Introduction to Databases And Querying

What is a database?

A database is an organized collection of data that can be accessed, managed, and updated.

Some examples you have probably encountered:

- A shopping site (like Amazon) that has data representing products, users, purchases, and so on.
- A video site (like Netflix or YouTube) that might keep track of what people watch, let them create favorites or lists to watch, and maybe even recommend things based on how they rate what they've watched.
- Something like a Wordpress blog or even Facebook that would store data for posts, comments, and so on.

A Few More Tidbits…

- Storing data so it can be used later is called **persisting** data.

- Most databases provide intelligent ways to work with data to provide reports of all sorts and add and modify data as needed.

Some Reasons to Use a Database

- Easily support storing large number of records

- Central storage

- Support structured query syntax to retrieve data

- Enforces consistency and integrity of data

- Guarantee data type

Relational Databases

- **<u>Relational databases</u>** allow data to be accessed and reassembled in many different ways by **<u>Relating</u>** the data rather than reassembling it.

- Real world problems are modeled into **<u>Entities</u>** such as users, posts comments, products, etc.

TABLE (Entity)

COLUMN (Attribute)

ROW
(Record or
Instance)

Product

ID	BRAND	MODEL	DESCRIPTION	PRICE
1453672	DELL	Vostro 5590	Great laptop.	679.00
1453673	APPLE	MacBook Pro	13" 2020 2 Thunderbold 3 Ports	1299.00

Database Management Systems (DBMS)

A DBMS helps manage a database. We will be working with a Relational Database Management System (RDBMS).

Provides for 4 basic functions:
- Data definition
- Data storage
- Data retrieval
- Administration

Common Database Engines

- Oracle
- SQL Server
- MS Access
- PostgreSQL
- DB2
- MySQL

Database Columns

- Each column in a database has a data type.

- ANSI SQL (Structured Query Language):
  The standard for SQL that is defined by ANSI (American National Standards Institute).

  https://en.wikipedia.org/wiki/SQL#SQL_data_types

Database Columns

ANSI SQL defines many common data types:.

Examples:
- char, varchar, nvarchar
- int, decimal, bigint
- boolean/bit
- Datetime
- More info: https://www.w3resource.com/sql/data-type.php
- Postgres Data Types: https://www.postgresql.org/docs/11/datatype-character.html

Database Table Example

Each table column has a name and data type.
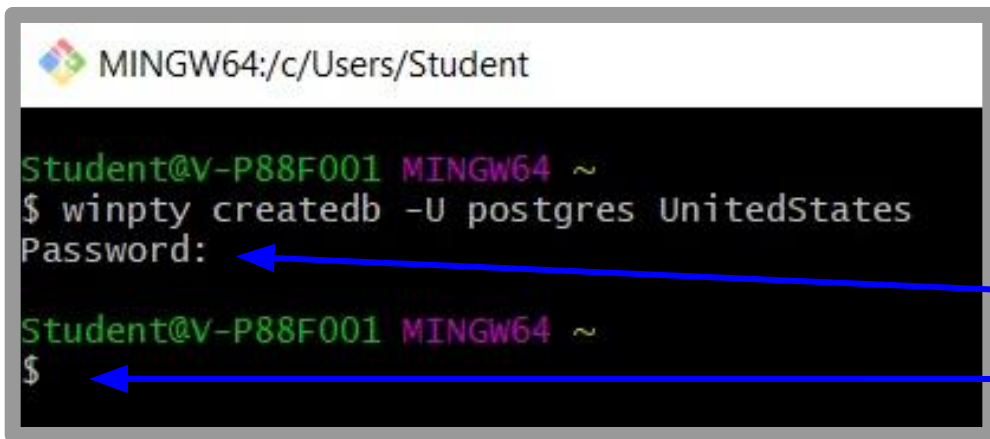
id	name	countrycode	district	population
1	Kabul	AFG	Kabol	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200
6	Rotterdam	NLD	Zuid-Holland	593321
7	Haag	NLD	Zuid-Holland	440900
8	Utrecht	NLD	Utrecht	234323
9	Eindhoven	NLD	Noord-Brabant	201843
10	Tilburg	NLD	Noord-Brabant	193238
11	Groningen	NLD	Groningen	172701
12	Breda	NLD	Noord-Brabant	160398
13	Apeldoorn	NLD	Gelderland	153491
14	Nijmegen	NLD	Gelderland	152463
15	Enschede	NLD	Overijssel	149544

city

id	INTEGER
name	CHARACTER VARYING(64)
countrycode	CHARACTER(3)
district	CHARACTER VARYING(64)
population	INTEGER

