DAT601 assessment 1

Ollie Moss

Contents

[Assignment 1 Part 1: Conceptual database design 1](#_Toc163551980)

[Overview 1](#_Toc163551981)

[Scenario 1](#_Toc163551982)

[Introduction to data modelling in information systems. 2](#_Toc163551983)

[Extended (aka Enhanced) Chen Entity Relationship Diagramming (ERD) 3](#_Toc163551984)

[Conceptual ER model 4](#_Toc163551985)

[Data dictionary 4](#_Toc163551986)

[Assumptions made about the business rules 4](#_Toc163551987)

[Assignment 1 Part 2: TSQL 5](#_Toc163551988)

[Overview 5](#_Toc163551989)

[Adding Foreign Keys 5](#_Toc163551990)

[SQL Queries 5](#_Toc163551991)

[Sub Queries 7](#_Toc163551992)

[Combined Query 8](#_Toc163551993)

[Views 9](#_Toc163551994)

[NOTES 10](#_Toc163551995)

[**References** 11](#_Toc163551996)

# Assignment 1 Part 1: Conceptual database design

# Overview

This part of the assignment is focused on the conceptual model to be used in the context of the scenario described below. This is the first stage of the project that will be expanded upon using the logical and physical model in assignment 2.

# Scenario

FlightStream is a “not for profit” organisation that provides remote video streaming and data sensing to government scientific services and commercial organisations and presence services to subscribers. For example, data required by government departments such as MFAT and the Department of Conservation. FlightStream deploys data sensing, collecting, and transmitting drones that can be managed by specific mobile devices. FlightStream brands their drones as “DataScoop” with an accompanying application for mobile devices. The “DataScoop”, facilitates data sensing and video streaming to be used for scientific and commercial purposes. The “DataScoop”, maintains its communication with FlightStream by satellite and/or ground based mobile cellular communication networks.

Introduction to data modelling in information systems.

Includes an outline of the fundamental principles of effective data management. Include an overview of Conceptual, Logical data modelling and Physical implementation. (5 marks) (LO1, LO4)

## Conceptual model

A model of the information used in an enterprise independent of all physical considerations.

The conceptual model involves forming a basis and a solid structure to work with in the later stages. Although using a higher-level model it is not necessarily less detailed. The stages to this include defining the scope, boundaries of the database, its users, and areas of use. This helps to further understand the databases use cases and purpose.

Furthering upon this gathering requirement and analysing use cases, scenarios, and roles to further understand how the database will function.

This information can be used to identify entities, in the system will help to provide a basis for developing a conceptual ER model. There are various ER models in use today, however in the context of this assignment the extended ER model will be used for this.

This is the outline for the conceptual stage of data modelling in information systems and how it develops a structured framework to work with in the later stages of development.

## Logical Model

Model of the information used in an enterprise based on specific data model independent of DBMS and physical systems.

Physical implementation:

Producing a description of the implementation of the database on secondary storage; it describes the storage structures and the access methods to achieve efficiency.

Involves translating the logical data model into a physical database schema that can be implemented on a specific database management system (DBMS).

Defines data types, indexes, constraints, and storage structures to optimize performance and meet the requirements of the target DBMS.

Considers factors such as scalability, availability, and security when designing the physical database schema.

## Database lifecycle

Database planning: how to realise the stages efficiently and effectively.

System definition: Scope, boundaries of app database, app users, and app areas

Requirements collection and analysis: use case, scenarios and roles

Conceptual design

Logical design

Physical design

Application design

Implementation, data conversion and loading, and testing and maintenance.

Prototyping, Agile sprints, Testing

Chens notation – used in conceptual modelling.

# Extended (aka Enhanced) Chen Entity Relationship Diagramming (ERD)

An introduction to conceptual modelling using Chen ERDs with a depiction and description of all components of a Chen ERD, include extended components. Describe how the fundamental principles of data management are applied through of the extended Chen ERD in enterprise modelling. (10 marks) (LO1, LO4)

CHENs Notation is a high-level data model used for the conceptual design of a database. This is still not less detailed than further stages of the data modelling stage. Making use of the ER diagram helps to specify the desired components of a database and the relationships within it. It provides flexibility so that it can be used in any environments that require information to be modelled.

