

Databases

Contents

Definitions	2
Diagrams	3

Definitions

An **entity** is a category of object, person, event, or thing about which data needs to be recorded.

A simple (flat file) database is a single file containing information on a single entity.

An entity maps to a table. A table contains many records, each holding information about a particular instance of an entity. For example, a `tblCustomer` table will contain many records, each representing a single customer.

Each record of an entity table needs an identifier to uniquely identify it. The identifier is known as the **primary key**.

A composite primary key is a combination of keys in a record, where the *combination* uniquely identifies the record rather than having a separate ID key.

Most databases automatically index the primary key field so that any particular record can be found quickly.

If another key field also needs to be searched for, then it can be called a **secondary key**, and this field will be indexed.

Entities can be linked by one-to-one, one-to-many, or many-to-many relationships. See the diagrams.

To create a relationship between entities, the ‘many’ side must have a **foreign key**, which matches the primary key in another table. Say a customer can have many subscriptions. Each record must have a fixed number of fields, so they can’t have an arbitrary number of foreign keys to link to subscriptions. Therefore, each subscription record should have a foreign key to point to a customer.

Referential integrity means that every foreign key must reference a existent record in a related table that actually exists. Dangling foreign keys are forbidden.

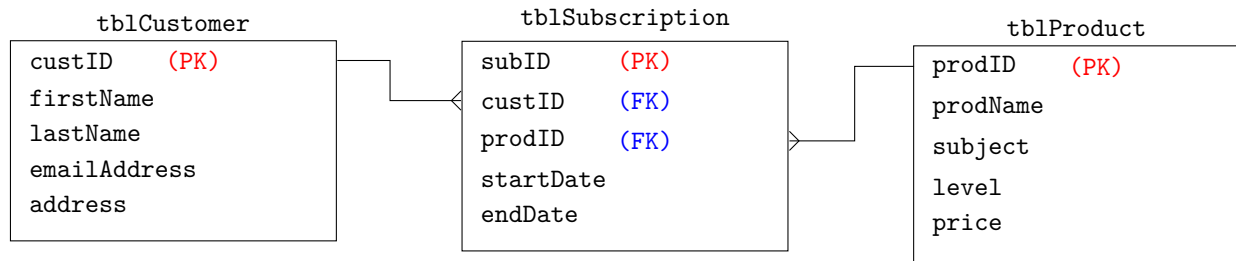
As a canonical example, a customer can have many orders; an order can have many orderLines (an order is like a shopping cart, and an orderLine is like a single order with product, quantity, price); a product can appear in many orderLines.

Many-to-many relationships cause problems with unknowable column numbers. This can be fixed with a **linking table**. Consider a fitness center - members can take many classes, and classes can have many members. Instead, have an enrolment table in the middle. A member can have many enrolments, and a class can be part of many enrolments. Each enrolment record is a single link between a member and a class.

Diagrams

Boxes contain entities, or the fields of a record in that table. Relationship lines have a single line connection to the box for a *one* connection, and a crow's foot connection for *many*.

In this example, there is an online revision service which provides textbooks. A customer can subscribe to different tiers. A customer has personal details; a product is a book with a name, subject, level, and price; and a subscription links these two together with foreign keys for each, as well as a start and end date.



The **tblCustomer** table has personal details and a unique ID, which is its primary key. The **tblProduct** table has details about the product and its own unique ID primary key. The **tblSubscription** table has a unique ID primary key, a start and end date, and two foreign keys. Each subscription references the customer that subscribes to it and the product they subscribe to, in the form of two primary keys.