

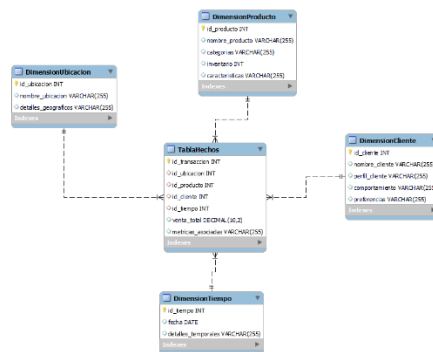
INFORME DE TRABAJO GRUPAL

MATERIA	Sistemas Avanzados de bases de datos	NRC	15031
CARRERA	Ing. de Software	Docente	Ing. Alexis Estevez
PERIODO ACADÉMICO	Noviembre 2023 – marzo 2024	FECHA	23 – 11 - 2023
TEMA	Consulta		
ESTUDIANTE(S)	Joel Arguello		

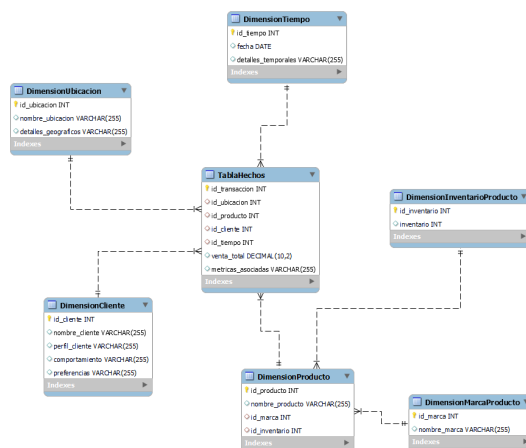
Los códigos se subirán a un repositorio para cualquier revisión

Modelos

Estrella:



Copo de nieve:



INFORME DE TRABAJO GRUPAL**SQL**

Al ser largos los archivos se incluirán en el repositorio

Crear red

Estrella:

- Create network copo

Copo de nieve

- Create newtork Estrella

Crear contenedor

Estrella:

```
docker run --name estrella-mysql-master -p 3306:3306 -e  
MYSQL_ROOT_PASSWORD=root -e MYSQL_ALLOW_EMPTY_PASSWORD=yes -d -  
-cpus=1 --memory=700m --storage-opt size=1g mysql:latest --default-authentication-  
plugin=mysql_native_password
```

```
docker network connect estrella estrella-mysql-master
```

```
docker run --name estrella-mysql-slave -p 3307:3306 -e  
MYSQL_ROOT_PASSWORD=root -e MYSQL_ALLOW_EMPTY_PASSWORD=yes -d -  
-cpus=0.5 --memory=700m --storage-opt size=1g mysql:latest --default-authentication-  
plugin=mysql_native_password --require_secure_transport=OFF
```

```
docker network connect etrella etrella-mysql-slave
```

Copo de nieve:

```
docker run --name copo-mysql-master -p 3308:3306 -e  
MYSQL_ROOT_PASSWORD=root -e MYSQL_ALLOW_EMPTY_PASSWORD=yes -d -  
-cpus=0.5 --memory=700m --storage-opt size=1g mysql:latest --default-authentication-  
plugin=mysql_native_password --require_secure_transport=OFF
```

```
docker network connect copo copo-mysql-master
```

```
docker run --name copo-mysql-slave2 -p 3309:3306 -e  
MYSQL_ROOT_PASSWORD=root -e MYSQL_ALLOW_EMPTY_PASSWORD=yes -d -  
-cpus=0.5 --memory=700m --storage-opt size=1g mysql:latest --default-authentication-  
plugin=mysql_native_password --require_secure_transport=OF
```

```
docker network connect copo copo-mysql-slave
```

INFORME DE TRABAJO GRUPAL**Comandos SQL**

Los comandos se deben ejecutar en el orden que se encuentra en el documento

Primer paso

Master

```
SET @global.server_id = 1;
```

Slave

```
SET @global.server_id = 2;
```

Segundo paso:

En el Master

```
DROP USER IF EXISTS 'repl'@'%';
```

```
CREATE USER 'repl'@ '%' IDENTIFIED WITH 'mysql_native_password' BY 'repl';
```

```
GRANT REPLICATION SLAVE ON *.* TO 'repl'@ '%';
```

```
FLUSH PRIVILEGES;
```

```
FLUSH TABLES WITH READ LOCK;
```

```
SHOW MASTER STATUS;
```

Obtenemos la información que tenemos en master status

Tercer paso

Obtenemos la IP del nodo master

```
Docker inspect ____-mysql-master
```

En el Slave

```
CHANGE MASTER TO MASTER_HOST='ipdelhost', MASTER_USER='repl',  
MASTER_PASSWORD='repl', MASTER_LOG_FILE='Se obtiene del paso anterior',  
MASTER_LOG_POS=Se-obtiene del paso anterior;
```

```
START SLAVE;
```

Esto lo repetimos según las redes o modelos

Cuarto paso

INFORME DE TRABAJO GRUPAL

Master:

Unlock tables;

Exclavo:

Verificamos el estado:

SHOW SLAVE STATUS;

Imágenes

```
C:\Users\Joel>docker network create estrella  
37c2f6d523dbe384a4edff92ca587d7e03b0f1f3fed22813db12bf42669483a0
```

```
C:\Users\Joel>docker network create copo  
17e83a42777d7d370254135225493bf66815cae62997ee3bfb7d2b939d6f80f7
```

```
C:\Users\Joel>docker run --name estrella-mysql-master -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root -e MYSQL_ALLOW_EMPTY_PASSWORD=yes -d mysql:latest --default-authentication-plugin=mysql_native_password  
f65db506fb860c8b9daa16f58bae4c0c1ce2fa61f9237cab210ceedb09857acb
```

```
C:\Users\Joel>docker run --name estrella-mysql-slave -p 3307:3306 -e MYSQL_ROOT_PASSWORD=root -e MYSQL_ALLOW_EMPTY_PASSWORD=yes -d mysql:latest --default-authentication-plugin=mysql_native_password --require_secure_transport=OFF  
d1cbff6679852dee110b1e2c9313db0b32fd6b1033545cecf4c6c68cc073adb2
```

```
C:\Users\Joel>docker network connect estrella estrella-mysql-master
```

