# Module 3
# Processing MOVES Output

MOVES

Motor Vehicle Emission Simulator

March 2015

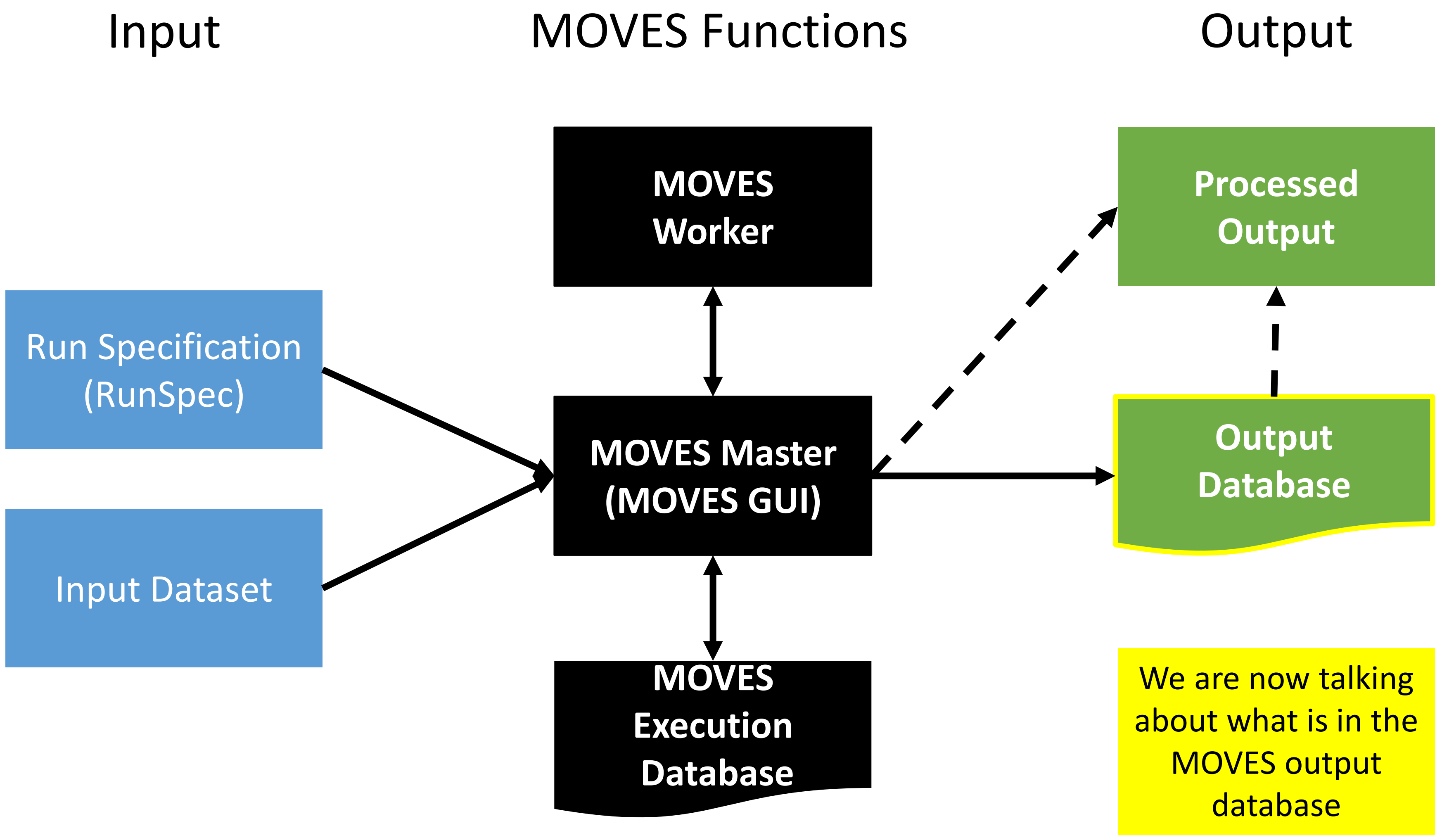
## Module Overview

- Describe what is contained in the MOVES output tables
- Use the Post-Processing Menu and post-processing MySQL scripts
- View and manipulate MOVES output using the MySQL Workbench
- Explore the results of the MOVES County-scale inventory exercise using MySQL Workbench

MOVES
Motor Vehicle Emission Simulator

EPA

# Using MOVES: Where Are We?



We are now talking about what is in the MOVES output database

# MOVES Output: General

- All of the results of a MOVES run are stored in MySQL database tables
- These results can be accessed by
  - Using MOVES summary reporter
  - Using MySQL query commands and/or the MySQL Workbench
  - Using Microsoft Access with a MySQL Open Database Connectivity (ODBC)
- Any table may be exported to other applications (e.g, MS Excel) for further processing

## Output Database Tables

- The MOVES output database contains numerous output tables with results of the run, input data, and other information about the run

- MOVESOutput table
  - Contains the quantity of emissions (by sourcetype, pollutant/process, etc., based on output detail selections made in the RunSpec)

- MOVESActivityOutput table
  - Contains the distance (useful to ensure no VMT was “lost”)

- MOVESRun table
  - Information about the run (e.g., date/time of run, domain and scale, units selected)

MOVES Motor Vehicle Emission Simulator

# Output Database Tables

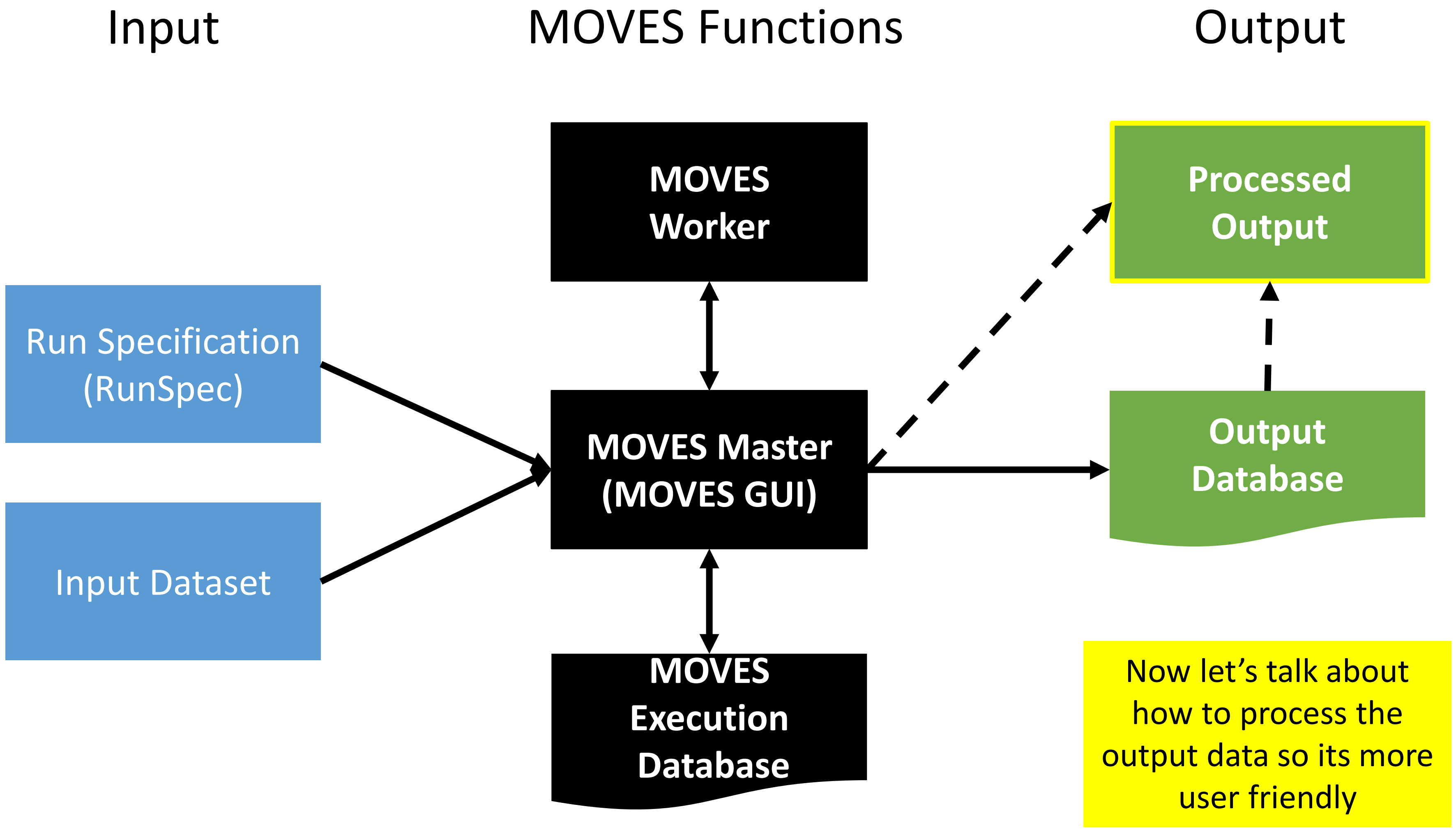
- Some tables are only populated when doing an Emission Rates run (not relevant to Inventory run)
  - RatePerDistance
  - RatePerVehicle
  - RatePerProfile
  - RatePerStart
  - RatePerHour
  - StartsPerVehicle

- Some tables are useful for diagnostic purposes
  - ActivityType
  - MOVESError
  - MOVESTablesUsed
  - MOVESWorkersUsed
  - MOVESEventLog

