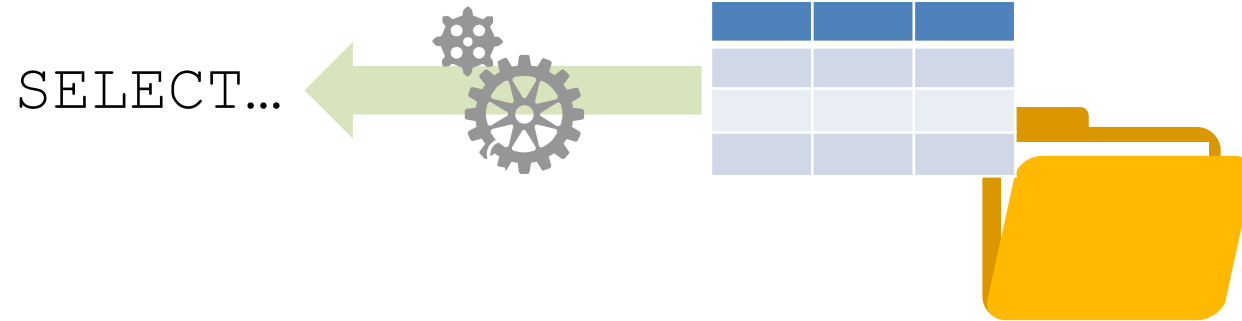


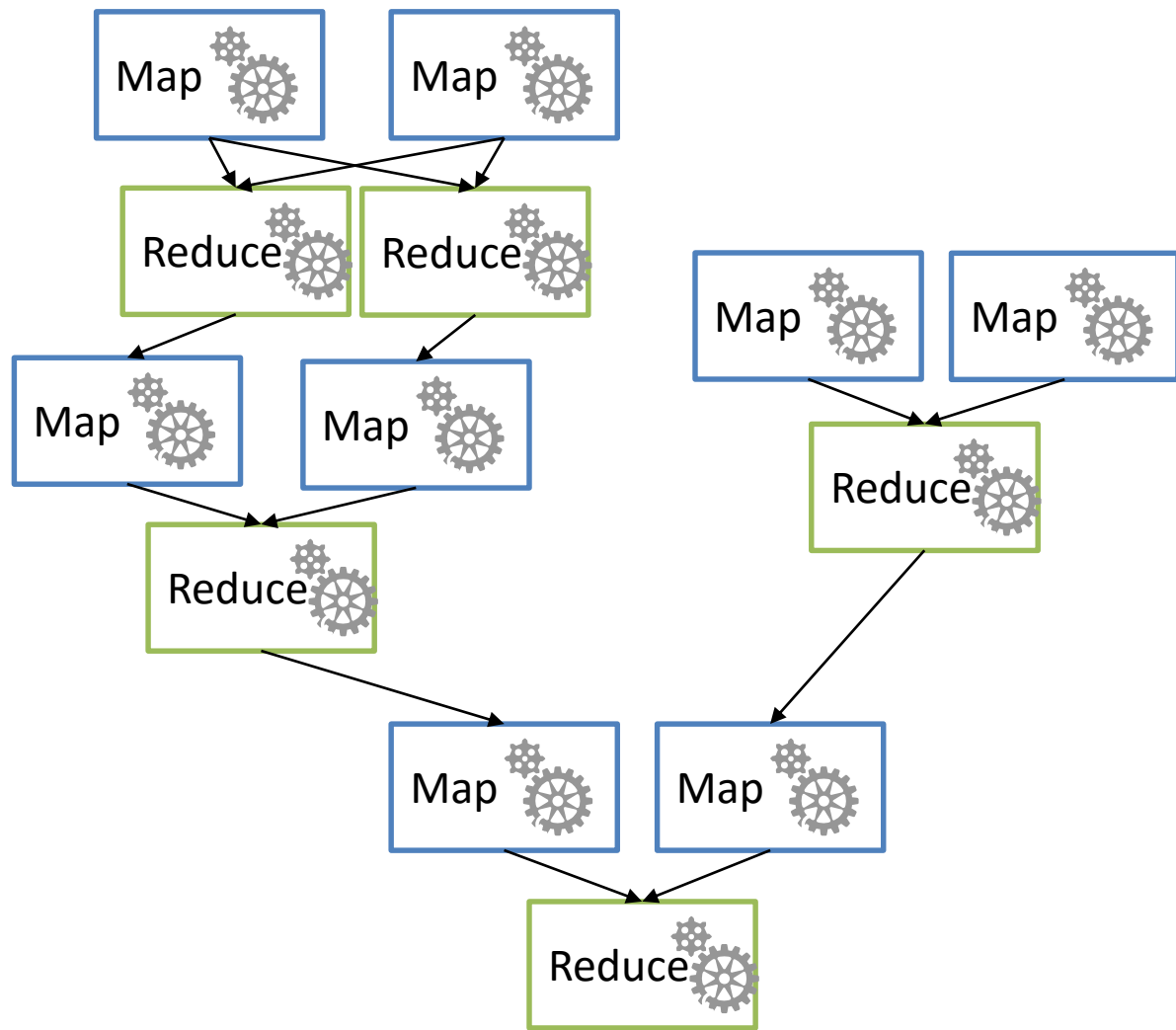
Processing Big Data with Hive

What is Hive?

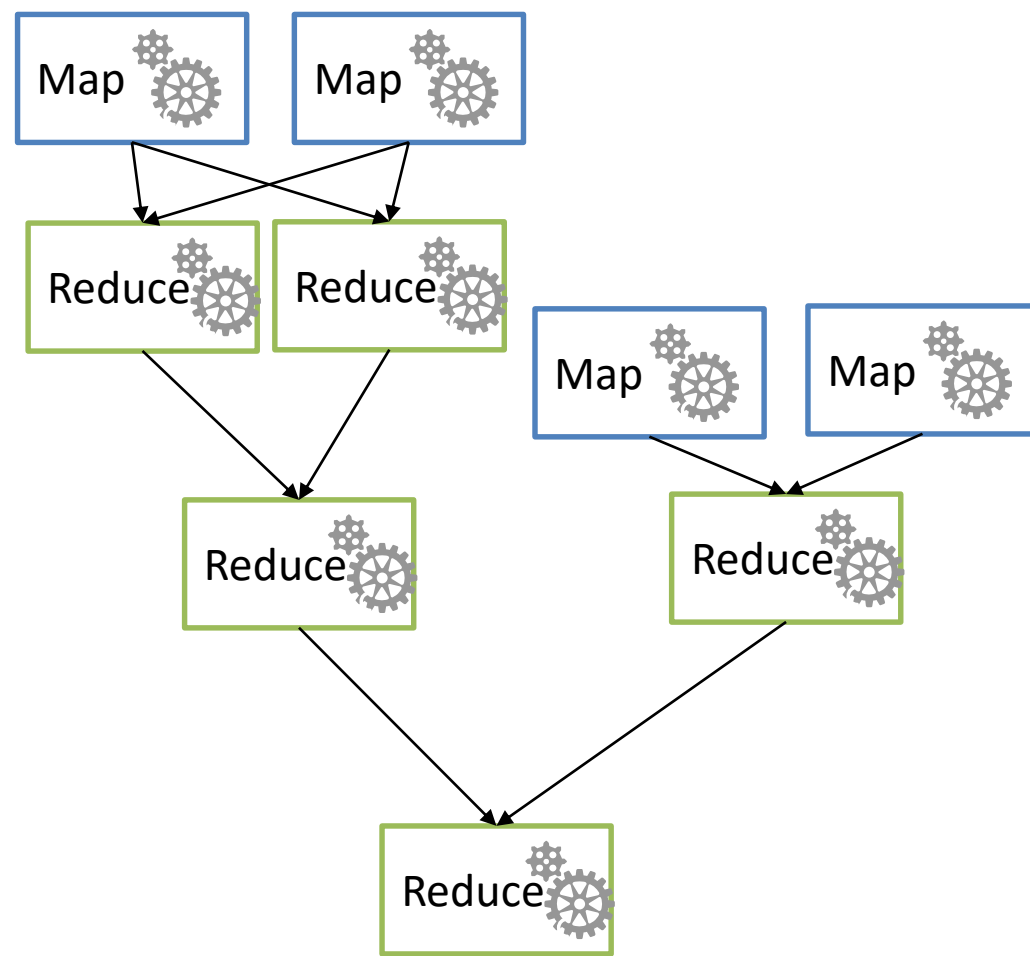


- A metadata service that projects tabular schemas over folders
- Enables the contents of folders to be queried as tables, using SQL-like query semantics
- Queries are translated into jobs
 - Execution engine can be Tez or MapReduce

```
set hive.execution.engine=mr;  
SELECT...
```

