**HIVE**

Mostrar las tablas existentes en la base de datos actual

**SHOW TABLES;**

Mostrar las tablas existentes que comiencen por el prefijo page

**SHOW TABLES** 'page.\*'**;**

Mostrar las particiones de la tabla page\_view

**SHOW PARTITIONS** page\_view**;**

Mostrar información básica (columnas y sus tipos) de la tabla page\_view

**DESCRIBE** page\_view**;**

Mostrar la información básica de la tabla page\_view y otras propiedades.

Útil para depuración.

**DESCRIBE EXTENDED** page\_view**;**

Cambio de nombre de una tabla

**ALTER** **TABLE** old\_table\_name **RENAME** **TO** new\_table\_name**;**

Modificación de nombres y/o tipos de columnas

**ALTER** **TABLE** old\_table\_name **REPLACE** **COLUMNS** **(**col1 **TYPE,** **...);**

Añadir columnas a una tabla

**ALTER** **TABLE** tab1 **ADD COLUMNS** **(**c1 **INT** **COMMENT** 'a new int column'**,** c2 **STRING DEFAULT** 'def val'**);**

Borrar una tabla

**DROP** **TABLE** pv\_users**;**

Borrar una partición

**ALTER** **TABLE** pv\_users **DROP** **PARTITION** **(**ds**=**'2008-08-08'**);**

Copiar datos de un fichero local a una tabla

**LOAD DATA** **LOCAL** **INPATH** '/tmp/pv\_2008-06-08\_us.txt'

**INTO** **TABLE** page\_view **PARTITION(**date**=**'2008-06-08'**,** country**=**'US'**)**

Mover datos de un fichero en HDFS a una tabla

**LOAD DATA** **INPATH** '/user/data/pv\_2008-06-08\_us.txt'

**INTO** **TABLE** page\_view **PARTITION(**date**=**'2008-06-08'**,** country**=**'US'**)**

Sobrescribir la tabla/ficheros en destino

**LOAD DATA** **INPATH** '/user/data/pv\_2008-06-08\_us.txt'

**OVERWRITE INTO** **TABLE** page\_view **PARTITION(**date**=**'2008-06-08'**,** country**=**'US'**)**

Copiar datos de una tabla a un fichero local

**INSERT OVERWRITE LOCAL DIRECTORY** '/tmp/ca\_employees'

**SELECT** name**,** salary**,** address **FROM** employees **WHERE** se.state **=** 'CA'**;**

Cláusula LIMIT: selección de un cierto número de filas

**SELECT** \*

**FROM** page\_views

**LIMIT** 10**;**

Sentencias SELECT anidadas

**SELECT** (

**SELECT** round(salary - taxes) **AS** salary\_minus\_taxes

**FROM** employees

)

**FROM** e.salary\_minus\_taxes**;**

Sentencias CASE…WHEN…THEN

**SELECT** name**,** salary**,**

**CASE**

**WHEN** salary **<** 50000.0 **THEN** 'low'

**WHEN** salary **>=** 50000.0 **AND** salary **<** 70000.0 **THEN** 'middle'

**WHEN** salary **>=** 70000.0 **AND** salary **<** 100000.0 **THEN** 'high'

**ELSE** 'very high'

**END** **AS** bracket

**FROM** employees**;**

Uso de cálculos en los predicados

**SELECT** e**.\*** **FROM**

**(SELECT** name**,** salary**,** deductions**[**"Federal Taxes"**]** **AS** ded**,**

salary **\*** **(**1 **-** deductions**[**"Federal Taxes"**])** **AS** salary\_minus\_fed\_taxes

**FROM** employees**)** e

**WHERE** **round(**e**.**salary\_minus\_fed\_taxes**)** **>** 70000**;**

SORT BY: orden local, cada reducer ordena sus resultados

**SELECT** s**.**ymd**,** s**.**symbol**,** s**.**price\_close

**FROM** stocks s

**SORT BY** s**.**ymd **ASC,** s**.**symbol **DESC;**

La función cast() permite convertir explícitamente un valor de un tipo a otro tipo diferente

* + Por ejemplo, si el campo salary es de tipo STRING, lo podemos convertir a un número en coma flotante de la siguiente forma:

**SELECT** name**,** salary

**FROM** employees

**WHERE** **cast(**salary **AS** **FLOAT)** **<** 100000.0**;**

* + Para hacer casting de tipos BINARY hay que convertirlos primero a STRING

**SELECT** **(**2.0**\*cast(cast(**b **AS STRING)** **AS DOUBLE))** **from** src**;**

Seleccionar datos de un array:

**SELECT** name, phones[0], phones[1] **FROM customers\_phones;**

Modificar la propiedad correspondiente para mostrar por pantalla las cabeceras de las tablas

“set hive.cli.print.header=true;”

Importa el fichero en la tabla iris que acabamos de crear desde HDFS

Load data inpath ‘/user/Cloudera/hive/iris\_completo.txt’ into table iris;

Utilizando la función LEAD, ejecutar una query que devuelva la clase, p\_length y el LEAD de p\_length con Offset=1 y Default\_Value =0, particionado por clase y ordenado por p\_length.

select clase, p\_length, LEAD(p\_length,1,0) OVER (PARTITION BY clase ORDER BY p\_length) as Lead from iris;