MOVES Motor Vehicle Emission Simulator

EPA

## Using MOVES: Where Are We?



Now let’s talk about how to process the output data so its more user friendly

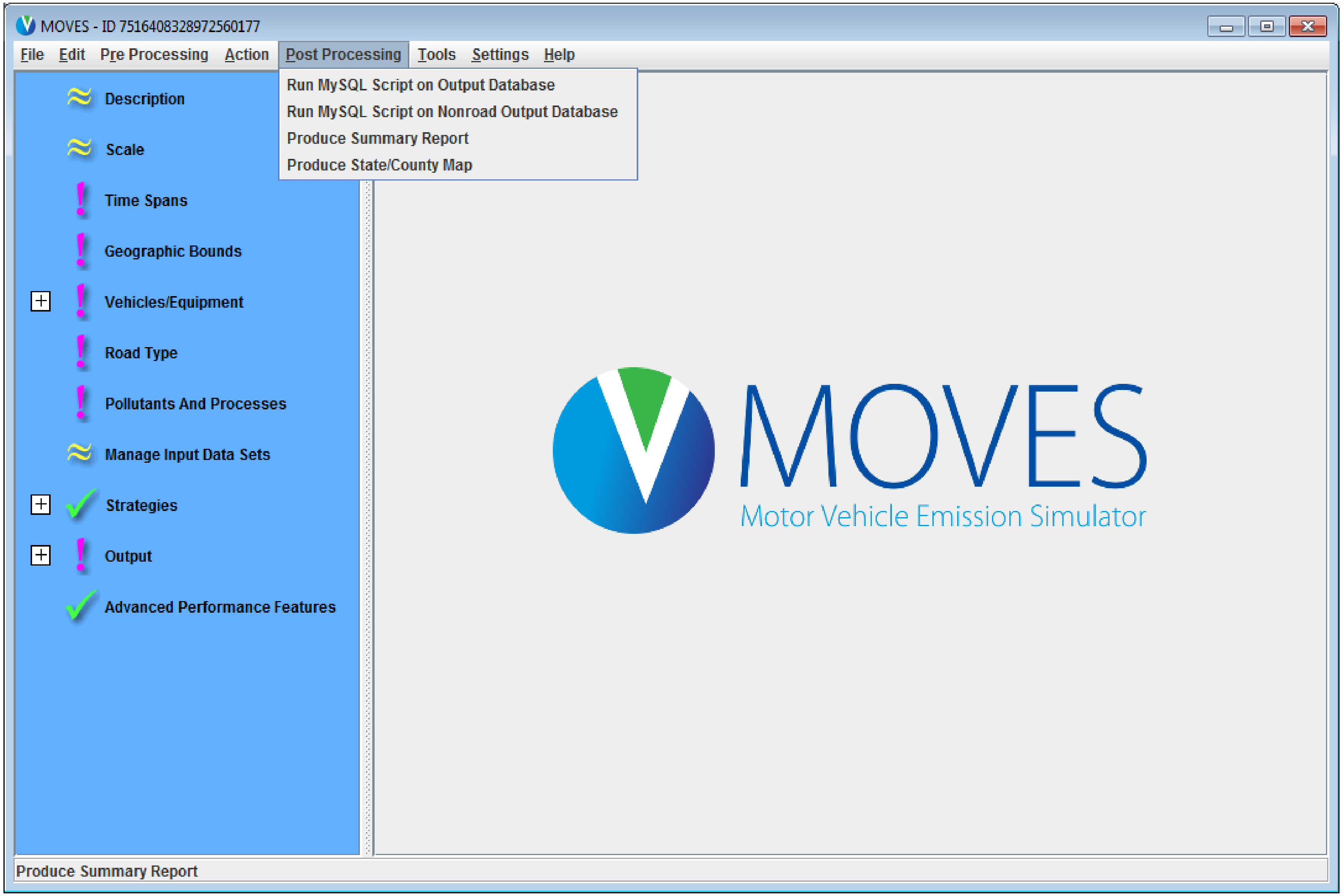
# MOVES Post-Processing Menu

- Use this menu option only after you have completed a run

- Select an existing output database using the RunSpec used to generate the results
  - If you are interesting in doing any post-processing from the Post-Processing Menu, it's often easiest if you immediately do so upon conclusion of the run

- Options for processing output include
  - Execute any MySQL scripts that come embedded in MOVES
  - Summarize results into text files
  - Graphically represent results in a county map

# MOVES Post-Processing Menu



## Post-Processing Scripts

- The scripts are applied to the current output database selected in the RunSpec
- You can select previous runs from the database using the MOVES Run Error Log window from the pull down Action menu
- There are several MySQL command scripts stored in the /database/OutputProcessingScripts folder of the MOVES application installation
- Users may write their own scripts and add them to the folder or add scripts obtained from other users

MOVES
Motor Vehicle Emission Simulator

EPA

## Post-Processing Scripts

- Read the script documentation before running MOVES
  - Project-scale scripts may require that you run MOVES in a particular way
  - Scripts may require running with a specific calculation type or certain units in the output

- Send ideas for useful scripts to mobile@epa.gov

MOVES Motor Vehicle Emission Simulator

EPA

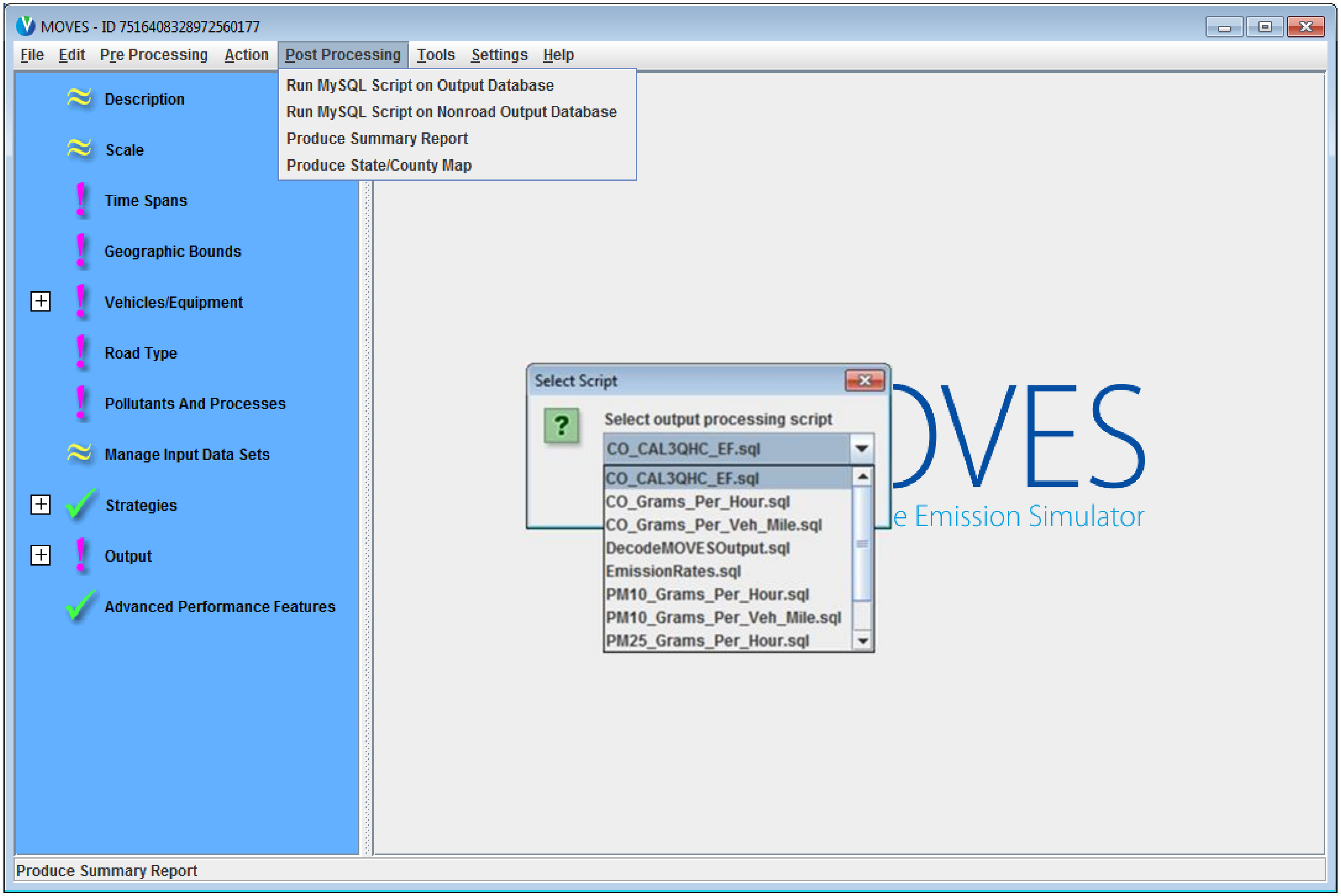
## Post-Processing Script

| Script Title | Description |
|---|---|
| DecodeMOVESOutput.sql | Decodes most key fields of MOVESOutput and MOVESActivityOutput tables |
| EmissionRates.sql | Produces an output table which reports the emission results in units of mass per distance |
| TabbedOutput.sql | Produces tab-delimited output suitable for reading into an EXCEL Spreadsheet from the MOVES MySQL database output tables |

# Post-Processing Scripts: Project Scale

| Script Title | Description |
|---|---|
| CO_CAL3QHC_EF.sql | Produces CO emission rates for use in the CAL3QHC air quality model |
| CO_Grams_Per_Hour.sql | Produces CO emission rates as grams per hour for each link (project-scale runs) |
| CO_Grams_Per_Veh_Mile.sql | Produces CO emission rates as grams per vehicle-mile for each link (project-scale runs) |
| PM10_Grams_Per_Hour.sql | Produces PM10 emission rates as grams per hour for each link (project-scale runs) |
| PM10_Grams_Per_Veh_Mile.sql | Produces PM10 emission rates as grams per vehicle-mile for each link (project-scale runs) |
| PM25_Grams_Per_Hour.sql | Produces PM2.5 emission rates as grams per hour for each link (project-scale runs) |
| PM25_Grams_Per_Veh_Mile.sql | Produces PM2.5 emission rates as grams per vehicle-mile for each link (project-scale runs) |