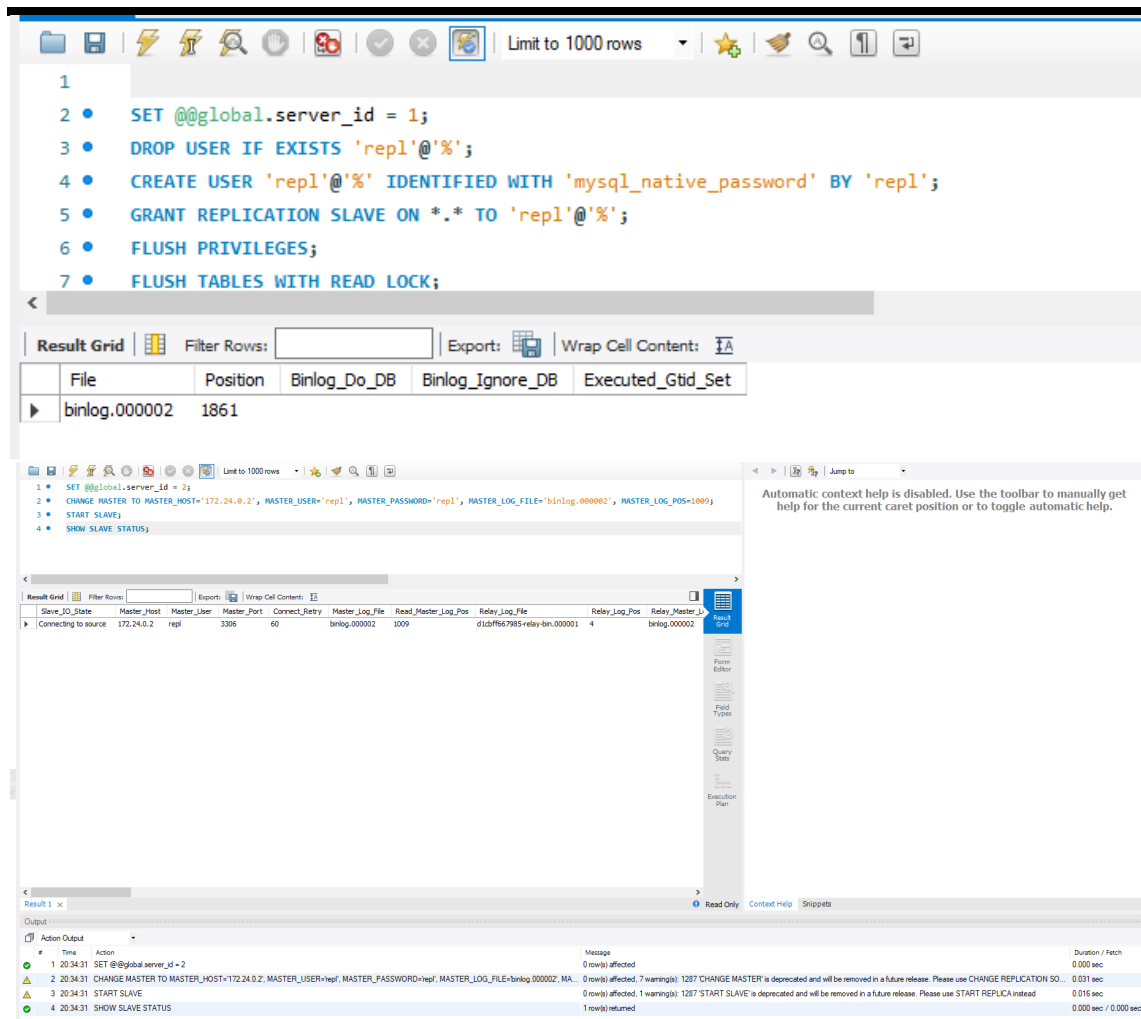
```
C:\Users\Joel>docker network connect estrella estrella-mysql-slave
```

```
Simbolo del sistema
```

```
{  
  "NetworkID": "3ca0c8ab29094695901bc21fba24ab9b44929f49aa89d63013912ddc7603fbc",  
  "EndpointID": "35ddb7b6c2c2dd4957ef4b674f3d2017bf98f6f5fe6414a1af5890e104d35291",  
  "Gateway": "172.17.0.1",  
  "IPAddress": "172.17.0.2",  
  "IPPrefixLen": 16,  
  "IPv6Gateway": "",  
  "GlobalIPv6Address": "",  
  "GlobalIPv6PrefixLen": 0,  
  "MacAddress": "02:42:ac:11:00:02",  
  "DriverOpts": null  
},  
  "estrella": {  
    "IPAMConfig": {},  
    "Links": null,  
    "Aliases": [  
      "f65db506fb86"  
    ],  
    "NetworkID": "37c2f6d523dbe384a4edff92ca587d7e03b0f1f3fed22813db12bf42669483a0",  
    "EndpointID": "5eeaceb58c9d1b5ff235891bdf2a72632c9e8612923e0840ee6d79ca3087cb8",  
    "Gateway": "172.26.0.1",  
    "IPAddress": "172.26.0.2",  
    "IPPrefixLen": 16,  
    "IPv6Gateway": "",  
    "GlobalIPv6Address": "",  
    "GlobalIPv6PrefixLen": 0,  
    "MacAddress": "02:42:ac:1a:00:02",  
    "DriverOpts": {}  
  }  
}
```

```
C:\Users\Joel>
```

