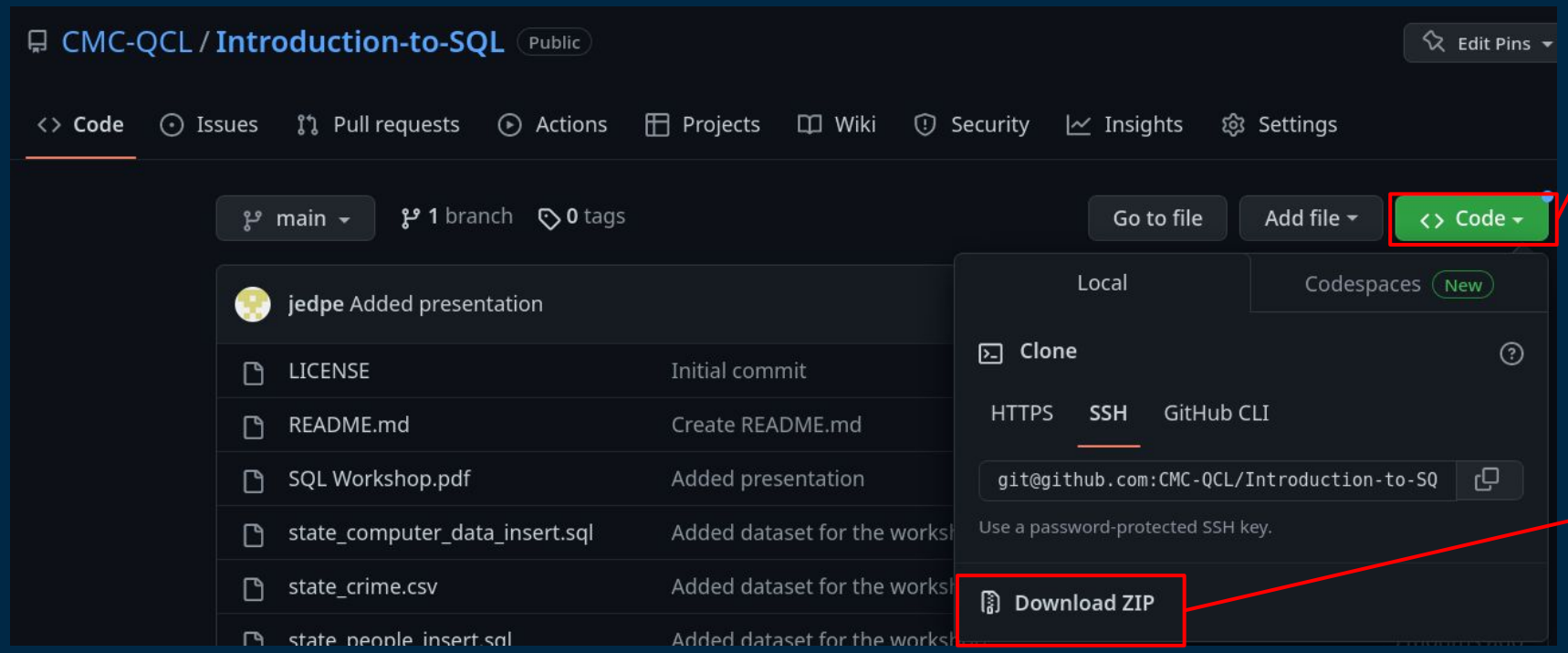
Introduction to SQL

Jorge Peña
QCL Graduate Fellow

Before we start

- Download the materials for this workshop
  - https://github.com/CMC-QCL/Introduction-to-SQL



Click on the "Code" button

Download it as a ZIP file

What is SQL?

- SQL stands for Structured Query Language
- DBeaver is a Database Administration Software
  - Connect to Databases
  - Add data to your databases and tables
  - Retrieve data

