

Apache Hive

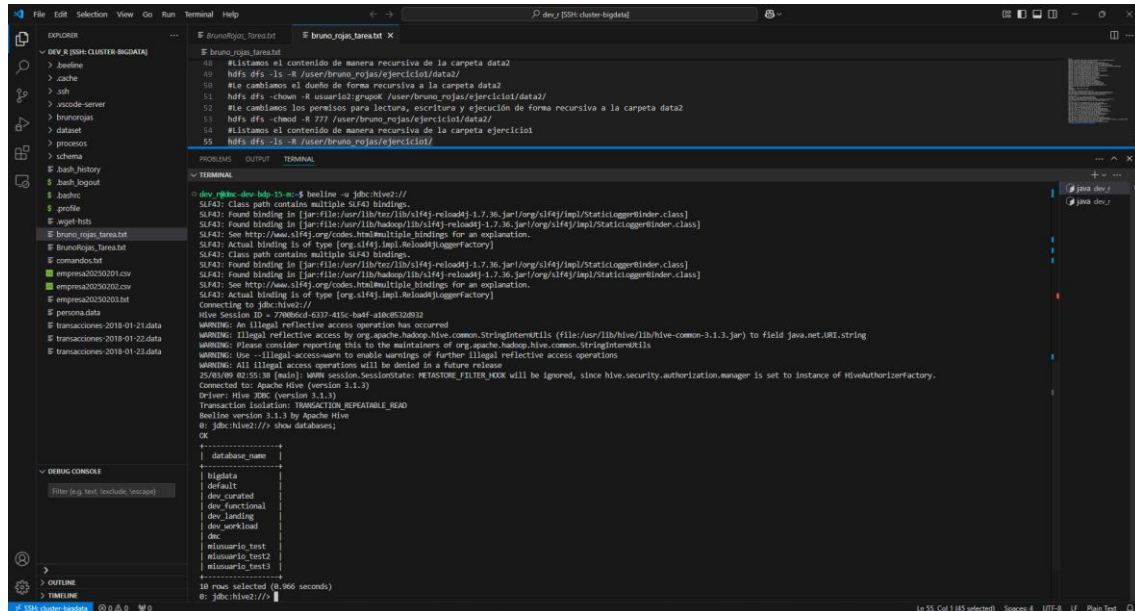
Conexión con Apache Hive

Nos conectamos a través del cliente beeline

```
beeline -u jdbc:hive2://
```

```
beeline -u jdbc:hive2://localhost:10000/default -n usr -p pwd
```

Para mostrar las bases de datos usamos el comando SHOW DATABASES;

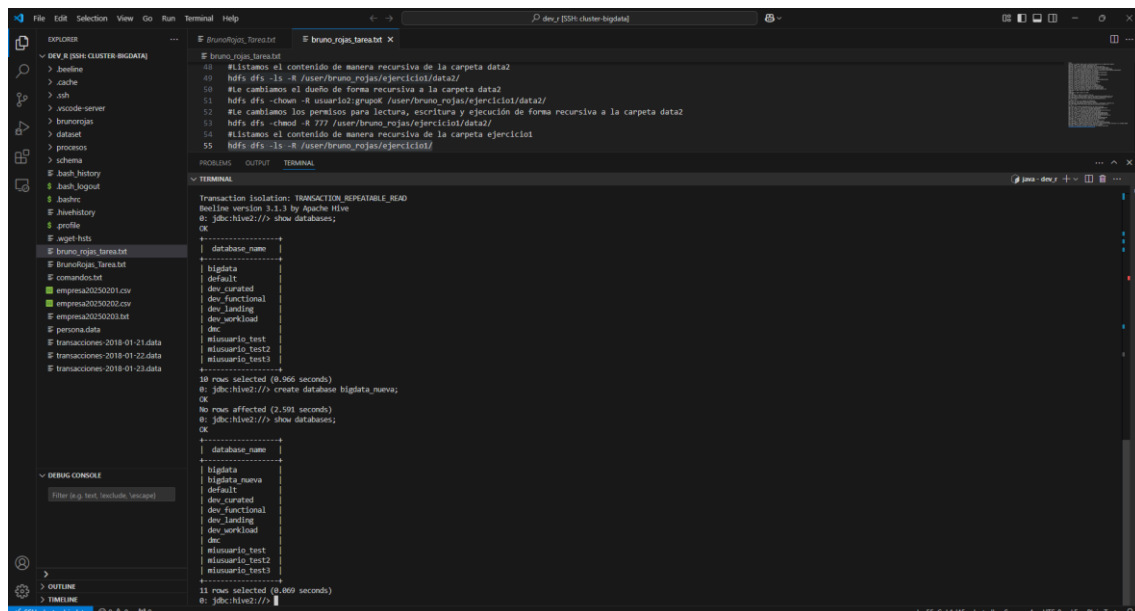


```
dev@dev:~$ beeline -u jdbc:hive2://
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/tez/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/tez/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
Connecting to jdbc:hive2://
Hive Session ID = 770806c4-6337-415c-ba4f-a8c85320932
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.hive.common.StringInternerUtil (file:/usr/lib/hive/lib/hive-common-3.13.3.jar) to field java.net.URI.string
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.hive.common.StringInternerUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
25/01/09 02:55:38 (main): WARN session.SessionState: METASTORE_FILTER_HOOK will be ignored, since hive.security.authorization.manager is set to instance of HiveAuthorizerFactory.
Connected to: Apache Hive (version 3.13.3)
Driver: Hive JDBC (version 3.13.3)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.13.3 by Apache Hive
0: jdbc:hive2://> show databases;
OK
=====
| database_name |
=====+=====
| bigdata       |
| default       |
| dev_curated   |
| dev_functional|
| dev_landing   |
| dev_workload  |
| dev           |
| minuario_test |
| minuario_test2|
| minuario_test3|
=====+=====
10 rows selected (0.966 seconds)
0: jdbc:hive2://>
```

Crear y mostrar bases de datos

```
create database bigdata_nueva;
```

```
show databases;
```



```
dev@dev:~$ beeline -u jdbc:hive2://
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/tez/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/tez/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
Connecting to jdbc:hive2://
Hive Session ID = 770806c4-6337-415c-ba4f-a8c85320932
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.hive.common.StringInternerUtil (file:/usr/lib/hive/lib/hive-common-3.13.3.jar) to field java.net.URI.string
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.hive.common.StringInternerUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
25/01/09 02:55:38 (main): WARN session.SessionState: METASTORE_FILTER_HOOK will be ignored, since hive.security.authorization.manager is set to instance of HiveAuthorizerFactory.
Connected to: Apache Hive (version 3.13.3)
Driver: Hive JDBC (version 3.13.3)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.13.3 by Apache Hive
0: jdbc:hive2://> show databases;
OK
=====
| database_name |
=====+=====
| bigdata       |
| bigdata_nueva |
| default       |
| dev_curated   |
| dev_functional|
| dev_landing   |
| dev_workload  |
| dev           |
| minuario_test |
| minuario_test2|
| minuario_test3|
=====+=====
11 rows selected (0.869 seconds)
0: jdbc:hive2://>
```

Creación de un schema o base de datos: create schema miusuario_test_nuevo;

```
11 rows selected (0.000 seconds)
0: jdbc:hive2://> create schema miusuario_test_nuevo;
OK
No rows affected (0.175 seconds)
0: jdbc:hive2://> show databases;
OK
+-----+
| database_name |
+-----+
| bigdata        |
| bigdata_nueva  |
| default        |
| dev_curated    |
| dev_functional |
| dev_landing    |
| dev_workload   |
| dm             |
| miusuario_test |
| miusuario_test2 |
| miusuario_test3 |
| miusuario_test_nuevo |
+-----+
12 rows selected (0.000 seconds)
0: jdbc:hive2://>
```

Crear tabla (miusuario_test_nuevo) y mostrarla

create table miusuario_test_nuevo.persona(id string, nombre string, telefono string, correo string, fecha_ingreso string, edad int, salario double, id_empresa string) row format delimited fields terminated by '|' lines terminated by '\n' stored as textfile;
show tables in miusuario_test_nuevo;

```
PROBLEMS OUTPUT TERMINAL
▼ TERMINAL
0: jdbc:hive2://> CREATE TABLE miusuario_test_nuevo.persona(
+-----+
| ID STRING,
+-----+
| NOMBRE STRING,
+-----+
| TELEFONO STRING,
+-----+
| CORREO STRING,
+-----+
| FECHA_INGRESO STRING,
+-----+
| EDAD INT,
+-----+
| SALARIO DOUBLE,
+-----+
| ID_EMPRESA STRING)
+-----+
| ROW FORMAT DELIMITED
+-----+
| FIELDS TERMINATED BY '|'
+-----+
| LINES TERMINATED BY '\n'
+-----+
| STORED AS TEXTFILE;
+-----+
OK
No rows affected (0.053 seconds)
0: jdbc:hive2://> show tables in miusuario_test_nuevo;
OK
+-----+
| tab_name |
+-----+
| persona  |
+-----+
1 row selected (0.003 seconds)
0: jdbc:hive2://>
```

La metadata de la base de datos que hemos creado se almacena en:
hdfs dfs -ls /user/hive/warehouse/miusuario_test_nuevo.db/persona

```
PROBLEMS OUTPUT TERMINAL
▼ TERMINAL
dev_r@hdc-dev-bdp-15-m:~$ hdfs dfs -ls /user/hive/warehouse
Found 6 items
drwxr-xr-x - dev_r hadoop 0 2025-02-18 02:25 /user/hive/warehouse/bigdata.db
drwxr-xr-x - dev_r hadoop 0 2025-03-09 03:06 /user/hive/warehouse/bigdata_nueva.db
drwxr-xr-x - dev_r hadoop 0 2025-02-20 02:47 /user/hive/warehouse/dmc.db
drwxr-xr-x - dev_r hadoop 0 2025-02-18 02:32 /user/hive/warehouse/miusuario_test.db
drwxr-xr-x - dev_r hadoop 0 2025-02-20 02:54 /user/hive/warehouse/miusuario_test3.db
drwxr-xr-x - dev_r hadoop 0 2025-03-09 03:17 /user/hive/warehouse/miusuario_test_nuevo.db
dev_r@hdc-dev-bdp-15-m:~$ hdfs dfs -ls /user/hive/warehouse/miusuario_test_nuevo.db
Found 1 items
drwxr-xr-x - dev_r hadoop 0 2025-03-09 03:17 /user/hive/warehouse/miusuario_test_nuevo.db/persona
dev_r@hdc-dev-bdp-15-m:~$ hdfs dfs -ls /user/hive/warehouse/miusuario_test_nuevo.db/persona
dev_r@hdc-dev-bdp-15-m:~$
```

Subimos el archivo a la carpeta de la tabla persona creada

hdfs dfs -put /home/dev_r/persona.data /user/hive/warehouse/miusuario_test_nuevo.db/persona

Listamos

hdfs dfs -ls /user/hive/warehouse/miusuario_test_nuevo.db/persona

```
▼ TERMINAL
dev_r@hdc-dev-bdp-15-m:~$ hdfs dfs -ls /user/hive/warehouse
Found 6 items
drwxr-xr-x - dev_r hadoop 0 2025-02-18 02:25 /user/hive/warehouse/bigdata.db
drwxr-xr-x - dev_r hadoop 0 2025-03-09 03:06 /user/hive/warehouse/bigdata_nueva.db
drwxr-xr-x - dev_r hadoop 0 2025-02-20 02:47 /user/hive/warehouse/dmc.db
drwxr-xr-x - dev_r hadoop 0 2025-02-18 02:32 /user/hive/warehouse/miusuario_test.db
drwxr-xr-x - dev_r hadoop 0 2025-02-20 02:54 /user/hive/warehouse/miusuario_test3.db
drwxr-xr-x - dev_r hadoop 0 2025-03-09 03:17 /user/hive/warehouse/miusuario_test_nuevo.db
dev_r@hdc-dev-bdp-15-m:~$ hdfs dfs -ls /user/hive/warehouse/miusuario_test_nuevo.db
Found 1 items
drwxr-xr-x - dev_r hadoop 0 2025-03-09 03:17 /user/hive/warehouse/miusuario_test_nuevo.db/persona
dev_r@hdc-dev-bdp-15-m:~$ hdfs dfs -ls /user/hive/warehouse/miusuario_test_nuevo.db/persona
dev_r@hdc-dev-bdp-15-m:~$ hdfs dfs -put /home/dev_r/persona.data /user/hive/warehouse/miusuario_test_nuevo.db/persona
dev_r@hdc-dev-bdp-15-m:~$ hdfs dfs -ls /user/hive/warehouse/miusuario_test_nuevo.db/persona
Found 1 items
-rw-r--r-- 2 dev_r hadoop 7282 2025-03-10 14:03 /user/hive/warehouse/miusuario_test_nuevo.db/persona/persona.data
dev_r@hdc-dev-bdp-15-m:~$
```

