In this notebook we'll go over a brief introduction to the structure of the Sakila Database and setting up SQL in your Python Environment.

####DISCLAIMER:

There are many ways to browse through a SQL database, throughout this Appendix we are only going to be focusing on learning about SQL queries using a combination of SQLite,Python,pandas, and SQLAlchemy. Please note that this is a pretty specific way of operating with a SQL Database, and may or may not fit other general needs. The primary goal of this section is to teach you how to use SQL queries to grab information and set it as a pandas DataFrame. We will not be going over more general topics of relational databases, MySQL, or using a SQL console directly.

To fully understand the content of this Appendix, I suggest you complete the course up to at least Lecture 28, although I really recommend completing up to Lecture 46 to get the most out of this Appendix!

Great, let's begin!

Step 1: Download SQL Alchemy

To start this appendix, download SQLAlchemy. You can do this by either downloading it here (http://www.sqlalchemy.org/download.html)

Or - by typing pip install sqlalchemy in your command line.

Or - by typing conda install sqlalchemy if you are using the Anaconda installation of Python. *(recommended)*

Step 2: Download SQLite Broswer

Next up we will download a sql browser. We will be using SQLite Browser because it is lightweight and free to use. There are many alternatives you can use, check out a list of 10 free ones here (http://www.gfi.com/blog/top-10-free-database-tools-for-sys-admins/)

Download SQLite Browser here: http://sqlitebrowser.org/ (http://sqlitebrowser.org/)

Step 3: Download the sakila Database

You can download the fully constructed database here (https://www.dropbox.com/s/t049qmjzycrakro/sakila.db?dl=0)

Or - you can download the .sql file to construct the database yourself: http://dev.mysql.com/doc/index-other.html (http://dev.mysql.com/doc/index-other.html) Then use SQLite Browser to construct the database by running the .sql

Either way, make sure to save it in the same directory as your iPython notebooks, or remember the file path for later so we can tell pandas exactly where to look for it.

All done! Now let's look at the database before diving into how to work with it in Python.

Check out the database either by opening it up using SQLite Browser or by checking out the diagram at this link: Diagram (http://www.dbquanti.eu/css/images/database.png)

I've posted it below as well inside this notebook, but fair warning, the picture is huge!

In [11]:

```
# Note: The picture is really big, I suggest you check out the link directly!

from IPython.display import Image
Image(url='http://www.dbquanti.eu/css/images/database.png')
```

Out[11]:

Now that we have seen an overview of what the database looks like, let's go ahead and learn how to communicate with it with Python and pandas.

Python comes with SQLite3, which provides a lightweight disk-based database that doesn't require a seperate server process. It's useful to prototyp with SQLite and then port the code to a larger database system, like MySQL. Python comes with a pretty awesome module to connect to a SQL database with SQLite. The module is SQLite3, let's go ahead and import it (and pandas as well).

In [13]:

```
# imports!
import sqlite3
import pandas as pd
```

To use the module, you must first create a Connection object that represents the database. If the database name already exists SQLite3 will automatically connect to it, if it does not exsist, SQLite3 will automatically create.

For experienced users: You can also supply the special name :memory: to create a database in RAM.

Let's make the connection!

In [15]:

```
# Connect to the database (again, downloaded from here: https://www.dropbox.com/s
con = sqlite3.connect("sakila.db")
```

Now we can run a basic SQL query, pass it with pandas, and display the output as a DataFrame! Don't worry if you don't understand the query completely yet, this is just a usage example for connecting to the database, other lectures will dive deeper into SQL queries.

In [16]:

```
# Set SQL query as a comment
sql_query = ''' SELECT * FROM customer '''

# Use pandas to pass sql query using connection form SQLite3
df = pd.read_sql(sql_query, con)

# Show the resulting DataFrame
df
```

Out[16]:

	customer_id	store_id	first_name	last_name	email
0	1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.or
1	2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacust
2	3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustom
3	4	2	BARBARA	JONES	BARBARA.JONES@sakilacustom
4	5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacust
5	6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustom
6	7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.
7	8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustome
8	9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacust
9	10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacusto
10	11	2	LISA	ANDERSON	LISA.ANDERSON@sakilacustom
11	12	1	NANCY	THOMAS	NANCY.THOMAS@sakilacustome
12	13	2	KAREN	JACKSON	KAREN.JACKSON@sakilacustom
13	14	2	BETTY	WHITE	BETTY.WHITE@sakilacustomer.c
14	15	1	HELEN	HARRIS	HELEN.HARRIS@sakilacustomer
15	16	2	SANDRA	MARTIN	SANDRA.MARTIN@sakilacustom

	customer_id	store_id	first_name	last_name	email
16	17	1	DONNA	THOMPSON	DONNA.THOMPSON@sakilacust
17	18	2	CAROL	GARCIA	CAROL.GARCIA@sakilacustome
18	19	1	RUTH	MARTINEZ	RUTH.MARTINEZ@sakilacustom

Congratulations! You just passed a SQL Query using pandas and Python! You're amazing! Subsequent lectures will go further into how to query with SQL, but if you already know SQL, you're good to go!