Interface

Choose your connection and database

SQL Editor controls

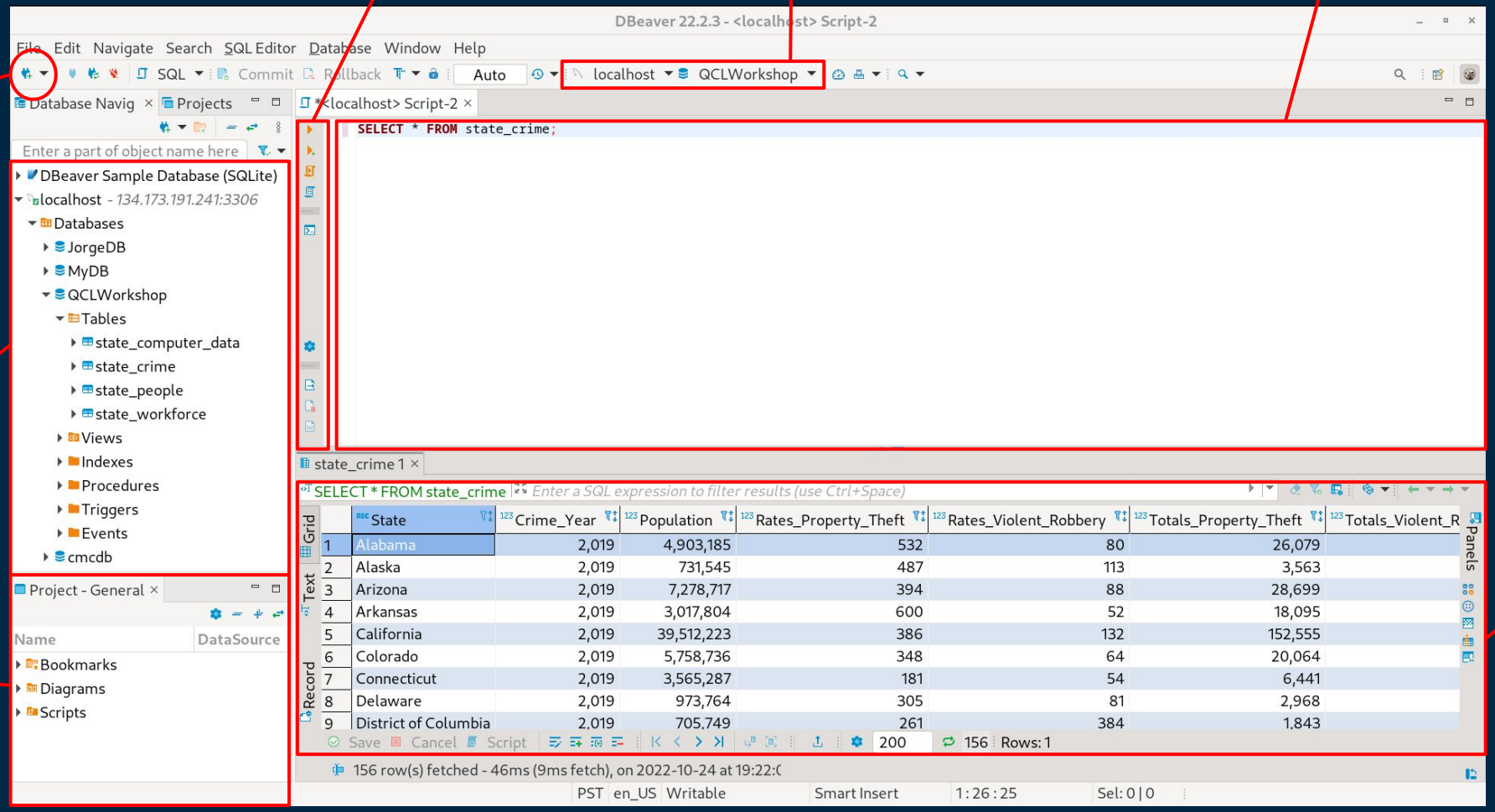
SQL Editor

Create a new connection

Databases and tables

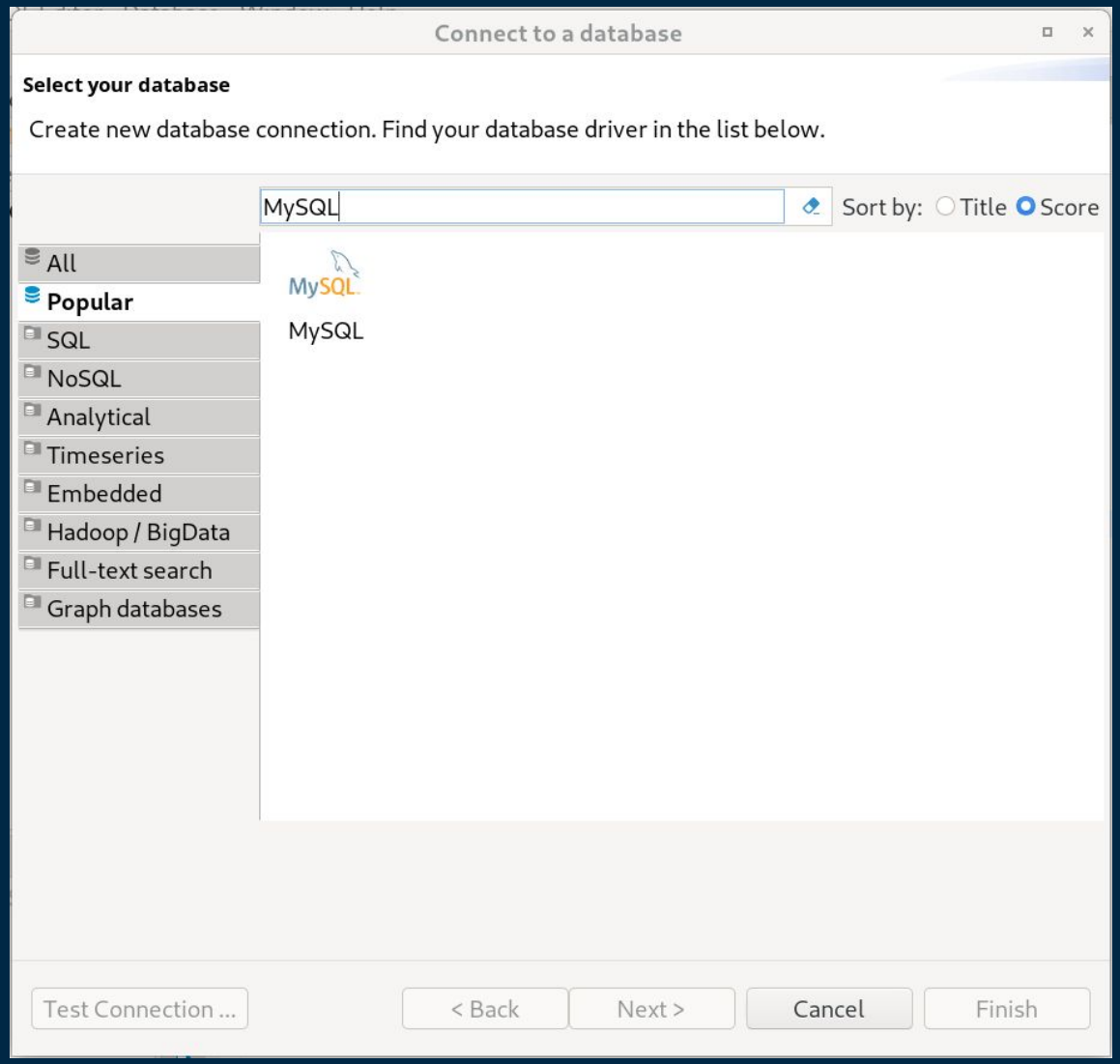
Results view

Project view



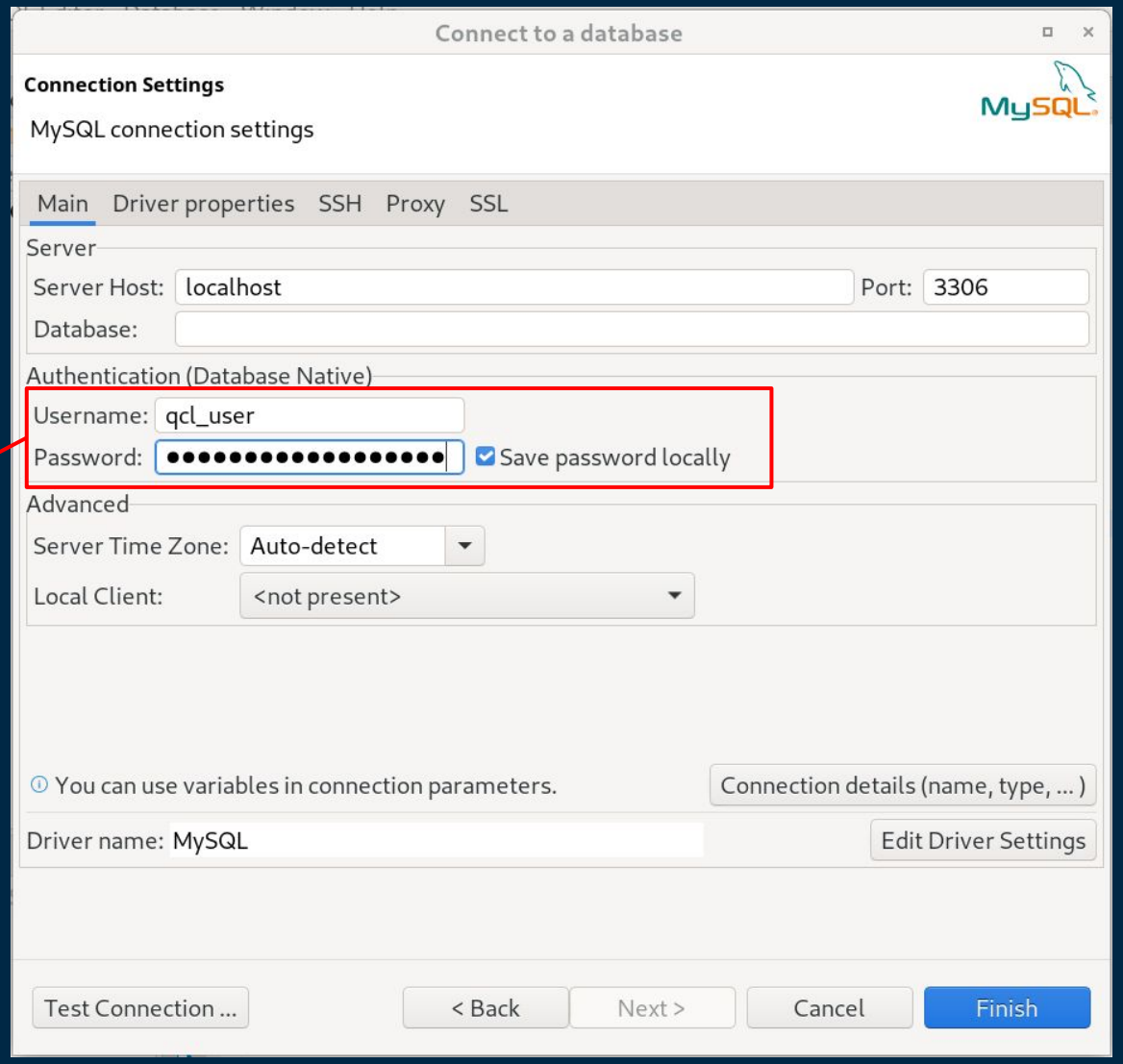
Create a Connection

- Create a new connection using MySQL



Database Settings

Enter the Username and Password for the database



SSH Tunnel

Select the SSH tab

Enter the IP **134.173.191.241**

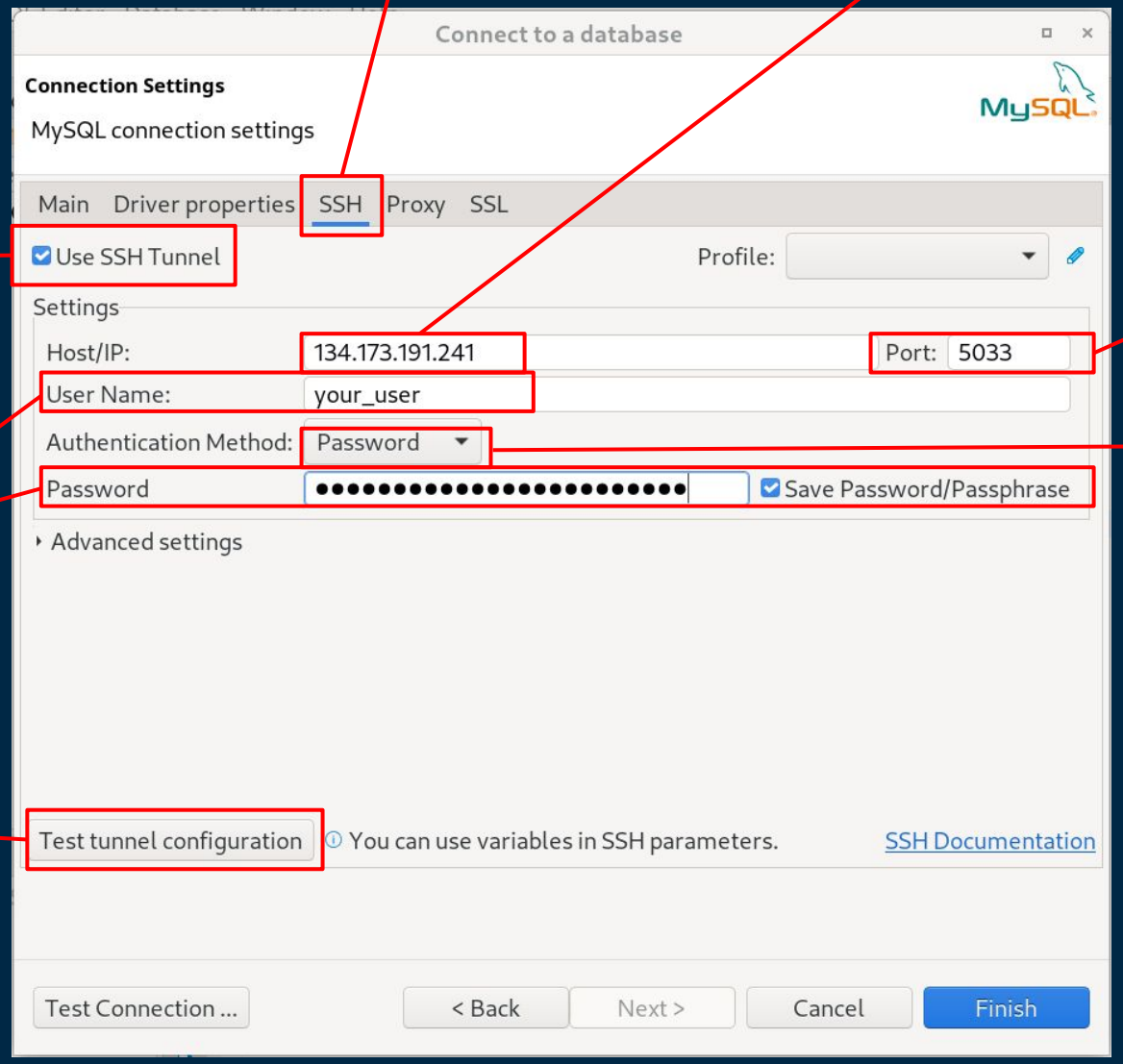
Select "Use SSH Tunnel"

Enter the port **5033**

Enter the Username and Password you receive through email

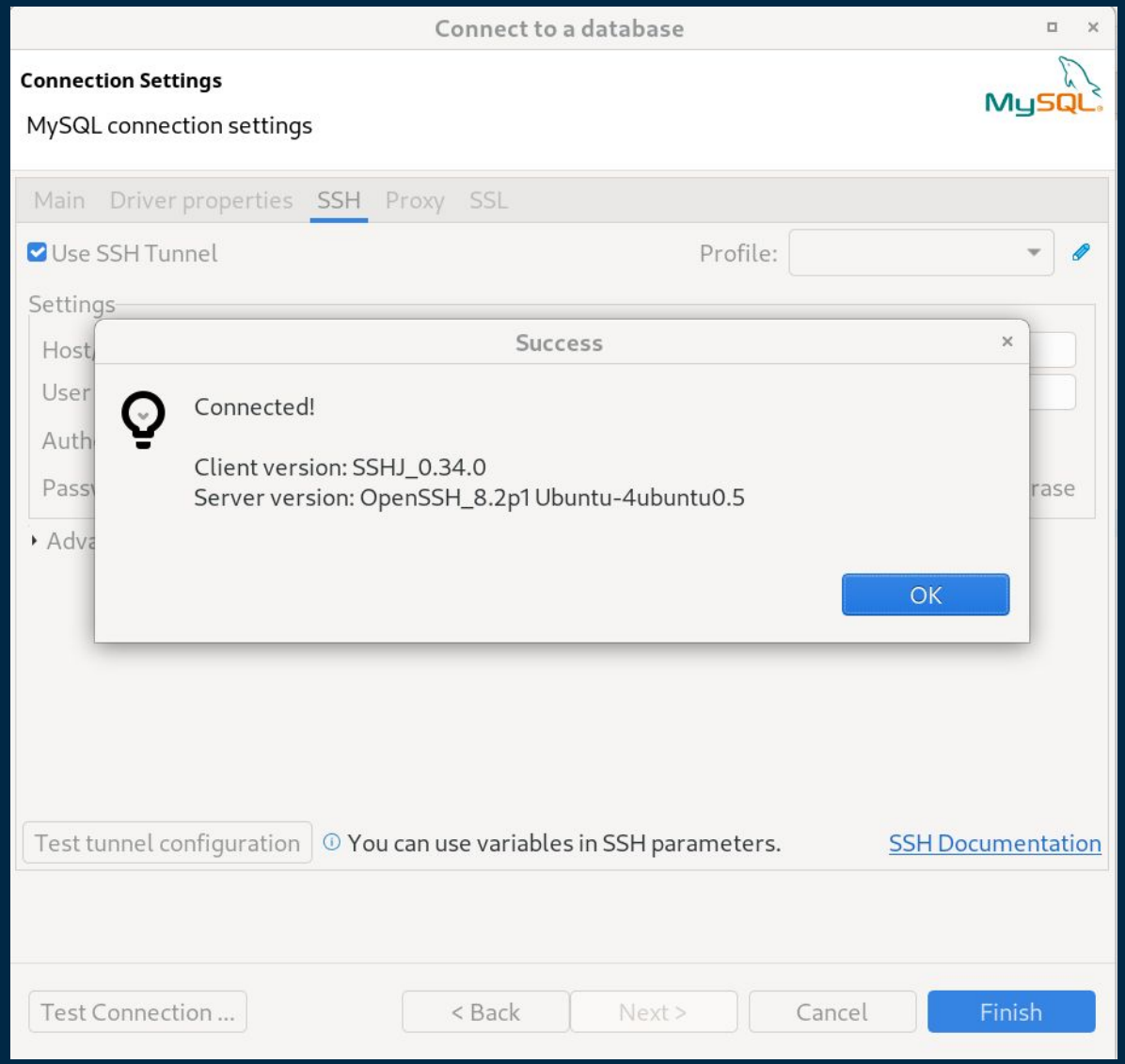
Select "Password" for authentication method

Finally test your configuration



Successful SSH

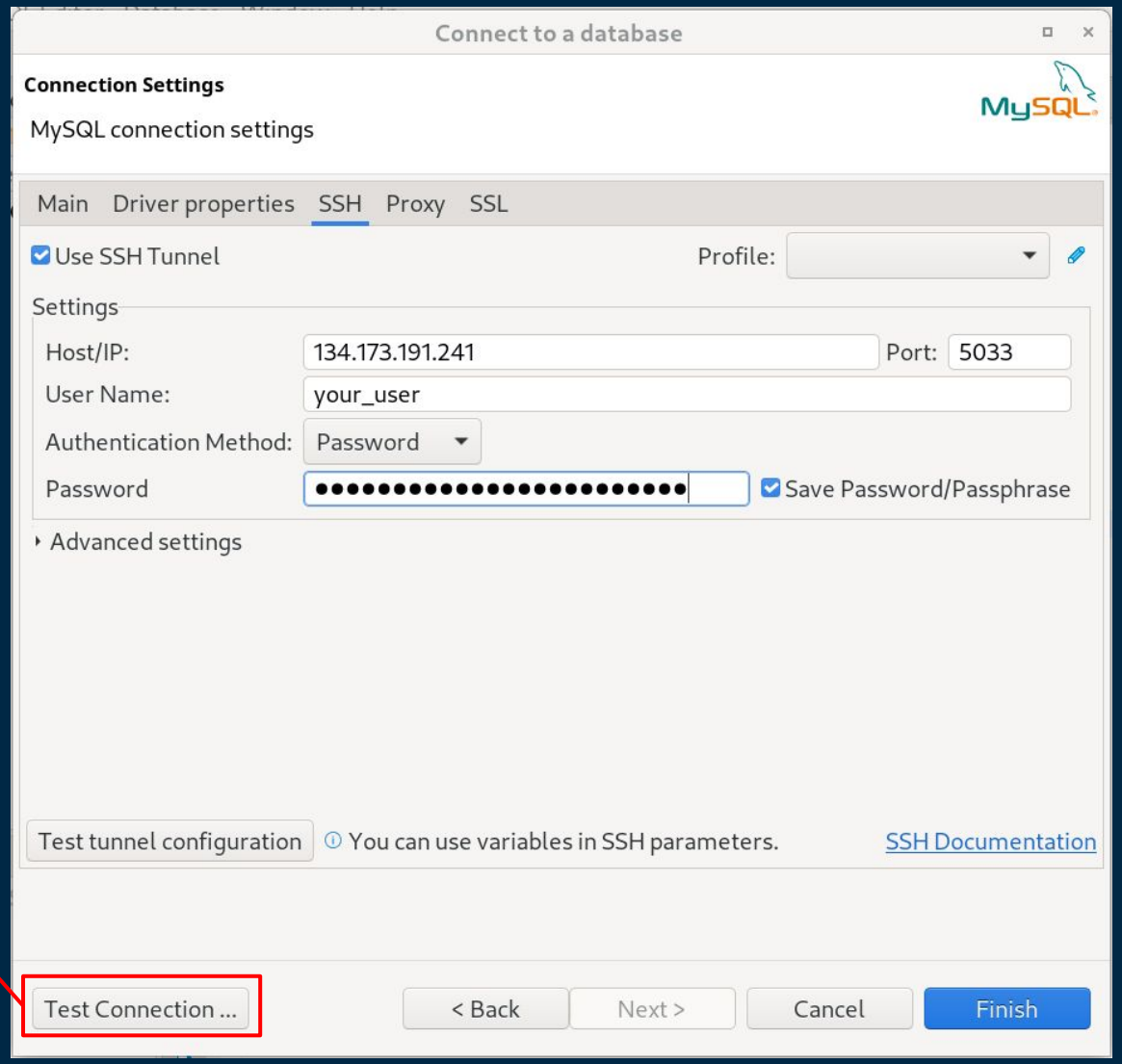
- Make sure to approve the push notification on DUO
- You will see a success message if the SSH connection works



Test the connection

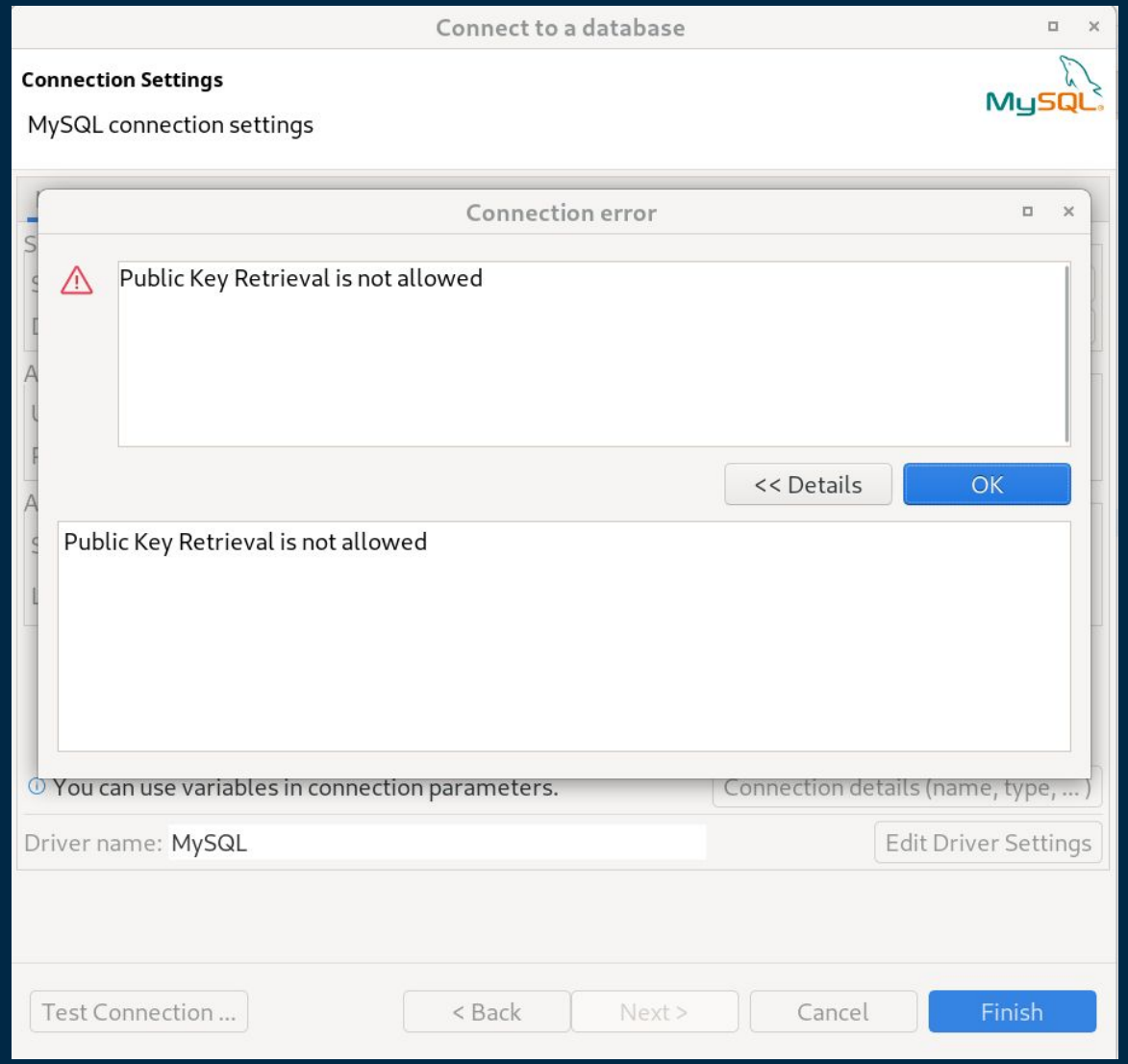
- Now you need to test the database connection

