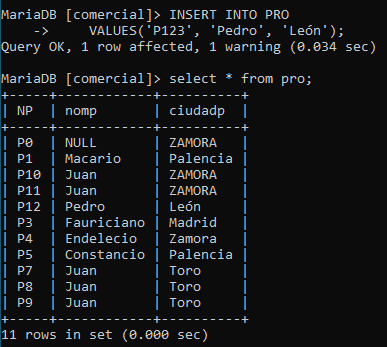
INSERCCIÓN DE VALORES EN UNA TABLA (INSERT)

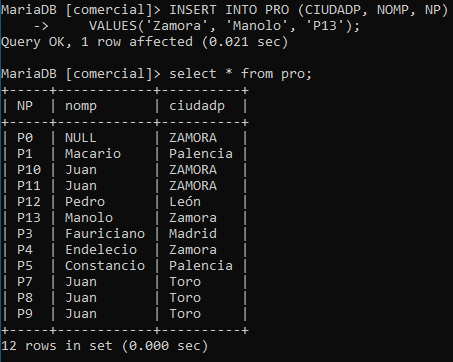
INSERT [IGNORE] [INTO][SET] nombre\_Tabla [(columnas)] VALUES (VALORES |DEFAULT |NULL) [ON DUPLICATE KEY UPDATE columna = expresión, …]

Cosas a tener en cuenta a la hora de insertar valores:

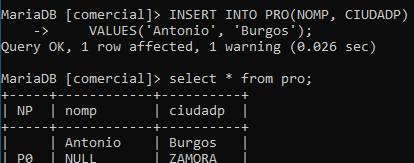
* La cláusula INTO es obligatoria en algunos SGBD.
* A la hora de insertar valores con más caracteres del limitado, se corta el valor insertado a los caracteres límite.



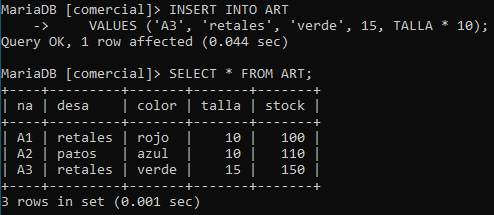
* Podemos insertar los valores en distinto orden poniendo los nombres de las columnas en el orden que queramos. También se pueden poner solo ciertas columnas, dejando las demás columnas a su valor por defecto, “”, 0 o NULL dependiendo de la situación.



* Si un campo tiene la restricción de no nulo, pero su valor por defecto es nulo, el quedará como cadena vacía.



* Es posible hacer que el contenido de un campo sea el resultado de una operación de otro de los campos.



* Con la cláusula IGNORE se desprecian los errores y se intenta continuar con la sentencia.

INSERT INTO TIN

    VALUES  ('T00', 'ABC', 2),

            ('T01', 'ABCAS', 4),

            ('T02', 'ABC', 3);

INSERT IGNORE INTO TIN

    VALUES  ('T00', 'AC', 1),

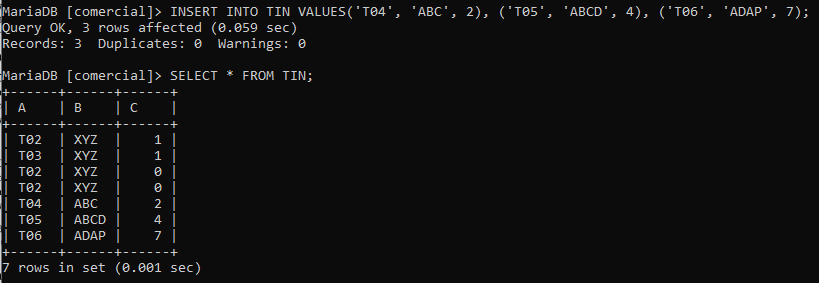
            ('T03', 'ABCS', 6),

            ('T02', 'XYZ', 3);

A la hora de ejecutar estas líneas de código, la cláusula IGNORE nos dará un aviso de que los valores ‘T00’ y ‘T02’ están repetidos, pero insertará sin problemas ‘T03’.

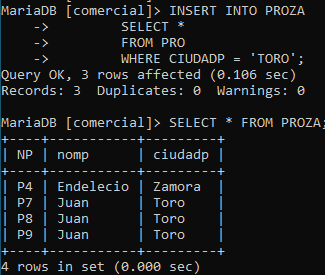
INSERTAR VARIOS VALORES EN UNA SOLA SENTENCIA

INSERT INTO <TABLA> VALUES (VALORES), (VALORES2)…



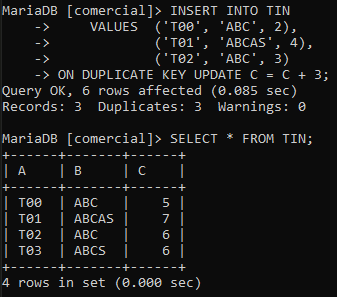
INSERTAR VALORES DE UNA CONSULTA

Podemos conseguir que el resultado propuesto de una consulta se aplique a los valores que queremos modificar de alguna tabla.



