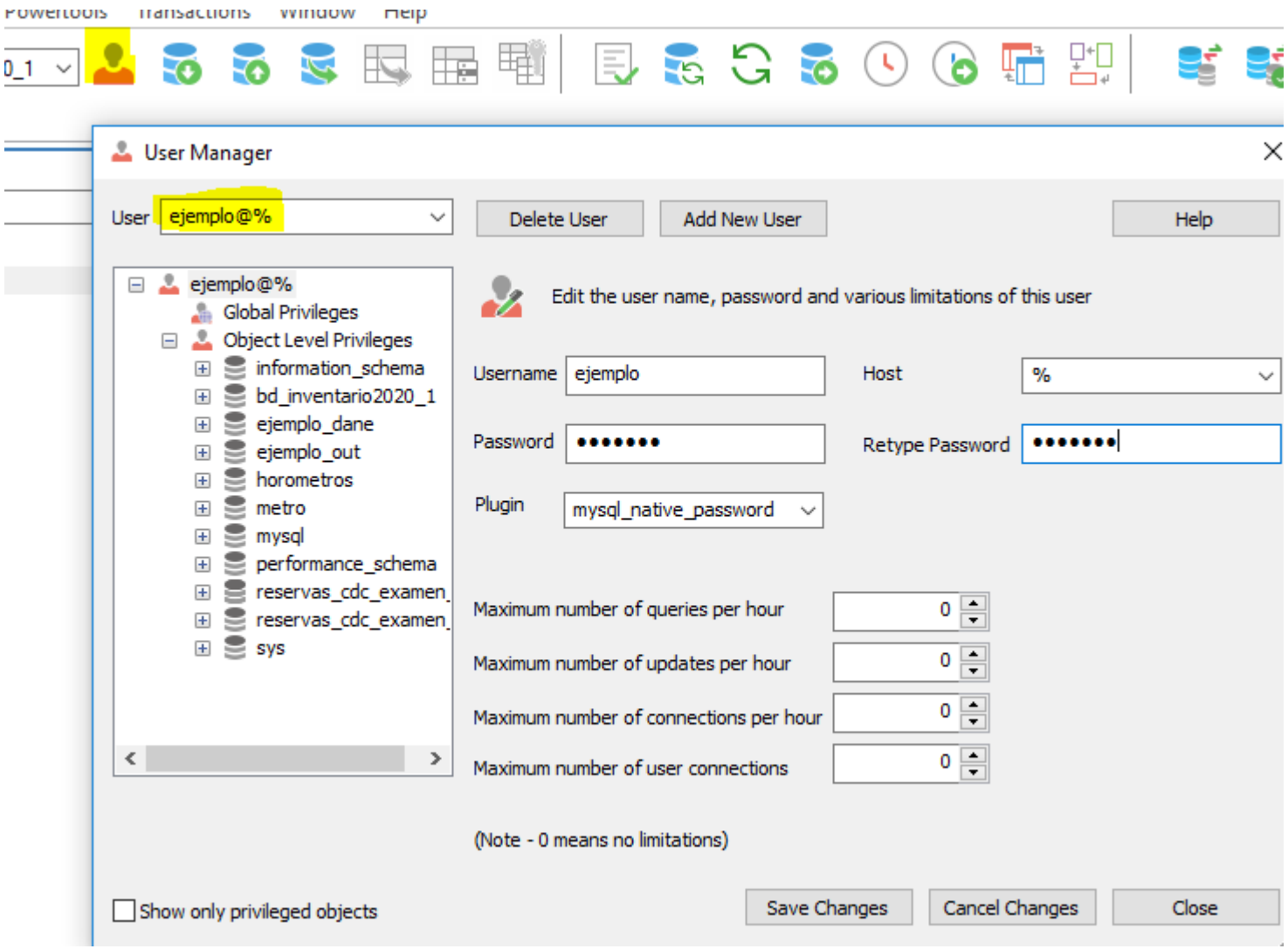
PERMISOS Y CANDADOS MYSQL

SERGIO ÁLVAREZ

VERSIÓN 1.0

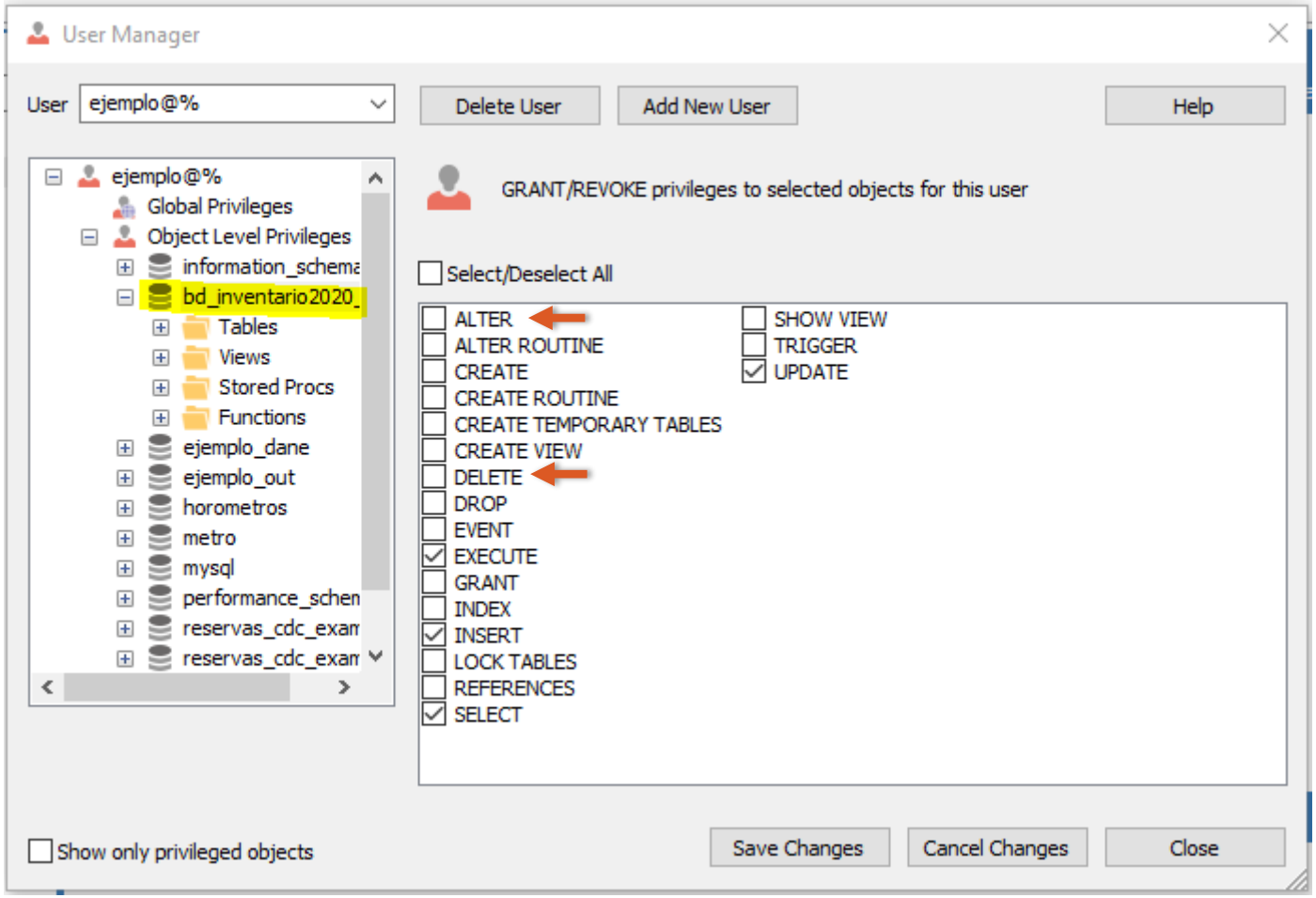
PERMISOS

CREAR USUARIO