## Entities

Entities represent real-world elements that will be involved in the system.

Describing entities in the extended CHENs Notation are as follows:

* Entities are drawn in a box.
* Entity type defines a collection of entities that have same attributes.
* Entity type is defined by its name and attribute.
* Entity type is named as a singular noun. E.g. person instead of people.
* Entity instance is a single occurrence of an entity type.
* Collection of entities is called an entity set. Contains all records of entities.

Weak entities

## Attributes

Attributes are properties that describe entities.

Describing attributes in the extended CHENs Notation are as follows:

* Enclosed by ovals connected to their entities with a single line.
* Value set or domain – represents all the different values an attribute can have (Data type)
* An attribute that’s underlined represents a primary key or candidate key.

## Attribute types:

* Simple attribute – represents a single data type (can’t be split into separate attributes) e.g. IRD number.
* Composite attribute – can be divided into smaller subparts. E.g. address can be split into street number, street name, etc.
* Single valued attribute - single value for a particular entity (single oval)
* Multi valued attribute – Set of values (doubled circled oval)
* Derived attribute – Attribute that can be derived from another attribute e.g. age can be derived from birthdate. (dotted oval)
* Stored attribute - Attribute from which other attributes are derived from e.g. birthdate is used to derive age.

Determining whether an attribute should be split into a composite attribute can be done by looking at use cases and functions for attributes and deciding whether they should be composite or simple attributes e.g. Should address be split into its components, if there is no need to refer to the individual components then there is no need to split it into simple attributes.

## Keys

Keys play a crucial role in defining the uniqueness of data within a database model. Keys help identify individual instances of entities and establish relationships between entities.

## Types of Keys:

* Primary Key (PK) - The primary key identifies each entity within a set of entities. Represented by underlining the attribute that is defined as the primary key.
* Composite Key - A composite key is made up of multiple primary keys. Represented by underlining multiple attributes within an entity. Composite keys are used when a single attribute cannot uniquely identify each entity instance, but a combination of attributes can.
* Foreign Key (FK) - A foreign key defines a relationship between two entities by referencing the primary key of another entity set. Represented by connecting an attribute in one entity to the primary key of another entity using a dashed line.
* Partial Key – A partial key serves as a unique identifier for a weak entity as it only uniquely identifies an entity in the context of a relationship and not the entity set as a whole. Represented using a dotted underline on the chosen attribute.

## Relationships

Relationships describe the connection between two entities. They are Represented with a diamond and lines connecting to the related entities.

## Relationship Attributes:

Describe properties or characteristics of relationships, represented by ovals connected to the relationship lines. Relationships have attributes that are related to the relationship itself, rather than to the entities connected in the relationship.

## Cardinalities:

Indicate the minimum and maximum number of occurrences of one entity that can be associated with another entity through a relationship. Represented using numbers: "1" for one occurrence, "N" for many occurrences.

## Participation Constraints:

Specifies whether the participation of an entity in a relationship is mandatory or optional.

Represented with a double line on desired side of the relationship for mandatory participation.

## Specialization and Generalization:

Used to represent inheritance between entity types.

Specialization represents the process of defining subtypes based on common attributes.

Generalization represents the process of defining a supertype that subtypes can be derived from.

Aggregation represents a relationship where one entity is made up of multiple entities.

## Relationships Degrees

Relationship degrees describe how many entities are participating in a given relationship.

