The features of a relational database

**Features**

Entities

When data is captured, it is stored in entities. These can be tables containing data based the importance of a thing or an object. For example, in an organisation, they would have a database with tables containing data such as customer details like name, surname, address, mobile number etc.

Primary Keys

Primary keys are unique numbers given to each individual in a database (or if it is to do with stocking, a primary key would be given to each item on the stock list). They are given in order to identify a person or product, so for example when customers are entered into a database, as they are each given unique numbers, if a customer were to ring in and request information, they would be asked to give their primary key number to identify them (instead of having the organisation to search for the hundreds of people with the same names as them).

Foreign Keys

Foreign keys link tables together. A primary key would be used in one table, and if used in another table, as it will be identical, once linked together they will be known as foreign keys. For example, if another table is made using new and current primary keys, it is best that if one item (or customer) in the table remains with the primary key in which they were issued (which then makes it a foreign key in the new table as it would be linked to where it came from) so that no confusion could be made, and also so that a second key is not made for that same item.

Data Redundancy

This occurs when data that is inputted into a database has been entered in more than one table within the database, having the data being considered as duplicated data. It is something that most organisations tend to avoid, as it means that duplicated data would need to be updated all at the same time in order to all be accurate. For example in an organisation that makes and sells clothes, if data redundancy happens, it could result in loss of money, and can be time consuming to deal with as it means updating each and every duplicate. Data redundancy also wastes quite an amount of space.

**Attributes**Data TypesWhen creating tables in a database, users are given the option to add further description to what they are inserting into their cells, for example apart from field names and data types, you can choose from having text, number, date/time, currency, an AutoNumber, yes/no, OLE object, lookup wizard or a hyperlink. This can be done mostly to define exactly what is going to be inputted in each cell, for example by having the date/time selection; the date would need to be in this format: 00/00/0000. So if a user were to input 29th April 1999, it will be false. By having these data types, it saves memory.

Other Field Properties

Data that is to be inputted into fields should always be valid, or the database wouldn’t be reliable. In order to validate data that is inputted, an input data should be used, for example in a field that requires an address to be filled in, an address would require both numbers and letters to be filled into that field, so by inserting “LL99” into the input mask it means that the user can use both alphabetic characters and numeric character in order to type in their address, therefore making it valid.

One-to-One One-to-Many Many-to-Many

A one to one relationship is by having one table linked to another table in a database. One to many relationships would be if one table is linked to more than one tables in a database, and a many to many would be if many tables in a database are linked to many other tables in the database. For example, in a business which sells products, customers can order almost each item in a stock table, and with stock it can contain more than one fields in the database linked to one other table, making it a one to many relationship.

Normalisation

This is the process of having data in the database be organised by having it analysed and rearranged into tables. By arranging data into tables, you can define them by creating relationships between them. With normalisation it can avoid data redundancy and would take up less space in the database.