

Introduction to SQL Syntax
A Tip Sheet for MOVES Users

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 sourceTypeID=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

Tips:

- Syntax must be used in the order given above. 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 must be used to separate multiple fields following a command
- AS command is useful to name a new fields created with the SUM command
- When using a SUM command, all fields that were selected that are not summed should be included in the GROUP BY command

Examples of simple 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. **The non-grouped and non-summed fields here are not useful, so this is an example of a poorly structured query.***

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

Same as above, but only selects the useful columns (movesRunID and summed emissionQuant) instead of all data.

```
SELECT movesRunID, SUM(emissionQuant) FROM lake_2015_training_out.movesoutput
WHERE pollutantID=1
GROUP BY movesRunID;
```

Same result as above, but only adds up emissions quantity for pollutantID 1 (total gaseous hydrocarbons)

```
SELECT movesRunID, pollutantID, processID, SUM(emissionQuant) AS grams
FROM lake_2015_training_out.movesoutput
WHERE pollutantID=1
GROUP BY movesRunID, pollutantID, processID
ORDER BY movesRunID;
```

Sums the emissions quantity in a new field named "grams" for pollutantID 1 and groups the results by movesRunID, pollutantID, and processID. This will produce rows for each process selected in each run and orders query results by movesRunID (all run 1 results will be listed in a row before run 2 results, etc.)