Haciendo un select en hive

	persona_id	persona_nombre	persona_telefono	persona_correo	persona_fecha_ingreso	persona_edad	persona_salario	persona_id_empresa
1	ID	NOMBRE	TELEFONO	COMBO	FECHA_INGRESO	EDAD	SALARIO	ID_EMPRESA
2	1	Carl	1-765-613-9145	carl.Sed@bigcorp.co.uk	2000-04-23	34	20007.0	5
3	1	Fredrick	1-552-3405	fredrick.alijian@bigcorpnet.edu	2000-04-23	34	20008.0	2
4	1	Jocelyn	1-204-696-8004	amr.d.fish@bigcorp.co.uk	2000-04-01	27	10003.0	3
5	1	Alden	1-508-666-8004	alden@bigcorp.com	2000-11-06	22	10004.0	1
6	1	Leandra	1-608-8044	leandra@bigcorpnet.com	2000-10-10	41	22200.0	1
7	6	Bert	797-4453	a.fells@bigcorpnet.org	2000-04-25	70	7000.0	7
8	1	Paul	1-608-380-4592	quay@bigcorp.net	2000-04-21	52	1112.0	2
9	6	Jonah	214-2975	quay@bigcorp.net	2000-10-07	23	12000.0	5
10	6	Helen	615-2277	quay@bigcorp.net	2000-04-25	69	6000.0	6
11	1	Caban	1-508-562-2781	quay@bigcorp.net	2000-05-19	19	7000.0	7
12	1	Melyssa	506-7730	quay@bigcorp.net	2000-10-14	48	4013.0	8
13	1	Tensor	1-770-776-7007	quay@bigcorp.net	2000-04-19	1	10040.0	4
14	1	Trevor	512-1955	quay@bigcorp.net	2000-06-08	34	9001.0	5
15	1	Alden	733-2795	quay@bigcorp.net	2000-04-19	39	20000.0	7
16	1	Wanda	359-6911	quay@bigcorp.net	2000-04-27	27	15350.0	1
17	1	Alison	341-8522	quay@bigcorp.net	2000-12-05	66	1107.0	12
18	1	Don	724-3443	quay@bigcorp.net	2000-06-24	48	10013.0	4
19	1	Quin	1-187-316-7414	quay@bigcorp.net	2000-04-04	34	4709.0	7
20	1	Laure	1-914-623-2057	quay@bigcorp.net	2000-03-09	60	12003.0	4
21	1	Quary	1-432-806-8004	quay@bigcorp.net	2000-03-07	67	45752.0	1
22	1	Carissa	1-308-777-8059	quay@bigcorp.net	2000-03-16	31	10512.0	10
23	1	Samson	1-438-188-6663	quay@bigcorp.net	2000-04-05	22	7400.0	2
24	1	Amey	1-448-826-8007	quay@bigcorp.net	2000-03-17	24	1001.0	6
25	1	Paul	1-608-260-1173	quay@bigcorp.net	2000-12-21	52	14750.0	1
26	1	Brayden	1-457-726-9413	quay@bigcorp.net	2000-03-17	33	20549.0	7
27	1	Alexander	1-508-666-8004	quay@bigcorp.net	2000-04-27	50	11013.0	1
28	1	Stephan	232-2020	quay@bigcorp.net	2000-04-04	53	9600.0	9
29	1	Jana	1-564-186-5562	quay@bigcorp.net	2000-10-10	30	6403.0	12
30	1	Clayton	1-508-666-8004	quay@bigcorp.net	2000-04-04	50	9600.0	9
31	1	Rylee	306-5447	quay@bigcorp.net	2000-07-07	47	21593.0	13
32	1	Gisela	408-3033	quay@bigcorp.net	2000-08-21	67	6007.0	3

Ver la descripción formateada de la tabla
desc formatted miusuario_test_nuevo.persona;

```

TERMINAL
40 rows selected (0.287 seconds)
@ jdbc:hive2:// desc formatted elisuario_test_nuevo_persona;
OK
+-----+-----+-----+
| col_name | data_type | comment |
+-----+-----+-----+
# col_name | data_type | comment
id | string |
nombre | string |
telefono | string |
correo | string |
fecha_ingreso | string |
edad | int |
salario | double |
id_empresa | string |
id_empresa | NULL | NULL
# Detailed Table Information
Database | elisuario_test_nuevo | NULL
Owner | USER | NULL
CreatedTime | Sun Mar 09 09:17:03 UTC 2025 | NULL
LastAccessTime | UNKNOWN | NULL
Retention | 0 | NULL
Location | hdfs://dnc-dev-bdp-15-n/warehouse/elisuario_test_nuevo.db/persona | NULL
Table Type | MANAGED TABLE | NULL
Table Parameters |
+-----+-----+-----+
| COLUMN_STATS_ACCURATE | "BASIC_STATS" | "true" |
| "COLUMN_STATS" | "correct" | "true" |
| "valid" | "true" |
| "Fecha_ingreso" | "true" |
| "id" | "true" |
| "id_empresa" | "true" |
| "nombre" | "true" |
| "Vendedor" | "true" |
| "telefono" | "true" |
+-----+-----+-----+
| bucketing_version | 2 |
| numfiles | 0 |
| nummapes | 0 |
| rowDataSize | 0 |
| totalSize | 0 |
| transient_lastDdlTime | 1741800223 |
| NULL | NULL
# Storage Information
Serde Library: | org.apache.hadoop.hive.serde2.lazylazySimpleSerde | NULL
InputFormat: | org.apache.hadoop.mapred.TextInputFormat | NULL
+-----+-----+-----+

```

Ver descripción simple de la tabla
desc miusuario test nuevo.tabla;

```
0: jdbc:hive2:///> desc miusuario_test_nuevo.persona;
OK
+-----+-----+-----+
| col_name | data_type | comment |
+-----+-----+-----+
| id        | string    |         |
| nombre    | string    |         |
| telefono  | string    |         |
| correo    | string    |         |
| fecha_ingreso | string    |         |
| edad      | int       |         |
| salario   | double    |         |
| id_empresa | string    |         |
+-----+-----+-----+
8 rows selected (0.156 seconds)
0: jdbc:hive2:///>
```

Haciendo un select * from miusuario_test_nuevo.persona where edad = 30;

```
0: jdbc:hive2:// select * from miusuario17nuevo.persona where edad = 30;
Query ID = dev_r_20250310141332_9225b9c0-91a7-4c5a-acf7-ec27/cb8271c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1741614893762_0001)

OK

+-----+-----+-----+-----+-----+-----+-----+-----+
| persona.id | persona.nombre | persona.telefono | persona.correo | persona.fecha_ingreso | persona.edad | persona.salario | persona.id_empresa |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 46 | Camden | 1-663-548-3304 | lec@digmissimpharetraNam.net | 2003-10-19 | 30 | 7860.0 | 10 |
| 69 | Hallee | 1-288-996-8549 | interdum.feugiat@imperdietornare.edu | 2008-12-25 | 30 | 16782.0 | 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+

2 rows selected (31.842 seconds)
0: jdbc:hive2://
```

Hive crea las bases de datos y tablas por defecto en la ruta /user/hive/warehouse/. Si quisiéramos especificar una ruta distinta se usa la sentencia LOCATION. Creamos base de datos con location

```
create database miusuario_test2_nuevo location "/user/miusuario/bd/miusuario_test2";
```

```
g: jdbc:hive2://> CREATE DATABASE MIUSUARIO_TEST2_NUEVO LOCATION "/user/miusuario/bd/miusuario_test2_nuevo";
OK
No rows affected (2.689 seconds)
g: jdbc:hive2://> show databases;
OK
+-----+
| database_name |
+-----+
| bigdata       |
| bigdata.nueva |
| default       |
| dev_curated   |
| dev_functional |
| dev_landing    |
| dev_workload  |
| dmc            |
| miusuario_test |
| miusuario_test2 |
| miusuario_test2_nuevo |
| miusuario_test3 |
| miusuario_test_nuevo |
+-----+
13 rows collected (0.072 seconds)
g: jdbc:hive2://>
```

```
create table miusuario_test2_nuevo.persona( id string, nombre string, telefono string, correo
string, fecha_ingreso string, edad int, salario double, id_empresa string ) row format delimited
fields terminated by '|' lines terminated by '\n' stored as textfile location
'/user/miusuario/bd/miusuario_test2/persona';
```

```
0: jdbc:chiw2://> CREATE TABLE MIUSUARIO_TEST2_MUVO.PERSONA(
+ .....> ID STRING,
+ .....> NOMBRE STRING,
+ .....> TELEFONO STRING,
+ .....> CORREO STRING,
+ .....> FECHA_INGRESO STRING,
+ .....> EDAD INT,
+ .....> SALARIO DOBLE,
+ .....> ID_EMPRESA STRING
+ .....> )
+ .....> ROW FORMAT DELIMITED
+ .....> FIELDS TERMINATED BY '|'
+ .....> LINES TERMINATED BY '\n'
+ .....> STORED AS TEXTFILE
+ .....> LOCATION '/user/miusuario/bd/miusuario_test2_muvo/persona';
OK
No rows affected (0.441 seconds)
0: jdbc:chiw2://> show tables in miusuario_test2_muvo;
OK
+-----+
| tab_name |
+-----+
| persona |
+-----+
1 row selected (0.086 seconds)
0: jdbc:chiw2://>
```

Si ejecutamos un: `desc formatted miusuario_test2_nuevo.persona;`
Podemos ver la ruta del archivo

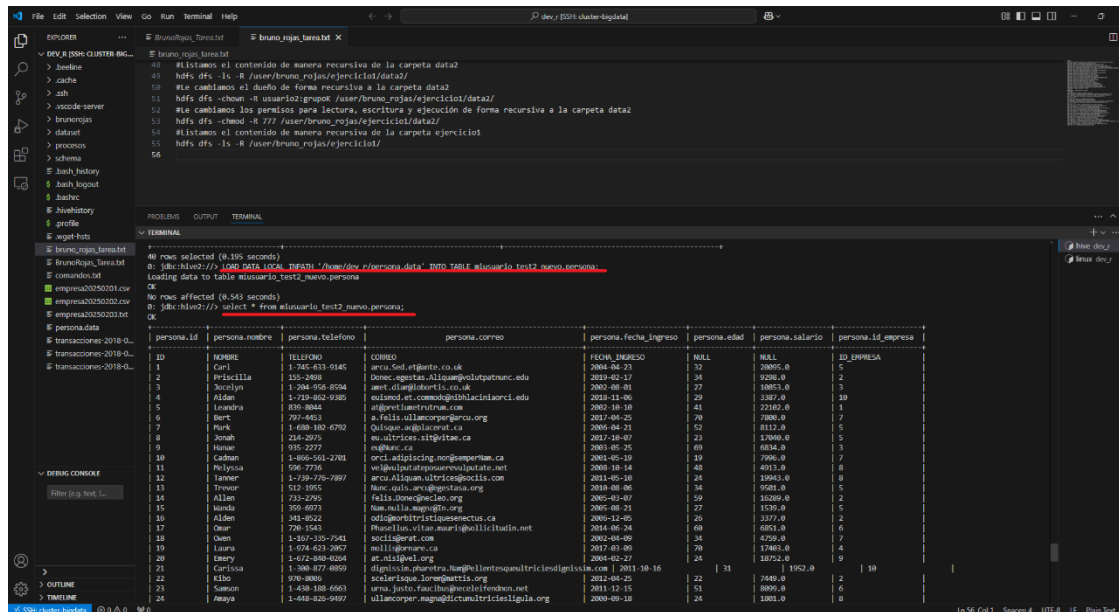
```

--> # Create table
CREATE TABLE miscurso_test2_nuevo_persona (
  id integer NOT NULL,
  nombre character varying(50) NOT NULL,
  telefono character varying(15) NOT NULL,
  correo character varying(50) NOT NULL,
  fecha_ingreso date NOT NULL,
  edad integer NOT NULL,
  salario double precision NOT NULL,
  id_empresa integer NOT NULL,
  CONSTRAINT pk_miscurso_test2_nuevo_persona PRIMARY KEY (id)
);

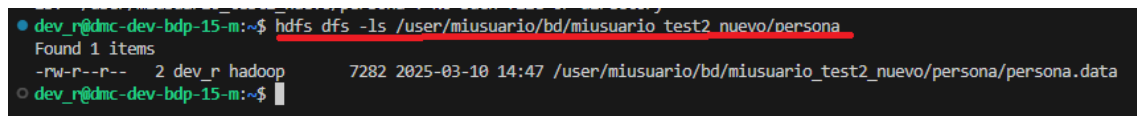
--> # Detailed Table Information
\d miscurso_test2_nuevo_persona