Creating the UnitedStates Database

In a Git Bash window, enter the command below, then push the enter key. It doesn't matter what directory you are in - this will work anywhere.

```
winpty createdb -U postgres UnitedStates
```

MINGW64:/c/Users/Student

```
Student@V-P88F001 MINGW64 ~
$ winpty createdb -U postgres UnitedStates
Password:

Student@V-P88F001 MINGW64 ~
$
```

When prompted for a password, enter `postgres1` and push the enter key.

What you type will not be visible, so type it carefully

If you see no error after you push the enter key, you are good to go!

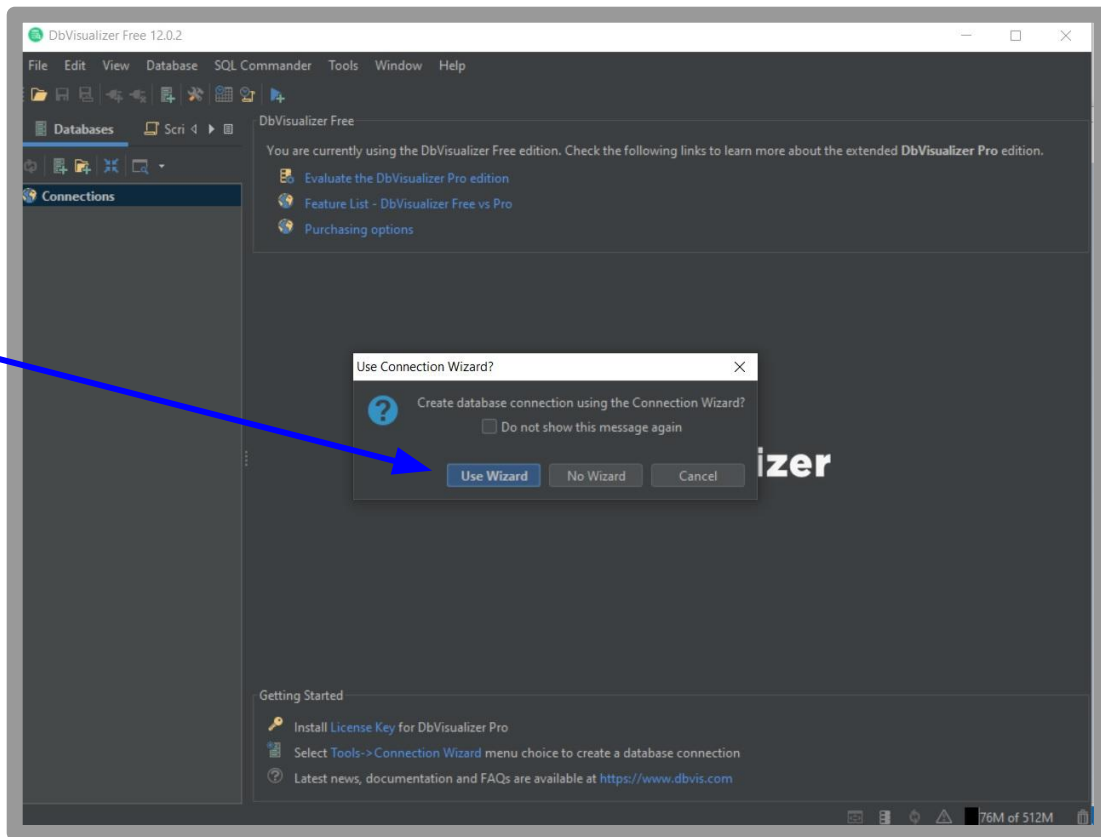# Time To Set Up DbVisualizer

Creating a Connection

In order to connect to database in DBVisualizer, we need to create a database connection.