Test your connection



Connection error

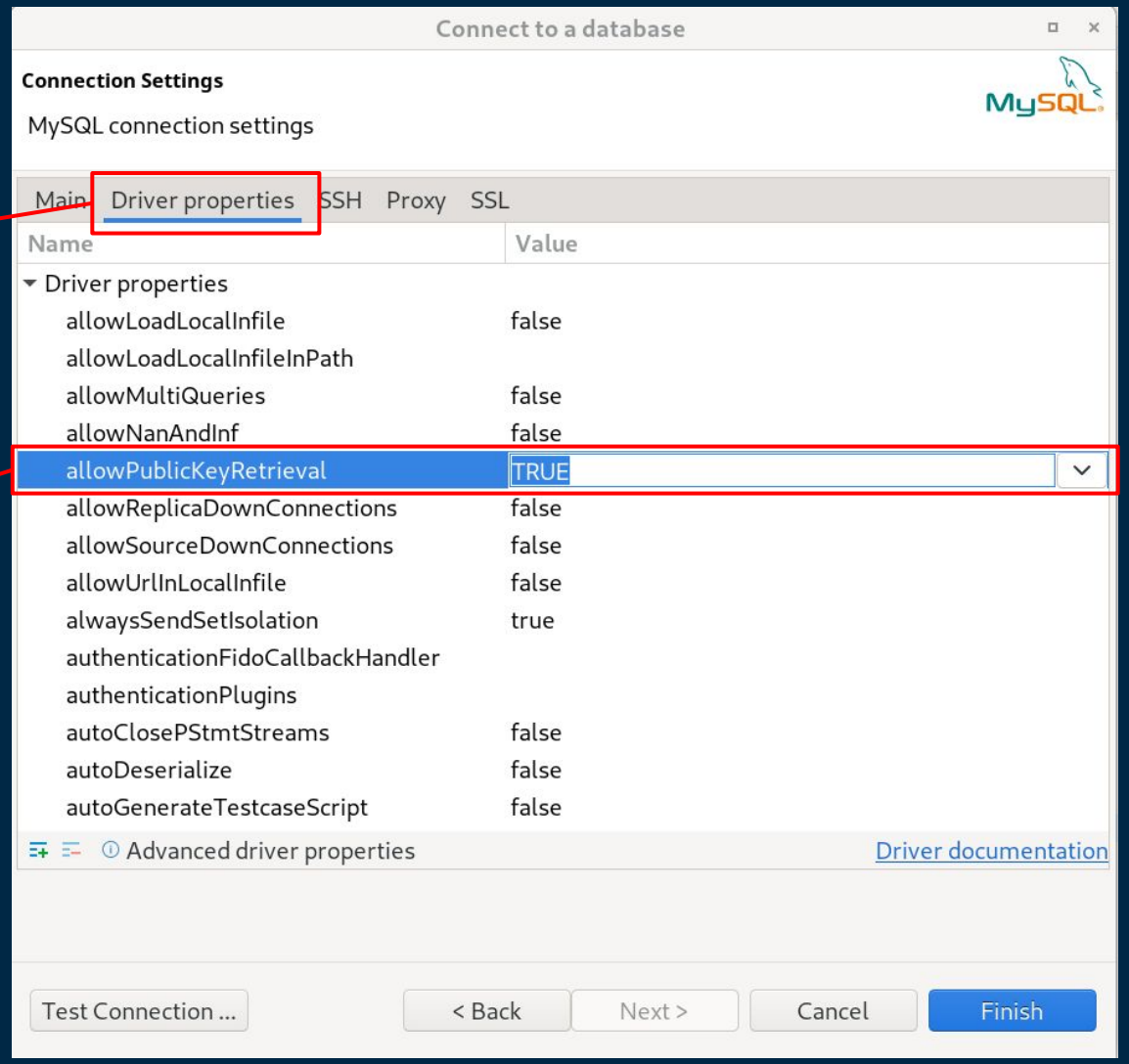
- If you get a connection error like this one follow the steps on the following slides



Driver properties

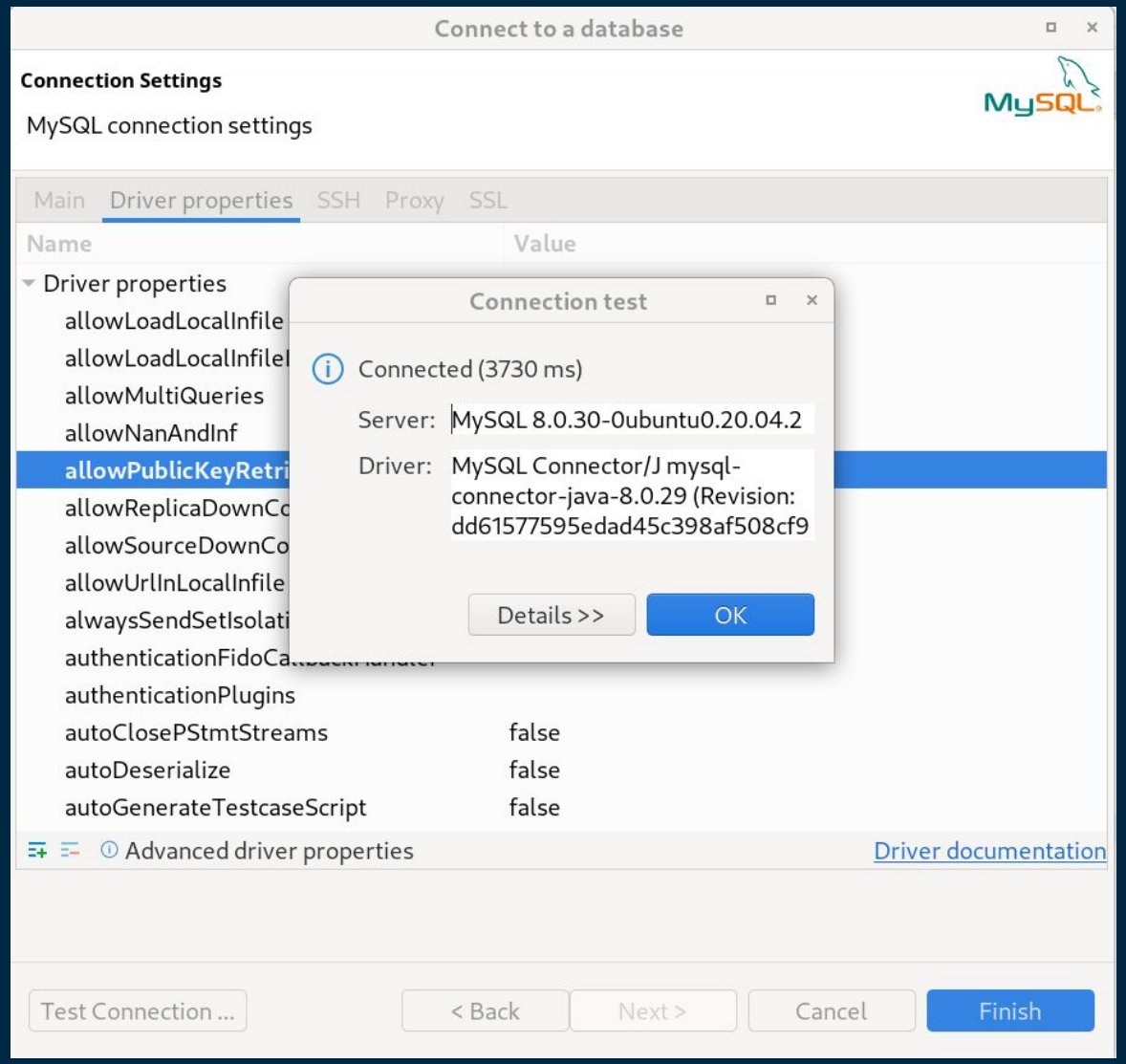
Click on the Driver properties tab

Change the value of **allowPublicKeyRetrieval** to TRUE



Successful connection

- Test your connection once again and you should see the success



Agenda

Databases and SQL

- Relational Databases
- SQL Overview

Tables and Data

- Databases and Tables
- Today's data

Writing Queries

- Retrieving Data
- Sorting and Filtering

Subqueries and Joins

- Subqueries and Aliases
- Joining Tables

Databases and SQL

Database Advantages

01

BETTER DATA INTEGRATION

Improves data handling and reduces redundancy

02

STORAGE IS MORE SECURE

Provides better privacy and security policies

03

FASTER DATA ACCESS

Produce quick answers to data queries

Relational Databases

Students_Registration

StudentID	ClassID	Semester
71225	1005	Fall21
86634	1006	Spr22
32238	1006	Spr22

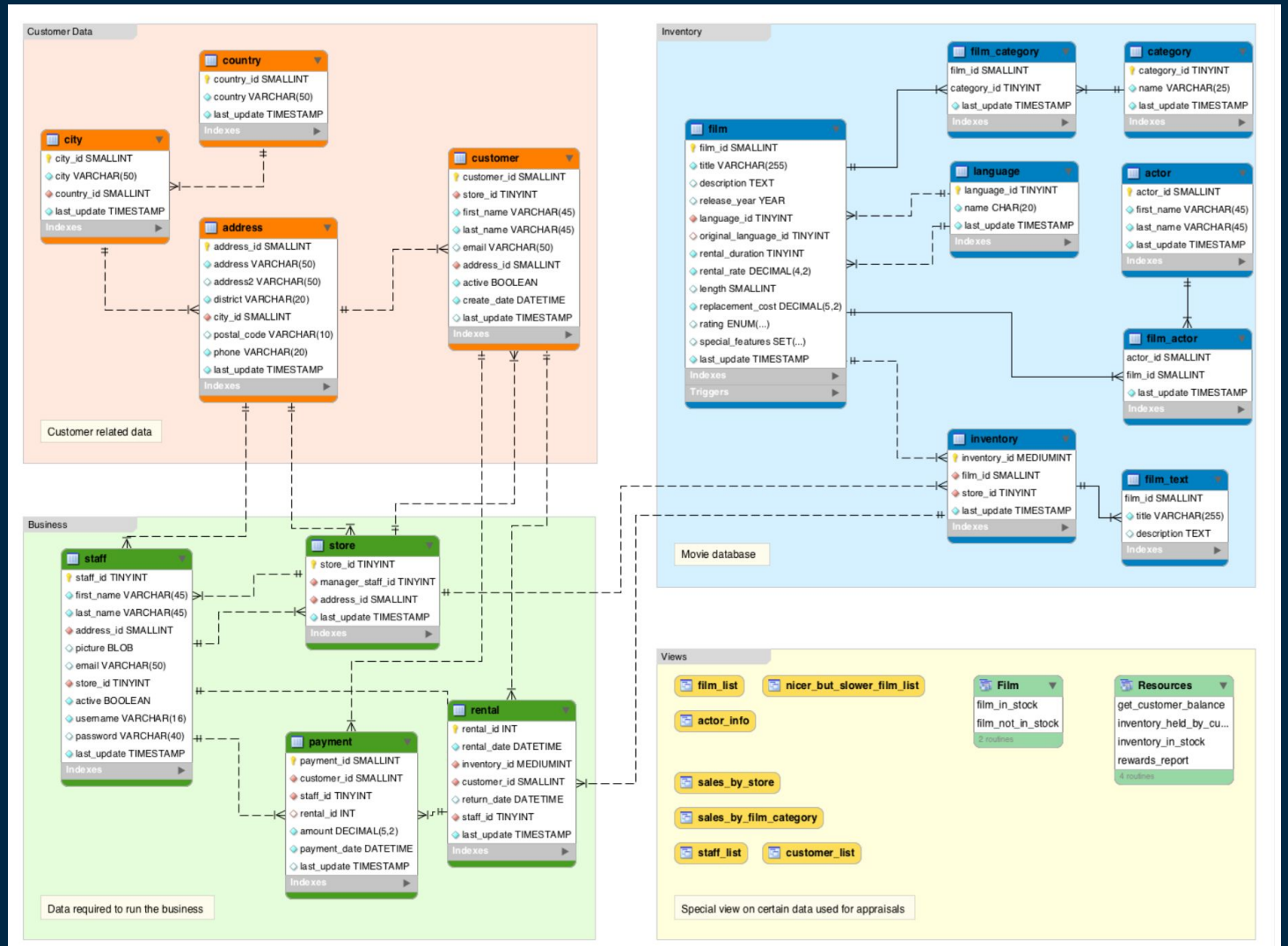
School_Courses

ClassID	Title	ClassNum
1005	Intro to Art History	500
1006	Intro to SQL	501
1009	Intro to Databases	300

Students

ID	Name	Grade	DOB	GPA
71225	Lili	Freshman	1995-03-12	3.5
32238	Brenda	Senior	1989-05-28	3.9
86634	James	Freshman	1999-09-20	4.0

Database Schema



A database schema diagram of the Sakila Sample Database.

https://database.guide/what-is-a-database-schema/

Primary and Foregin Keys

Keys are used to create relationships between tables

- *Primary Keys* (PK) are columns that uniquely identified a row in a table
- *Foreign Keys* (FK) are columns that correspond to the primary key in another table