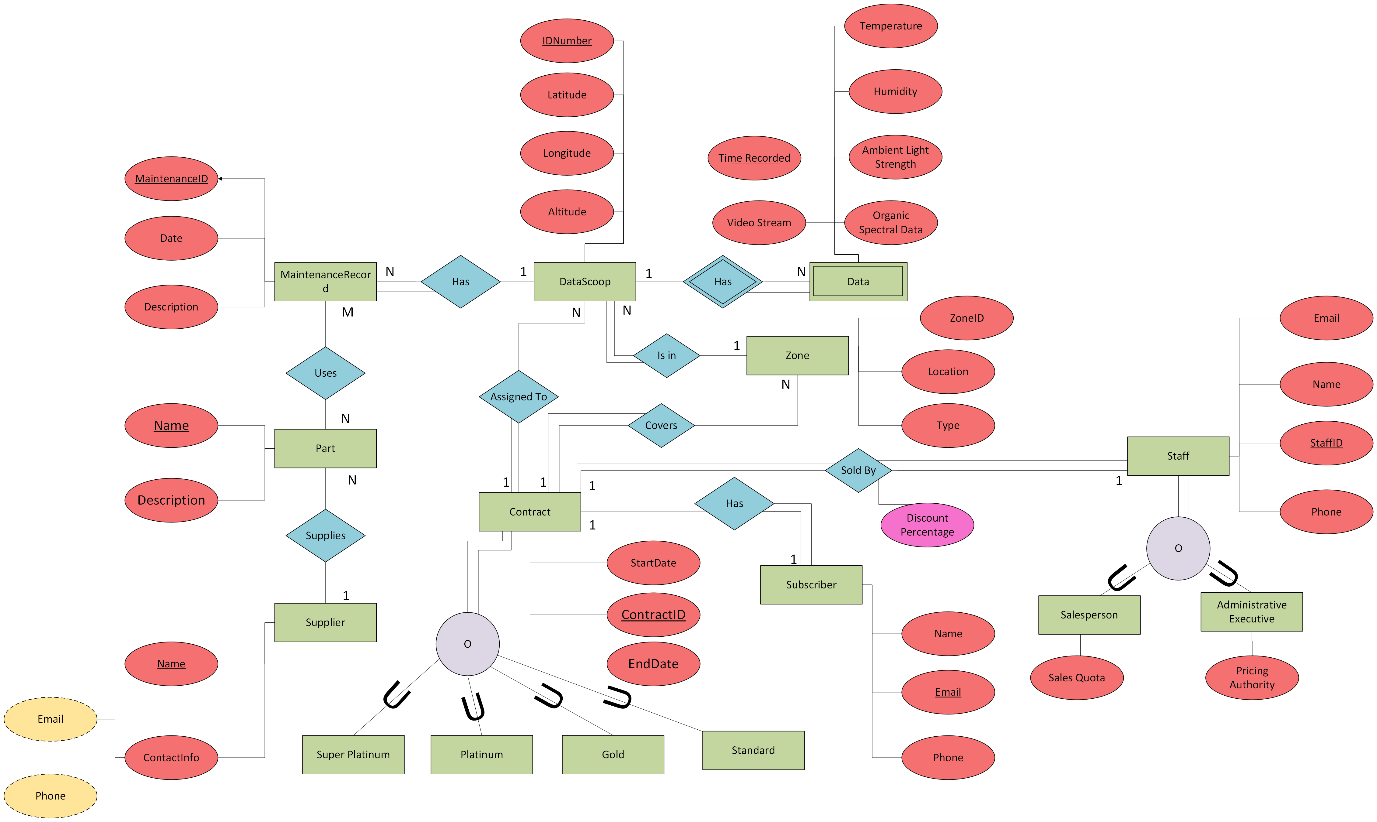
Unary Relationship: Relates an entity to itself.

Binary Relationship: Relates two different entities.

Ternary Relationship: Relates three entities.

Conceptual ER model

Accompany this with your rationale; describe and explain the reasoning and purpose of all parts of the model. Write and reflect on how the fundamental principles of data management are applied through conceptual ER modelling using the Chen Extended ERD . (20 Marks) (LO1, LO4)



Data dictionary

Covers the following: ○ Entities (e.g. name description, aliases, occurrences) ○ Relationships (e.g. name, multiplicity) ○ Attributes (e.g. name, description, domain, aliases, composite, derived, nulls, key, default value). Accompany this with your rationale; describe and explain the reasoning and purpose of all parts of the model. Write and reflect on how the fundamental principles of data management are applied by using data dictionaries. (LO1, LO4)(10 marks)

Business Rules Assumptions

Include a reflection in outline of how the fundamental principles of data management as they are applied to assumptions about the business rules. (LO1, LO4)(5 marks)

## Data Integrity

Assumes that data recorded by DataScoops is accurate and reliable.

## Data Security

Assumes that effective security measures are put in place ensuring that privacy is upheld, and data breaches are minimised.

## Data Governance

Assumes that FlightStream adheres to regulations and standards in the context of data collection, and transmission.