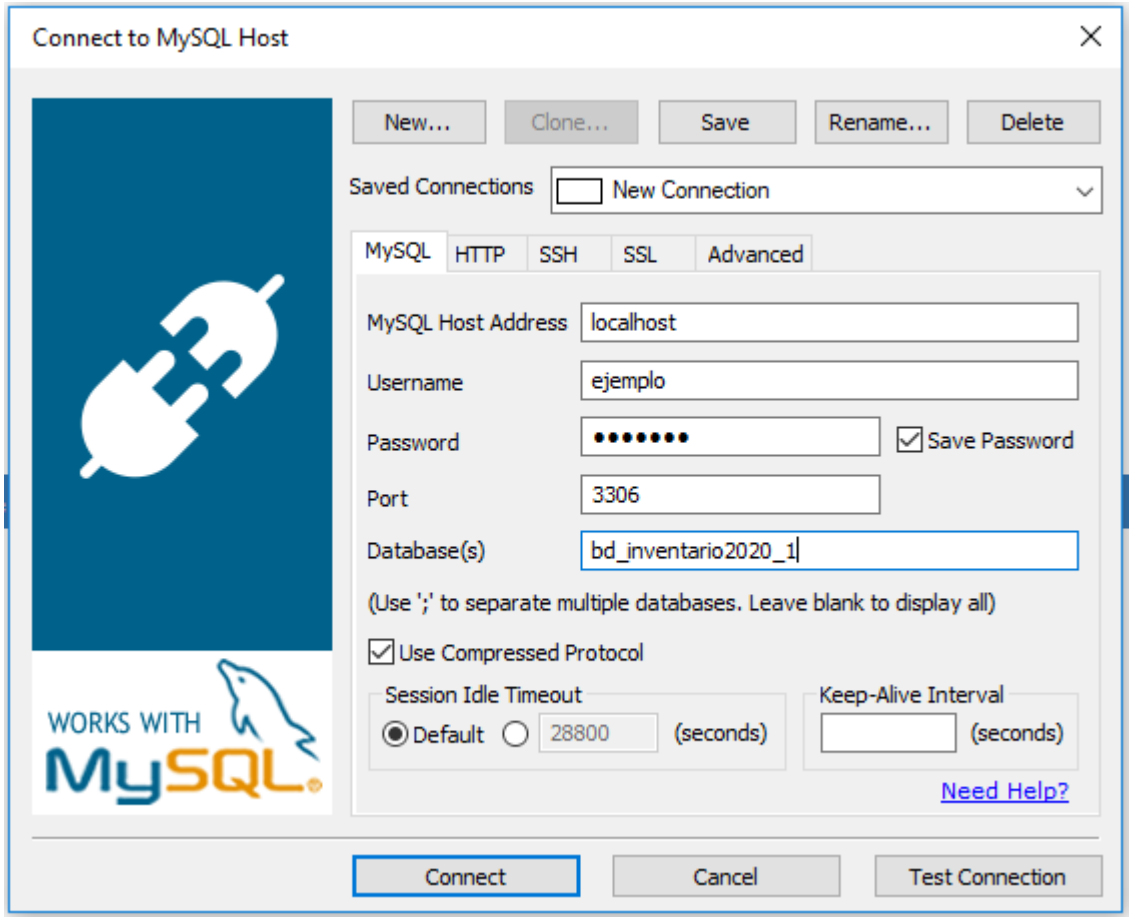
Sergio Álvarez

PERMISOS X BASE DE DATOS

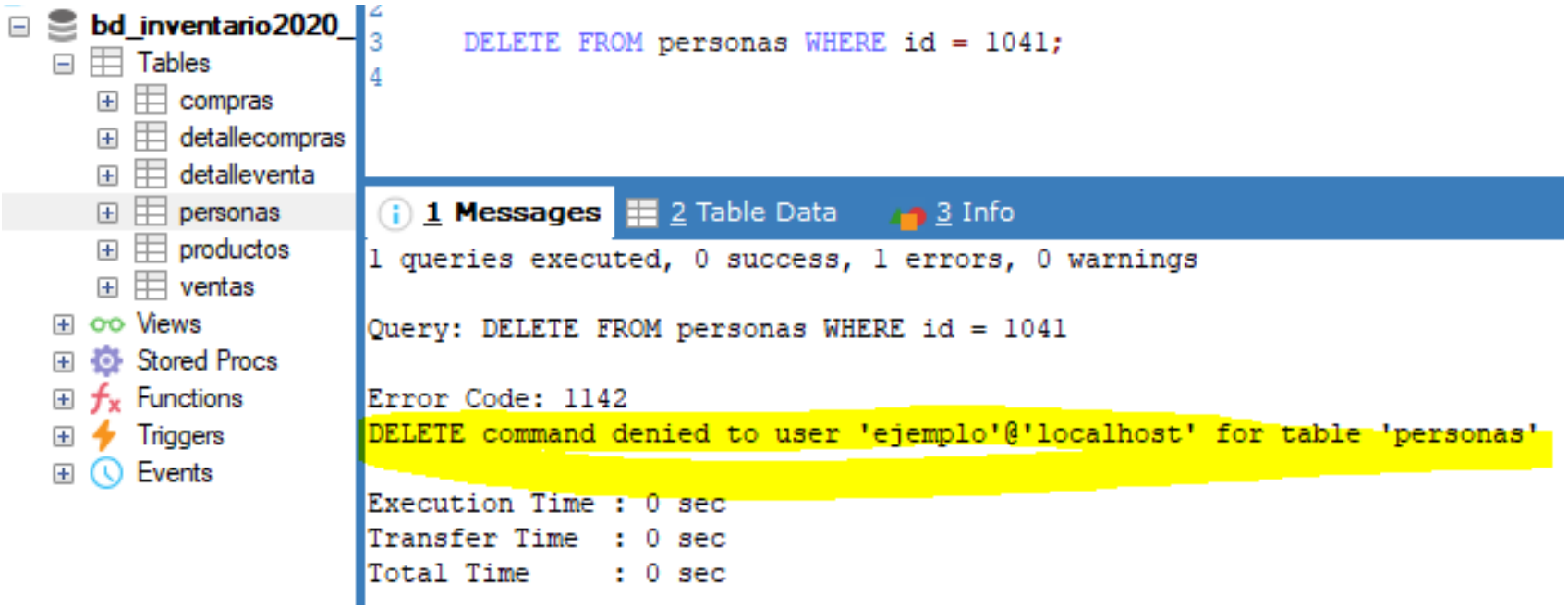


Sergio Álvarez

OTRA INSTANCIA SQLYOG Y NUEVA CONEXIÓN

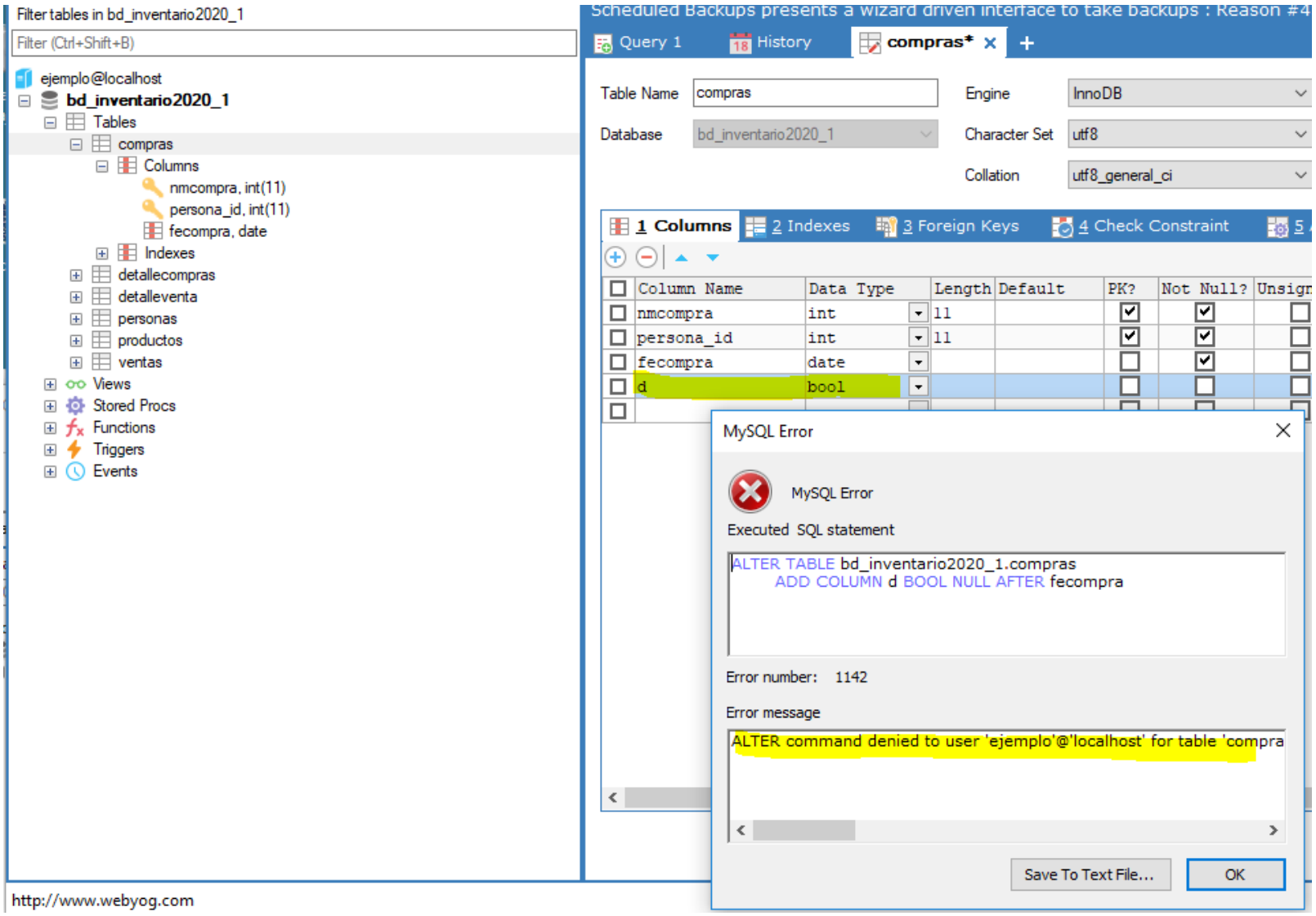


