# Database Design Assessment #1 by Aleksei Stukalov

*Note: I apologies for this late assessment work, passed month was a very much a down period for me, after I got better from COVID the amount of schoolwork and personal stuff was overwhelming and I got lost. This is not an excuse at any point, and I understand and accept my responsibilities at full. Since this assessment is way overdue, I fully accept that I would not get marks on it, finishing it was my personal achievement and challenge. I think after all I have done an okay job at designing this Database and documenting it. Thank you for your attention.*

## File system structure and logic

In this assessment I have included several files to help with navigation around the project folder. This should help a new-to-project person to catch up on hierarchy and logic of the project.

**Aleksei Stukalov Design Assignment Folder Hierarchy.txt**

*Includes a breakdown of the project folder structure in the markdown syntax.*

**Aleksei Stukalov Design Assignment Error List.docx**

*Includes a list of errors happen during a Data preparation process. Every error has a short description of the cause of the problem and a temporary solution I have implemented for it.*

**Aleksei Stukalov Design Assignment Data Dictionary.docx**

*Data Dictionary for the Database.*

**DATA**

*This folder contains all the Data that been used in this assessment, including the original Excel file* ***ARA DATA Wait Lists Feb 2021.xlsx****. Other files in the folder are row data exports from the Excel file and processed/prepared data for database, more about it in the* ***Notes*** *section of this document.*

**Diagrams**

*This folder contains Draw.IO ERDs and PDF exports. It is possible to see the progression of the ERD design by looking at the number the file name starts with, where* ***!01*** *is the very first design I made and* ***!11*** *is the very last.*

**Python**

*This folder contains python script I used to prepare data for insertion. Python file* ***DATA\_Shaper.py*** *itself has lots of commented out sections explaining the process and the logic of the conversion. Highly proud of that script, please look at it.*

**Queries**

*This folder contains all the queries for the database creation, data insertion and data presentation. In the root of the folder, you can see the progression of the queries I have been making, but it won’t show the challenges I have met and solutions I came up with, that information is below in the* ***Notes*** *section of this document. Queries folder also includes 2 subfolders:*

**DATABASE AND DATA**

*This folder contains two query files needed for Database creation and Data insertion, named appropriately* ***SysmexReferralDB.sql*** *and* ***Data Insertion.sql****.*

**TABLE QUERIES**

*This folder contains all the queries needed for the assessment’s questions. More on this in* ***Queries*** *section of this document.*

## Tasks

1. **A relational Schema (an extended ERD)** *using Visio, that includes entities, relationships, cardinality, attributes, and the PK indicated.*

For this task I used Draw.IO application to design ERDs and then exported them to PDF format. The current ERD solution called **!11 Assessment-1 ERD Simplified REWORK.pdf** is in the folder **Diagrams** of the project root. You can also see the iterations of the design by looking at the number at the start of the file name. This project also been maintained on GitHub, and you can see the progress of it in there. Link is at the bottom of **Tasks** section.

1. **A data dictionary o***f normalized tables (fields, data type, size, constrains)*

Data dictionary is in the DOCX file named **Aleksei Stukalov Design Assignment Data Dictionary.docx** and is in the root folder of the project.

1. **A report on the design process** *and the design issues encountered and how you chose to resolve these e.g., choice of entities, relationships, choice of keys, extent of normalization, multi-valued or composite attributes etc. Comment on any optional, mandatory, recursive, weak, subtype/supertype and composite entities.*

The way I did this task is by making a Notes section in this document and including all the notes I have had during the design and development of this project. It includes points from the task description so as other notes I felt like worth adding to it.

1. **Maintain and provide evidence of version control** *throughout the assignment.*

Few folders in this project contain iterations of the design, which I was labeling by adding a number at the start of the file name, where the lowest order number is the very first attempt of the design and the highest number would represent the latest and current solution I have came up with. On top of that I have been maintaining GitHub repository of this project, which you can access [**by this link**](https://github.com/yUtopist/DE103-Assessment-1). Repository is in public access and has all the changes shown. The link to the repository is also added at the bottom of **Tasks** section for the ease of find.

1. **Tables and relationships created in SQL Server** *– with sample data entered and answers for queries.*

Main part of this task is solved in 2 Query file for Database and Tables structure creation and Data insertion. Working Queries can be found in the folder **Queries > DATABASE AND DATA > SysmexReferralDB.sql** needs to be executed first because it is an initial query for Database and Tables creation, and then **Data Insertion.sql** can be initiated to insert prepared Data to Database. Keep it in mind that these two queries only need to be run ones to create the Database and insert Data to it, from then on other queries should be used to display Data.