Start by selecting
Create Database Connection
from the **Database** menu.
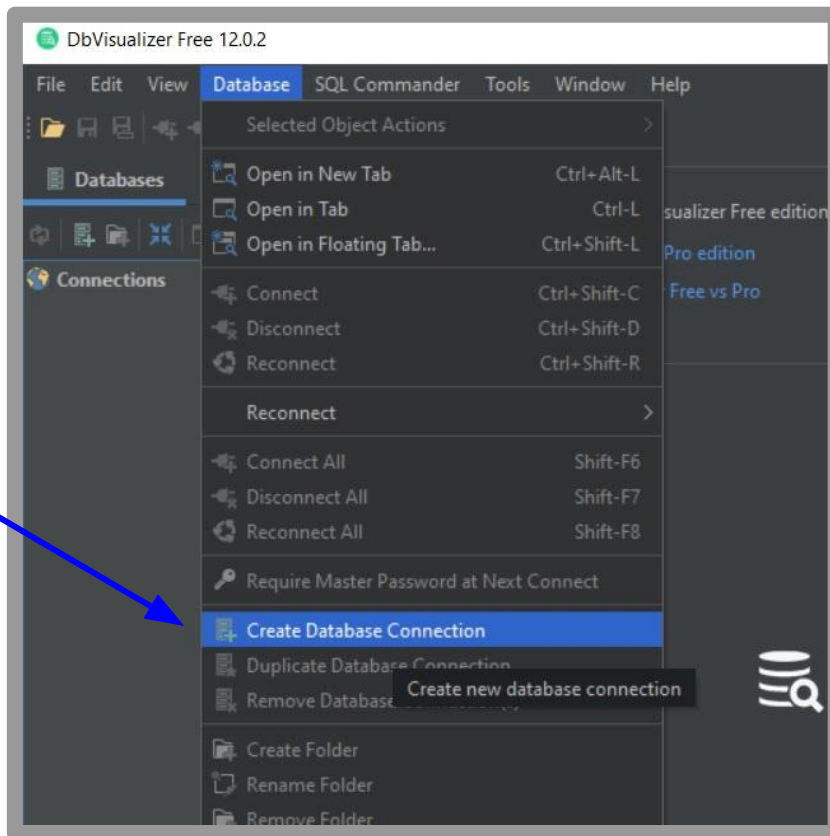
Creating a Connection

When the
Use Connection Wizard?
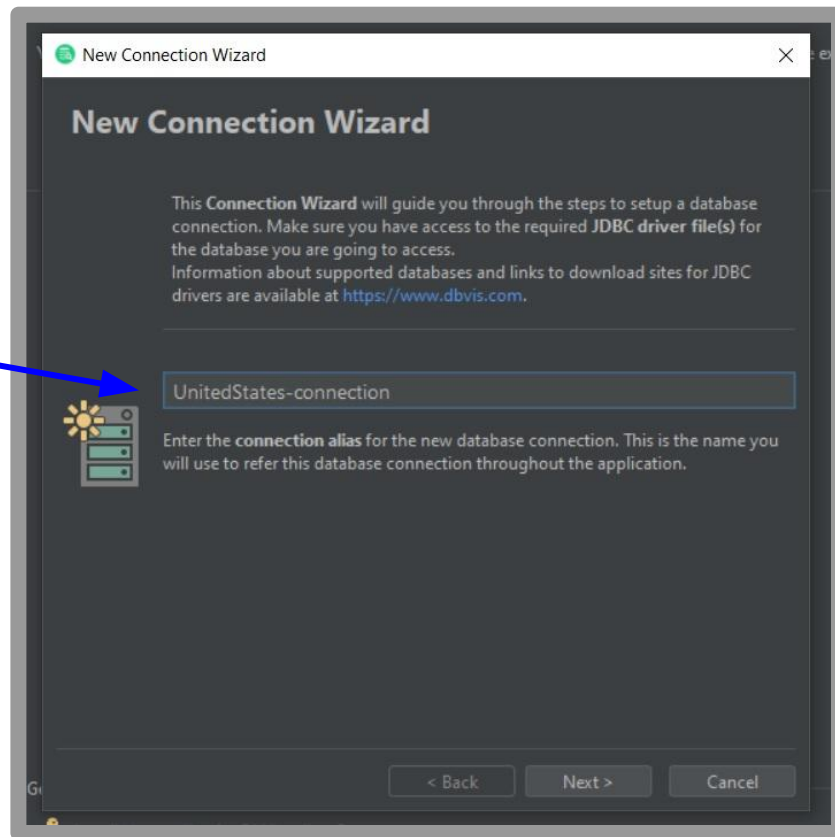pop-up appears, click on the
Use Wizard button.

Creating a Connection

Enter an alias for the connection.

This is not the name of the database, it is just a name for the connection we are creating **TO** the database.