SIN PERMISO PARA BORRAR



Sergio Álvarez

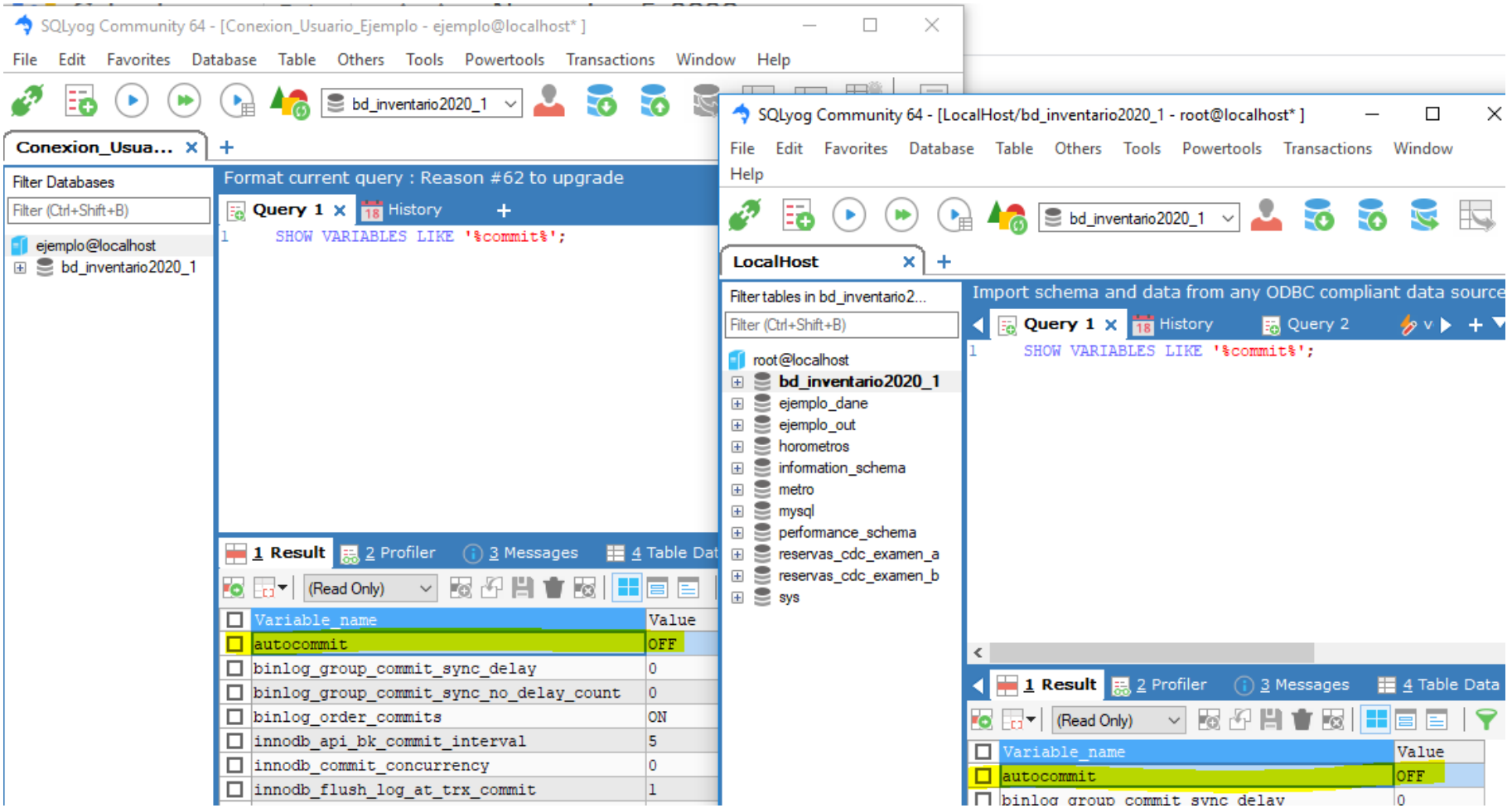
PERMISOS MODIFICAR



Sergio Álvarez

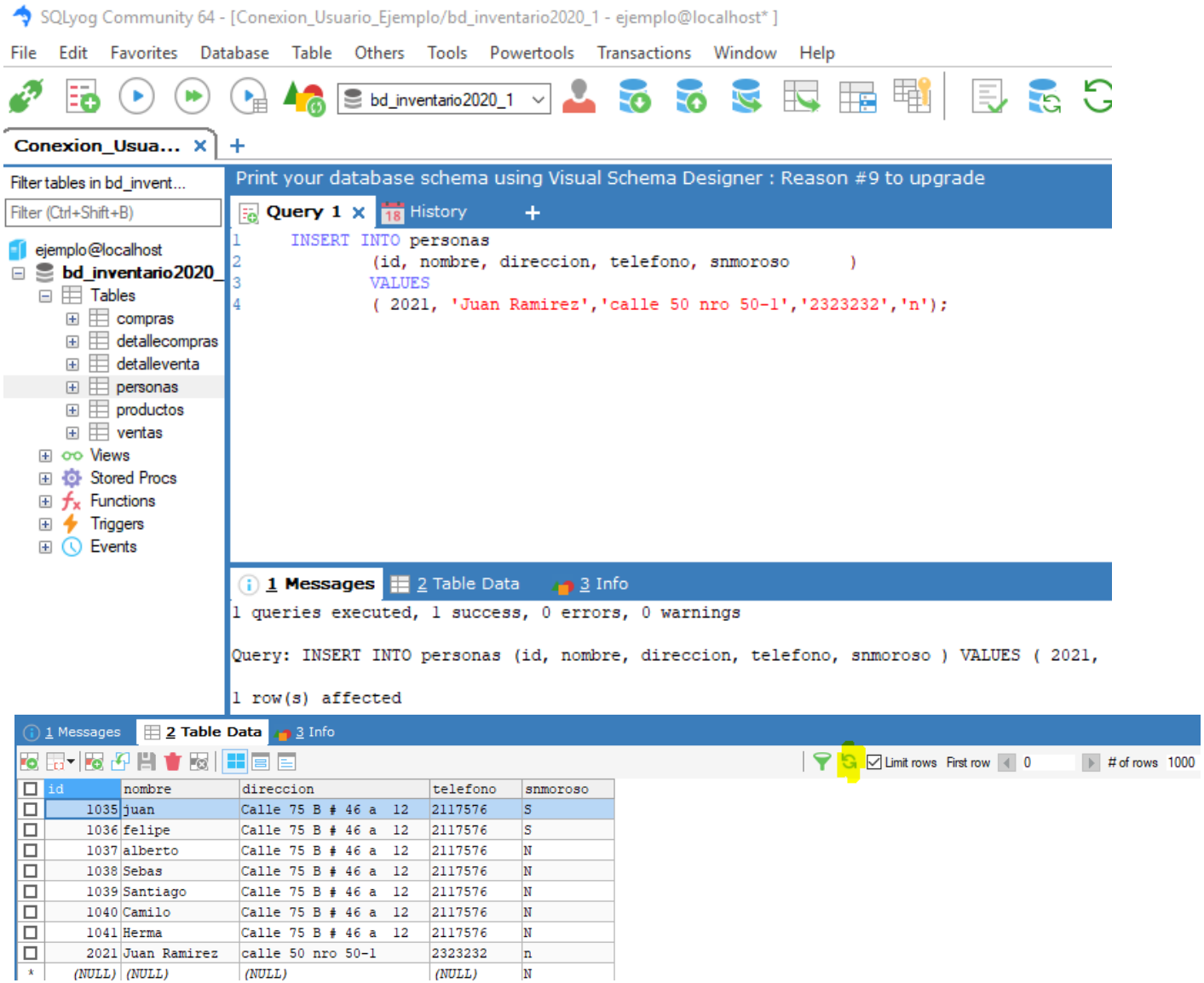