## Selecting a MySQL Output Processing Script



# Post-Processing Menu: Summary Report

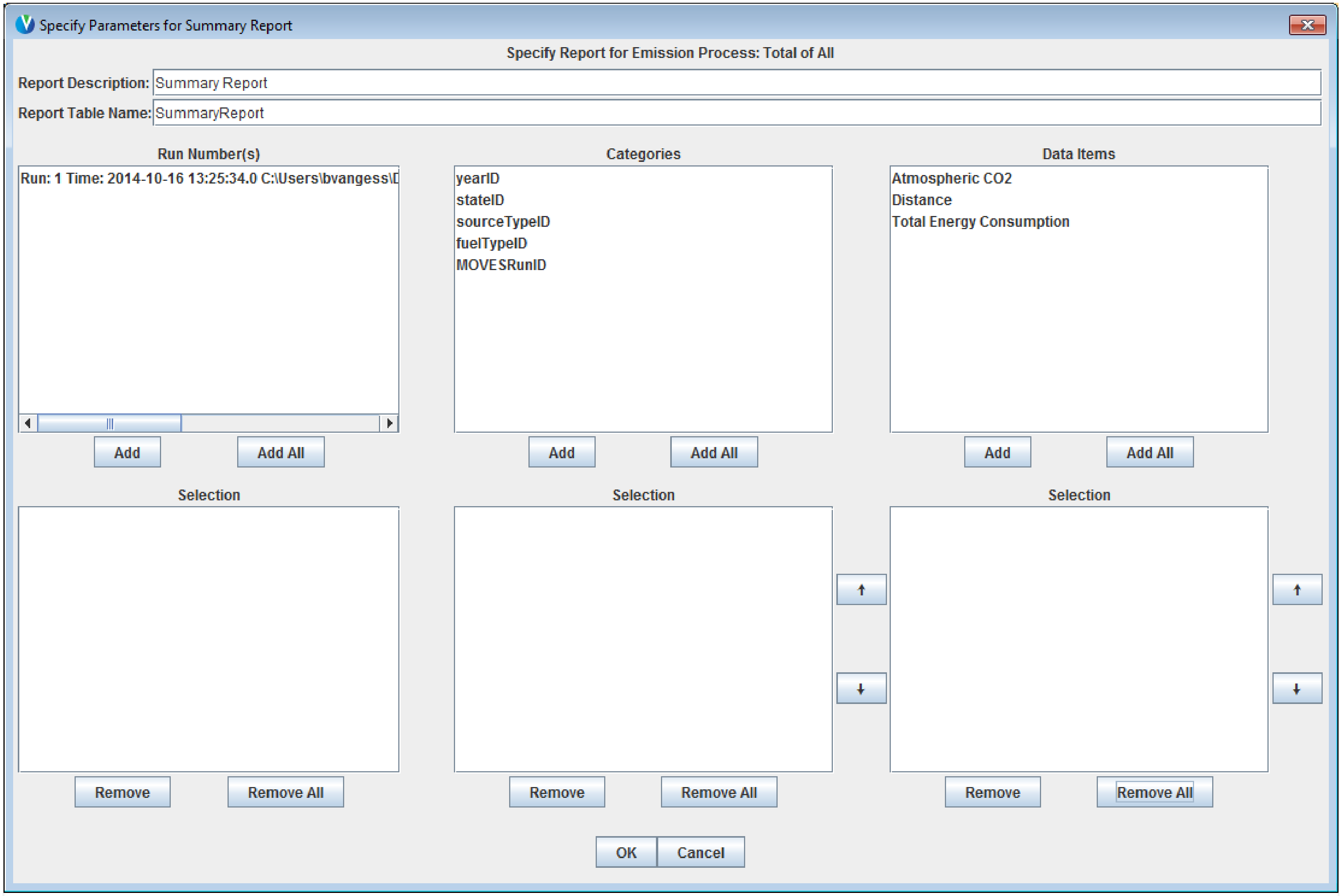
- Uses the output tables in the database referenced in the current RunSpec

- Reports output emissions and activity in varying levels of detail, based on selections by the user

MOVES
Motor Vehicle Emission Simulator

EPA

# Post-Processing Menu: Summary Report


Specify Parameters for Summary Report

Specify Report for Emission Process: Total of All

Report Description: Summary Report

Report Table Name: SummaryReport

Run Number(s)

Run: 1 Time: 2014-10-16 13:25:34.0 C:\Users\bvangess\D

Categories

yearID
stateID
sourceTypeID
fuelTypeID
MOVESRunID

Data Items

Atmospheric CO2
Distance
Total Energy Consumption

# Other Post-Processing Options

- Export the data using the MySQL script
- Export the data using the MySQL Workbench
  - CSV
  - MS Excel
  - PLIST
- Use the data from MS Access using the ODBC

MOVES
Motor Vehicle Emission Simulator

EPA

# MySQL Workbench

- Provided with the MOVES model suite
- Windows tool for viewing databases, executing queries, and editing tables
- Results can be exported as .csv or MS Excel files
- Built-in Help files
- Query history recorded, so you can repeat queries without retyping them
- Tables can be edited directly, rather than using MySQL commands

MOVES
Motor Vehicle Emission Simulator

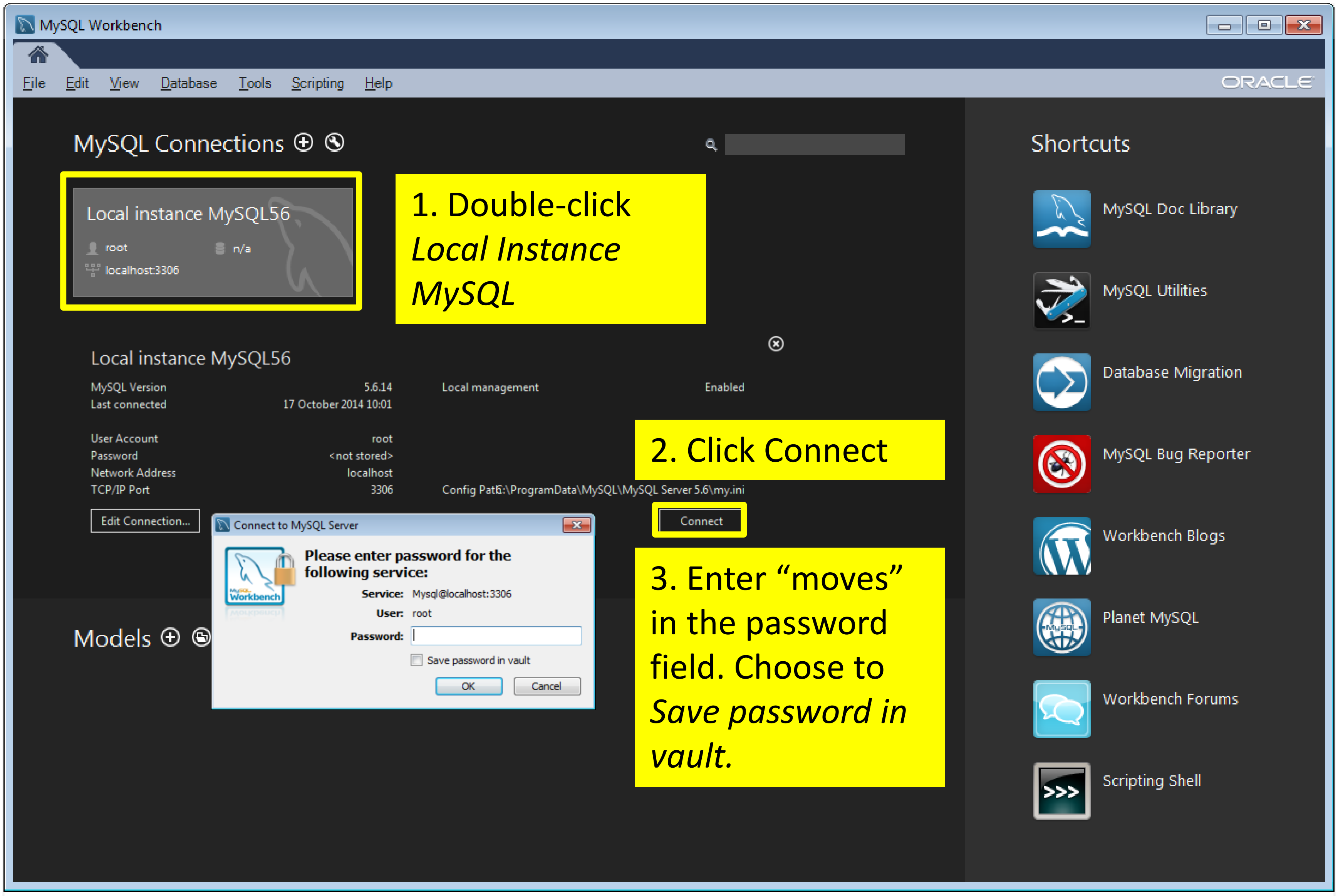
EPA

# Exploring MOVES Databases with MySQL

- Introduction to Workbench and hands-on practice using our Module 3 County-scale inventory output database

- Instructions for Exploring MOVES Databases Exercise:
  - Open MySQL Workbench
  - Start/Programs/MySQL/MySQL Workbench
  - Click *Local instance MySQL56*
  - Click *Connect*
  - Enter “moves” for Password and click *OK*

# MySQL Workbench – Basics: Password



1. Double-click *Local Instance MySQL*

Local instance MySQL56

| | | | |
|---|---|---|---|
| MySQL Version | 5.6.14 | Local management | Enabled |
| Last connected | 17 October 2014 10:01 | | |
| User Account | root | | |
| Password | <not stored> | | |
| Network Address | localhost | | |
| TCP/IP Port | 3306 | Config Path: | C:\ProgramData\MySQL\MySQL Server 5.6\my.ini |