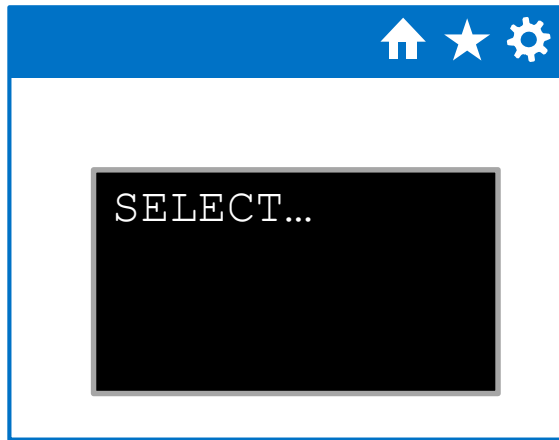


```
set hive.execution.engine=tez;  
SELECT...
```

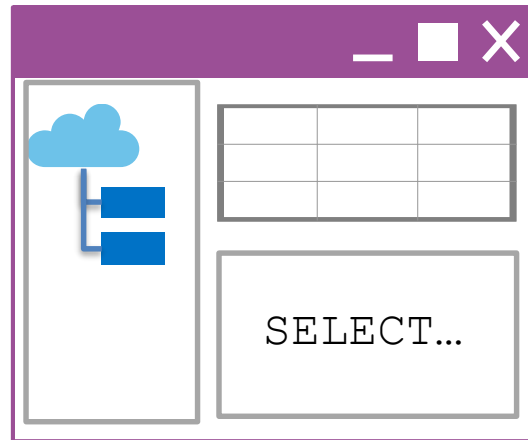


Hive client tools include...

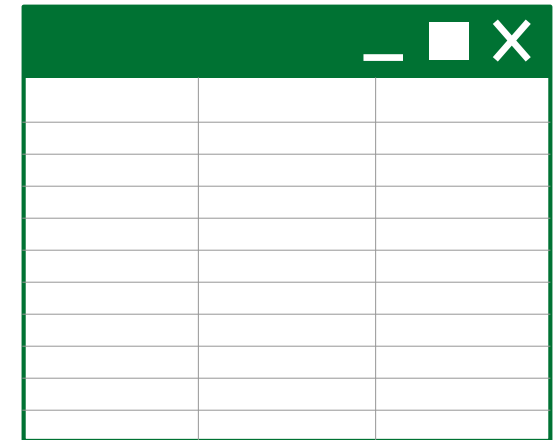
Hive Shell



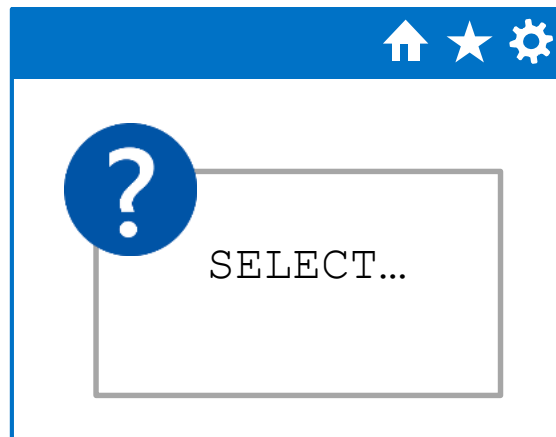
Visual Studio



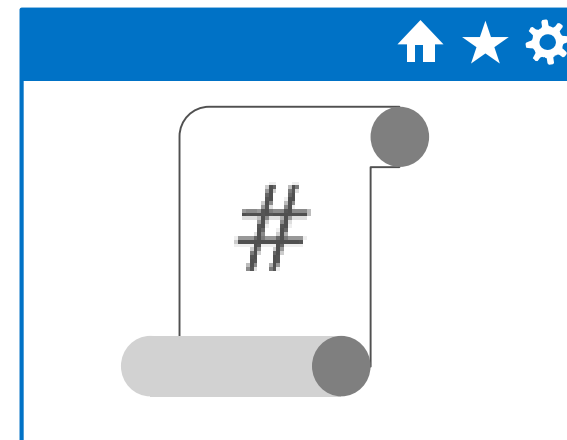
Any ODBC Client



Query Console (Hue)



PowerShell



How do I create and load Hive tables?