```

```
miusuario_test2_nuevo.persona;
```

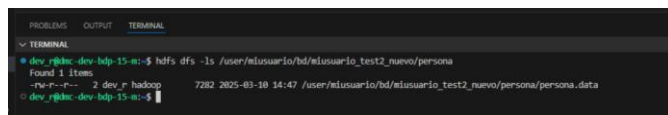


Comprobamos la ruta en hdfs

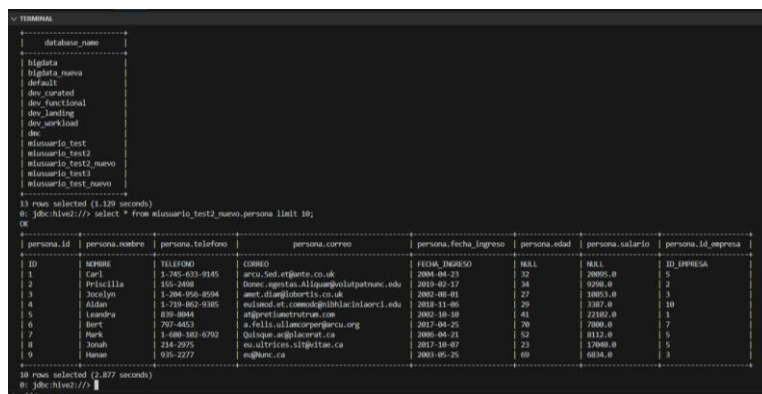


Hemos verificado que el sistema crea la ruta que le indicamos en el location al momento de la creación de la tabla.

```
hdfs dfs -ls /user/miusuario/bd/miusuario test2 nuevo/persona
```



Al hacerle un select vemos que si tiene data.



Al hacer un drop a la tabla
drop table miusuario_test2_nuevo.persona;

```
0: jdbc:hive2://> drop table miusuario_test2_nuevo.persona;
OK
No rows affected (2.649 seconds)
0: jdbc:hive2://>
```

Y luego un select, aparece un error de table not found.

```
Error: Error while compiling statement: FAILED: SemanticException [Error 10001]: Line 1:14 Table not found 'persona' (state=42502,code=10001)
0: jdbc:hive2://>
```

Si hacemos la consulta en hdfs de la ruta de la carpeta persona, nos dirá que no existe.
Por ser una tabla interna (internal) se han eliminado tanto la metada como la información.
Si fuese una tabla externa (external) solo se elimina la metadata pero el archivo en HDFS
(persona.data) seguiría existiendo. Las tablas externas se declaran con el “create external table
...”

```
PROBLEMS 4 OUTPUT TERMINAL
✓ TERMINAL
dev_r@dmc-dev-bdp-15-m:~$ hdfs dfs -ls /user/miusuario/bd/miusuario_test2_nuevo/persona
ls: '/user/miusuario/bd/miusuario_test2_nuevo/persona': No such file or directory
dev_r@dmc-dev-bdp-15-m:~$
```

Entonces, vamos a crear una tabla external para validar.

create external table miusuario_test2_nuevo.persona (id string, nombre string, telefono string,
correo string, fecha_ingreso string, edad int, salario double, id_empresa string) row format
delimited fields terminated by '|' lines terminated by '\n' stored as textfile location
'/user/miusuario/bd/miusuario_test2/persona';

```
0: jdbc:hive2://> CREATE EXTERNAL TABLE MIUSUARIO_TEST2_NUEVO.PERSONA(
. . . . . > ID STRING,
. . . . . > NOMBRE STRING,
. . . . . > TELEFONO STRING,
. . . . . > CORREO STRING,
. . . . . > FECHA_INGRESO STRING,
. . . . . > EDAD INT,
. . . . . > SALARIO DOUBLE,
. . . . . > ID_EMPRESA STRING)
. . . . . > ROW FORMAT DELIMITED
. . . . . > FIELDS TERMINATED BY '|'
. . . . . > LINES TERMINATED BY '\n'
. . . . . > STORED AS TEXTFILE
. . . . . > LOCATION '/user/miusuario/bd/miusuario_test2_nuevo/persona';
OK
No rows affected (1.003 seconds)
0: jdbc:hive2://>
```

0

Ln 56, Col 1 Spaces: 4 UTF-8 LF Plain Text

Vemos que se creo nuevamente la ruta

```
PROBLEMS OUTPUT TERMINAL
dev_r@mc-dev-bdp-15-m:~$ hdfs dfs -ls /user/miusuario/bd/miusuario_test2_nuevo/persona
dev_r@mc-dev-bdp-15-m:~$
```

Cargamos el archivo persona.data a la ruta

LOAD DATA LOCAL INPATH '/home/dev_r/persona.data' INTO TABLE
miusuario_test2_nuevo.persona;

```
0: jdbc:hive2://> LOAD DATA LOCAL INPATH '/home/dev_r/persona.data' INTO TABLE miusuario_test2_nuevo.persona;
Loading data to table miusuario_test2_nuevo.persona
OK
No rows affected (1.659 seconds)
0: jdbc:hive2://> select * from miusuario_test2_nuevo.persona limit 10;
OK
+-----+-----+-----+-----+-----+-----+-----+-----+
| persona.id | persona.nombre | persona.telefono | persona.correo | persona.fecha_ingreso | persona.edad | persona.salario | persona.id_empresa |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | NOMBRE | TELEFONO | CORREO | FECHA_INGRESO | NULL | NULL | ID_EMPRESA |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Carl | 1-745-633-9145 | arcu.Sed.et@ante.co.uk | 2004-04-23 | 32 | 20095.0 | 5 |
| 2 | Priscilla | 155-2498 | Donec.egestas.Aliquam@volutpatnunc.edu | 2019-02-17 | 34 | 9288.0 | 2 |
| 3 | Jocelyn | 1-204-956-8594 | amet.diam@lobortis.co.uk | 2002-08-01 | 27 | 10853.0 | 3 |
| 4 | Aidan | 1-719-862-9385 | euismod.et.commodo@nibhislaciniacrci.edu | 2018-11-06 | 29 | 3387.0 | 10 |
| 5 | Leandra | 839-8844 | at@pretiumetrutrum.com | 2002-10-10 | 41 | 22102.0 | 1 |
| 6 | Bert | 797-4453 | a.felis.ullamcorper@arcu.org | 2017-04-25 | 70 | 7800.0 | 7 |
| 7 | Mark | 1-680-102-6792 | Quisque.ac@placerat.ca | 2006-04-21 | 52 | 8112.0 | 5 |
| 8 | Jonah | 214-2975 | eu.ultrices.sit@vitae.ca | 2017-10-07 | 23 | 17040.0 | 5 |
| 9 | Hanae | 935-2277 | eu@nunc.ca | 2003-05-25 | 69 | 6834.0 | 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows selected (2.284 seconds)
0: jdbc:hive2://> show tables in miusuario_test2_nuevo;
OK
+-----+
| tab_name |
+-----+
| persona |
+-----+
1 row selected (0.125 seconds)
0: jdbc:hive2://>
```

Verificamos también que en la ruta estén los datos.

```
PROBLEMS OUTPUT TERMINAL
dev_r@mc-dev-bdp-15-m:~$ hdfs dfs -ls /user/miusuario/bd/miusuario_test2_nuevo/persona
dev_r@mc-dev-bdp-15-m:~$ hdfs dfs -ls /user/miusuario/bd/miusuario_test2_nuevo/persona
Found 1 items
-rw-r--r-- 2 dev_r hadoop 7282 2025-03-11 15:47 /user/miusuario/bd/miusuario_test2_nuevo/persona/persona.data
dev_r@mc-dev-bdp-15-m:~$
```

Luego de hacer el drop table validamos que la tabla ya no existe en la base de datos, y al hacer select nos muestra error. Es decir, se eliminó la metadata.

```
0: jdbc:hive2://> drop table miusuario_test2_nuevo.persona;
OK
No rows affected (0.181 seconds)
0: jdbc:hive2://> show tables in miusuario_test2_nuevo;
OK
+-----+
| tab_name |
+-----+
No rows selected (0.083 seconds)
0: jdbc:hive2://> select * from miusuario_test2_nuevo.persona limit 10;
25/03/11 15:52:06 [8052a0a3-7ca9-4045-be06-9df8af324afc main]: ERROR parse.CalcitePlanner: org.apache.hadoop.hive.ql.parse.SemanticException: Line 1:14 Table not found 'persona'
```

Sin embargo, la data y el directorio siguen existiendo si lo validamos en hdfs.

```
PROBLEMS OUTPUT TERMINAL
dev_r@mc-dev-bdp-15-m:~$ hdfs dfs -ls /user/miusuario/bd/miusuario_test2_nuevo/persona
Found 1 items
-rw-r--r-- 2 dev_r hadoop 7282 2025-03-11 15:47 /user/miusuario/bd/miusuario_test2_nuevo/persona/persona.data
dev_r@mc-dev-bdp-15-m:~$
```

Diferencia entre drop y truncate? Truncate solo elimina la data pero mantiene la estructura, el drop elimina la data (contenido) y la estructura de la tabla.

Truncate no funciona sobre tablas externas porque no permite eliminar el contenido o la data.

Al ejecturar comandos sql en hive, se pueden utilizar validadores que eviten que la secuencia de comandos se caiga, como por ejemplo IF EXISTS, IF NOT EXISTS.

Creamos una nueva base de datos
create database dmc_nuevo;

```
0: jdbc:hive2://> create database dmc_nuevo;
OK
No rows affected (0.094 seconds)
0: jdbc:hive2://> create database dmc_nuevo;
25/03/11 16:13:46 [HiveServer2-Background-Pool: Thread-67]: ERROR exec.DDLTask: Failed
org.apache.hadoop.hive.ql.metadata.HiveException: Database dmc_nuevo already exists
```

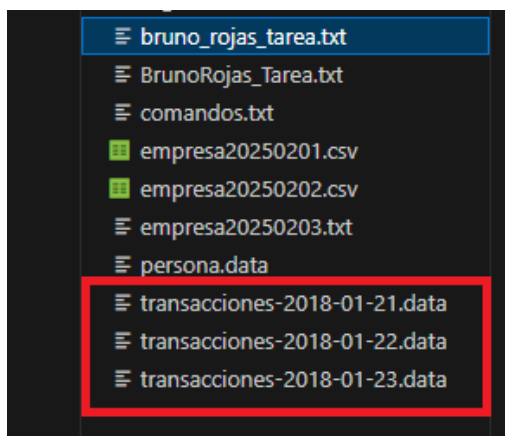
Crear tabla en formato PARQUET
En la tabla en formato PARQUET no es necesario indicar:
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
LINES TERMINATED BY '\n'

create table miusuario_test3_nuevo.persona (id string, nombre string, telefono string, correo string, fecha_ingreso string, edad int, salario double, id_empresa string) stored as parquet location '/user/miusuario/bd/miusuario_test3/persona_parquet';

```
0: jdbc:hive2://> create database if not exists miusuario_test3_nuevo;
OK
No rows affected (0.086 seconds)
0: jdbc:hive2://> CREATE TABLE IF NOT EXISTS MIUSUARIO_TEST3_NUEVO.PERSONA(
. . . . . > ID STRING,
. . . . . > NOMBRE STRING,
. . . . . > TELEFONO STRING,
. . . . . > CORREO STRING,
. . . . . > FECHA_INGRESO STRING,
. . . . . > EDAD INT,
. . . . . > SALARIO DOUBLE,
. . . . . > ID_EMPRESA STRING)
. . . . . > STORED AS PARQUET
. . . . . > LOCATION '/user/miusuario/bd/miusuario_test3_nuevo/persona_parquet';
OK
No rows affected (0.129 seconds)
0: jdbc:hive2://>
```

Tablas particionadas, permiten una búsqueda más rápida por el campo que se ha realizado la partición.

