# Worksheet 3 Relational databases and normalisation

# 1. Write definitions for

# First Normal Form (1NF)

The data is atomic meaning it can not be broken down any further. There is a primary key and all the field names are unique. It contains no repeating attributes or groups of attributes.

# Second Normal Form (2NF)

There are no more partial dependencies in the entities. This means that the secondary keys are all dependent on the primary key, and not for example, one key is dependent on one primary key while another is on another primary key in a composite key. The database is already in first form.

# Third Normal Form (3NF)

There is no more transitive dependencies. This means none of the data is indirectly linked to the primary key via another key. The database is also already in 2nd form.

Nonkey dependencies/transitive dependencies

# 2. A car dealer has several different branches which each sell cars, and a database is being designed to hold data about the cars they sell and the salespeople who sell them.

# Each branch is identified by town. There is a maximum of one branch in each town

# Each make of car is identified by a unique model name

# Each model of car is made by only one manufacturer

# Each salesperson is identified by their SalesID. The number of each model of car that they sell (SalesVol) is recorded

# A first attempt at designing a table to hold the data has been made. The table, called CarSales, is shown below with some sample data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SalesID** | **Name** | **Branch** | **Model** | **SalesVol** | **Manufacturer** |
| S123 | Gerry | Norwich | Clio  C3 Picasso  Civic | 3  4  5 | Renault  Citroen  Honda |
| S555 | Shirley | Cromer | Juke  C4  Octavia | 1  2  4 | Nissan  Citroen  Skoda |
| S442 | Dave | Cromer | C3 Picasso  Octavia | 5  1 | Citroen  Skoda |

(a) Why is this table not in First Normal Form (1NF)?

Because the data in the database is not yet atomic. Like for example, in SalesVol, it can be broken down further since sales volume column has multiple sales for different models and by different manufacturers. It contains repeating groups of attributes (Model, SalesVol, Manafacturer columns)

(b) The data is split into two tables. Show the contents of the two tables

**Table: SalesPerson**

|  |  |  |
| --- | --- | --- |
| SalesID | Name | Branch |
| S123 | Gerry | Norwich |
| S555 | Shirley | Cromer |
| S442 | Dave | Cromer |

# 

|  |  |  |  |
| --- | --- | --- | --- |
| SalesID | Model | SalesVol | Manufacturer |
| S123 | Clio | 3 | Renault |
| S123 | C3 Picasso | 4 | Citroen |
| S123 | Civic | 5 | Honda |
| S555 | Juke | 1 | Nissan |
| S555 | C4 | 2 | Citroen |
| S555 | Octavia | 4 | Skoda |
| S442 | C3 Picasso | 5 | Citroen |
| S442 | Octavia | 1 | Skoda |

**Table: ProductSales**

(c) A relationship between the tables has been implemented. Explain how this has been done.

There is a one to many

(d) Explain why the ProductSales table is not in Third Normal Form (3NF)

Its not even in 2nd normal form yet, so it cant be in 3rd normal form since manufacturer is dependent on Model, but there is no 1 primary key, the primary key is a composite key, so its made up of salesID and Model since SalesID isn’t all unique by itself. Since manufacturer is only dependent on Model, its not even in 2nd form since it has partial dependencies.

(e) Draw an entity relationship diagram to show the entitities in the database in 3NF.

SalesPerson one to many ProductSales many to one Product

(f) Write the table definitions for the database in 3NF. Use the notation

*Tablename* (*keyfield, Attribute1, Attribute2,* …)

SalesPerson (SalesID, Name, Branch)

ProductSales (SalesID, Model, SalesVol)

Product (Model, Manafacturer)

(g) Identify the foreign key(s) in one of the tables.

SalesID