- Use the CREATE TABLE HiveQL statement
 - Defines schema metadata to be projected onto data in a folder when the table is queried (*not* when it is created)
- Specify file format and file location
 - Defaults to textfile format in the `<database>/<table_name>` folder
 - Default database is in `/hive/warehouse`
 - Create additional databases using CREATE DATABASE
- Create *internal* or *external* tables
 - Internal tables manage the lifetime of the underlying folders
 - External tables are managed independently from folders

```
CREATE TABLE table1  
(col1 STRING,  
 col2 INT)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ' ';
```

Internal table (folders
deleted when table is
dropped)

Default location
(/hive/warehouse/table1)

```
CREATE TABLE table2  
(col1 STRING,  
 col2 INT)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ' '  
STORED AS TEXTFILE LOCATION '/data/table2';
```

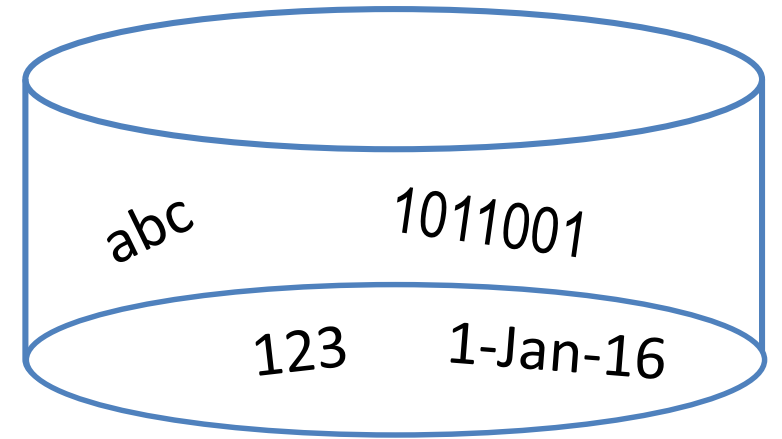
Stored in a custom folder (but
still internal, so the folder is
deleted when table is dropped)

```
CREATE EXTERNAL TABLE table3  
(col1 STRING,  
 col2 INT)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ' '  
STORED AS TEXTFILE LOCATION '/data/table3';
```

External table (folders and files
are left intact in Azure Blob Store
when the table is dropped)

Hive data types :

- Numeric
 - Integers: TINYINT, SMALLINT, INT, BIGINT
 - Fractional: FLOAT, DOUBLE, DECIMAL
- Character
 - STRING, VARCHAR, CHAR
- Date/Time
 - TIMESTAMP
 - DATE
- Special
 - BOOLEAN, BINARY, ARRAY, MAP, STRUCT, UNIONTYPE



- Save data files in table folders (or create table on existing files!)

```
PUT myfile.txt /data/table1
```

- Use the LOAD statement

```
LOAD DATA [LOCAL] INPATH '/data/source' INTO TABLE MyTable;
```

- Use the INSERT statement

```
INSERT INTO TABLE Table2  
SELECT Col1, UPPER(Col2),  
FROM Table1;
```

- Use a CREATE TABLE AS SELECT (CTAS) statement

```
CREATE TABLE Table3  
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'  
STORED AS TEXTFILE LOCATION '/data/summarytable'  
AS  
SELECT Col1, SUM(Col2) As Total  
FROM Table1  
GROUP BY Col1;
```

How do I query Hive tables?

