

# TAREA UT05 – BASES DE DATOS

ANTONIO SANZ PANS  
IES MAESTRE DE CALATRAVA

## Contenido

Primeros pasos .....	3
Ejercicio 1 .....	4
Enunciado.....	4
Capturas de pantalla .....	4
Sentencia SQL.....	6
Ejercicio 2 .....	7
Enunciado.....	7
Capturas de pantalla .....	7
Sentencia SQL.....	8
Ejercicio 3 .....	9
Enunciado.....	9
Capturas de pantalla .....	9
Sentencia SQL.....	9
Ejercicio 4 .....	10
Enunciado.....	10
Respuesta .....	10
Capturas de pantalla .....	10
Sentencias SQL .....	11
Ejercicio 5 .....	12
Enunciado.....	12
Capturas de pantalla .....	12
Sentencia SQL.....	12
Ejercicio 6 .....	13
Enunciado.....	13
Capturas de pantalla .....	13
Sentencia SQL.....	13
Ejercicio 7 .....	14
Enunciado.....	14
Capturas de pantalla .....	14
Sentencia SQL.....	15
Ejercicio 8 .....	17
Enunciado.....	17
Capturas de pantalla .....	17
Sentencia SQL.....	17



## PRIMEROS PASOS

Crear el usuario y la conexión para la práctica BD05

Nueva / Seleccionar Conexión a Base de Datos

Nombre de Cone... Detalles de Cone...  
PRACTICA-BD05 BD05@//localhos...  
PRACTICA04-AN... ANTONIO@//loca...  
XE-SYSTEM SYSTEM@//localh...  
XEpdb1-ANTONIO ANTONIO@//loca...

Nombre: PRACTICA-BD05

Tipo de Base de Datos: Oracle

Información de usuario: Usuario de Proxy

Tipo de autenticación: Por defecto

Usuario: BD05 Rol: valor por defecto

Contraseña: \*\*\*\* ☒ Guardar Contraseña

Tipo de Conexión: Básico

Detalles: Avanzado

Nombre del Host: localhost

Puerto: 1521

☐ SID

☒ Nombre del Servicio: XEpdb1

Estado: Correcto

Ayuda Guardar Borrar Probar Conectar Cancelar

Cargar las tablas del archivo Tarea-BD05-TABLAS.sql

Oracle SQL Developer - D:\Curso\DAW\BASES DE DATOS\UT05\Tarea para BD05-2022-23\Tarea-BD05-TABLAS.sql

Archivo Editar Ver Ejecutar Origen Equip Herramientas Ventana Ayuda

Conexiones: PRACTICA-BD05

Hoja de Trabajo de SQL: Historial

Hoja de Trabajo: Generador de Consultas

ALTER SESSION SET NLS\_DATE\_FORMAT='DD/MM/YYYY';

DROP TABLE "M\_FAMILIAS" cascade constraints;

DROP TABLE "M\_LABORATORIOS" cascade constraints;

DROP TABLE "M\_MEDICAMENTOS" cascade constraints;

DROP TABLE "M\_MEDICOS" cascade constraints;

DROP TABLE "M\_PACIENTES" cascade constraints;

DROP TABLE "M\_PRESCRIPCIONES" cascade constraints;

DROP TABLE "M\_VENTAS\_MED" cascade constraints;

DROP TABLE "M\_VENTAS\_RECETAS" cascade constraints;

CREATE TABLE "M\_FAMILIAS"

Table "M\_MEDICAMENTOS" alterado.

Table "M\_MEDICAMENTOS" alterado.

Table "M\_VENTAS\_MED" alterado.

Table "M\_VENTAS\_RECETAS" alterado.

Table "M\_VENTAS\_RECETAS" alterado.

Table "M\_VENTAS\_RECETAS" alterado.

Table "M\_VENTAS\_RECETAS" alterado.

