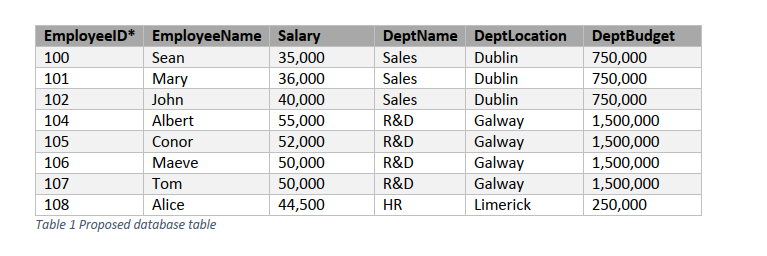
# 4.3 Normalisation.

This document contains analysis of the following database and my evaluation of its design.



## Reflexion of the purpose in the Database.

The purpose of the database is to store information on company employees and departments. The purpose is not reflected well in the current table as it does not include information on employee surnames, recruitment dates, home addresses, phone numbers, department phone number etc., and as such is somewhat incomplete.

## Organization of Tables.

The organisation of the current database is not well divided into subject-based tables, since a separate table should have been created for each related data set, thus minimizing data redundancy. Considering the information provided, the inclusion of two tables would have been more appropriate: an Employee data table (with a primary key as it is EmployeeID that cannot be NULL) and a Department data table (with a primary key DeptID that cannot be NULL) linked via a foreign key DeptID.

Below is an example of the suggested division:

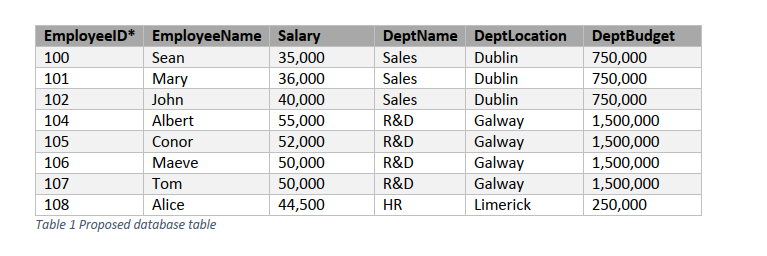
**Employee Table**

|  |  |  |  |
| --- | --- | --- | --- |
| EmployeeID\* | EmployeeName | Salary | DeptID |
| 100 | Sean | 35,000 | 1 |
| 101 | Mary | 36,000 | 1 |
| 102 | John | 40,000 | 1 |
| 104 | Albert | 55,000 | 3 |
| 105 | Conor | 52,000 | 3 |
| 106 | Maeve | 50,000 | 3 |
| 107 | Tom | 50,000 | 3 |
| 108 | Alice | 44,500 | 2 |

**Department Table**

|  |  |  |  |
| --- | --- | --- | --- |
| DeptID\* | DeptName | DeptLocation | DeptBudget |
| 1 | Sales | Dublin | 750,000 |
| 2 | HR | Limerick | 250,000 |
| 3 | R&D | Galway | 1,500,000 |

This will exclude duplication of Department Name and Department Location.



Additionally, separation of data into two tables will save time when an update is required. For example, if a new employee has joined the company, it will be necessary to add an EmployeeID, Employee Name, Salary and DeptID – the rest of the information will be referenced to via Foreign Key in the Department Table (where EmploeeID is set as Auto Incremented it will be generated automatically).

On the other hand, assuming a department called Production is introduced in Cork with a budget of €900,000 and does not yet have any employees. The current table has the primary key, EmployeeID, which cannot be omitted; the result would be the inaccurate entry line below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EmployeeID\* | EmployeeName | Salary | DeptName | DeptLocation | DeptBudget |
| 109 | NULL | NULL | Production | Cork | 900,000 |

If provided with more information on, for example, the from/to dates of employees’ current salary, I would suggest creating another table called Salaries with the primary keys, EmployeeID and FromDate, and the foreign key, EmployeeID, to reference the Employee Table.

**Employee Table**

|  |  |  |
| --- | --- | --- |
| EmployeeID\* | EmployeeName | DeptID |
| 100 | Sean | 1 |
| 101 | Mary | 1 |
| 102 | John | 1 |
| 104 | Albert | 3 |
| 105 | Conor | 3 |
| 106 | Maeve | 3 |
| 107 | Tom | 3 |
| 108 | Alice | 2 |

**Department Table**

|  |  |  |  |
| --- | --- | --- | --- |
| DeptID\* | DeptName | DeptLocation | DeptBudget |
| 1 | Sales | Dublin | 750,000 |
| 2 | HR | Limerick | 250,000 |
| 3 | R&D | Galway | 1,500,000 |

**Salaries Table**

|  |  |  |  |
| --- | --- | --- | --- |
| EmployeeID\* | FromDate | ToDate | Salary |
| 100 | XXXX-XX-XX | XXXX-XX-XX | 35,000 |
| 101 | XXXX-XX-XX | XXXX-XX-XX | 36,000 |
| 102 | XXXX-XX-XX | XXXX-XX-XX | 40,000 |
| 104 | XXXX-XX-XX | XXXX-XX-XX | 55,000 |
| 105 | XXXX-XX-XX | XXXX-XX-XX | 52,000 |
| 106 | XXXX-XX-XX | XXXX-XX-XX | 50,000 |
| 107 | XXXX-XX-XX | XXXX-XX-XX | 50,000 |
| 108 | XXXX-XX-XX | XXXX-XX-XX | 44,500 |

The above example shows that the inclusion of separate subject-based tables will better facilitate updates as the database grows.

## Summary

To summarize, the above analysis indicates that the database is not optimally designed. Although it is acceptable in practice, it is difficult and time-consuming to maintain as it is not separated into logical blocks.