INFORME DE TRABAJO GRUPAL



The screenshot displays the MySQL Workbench interface. The top pane shows a series of SQL commands for setting up a replication slave:

```

1
2 • SET @@global.server_id = 1;
3 • DROP USER IF EXISTS 'repl'@'%';
4 • CREATE USER 'repl'@'%' IDENTIFIED WITH 'mysql_native_password' BY 'repl';
5 • GRANT REPLICATION SLAVE ON *.* TO 'repl'@'%';
6 • FLUSH PRIVILEGES;
7 • FLUSH TABLES WITH READ LOCK;
  
```

The middle pane shows the 'Result Grid' with a single row of data:

File	Position	Binlog_Do_DB	Binlog_Ignore_DB	Executed_Gtid_Set
binlog.000002	1861			

The bottom pane shows the 'Output' tab with the following actions and messages:

#	Time	Action	Message	Duration / Fetch
1	20:34:31	SET @@global.server_id = 2	0 row(s) affected	0.000 sec
2	20:34:31	CHANGE MASTER TO MASTER_HOST='172.24.0.2', MASTER_USER='repl', MASTER_PASSWORD='repl', MASTER_LOG_FILE='binlog.000002', MA...	0 row(s) affected, 7 warning(s): 1287 CHANGE MASTER is deprecated and will be removed in a future release. Please use CHANGE REPLICATION SO...	0.031 sec
3	20:34:31	START SLAVE	0 row(s) affected, 1 warning(s): 1287 START SLAVE is deprecated and will be removed in a future release. Please use START REPLICA instead	0.016 sec
4	20:34:31	SHOW SLAVE STATUS	1 row(s) returned	0.000 sec / 0.000 sec

Se tuvo coneccion exitosa

Inserts

Los inserts se los realizo con Python usando la extensión Faker

INFORME DE TRABAJO GRUPAL

```
estrella > ◆ insert.py > ...
1 import mysql.connector
2 from faker import Faker
3 from datetime import datetime, timedelta
4
5 conexion = mysql.connector.connect(
6     host="localhost",
7     user="root",
8     password="root",
9     port=3306,
10    database="uso"
11 )
12
13 cursor = conexion.cursor()
14
15 fake = Faker('es_ES')
16
17 num_registros = 20
18
19 for i in range(1, num_registros + 1):
20     nombre_ubicacion = fake.city()
21     detalles_geograficos = fake.address()
22
23     consulta_insert = "INSERT INTO DimensionUbicacion (id_ubicacion, nombre_ubicacion, detalles_geograficos) VALUES (%s, %s, %s);"
```

```
24
25     valores = (i, nombre_ubicacion, detalles_geograficos)
26
27     cursor.execute(consulta_insert, valores)
28
29
30
31 for i in range(1, num_registros + 1):
32     nombre_producto = fake.word()
33     categorias = fake.word()
34     inventario = fake.random_int(1, 100)
35     caracteristicas = fake.text()
36
37     consulta_insert = "INSERT INTO DimensionProducto (id_producto, nombre_producto, categorias, inventario, caracteristicas) VALUES (%s, %s, %s, %s, %s);"
```

```
38
39     valores = (i, nombre_producto, categorias, inventario, caracteristicas)
40
41     cursor.execute(consulta_insert, valores)
42
43
44 for i in range(1, num_registros + 1):
45     nombre_cliente = fake.name()
46     perfil_cliente = fake.word()
47     comportamiento = fake.word()
48     preferencias = fake.text()
49
50     consulta_insert = "INSERT INTO DimensionCliente (id_cliente, nombre_cliente, perfil_cliente, comportamiento, preferencias) VALUES (%s, %s, %s, %s, %s);"
```

Ln. 22, col. 1 Espacios: 4 UTF-8 CRLF Python 3.11.7 64-bit (Microsoft Store)

