date March 29th (Monday) at 8am

Assignment 1 is due on March 29th (Monday) at 8am. Each team will

- upload the assignment to UOnline by March 29th (Monday) at 8am and
- demonstrate the assignment to the tutor in the lab session on week 6 (commencing on March 29th)

Weighting

25% of course mark

Assignment Team

You need to attempt the assignment in a group of 2 members (3 members in exceptional circumstances). You need to fill in the Group Formation Form at the end of the assignment and submit it to the tutor on March 4th during the tutorial session. Note that your group members must be enrolled in the same tutorial session as you. Members in the same group may be awarded different marks based on their contribution to the assignment.

This assignment has 6 sections to it. Progress in each part for the group is monitored during tutorials. Note that all SQL scripts and Java code created in this assignment needs to be commented for readability (Marks will be deducted for uncommented, illegible SQL scripts/Java code).

Assignment Background

You are asked to develop a conceptual database design using Enhanced Entity Relationship model for a database for *Tasty'n'Yummy Pizzas*.

Tasty'n'Yummy Pizzas is a pizza outlet at Charlestown. After years of managing the records manually, *Tasty'n'Yummy Pizzas* has decided to computerise its records. You are tasked to design the conceptual database design for *Tasty'n'Yummy Pizzas*'s database based on the business requirements provided in this document.

Your lecturer will act as your client and you can query him for any further information and clarifications.

Business Requirements

Order Processing

Tasty'n'Yummy Pizzas takes orders via phone, online (via an app and website) as well as through walk-in customers. *Tasty'n'Yummy Pizzas* provide both delivery and pickup services.

When a customer orders via the phone, the customer's phone number is entered to the system along with the id of the person taking the order. If the customer has previously ordered by phone, the name and address appears on the screen. The customer is then asked for his name and address and then takes the order. If the customer has not ordered before or if the name and address given does not correspond with that in the computer, a new customer record is created and the order is taken. For a walk-in customer, the customer's name is recorded to identify the order. If the customer is using the app or website, the customer can login using his/her email or phone number and password. Alternatively, the customer can order as a guest.

Each order contains date and time of order, one or more items, quantity of each item, price of each item, subtotal, discount amount (if any), discount code (if any), tax, total amount due, payment method, status, type (phone/walk-in/online) and description. If the payment is via card, a payment approval number is recorded. If the order is a delivery order, the driver who delivered the order is recorded. It is possible to pre-order for the order to be fulfilled at a later date/time.

There are different types of discount programs offered by *Tasty'n'Yummy Pizzas*. All discount programs are recorded in the database. A discount program has a discount code, description, start date of discount program, end date of discount program, requirements for discount, discount percentage (e.g. 5%) and a description.

Menu Items, Ingredients and Suppliers

Each item in the menu has an item code (unique), name, size and a current selling price. An item in the menu is made up of a number of ingredients. The ingredients and quantities used for the item are recorded in the database.

Each ingredient has a code (unique), name, type, description, stock level at stocktake, date last stocktake was taken, suggested current stock level, reorder level and a list of suppliers who supply the ingredients. A supplier can supply many ingredients.

A stocktake is taken each week, where the actual levels of ingredients in store are recorded. This is then compared with suggested levels (based on orders for the week). This report is used by the store manager to order ingredients for the following week. Information about supplier orders needs to be maintained in the database.

Employees

Employees at the store can be divided into two types: those who work in the shop are paid hourly and those who carry out deliveries are paid by the number of deliveries. For each employee, there is an employee number, firstname, lastname, postal address, contact number, tax file number, bank details (bank code, bank name, account number), a payment rate, status, and a description. Drivers also have a driver's license number. Hours are not regular and a record is kept for each time an employee works – a shift (start date, start time, end date, end time). The orders a driver delivers during a shift is kept on record.

Payment rates for shop workers and drivers are maintained in the database. Employee payments are made for each shift to the employee's bank account at the end of each week. Employee payment records needs to be maintained in the database.

Reference: Requirements modified from case study at
<http://programmingou2k9.blogspot.com.au/2011/08/case-study-scenario-1-er-model.html>

Section 1: Conceptual DB Design (10 marks)

Based on the business requirements, develop a conceptual database design using EER model. Your lecturer will act as the client and you can speak to him to clarify any questions regarding the requirements.

Section 2: EER – Relational Mapping & Normalisation (2 marks)

The EER diagram needs designed in section 1 needs to be mapped to a relational schema and normalised.

The relational model needs to be described in DBDL format. Sample DBDL format is given in the 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 3: Database Implementation – Database Script (3 marks)

Create a T-SQL script for the database design in section 2. 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 purpose.

Save this SQL script as *CreateDB.sql*.