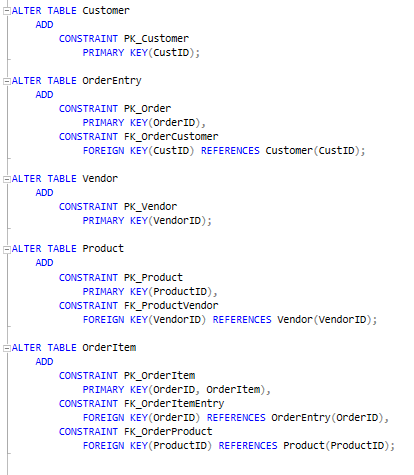
# Assignment 1 Part 2: TSQL

# Overview

Part 2 of this assignment focuses on SQL knowledge and practice. Using the provided database and data, using SQL Server tasks surrounding SQL queries and practice will be completed to demonstrate these skills. Including adding relationships, data and creating queries and views to retrieve data.

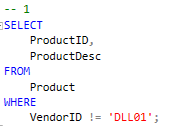
# Adding Foreign Keys

The database does not contain any Foreign Key constraints – add all the required Foreign Keys to your copy of the database using SQLServer SQL ALTER TABLE commands. Write an ALTER TABLE statement for each Foreign Key.



# SQL Queries

1. List all products not made by vendor DLL01.



1. List all the products with a price between $5.00 and $10.00.

A screen shot of a computer

Description automatically generated

1. List any products made by either vendor DLL01 or vendor BRS01 costing $10.00 or greater.

A computer screen shot of a computer

Description automatically generated with medium confidence

1. Return the average price of all the products in the Products table.

A close-up of a text

Description automatically generated

1. Return the total number of customers in the Customers table.

A close up of a number

Description automatically generated

1. Return the number of customers in the Customers table with an e-mail address.

A close up of a number

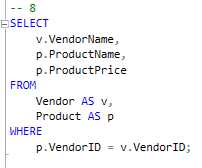
Description automatically generated

1. Return the number of product types, minimum, maximum and average product price from the products table.

A screenshot of a computer code

Description automatically generated

1. Return the vendor name, product price and product name from the vendors and products tables.



1. Return the product name, vendor name, product price and quantity for each item in order number 20007.

A screenshot of a computer

Description automatically generated

## Sub Queries

1. Create a list of all the customers (customer name and customer contact) who ordered item RGAN01.

A screenshot of a computer

Description automatically generated

1. Display the total number of orders placed by every customer in the Customers table, as well as the city the customer is in.

A screenshot of a computer program

Description automatically generated

## Combined Query

1. Create a report on all the customers in Nelson and Wellington. You also should include all Fun4All locations, regardless of city. The resulting customers should be in alphabetical order of customer name then customer contact.

A screenshot of a computer program

Description automatically generated

# Views

1. Create a view called vProductCustomer which joins the Customer, Order and OrderItem tables to return a list of all customers who have ordered any product. Now retrieve from that view a list of customers who ordered product RGAN01.

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

1. Add a customer to the database: CustID = 1000000006 CustName = The Toy Emporium CustPhone = 09-546-8552

Using a view to format mailing list data: - First create a query that will display the customer name and then the address in the following format: CustName Customer address City/town, Phone number

Next turn this query into a view called vCustomerMailingLabel

Display all the “entries” in vCustomerMailingLabel

Try defining the customer mailing label view so that it filters out any incomplete addresses as these cannot be used for mailing labels.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## NOTES

Hand in the URL to a GitHUB repository created and owned by you that contains the work from parts one and two above.

<https://github.com/Ollie-Moss/DAT601-AS1>

Notes:

Rich modelling, story boards, use case diagrams (function centric, not data centric harder to use in data modelling)

Data architecture

External level

Users, customers, clients, employees.

This refers how individual users see the data.

Conceptual level

This refers to the description of data of interest to the organization. (Enterprise organization user view)

NOT LESS DETAILED

Internal level

How the data is physically stored.

Domains should be more specific rather than broader such as restrictions for certain attributes.

What is an object?

Codeasyl – Was one of the first hierarchical databases, like OO databases. Was used as a standard for many organizations.

# **References**