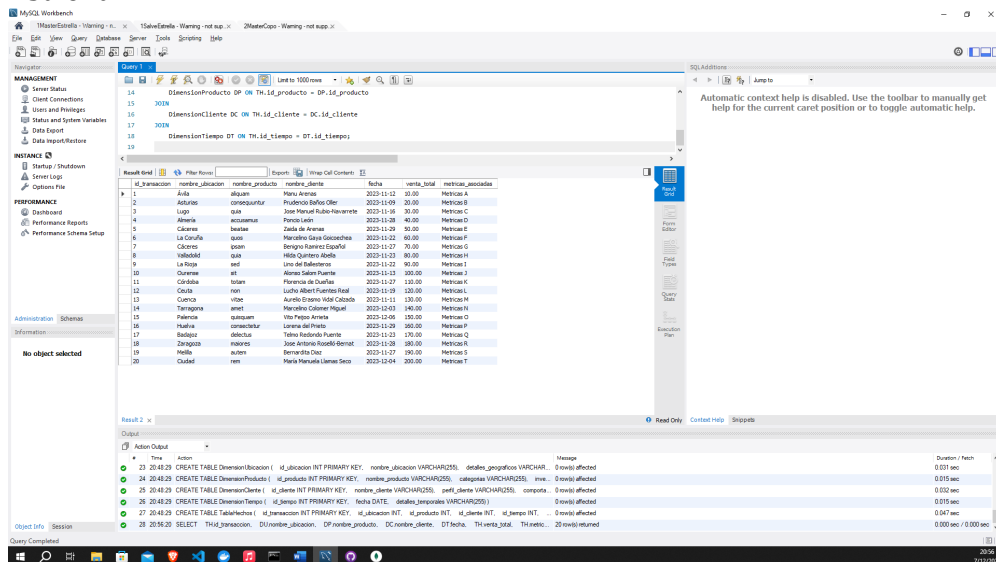
```
estrella > ◆ mongo.py > ...
1 import pymongo
2 import mysql.connector
3 from faker import Faker
4 from datetime import datetime, timedelta
5 from decimal import Decimal
6
7 mysql_connection = mysql.connector.connect(
8     host="localhost",
9     user="root",
10    password="secret",
11    port=3306,
12    database="uso"
13 )
14 mysql_cursor = mysql_connection.cursor(dictionary=True)
15
16 mongo_client = pymongo.MongoClient("mongodb://localhost:27017/")
17 mongo_db = mongo_client["uso"]
18
19 fake = Faker('es_ES')
20
21 mysql_cursor.execute("SELECT * FROM DimensionUbicacion")
22 ubicaciones = mysql_cursor.fetchall()
23 ubicacion_collection = mongo_db["DimensionUbicacion"]
24 ubicacion_collection.insert_many(ubicaciones)
25
26 mysql_cursor.execute("SELECT * FROM DimensionProducto")
27 productos = mysql_cursor.fetchall()
28 producto_collection = mongo_db["DimensionProducto"]
29 producto_collection.insert_many(productos)
30
31 mysql_cursor.execute("SELECT * FROM DimensionCliente")
32 clientes = mysql_cursor.fetchall()
33 cliente_collection = mongo_db["DimensionCliente"]
34 cliente_collection.insert_many(clientes)
35
36 mysql_cursor.execute("SELECT * FROM DimensionTiempo")
37 tiempos = mysql_cursor.fetchall()
38 tiempo_collection = mongo_db["DimensionTiempo"]
39 for tiempo in tiempos:
40     tiempo['fecha'] = datetime(tiempo['fecha'].year, tiempo['fecha'].month, tiempo['fecha'].day)
41     tiempo_collection.insert_many(tiempos)
42
43 mysql_cursor.execute("SELECT * FROM TablaHechos")
44 hechos = mysql_cursor.fetchall()
45 hechos_collection = mongo_db["TablaHechos"]
46 for hecho in hechos:
47     hecho['venta_total'] = float(hecho['venta_total'])
48     hechos_collection.insert_many(hechos)
49
```

INFORME DE TRABAJO GRUPAL



Vista

Los SQL para las vistas se incluyen en el repositorio Estrella:



Copo:

