COMP1140 S2 2023

**Assignment 1**

**Project: Database design of Numberone Pizza**

Requirement Analysis and Conceptual Design

(Please note: this is a templet only. You are suggested to follow this templet in completing your A1, with all needed content.)

Table of Contents

Preface: 1

[Part 1: Requirements x](#_Toc240478388)

[Data Requirements x](#_Toc240478389)

[Transaction Requirements x](#_Toc240478390)

[Business Rules x](#_Toc240478391)

[Part 2: EER Model with Data Dictionary x](#_Toc240478392)

[EER Model x](#_Toc240478393)

[Data Dictionary x](#_Toc240478394)

[Entities x](#_Toc240478394)

[Relationships x](#_Toc240478394)

[Attributes x](#_Toc240478394)

Summary ………………………………………………………………………………….…………x

# Preface

This is the requirements analysis and conceptual design for the implementation of the database of *Numberone Pizzas*.

The requirements analysis includes the data requirements, transaction requirements and business rules. This will ensure that all the data required to be stored is identified, can be manipulated and is managed according to the store’s business policies.

The conceptual design includes an extended entity relationship diagram in UML which describes the required entities, their attributes and their relationships. It is further clarified with a data dictionary for the entities, attributes and also the relationships. This conceptual model was developed from the requirements analysis.

This document …

# Part 1: Requirements

## Data Requirements

Data Requirements refer to the specific types of data that a system must be able to accept, store, process, and present. In the context of your question, the major data items that your system needs to handle are outlined as follows:

1. **Production Data**: The system's database must be capable of accepting and storing production data. This includes a 'Lot' (likely a string or numeric identifier for a batch of product), a 'Product Number' (an integer that uniquely identifies a product), and a 'Date' (a date that could represent when the product was manufactured or added to the system). These data types correspond to standard database data types: VARCHAR for strings, INT for integers, and DATE for dates [w3schools.com](https://www.w3schools.com/sql/sql_datatypes.asp).
2. **Image Data**: The system should have the capacity to store image data. This is typically accomplished using the Binary Large Object (BLOB) data type, which can store large amounts of binary data, such as image files [stackoverflow.com](https://stackoverflow.com/questions/5414551/what-is-it-exactly-a-blob-in-a-dbms-context), [digitalocean.com](https://www.digitalocean.com/community/tutorials/how-to-use-the-mysql-blob-data-type-to-store-images-with-php-on-ubuntu-18-04).
3. **Report Generation**: The system should be able to generate and print reports without any delays. This requirement is more about the system's performance than about a specific data type. It implies that the system should be designed and optimized to retrieve and process data efficiently.
4. **User Details**: The system should be able to store user details as strings. In database terms, this is typically done using the VARCHAR data type, which can store alphanumeric characters [w3schools.com](https://www.w3schools.com/sql/sql_datatypes.asp).
5. **User Status**: The system should store boolean values to indicate the status of the user. A boolean is a data type that has two possible values: true or false. It might be used to indicate whether a user's account is active, whether a user is online, or any other binary state [w3schools.com](https://www.w3schools.com/sql/sql_datatypes.asp).

In summary, these data requirements outline the major types of data that your system needs to handle, and they suggest the database design and performance characteristics that your system should have.

* **Order Processing**

Order

An order can be placed over the phone or instore and be picked up or delivered. When an order is created, information is recorded in the system such as OrderNo, OrderDateTime, CustomerInfo, StaffInfo, TypeOfOrder (phone or walkin inheritance), TotalAmountDue, PaymentMethod, PaymentApprovalNo, MenuItemInfo, QuantityOfEachOrderedMenuItem, and order Status.

An order can be either phone order or walkin order. A WalkinOrder includes WalkinTime. A PhoneOrder includes TimeCallAnswered, TimeCallTerminated, and delivery or pickup. A Delivery phone order includes DeliveryAddress, DeliveryTime, and the shift information. A Pickup phone order includes PickupTime.

More …

## Transaction Requirements

Transaction requirements outline the types of operations that your system should be able to perform on data. These operations are typically divided into Data Manipulation Language (DML) operations, which involve changing data, and queries, which involve retrieving data. Here are the details of the transaction requirements mentioned:

**Data Manipulation:**

Data manipulation refers to the process of modifying data to make it easier to read or analyze. DML is a language that includes commands used to add, remove or alter data in a database. The main commands in DML are:

1. **INSERT**: This command is used to add new records into a database table. The basic syntax is:

INSERT INTO table\_name (column1, column2, ...)

