Module 2-5

Database Design

DQL vs DML vs DDL

The SQL statements we have seen so far fall into a number of different categories:

- Data Query Language (**DQL**): SELECT
- Data Manipulation Language (**DML**): INSERT, UPDATE, DELETE
- Data Definition Language (**DDL**): CREATE, ALTER

The focus of this lecture will be DDL statements with appropriate constraints.

Normal Forms

Before a single CREATE statement is run, the tables and their relationships need to be well thought out.

One design methodology commonly used is Boyce-Codd Normal Form (BCNF).

Normal Forms: 3NF

There are several levels of “normal form” compliance, but generally the third normal form is good enough for 99% of all situations.

An informal intuitive definition of 3NF is as follows:

There are no fields in a table that are not directly determined by the values of the primary key.

Normal Forms: 3NF Example

Suppose we have the following table:

InvoiceNumber (PK)	**InvoiceDate**	**Inventory ID**	**Inventory Description**
1000	10/1/2019	45	Hammer
1001	10/3/2019	28	Nails
1002	10/3/2019	17	Screwdriver
1003	10/4/2019	45	Hammer

Some questions to consider:

- Is an invoice date directly related to an invoiceNumber? $\rightarrow$ Yes
- Is an inventory description directly related to an invoiceNumber? $\rightarrow$ No

Normal Forms: 3NF Example

Suppose we need a Spanish version of this database, and we need to value to show *Martillo* instead of Hammer. This would entail an UPDATE statement that targets 2 rows.

InvoiceNumber (PK)	InvoiceDate	Inventory ID	Inventory Description
1000	**10/1/2019**	**45**	**Martillo**
1001	10/3/2019	28	Nails
1002	10/3/2019	17	Screwdriver
1003	**10/4/2019**	**45**	**Martillo**

Normal Forms: 3NF Example

In this situation, we could have split up the data into 2 tables, thus we end up with a less risky query, affecting only 1 row:

InvoiceNumber (PK)	InvoiceDate	Inventory ID
1000	**10/1/2019**	**45**
1001	10/3/2019	28
1002	10/3/2019	17
1003	**10/4/2019**	**45**

Inventory ID (pk)	Description
28	Nails
17	Screwdriver
45	Martillo

Many to Many relationships

Generally speaking, when there are 2 entities for which there is a “many to many” relationship, we will end up with 3 tables when considering 3NF as part of our design.

Many to Many relationships Example

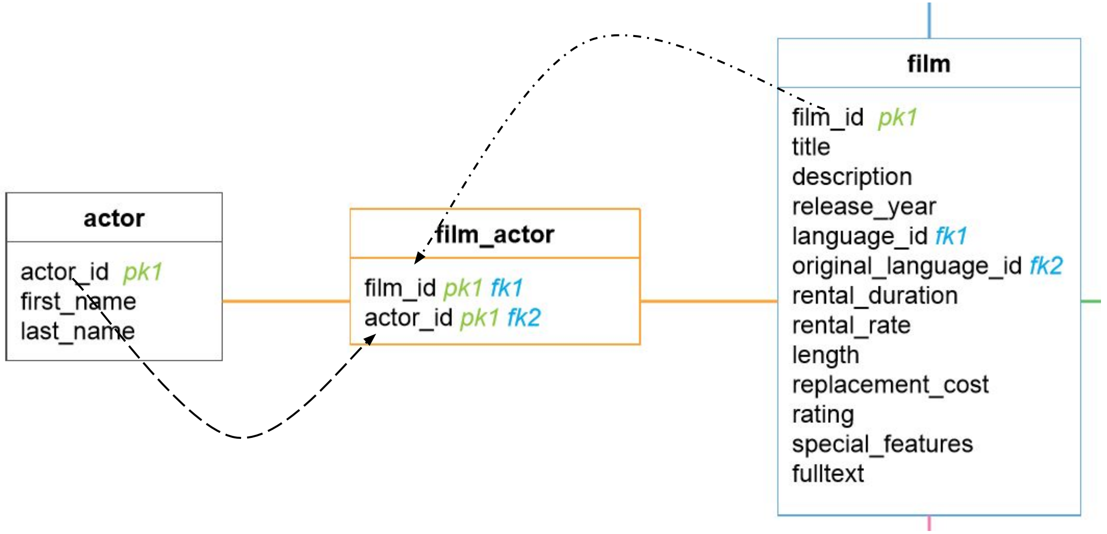
Consider the dvdstore example:

- An actor can be a cast member of several movies.
    - A movie can have several actors.

This is a “many to many” relationship.

Many to Many relationships Example

Consequently we end up with three tables to describe this relationship:



For this relationship to work we have defined two foreign keys in the film_actor table, the primary keys of each of the other two tables.