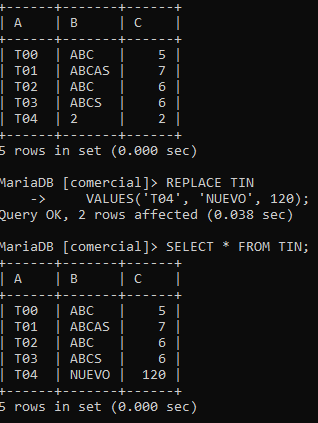
ON DUPLICATE KEY UPDATE

Esta cláusula hace que en caso de que se repita el valor de alguna clave primaria, se ejecutará la expresión puesta a continuación.



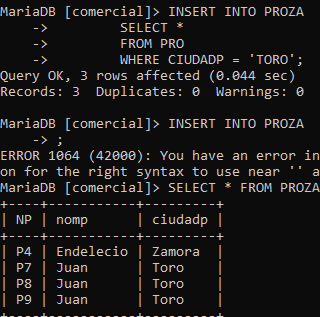
REPLACE

Hace lo mismo que INSERT, pero en el caso de que ya exista el valor, este se borrará y se insertará el nuevo a continuación.



INSERTAR EL RESULTADO DE UNA CONSULTA

Incorporar a una tabla el resultado de una consulta.



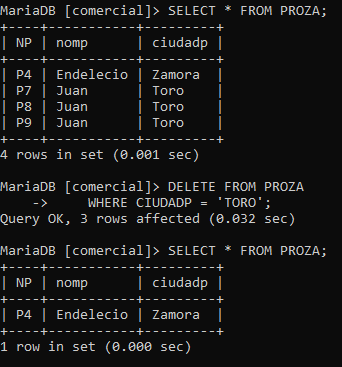
BORRAR VALORES DE UNA TABLA(DELETE)

DELETE [IGNORE] FROM <TABLA>

[CONDICIÓN WHERE …]

[ORDER BY …]

[LIMIT *FILAS*];



Para asegurarnos de que borramos lo correcto, es recomendable hacer antes una consulta y verificar los datos que vayamos a eliminar.

ORDER BY y LIMIT

Las cláusulas ORDER BY y LIMIT tienen sentido si se usan juntas, ya que con esto se consiguen eliminar x registros desde un orden:

-- BORRAR EL PRIMERO ORDENADO POR NOMBRE --

-- ANTES --

SELECT \* FROM PRO;

DELETE FROM PRO

    ORDER BY NOMP DESC

    LIMIT 1;

-- DESPUÉS --

SELECT \* FROM PRO;

MODIFICIACIÓN DE VALORES DE UNA TABLA(UPDATE)

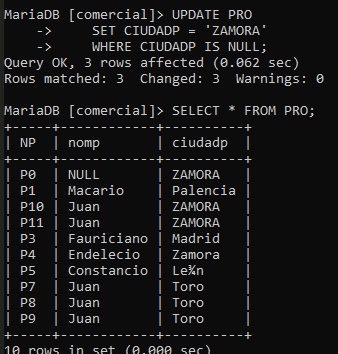
UPDATE [IGNORE] <TABLA>

SET *COLUMNA = EXPRESION …*

[CONDICIÓN WHERE]

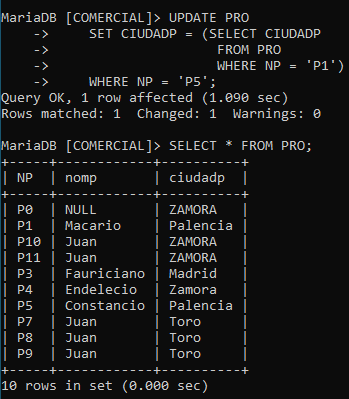
[ORDER BY]

[LIMIT *FILAS*]



MODIFICACIÓN DE VALORES POR CONSULTA

Se pueden modificar valores con el resultado de una consulta al igual que con casos anteriores.



MODIFICACIÓN/BORRADO SOBRE DISTINTAS TABLAS

**DELETE FROM** PRO, FAB **USING** PRO, PED, FAB

**WHERE** PRO.NP = PED.NP **AND** FAB.NF = PED.NF **AND** PRO.NP = ‘P4’;

“Borra las filas de los proveedores P4 que se encuentren en el conjunto de las tablas PRO, PED y FAB, pero que solo afecte a las tablas PRO y FAB”

SET AUTOCOMMIT = 0;

SELECT \* FROM PED;

SELECT \* FROM PRO;

SELECT \* FROM FAB;

DELETE FROM PRO, FAB USING PRO, PED, FAB

    WHERE PRO.NP = PED.NP AND FAB.NF = PED.NF AND PRO.NP = 'P4';

SELECT \* FROM PED;

SELECT \* FROM PRO;

SELECT \* FROM FAB;

ROLLBACK;

REVERTIR CAMBIOS

Usando AUTOCOMMIT = 0 podremos hacer que cuando escribamos ROLLBACK al final, se revertirán los cambios realizados en las tablas.