Tarea terminada en 1.664 segundos

## EJERCICIO 1

### Enunciado

Utiliza la herramienta gráfica de SQL Developer para insertar en la tabla **M\_VENTAS\_MED** las siguientes ventas:

ID_VENTA	ID_MED	Fecha Venta	Unidades
25	9	12/01/2021	4
26	10	21/01/2021	5
27	11	3/02/2021	10
28	12	7/02/2021	1

Utiliza la herramienta gráfica de SQL Developer para insertar en la tabla **M\_VENTAS\_RECETAS** los registros cuyo ID sea 25 y 28 a las ventas anteriores, pero sin asignarle médicos ni pacientes.

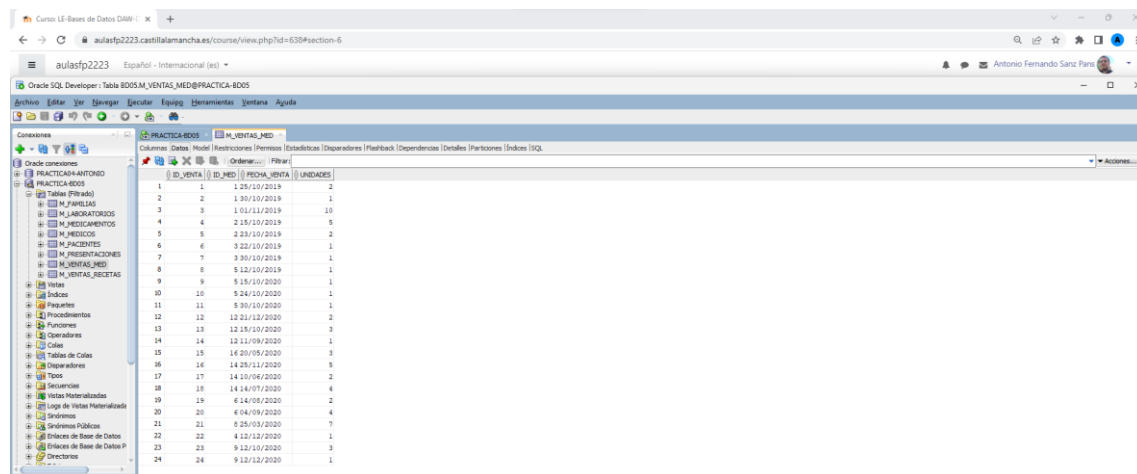
Mediante sentencias SQL: Modifica los registros de la tabla **M\_VENTAS\_RECETAS** para asignar a cada venta introducida anteriormente médico y un paciente. El profesorado que debes asignar a cada curso es:

ID_VENTA	DNI Médico	DNI Paciente
25	22222222B	10000000A
28	11111111A	20000000B

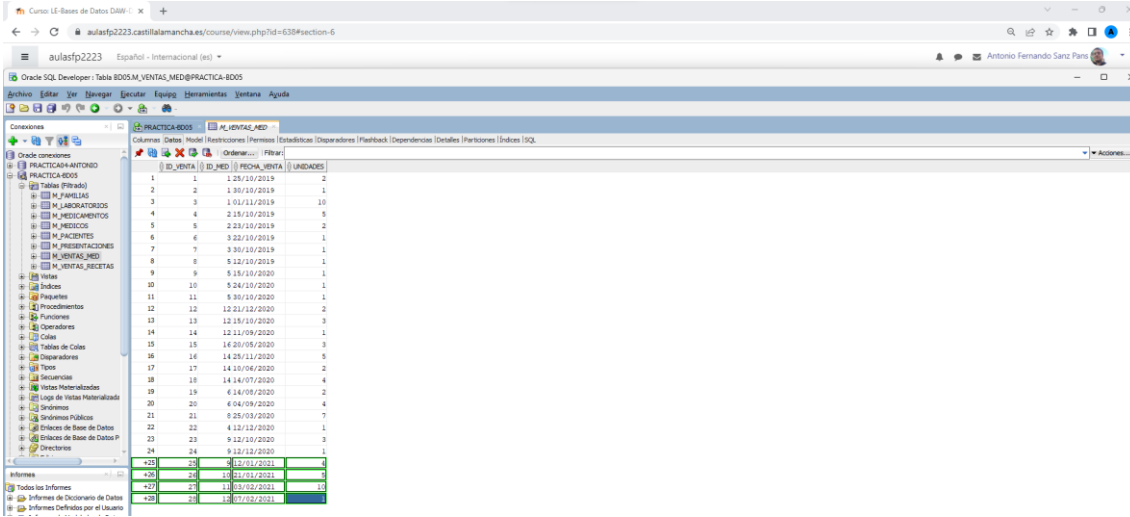
### Capturas de pantalla

Añadir datos en M\_VENTAS\_MED mediante la herramienta gráfica

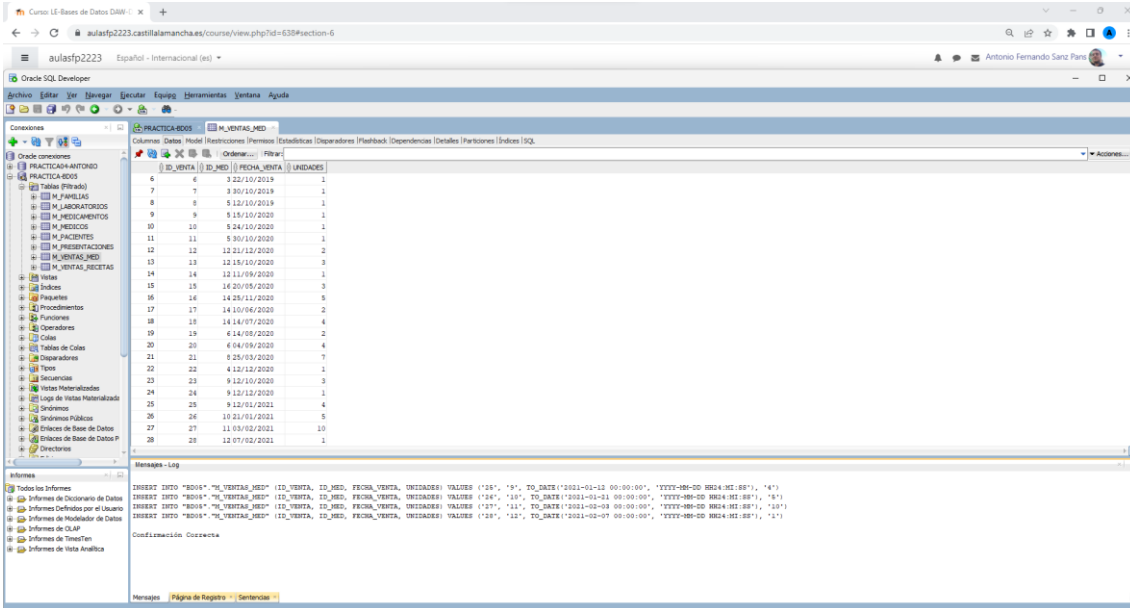
Antes



## Después

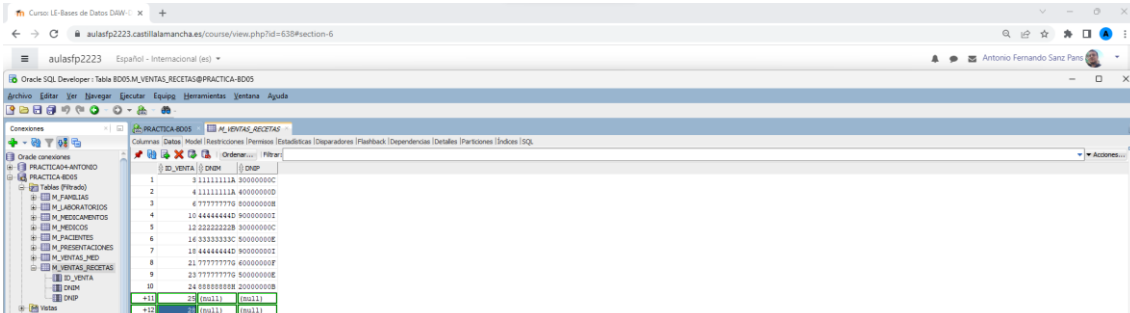


ID_VENTA	ID_MED	FECHA_VENTA	UNIDADES
1	1	1/25/10/2019	2
2	2	1/30/10/2019	1
3	3	1/01/11/2019	10
4	4	2/15/10/2019	5
5	5	2/23/10/2019	2
6	6	3/22/10/2019	1
7	7	3/30/10/2019	1
8	8	5/12/10/2019	1
9	9	5/15/10/2020	1
10	10	5/24/10/2020	1
11	11	5/30/10/2020	1
12	12	12/21/12/2020	2
13	13	12/15/10/2020	3
14	14	12/11/09/2020	1
15	15	14/20/09/2020	3
16	16	14/25/11/2020	5
17	17	14/10/04/2020	2
18	18	14/14/07/2020	4
19	19	6/14/09/2020	2
20	20	6/04/09/2020	4
21	21	8/25/09/2020	7
22	22	4/12/12/2020	1
23	23	9/12/10/2020	3
24	24	9/12/12/2020	1
25	25	12/02/2021	4
26	26	10/21/01/2021	5
27	27	11/03/02/2021	10
28	28	12/07/02/2021	1

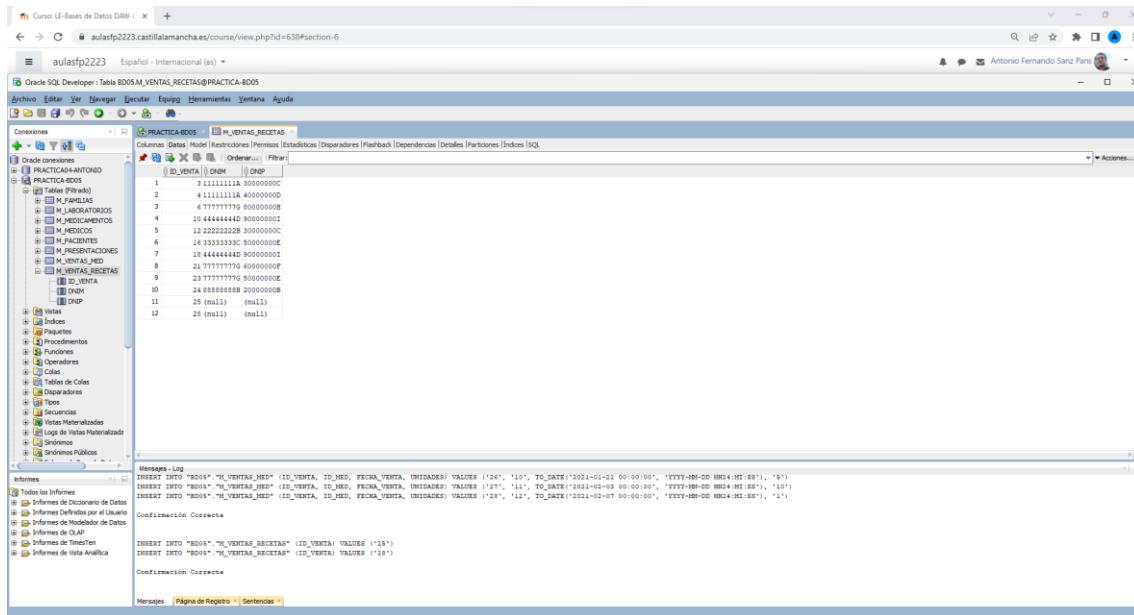


```
INSERT INTO "BD05"."M_VENTAS_MED" ("ID_VENTA", "ID_MED", "FECHA_VENTA", "UNIDADES") VALUES ('24', '9', TO_DATE('2021-09-12 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), '1');
INSERT INTO "BD05"."M_VENTAS_MED" ("ID_VENTA", "ID_MED", "FECHA_VENTA", "UNIDADES") VALUES ('25', '10', TO_DATE('2021-01-02 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), '4');
INSERT INTO "BD05"."M_VENTAS_MED" ("ID_VENTA", "ID_MED", "FECHA_VENTA", "UNIDADES") VALUES ('26', '26', TO_DATE('2021-01-21 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), '5');
INSERT INTO "BD05"."M_VENTAS_MED" ("ID_VENTA", "ID_MED", "FECHA_VENTA", "UNIDADES") VALUES ('27', '27', TO_DATE('2021-02-03 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), '10');
INSERT INTO "BD05"."M_VENTAS_MED" ("ID_VENTA", "ID_MED", "FECHA_VENTA", "UNIDADES") VALUES ('28', '28', TO_DATE('2021-02-07 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), '1');
```