- Query data using the SELECT HiveQL statement

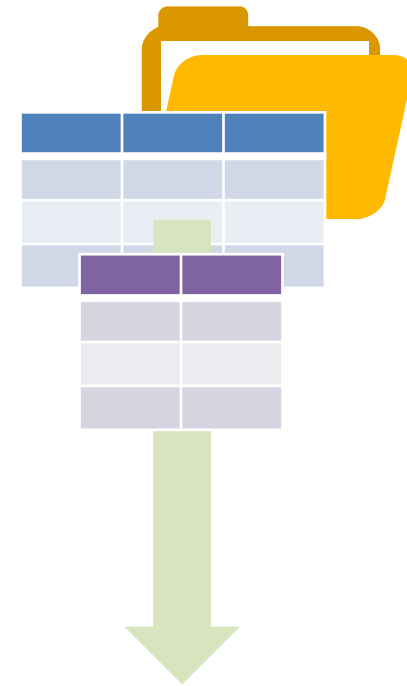
```
SELECT Col1, SUM(Col2) AS TotalCol2  
FROM MyTable  
WHERE Col3 = 'ABC' AND Col4 < 10  
GROUP BY Col1  
ORDER BY Col4;
```

- Hive translates the query into jobs and applies the table schema to the underlying data files

- Views are named queries that abstract underlying tables

```
CREATE VIEW v_SummarizedData  
AS  
SELECT col1, SUM(col2) AS TotalCol2  
FROM mytable  
GROUP BY col1;
```

```
SELECT col1, TotalCol2  
FROM v_SummarizedData;
```



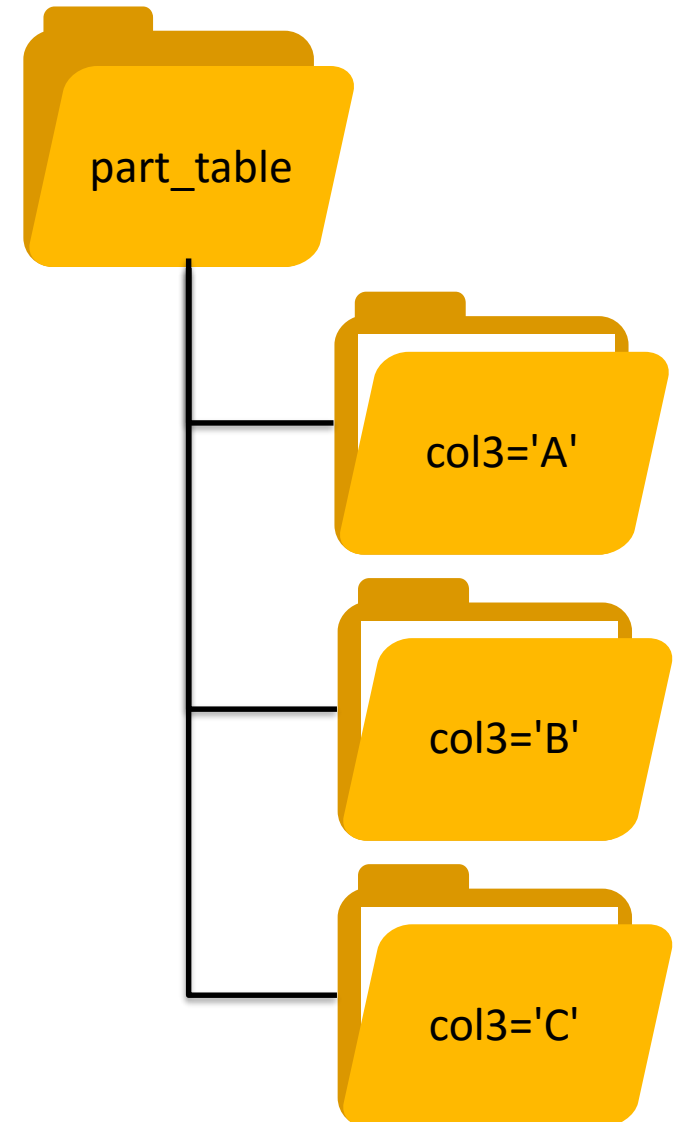
Partitioning, Skewing, and Clustering Tables

```
CREATE TABLE part_table
(col1 INT,
 col2 STRING)
PARTITIONED BY (col3 STRING);

INSERT INTO TABLE part_table PARTITION(col3='A')
SELECT col1, col2, col3
FROM stg_table
WHERE col3 = 'A';

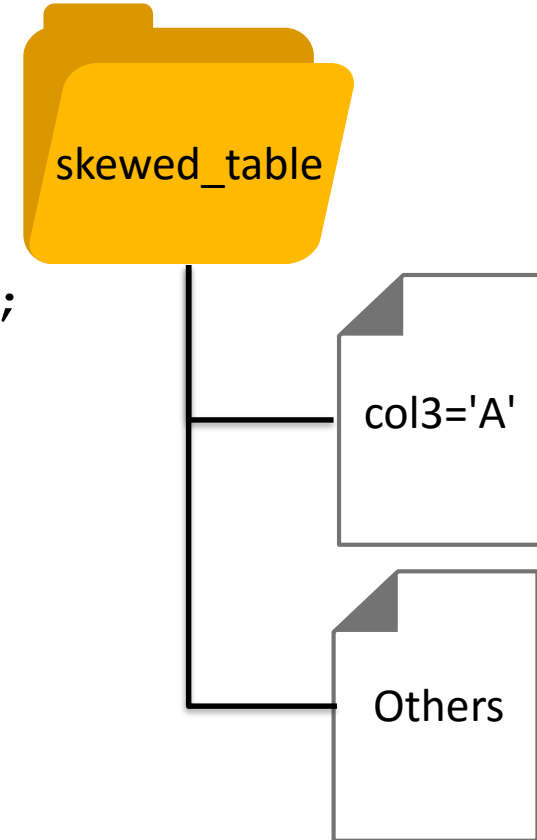
SET hive.exec.dynamic.partition = true;
SET hive.exec.dynamic.partition.mode=nonstrict;

INSERT INTO TABLE part_table PARTITION(col3)
SELECT col1, col2, col3
FROM stg_table;
```



