13 Databases and Software Development

13.1 Entity Relationship Modelling

An **entity** is a category of object about which data is to be recorded.

- Each entity in a database system has **attributes**.
- Each entity needs an **entity identifier** which uniquely identifies the entity.
- The **primary key** is the entity identifier in a **relational database**.

An **entity description** is written using the format.

EntityName (PrimaryKey, Attribute1, Attribute2)

Two entities are said to be **related** if they are linked in some way.

The **degrees of relationship** between two entities can be

- One-to-one
- One-to-many
- Many-to-many

An **entity relationship diagram** is a diagrammatic way of representing the relationship between the entities in a database. Shows the

- Degree of relationship.
- Name of the relationship.

Relational Database

In a relational database, a separate **table** is created for each entity identified in the system.

- Where a relationship exists between entities, an extra field called a **foreign key** links the two tables.
- A foreign key is an attribute that creates a join between two tables - it is the attribute that is **common to both tables**.
- The primary key in one table is the foreign key in the table to which it is linked.

Many-to-many Relationships

Tables cannot be linked directly in a many-to-many relationship. Instead, create a **link table** with two foreign keys, each linking to one of the two tables. The two foreign keys also act as the primary key of the table.

A primary key which consists of more than one attribute is called a **composite primary key**.

13.2 Relational Databases and Normalisation

A **relational database** is a collection of tables which relationships are modelled by shared attributes.

Tables can be linked through the use of **common attributes**. This attribute must be a primary key of one of the tables, and is know as a **foreign key** in the second table.

Normalisation

Normalisation is a process used to come up with the **best possible design** for a relational database. Tables are organised in a way that

- No data is **unnecessarily duplicated**.
- Data is **consistent** throughout the database. This means anomalies will not arise when the data is inserted, amended or deleted.

  Consistency should be an automatic consequence of no holding any duplicated data.
- The structure of each table is enough to allow you to enter **as many or as few items** as required.
- The structure should enable a sure to make all kinds of **complex queries** relating data from different tables.

1. A table is in **first normal form** if it contains **no repeating attribute** or group of attributes.
2. A table is in **second normal form** if it is in first normal form and contains **no partial dependencies**.

   A partial dependency is where one or more attributes **depends on only part of the primary key**, which can only occur if the primary key is a **composite key**.
3. A table is in **third normal form** if it is in second normal form and contains **no non-key dependencies**.

   A non-key dependency is one where the value of an attribute is determined by the value of another attribute which is not part of the key.

Advantages of Normalisation

- Easier to **maintain and change**.

Data integrity is maintained since there is no unnecessary duplication of data, it will also be impossible to reference a non-existing record on another table.

- **Faster** sorting and searching, as normalisation produce **smaller tables with fewer fields**, searching is faster because less data is involved.

  Holding data once saves storage space.

- A normalised data base with **correctly defined relationships** between tables will not allow records in a table on the 'one' side of a one-to-many relationship to be deleted accidentally.