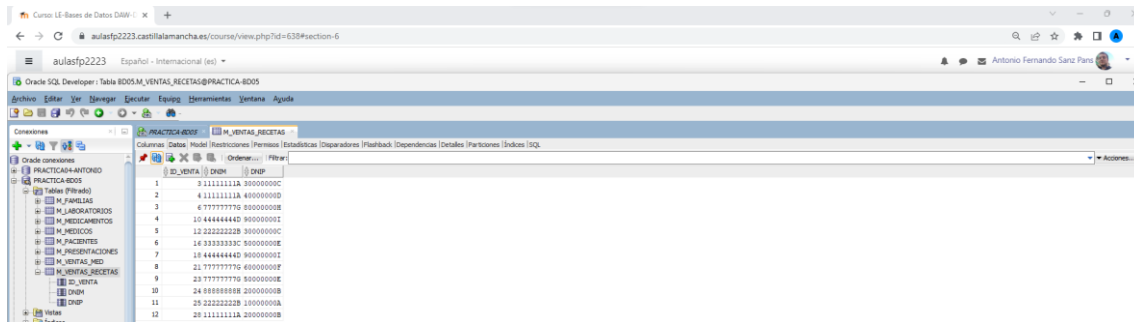
Añadir los registros con ID 24 y 25 pero sin DNI para el medico ni el paciente.



ID_VENTA	DNI_MED	DNI_PAC
1	311111111A	300000000C
2	311111111A	400000000D
3	677777777B	800000000E
4	104444444D	900000000F
5	122222222B	300000000C
6	143333333C	500000000E
7	164444444D	900000000F
8	217777777G	600000000F
9	237777777G	500000000E
10	246666666B	200000000D
24	NULL	NULL
25	NULL	NULL



## Captura de pantalla después del update mediante sentencia SQL



## Sentencia SQL

```
UPDATE m_ventas_recetas SET dnim='2222222B', dni='10000000A' WHERE id_venta=25;
UPDATE m_ventas_recetas SET dnim='1111111A', dni='20000000B' WHERE id_venta=28;
```

## EJERCICIO 2

### Enunciado

Inserta varios registros en la tabla **M\_MEDICOS** utilizando sentencias SQL. En la entrega de la tarea debes copiar las sentencias que has utilizado. Los datos deben ser los siguientes:

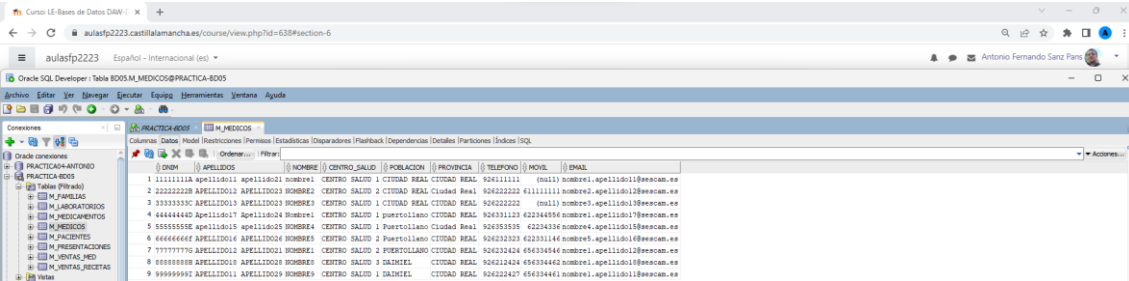
Tabla **M\_MEDICOS**

DNIM	APELLIDOS	NOMBRE	CENTRO SALUD	POBLACIÓN	PROVINCIA	TELEFONO	MOVIL	EMAIL
11111112B	Sanz Hervás	Maria	Centro Salud 1	Ciudad Real	Ciudad Real	926212121		
11111113B	RAMOS CRUZ	JUAN	Centro Salud 3	DAIMIEL	CIUDAD REAL			
11111114B	LAOS MIS	PEDRO		PUERTOLLANO				PLM@gmail.com
11111115B	Lagos Cortés	Marina			TOLEDO		622622622	

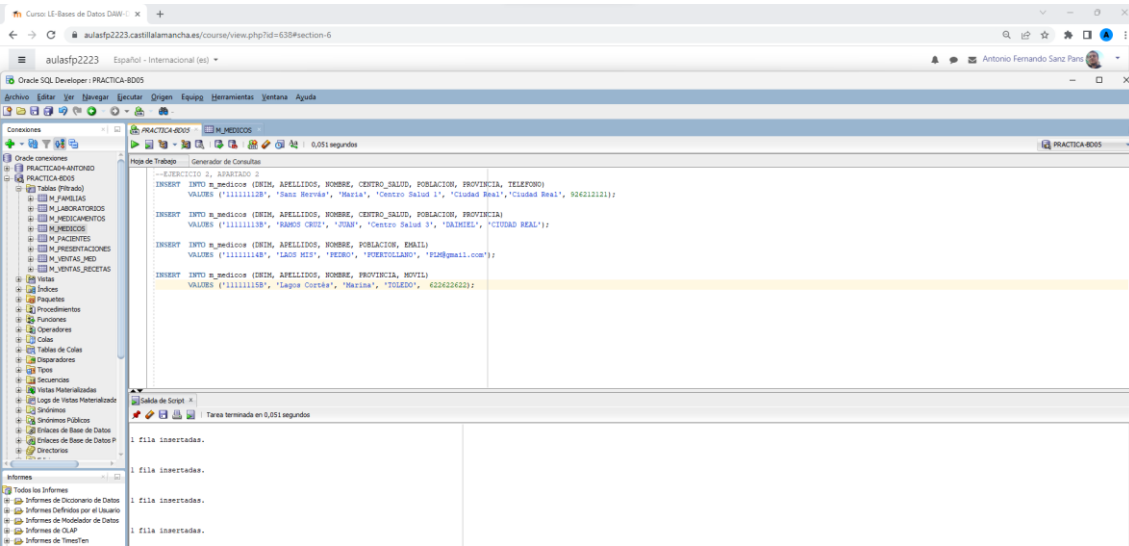
Las columnas con los datos que aparecen en blanco no deben utilizarse en las sentencias INSERT.