Students_Registration

FK

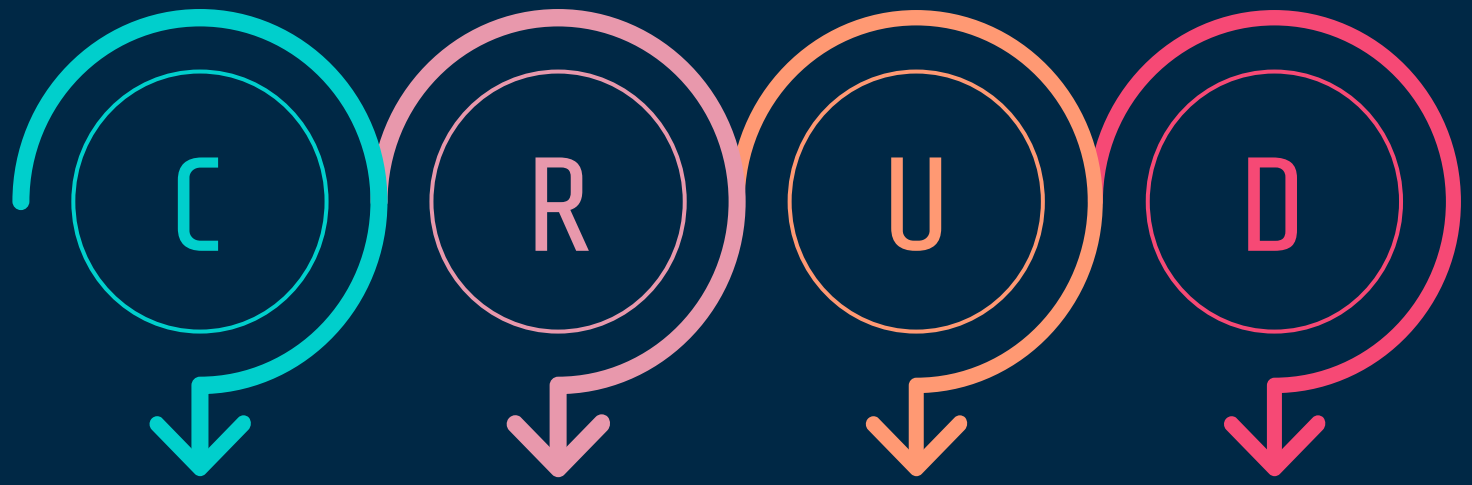
StudentID	ClassID	Semester
71225	1005	Fall21
86634	1006	Spr22
32238	1006	Spr22

PK

School_Courses

ClassID	Title	ClassNum
1005	Intro to Art History	500
1006	Intro to SQL	501
1009	Intro to Databases	300

SQL Overview



CREATE

Create databases and tables

READ

Look at specific data

UPDATE

Make changes to data

DELETE

Remove data

Vocabulary

- Data Definition Language (DDL):
  - CREATE, DROP, ALTER, TRUNCATE
- Data Manipulation Language (DML):
  - INSERT, UPDATE, DELETE
- Data Query Language (DQL):
  - SELECT, JOIN
- Data Control Language (DCL) or Transaction Control Language (TCL):
  - GRANT, REVOKE

SQL Data Types

Students

ID	Name	Grade	DOB	GPA
71225	Lili	Freshman	1995-03-12	3.5
32238	Brenda	Senior	1989-05-28	3.9
86634	James	Freshman	1999-09-20	4.0

- Each column stores only one type of data
- Data types determine the available functions
- Different types of data take up different space

Integers

Students

ID	Name	Grade	DOB	GPA
71225	Lili	Freshman	1995-03-12	3.5
32238	Brenda	Senior	1989-05-28	3.9
86634	James	Freshman	1999-09-20	4.0

Integer field

- `INT` / `INTEGER` is a signed whole number

Strings

String fields

Students

ID	Name	Grade	DOB	GPA
71225	Lili	Freshman	1995-03-12	3.5
32238	Brenda	Senior	1989-05-28	3.9
86634	James	Freshman	1999-09-20	4.0

- Can contain letters, numbers and special characters
- VARCHAR(size) is a variable length string of maximum size length (up to 65,535)

Date and Time

Date field

Students

ID	Name	Grade	DOB	GPA
71225	Lili	Freshman	1995-03-12	3.5
32238	Brenda	Senior	1989-05-28	3.9
86634	James	Freshman	1999-09-20	4.0

- Represent temporal values of date, time and datetime values
- `DATE` with format `YYYY-MM-DD`
- `TIME` with format `hh:mm:ss`
- `DATETIME` with a combination with format `YYYY-MM-DD hh:mm:ss`
- MySQL can store years with the `YEAR` data type with a range of 1901 to 2155

Floats

Students

ID	Name	Grade	DOB	GPA
71225	Lili	Freshman	1995-03-12	3.5
32238	Brenda	Senior	1989-05-28	3.9
86634	James	Freshman	1999-09-20	4.0

Float field

- Numbers with a fractional part
- In MySQL `NUMERIC` and `DECIMAL` are equivalent
- `DECIMAL(M, d)` has `M` digits with `d` of them being digits after decimal point
  - e.g. `DECIMAL(5,2)` stores any number between -999.99 and 999.99

Data types summary

String field

Date fields

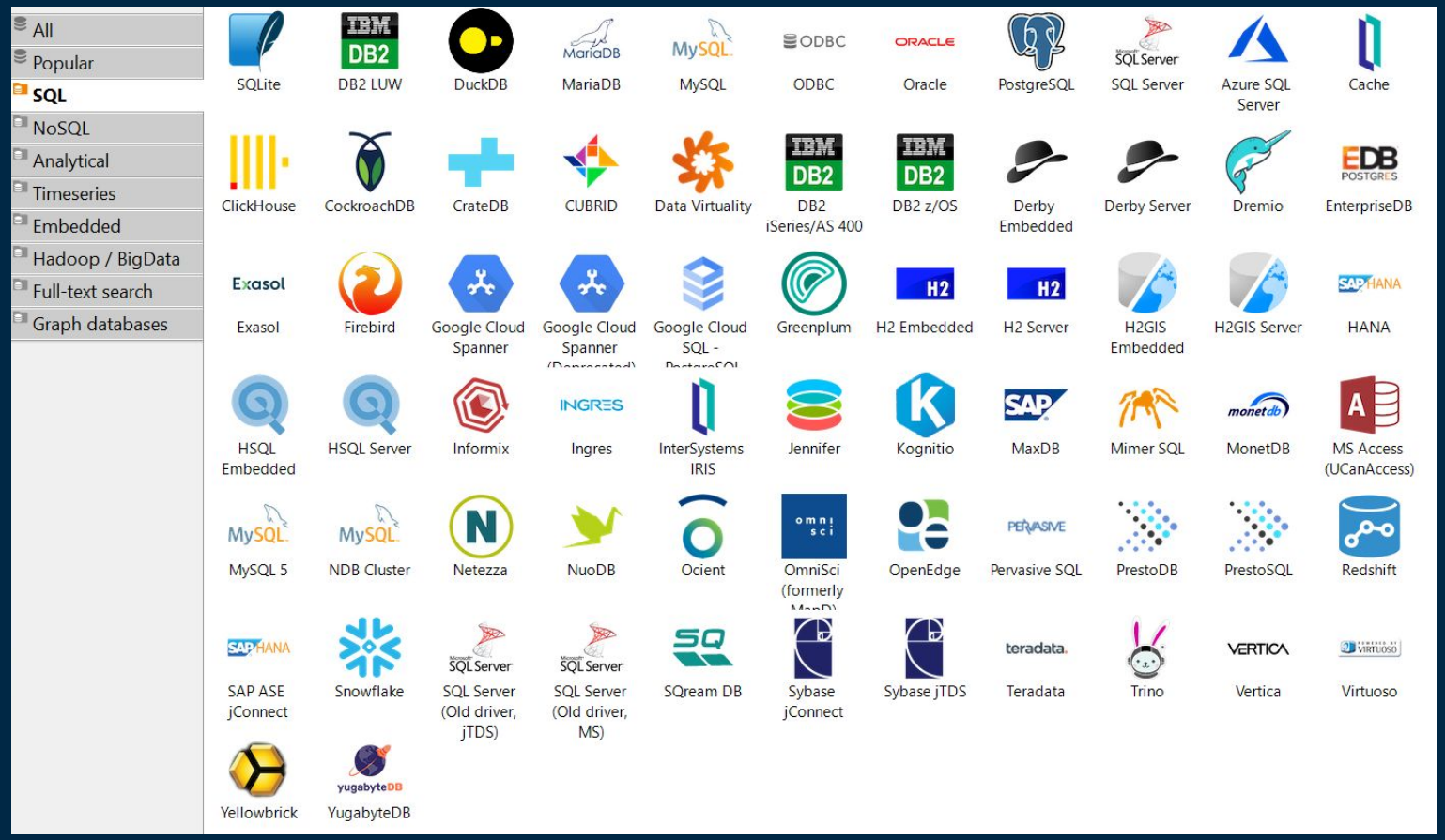
Float field

Integer field

Students

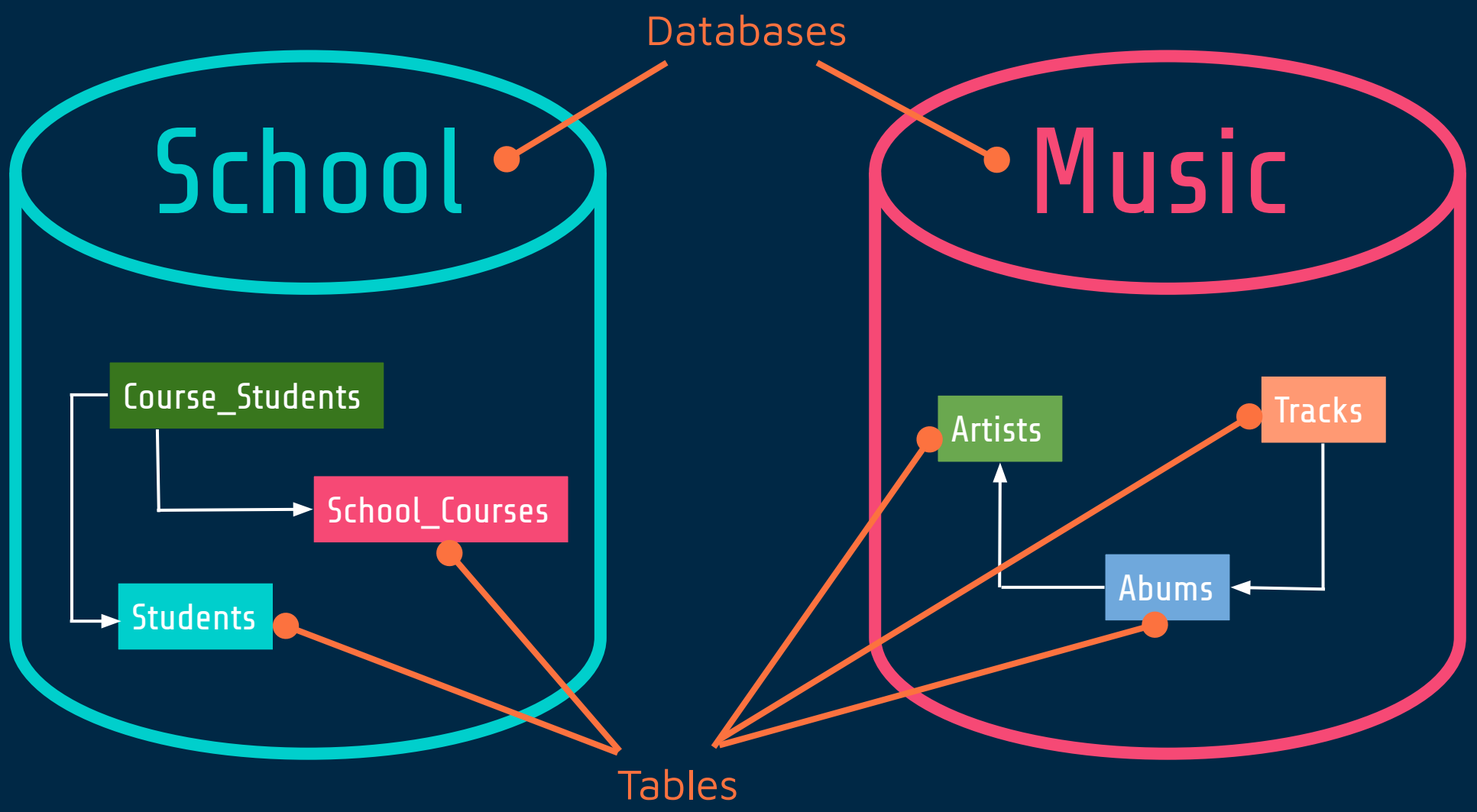
ID	Name	Grade	DOB	GPA
71225	Lili	Freshman	1995-03-12	3.5
32238	Brenda	Senior	1989-05-28	3.9
86634	James	Freshman	1999-09-20	4.0

SQL Flavors


All
Popular
SQL
NoSQL
Analytical
Timeseries
Embedded
Hadoop / BigData
Full-text search
Graph databases
SQLite
DB2 LUW
DuckDB
MariaDB
MySQL
ODBC
Oracle
PostgreSQL
SQL Server
Azure SQL Server
Cache
ClickHouse
CockroachDB
CrateDB
CUBRID
Data Virtuality
DB2 iSeries/AS 400
DB2 z/OS
Derby Embedded
Derby Server
Dremio
EnterpriseDB
Exasol
Firebird
Google Cloud Spanner
Google Cloud Spanner
Google Cloud SQL -
Greenplum
H2 Embedded
H2 Server
H2GIS Embedded
H2GIS Server
HANA
HSQL Embedded
HSQL Server
Informix
Ingres
InterSystems IRIS
Jennifer
Kognitio
MaxDB
Mimer SQL
MonetDB
MS Access (UCanAccess)
MySQL 5
NDB Cluster
Netezza
NuoDB
Ocient
OmniSci (formerly
OpenEdge
Pervasive SQL
PrestoDB
PrestoSQL
Redshift
SAP ASE jConnect
Snowflake
SQL Server (Old driver, jTDS)
SQL Server (Old driver, MS)
SQream DB
Sybase jConnect
Sybase jTDS
Teradata
Trino
Vertica
Virtuoso
Yellowbrick
YugabyteDB

Tables and Data

Databases and Tables



Records and Fields

Students

ID	Name	Grade	DOB	GPA
71225	Lili	Freshman	1995-03-12	3.5
32238	Brenda	Senior	1989-05-28	3.9
86634	James	Freshman	1999-09-20	4.0

Row = Tuple or Record

Holds information for one observation

Column = Attribute or Field

Holds specific information about all observations

Today's data

- Modified datasets for workshop (4 files total)
- **State Crime CSV File**
  - **state_crime.csv**
  - information on the crime rates and totals for states across the United States for a wide range of years
  - reports go from 1960 to 2019 (only used 2010, 2014 and 2019)
  - https://corgis-edu.github.io/corgis/csv/state_crime/
- **State Demographics CSV and SQL Files**
  - **state_computer_data.sql, state_workforce.csv, state_people.sql**
  - summarized information obtained about states in the United States from 2015 through 2019 through the United States Census Bureau
  - just the summarized data as of 2019
  - https://corgis-edu.github.io/corgis/csv/state_demographics/