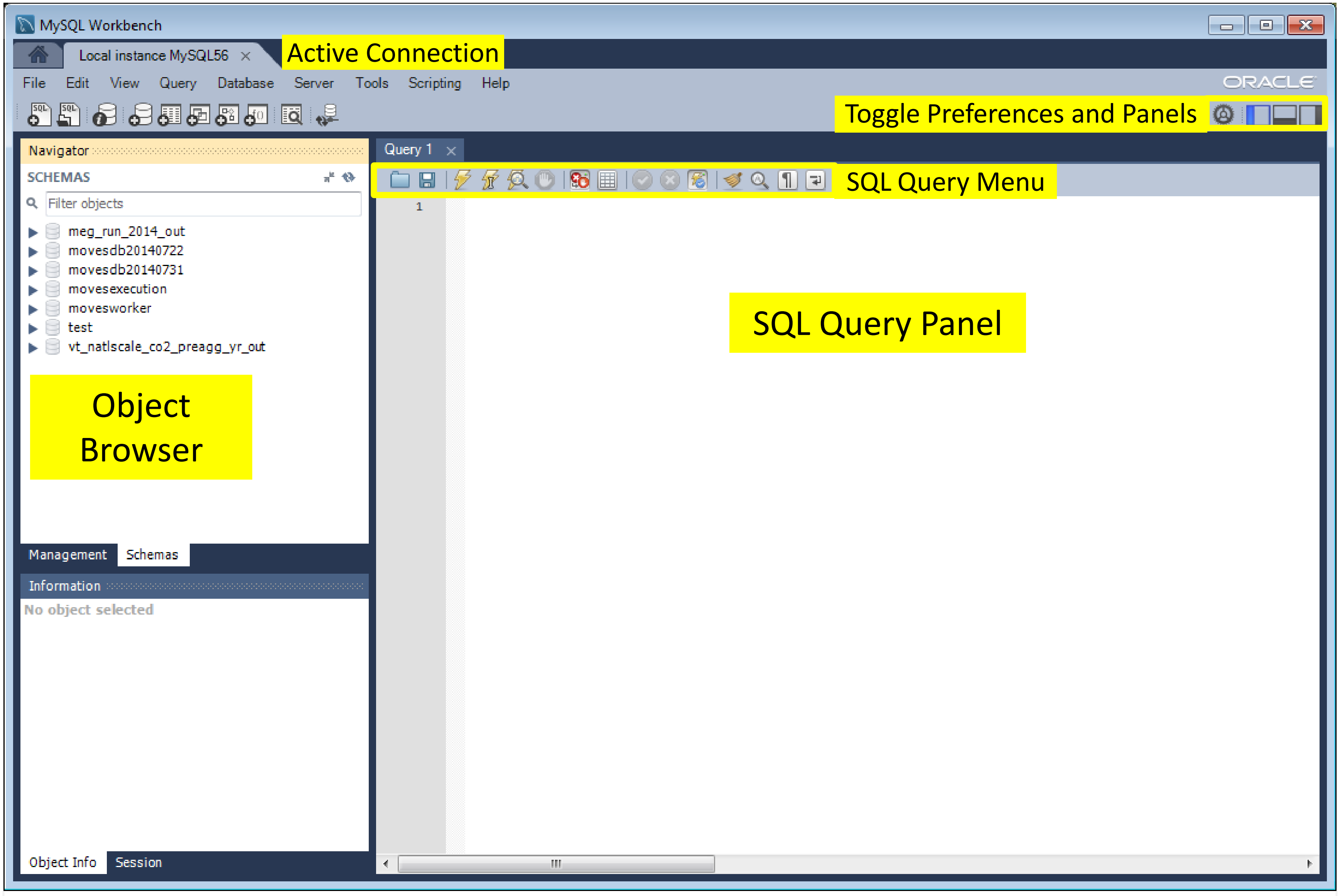
2. Click Connect

Please enter password for the following service:

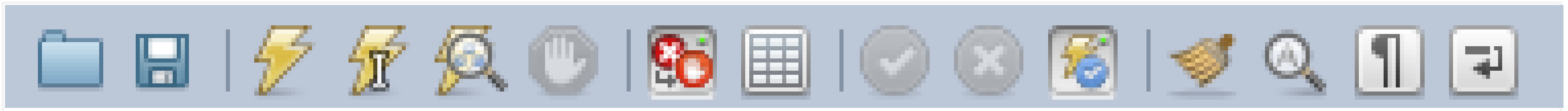
Service: Mysql@localhost:3306
User: root
Password:
☐ Save password in vault

3. Enter “moves” in the password field. Choose to *Save password in vault.*

# MySQL Workbench – Layout



# MySQL Workbench – Basics



From left to right, these buttons are:

**Open a SQL Script File**: Loads a saved SQL script to be ready for execution.

**Save SQL Script to File**: Saves the current SQL script to a specified file.

**Execute SQL Script**: Executes the selected portion of the query, or the entire query if nothing is selected.

**Execute Current SQL script**: Execute the statement under the keyboard cursor.

**Explain**: Execute the **EXPLAIN** command on the query after the keyboard cursor.

**Stop the query being executed**: Halts execution of the currently executing SQL script.

**Toggle whether execution of SQL script should continue after failed statements**: If the red "breakpoint" circle is displayed, the script terminates on a statement that fails. If the button is depressed so that the green arrow is displayed, execution continues past the failed code, possibly generating additional result sets.

**Toggle row limit:** turns on and off the row limit (default 1000) on the result set

**Commit**: Commits the current transaction. Note: All query tabs in the same connection share the same transactions. To have independent transactions, a new connection must be opened.

**Rollback**: Rolls back the current transaction. Note: All query tabs in the same connection share the same transactions. To have independent transactions, a new connection must be opened.

**Toggle Auto-Commit Mode**: If selected, each statement will be committed independently. Note: All query tabs in the same connection share the same transactions. To have independent transactions, a new connection must be opened.

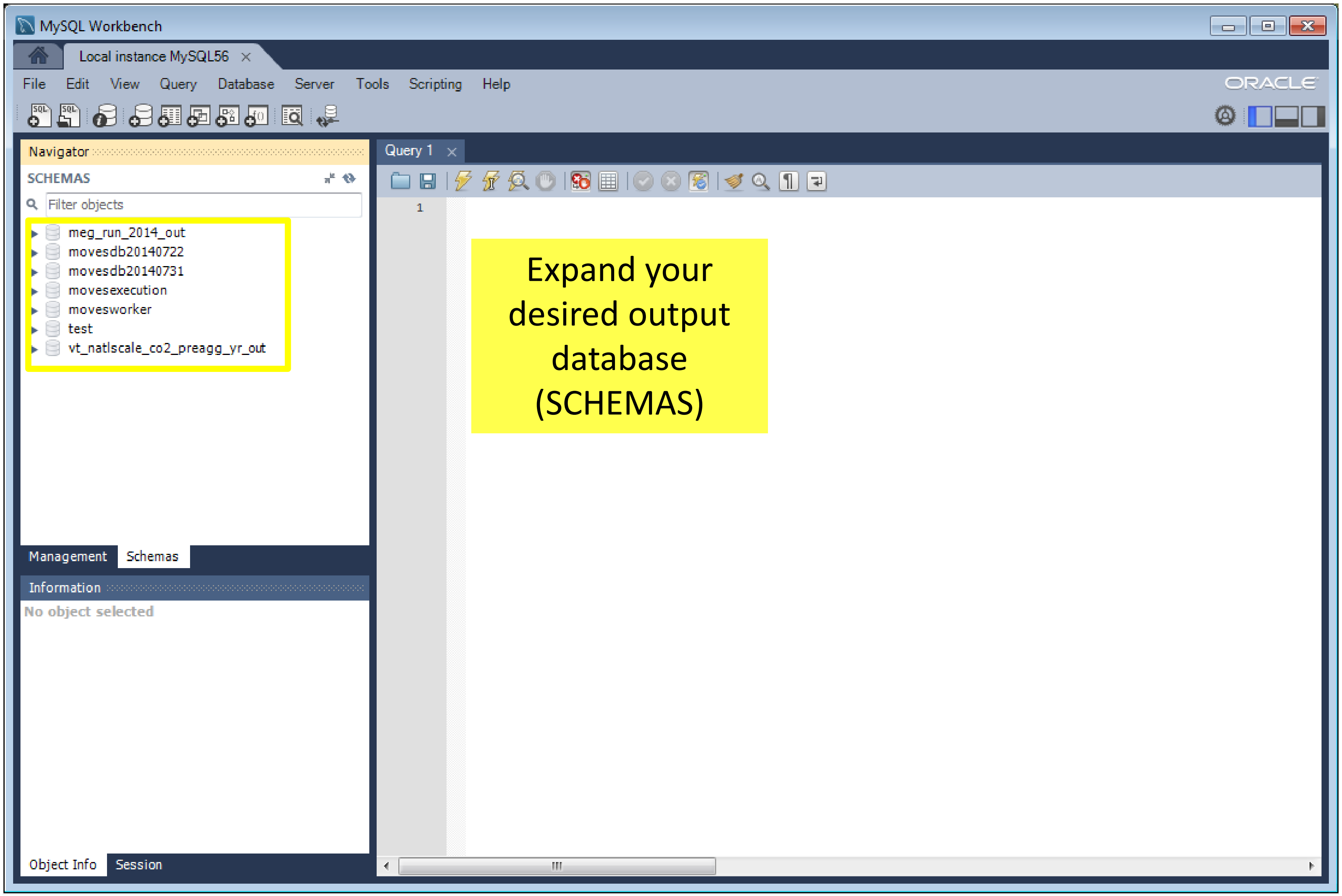
**Beautify SQL**: Beautify/reformat the SQL script.

