CITS2232 Databases Design Document

Dominic Cockman(20927611), Alastair Mory (21120848),   
Marcus Green (20947214)

# Design Reasoning

The driving force behind the decisions made in the design of the database was that of application, or how the database would function upon deployment to the real world, which in our case was as a staff and office management system for Centrelink. To this end, the database has been kept organised, with as few keys as possible. This is reflected by the UML diagram, in which there are only two primary keys: the Office Code and the Staff Identification Number. It is from these two keys that much of the other data can be determined; the Office Code will determine every other element in the Office table, as each piece of data is a unique set that is tied to the unique Office Codes. Similarly, the Staff ID determines each other element in the Staff table. On top of this, both the Office Code and the Staff ID form a multivalued dependency that allows us to determine the position that each staff member holds within each office, again a relationship that is unique to each combination of Staff ID and Office Code.

The relationships between each table are also based on perceived real life applications. Working up from the ‘first’ relationship, it can be seen that each Staff member can have multiple Staff Office Memberships, to reflect that a single Staff member could potentially work at different Centrelink offices during their employment. Similarly, each Office can be attached to multiple Staff Office Memberships, signifying that each office can employ multiple staff members. On top of this, our database has left room for the possibility that a Staff member could be an Office Manager at one Centrelink location, while still being a regular Staff member at another.

# Process of Normalisation

Normalising our databases was a fairly straightforward process, due to the simplicity of our table structure. The three tables we have finished with are representative of this; each primary key and its functional dependencies are in their own tables, and the only multivalued dependency in the database has its own table. This has reduced possible redundancies that would have come from having the Staff Office Membership data sets in either of the tables, as there is not the possibility of having multiple entries that determine the same thing.

The actual process involved determining which data sets were to be held unique, and which pieces of data would be intrinsically tied to them. As was obvious, the Office Code and Staff ID were created in order to facilitate this, both of which are determined independent of any other pieces of data in the database.

All this being said, there are potentially some other relationships that could have been given their own table. For example, it could be said that the Postcode in both the Office and Staff tables functionally determines the State, and thus should be given its own table. Similarly, the Street Address, Suburb, and Postcode could form a multivalued dependency that determines the Longitude and Latitude. However, we felt that these additional tables would not add any extra functionality that could not be facilitated by simply having them in the tables with which they were associated.

In summary, the design and functionality of our database was driven by both functionality and our perception of its application in the field. This has resulted in our three tables that we felt were necessary to convey all the information that would be needed to be extracted from our database application. We have taken steps to ensure that there is as little redundancy in the table as possible, hence normalising the database in the process, putting it, in our opinion, functionally in both Boyce-Codd Normal Form and 4th Normal Form.