Create a Database

```sql
-- Create a database
CREATE DATABASE QCLWorkshop;

-- Calling the database
USE QCLWorkshop;
```

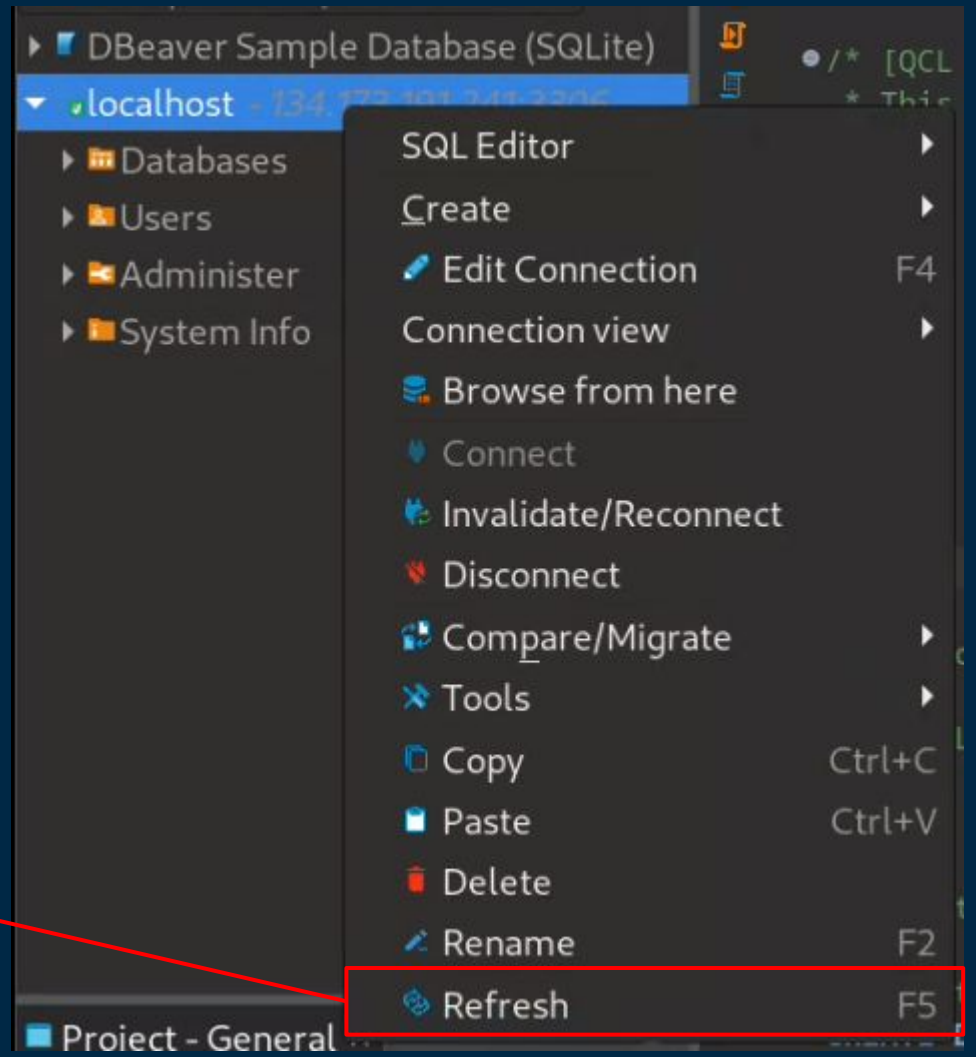
Statistics 1 ×

Name	Value
Updated Rows	1
Query	-- Create a database CREATE DATABASE QCLWorkshop
Finish time	Thu Oct 20 13:48:51 PDT 2022

Refresh

- Right click on your connection and click on refresh

Click on Refresh



Comments

```
-- This is a single line comment


/* [QCL Workshop] Introduction to SQL
 * This is a multiline comment.
 */
```

Create a Table

```sql
-- Create a table
CREATE TABLE state_computer_data (
    State VARCHAR(50),
    Persons_per_household NUMERIC,
    Households_with_computer NUMERIC,
    Households_with_internet NUMERIC
);

-- Insert values
INSERT INTO state_computer_data
VALUES('Alabama', 2.55, 85.5, 76.4);
```

- After creating the table click on Refresh once more

Insert multiple values

DBeaver 22.2.2 - <localhost> state_computer_data_insert.sql

File Edit Navigate Search SQL Editor Database Window Help

SQL Commit Rollback Auto localhost QCLWorkshop

Database Navigator × Projects

Enter a part of object name here

- DBeaver Sample Database (SQLite)
- localhost - 134.173.191.241:3306
  - Databases
    - QCLWorkshop
      - Tables
        - state_computer_data
        - state_crime
        - state_workforce
      - Views
      - Indexes
      - Procedures
      - Triggers
      - Events
    - cmcdb
    - sys
    - testdb
    - toandb

Project - General ×

Name	DataSource
Bookmarks	
Diagrams	
Scripts	

*<localhost> Script-1 | <localhost> state_computer_data_insert.sql ×

```sql
INSERT INTO state_computer_data (State,Persons_per_Household,Househol
VALUES('Alaska',2.8,94.1,85.5),
      ('Arizona',2.68,91.7,84.1),
      ('Arkansas',2.52,86.2,73.0),
      ('California',2.95,93.0,86.7),
      ('Colorado',2.56,93.9,87.6),
      ('Connecticut',2.53,90.8,85.5),
      ('Delaware',2.57,91.6,85.0),
      ('District of Columbia',2.3,91.8,82.6),
      ('Florida',2.65,91.5,83.0),
      ('Georgia',2.7,90.2,81.3),
      ('Hawaii',3.0,91.2,84.8),
      ('Idaho',2.68,91.8,82.7),
      ('Illinois',2.57,89.9,82.7),
      ('Indiana',2.52,88.7,80.1),
      ('Iowa',2.4,89.0,80.8),
      ('Kansas',2.51,90.0,81.8),
      ('Kentucky',2.49,86.4,78.4),
```

Output ×

```
Data truncated for column 'Households_with_internet' at row 45
Data truncated for column 'Persons_per_household' at row 46
Data truncated for column 'Households_with_computer' at row 46
Data truncated for column 'Households_with_internet' at row 46
Data truncated for column 'Persons_per_household' at row 47
Data truncated for column 'Households_with_computer' at row 47
Data truncated for column 'Households_with_internet' at row 47
Data truncated for column 'Persons_per_household' at row 48
Data truncated for column 'Households_with_computer' at row 48
Data truncated for column 'Persons_per_household' at row 49
Data truncated for column 'Households_with_computer' at row 49
Data truncated for column 'Households_with_internet' at row 49
Data truncated for column 'Persons_per_household' at row 50
Data truncated for column 'Households_with_computer' at row 50
Data truncated for column 'Households_with_internet' at row 50
```

Statistics 1 ×

Name	Value
Updated Rows	50
Query	INSERT INTO state_computer_data (State,Persons_per_Household,Households_with_computer,Households_with_Internet) VALUES('Alaska',2.8,94.1,85.5), ('Arizona',2.68,91.7,84.1), ('Arkansas',2.52,86.2,73.0), ('California',2.95,93.0,86.7), ('Colorado',2.56,93.9,87.6), ('Connecticut',2.53,90.8,85.5),

Save Cancel Script 200 1

50 row(s) updated - 46ms, on 2022-10-21 at 01:05:35

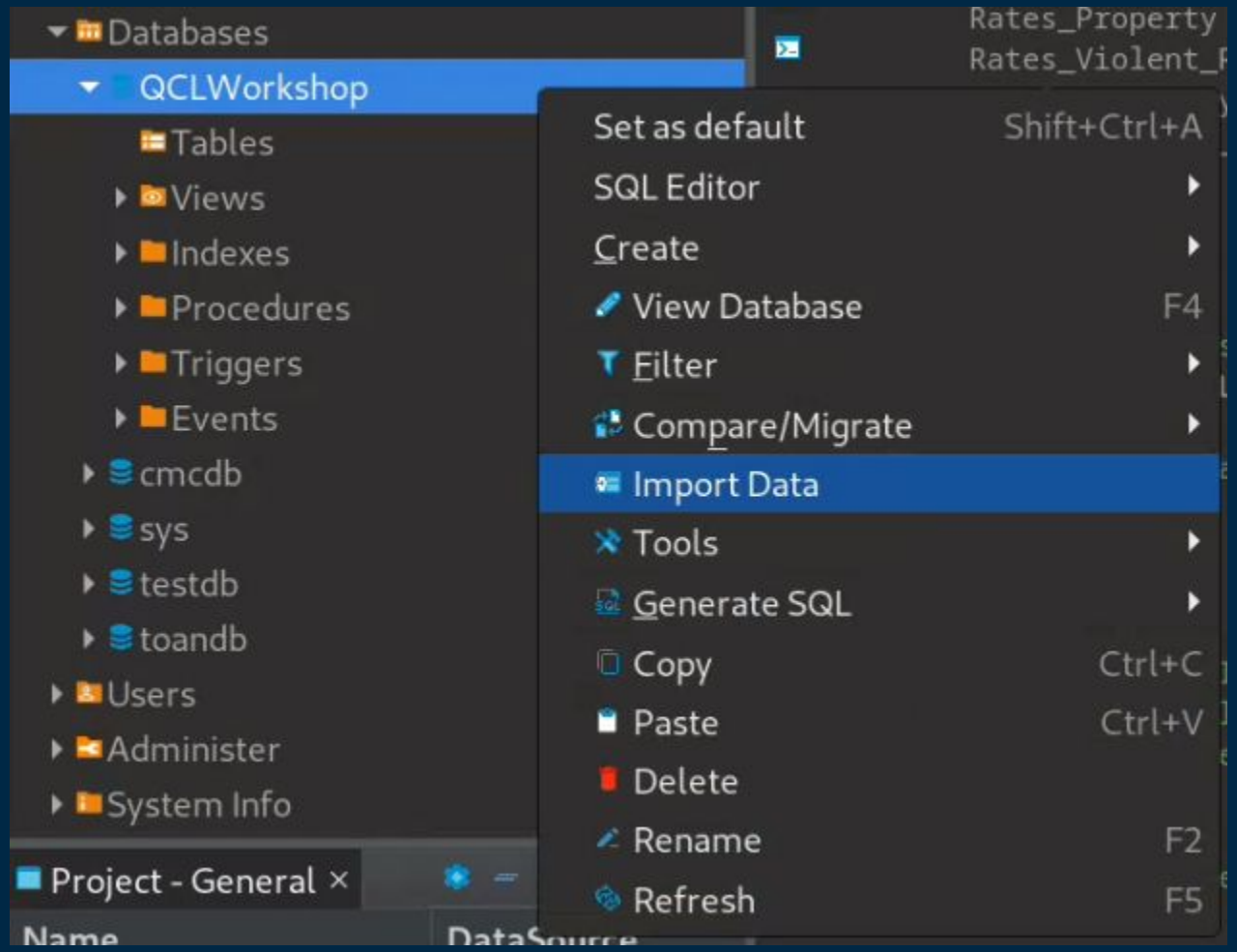
PST en_US Writable Smart Insert 1:1:0 Sel: 0 | 0

Hands-on

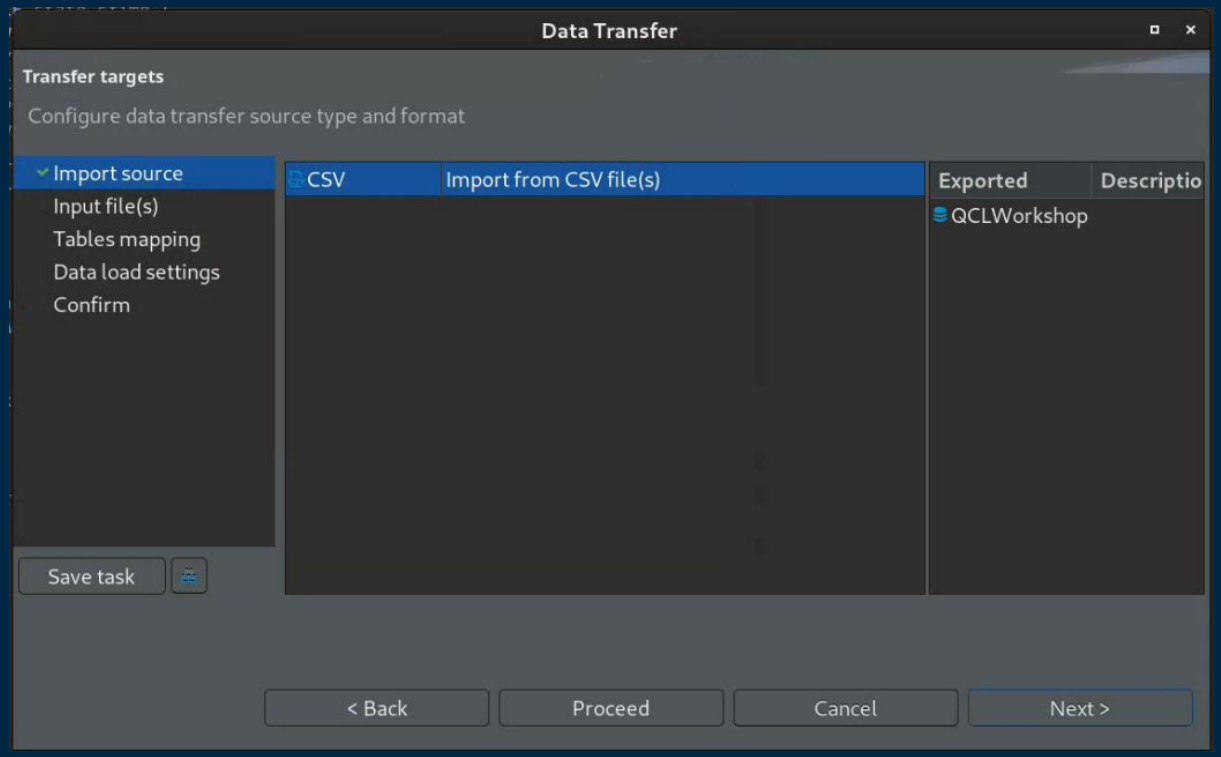
- Create a table named **state_people** and add the attributes
  1. State as VARCHAR with 20 characters
  2. Employment_Firms_Total as INT
  3. Age_Percent_Under_18_Years as NUMERIC
  4. Age_Percent_65_and_Older as NUMERIC