**Find panel**: Show the Find panel for the editor.

**Invisible characters**: Toggle display of invisible characters, such as newlines, tabs or spaces.

**Wrapping**: Toggles the wrapping of long lines in the SQL editor window.

# MySQL Workbench – Basics

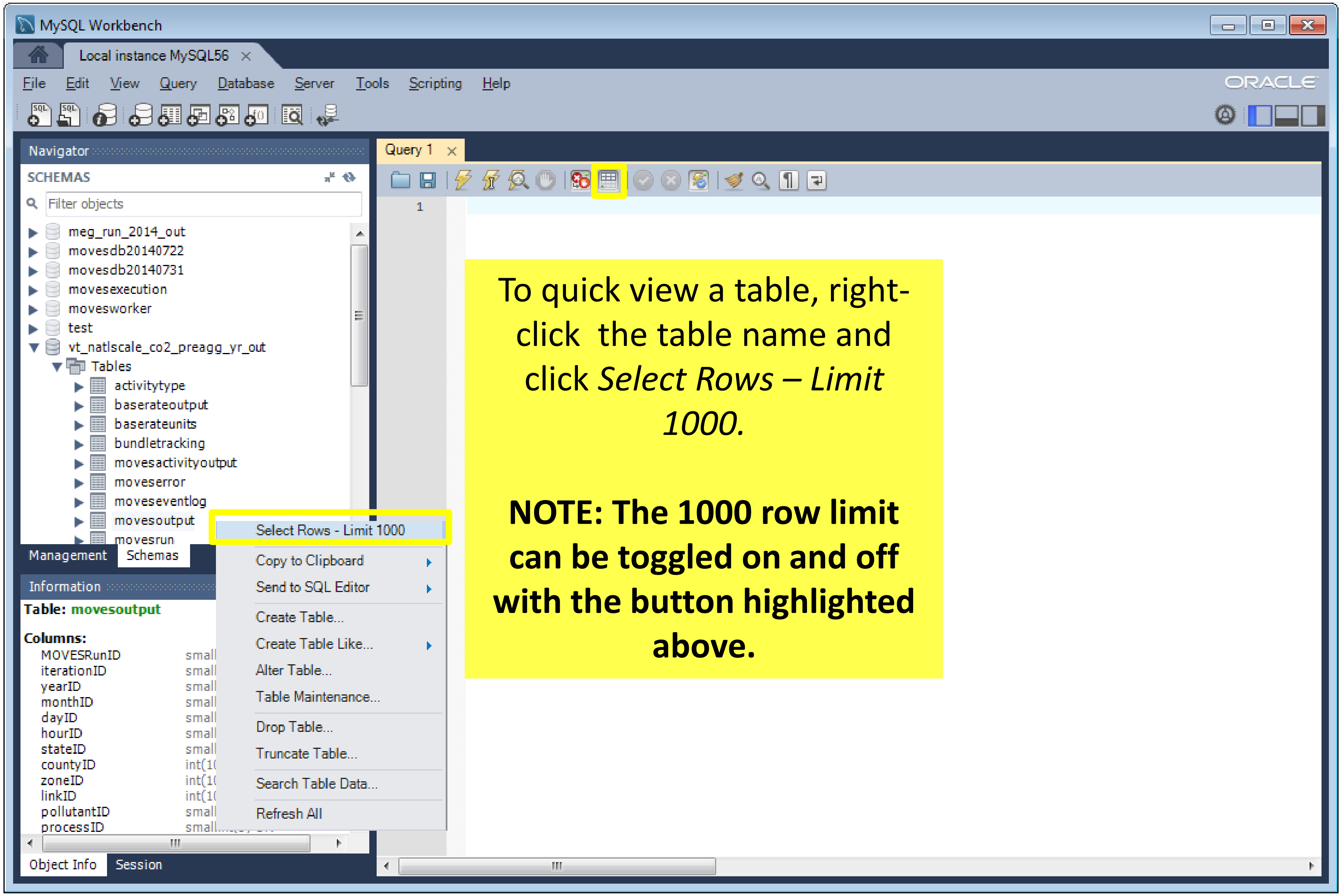


meg_run_2014_out
movesdb20140722
movesdb20140731
movesexecution
movesworker
test
vt_natlscale_co2_preagg_yr_out

No object selected

Expand your desired output database (SCHEMAS)

# MySQL Workbench – Basics



Information

**Table: movesoutput**

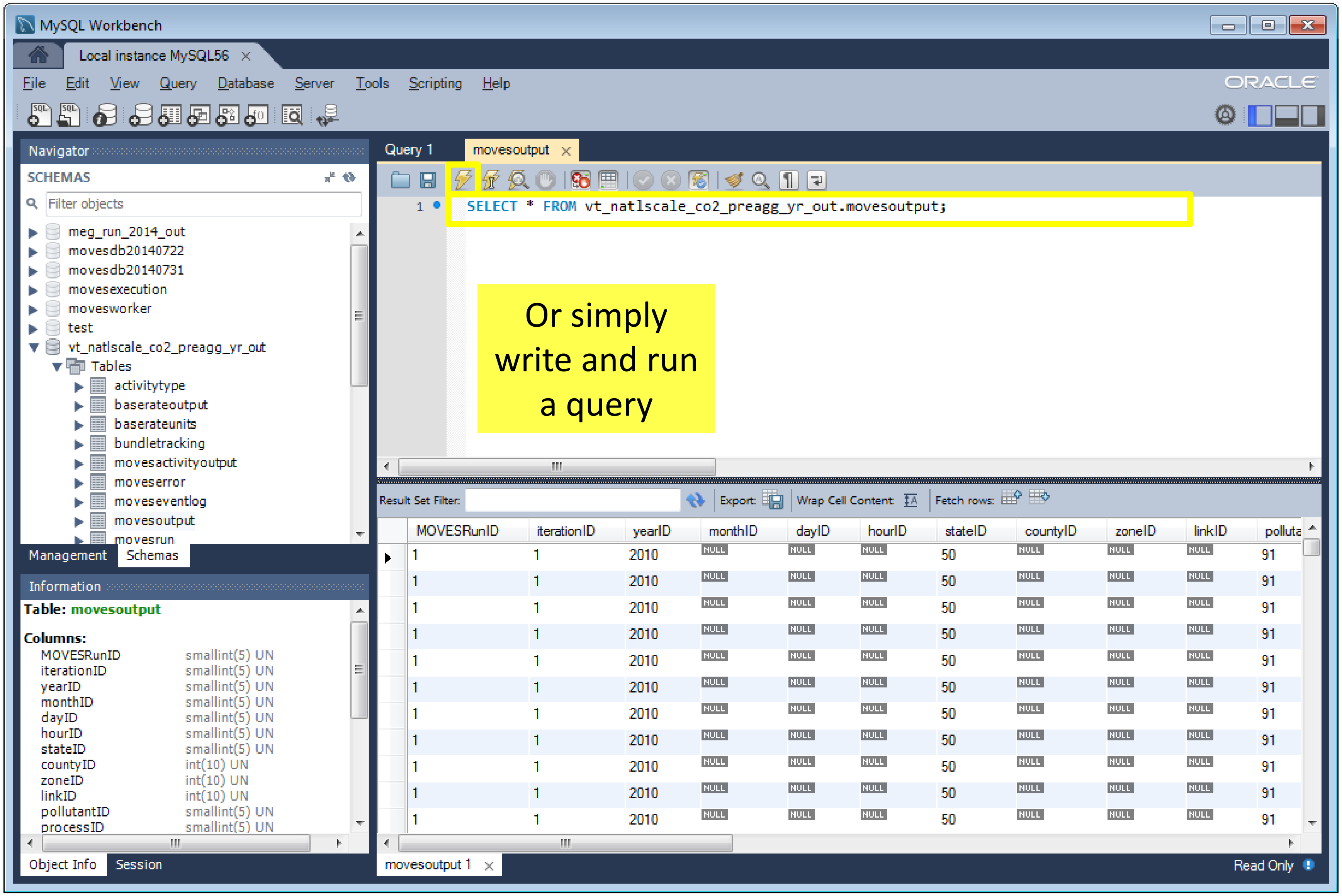
**Columns:**

| | |
|---|---|
| MOVESRunID | small |
| iterationID | small |
| yearID | small |
| monthID | small |
| dayID | small |
| hourID | small |
| stateID | small |
| countyID | int(1 |
| zoneID | int(1 |
| linkID | int(1 |
| pollutantID | small |
| processID | small |

To quick view a table, right-click the table name and click *Select Rows – Limit 1000.*

**NOTE: The 1000 row limit can be toggled on and off with the button highlighted above.**

# MySQL Workbench – Basics



```
SELECT * FROM vt_natlscale_co2_preagg_yr_out.movesoutput;
```

Or simply write and run a query

- meg_run_2014_out
- movesdb20140722
- movesdb20140731
- movesexecution
- movesworker
- test
- vt_natlscale_co2_preagg_yr_out
  - Tables
    - activitytype
    - baserateoutput
    - baserateunits
    - bundletracking
    - movesactivityoutput
    - moveserror
    - moveseventlog
    - movesoutput
    - movesrun

Result Set Filter: Export: Wrap Cell Content: Fetch rows:

| MOVESRunID | iterationID | yearID | monthID | dayID | hourID | stateID | countyID | zoneID | linkID | polluta |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |

**Table: movesoutput**

**Columns:**

| | |
|---|---|
| MOVESRunID | smallint(5) UN |
| iterationID | smallint(5) UN |
| yearID | smallint(5) UN |
| monthID | smallint(5) UN |
| dayID | smallint(5) UN |
| hourID | smallint(5) UN |
| stateID | smallint(5) UN |
| countyID | int(10) UN |
| zoneID | int(10) UN |
| linkID | int(10) UN |
| pollutantID | smallint(5) UN |
| processID | smallint(5) UN |

## Simple MySQL Queries: Syntax

- Queries are used to display or aggregate output database data

- A query can be typed into the Query Area

- **The order and arrangement of query commands is important!**
  - Use the "**Introduction to MYSQL Workbench Syntax**" handout.
  - Syntax must be used in the order given on the handout/next slide.
  - Not all commands are needed to complete a query.
  - To identify the table to be queried from, the syntax is database name followed by "." and the table name.
  - Commas can be used to separate multiple fields following a command
  - All of this will be covered more on the next few slides

- Click the ⚡ button or hit CTL/ENTER to execute queries

# Simple MySQL Queries: Commands

| Syntax | Function | Example |
|---|---|---|
| **SELECT** | Selects one or more data fields, separated by commas. A "*" following the SELECT command indicates "all fields" | SELECT sourcetypeID, activity |
| **SUM** | Adds up the data in the field indicated in parentheses and creates a new field for the results (note required spacing) | SUM(activity) |
| **AS** | Used with the SUM command to name the new field containing the results of the SUM command (optional). | AS grams |
| **FROM** | Indicates the database and table the SELECT command is pulling from. Database and table must be separated by period. | FROM co_2015_out.movesoutput |
| **WHERE** | Used to specify the value(s) of the field to be selected. | WHERE sourcetypeID=21 |
| **AND** | Used to specify more than one field when using the WHERE command. | WHERE source typeiD=21 AND dayID=2 |
| **GROUP BY** | Groups data together by the field indicated. | GROUP BY movesrunid |
| **ORDER BY** | Specifies the order of data presented in the field(s) following the command (e.g., will rank data high to low). | ORDER BY pollutantid |

MOVES Motor Vehicle Emission Simulator

EPA

# Examples of MySQL Queries

```
SELECT * FROM lake_2015_training_out.movesactivityoutput;
```

- Selects data from **all field columns** from the movesactivityoutput table of the Lake_2015_training_out database

```
SELECT *, SUM(emissionQuant) FROM
lake_2015_training_out.movesoutput
GROUP BY movesrunid;
```