Cargamos archivos de transacciones al servidor Linux




```
create table miusuario_test_nuevo.transaccion( id_persona string, id_empresa string, monto
double ) partitioned by (fecha string) row format delimited fields terminated by '|' lines
terminated by '\n' stored as textfile location '/user/miusuario/bd/miusuario_test/transaccion';
show tables in miusuario_test;
select * from miusuario_test_nuevo.transaccion limit 10;
```

Si queremos que se salte el encabezado se puede realizar en la definición de la tabla o alterando la tabla.

```
CREATE TABLE temp
```

```
(
    name STRING,
    id INT
)
```

```
row format delimited fields terminated BY '\t'
```

```
lines terminated BY '\n'
```

```
tblproperties("skip.header.line.count"="1");
```

```
alter table tablename set tblproperties ("skip.header.line.count"="1");
```

```
load data local inpath '/home/dev_r/transacciones-2018-01-22.data' overwrite into table
miusuario_test.transaccion partition (fecha='2018-01-22');
```

```
1 row selected (0.721 seconds)
0: jdbc:hive2://> ALTER TABLE MIUSUARIO_TEST_NUEVO.TRANSACCION SET TBLPROPERTIES ("skip.header.line.count"="1");
OK
No rows affected (0.18 seconds)
0: jdbc:hive2://> LOAD DATA LOCAL INPATH '/home/dev_r/transacciones-2018-01-22.data' OVERWRITE INTO TABLE MIUSUARIO_TEST_NUEVO.TRANSACCION PARTITION (FECHA="2018-01-22");
Loading data to table miusuario_test_nuevo.transaccion partition (fecha=2018-01-22)
OK
No rows affected (0.581 seconds)
0: jdbc:hive2://> select count(*) from MIUSUARIO_TEST_NUEVO.TRANSACCION;
Query ID = dev_r_20250311170926_9c17e9c6-3844-4797-bdb8-af0c64870efe
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1741701656871_0002)

OK
+-----+
| _c0 |
+-----+
| 111680 |
+-----+
1 row selected (25.644 seconds)
0: jdbc:hive2://> SHOW PARTITIONS MIUSUARIO_TEST_NUEVO.TRANSACCION;
OK
+-----+
| partition |
+-----+
| fecha=2018-01-21 |
| fecha=2018-01-22 |
+-----+
2 rows selected (0.157 seconds)
0: jdbc:hive2://> █
```

```
load data local inpath '/home/dev_r/transacciones-2018-01-23.data' overwrite into table
miusuario_test.transaccion partition (fecha='2018-01-23');
show partitions miusuario_test_nuevo.transaccion;
```

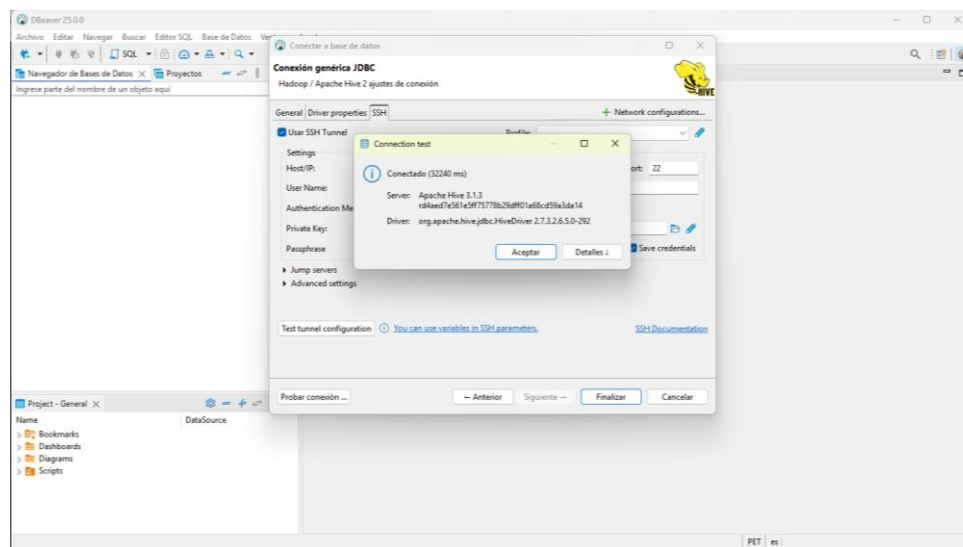
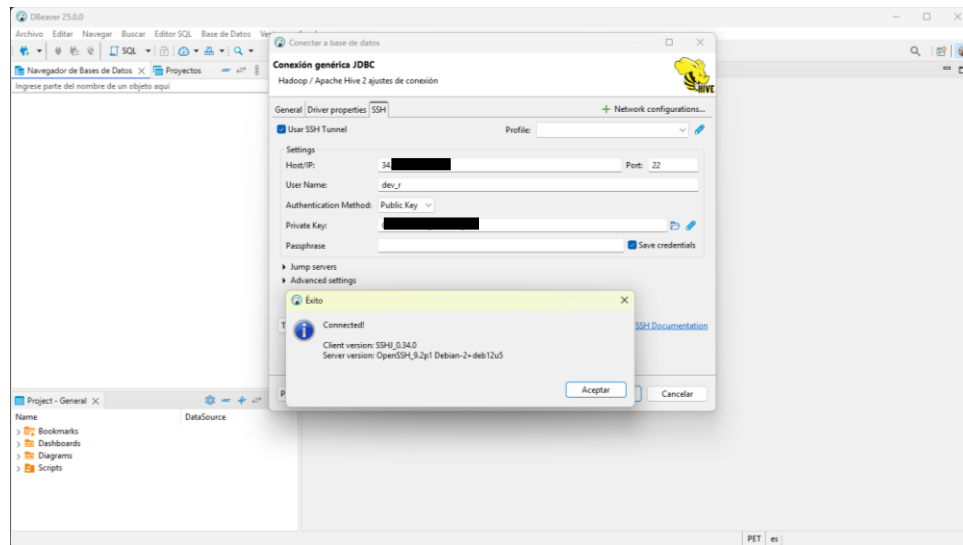
```
0: jdbc:hive2://> LOAD DATA LOCAL INPATH '/home/dev_r/transacciones-2018-01-23.data' OVERWRITE INTO TABLE MIUSUARIO_TEST_NUEVO.TRANSACCION PARTITION (FECHA="2018-01-23");
Loading data to table miusuario_test_nuevo.transaccion partition (fecha=2018-01-23)
OK
No rows affected (0.499 seconds)
0: jdbc:hive2://> select count(*) from MIUSUARIO_TEST_NUEVO.TRANSACCION;
Query ID = dev_r_20250311171113_a6e3f939-4cdd-472e-8053-e3e3f0aaa965
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1741701656871_0002)

OK
+-----+
| _c0 |
+-----+
| 235040 |
+-----+
1 row selected (9.68 seconds)
0: jdbc:hive2://> SHOW PARTITIONS MIUSUARIO_TEST_NUEVO.TRANSACCION;
OK
+-----+
| partition |
+-----+
| fecha=2018-01-21 |
| fecha=2018-01-22 |
| fecha=2018-01-23 |
+-----+
3 rows selected (0.174 seconds)
0: jdbc:hive2://> █
```

En hdfs validamos las rutas que se han creado para las pariticones
hdfs dfs -ls /user/miusuario/bd/miusuario_test_nuevo/transacción

```
dev_r@dev-bdp-15-m:~$ hdfs dfs -ls /user/miusuario/bd/miusuario_test_nuevo/transaccion/
Found 3 items
drwxr-xr-x - dev_r hadoop 0 2025-03-11 16:51 /user/miusuario/bd/miusuario_test_nuevo/transaccion/fecha-2018-01-21
drwxr-xr-x - dev_r hadoop 0 2025-03-11 17:09 /user/miusuario/bd/miusuario_test_nuevo/transaccion/fecha-2018-01-22
drwxr-xr-x - dev_r hadoop 0 2025-03-11 17:11 /user/miusuario/bd/miusuario_test_nuevo/transaccion/fecha-2018-01-23
dev_r@dev-bdp-15-m:~$ hdfs dfs -ls -R /user/miusuario/bd/miusuario_test_nuevo/transaccion/
drwxr-xr-x - dev_r hadoop 0 2025-03-11 16:51 /user/miusuario/bd/miusuario_test_nuevo/transaccion/fecha-2018-01-21
538827 2025-03-11 16:51 /user/miusuario/bd/miusuario_test_nuevo/transaccion/fecha-2018-01-21/transacciones-2018-01-21.data
drwxr-xr-x - dev_r hadoop 0 2025-03-11 17:09 /user/miusuario/bd/miusuario_test_nuevo/transaccion/fecha-2018-01-22
668498 2025-03-11 17:09 /user/miusuario/bd/miusuario_test_nuevo/transaccion/fecha-2018-01-22/transacciones-2018-01-22.data
drwxr-xr-x - dev_r hadoop 0 2025-03-11 17:11 /user/miusuario/bd/miusuario_test_nuevo/transaccion/fecha-2018-01-23
1336418 2025-03-11 17:11 /user/miusuario/bd/miusuario_test_nuevo/transaccion/fecha-2018-01-23/transacciones-2018-01-23.data
dev_r@dev-bdp-15-m:~$
```

Conexión a través de DBeaver



DBeaver 25.0.0 - localhost - Script

Archivo Editor Navegador Base de Datos Ventana Ayuda

Auto SQL Connect Refresh Bulk Edit Localhost default localhost Script X

Navegador de Base de Datos X Proyectos

Ingrese parte del nombre de un objeto aquí

localhost

- bigdata
- bigdata_nuevo
- default
- den_curated
- den_functional
- den_landing
- den_workload
- dmic
- dmic_nuevo
- miusuario_test
- miusuario_test2
- miusuario_test2_nuevo
- miusuario_test3
- miusuario_test2_nuevo
- miusuario_test_nuevo
- Tablas
 - persona
 - Columnas
 - Claves
 - Columnas de clave externa
 - Referencias
 - transaccion
 - Columnas
 - id_persona (STRING)
 - id_empresa (STRING)
 - monto (DOUBLE)
 - fecha (STRING)
 - Claves
 - Columnas de clave externa
 - Referencias
 - Vistas
 - Procedimientos

Project - General X

DataSource

Name

- Bookmarks
- Dashboards
- Diagrams
- Scripts

select * from miusuario_test_nuevo.transaccion

Resultados 1 X

select * from miusuario_test_nuevo.transaccion Enter a SQL expression to filter results (use Ctrl+Space)

	id_persona	id_empresa	monto	fecha
1	31	1	3.142	2018-01-21
2	43	5	962	2018-01-21
3	40	3	264	2018-01-21
4	83	9	2.996	2018-01-21
5	49	5	3.418	2018-01-21
6	29	4	2.071	2018-01-21
7	97	6	2.589	2018-01-21
8	37	4	579	2018-01-21
9	53	4	206	2018-01-21
10	76	6	2.878	2018-01-21
11	14	8	1.610	2018-01-21
12	42	5	2.697	2018-01-21
13	45	9	2.011	2018-01-21
14	11	3	2.903	2018-01-21
15	66	10	1.771	2018-01-21
16	42	3	1.621	2018-01-21
17	87	1	3.036	2018-01-21
18	04	1	1.764	2018-01-21
total				

Valor X

31

Remove Save Cancel Exportar datos... 200 200 200 row(s) fetched - 1.872s (5.970s fetch), on 2023-03-11 at 15:37:36