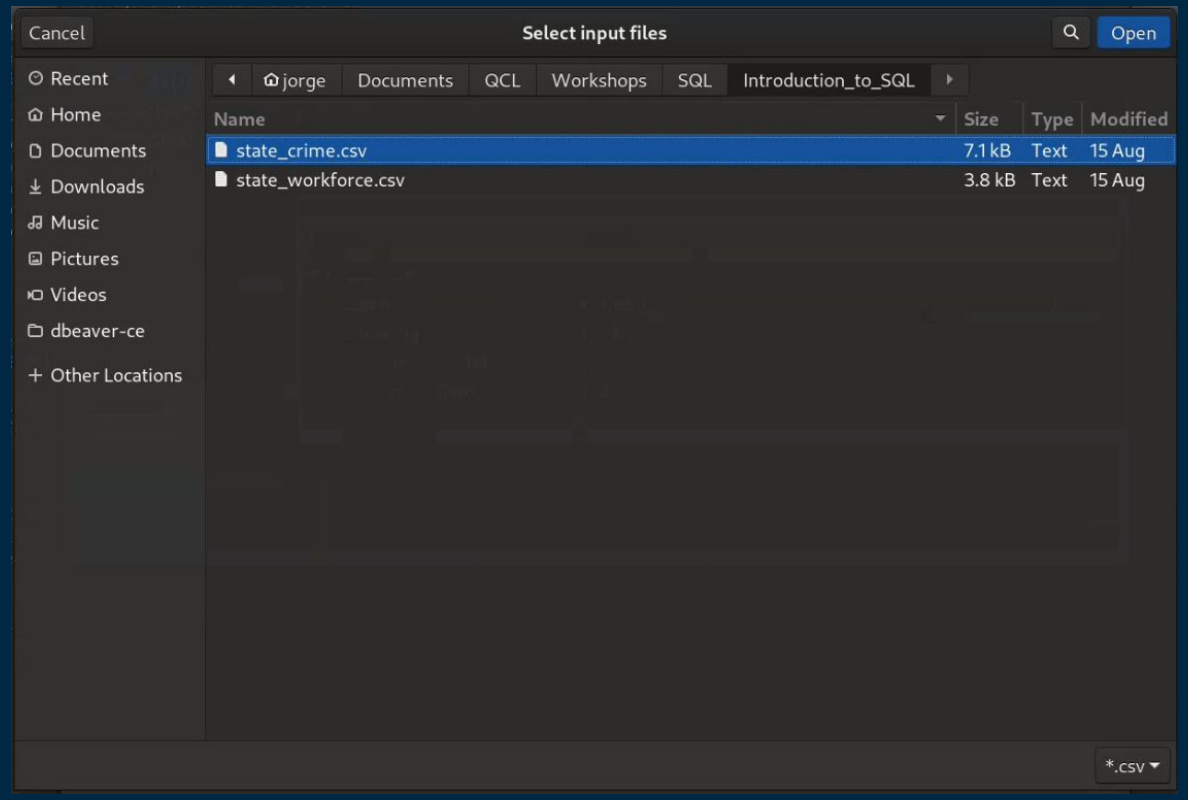
- Insert data into the new table with the file **state_people_insert.sql**

Import your data

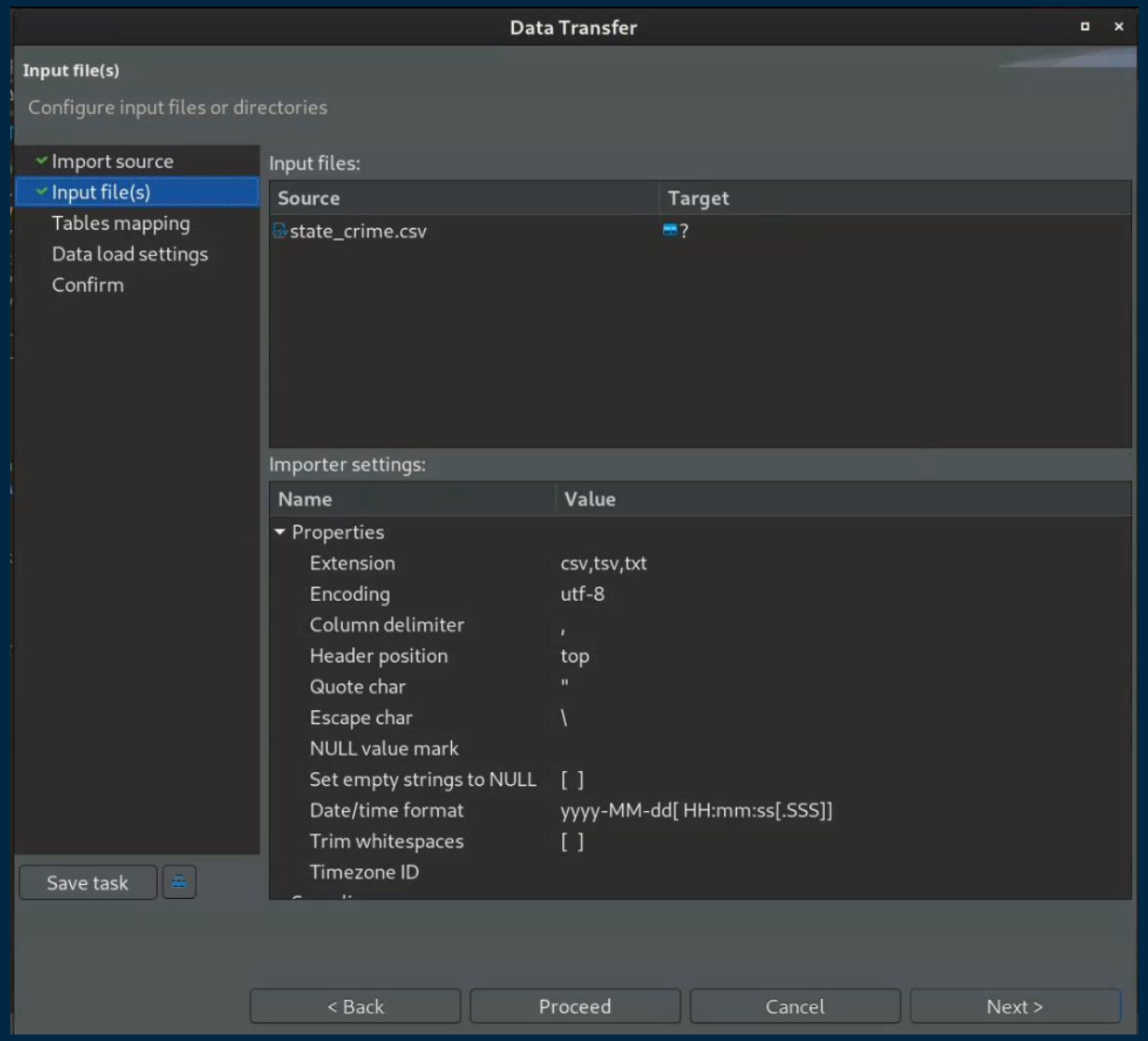


Import source

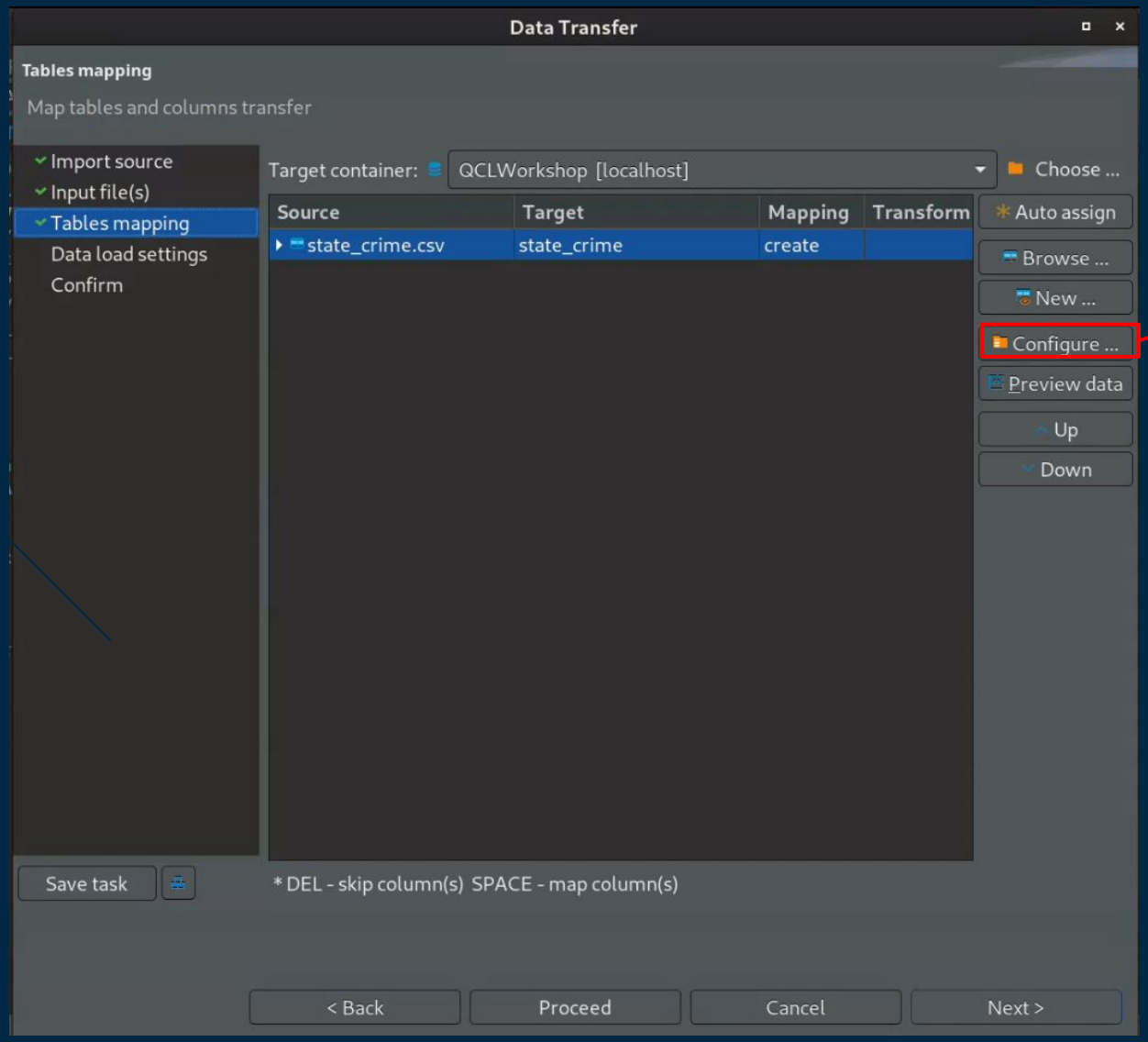




Input file

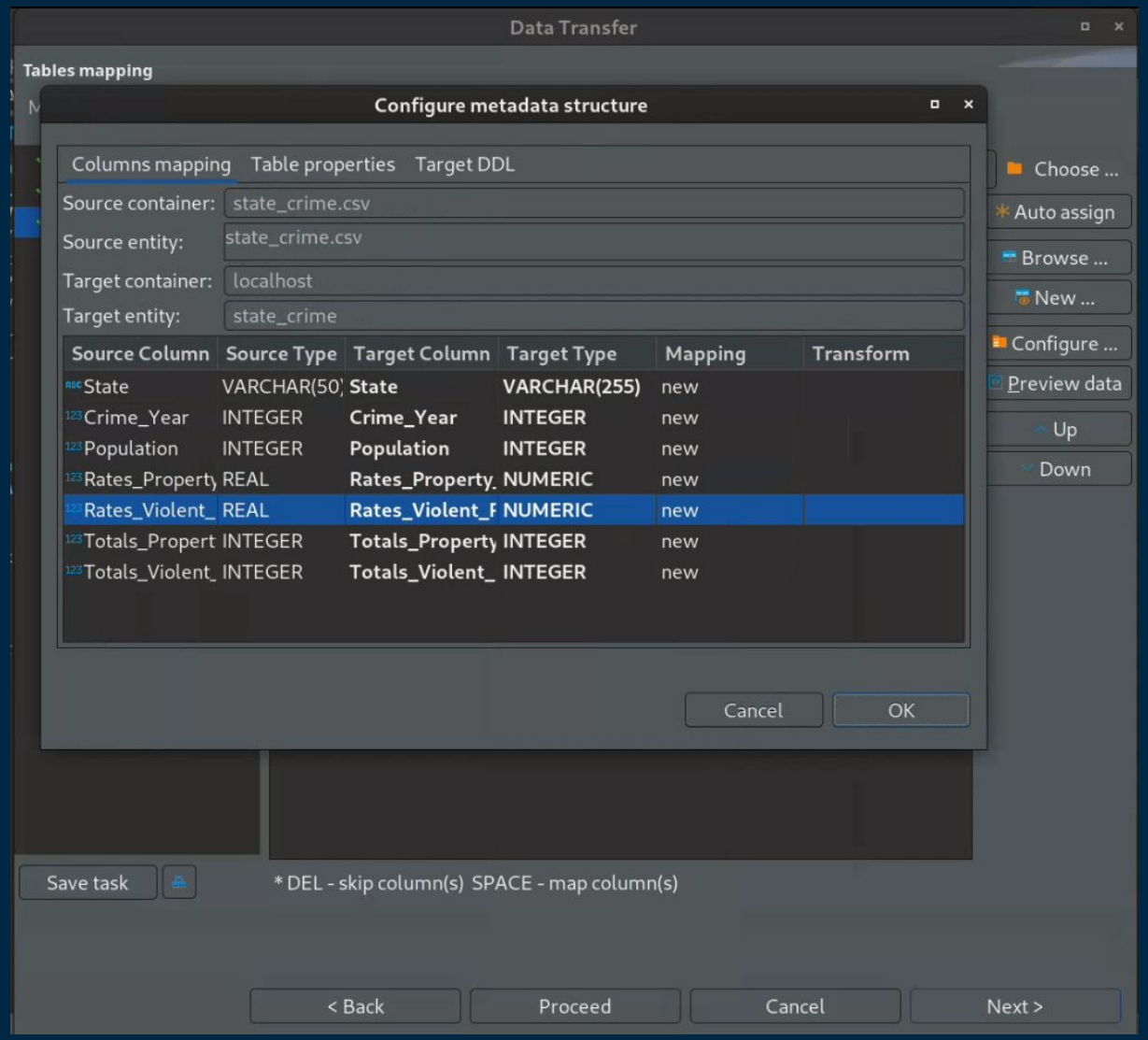

Data Transfer
Input file(s)
Configure input files or directories
Import source
Input file(s)
Tables mapping
Data load settings
Confirm
Input files:
Source
Target
state_crime.csv
?
Importer settings:
Name
Value
Properties
Extension
csv,tsv,txt
Encoding
utf-8
Column delimiter
,
Header position
top
Quote char
"
Escape char
\
NULL value mark
Set empty strings to NULL
[ ]
Date/time format
yyyy-MM-dd[HH:mm:ss[.SSS]]
Trim whitespaces
[ ]
Timezone ID
Save task
< Back
Proceed
Cancel
Next >