- Selects all field columns from the movesoutput table and adds up the emissionQuant field across all source types, pollutant types, etc. Groups the results by movesrunid

```
SELECT movesrunid, emissionquant, SUM(emissionquant) FROM
lake_2015_training_out.movesoutput
GROUP BY movesrunid;
```

- Same result as above, but only selects two field columns (movesrunid and emissionsquant) instead of all data

MOVES
Motor Vehicle Emission Simulator

EPA

# Simple MySQL Queries

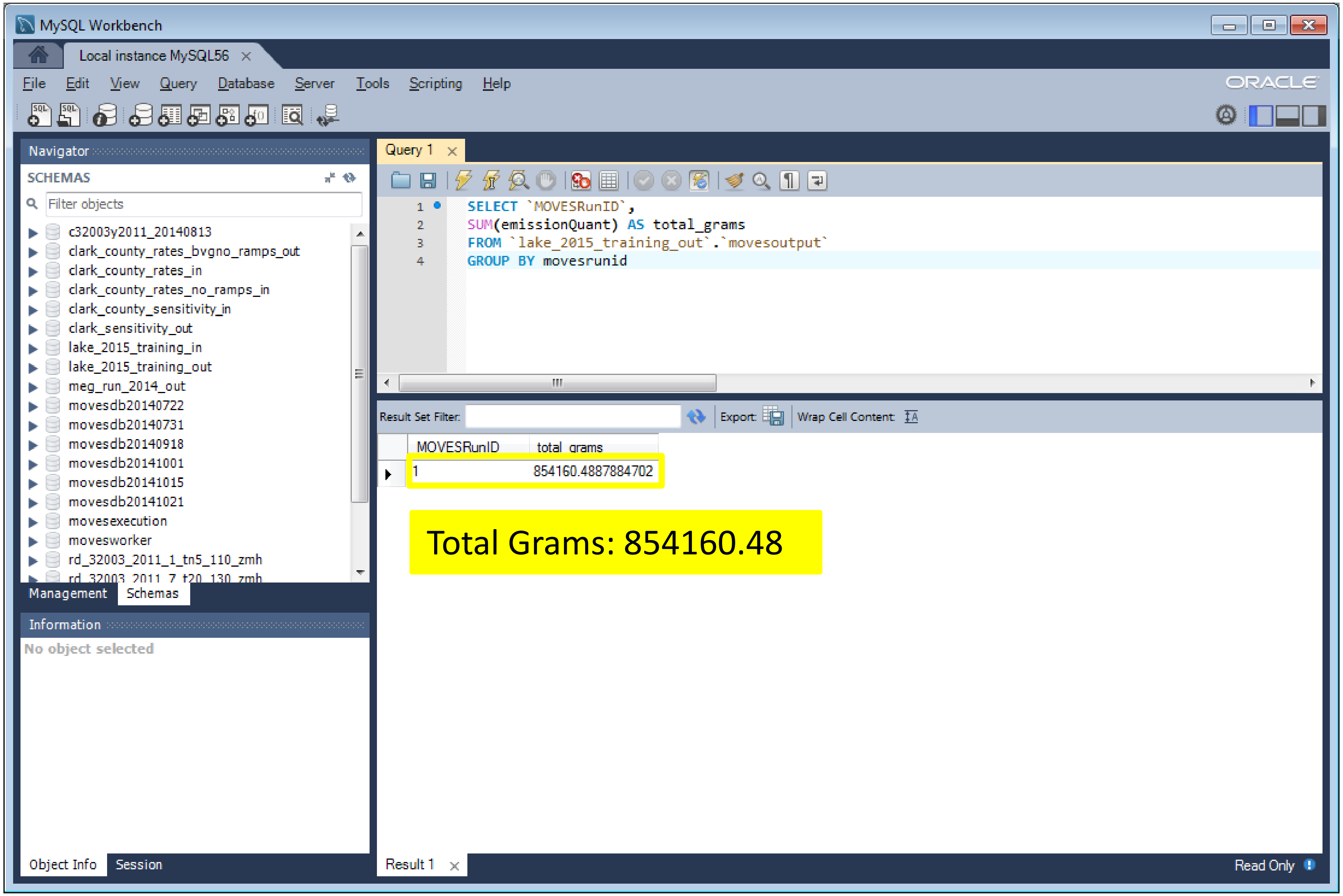
- Let’s run some example queries on our County-scale inventory results from Module 3
- Instructions for Accessing County-scale Results:
  - Open file *post-processing.txt*, copy “Script 1” and paste into Query area of MSQL Browser, then click the ⚡ button to calculate the total inventory

```
SELECT `MOVESRunID`,
SUM(emissionQuant) AS total_grams
FROM `lake_2015_training_out`.`movesoutput`
GROUP BY movesrunid
```

MOVES Motor Vehicle Emission Simulator

EPA

# Simple MySQL Queries: County-scale Results



c32003y2011_20140813
clark_county_rates_bvgno_ramps_out
clark_county_rates_in
clark_county_rates_no_ramps_in
clark_county_sensitivity_in
clark_sensitivity_out
lake_2015_training_in
lake_2015_training_out
meg_run_2014_out
movesdb20140722
movesdb20140731
movesdb20140918
movesdb20141001
movesdb20141015
movesdb20141021
movesexecution
movesworker
rd_32003_2011_1_tn5_110_zmh
rd_32003_2011_7_t20_130_zmh

```
SELECT `MOVESRunID`,
SUM(emissionQuant) AS total_grams
FROM `lake_2015_training_out`.`movesoutput`
GROUP BY movesrunid
```

| MOVESRunID | total_grams |
|---|---|
| 1 | 854160.4887884702 |

Total Grams: 854160.48

## Simple MySQL Queries

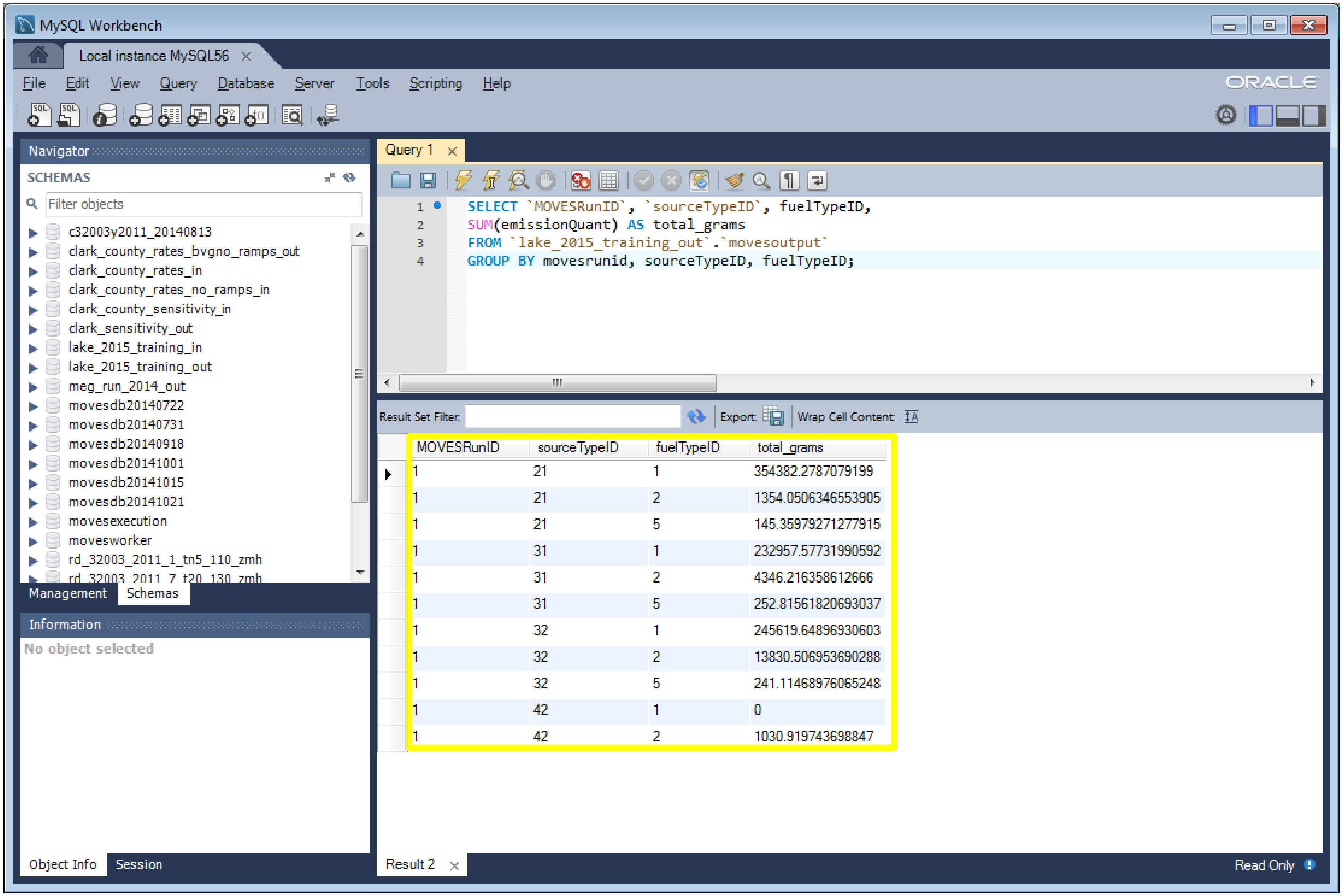
- Instructions for Accessing County-scale Results:
  - Open file *post-processing.txt*, copy "Script 2" and paste into Query area of MSQL Browser, then click the ⚡ button to calculate the inventory by source type and fuel type.

```
SELECT `MOVESRunID`, `sourceTypeID`, fuelTypeID,
SUM(emissionQuant) AS total_grams
FROM `lake_2015_training_out`.`movesoutput`
GROUP BY movesrunid, sourceTypeID, fuelTypeID;
```

MOVES Motor Vehicle Emission Simulator

EPA

# Simple MySQL Queries: County-scale Results



```sql
SELECT `MOVESRunID`, `sourceTypeID`, fuelTypeID,
SUM(emissionQuant) AS total_grams
FROM `lake_2015_training_out`.`movesoutput`
GROUP BY movesrunid, sourceTypeID, fuelTypeID;
```