The task includes five subtasks for Sysmex Hospital Data Queries to display certain data and all the Queries are in the folder **Queries > TABLE QUERIES** and each of the query contain detailed comments inside.

* 1. **How many people have been referred for cardiothoracic?**

**Cardiothoracic\_patients.sql**

* 1. **What is the average time taken (in days) to see a Surgeon by Department?**

**Average\_Time\_By\_Surgeon\_and\_Department.sql**

*NOTE: I couldn’t understand what exactly is needed to be displayed – average time takes to see a specific surgeon or all the surgeons in each department, so I made a query for both possibilities.*

* 1. **Who has each Surgeon had on their list and how long have they been waiting, or did they wait?**

**Patients\_Days\_Waited.sql**

*NOTE: Since there are entries of patient referrals with no FSA (First Specialist Appointment Date) I made a separate query to display these.*

* 1. **Assuming that all patients under 18 need to be seen by Pediatric Surgery, are there any patients who need to be reassigned?**

**Underage\_patients\_to\_reassign.sql**

*NOTE: Initial data has few errors and misspells, which are supposedly fixed by this point. It is important since it would change the outcome of this query.*

* 1. **What percentage of patient were seen within the target of 75 days by department?**

**Seen\_In\_75\_Days\_By\_Department.sql**

Link to the project repository: <https://github.com/yUtopist/DE103-Assessment-1>

## Notes

**Data preparation**

Although it is possible to create a list of unique Departments and assign unique Primary Keys to them in SQL Server, I prefer to do it with python, since it is still a Data preparation stage, and I would need to change data formats in python anyway, so I just wrote a simple script to convert the Data to desire shape.

All the raw Data is saved in correspondent files with file’s name representing what data the file contains and shaped data has an exclamation mark afront of the name, for the ease of read.

**Patient NHI Duplication, Date of Birth error**

Raw data in the excel sheet has several data imperfections and errors: duplication of patient’s NHI, where different patients having the same NHI number, which cannot be the case. This error was resolved with just manually changing all the duplicates of NHI number. In my python script I have made it so new NHI starts with the space bar, followed by string ‘err’ and then followed by three digits which are an order number of the error. So duplicate NHI numbers become to be ‘ err001’, ‘ err002’ and so on.

Additionally, some of the referral entries have had day of birth set up incorrectly, I have changed those the way I felt was the most accurate, but I also kept the track of these entries in the file **Error List.docx**.

**ERD design**

Current solution has following entities:

**Patient -** includes patient information such as their name, NHI, day of birth and gender. Since NHI is unique for each patient, it is used as a Primary Key.

**Department –** a list of unique departments with their corresponding ID.

**Surgeon** - surgeon information in the clinic - their name, the ID of department they are working at and their ID.

**Referrer** - referrer name, type, and ID.

**Referral** - includes every referral that has been sent to the clinic. Every wanted attribute that could be referenced with Foreign Key was done that way – Referrer, Patient and Surgeon.

**Aleksei Stukalov Design Assignment Data Dictionary.docx** has detailed information on relationship between entities.

Final Data Display Query named **! Referrals.sql** includes almost every attribute (column) from the original Excel file, and exclude the very last unnamed column, which is corresponded to patient NHI and doesn’t seem to have any value of information. Apart of all the original columns the Medial Staff would like to see two additional fields (Patient Age at Referral and Day Waited from Referral Date to FSA) which are **Derived Attributes** and do not need to be physically stored in the database. Also, one column from the original Excel file (Year-Month) can be a derived attribute, which is the case in my solution.

**Referrer Type is fluid**

Even though referrer type swap for Hilary Trump, Linda Moore and Ralph Smith are most likely a data insertion error, there is a possibility that referrer could change the place of work from other clinic to Sysmex clinic and therefore would change its type. From there we would have a problem with this database design - in the record referral going to have a current referrer type and not the type of them on the moment of record creation. The proper way, in my opinion, would be to have an attribute on Referral entity which corresponds to referrer type on the moment of record creation. but in this database design I would not develop it this way, instead I would assume that the referrer type swap was a human mistake and should not be the case.

**Underaged patient’s reassignment**

Several patients under age of 18 supposed to have assigned department as Pediatric, but have another departments instead. Since I do not have enough information on which surgeon they should to be assigned – I have made a Queary called **!Underage-patients-to-reassign.sql** which will display a list of these patients so someone with more information could make a change.

**FSA date not set**

Some of the FSA are not set, and I am assuming in the original Database assessment in 2016ish these would mean that Surgeon hasn’t seen a patient in around 100 days, but now in 2022 it gives me the answer that Surgeon hasn’t seen the patient in 2000 days, which wouldn’t be the case.