After entering the connection name, click the *Next* button.

New Connection Wizard ✕

New Connection Wizard

This **Connection Wizard** will guide you through the steps to setup a database connection. Make sure you have access to the required **JDBC driver file(s)** for the database you are going to access.
Information about supported databases and links to download sites for JDBC drivers are available at https://www.dbvis.com.

UnitedStates-connection

Enter the **connection alias** for the new database connection. This is the name you will use to refer this database connection throughout the application.

< Back   Next >   Cancel

Creating a Connection

This will be a connection to a PostgreSQL database, so we need to select **PostgreSQL** as the database driver.

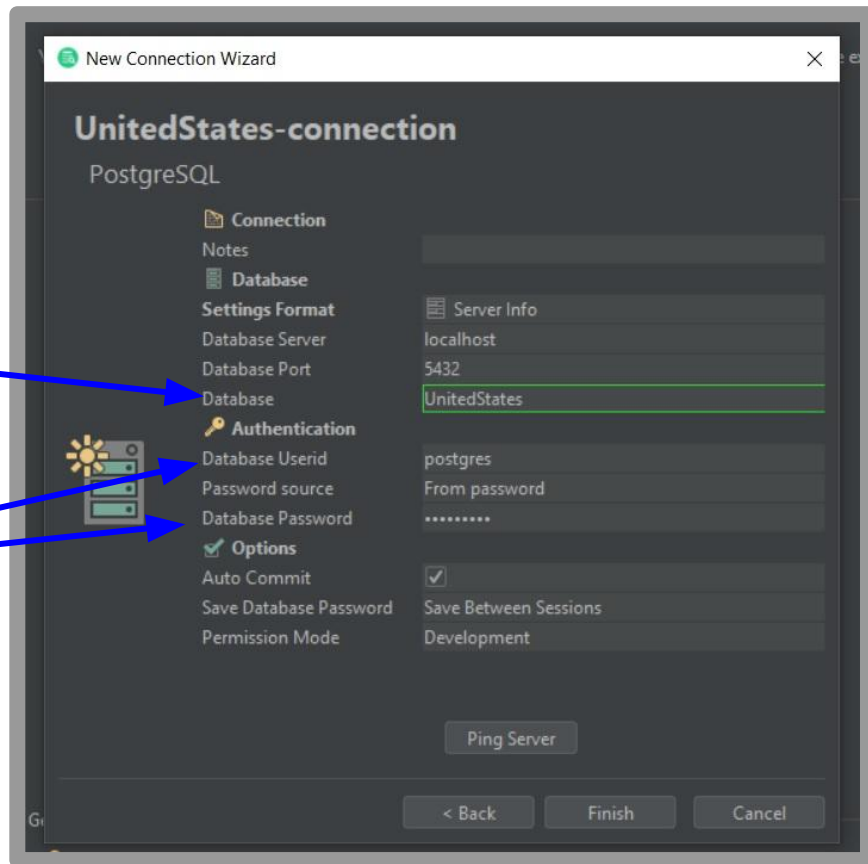After selecting the driver, click the **Next** button.
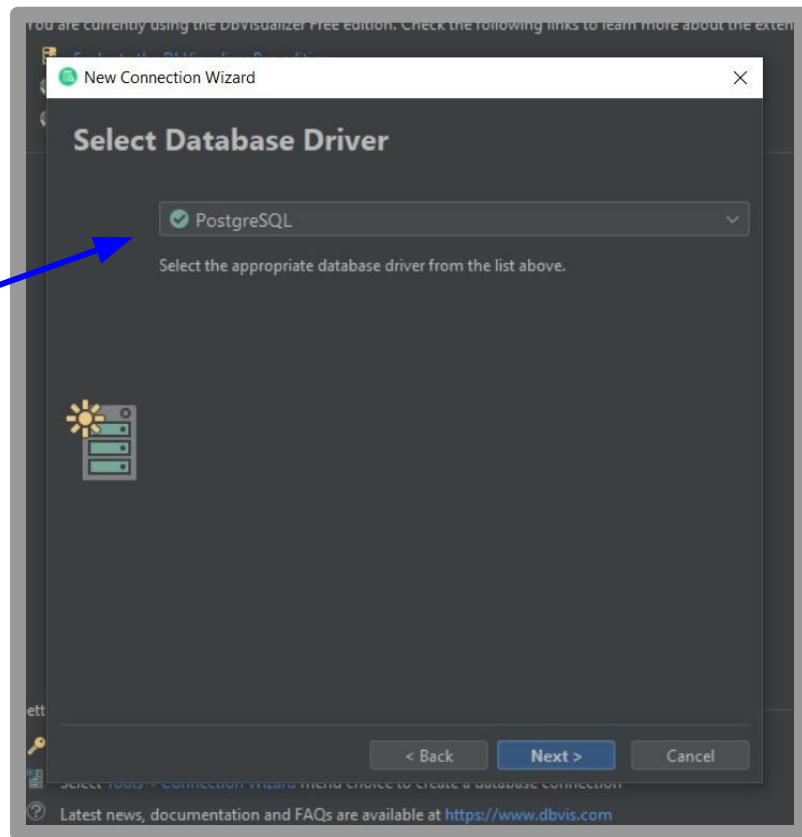
Creating a Connection