Tables mapping

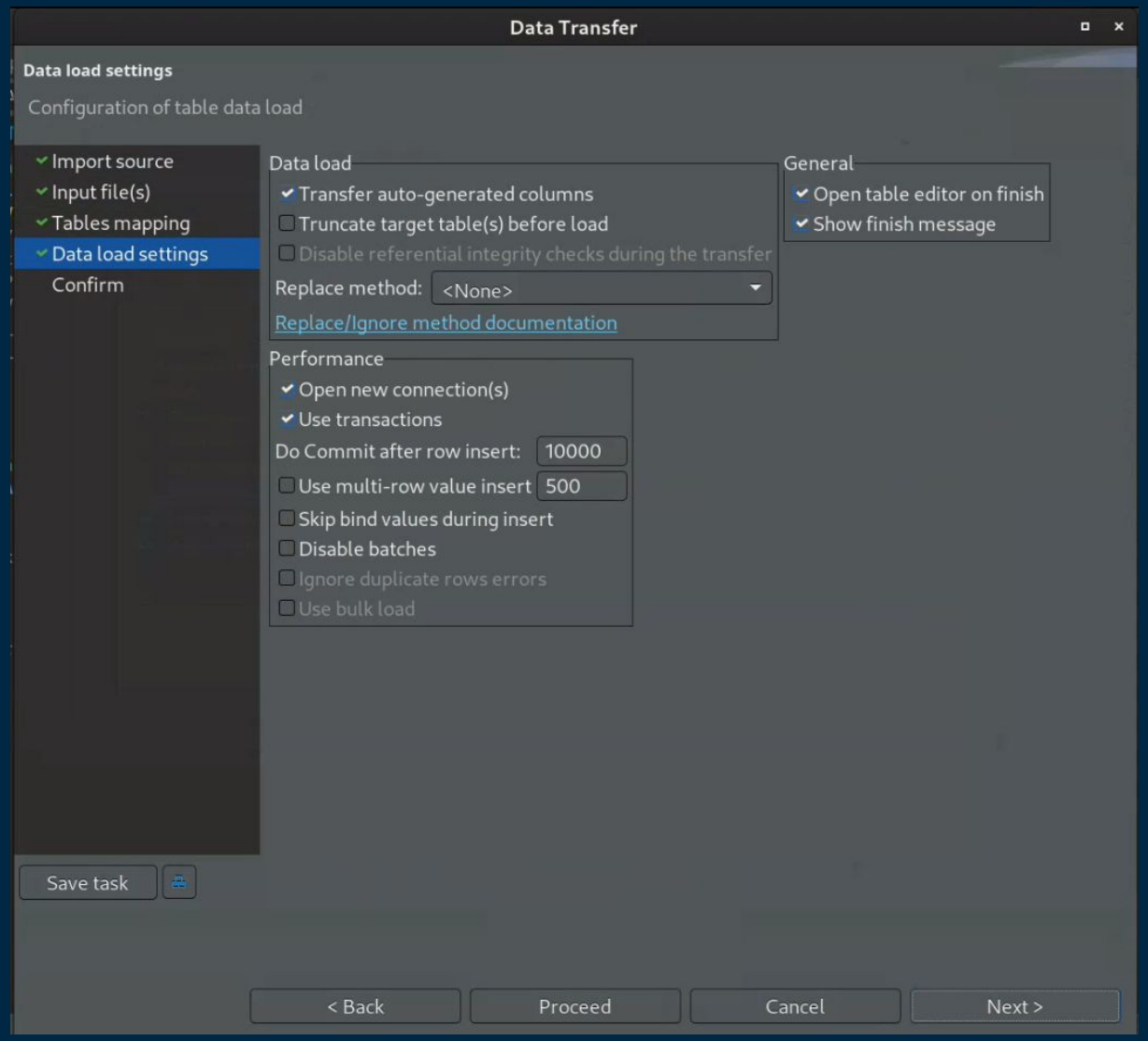


Configure the table mappings

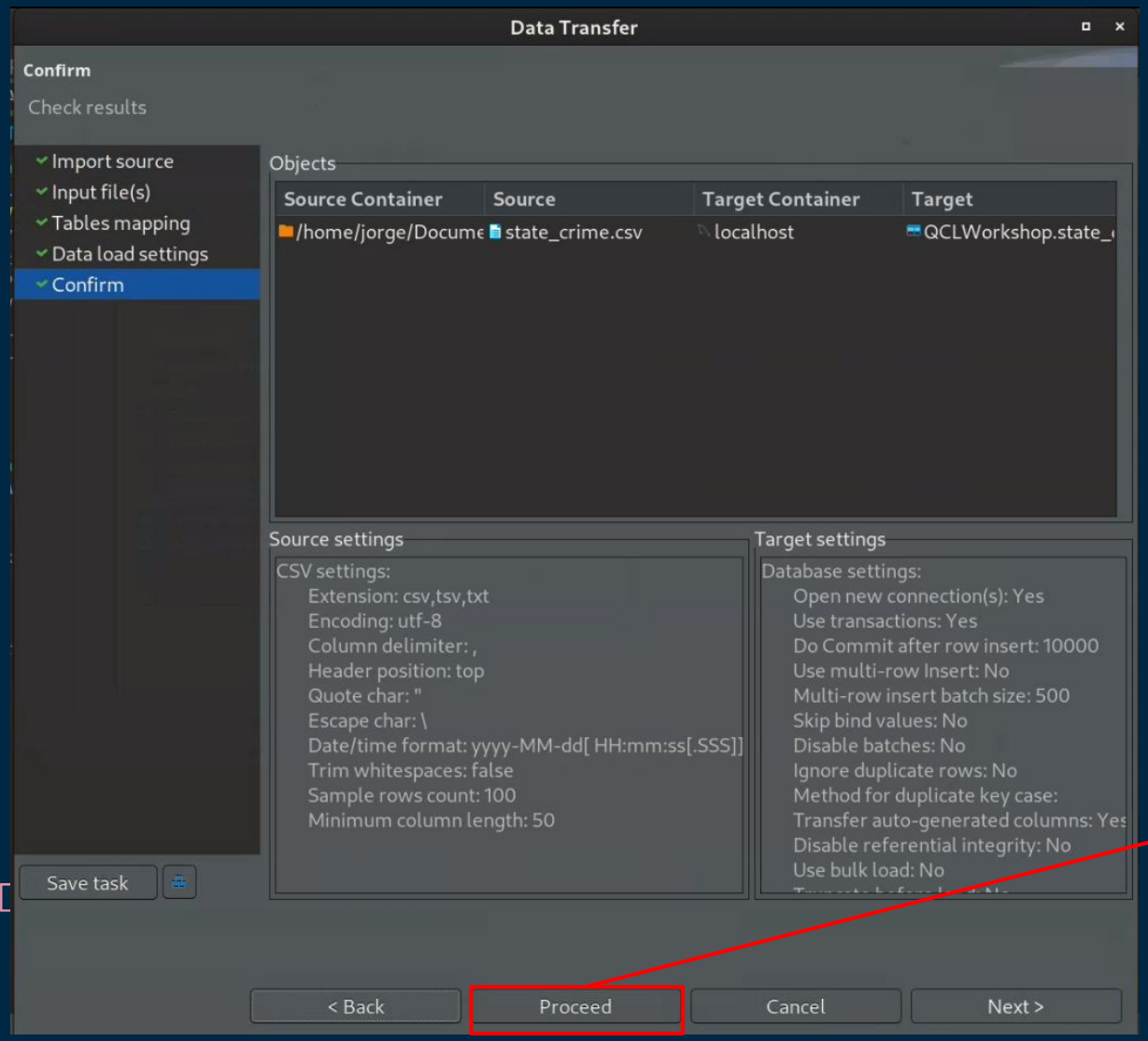
Configure mappings

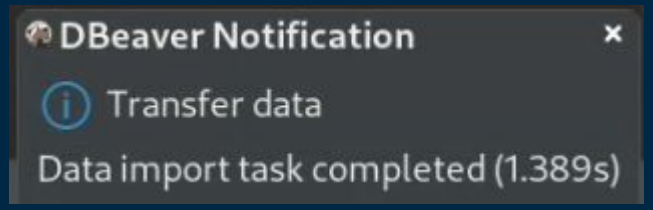


Data load settings



Confirmation





Click on “Proceed”

Hands-on

- Import the file named state_workforce

Writing Queries

Retrieving Data

```sql
-- Select statement with wildcard "*"
SELECT * FROM state_crime

-- Select using column names
SELECT Population,
       Rates_Property_Theft,
       Totals_Violent_Robbery
FROM state_crime
```

- Wildcard selects all columns from the given table
- You can also specify the columns by name

Hands-on

- Retrieve the data from the state_workforce table
  1. How many rows and attributes does this table have?

Filtering

```sql
-- Where clause
SELECT Population,
       Rates_Property_Theft,
       Totals_Violent_Robbery
FROM state_crime
WHERE Totals_Violent_Robbery >= 3000;
```

- Used to filter records based on a condition
- Use `AND` | `OR` operators to use multiple conditions

Filtering Operators

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
!=	Not equal
NOT	Negates the boolean value
BETWEEN	Whether a value is within a range
LIKE	Search for a pattern

- These are only some of the operators
- You can find all the operators here:
  - https://dev.mysql.com/doc/refman/8.0/en/non-typed-operators.html

Wildcards for LIKE

Wildcard	Description
%	Represents zero or more characters
_	Represents a single character

- Not all flavors of SQL support the same wildcards
- Take longer to run compared with using other operators

Starts with

DBeaver 22.2.2 - <localhost> Script-1

```
/* Using "%" wildcard with LIKE operator
 * Grab anything that starts with "South" */
SELECT State,
       Population
FROM state_crime
WHERE State LIKE 'South %';

-- Grab anything that ends with "Carolina"
SELECT State,
       Population
FROM state_crime
WHERE State LIKE '% Carolina';

-- Grab anything that contains
SELECT State,
       Population
FROM state_crime
WHERE State LIKE '% of %';
```

	State	Population
1	South Carolina	5,148,714
2	South Dakota	884,659
3	South Carolina	4,637,106
4	South Dakota	816,598
5	South Carolina	4,832,482
6	South Dakota	853,175

6 row(s) fetched - 24ms, on 2022-10-20 at 23:01:14

Ends with

DBeaver 22.2.2 - <localhost> Script-1

```
/* Using "%" wildcard with LIKE operator
 * Grab anything that starts with "South" */
SELECT State,
       Population
FROM state_crime
WHERE State LIKE 'South %';

-- Grab anything that ends with "Carolina"
SELECT State,
       Population
FROM state_crime
WHERE State LIKE '% Carolina';

-- Grab anything that contains
SELECT State,
       Population
FROM state_crime
WHERE State LIKE '% of %';
```

	State	Population
1	North Carolina	10,488,084
2	South Carolina	5,148,714
3	North Carolina	9,560,234
4	South Carolina	4,637,106
5	North Carolina	9,943,964
6	South Carolina	4,832,482

6 row(s) fetched - 21ms, on 2022-10-20 at 23:01:33

Contains

DBeaver 22.2.2 - <localhost> Script-1

```
/* Using "%" wildcard with LIKE operator
 * Grab anything that starts with "South" */
SELECT State,
       Population
FROM state_crime
WHERE State LIKE 'South %';

-- Grab anything that ends with "Carolina"
SELECT State,
       Population
FROM state_crime
WHERE State LIKE '% Carolina';

-- Grab anything that contains
SELECT State,
       Population
FROM state_crime
WHERE State LIKE '% of %';
```

	State	Population
1	District of Columbia	705,749
2	District of Columbia	604,912
3	District of Columbia	658,893

3 row(s) fetched - 21ms, on 2022-10-20 at 23:01:51

Sorting

```sql
-- Order by in descending order
SELECT State,
       Population,
       Totals_Violent_Robbery
FROM state_crime
WHERE Totals_Violent_Robbery >= 3000
ORDER BY Population DESC;
```

- Sorts the records by the given field in ascending order by default
- `ASC | DESC` for ascending or descending order

Hands-on

- Find out how many people on average take longer than 20 mins to get to work
- Sort the results to find out what state has the longest on average time it takes to get to work
- Modify your query to show only the records of New York, New Jersey, New Hampshire and New Mexico

Aggregate Functions

Function	Description
COUNT()	Counts the number of records
MAX()	Get maximum value
MIN()	Get minimum value
SUM()	Sums the field values
AVG()	Average of column values

Various data types

Numerical fields only

- Aggregate functions are used to summarize data

Syntax examples

```sql
-- Count example
SELECT COUNT(*)
FROM state_crime

-- Average example
SELECT AVG(Rates_Property_Theft)
FROM state_crime
WHERE Crime_Year = 2019
```

Subqueries and Joins

Subqueries

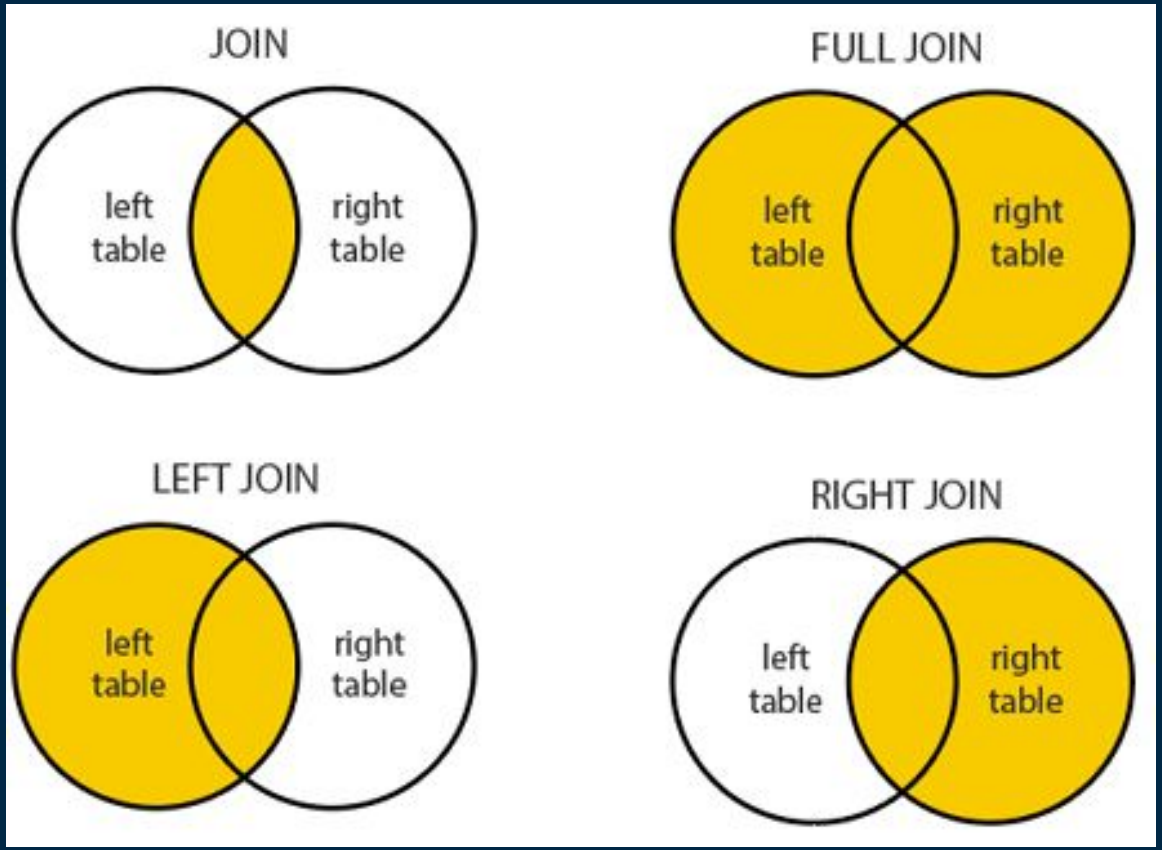
```sql
-- Subquery from computer data
SELECT State,
       Population,
       Totals_Violent_Robbery
FROM state_crime
WHERE State IN (SELECT State
                FROM state_computer_data
                WHERE Households_with_computer >= 93)
```