### Capturas de pantalla

Tabla M\_MEDICOS antes de insertar los registros

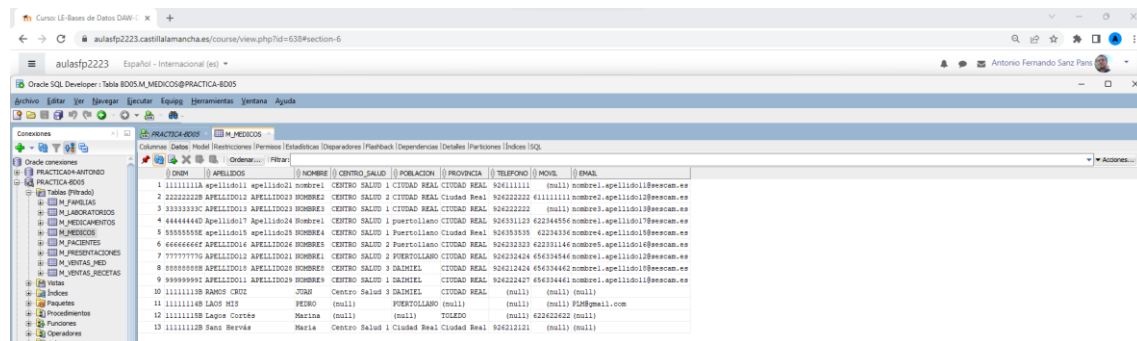


Insert de los registros





## Tabla M\_MEDICOS después de insertar los registros



The screenshot shows the Oracle SQL Developer interface with the 'M\_MEDICOS' table selected. The table contains 13 rows of data. The columns are: DNIM, APELLIDOS, NOMBRE, CENTRO\_SALUD, POBLACION, PROVINCIA, TELEFONO, MOVIL, and EMAIL. The data is as follows:

DNIM	APELLIDOS	NOMBRE	CENTRO_SALUD	POBLACION	PROVINCIA	TELEFONO	MOVIL	EMAIL
1 1111111A	apellido11	apellido11	nombre1	CENTRO SALUD 1	CIUDAD REAL	926111111	(null)	nombre1.apellido11@escan.es
2 2222222B	apellido12	apellido12	nombre2	CENTRO SALUD 2	CIUDAD REAL	926222222	611111111	nombre2.apellido12@escan.es
3 3333333C	apellido13	apellido13	nombre3	CENTRO SALUD 3	CIUDAD REAL	926333333	(null)	nombre3.apellido13@escan.es
4 4444444D	apellido14	apellido14	nombre4	CENTRO SALUD 4	PUERTOLLANO	926444444	622444444	nombre4.apellido14@escan.es
5 5555555E	apellido15	apellido15	nombre5	CENTRO SALUD 5	PUERTOLLANO	926555555	622555555	nombre5.apellido15@escan.es
6 6666666F	apellido16	apellido16	nombre6	CENTRO SALUD 6	PUERTOLLANO	926666666	622666666	nombre6.apellido16@escan.es
7 7777777G	apellido17	apellido17	nombre7	CENTRO SALUD 7	PUERTOLLANO	926777777	622777777	nombre7.apellido17@escan.es
8 8888888H	apellido18	apellido18	nombre8	CENTRO SALUD 8	PUERTOLLANO	926888888	622888888	nombre8.apellido18@escan.es
9 9999999I	apellido19	apellido19	nombre9	CENTRO SALUD 9	PUERTOLLANO	926999999	622999999	nombre9.apellido19@escan.es
10 1111111B	RAMOS CRUZ	JUAN	Centro Salud 3	DAIMIEL	CIUDAD REAL	(null)	(null)	(null)
11 1111111B	LAOS MIS	PEDRO	(null)	PUERTOLLANO	(null)	(null)	(null)	PLM@gmail.com
12 1111111B	Lagos Cortés	Marina	(null)	(null)	TOLEDO	(null)	622622622	(null)
13 1111111B	Sanz Hervás	Maria	Centro Salud 1	Ciudad Real	Ciudad Real	926121212	(null)	(null)

## Sentencia SQL

```
INSERT INTO m_medicos (DNIM, APELLIDOS, NOMBRE, CENTRO_SALUD, POBLACION, PROVINCIA, TELEFONO)
VALUES ('11111112B', 'Sanz Hervás', 'Maria', 'Centro Salud 1', 'Ciudad Real', 'Ciudad Real', 926212121);
```

```
INSERT INTO m_medicos (DNIM, APELLIDOS, NOMBRE, CENTRO_SALUD, POBLACION, PROVINCIA)
VALUES ('11111113B', 'RAMOS CRUZ', 'JUAN', 'Centro Salud 3', 'DAIMIEL', 'CIUDAD REAL');
```

```
INSERT INTO m_medicos (DNIM, APELLIDOS, NOMBRE, POBLACION, EMAIL)
VALUES ('11111114B', 'LAOS MIS', 'PEDRO', 'PUERTOLLANO', 'PLM@gmail.com');
```

```
INSERT INTO m_medicos (DNIM, APELLIDOS, NOMBRE, PROVINCIA, MOVIL)
VALUES ('11111115B', 'Lagos Cortés', 'Marina', 'TOLEDO', 622622622);
```