PRUEBA DE CONCURRENCIA

VALIDAR AUTOCOMMIT



Sergio Álvarez

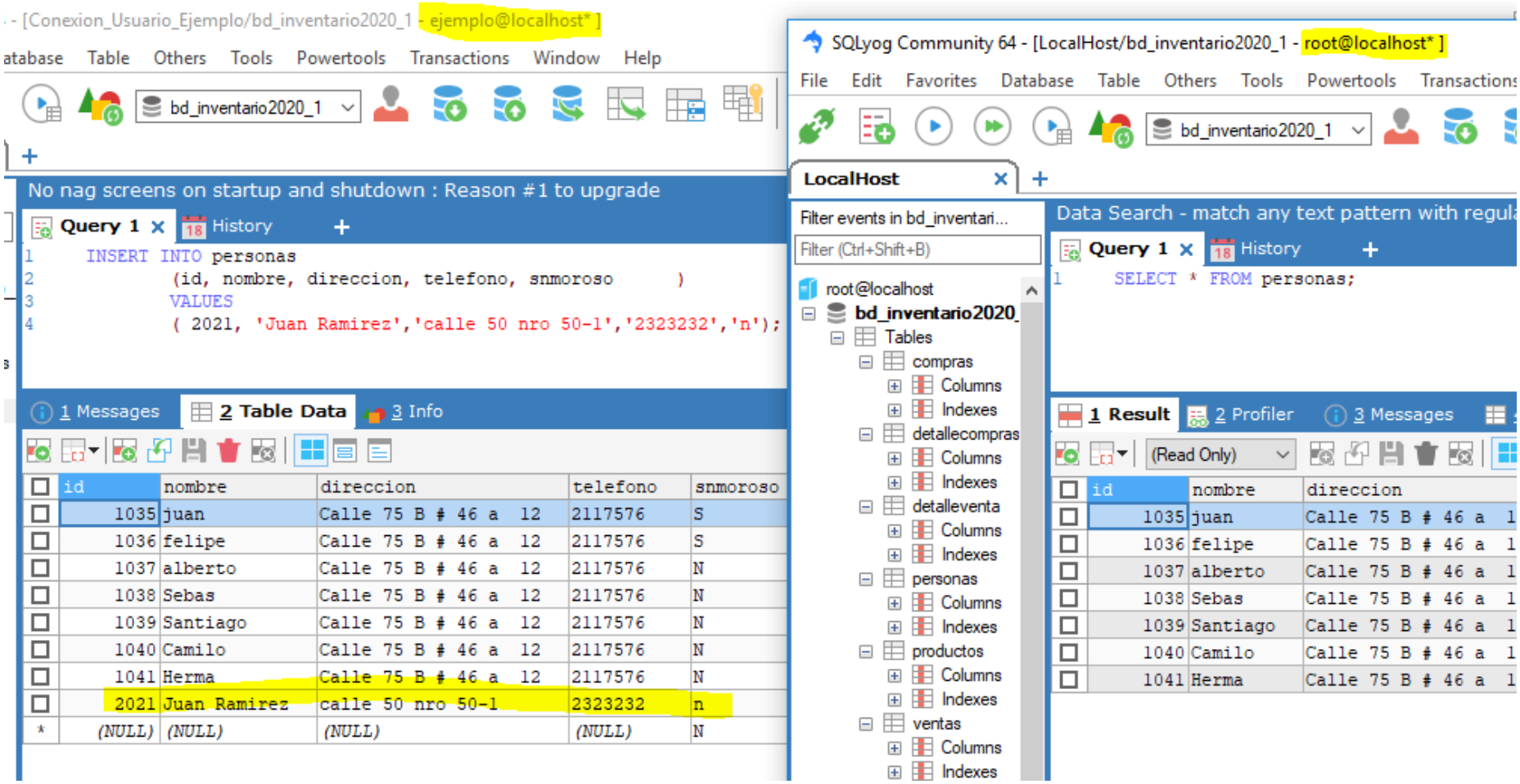
SIN COMMIT – USUARIO EJEMPLO



id	nombre	direccion	telefono	snmoroso
1035	juan	Calle 75 B # 46 a 12	2117576	S
1036	felipe	Calle 75 B # 46 a 12	2117576	S
1037	alberto	Calle 75 B # 46 a 12	2117576	N
1038	Sebas	Calle 75 B # 46 a 12	2117576	N
1039	Santiago	Calle 75 B # 46 a 12	2117576	N
1040	Camilo	Calle 75 B # 46 a 12	2117576	N
1041	Herma	Calle 75 B # 46 a 12	2117576	N
2021	Juan Ramirez	calle 50 nro 50-1	2323232	n
(NULL)	*(NULL)*	*(NULL)*	*(NULL)*	N