PET es Editable Inserción inteligente 1:48:47 Set 0/0

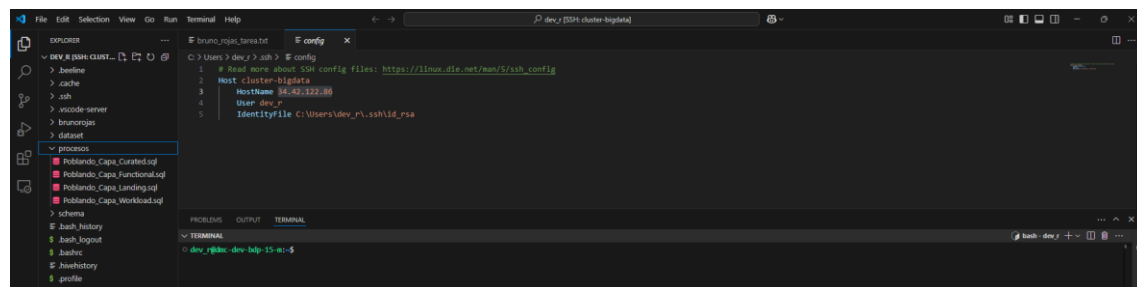
CLASE 06 -PROCESAMIENTO DATALAKE CON APACHE HIVE

CAPAS: WORKLOAD – LANDING – CURATED(Historia + Reglas de negocio) –
FUNCTIONAL(Transacción Enriquecida)

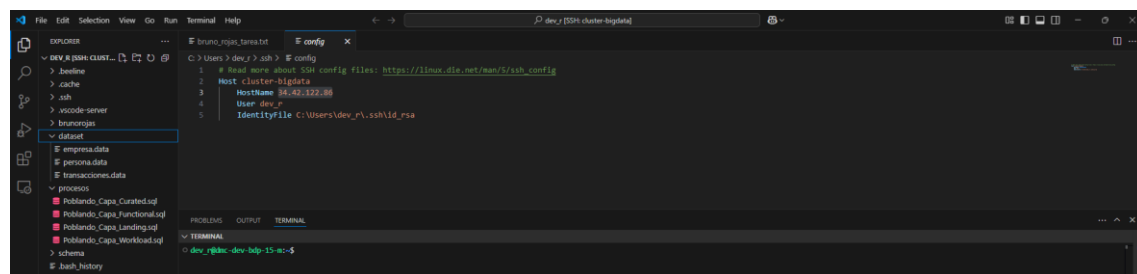
Con los archivos que nos ha pasado el profesor hacemos lo siguiente

Nombre	Fecha de modificación	Tipo	Tamaño
_MACOSX	24/02/2025 19:59	Carpeta de archivos	
empresa.avsc	20/07/2023 20:10	Archivo AVSC	1 KB
empresa.data	5/06/2023 20:08	Archivo DATA	1 KB
persona.avsc	20/07/2023 20:10	Archivo AVSC	1 KB
persona.data	5/06/2023 20:08	Archivo DATA	8 KB
Poblando Datalake.zip	12/12/2024 20:41	Carpeta compres...	1,160 KB
Poblando_Capa_Curated.sql	5/06/2023 20:08	Archivo de origen ...	6 KB
Poblando_Capa_Functional.sql	5/06/2023 20:08	Archivo de origen ...	9 KB
Poblando_Capa_Landing.sql	5/06/2023 20:08	Archivo de origen ...	5 KB
Poblando_Capa_Workload.sql	5/06/2023 20:08	Archivo de origen ...	5 KB
transaccion.avsc	20/07/2023 20:10	Archivo AVSC	1 KB
transacciones.data	15/12/2018 05:59	Archivo DATA	5,009 KB

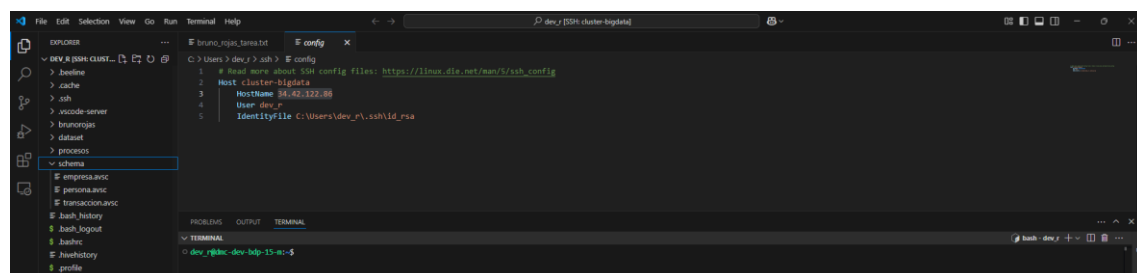
Creamos una carpeta procesos en el servidor y agregamos los archivos sql.



Creamos otra carpeta dataset y agregamos los archivos .data



Creamos otra carpeta que se llame schema y guardamos los archivos .avsc.



En lugar de abrir una ventana interactiva en Beeline, vamos a decirle que cargue un archivo.

Para ello primero verificamos la ruta donde nos encontramos con pwd.

Verificamos que la ruta de la carpeta procesos sería: /home/dev_r/procesos

```
✓ TERMINAL
● dev_r@dmc-dev-bdp-15-m:~$ pwd
/home/dev_r
● dev_r@dmc-dev-bdp-15-m:~$ ls -la
total 2588
drwxr-xr-x 10 dev_r dev_r 4096 Mar  9 03:01 .
drwxr-xr-x  5 root  root 4096 Mar 11 01:42 ..
-rw-r----- 1 dev_r dev_r 11492 Mar 11 20:38 .bash_history
-rw-r--r--  1 dev_r dev_r 220 Mar 29 2024 .bash_logout
-rw-r--r--  1 dev_r dev_r 3526 Mar 29 2024 .bashrc
drwxr-xr-x  2 dev_r dev_r 4096 Feb 18 02:23 .beeline
drwxr-xr-x  3 dev_r dev_r 4096 Feb 11 03:32 .cache
-rw-r--r--  1 dev_r dev_r 16 Mar  9 03:01 .hivehistory
-rw-r--r--  1 dev_r dev_r 807 Mar 29 2024 .profile
drwx----- 2 dev_r dev_r 4096 Mar 17 02:48 .ssh
drwxr-x---  5 dev_r dev_r 4096 Mar 17 02:49 .vscode-server
-rw-r--r--  1 dev_r dev_r 183 Mar 11 14:03 .wget-hsts
-rw-r--r--  1 dev_r dev_r 2947 Feb 18 00:57 BrunoRojas_Tarea.txt
-rw-r--r--  1 dev_r dev_r 3112 Mar  8 03:39 bruno_rojas_tarea.txt
drwxr-xr-x  2 dev_r dev_r 4096 Feb 17 23:30 brunorojas
-rw-r--r--  1 dev_r dev_r 5115 Feb 25 03:46 comandos.txt
drwxr-xr-x  2 dev_r dev_r 4096 Feb 25 01:02 dataset
-rw-r--r--  1 dev_r dev_r  0 Feb 13 02:56 empresa20250201.csv
-rw-r--r--  1 dev_r dev_r  0 Feb 13 02:56 empresa20250202.csv
-rw-r--r--  1 dev_r dev_r  0 Feb 13 02:57 empresa20250203.txt
-rw-r--r--  1 dev_r dev_r 7282 Feb 13 02:39 persona.data
drwxr-xr-x  2 dev_r dev_r 4096 Feb 25 01:01 procesos
drwxr-xr-x  2 dev_r dev_r 4096 Feb 25 01:03 schema
-rw-r--r--  1 dev_r dev_r 538827 Feb 20 03:04 transacciones-2018-01-21.data
-rw-r--r--  1 dev_r dev_r 668498 Feb 20 03:04 transacciones-2018-01-22.data
-rw-r--r--  1 dev_r dev_r 1336418 Feb 20 03:04 transacciones-2018-01-23.data
○ dev_r@dmc-dev-bdp-15-m:~$
```

La Capa Workload

Dentro del script que nos han pasado “Poblando_Capa_Workload.sql” modificamos la ruta del archivo.

```
bruno_rojas_tarea.txt Poblando_Capa_Workload.sql X
procesos > Poblando_Capa_Workload.sql
1  -- -----
2  -- COMANDO DE EJECUCION
3  --Esta no usamos
4  -- beeline -u jdbc:hive2:// -f /home/dev_r/procesos/Poblando_Capa_Workload.sql --hiveconf "PARAM_USERNAME=juan" --hiveconf "ENV=DEV"
5  --Usamos esta
6  --beeline -u jdbc:hive2:// -f /home/dev_r/procesos/Poblando_Capa_Workload.sql
7  -- -----
8  --
9  -- @section 1. Definición de parámetros
10 --
11 -- -----
12
13 -- [HIVE] Creamos una variable en HIVE
14 SET ENV=dev;
15 SET PARAM_USERNAME=dev_r;
16 -- -----
17 --
18 -- @section 2. Eliminación de base de datos
19 --
20
```

```

-- -----
-- COMANDO DE EJECUCION
-- Esta no usamos
-- beeline -u jdbc:hive2://-f /home/dev_r/procesos/Poblando_Capa_Workload.sql --hiveconf "PARAM_USERNAME=juan" --hiveconf "ENV=DEV"
-- Usamos esta
-- beeline -u jdbc:hive2://-f /home/dev_r/procesos/Poblando_Capa_Workload.sql
-- -----
--
-- @section 1. Definición de parámetros
-- -----

-- [HIVE] Creamos una variable en HIVE
SET ENV=dev;
SET PARAM_USERNAME=dev_r;
-- -----
--
-- @section 2. Eliminación de base de datos
-- -----

-- Eliminación de bases de datos
DROP DATABASE IF EXISTS ${hiveconf:ENV}_workload CASCADE;
-- -----
--
-- @section 3. Creación de base de datos
-- -----

-- Creación de base de datos
CREATE DATABASE IF NOT EXISTS ${hiveconf:ENV}_workload LOCATION 'user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_workload/';
-- -----
--
-- @section 4. Despliegue de tabla PERSONA
-- -----

-- Creación de tabla
-- Creación de tabla PERSONA
CREATE TABLE ${hiveconf:ENV}_workload.PERSONA(
  ID STRING,
  NOMBRE STRING,
  TELEFONO STRING,
  CORREO STRING,
  FECHA_INGRESO STRING,
  EDAD STRING,
  SALARIO STRING,
  ID_EMPRESA STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION 'user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_workload/persona'
TBLPROPERTIES(
  'skip.header.line.count'='1',
  'store.charset'='ISO-8859-1',
  'retrieve.charset'='ISO-8859-1'
);

-- Subida de datos
LOAD DATA LOCAL INPATH '/home/${hiveconf:PARAM_USERNAME}/dataset/persona.data'
INTO TABLE ${hiveconf:ENV}_workload.PERSONA;

-- Impresión de datos
SELECT * FROM ${hiveconf:ENV}_workload.PERSONA LIMIT 10;
-- -----
--
-- @section 5. Despliegue de tabla EMPRESA
-- -----

-- Creación de tabla EMPRESA
CREATE TABLE ${hiveconf:ENV}_workload.EMPRESA(
  ID STRING,
  NOMBRE STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION 'user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_workload/empresa'
TBLPROPERTIES(
  'skip.header.line.count'='1',
  'store.charset'='ISO-8859-1',
  'retrieve.charset'='ISO-8859-1'
);

-- Subida de datos
LOAD DATA LOCAL INPATH '/home/${hiveconf:PARAM_USERNAME}/dataset/empresa.data'
INTO TABLE ${hiveconf:ENV}_workload.EMPRESA;

-- Impresión de datos
SELECT * FROM ${hiveconf:ENV}_workload.EMPRESA LIMIT 10;
-- -----