Finally, we need to provide the details for connection to the database.

The name of the database (`UnitedStates` in this case) goes in the *Database* field.

For the databases we create, we will use the postgres user. Enter `postgres` in the *Database Userid* field and `postgres1` in the *Database Password* field.

Once you have filled out these details, click the *Finish* button.
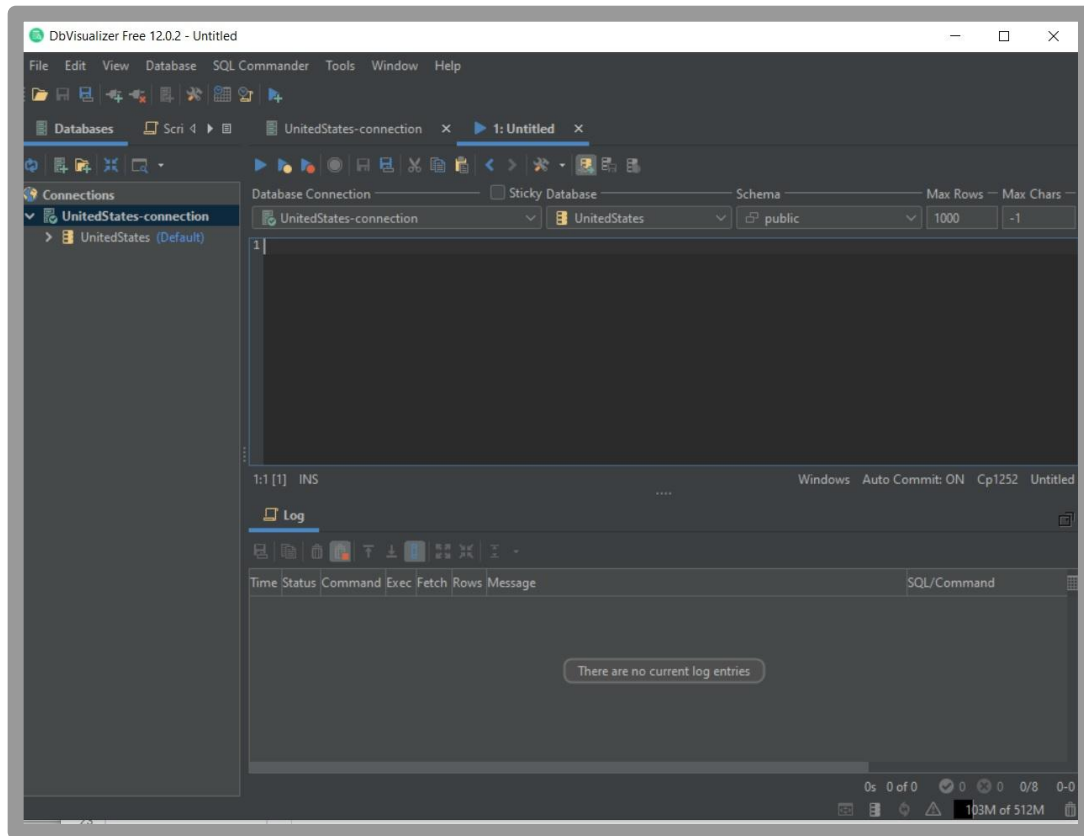
Creating a Connection

If you were able to connect to the database correctly, you should see something like this in the **Connections** column.