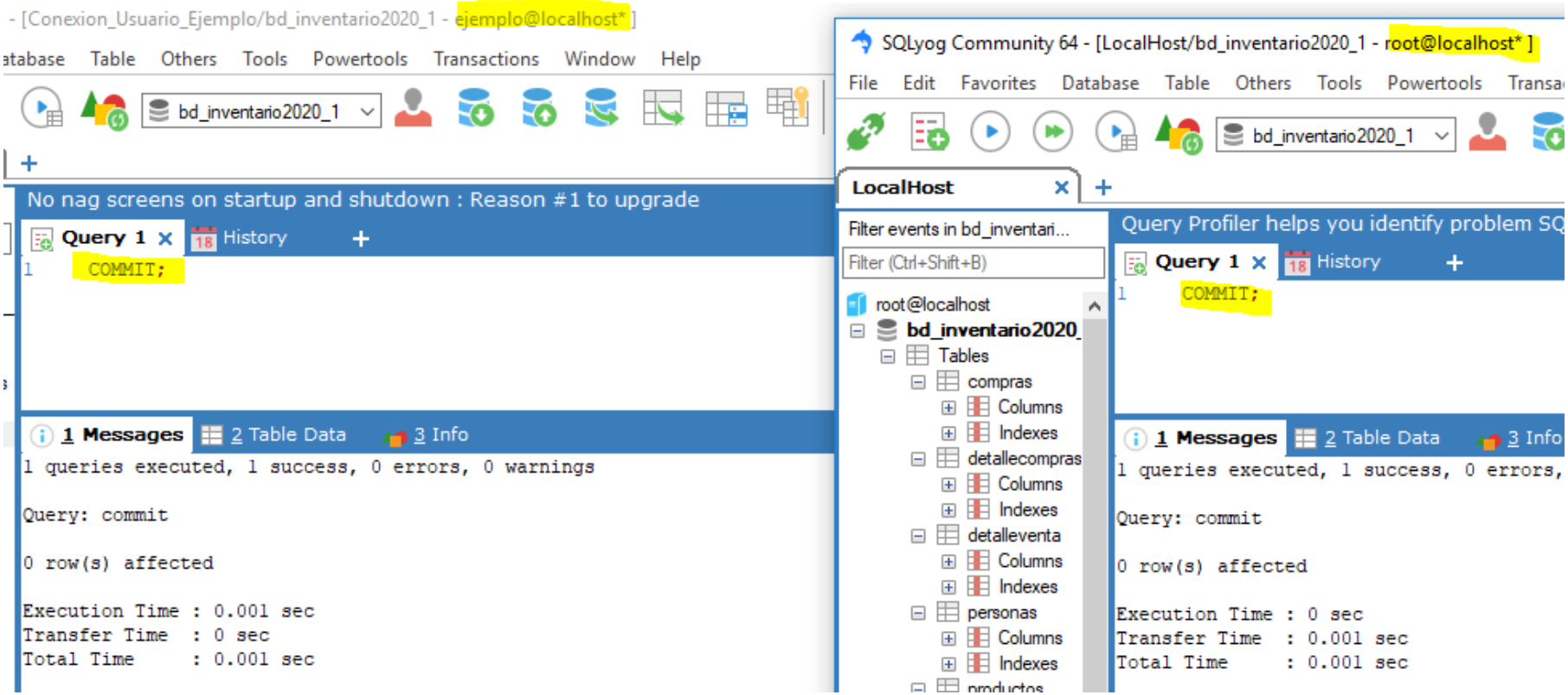
Sergio Alvarez

SIN COMMIT – USUARIO EJEMPLO



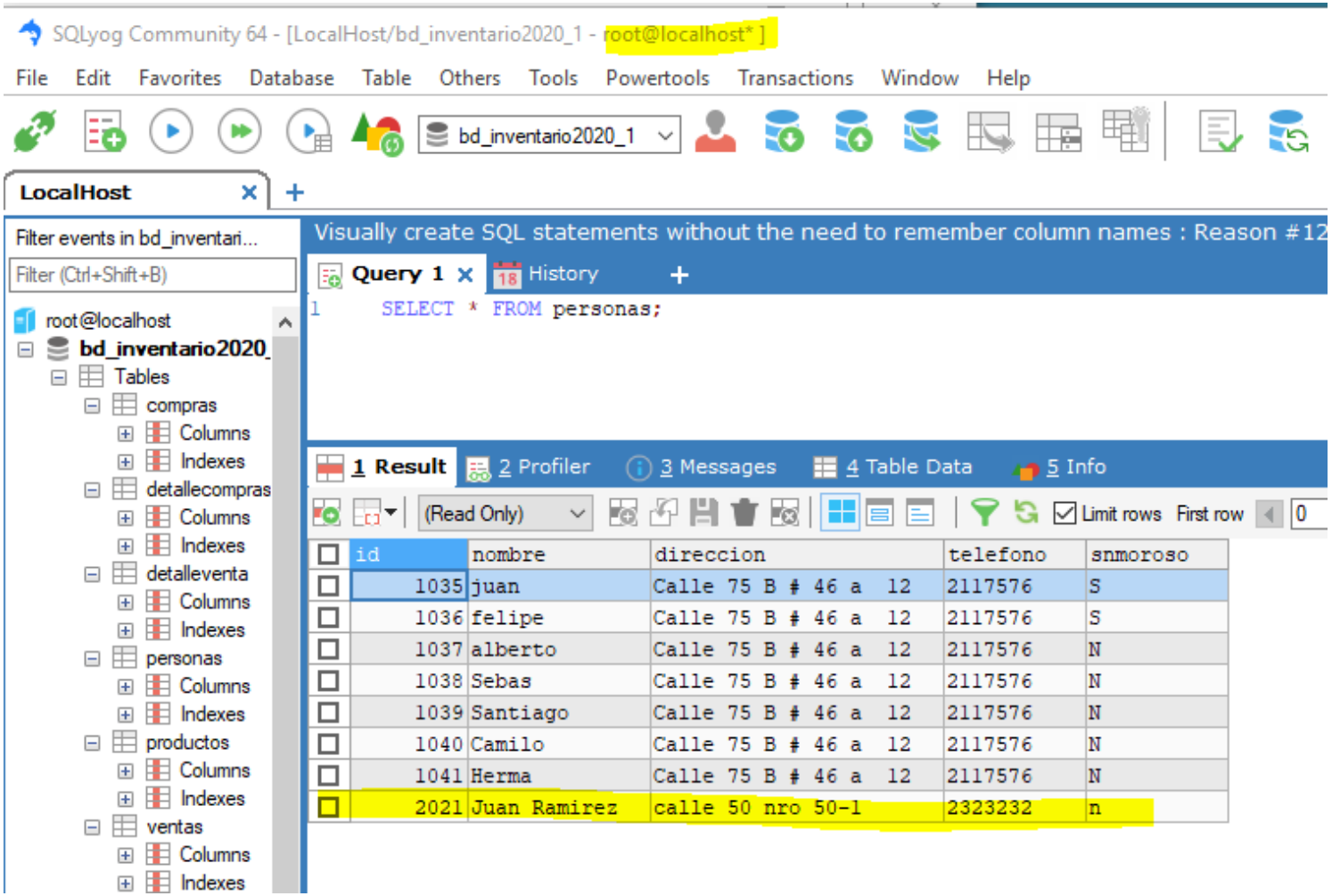
Sergio Alvarez

COMMIT EN AMBOS



Sergio Alvarez

SE REFRESCAN LOS CAMBIOS EN ROOT



SINTAXIS CORRECTA

```
START TRANSACTION;
INSERT INTO personas
    (id, nombre, direccion, telefono, snmoroso  )
    VALUES
    ( 2022, 'Diana Ochoa','calle 1 sur nro 10-1','454545','N');
COMMIT ;
```


Sergio Alvarez

CANDADOS

FORMA MANUAL, SIN TRANSACCIONES

```
LOCK TABLES
    tbl_name [[AS] alias] lock_type
    [, tbl_name [[AS] alias] lock_type] ...

lock_type: {
    READ [LOCAL]
  | [LOW_PRIORITY] WRITE
}

UNLOCK TABLES
```

MySQL enables client sessions to acquire table locks explicitly for the purpose of cooperating with other sessions for access to tables, or to prevent other sessions from modifying tables during periods when a session requires exclusive access to them. A session can acquire or release locks only for itself. One session cannot acquire locks for another session or release locks held by another session.

Locks may be used to emulate transactions or to get more speed when updating tables. This is explained in more detail in Table-Locking Restrictions and Conditions.

Sergio Alvarez

LOCKING READS

- `SELECT ... LOCK IN SHARE MODE`

  Sets a shared mode lock on any rows that are read. Other sessions can read the rows, but cannot modify them until your transaction commits. If any of these rows were changed by another transaction that has not yet committed, your query waits until that transaction ends and then uses the latest values.

- `SELECT ... FOR UPDATE`

  For index records the search encounters, locks the rows and any associated index entries, the same as if you issued an `UPDATE` statement for those rows. Other transactions are blocked from updating those rows, from doing `SELECT ... LOCK IN SHARE MODE`, or from reading the data in certain transaction isolation levels. Consistent reads ignore any locks set on the records that exist in the read view. (Old versions of a record cannot be locked; they are reconstructed by applying undo logs on an in-memory copy of the record.)

```
1  SELECT * FROM parent WHERE NAME = 'Jones' LOCK IN SHARE MODE;
```

https://dev.mysql.com/doc/refman/5.6/en/innodb-locking-reads.html

Sergio Alvarez