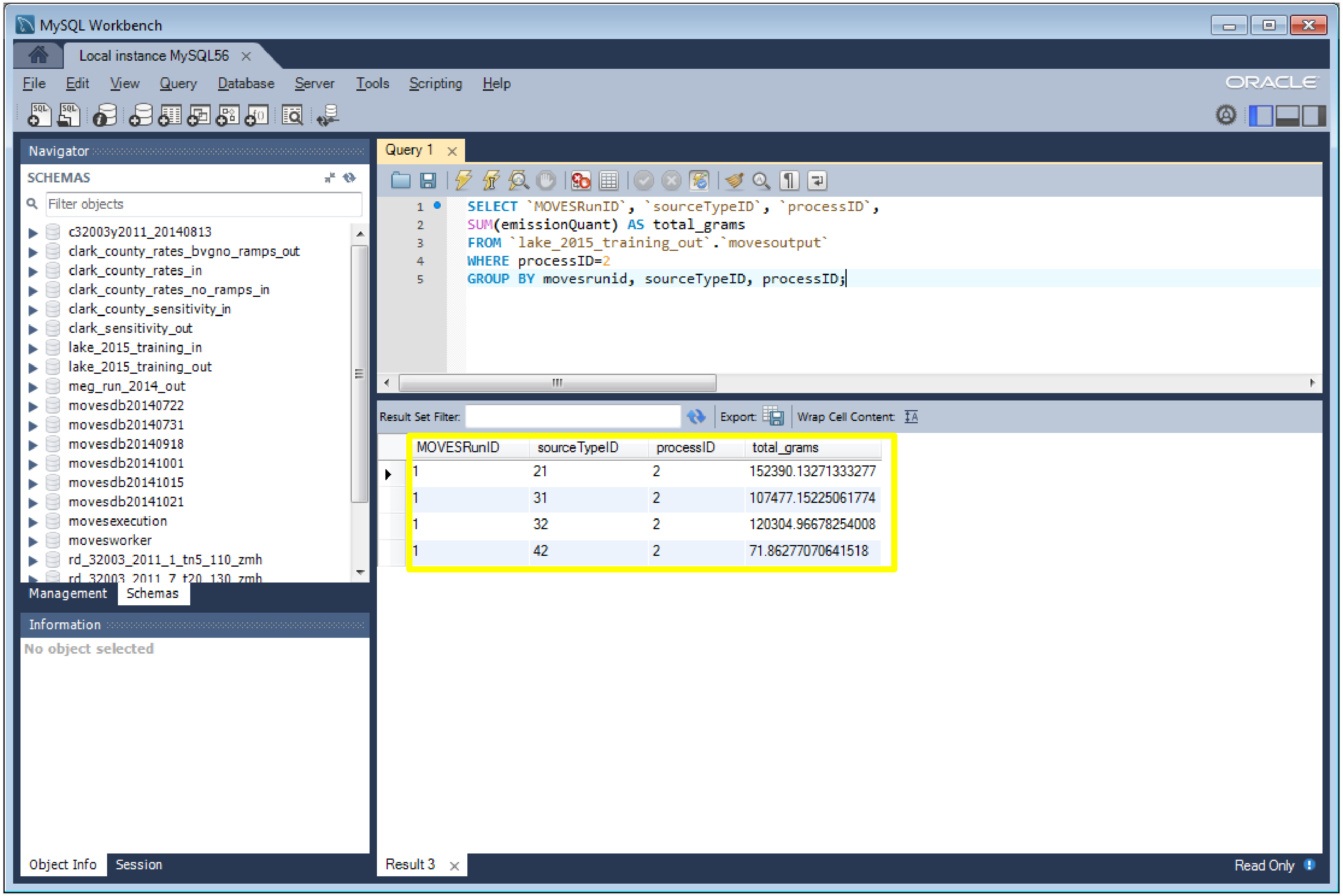
| MOVESRunID | sourceTypeID | fuelTypeID | total_grams |
|---|---|---|---|
| 1 | 21 | 1 | 354382.2787079199 |
| 1 | 21 | 2 | 1354.0506346553905 |
| 1 | 21 | 5 | 145.35979271277915 |
| 1 | 31 | 1 | 232957.57731990592 |
| 1 | 31 | 2 | 4346.216358612666 |
| 1 | 31 | 5 | 252.81561820693037 |
| 1 | 32 | 1 | 245619.64896930603 |
| 1 | 32 | 2 | 13830.506953690288 |
| 1 | 32 | 5 | 241.11468976065248 |
| 1 | 42 | 1 | 0 |
| 1 | 42 | 2 | 1030.919743698847 |

# Simple MySQL Queries

- Instructions for Accessing County-scale Results:
  - Open file *post-processing.txt*, copy “Script 3” and paste into Query area of MSQL Browser, then click the ⚡ button to calculate start emissions only, by source type.

```
SELECT `MOVESRunID`, `sourceTypeID`, `processID`,
SUM(emissionQuant) AS total_grams
FROM `lake_2015_training_out`.`movesoutput`
WHERE processID=2
GROUP BY movesrunid, sourceTypeID, processID;
```

# Simple MySQL Queries: County-scale Results



```
SELECT `MOVESRunID`, `sourceTypeID`, `processID`,
SUM(emissionQuant) AS total_grams
FROM `lake_2015_training_out`.`movesoutput`
WHERE processID=2
GROUP BY movesrunid, sourceTypeID, processID;
```

| MOVESRunID | sourceTypeID | processID | total_grams |
|---|---|---|---|
| 1 | 21 | 2 | 152390.13271333277 |
| 1 | 31 | 2 | 107477.15225061774 |
| 1 | 32 | 2 | 120304.96678254008 |
| 1 | 42 | 2 | 71.86277070641518 |

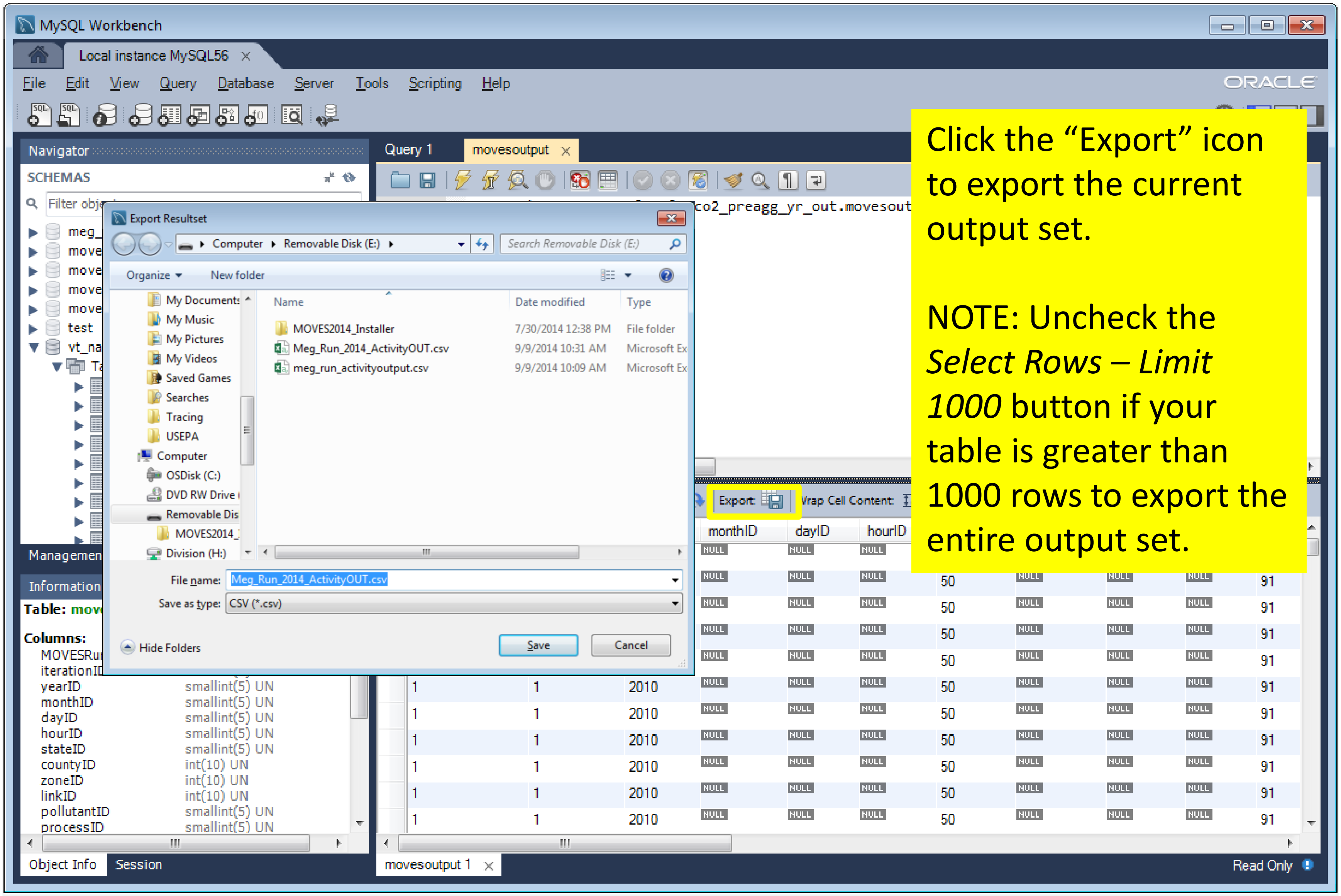
## Other MySQL Features

- Exporting result sets
- Viewing query history
- Saving and bookmarking queries

MOVES
Motor Vehicle Emission Simulator

EPA

# Other MySQL Features: Exporting



Click the “Export” icon to export the current output set.

