Module 2 Assignment

**Part 1**

A Relational Database is a database where data are organised into tables and those tables may have defined relations between them. One of the main Databases that I work with is the Savings Database within the NHS Supply Chain’s system. The Savings Database is split into 4 key Relational Databases; Category Towers Service Providers (CTSP), APELS, Libra and Housekeeping. For this account, I will be looking at the Category Towers Service Providers (CTSP) Relational Database.

CTSP is split into 11 different entities (which are known as Category Towers):

1. NHS Supply Chain: Ward Based Consumables
2. NHS Supply Chain: Sterile Intervention Equipment and Associated Consumables
3. NHS Supply Chain: Infection Control and Wound Care
4. NHS Supply Chain: Orthopaedics, Trauma and Spine, and Ophthalmology
5. NHS Supply Chain: Rehabilitation, Disabled Services, Women’s Health and Associated Consumables
6. NHS Supply Chain: Cardio-vascular, Radiology, Endoscopy, Audiology and Pain Management
7. NHS Supply Chain: Large Diagnostic Capital Equipment Including Mobile and Services
8. NHS Supply Chain: Diagnostic, Pathology and Therapy Technologies, and Services
9. NHS Supply Chain: Office Solutions
10. NHS Supply Chain: Food
11. NHS Supply Chain: Hotel Services

The CTSP Relational Database is used to develop frameworks which will ensure delivery of clinically evaluated, quality products at the best possible value.

Some advantages of a relational database include:

* Efficiency – increase speed and storage.
* Redundancy – less redundant and duplicated data.
* Update issues – fewer problems updating data.
* Deletion – less chance of deleting important data.
* Standardisation – database design follows consistent design principles.

With regards to the CTSP Relational Database, a problem currently is there are issues with efficiency and redundancy.

The issues with efficiency and redundancy come from the database being so large – approximately 1 million rows are added to each table in the database each month and nothing can be deleted from the table. The size of each table makes it difficult for the database to adhere to the three stages of normalisation.

Normalisation is a step-by-step process for taking a table and converting it into an efficient relational database structure. Typically, a 3-step process, which is outlined below.

1st Normal Form Criteria:

* Each cell contains only one data point.
* Each column contains only one data type.
* Columns should have a unique name.

2nd Normal Form Criteria:

* Each table contains relevant data about one type of entity.
* There are no partial dependencies.

3rd Normal Form:

* There are no transitive dependencies.

My role with, regards to this database, is helping to solve the issues with efficiency and redundancy. The CTSP Relational Database is made and updated manually. This leads to a lot of errors in the database, such as columns containing multiple datatypes due to mistypes. Part of my role consists at applying normalisation techniques to the database with the help of the erwin Data Intelligence tool.

With regards to the 1st Normal Form criteria, the columns in the database all have a unique name, however, often there are columns which have multiple data points or multiple data types. These do get rectified; it is just a very time-consuming process. The database meets all criteria of 2nd Normal Form. However, there are transitive dependencies in the database, which means 3rd Normal Form is not met.

A transitive dependency occurs when a field can be inferred from another field that is not the primary key – i.e., you can transition from a field to the primary key via another field.

Therefore, the database does not yet adhere to the 3 stages of normalisation.

**Part 2**

The open-source dataset on Kaggle that I decided on was a films database (Link: [Movies on Netflix, Prime Video, Hulu and Disney+ | Kaggle](https://www.kaggle.com/ruchi798/movies-on-netflix-prime-video-hulu-and-disney)). The dataset comprised a comprehensive list of movies available on various streaming platforms.

Raw dataset:



The columns included in the raw dataset are:

* (Blank)
* ID
* Title
* Year
* Age
* IMDb
* Rotten Tomatoes
* Netflix
* Hulu
* Prime Video
* Disney+
* Type
* Directors
* Genres
* Country
* Language
* Runtime

To convert this into a relational database, normalisation first needed to take place.

1st Normal Form Criteria:

* Each cell contains only one data point.
* Each column contains only one data type.
* Columns should have a unique name.

2nd Normal Form Criteria:

* Each table contains relevant data about one type of entity.
* There are no partial dependencies.

3rd Normal Form:

* There are no transitive dependencies.

To begin with the redundant columns, ‘(Blank)’ and ‘Type’ were removed, since ‘Type’ had only had 1 value (0) and ‘(Blank)’ was a repeated ID column.

‘Directors’, ‘Genres’, ‘Country’ and ‘Language’ were then also updated to make sure they only had 1 value in each field.

The dataset was then split into multiple tables to form a relational database. These tables were:

* Film
* Information
* Platform
* Rating
* Genre
* Director
* Country

Column names were also updated where appropriate and other ID columns were added to complete the normalisation process.

The head of each table can be seen in this embedded folder:



An entity relationship diagram was then created to show how the different entities relate to each other within the system.

Graphical user interface, application, Teams

Description automatically generated