```



```
--
-- @section 6. Despliegue de tabla TRANSACCION
--
-----

-- Creaci3n de tabla Transacciones

CREATE TABLE ${hiveconf:ENV}_workload.TRANSACCION(
  ID_PERSONA STRING,
  ID_EMPRESA STRING,
  MONTO STRING,
  FECHA STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_workload/transaccion'
TBLPROPERTIES(
  'skip.header.line.count'='1',
  'store.charset'='ISO-8859-1',
  'retrieve.charset'='ISO-8859-1'
);

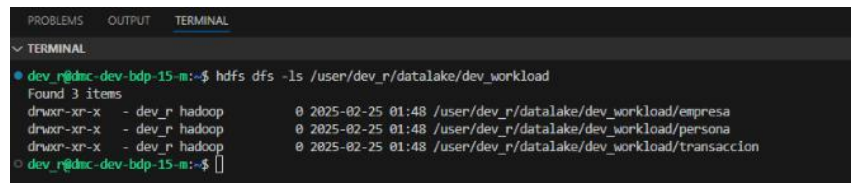
-- Subida de datos
LOAD DATA LOCAL INPATH '/home/${hiveconf:PARAM_USERNAME}/dataset/transacciones.data'
INTO TABLE ${hiveconf:ENV}_workload.TRANSACCION;

-- Impresi3n de datos
SELECT * FROM ${hiveconf:ENV}_workload.TRANSACCION LIMIT 10;
```

Se carga el archivo con el comando:

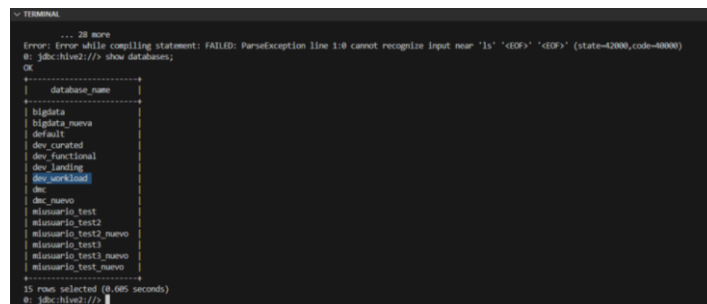
```
beeline -u jdbc:hive2:// -f /home/dev_r/procesos/Poblando_Capa_Workload.sql
```

Validamos los directorios creados para las tablas

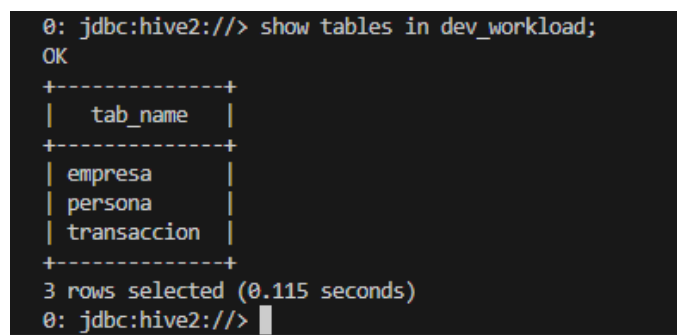


```
dev_r@dmcc-dev-bdp-15-m:~$ hdfs dfs -ls /user/dev_r/datalake/dev_workload
Found 3 items
drwxr-xr-x - dev_r hadoop      0 2025-02-25 01:48 /user/dev_r/datalake/dev_workload/empresa
drwxr-xr-x - dev_r hadoop      0 2025-02-25 01:48 /user/dev_r/datalake/dev_workload/persona
drwxr-xr-x - dev_r hadoop      0 2025-02-25 01:48 /user/dev_r/datalake/dev_workload/transaccion
dev_r@dmcc-dev-bdp-15-m:~$
```

Luego de la carga podemos entrar a beeline y validar con un show databases;



```
... 28 more
Error: Error while compiling statement: FAILED: ParseException line 1:0 cannot recognize input near 'ls' '<EOF>' '<EOF>' (state=2000,code=40000)
0: jdbc:hive2://> show databases;
OK
+-----+
| database_name |
+-----+
| hqdata        |
| hqdata_nueva  |
| default       |
| dev_curated   |
| dev_functional |
| dev_landing   |
| dev_workload  |
| dmcc          |
| dmcc_nuevo    |
| minisuario_test |
| minisuario_test2 |
| minisuario_test2_nuevo |
| minisuario_test3 |
| minisuario_test3_nuevo |
| minisuario_test_nuevo |
+-----+
15 rows selected (0.005 seconds)
0: jdbc:hive2://>
```



```
0: jdbc:hive2://> show tables in dev_workload;
OK
+-----+
| tab_name |
+-----+
| empresa  |
| persona  |
| transaccion |
+-----+
3 rows selected (0.115 seconds)
0: jdbc:hive2://>
```

La Capa Landing

```
procesos > Poblado_Capa_Landing.sql
1  -----
2
3  -- COMANDO DE EJECUCION
4  -- beeline -u jdbc:hive2:// -f /home/dev_r/procesos/Poblado_Capa_Landing.sql --hiveconf "PARAM_USERNAME=juan" --hiveconf "ENV=DEV"
5
6  -----
7  --
8  -- @section 1. Definición de parámetros
9  --
10 -----
11
12 -- [HIVE] Creamos una variable en HIVE
13 SET ENV=dev;
14 SET PARAM_USERNAME=dev_r;
15
16 -----
```

Dentro de las opciones, se le indica el tipo de compresión del archivo, hoy en día el estándar es snappy. Hay otros como Gzip o Bzip.

Activamos el particionamiento dinámico en Hive, lo que permite que Hive cree particiones en una tabla de manera automática.

El indicador nonstrict indica que se debe agregar la partición sin eliminar las anteriores.

```
41 -- Compresión
42 SET hive.exec.compress.output=true;
43 SET avro.output.codec=snappy; Gzip
44
45 -- Particionamiento dinámico
46 SET hive.exec.dynamic.partition=true;
47 SET hive.exec.dynamic.partition.mode=nonstrict;
48
```

Podemos observar que en la definición de la tabla persona para la capa landing ya no se define el esquema, sin embargo si va a utilizar el esquema que subimos a la carpeta schema en formato avsc.

También se indica que el formato para almacenar será AVRO y ya no textfield.

La carga de datos ya no se hace de archivos externos, si no de la capa anterior.

```
42 CREATE TABLE ${hiveconf:ENV}_workload.PERSONA(
43     ID STRING,
44     NOMBRE STRING,
45     TELEFONO STRING,
46     CORREO STRING,
47     FECHA_INGRESO STRING,
48     EDAD STRING,
49     SALARIO STRING,
50     ID_EMPRESA STRING
51 )
52 ROW FORMAT DELIMITED
53 FIELDS TERMINATED BY '|'
54 LINES TERMINATED BY '\n'
55 STORED AS TEXTFILE
56 LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveco
57
58 -- Creación de tabla
59 CREATE TABLE ${hiveconf:ENV}_LANDING.PERSONA
60 STORED AS AVRO
61 LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveco
62
63 -- Inserción de datos
64 INSERT INTO TABLE ${hiveconf:ENV}_LANDING.PERSONA
65 SELECT * FROM ${hiveconf:ENV}_workload.PERSONA;
66
67 -- Impresión de datos
```

```
EXPLORER
DEV_R [SSH: CLUST...
> .beeline
> .cache
> .ssh
> .vscode-server
> brunorojas
> dataset
> procesos
  Poblado_Capa_Curated.sql
  Poblado_Capa_Functional.sql
  Poblado_Capa_Landing.sql
  Poblado_Capa_Workload.sql
> schema
  empresa.avsc
  persona.avsc
  transaccion.avsc

schema > persona.avsc
1  {
2    "name": "PERSONA",
3    "type": "record",
4    "fields": [
5      {"name": "ID", "type": ["string", "null"]},
6      {"name": "NOMBRE", "type": ["string", "null"]},
7      {"name": "TELEFONO", "type": ["string", "null"]},
8      {"name": "CORREO", "type": ["string", "null"]},
9      {"name": "FECHA_INGRESO", "type": ["string", "null"]},
10     {"name": "EDAD", "type": ["string", "null"]},
11     {"name": "SALARIO", "type": ["string", "null"]},
12     {"name": "ID_EMPRESA", "type": ["string", "null"]}
13   ]
14 }
```

A la tabla transaccion se le indica que va a estar particionada por el campo fecha tanto al momento de la definición como al de la inserción o población de los datos.

```
103 -- Creación de tabla
104 -- Creación de tabla TRANSACCION
105 CREATE TABLE ${hiveconf:ENV}_LANDING.TRANSACCION
106 PARTITIONED BY (FECHA STRING)
107 STORED AS AVRO
108 LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_LANDING/transaccion'
109 TBLPROPERTIES(
110     'store.charset'='ISO-8859-1',
111     'retrieve.charset'='ISO-8859-1',
112     'avro.schema.url'='hdfs:///user/${hiveconf:PARAM_USERNAME}/datalake/schema/${hiveconf:ENV}_LAN
113     'avro.output.codec'='snappy'
114 );
115
116 -- Inserción de datos por particionamiento dinámico
117 INSERT INTO TABLE ${hiveconf:ENV}_LANDING.TRANSACCION
118 PARTITION(FECHA)
119 SELECT * FROM ${hiveconf:ENV}_workload.TRANSACCION;
```

Sin embargo ese campo fecha no está en la definición de la tabla en el archivo .avsc

```
bruno_rojas_tarea.txt  Poblando_Capa_Landing.sql  transaccion.avsc X
schema > transaccion.avsc
1 {
2   "name": "TRANSACCION",
3   "type": "record",
4   "fields": [
5     {"name": "ID_PERSONA", "type": ["string", "null"]},
6     {"name": "ID_EMPRESA", "type": ["string", "null"]},
7     {"name": "MONTO", "type": ["string", "null"]}
8   ]
9 }
```

```
0: jdbc:hive2://> show partitions dev_landing.transaccion;
OK
+-----+
| partition |
+-----+
| fecha=2018-01-21 |
| fecha=2018-01-22 |
| fecha=2018-01-23 |
+-----+
3 rows selected (0.471 seconds)
```

Como la creación de las tablas tiene definida la siguiente ruta en hdfs, hay que crearla y subir ahí los archivos avsc que contienen el esquema.

```
TBLPROPERTIES(
  'store.charset'='ISO-8859-1',
  'retrieve.charset'='ISO-8859-1',
  'avro.schema.url'='hdfs:///user/${hiveconf:PARAM_USERNAME}/datalake/schema/${hiveconf:ENV}_LANDING/transaccion.avsc',
  'avro.output.codec'='snappy'
);
```

Creamos carpeta

Hdfs dfs -mkdir -p /user/dev_r/datalake/schema/dev_LANDING

Subimos esquemas

Hdfs dfs -put /home/dev_r/schema/*.avsc /user/dev_r/datalake/schema/dev_LANDING/

/user/dev_r/datalake/schema/dev_landing/persona.avsc

/user/dev_r/datalake/schema/dev_landing/empresa.avsc

/user/dev_r/datalake/schema/dev_landing/transaccion.avsc

```
dev_r@dmc-dev-bdp-15-m:~$ hdfs dfs -ls /user/dev_r/datalake/schema/dev_LANDING
Found 3 items
-rw-r--r--  2 dev_r hadoop      160 2025-02-25 02:42 /user/dev_r/datalake/schema/dev_LANDING/empresa.avsc
-rw-r--r--  2 dev_r hadoop      498 2025-02-25 02:42 /user/dev_r/datalake/schema/dev_LANDING/persona.avsc
-rw-r--r--  2 dev_r hadoop      226 2025-02-25 02:42 /user/dev_r/datalake/schema/dev_LANDING/transaccion.avsc
dev_r@dmc-dev-bdp-15-m:~$
```