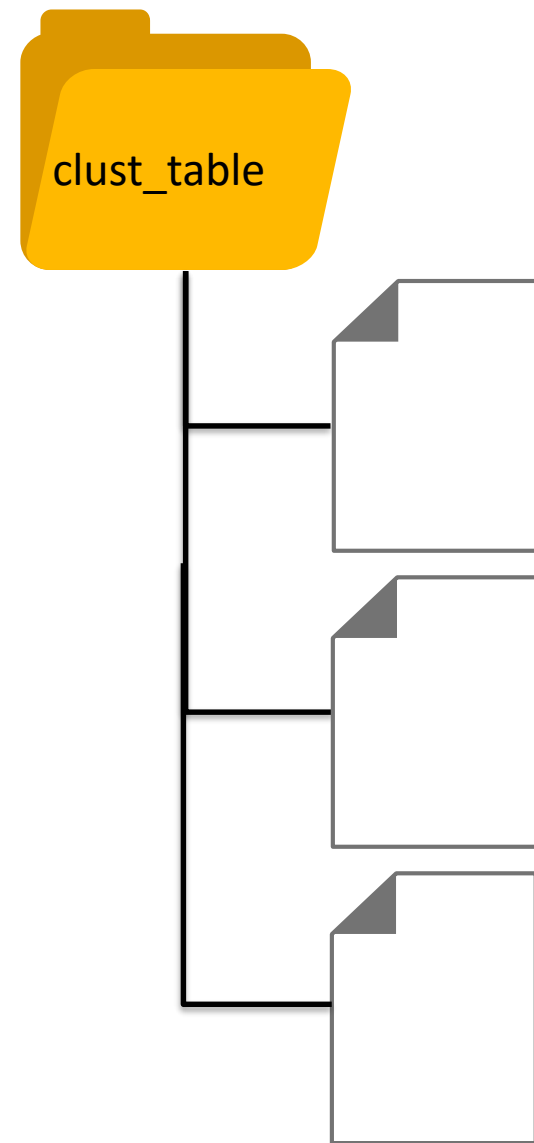
```
CREATE TABLE skewed_table  
(col1 INT,  
 col2 STRING,  
 col3 STRING)  
SKEWED BY (col3) ON ('A') [STORED AS DIRECTORIES];
```

```
INSERT INTO TABLE skewed_table  
SELECT col1, col2, col3  
FROM stg_table;
```




```
CREATE TABLE clust_table  
(col1 INT,  
 col2 STRING,  
 col3 STRING)  
CLUSTERED BY (col3) INTO 3 BUCKETS;
```