- Useful when you want to combine information from different tables
- SQL performs the innermost query first

Hands-on

- Find the maximum percentage of the people with education of high school or higher from states where either the rate of property theft is above 2,500 or the rate of violent robbery is above 30 during 2014

SQL Joins

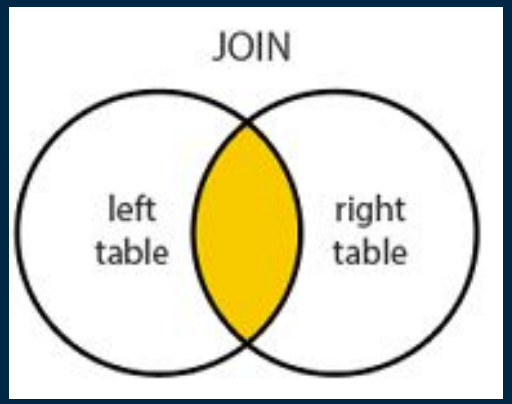

JOIN
left table
right table
FULL JOIN
left table
right table
LEFT JOIN
left table
right table
RIGHT JOIN
left table
right table

Aliases

```sql
-- Alias
SELECT scd.State,
       scd.Households_with_computer
FROM state_computer_data AS scd
```

(Inner) Join

```sql
-- Inner join
SELECT scd.State,
       scd.Households_with_computer,
       sc.Totals_Property_Theft,
       sc.Totals_Violent_Robbery
FROM state_computer_data AS scd
JOIN state_crime AS sc
ON scd.State = sc.State
```



Operation order

Select – Returns the data that was requested

From – choose a table to draw information from

Join – matches records from different tables

Where – filters data based on a condition

Group By – aggregates the data

Having – filers aggregated data

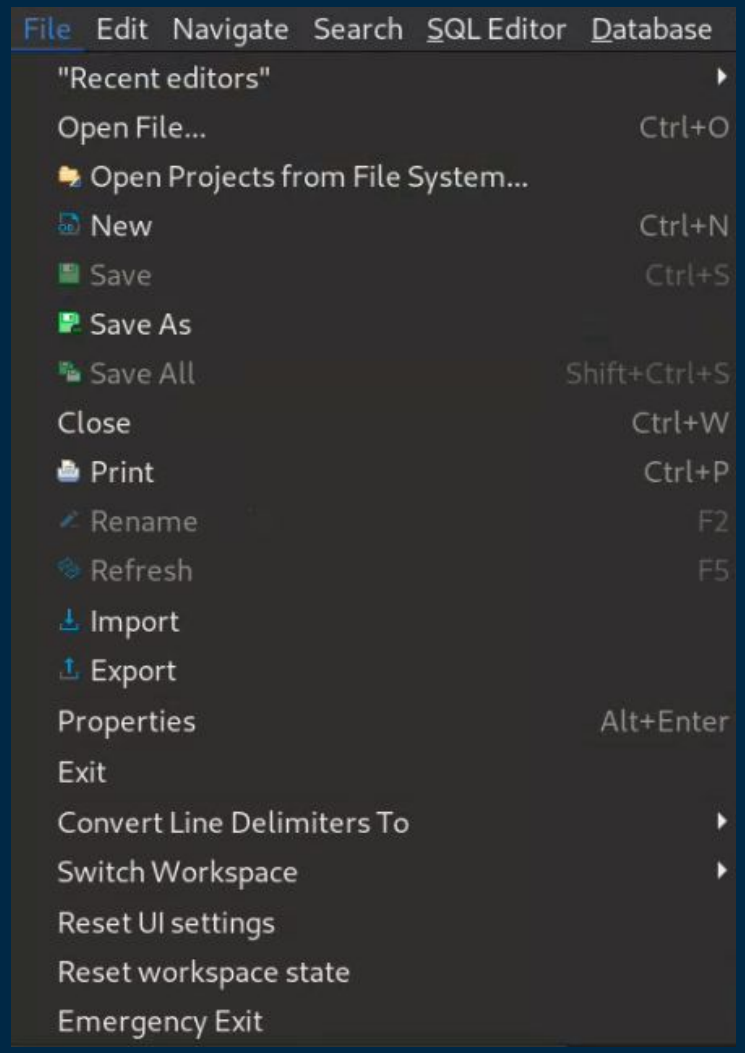
Order by – sorts the data

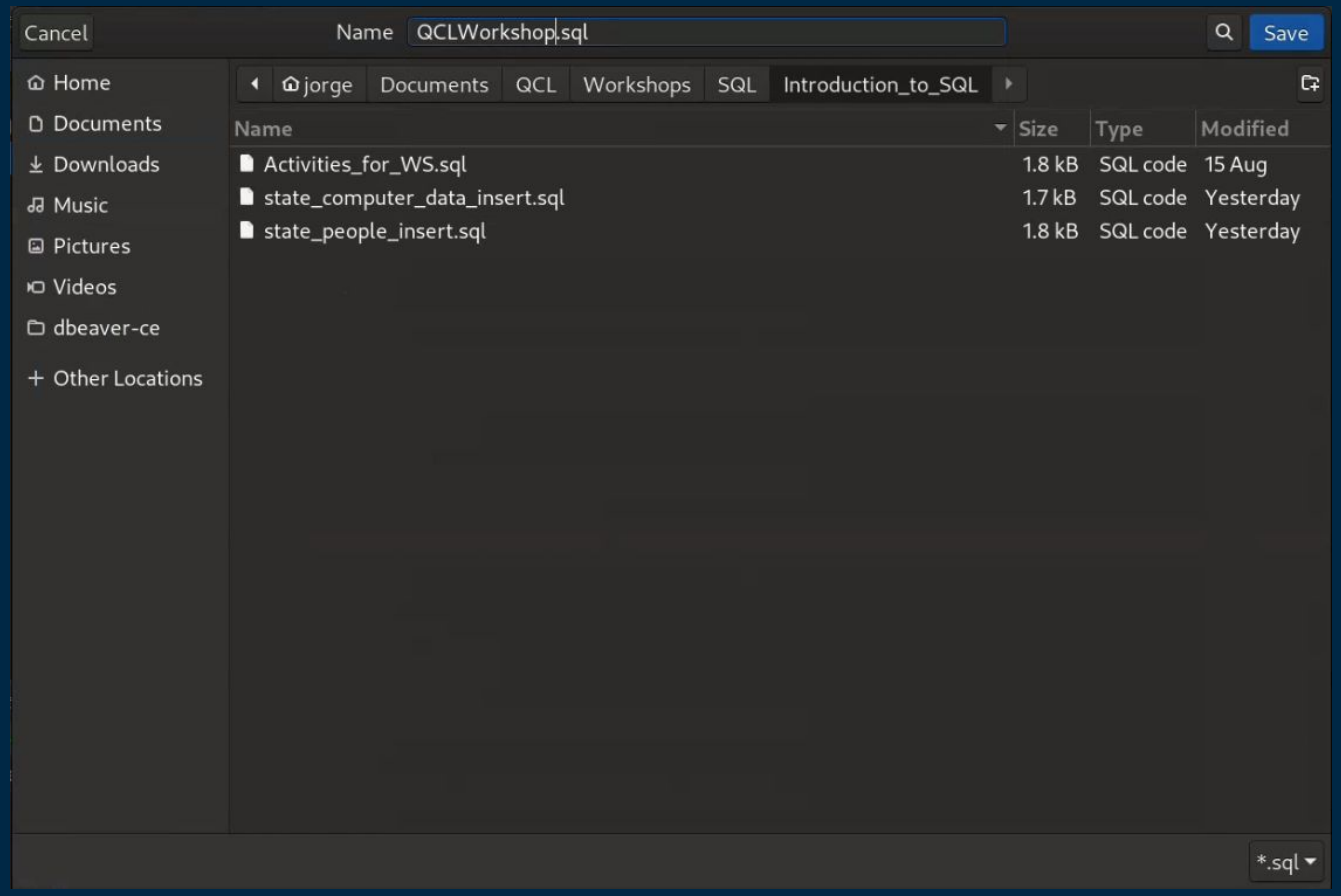
Limit – limit the number of rows returned

Hands-on

- Create an inner join using aliases with tables **state_workforce** and **state_people.** Make sure to view attributes:
  1. State
  2. Mean_Travel_Time_to_Work
  3. Employment_Firms_Total

Save your progress

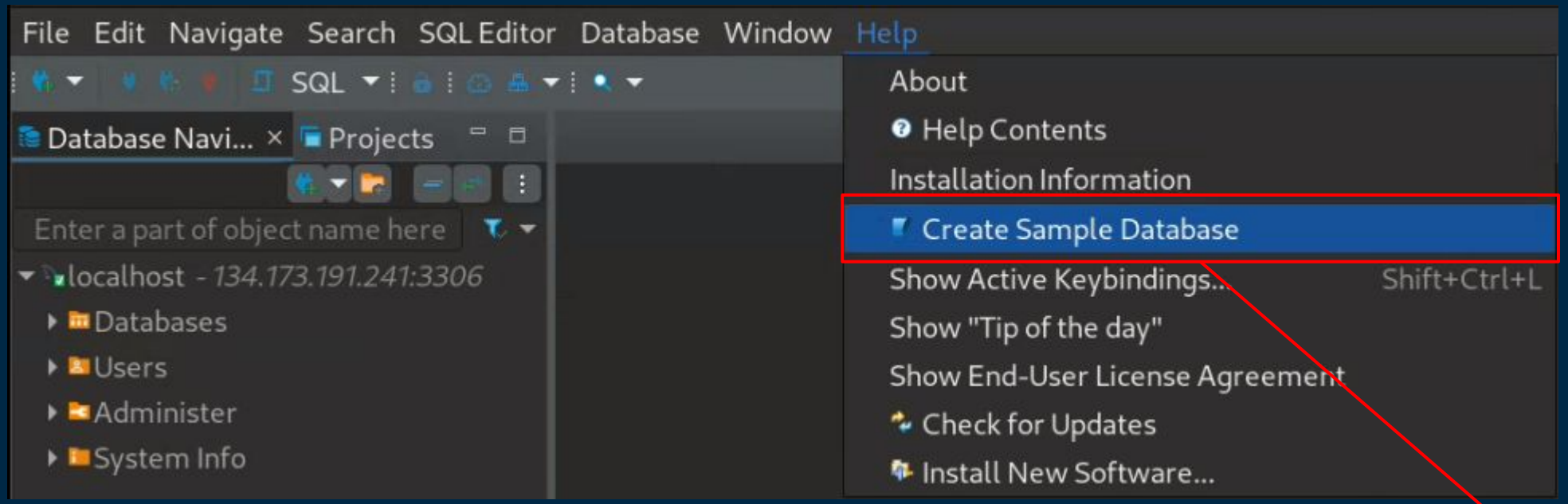




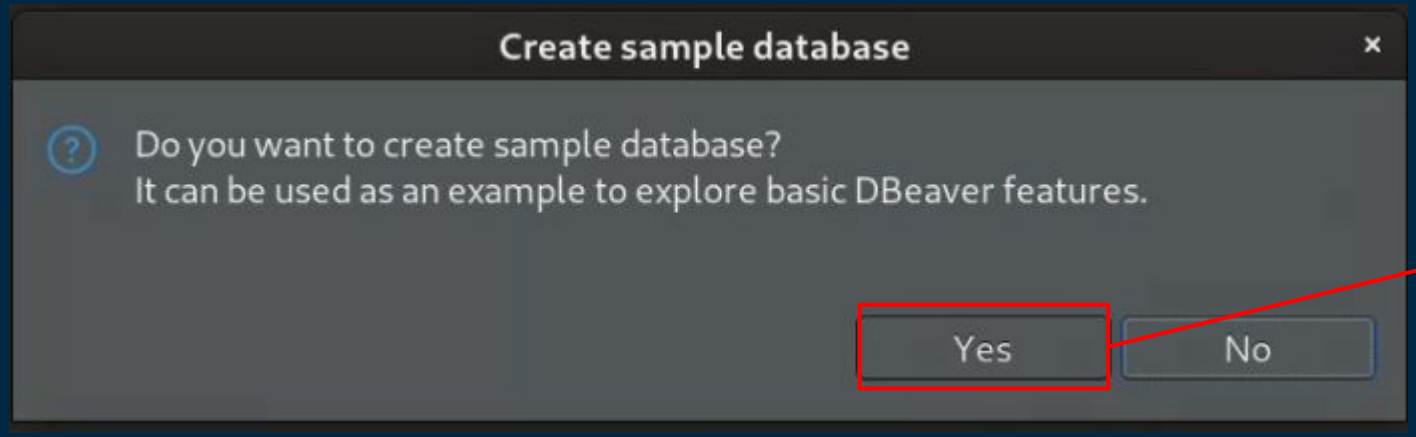
Send your results

- Finish today's hands-on activities
- Digital badge:
  - Create a sample database
  - How many artist collaborations are there in the Artist table? (keyword is "Feat.")
  - Show a table with the Artist name and their Album's titles as the only columns
  - What are the top 3 Albums with the most tracks?
- Send your hands-on activities and digital badge activities to
  - qcl@cmc.edu

Create Sample Database



Create a sample database



Confirmation

Resources

- Dbeaver Wiki - https://github.com/dbeaver/dbeaver/wiki
- W3schools - https://www.w3schools.com/sql/
- Codecademy - https://www.codecademy.com/learn/learn-sql

Best way to learn

- SQL Murder Mystery - https://mystery.knightlab.com/