VALUES (value1, value2, ...);

1. **UPDATE**: This command is used to modify existing records in a database table. The basic syntax is:

UPDATE table\_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

1. **DELETE**: This command is used to remove records from a database table. The basic syntax is:

DELETE FROM table\_name WHERE condition;

**Data Manipulating Queries:**

These are the queries that retrieve data from the database and present it in a useful way. The main query command in SQL is SELECT, which is used to select data from a database. The data returned is stored in a result table, called the result-set.

1. **SELECT**: This command is used to select data from a database. The data returned is stored in a result table, called the result-set. The basic syntax is:

SELECT column1, column2, ...

FROM table\_name;

1. **UPDATE**: This command is used to modify the existing records in a table. The basic syntax is:

UPDATE table\_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

1. **INSERT**: This command is used to insert new records in a table. The basic syntax is:

INSERT INTO table\_name (column1, column2, ...)

VALUES (value1, value2, ...);

1. **DELETE**: This command is used to delete existing records in a table. The basic syntax is:

DELETE FROM table\_name WHERE condition;

These commands provide the basic functionality for manipulating data in a SQL database [mssqltips.com](https://www.mssqltips.com/sqlservertip/7511/sql-cheat-sheet-for-select-insert-delete-and-update-commands/), [opensource.docs.scylladb.com](https://opensource.docs.scylladb.com/stable/cql/dml.html).

## Business Rules

* An order’s type can only be either:
  + Pick up
  + Delivery

Business rules define the operations, definitions, and constraints that apply to an organization in achieving its goals. They describe the operations, definitions, and constraints that apply to an organization. They are important in ensuring that the system's data and operations are in line with the organization's goals and requirements.

Here are the explanations for the provided business rules:

1. **Serving Multiple Users**: The system should be able to serve at least 20 users at a time without delays. This is a performance requirement that ensures that the system can handle multiple concurrent users. It means that the system's architecture and infrastructure should be designed to support this level of concurrency, possibly using techniques such as multi-threading, load balancing, and efficient database queries [ayende.com](https://ayende.com/blog/2278/input-validation-vs-business-rules-validation).
2. **Data Validation**: The system should be able to validate data fields. This means that before data is saved into the system, it should be checked to ensure it meets certain criteria. For example, email fields should be checked to ensure they contain a valid email address, numeric fields should be checked to ensure they contain numeric values, etc. This is typically done using validation rules or constraints in the database or the application code [corporatefinanceinstitute.com](https://corporatefinanceinstitute.com/resources/data-science/data-validation/), [docs.oracle.com](https://docs.oracle.com/middleware/11119/adf/develop/bcvalidation.htm).
3. **No Default Values**: The system should not add default values for any column. This means that if a value for a particular column is not provided, the system should not automatically fill it with a default value. Instead, it might leave it empty (NULL in database terms), or it might require the user to provide a value before the data can be saved.
4. **Data Type Definition**: The system must have data types defined in the user requirements. This means that for each piece of data that the system handles, there should be a defined data type. Data types might include integers, strings, dates, etc. The data type for a piece of data dictates what operations can be performed on it, how much space it takes up in storage, and how it should be validated [docs.oracle.com](https://docs.oracle.com/middleware/11119/adf/develop/bcvalidation.htm).

These business rules provide a set of guidelines that will dictate how the system should be designed and implemented. They are typically derived from the organization's operational requirements and strategic goals, and they help to ensure that the system will meet the organization's needs [learn.microsoft.com](https://learn.microsoft.com/en-us/sql/master-data-services/business-rules-master-data-services?view=sql-server-ver16).

# Part 2: EER Model

## EER Model



Data Dictionary

## Entity

Note: xxx entities (including sup & sub)

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Description** | **Aliases** | **Occurrences** |
| Order | describing orders customers have made | Pizza Order | When an order is made by a customer |
|  |  |  |  |

***Relationships***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Order | 0..\* | Consists of | 1..\* | ManuItem |
| (Man,Or) | Generalisation | (Man,Or) | WalkInOrder |
| (Man,Or) | Generalisation | (Man,Or) | PhoneOrder |
|  |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |  |

***Attributes:***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Derived** | **Default** |
| Order | OrderNo | Unique order identifier | char(10) | N | N | N |  |
|  | OrderDateTime | The date & time the order is made | datetime | Y | N | N |  |
|  | OrderType | Walkin or Phone order | varchar(10) | Y | N | N |  |
|  | TotalAmountDue | Total money for the order | float | Y | N | N |  |
|  | PaymentMethod | Total 3 kinds of payment | varchar(10) | Y | N | N |  |
|  | PaymentApprovalNo | Payment Approval No | varchar(10) | Y | N | N |  |
|  | Status | Current state of the order, including executed or not | varchar(10) | Y | N | N |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**Summary ………………………………………………………………………x**