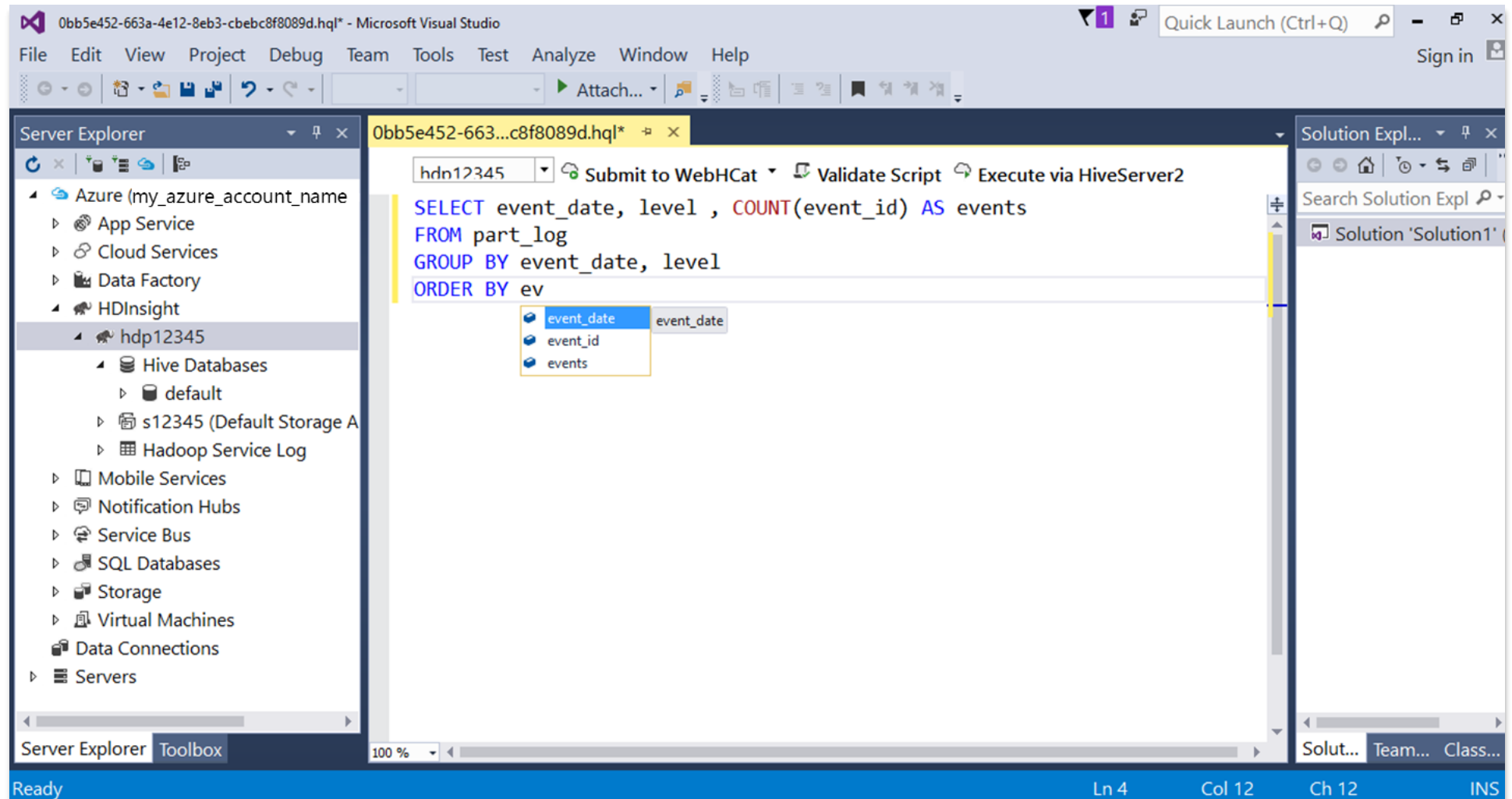
```
INSERT INTO TABLE clust_table  
SELECT col1, col2, col3  
FROM stg_table;
```



How do I use Hive in Visual Studio?

