Module 3
Processing MOVES Output
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
Using MOVES: Where Are We?

Input

- Run Specification (RunSpec)
- Input Dataset

MOVES Functions

- MOVES Worker
- MOVES Master (MOVES GUI)
- MOVES Execution Database

Output

- Processed Output
- Output Database

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)
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
Now let’s talk about how to process the output data so it’s 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 interested 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
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
## Post-Processing Script

<table>
<thead>
<tr>
<th>Script Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DecodeMOVESOutput.sql</td>
<td>Decodes most key fields of MOVESOutput and MOVESActivityOutput tables</td>
</tr>
<tr>
<td>EmissionRates.sql</td>
<td>Produces an output table which reports the emission results in units of mass per distance</td>
</tr>
<tr>
<td>TabbedOutput.sql</td>
<td>Produces tab-delimited output suitable for reading into an EXCEL Spreadsheet from the MOVES MySQL database output tables</td>
</tr>
</tbody>
</table>
## Post-Processing Scripts: Project Scale

<table>
<thead>
<tr>
<th>Script Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO_CAL3QHC_EF.sql</td>
<td>Produces CO emission rates for use in the CAL3QHC air quality model</td>
</tr>
<tr>
<td>CO_Grams_Per_Hour.sql</td>
<td>Produces CO emission rates as grams per hour for each link (project-scale runs)</td>
</tr>
<tr>
<td>CO_Grams_Per_Veh_Mile.sql</td>
<td>Produces CO emission rates as grams per vehicle-mile for each link (project-scale runs)</td>
</tr>
<tr>
<td>PM10_Grams_Per_Hour.sql</td>
<td>Produces PM10 emission rates as grams per hour for each link (project-scale runs)</td>
</tr>
<tr>
<td>PM10_Grams_Per_Veh_Mile.sql</td>
<td>Produces PM10 emission rates as grams per vehicle-mile for each link (project-scale runs)</td>
</tr>
<tr>
<td>PM25_Grams_Per_Hour.sql</td>
<td>Produces PM2.5 emission rates as grams per hour for each link (project-scale runs)</td>
</tr>
<tr>
<td>PM25_Grams_Per_Veh_Mile.sql</td>
<td>Produces PM2.5 emission rates as grams per vehicle-mile for each link (project-scale runs)</td>
</tr>
</tbody>
</table>
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
Post-Processing Menu: Summary Report

[Image of a software interface for specifying parameters for a summary report]
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
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
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
1. Double-click **Local Instance MySQL**

2. Click Connect

3. Enter “moves” in the password field. Choose to **Save password in vault.**
MySQL Workbench – Layout

- Active Connection
- Toggle Preferences and Panels
- SQL Query Menu
- SQL Query Panel
- Object Browser
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

Expand your desired output database (SCHEMAS)
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

Or simply write and run a query
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

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

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;
```
Simple MySQL Queries: County-scale Results
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
Other MySQL Features

- Exporting result sets
- Viewing query history
- Saving and bookmarking queries
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.
Other MySQL Features: History
Other MySQL Features: Saving Queries

![MySQL Workbench interface showing a query window and a save SQL script dialog box. The query is selecting a list of columns from a table named `movescb20140722` and saving the script to the desktop.]
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
Questions?