Section 4: Stored Procedures (5 marks)

Implement the following stored procedures. Ensure that each stored procedure is tested with appropriate sample data. Also ensure that appropriate error messages are generated 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 of items (item number, quantity) ordered Discount code – Discount code (A discount code is NULL if there is no discount) Type – Type of order (phone, in-store, online) OrderDateTime – Date and time of when order is made DateTimeOrderNeedsFulfilling – Date and time the order needs to be fulfilled DateTimeOrderComplete – Date and time order was fulfilled

	DeliveryMode - Delivery or pickup DeliveryAddress* – Delivery address if it is a delivery order. Delivery address is NULL for pickup orders. PaymentConfirmation – Id of payment confirmation OrderTakeBy – Staff id of staff taking the order if it is a phone order or in-store order
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 suggested 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_usp_createCustomerOrder.sql
Test script	test_usp_createCustomerOrder.sql

Section 5: Business Rules (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.

Section 6: Java Hibernate (2 bonus marks)

Implement the following Java application that has a Form that allows for an employee to make an order. Your form needs interact with the database using Hibernate ORM. You should call *usp_createCustomerOrder* stored procedure implemented in Section 5.

You need to test your form with valid and invalid data and ensure appropriate error messages are generated for incorrect input.

Submission Requirements

Each group will submit soft copy for the assignment.

Soft copy

Soft copy submission is via UOnline. Each group will submit a zip file which contains the following items:

1. Database documentation.
2. SQL Server Backup (.bak file)
3. SQL Script files
4. Java files

A Peer Group Evaluation Form needs to be submitted individually by each member via the TurnItIn link.

The **database documentation** must contain the following sections (in the order specified):

1. Assumptions: Assumptions made by your group with respect to the business requirements
2. ERD: The ERD must show all the entities, attributes and relationships.
3. Database Schema: The logical model must show all the tables, columns, column types, null/not null constraints, primary keys and foreign keys

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

Section	SQL scripts
Database	createDB.sql
Stored Procedures	usp_createCustomerOrder_<groupNo>.sql test_ usp_createCustomerOrder _<groupNo>.sql
Business Rule	create_ usp_OrderSatisfiability.sql test_ usp_OrderSatisfiability.sql

Progress Monitoring

Each groups need to show progress of their assignment during the tutorial sessions. Each member of group needs to attend these tutorial sessions.

<u>Section</u>	<u>Progress</u>
Sections 1	Week 3 tutorial (commencing on March 8 th)
Sections 2 and 3	Week 4 tutorial (commencing on March 15 th)
Sections 4 and 5 (optional 6)	Week 5 tutorial (commencing on March 22 nd)

Demonstration

All groups need to demonstrate their database implementation. Group demonstration for Assignment 1 will be held during **Week 6 tutorial/lab session** (commencing Mar 29th). Each member's contribution will be evaluated individually during demonstrations. Failure to attend the demonstration can result in a zero grade for the assignment by the member of the group.

The assessment RUBRIC is given below:

	Excellent	Good	Satisfactory	Poor	Fail
	(9-10)	(7-8)	(4-6)	(1-3)	(0)
Conceptual Database Design (10)	The requirements are clearly and correctly documented. All requirements modelled accurately without any errors.	The requirements are clearly and correctly documented. The EER model has minor errors	The requirements are clearly documented. Errors and minor mis-interpretations in requirements. EER model has errors.	Poor understanding of requirements and poor documentation. Errors in requirements reflected with major errors in EER.	No submission/ demonstration
EER – Relational Mapping & Normalisation	(2)		(1)		(0)
	The EER model is converted to the relational model and normalised without errors.		. 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 script maps to the earlier section design and documented. All necessary tables and constraints are clearly shown. The database is populated with sufficient and meaningful records for evaluation purpose.	The T-SQL database scripts maps to the earlier section design. Necessary tables and constraints are shown and commented. The database is populated but not sufficiently.	The T-SQL scripts has constraints missing and poorly commented. The table are partially populated.	T-SQL script has tables but missing constraints. Script is not commented. Table are not populated.	No submission/ demonstration
Stored Procedures (SP1 - 5)	(5)	(4)	(3)	(1-2)	(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 run and has at least 1 test case	The functionality has major errors. The code is not documented. The code run and little to no 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 rule are correctly implemented. The code is partially commented. Appropriate error messages are generated. The test script ensures major test cases verified.	Minor errors in implementing business rules. The code is partially commented. Error message generated. The testing is partial.	Major errors in enforcing rules. Poor comments. Code does not run or 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 submission for each group member.

Group Formation Form

GROUP NO: _____ (TO BE FILLED BY LECTURER)

GROUP MEMBERS:

I agree to participate in the mentioned group for COMP3350 Assignment 1

Student ID	Name	Signature