If you, do select **New SQL Commander** from the **SQL Commander** menu.
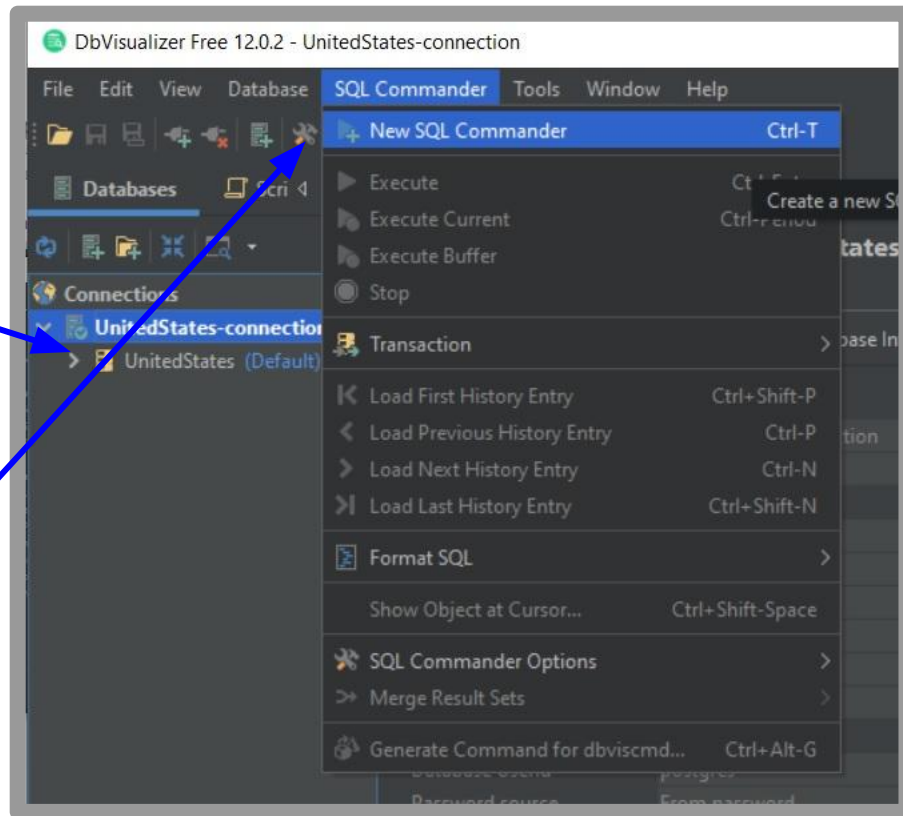
CREATING A CONNECTION

If you have completed all the steps successfully, you should see something like this

Intro To Queries

SQL stands for **structured query language** and is a **declarative programming language** to retrieve and update records from a database.

SQL consists of:
- **Data definition language** to define the data structures
- **Data manipulation language** to query and modify the data in a database
- **Data control language** to define access to a particular database

Intro To Queries

- The **SELECT** clause indicates what columns to get from a database table.

- The **FROM** clause indicates which table(s) to retrieve the data from.

```
SELECT city_name FROM city;
```

- If you want get all the columns in a table you can use * for the column.

```
SELECT * FROM city;
```

Intro To Queries

- The **SELECT** clause indicates what columns to get from a database table.

- The **FROM** clause indicates which table(s) to retrieve the data from.

- The **WHERE clause** is used to filter the result set using one or more criteria rules.

```
SELECT city_name FROM city WHERE
state_abbreviation='PA';
```

Intro To Queries

- Conditional clauses in the **WHERE clause** can include
  - `=, <>, !=, >, >=, <, <=`
  - `IN(values), NOT IN(values)`
  - `BETWEEN value AND value`
  - `IS NULL, IS NOT NULL`
  - `LIKE (with wildcard character)`

Casting in PostgreSQL

- In PostgreSQL you can cast a field to a different type by using ::
  followed by the type to cast to.

- To cast a literal, you can add the type before the value followed by a
  space..

```
select * from the_table
where the_timestamp_column::date =
date '2015-07-15';
```

Intro To Queries

- The **DISTINCT clause** indicates that duplicate values should not be included

- The **AS clause** establishes an alias for a particular column name

```
SELECT DISTINCT state_abbreviation AS state
FROM city;
```

Let's Write Some Queries!!!

EXERCISES

Notes on today's exercises

- GNP is expressed in millions of US Dollars.
- Average Life Expectancy just refers to value of the `lifeexpectancy` field.
- The value immediately after the problem statement is the expected number of rows that should be returned by the query.