4.3 Normalisation

4.3.1 Database Design

- Proposed database table (entity/relation) is similar in style to a spreadsheet - each row (record/tuple) contains data pertaining (relating) to a single item. In this **spreadsheet-style** database table each row represents an individual employee (EmployeeID - EmployeeName - Salary) along with his/her department (DeptName - DeptLocation) - in addition the column (attribute/field) **DeptBudget** characteristic of each department location is also recorded.

EmployeeID*	EmployeeName	Salary	DeptName	DeptLocation	DeptBudget
100	Sean	35,000	Sales	Dublin	750,000
101	Mary	36,000	Sales	Dublin	750,000
102	John	40,000	Sales	Dublin	750,000
104	Albert	55,000	R&D	Galway	1,500,000
105	Conor	52,000	R&D	Galway	1,500,000
106	Maeve	50,000	R&D	Galway	1,500,000
107	Tom	50,000	R&D	Galway	1,500,000
108	Alice	44,500	HR	Limerick	250,000

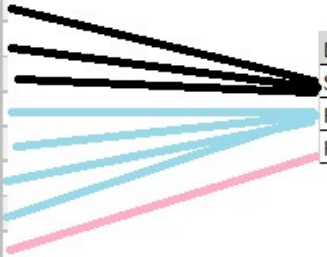
- Columns relating to the company's departments (DeptName - DeptLocation - DeptBudget) have a lot of duplication - each time an employee is added to the database will require information to be re-entered into the table. This duplication of information is error-prone - each new employee added results in details of the assigned department to be **duplicated** into the table. This database table is not well designed given this sort of duplication of information should not happen. If the company's R&D department budget reduced it would mean having this detail changed numerous times. Not having this updated on some employee records could have consequences for the company.

DeptName	DeptLocation	DeptBudget
Sales	Dublin	750,000
Sales	Dublin	750,000
Sales	Dublin	750,000
R&D	Galway	1,500,000
R&D	Galway	1,500,000
R&D	Galway	1,500,000
R&D	Galway	1,500,000
HR	Limerick	250,000

- To avoid such redundant data - updating the budget for the R&D department in one place and have each employee (EmployeeID) just **referencing** the name of the company department (DeptName) rather than duplicating would be the best option.

- Breaking the table down into two smaller related tables makes the data in the database easier to update and less prone to anomalies. Proposed database table broken down into smaller interlinked tables via **normalisation**. With less columns as opposed to one large table with many columns lends to the definition of a **database** as a **collection** of **related data** organised so that the **data** can be **easily accessed**, **managed** and **updated**.

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



DeptName*	DeptLocation	DeptBudget
Sales	Dublin	750000
R&D	Galway	1500000
HR	Limerick	250000

- The proposed database table design has an increased likeihood of some rows not updated or overlooked resulting in an **update anomaly**. In addition if it was decided to shut down a company department then deleting a department would result in the deletion of all related employees (**deletion anomaly**).

- **SUMMARY**: Normalised version of the table would mean splitting the proposed table into two tables - Employee - Department. The Employee table (column - EmployeeID - Primary Key) would hold a link to the Department table (column - DeptName - Foreign Key). Any employee associated with a department just references the DeptName.

END