Ahora ya se puede cargar el archivo .sql con el comando:

beeline -u jdbc:hive2:// -f /home/dev_r/procesos/Poblando_Capa_Landing.sql --hiveconf "PARAM_USERNAME=juan" --hiveconf "ENV=DEV"

```
-- COMANDO DE EJECUCION
-- beeline -u jdbc:hive2:// -f /home/dev_r/procesos/Poblando_Capa_Landing.sql --hiveconf "PARAM_USERNAME=juan" --hiveconf "ENV=DEV"
-----
-- @section 1. Definición de parámetros
-----
-- [HIVE] Creamos una variable en HIVE
SET ENV=dev;
SET PARAM_USERNAME=dev_r;
-----
-- @section 2. Eliminación de base de datos
-----
-- Eliminación de bases de datos
DROP DATABASE IF EXISTS ${hiveconf:ENV}_landing CASCADE;
-----
-- @section 3. Creación de base de datos
-----
-- Creación de base de datos
CREATE DATABASE IF NOT EXISTS ${hiveconf:ENV}_LANDING LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_LANDING';
-----
-- @section 4. Tunning
-- Compresión
SET hive.exec.compress.output=true;
SET avro.output.codec=snappy;
-- Particionamiento dinámico
SET hive.exec.dynamic.partition=true;
SET hive.exec.dynamic.partition.mode=nonstrict;
-----
-- @section 5. Despliegue de tabla PERSONA
-----
-- Creación de tabla
CREATE TABLE ${hiveconf:ENV}_LANDING.PERSONA
STORED AS AVRO
LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_LANDING/persona'
TBLPROPERTIES(
  'store.charset'=ISO-8859-1',
  'retrieve.charset'=ISO-8859-1',
  'avro.schema.url'=hdfs://user/${hiveconf:PARAM_USERNAME}/datalake/schema/${hiveconf:ENV}_LANDING/persona.avsc',
  'avro.output.codec'=snappy
);
-- Inserción de datos
INSERT INTO TABLE ${hiveconf:ENV}_LANDING.PERSONA
SELECT * FROM ${hiveconf:ENV}_workload.PERSONA;
-- Impresión de datos
SELECT * FROM ${hiveconf:ENV}_LANDING.PERSONA LIMIT 10;
-----
-- @section 6. Despliegue de tabla EMPRESA
-----
-- Creación de tabla
CREATE TABLE ${hiveconf:ENV}_LANDING.EMPRESA
STORED AS AVRO
LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_LANDING/empresa'
TBLPROPERTIES(
  'store.charset'=ISO-8859-1',
  'retrieve.charset'=ISO-8859-1',
  'avro.schema.url'=hdfs://user/${hiveconf:PARAM_USERNAME}/datalake/schema/${hiveconf:ENV}_LANDING/empresa.avsc',
  'avro.output.codec'=snappy
);
-- Inserción de datos
INSERT INTO TABLE ${hiveconf:ENV}_LANDING.EMPRESA
SELECT * FROM ${hiveconf:ENV}_workload.EMPRESA;
-- Impresión de datos
SELECT * FROM ${hiveconf:ENV}_LANDING.EMPRESA LIMIT 10;
-----
```

```

-- @section 7. Despliegue de tabla TRANSACCION
-----
-- Creaci3n de tabla
-- Creaci3n de tabla TRANSACCION
CREATE TABLE ${hiveconf:ENV}_LANDING.TRANSACCION
PARTITIONED BY (FECHA STRING)
STORED AS AVRO
LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_LANDING/transaccion'
TBLPROPERTIES(
  'store.charset'='ISO-8859-1',
  'retrieve.charset'='ISO-8859-1',
  'avro.schema.url'='hdfs:///user/${hiveconf:PARAM_USERNAME}/datalake/schema/${hiveconf:ENV}_LANDING/transaccion.avsc',
  'avro.output.codec'='snappy'
);
-- Inserci3n de datos por particionamiento dinámico
INSERT INTO TABLE ${hiveconf:ENV}_LANDING.TRANSACCION
PARTITION(FECHA)
SELECT * FROM ${hiveconf:ENV}_workload.TRANSACCION;
-- Impresi3n de datos
SELECT * FROM ${hiveconf:ENV}_LANDING.TRANSACCION LIMIT 10;
-- Verificamos las particiones
SHOW PARTITIONS ${hiveconf:ENV}_LANDING.TRANSACCION;

```

La Capa Curated

Cambia el formato de avro a parquet.

La definici3n de la tabla se indica en la creaci3n de la misma a diferencia de la capa anterior (landing) que se cargaba de un archivo avsc.

```

-- Compresi3n
SET hive.exec.compress.output=true;
SET avro.output.codec=snappy;

-- Particionamiento dinámico
SET hive.exec.dynamic.partition=true;
SET hive.exec.dynamic.partition.mode=nonstrict;

-----
--
-- @section 5. Despliegue de tabla PERSONA
--
-----

-- Creaci3n de tabla
CREATE TABLE ${hiveconf:ENV}_LANDING.PERSONA
STORED AS AVRO
LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_LANDING/persona'
TBLPROPERTIES(
  'store.charset'='ISO-8859-1',
  'retrieve.charset'='ISO-8859-1',
  'avro.schema.url'='hdfs:///user/${hiveconf:PARAM_USERNAME}/datalake/schema/${hiveconf:ENV}_LANDING/persona.avsc',
  'avro.output.codec'='snappy'
);

```

```

-- Compresi3n
SET hive.exec.compress.output=true;
SET parquet.compression=SNAPPY;

-- Particionamiento dinámico
SET hive.exec.dynamic.partition=true;
SET hive.exec.dynamic.partition.mode=nonstrict;

-----
--
-- @section 5. Despliegue de tabla PERSONA
--
-----

-- Creaci3n de tabla PERSONA
CREATE TABLE ${hiveconf:ENV}_curated.PERSONA(
  ID STRING,
  NOMBRE STRING,
  TELEFONO STRING,
  CORREO STRING,
  FECHA_INGRESO STRING,
  EDAD INT,
  SALARIO DOUBLE,
  ID_EMPRESA STRING
)
STORED AS PARQUET
LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_curated/PERSONA'

```

Al momento de poblar la capa curated se realizan las validaciones de tipo de dato y de negocio.

```
-- Inserción, casteo de datos y aplicación de reglas de limpieza
```

```
INSERT INTO TABLE ${hiveconf:ENV}_curated.PERSONA
```

```
SELECT
```

```
    CAST(T.ID AS STRING),
    CAST(T.NOMBRE AS STRING),
    CAST(T.TELEFONO AS STRING),
    CAST(T.CORREO AS STRING),
    CAST(T.FECHA_INGRESO AS STRING),
    CAST(T.EDAD AS INT),
    CAST(T.SALARIO AS DOUBLE),
    CAST(T.ID_EMPRESA AS STRING)
FROM
    ${hiveconf:ENV}_LANDING.PERSONA T
WHERE
    T.ID IS NOT NULL AND
    T.ID_EMPRESA IS NOT NULL AND
    CAST(T.EDAD AS INT) > 0 AND
    CAST(T.EDAD AS INT) < 100 AND
    CAST(T.SALARIO AS DOUBLE) > 0 AND
    CAST(T.SALARIO AS DOUBLE) < 10000000;
```

```
-- COMANDO DE EJECUCION
```

```
-- beeline -u jdbc:hive2:// -f /home/dev_r/procesos/Poblando_Capa_Curated.sql --hiveconf "PARAM_USERNAME=dev_r" --hiveconf "ENV=DEV"
```

```
-- @section 1. Definición de parámetros
```

```
-- [HIVE] Creamos una variable en HIVE
```

```
SET ENV=dev;
```

```
SET PARAM_USERNAME=dev_r;
```

```
-- @section 2. Eliminación de base de datos
```

```
-- Eliminación de bases de datos
```

```
DROP DATABASE IF EXISTS ${hiveconf:ENV}_curated CASCADE;
```

```
-- @section 3. Creación de base de datos
```

```
-- Creación de base de datos
```

```
CREATE DATABASE IF NOT EXISTS ${hiveconf:ENV}_curated LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_curated';
```

```
-- @section 4. Tuning
```

```
-- Compresión
```

```
SET hive.exec.compress.output=true;
```

```
SET parquet.compression=SNAPPY;
```

```
-- Particionamiento dinámico
```

```
SET hive.exec.dynamic.partition=true;
```

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

```
-- @section 5. Despliegue de tabla PERSONA
```

```
-- Creación de tabla PERSONA
```

```
CREATE TABLE ${hiveconf:ENV}_curated.PERSONA(
```

```
    ID STRING,
    NOMBRE STRING,
    TELEFONO STRING,
    CORREO STRING,
    FECHA_INGRESO STRING,
    EDAD INT,
    SALARIO DOUBLE,
    ID_EMPRESA STRING
```

```
)
```

```
STORED AS PARQUET
```

```
LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_curated/PERSONA'
```

```
TBLPROPERTIES(
```

```
    'store.charset'='ISO-8859-1',
    'retrieve.charset'='ISO-8859-1',
    'parquet.compression'='SNAPPY'
```

```
);
```

```
-- Inserción, casteo de datos y aplicación de reglas de limpieza
```

```
INSERT INTO TABLE ${hiveconf:ENV}_curated.PERSONA
```

```
SELECT
```

```
    CAST(T.ID AS STRING),
    CAST(T.NOMBRE AS STRING),
    CAST(T.TELEFONO AS STRING),
    CAST(T.CORREO AS STRING),
    CAST(T.FECHA_INGRESO AS STRING),
    CAST(T.EDAD AS INT),
    CAST(T.SALARIO AS DOUBLE),
    CAST(T.ID_EMPRESA AS STRING)
```

```
FROM
```

```
    ${hiveconf:ENV}_LANDING.PERSONA T
```

```

WHERE
  T.ID IS NOT NULL AND
  T.ID_EMPRESA IS NOT NULL AND
  CAST(T.EDAD AS INT) > 0 AND
  CAST(T.EDAD AS INT) < 100 AND
  CAST(T.SALARIO AS DOUBLE) > 0 AND
  CAST(T.SALARIO AS DOUBLE) < 10000000;
-- Impresión de datos
SELECT * FROM ${hiveconf:ENV}_curated.PERSONA LIMIT 10;

-----
-- @section 6. Despliegue de tabla EMPRESA
-----
-- Creación de tabla
CREATE TABLE ${hiveconf:ENV}_curated.EMPRESA(
  ID STRING,
  NOMBRE STRING
)
STORED AS PARQUET
LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_curated/EMPRESA'
TBLPROPERTIES(
  'store.charset'='ISO-8859-1',
  'retrieve.charset'='ISO-8859-1',
  'parquet.compression'='SNAPPY'
);
-- Inserción, casteo de datos y aplicación de reglas de limpieza
INSERT INTO TABLE ${hiveconf:ENV}_curated.EMPRESA
SELECT
  CAST(T.ID AS STRING),
  CAST(T.NOMBRE AS STRING)
FROM
  ${hiveconf:ENV}_LANDING.EMPRESA T
WHERE
  T.ID IS NOT NULL;
-- Impresión de datos
SELECT * FROM ${hiveconf:ENV}_curated.EMPRESA LIMIT 10;

-----
-- @section 7. Despliegue de tabla TRANSACCION
-----
-- Creación de tabla
CREATE TABLE ${hiveconf:ENV}_curated.TRANSACCION(
  ID_PERSONA STRING,
  ID_EMPRESA STRING,
  MONTO DOUBLE
)
PARTITIONED BY (FECHA STRING)
STORED AS PARQUET
LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_CURATED/TRANSACCION'
TBLPROPERTIES(
  'store.charset'='ISO-8859-1',
  'retrieve.charset'='ISO-8859-1',
  'parquet.compression'='SNAPPY'
);
-- Inserción por particionamiento dinámico, casteo de datos y aplicación de reglas de limpieza
INSERT INTO TABLE ${hiveconf:ENV}_curated.TRANSACCION
PARTITION(FECHA)
SELECT
  CAST(T.ID_PERSONA AS STRING),
  CAST(T.ID_EMPRESA AS STRING),
  CAST(T.MONTO AS DOUBLE),
  CAST(T.FECHA AS STRING)
FROM
  ${hiveconf:ENV}_LANDING.TRANSACCION T
WHERE
  T.ID_PERSONA IS NOT NULL AND
  T.ID_EMPRESA IS NOT NULL AND
  CAST(T.MONTO AS DOUBLE) >= 0;
-- Impresión de datos
SELECT * FROM ${hiveconf:ENV}_curated.TRANSACCION LIMIT 10;
-- Verificamos las particiones
SHOW PARTITIONS ${hiveconf:ENV}_curated.TRANSACCION;

```

Capa Funcional

Las tablas temporales solo existen durante la sesión. Cada vez que nos conectamos al Beeline se genera un id de sesión y al desconectarse acaba la sesión.

