Lesson 21: Inner joins

Different parts of information can be stored in different tables, and in order to put them together, we use `INNER JOIN ... ON`. Joining tables gets to the core of SQL functionality, but it can get very complicated. We will start with a simple example, and will start with an `INNER JOIN`.

As you can see below, there are 3 tables:
character: Each character is a row and is represented by a unique identifier (*id*), e.g. 1 is Doogie Howser
character_tv_show: For each character, which show is he/she in?
character_actor: For each character, who is the actor?

See that in **character_tv_show**, instead of storing both the character and TV show names (e.g. Willow Rosenberg and Buffy the Vampire Slayer), it stores the *character_id* as a substitute for the character name. This *character_id* refers to the matching *id* row from the **character** table.

This is done so data is not duplicated. For example, if the name of a character were to change, you would only have to change the name of the character in one row.

This allows us to "join" the tables together "on" that reference/common column.

To get each character name with his/her TV show name, we can write

```
SELECT character.name, character_tv_show.tv_show_name
FROM character
INNER JOIN character_tv_show
ON character.id = character_tv_show.character_id;
```

This puts together every row in **character** with the corresponding row in **character_tv_show**, or vice versa.

Note:
- We use the syntax **table_name**.*column_name*. If we only used *column_name*, SQL might incorrectly assume which table it is coming from.
- The example query above is written over multiple lines for readability, but that does not affect the query.

Can you use an inner join to pair each character name with the actor who plays them? Select the columns: **character**.*name*, **character_actor**.*actor_name*

Congrats! That is correct!
Next Lesson

SQL:

```
SELECT character.name,
character_actor.actor_name
FROM character
INNER JOIN character_actor
ON character.id = character_actor.character_id;
```

Run SQL

Result:

name	actor_name
Doogie Howser	Neil Patrick Harris
Barney Stinson	Neil Patrick Harris
Lily Aldrin	Alyson Hannigan
Willow Rosenberg	Alyson Hannigan

Current tables:

character

id	name
1	Doogie Howser
2	Barney Stinson
3	Lily Aldrin
4	Willow Rosenberg

character_tv_show

id	character_id	tv_show_name
1	4	Buffy the Vampire Slayer
2	3	How I Met Your Mother
3	2	How I Met Your Mother
4	1	Doogie Howser, M.D.

character_actor

id	character_id	actor_name
1	4	Alyson Hannigan
2	3	Alyson Hannigan
3	2	Neil Patrick Harris
4	1	Neil Patrick Harris

Expected Result:

name	actor_name
Doogie Howser	Neil Patrick Harris
Barney Stinson	Neil Patrick Harris
Lily Aldrin	Alyson Hannigan
Willow Rosenberg	Alyson Hannigan

Learn SQL on your own

This tutorial provides you with easy to understand SQL instructions and allows you to practice while you are learning, using an online SQL interpreter. To learn by practicing your SQL commands, seeing immediate results. You will be able to perform selects, inserts, updates, deletes, and drops on your tables. Note: This tutorial uses the SQLite database engine. The different variants of SQL use slightly different syntax.

If you're already familar with the basics of SQL, you can still use this as a refresher, and practice some SQL statements.

How long does it take to learn SQL? How hard is it to learn SQL? Is easy to learn?

Its not very hard and you can learn it very quickly. Follow this interactive online SQL training for beginners (and for FREE) and in no time you will learn all the necessary knowledge to start working and to be confident to say you know SQL in a job interview.

Are SQL queries/syntax case sensitive?

The SQL Keywords are case-insensitive (SELECT, FROM, WHERE, etc), but are often written in all caps. However in some setups table and column names are case-sensitive.

What is SQL?

SQL stands for Structured Query Language. SQL is used to communicate with a database and SQL is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc.

Easy to learn standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that you need to do with a database. This tutorial will provide you with the instruction on the basics of each of these commands as well as allow you to put them to practice using the SQL Interpreter.

What Can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

The Most Important SQL Commands

- **SELECT** - extracts data from a database
- **UPDATE** - updates data in a database
- **DELETE** - deletes data from a database
- **INSERT INTO** - inserts new data into a database
- **CREATE DATABASE** - creates a new database
- **ALTER DATABASE** - modifies a database
- **CREATE TABLE** - creates a new table
- **ALTER TABLE** - modifies a table
- **DROP TABLE** - deletes a table
- **CREATE INDEX** - creates an index (search key)
- **DROP INDEX** - deletes an index

Why should you learn SQL?

SQL is an incredibly important and valuable skill employers desire. You can earn really good money, SQL programmers are in high demand. As organizations seek to do more with their data, they will need more individuals with the skills to access and analyze that data. SQL is the skill that enables you to do just that.

Learning SQL will allow you to mine data with greater efficiency, as SQL queries can be easily saved and re-used at any point in time. You can do data manipulation, combine data from multiple sources and manage large pools of data. And you will not have to deal with Excel crashing anymore.

List of databases used by Internet's biggest websites

- *The king of scalability, Google, uses **BigTable.***
- *Facebook uses **Hive** (Data warehouse for Hadoop, supports tables and a variant of SQL called hiveQL) and **Cassandra** (Multi-dimensional, distributed key-value store) for Facebook's private messaging.*
- *Yahoo uses modified **PostgreSQL.***
- *YouTube uses **MySQL** but they are moving to Google's **BigTable**.*
- *Myspace uses **SQL Server**.*
- *Twitter and Wikipedia uses **MySQL**.*
- *Microsoft uses **SQL Server,** which is very obvious.*
- *Flickr uses **MySQL**.*

Facebook, Google, LinkedIn, and Twitter all use MySQL for at least some of their data management. As they all contribute some of their MySQL customizations to a project called WebScaleSQL.

Will SQL become obsolete?

My guess is not for a very, very long time, if ever. Business and specially small business will continue to organize data in a relational manner regardless of the underlying data storage and processing technology.