Database Design

InfoSafe

Seed Analytics

Team Name: FrAgile

Team Members: Christof Steyn
Chris Mittendorf
Karel Smit
Yané van der Westhuizen
Alistair Ross

Team Contact: fragile.cos301@gmail.com

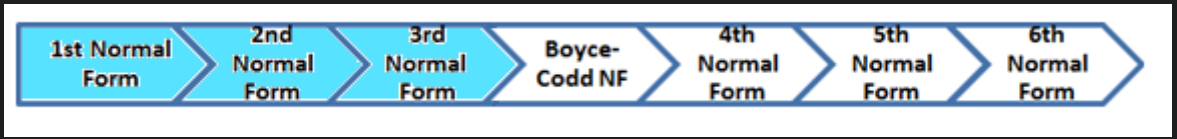
Table Of Contents

Introduction

It is important to understand what database normalisation is before delving into the design of the actual database. Normalisation is a design technique that reduces data redundancy and undesirable anomalies when creating, editing or deleting the contents of the database. The normalisation rules will divide large tables into smaller ones and then link them using relationships through primary and foreign keys. There are essentially seven database normal forms however the last four aren't as widely used and the first three rules are widely regarded as standard. We will also give further details about the relationships and the keys in the database.

The three rules we'll discuss here are:

- 1NF (First Normal Form)
- 2NF (Second Normal Form)
- 3NF (Third Normal Form)



Database Normal Forms

Keys in Relational Databases

Keys are used to identify records in a table uniquely. Usually it is an identifying feature like an ID or a specific number that is used to identify a row (or tuple) in a specific table. These keys can be used to identify duplicate information and help establish relationships between other tables in the database. The two keys we will use are Primary Keys and Foreign Keys.

Primary Keys are single column values used to identify a single database record uniquely. The Primary key has to be completely unique to a record and cannot be a null value. They should never or very rarely be changed and when a new record is created a new Primary Key should also be assigned to this record.

Foreign Keys are also used to uniquely identify records in a table. The foreign key is essentially a reference to a primary key and they can have the same value, they can be different or even null, unlike primary keys. The key in a table will reference the primary key to add more value to that record without the risk of running into redundancies or errors.

1NF (First Normal Form)

For a table to be in 1NF the following criteria need to be met:

- Each cell in a table should only contain a single value, no duplicates or multiple values are allowed
- Each record (or row) in a table should be unique, no duplicates allowed

2NF (Second Normal Form)

For a table to be in 2NF the following criteria need to be met:

- The table should meet all the criteria to be in 1NF
- Each table should have a single column for Primary Keys unique to each record of each table

3NF (Third Normal Form)

For a table to be in 3NF the following criteria need to be met:

- The table should meet all the criteria to be in 2NF
- The table should have no transitive functional dependencies, which means that a change in a non-key column might cause any other non-key column

Database Structure

Legend

Primary Key	*Foreign Key*	Non-key Attribute

Users

User_ID	*System_Role_ID*	Surname	Name	Email_Address

System Roles

System_Role_ID	Type

Hashed Passwords

User_ID	Hashed_Passwords

Roles

Role_ID	Role_Description

Data Scopes

Data_Scope_ID	Name	Description	Date_Captured	Status

Data Scope Roles

Data_Scope_ID	*User_ID*	*Role_ID*

Access Requests

Access_Request_ID	*User_ID*	*Data_Scope_ID*

Tasks / Compliance Matrix

Task_ID	Description	Status	Due_Date	Date_Completed

Assigned Tasks

Task_ID	*User_ID*	*Data_Scope_ID*

Documents

Document_ID	File

Emails

Email_ID	Email_File

Task Documents

Task_ID	*Document_ID*

Task Emails

Task_ID	*Email_ID*

Support Requests

Support_Request_ID	*User_ID*	Type	Description	Status	*Asset_ID*

Assigned Support Requests

Support_Request_ID	*User_ID*

Risks

Risk_ID	*Data_Scope_ID*	Impact_Rating	Description	Status

Assigned Risks

Risk_ID	*User_ID*

Assets

Asset_ID	Type	Description	Status	*Assigned_User_ID*

Asset Requests

Asset_Request_ID	*User_ID*	Reason	Desired_Date	Status