Se van creando tablas enriquecidas, primero persona con transacción, luego esa tabla que sale de la unión de ambas con la tabla empresa.

```

-- COMANDO DE EJECUCION

-- beeline -u jdbc:hive2://-f/home/dev_r/procesos/Poblando_Capa_Funcional.sql --hiveconf "PARAM_USERNAME=dev_r" --hiveconf "ENV=DEV"

-- @section 1. Definición de parámetros

-- [HIVE] Creamos una variable en HIVE

SET ENV=dev;

SET PARAM_USERNAME=dev_r;

```



```

-- @section 2. Eliminaci3n de base de datos

-- Eliminaci3n de bases de datos

DROP DATABASE IF EXISTS ${hiveconf:ENV}_FUNCTIONAL CASCADE;

--

-- @section 3. Creaci3n de base de datos

-- Creaci3n de base de datos

CREATE DATABASE IF NOT EXISTS ${hiveconf:ENV}_FUNCTIONAL LOCATION '/user/${hiveconf:ENV}/datalake/${hiveconf:ENV}_FUNCTIONAL';

-- COMANDO DE EJECUCION

-- beeline -u jdbc:hive2:// -f /home/userbda000/Laboratorio_022_proceso_tunning_codigo_y_tuning_recursos.sql --hiveconf "ENV=userbda000"

-- @section 1. Definici3n de par3metros

-- [HIVE] Creamos una variable en HIVE

-- SET ENV=userbda000;

-- @section 2. Tunning

--

-- Compresi3n

SET hive.exec.compress.output=true;

SET parquet.compression=SNAPPY;

-- Particionamiento din3mico

SET hive.exec.dynamic.partition=true;

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

SET hive.exec.max.dynamic.partitions=9999;

SET hive.exec.max.dynamic.partitions.pernode=9999;

-- Selecci3n del motor de ejecuci3n

SET hive.execution.engine=mr;

-- SET hive.execution.engine=spark;

-- SET hive.execution.engine=tez;

-- Tunning de recursos computacionales [mr]

SET mapreduce.job.maps=8;

SET mapreduce.map.cpu.vcores=2;

SET mapreduce.map.memory.mb=1024;

SET mapreduce.job.reduces=8;

SET mapreduce.reduce.cpu.vcores=2;

SET mapreduce.reduce.memory.mb=1024;

SET mapreduce.input.fileinputformat.split.maxsize = 1024000000;

SET mapreduce.input.fileinputformat.split.minsize = 1024000000;

-- Tunning de recursos computacionales [spark]

-- SET spark.driver.memory=1g;

-- SET spark.dynamicAllocation.maxExecutors=8;

-- SET spark.executor.cores=2;

-- SET spark.executor.memory=1g;

--SET spark.executor.memoryOverhead=100m;

-- Tunning de recursos computacionales [tez]

-- set mapred.reduce.tasks = -1;

-----

-- @section 3. Deploy de tablas temporales

--

```

```

-- Creaci3n de tabla

CREATE TABLE ${hiveconf:ENV}_FUNCTIONAL.TRANSAccion_ENRIQUECIDA(
  ID_PERSONA INT,
  NOMBRE_PERSONA STRING,
  EDAD_PERSONA INT,
  SALARIO_PERSONA DOUBLE,
  TRABAJO_PERSONA STRING,
  MONTO_TRANSAccion DOUBLE,
  EMPRESA_TRANSAccion STRING
)

PARTITIONED BY (FECHA_TRANSAccion STRING)

STORED AS PARQUET

LOCATION '/user/${hiveconf:PARAM_USERNAME}/datalake/${hiveconf:ENV}_FUNCTIONAL/TRANSAccion_ENRIQUECIDA'

TBLPROPERTIES(

  'store.charset'='ISO-8859-1',

  'retrieve.charset'='ISO-8859-1',

  'parquet.compression'='SNAPPY'
);

-- Eliminamos la tabla temporal

DROP TABLE IF EXISTS ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSAccion_ENRIQUECIDA_1;

-- Creamos la tabla temporal

CREATE TEMPORARY TABLE ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSAccion_ENRIQUECIDA_1(
  ID_PERSONA STRING,
  NOMBRE_PERSONA STRING,
  EDAD_PERSONA INT,
  SALARIO_PERSONA DOUBLE,
  ID_EMPRESA_PERSONA STRING,
  MONTO_TRANSAccion DOUBLE,
  FECHA_TRANSAccion STRING,
  ID_EMPRESA_TRANSAccion STRING
)

STORED AS PARQUET;

-- Eliminamos la tabla temporal

DROP TABLE IF EXISTS ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSAccion_ENRIQUECIDA_2;

-- Creamos la tabla temporal

CREATE TEMPORARY TABLE ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSAccion_ENRIQUECIDA_2(
  ID_PERSONA STRING,
  NOMBRE_PERSONA STRING,
  EDAD_PERSONA INT,
  SALARIO_PERSONA DOUBLE,
  TRABAJO_PERSONA STRING,
  MONTO_TRANSAccion DOUBLE,
  FECHA_TRANSAccion STRING,
  ID_EMPRESA_TRANSAccion STRING
)

STORED AS PARQUET;

-- Eliminamos la tabla temporal

DROP TABLE IF EXISTS ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSAccion_ENRIQUECIDA_3;

-- Creamos la tabla temporal

CREATE TEMPORARY TABLE ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSAccion_ENRIQUECIDA_3(
  ID_PERSONA STRING,
  NOMBRE_PERSONA STRING,
  EDAD_PERSONA INT,
  SALARIO_PERSONA DOUBLE,
  TRABAJO_PERSONA STRING,
  MONTO_TRANSAccion DOUBLE,
  FECHA_TRANSAccion STRING,
  EMPRESA_TRANSAccion STRING
)

STORED AS PARQUET;

@section 4. Proceso
-----

-- Truncamos la tabla

TRUNCATE TABLE ${hiveconf:ENV}_FUNCTIONAL.TRANSAccion_ENRIQUECIDA;

```

```

-- PASO 1: OBTENER LOS DATOS DE LA PERSONA QUE REALIZÓ LA TRANSACCIÓN

INSERT INTO ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_1

SELECT
    T.ID_PERSONA,
    P.NOMBRE,
    P.EDAD,
    P.SALARIO,
    P.ID_EMPRESA,
    T.MONTO,
    T.FECHA,
    T.ID_EMPRESA

FROM

    ${hiveconf:ENV}_curated.TRANSACCION T

    JOIN ${hiveconf:ENV}_curated.PERSONA P

    ON T.ID_PERSONA = P.ID;

-- Verificamos

SELECT * FROM ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_1 LIMIT 10;

-- PASO 2: OBTENER EL NOMBRE DE LA EMPRESA EN DONDE TRABAJA LA PERSONA

INSERT INTO ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_2

SELECT
    T.ID_PERSONA,
    T.NOMBRE_PERSONA,
    T.EDAD_PERSONA,
    T.SALARIO_PERSONA,
    E.NOMBRE,
    T.MONTO_TRANSACCION,
    T.FECHA_TRANSACCION,
    T.ID_EMPRESA_TRANSACCION

FROM

    ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_1 T

    JOIN ${hiveconf:ENV}_curated.EMPRESA E

    ON T.ID_EMPRESA_PERSONA = E.ID;

-- Verificamos

SELECT * FROM ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_2 LIMIT 10;

-- PASO 3: OBTENER EL NOMBRE DE LA EMPRESA EN DONDE SE REALIZÓ LA TRANSACCIÓN

INSERT INTO ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_3

SELECT
    T.ID_PERSONA,
    T.NOMBRE_PERSONA,
    T.EDAD_PERSONA,
    T.SALARIO_PERSONA,
    T.TRABAJO_PERSONA,
    T.MONTO_TRANSACCION,
    T.FECHA_TRANSACCION,
    E.NOMBRE

FROM

    ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_2 T

    JOIN ${hiveconf:ENV}_curated.EMPRESA E

    ON T.ID_EMPRESA_TRANSACCION = E.ID;

-- Verificamos

SELECT * FROM ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_3 LIMIT 10;

-- PASO 4: INSERTAR EN LA TABLA RESULTANTE FINAL

INSERT OVERWRITE TABLE ${hiveconf:ENV}_FUNCTIONAL.TRANSACCION_ENRIQUECIDA

PARTITION (FECHA_TRANSACCION)

SELECT
    T.ID_PERSONA,
    T.NOMBRE_PERSONA,
    T.EDAD_PERSONA,
    T.SALARIO_PERSONA,
    T.TRABAJO_PERSONA,
    T.MONTO_TRANSACCION,
    T.EMPRESA_TRANSACCION,
    T.FECHA_TRANSACCION

FROM

    ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_3 T;

-- Verificamos

SELECT * FROM ${hiveconf:ENV}_FUNCTIONAL.TRANSACCION_ENRIQUECIDA LIMIT 10;

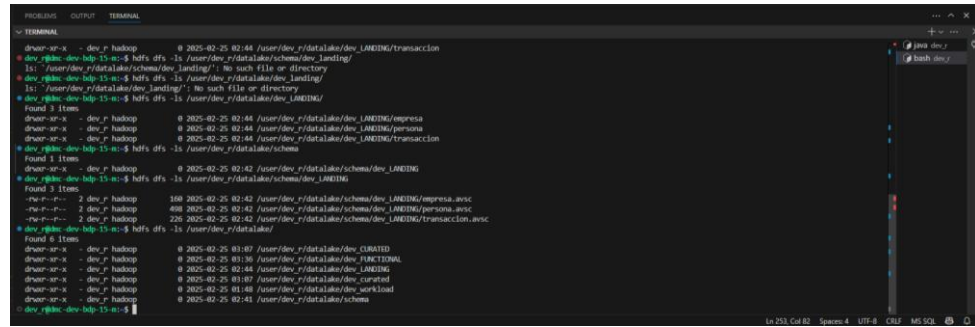
```

```
-- @section 4. Eliminaci3n de tablas temporales
```

```
-- DROP TABLE IF EXISTS ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_1;
```

```
-- DROP TABLE IF EXISTS ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_2;
```

```
-- DROP TABLE IF EXISTS ${hiveconf:ENV}_FUNCTIONAL.TMP_TRANSACCION_ENRIQUECIDA_3;
```



```
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/dev_LANDING/transaction
0 2025-02-25 02:44 /user/dev_r/datalake/dev_LANDING/transaction
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/schema/dev_LANDING/
ls: /user/dev_r/datalake/schema/dev_LANDING/: no such file or directory
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/dev_LANDING/
ls: /user/dev_r/datalake/dev_LANDING/: no such file or directory
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/dev_LANDING/
Found 3 items
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/dev_LANDING/empresa
0 2025-02-25 02:44 /user/dev_r/datalake/dev_LANDING/empresa
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/dev_LANDING/persona
0 2025-02-25 02:44 /user/dev_r/datalake/dev_LANDING/persona
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/dev_LANDING/transaction
0 2025-02-25 02:44 /user/dev_r/datalake/dev_LANDING/transaction
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/schema
Found 1 items
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/schema/dev_LANDING
0 2025-02-25 02:42 /user/dev_r/datalake/schema/dev_LANDING
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/schema/dev_LANDING
Found 3 items
-rw-r--r-- 2 dev_r hadoop 160 2025-02-25 02:42 /user/dev_r/datalake/schema/dev_LANDING/empresa.asc
-rw-r--r-- 2 dev_r hadoop 400 2025-02-25 02:42 /user/dev_r/datalake/schema/dev_LANDING/persona.asc
-rw-r--r-- 2 dev_r hadoop 220 2025-02-25 02:42 /user/dev_r/datalake/schema/dev_LANDING/transaction.asc
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/
Found 6 items
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/dev_CURATED
0 2025-02-25 03:30 /user/dev_r/datalake/dev_CURATED
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/dev_FUNCTIONAL
0 2025-02-25 03:30 /user/dev_r/datalake/dev_FUNCTIONAL
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/dev_LANDING
0 2025-02-25 02:44 /user/dev_r/datalake/dev_LANDING
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/dev_CURATED
0 2025-02-25 03:30 /user/dev_r/datalake/dev_CURATED
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/dev_sarkload
0 2025-02-25 03:30 /user/dev_r/datalake/dev_sarkload
dev_r@dev-hdp-15 ~$ hadoop fs -ls /user/dev_r/datalake/schema
0 2025-02-25 02:41 /user/dev_r/datalake/schema
dev_r@dev-hdp-15 ~$
```