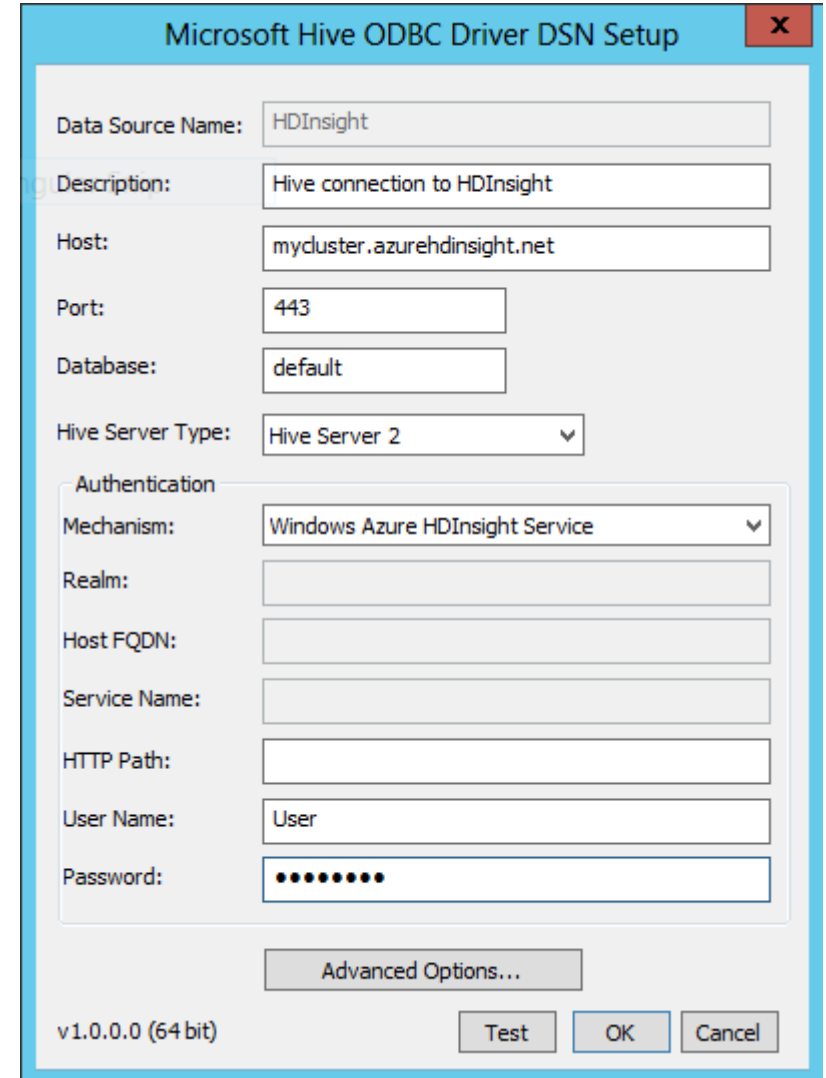
Azure SDK for .NET includes HDInsight tools for Visual Studio

- Visual Hive table designer
- Hive query editor



How do I access Hive via ODBC?

1. Download and install the Hive ODBC Driver for HDInsight
 - 32-bit and 64-bit versions
2. Optionally, create a data source name (DSN) for your HDInsight cluster
3. Use an ODBC connection to query Hive tables



The screenshot shows the "Microsoft Hive ODBC Driver DSN Setup" dialog box. It contains the following fields and options:

- Data Source Name:** HDInsight
- Description:** Hive connection to HDInsight
- Host:** mycluster.azurehdinsight.net
- Port:** 443
- Database:** default
- Hive Server Type:** Hive Server 2 (dropdown)
- Authentication:**
 - Mechanism:** Windows Azure HDInsight Service (dropdown)
 - Realm:** (empty text box)
 - Host FQDN:** (empty text box)
 - Service Name:** (empty text box)
 - HTTP Path:** (empty text box)
 - User Name:** User
 - Password:** (masked with dots)
- Advanced Options...** (button)
- Version:** v1.0.0.0 (64 bit)
- Buttons:** Test, OK, Cancel



Microsoft

©2014 Microsoft Corporation. All rights reserved. Microsoft, Windows, Office, Azure, System Center, Dynamics and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.