NOTE: Uncheck the *Select Rows – Limit 1000* button if your table is greater than 1000 rows to export the entire output set.

| Name | Date modified | Type |
|---|---|---|
| MOVES2014_Installer | 7/30/2014 12:38 PM | File folder |
| Meg_Run_2014_ActivityOUT.csv | 9/9/2014 10:31 AM | Microsoft Ex |
| meg_run_activityoutput.csv | 9/9/2014 10:09 AM | Microsoft Ex |

File name: Meg_Run_2014_ActivityOUT.csv

Save as type: CSV (*.csv)

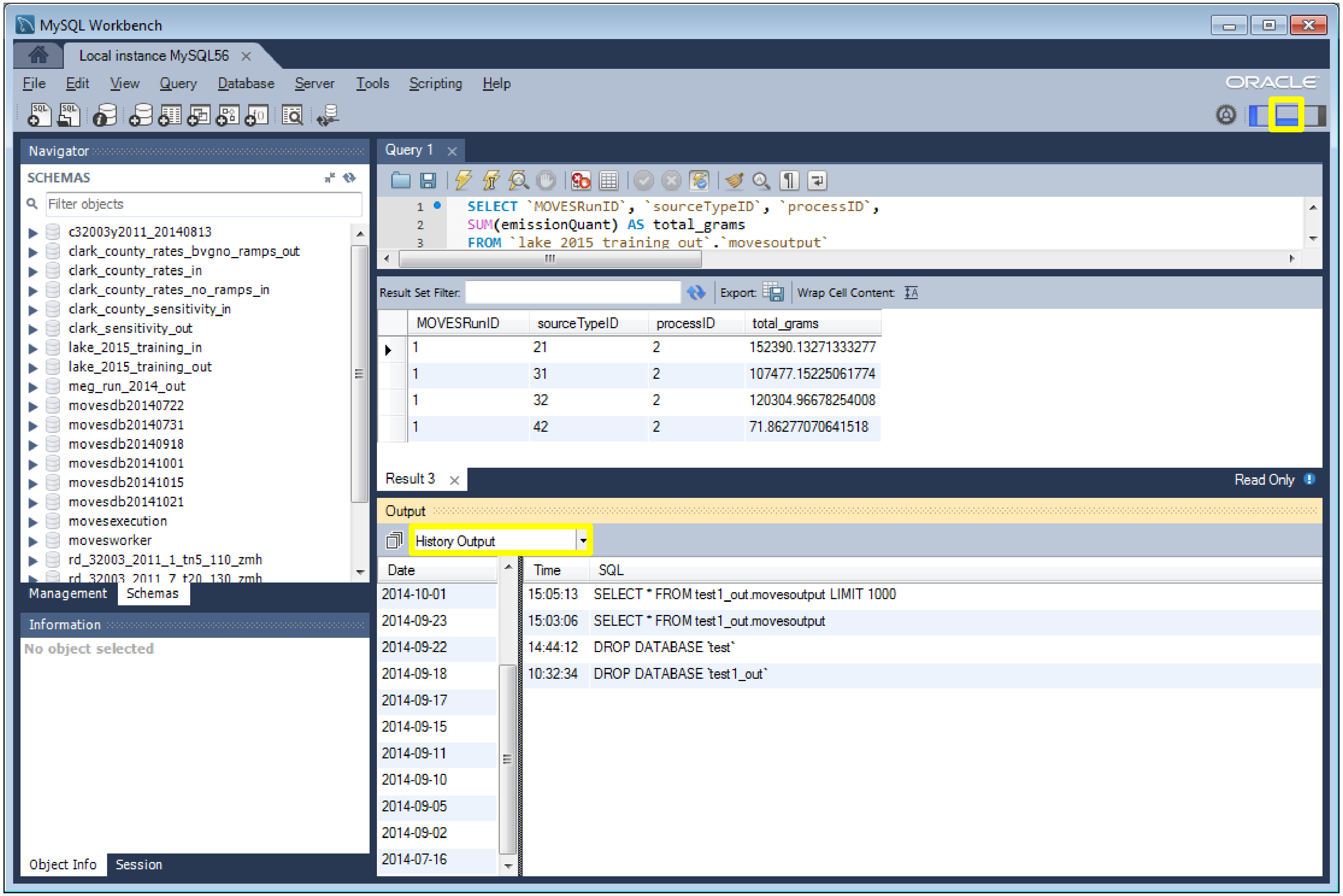
Table: mov

Columns:

| | |
|---|---|
| MOVESRu | |
| iterationI | |
| yearID | smallint(5) UN |
| monthID | smallint(5) UN |
| dayID | smallint(5) UN |
| hourID | smallint(5) UN |
| stateID | smallint(5) UN |
| countyID | int(10) UN |
| zoneID | int(10) UN |
| linkID | int(10) UN |
| pollutantID | smallint(5) UN |
| processID | smallint(5) UN |

| | | | monthID | dayID | hourID | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |
| 1 | 1 | 2010 | NULL | NULL | NULL | 50 | NULL | NULL | NULL | 91 |

# Other MySQL Features: History



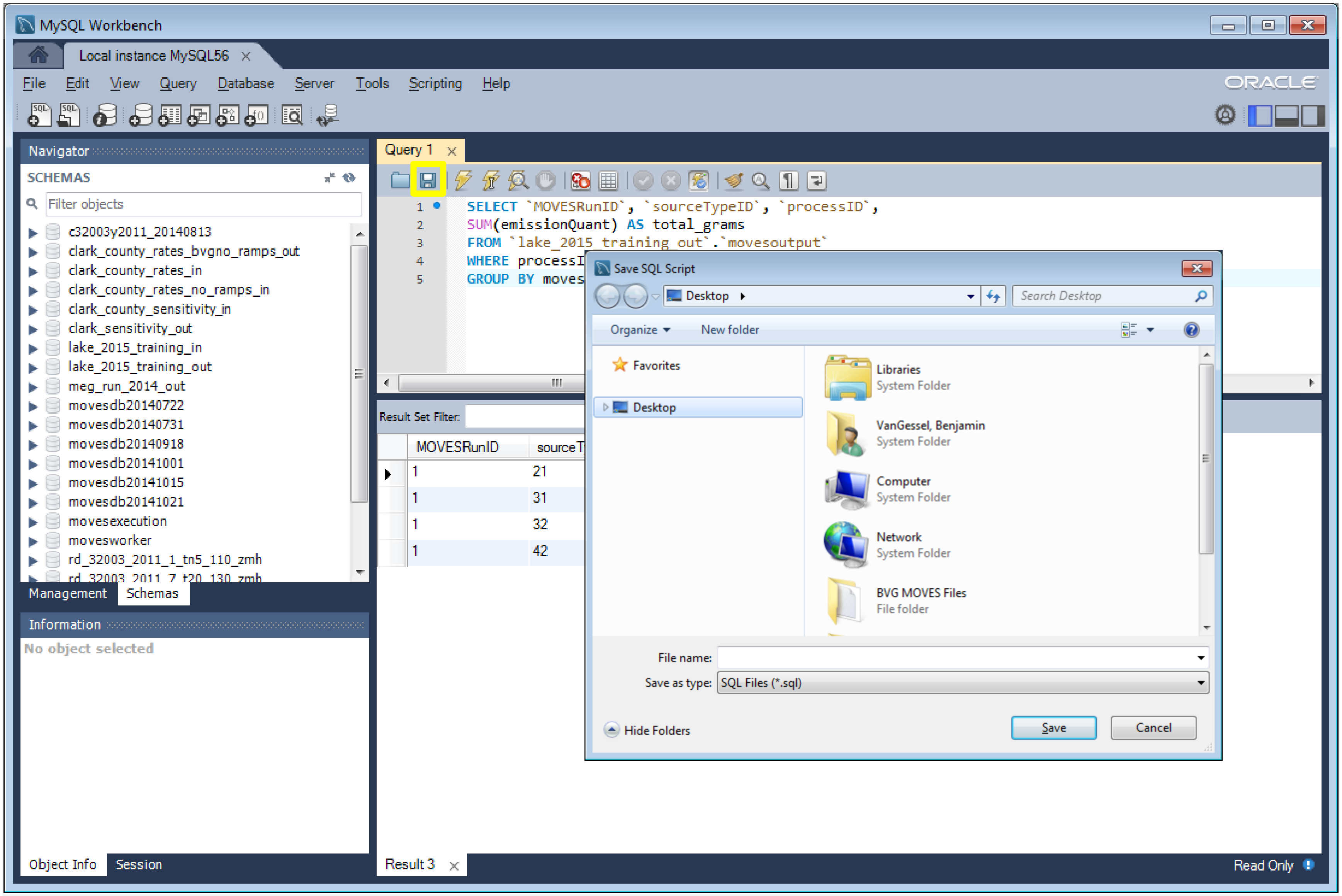
```
SELECT `MOVESRunID`, `sourceTypeID`, `processID`,
SUM(emissionQuant) AS total_grams
FROM `lake_2015_training_out`.`movesoutput`
```

| MOVESRunID | sourceTypeID | processID | total_grams |
|---|---|---|---|
| 1 | 21 | 2 | 152390.13271333277 |
| 1 | 31 | 2 | 107477.15225061774 |
| 1 | 32 | 2 | 120304.96678254008 |
| 1 | 42 | 2 | 71.86277070641518 |

| Date |
|---|
| 2014-10-01 |
| 2014-09-23 |
| 2014-09-22 |
| 2014-09-18 |
| 2014-09-17 |
| 2014-09-15 |
| 2014-09-11 |
| 2014-09-10 |
| 2014-09-05 |
| 2014-09-02 |
| 2014-07-16 |

| Time | SQL |
|---|---|
| 15:05:13 | SELECT * FROM test1_out.movesoutput LIMIT 1000 |
| 15:03:06 | SELECT * FROM test1_out.movesoutput |
| 14:44:12 | DROP DATABASE `test` |
| 10:32:34 | DROP DATABASE `test1_out` |

# Other MySQL Features: Saving Queries



```
SELECT `MOVESRunID`, `sourceTypeID`, `processID`,
SUM(emissionQuant) AS total_grams
FROM `lake_2015_training_out`.`movesoutput`
WHERE processI
GROUP BY moves
```

| MOVESRunID | sourceT |
|---|---|
| 1 | 21 |
| 1 | 31 |
| 1 | 32 |
| 1 | 42 |

## Copying and sending MySQL databases

- Input and output databases are stored within the folder.
- Data folder location varies by operating system.
- Databases may be copied and zipped for email and review

MOVES
Motor Vehicle Emission Simulator

EPA

## Questions?

MOVES

Motor Vehicle Emission Simulator

EPA