## EJERCICIO 3

### Enunciado

Inserta un laboratorio con los siguientes datos:

ID DEL LABORATORIO: 5, NOMBRE LABORATORIO: FARMAREAL, DIRECCION: AVENIDA UNIVERSIDAD, TELEFONO: 926212121, FAX: 926222222, EMAIL: info@farmareal.com

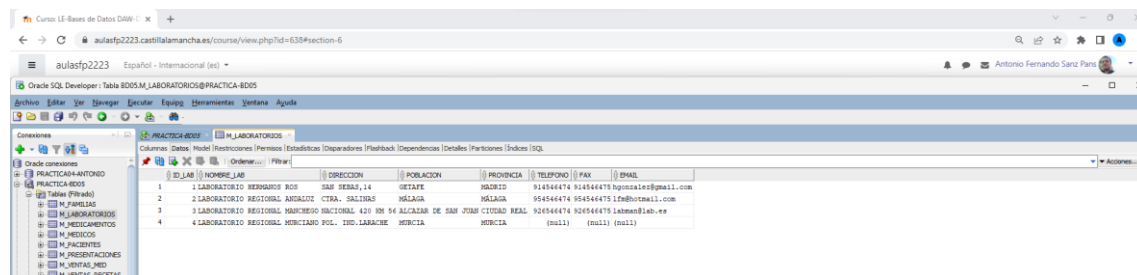
La población y la provincia serán los mismos que los del laboratorio cuyo ID es 3.

Se realizará en una sola sentencia insert (utilizando una select si es necesario).

No vale averiguar los valores y ponerlos después en la sentencia de inserción.

### Capturas de pantalla

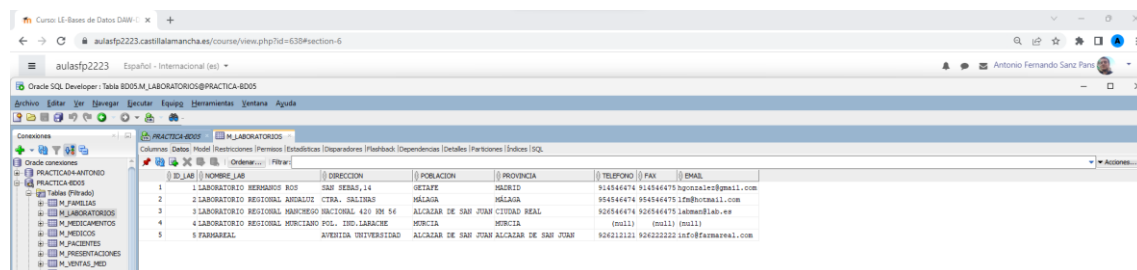
M\_LABORATORIOS antes:



The screenshot shows the Oracle SQL Developer interface with the 'M\_LABORATORIOS' table selected. The table contains four rows of data:

ID_LAB	NOMBRE_LAB	DIRECCION	POBLACION	PROVINCIA	TELEFONO	FAX	EMAIL
1	LABORATORIO HERMANOS ROS	SAN DEIAS,14	GETAFE	MADRID	914544474	914544475	hrosales@gmail.com
2	LABORATORIO REGIONAL ARGALIST	CTRA. SALINOS	MELILLA	MELILLA	954544474	954544475	lne@rcomail.com
3	LABORATORIO REGIONAL HERNANDEZ NACIONAL	425 KM 54	ALCAZAR DE SAN JUAN	CIUDAD REAL	926544474	926544475	lahernan@lab.es
4	LABORATORIO REGIONAL MENCIANO PCL.	IND. LARACHE	MERICIA	MERICIA	(null)	(null)	(null)

M\_LABORATORIOS después:



The screenshot shows the Oracle SQL Developer interface with the 'M\_LABORATORIOS' table selected. The table now contains five rows of data, with the new row added at the bottom:

ID_LAB	NOMBRE_LAB	DIRECCION	POBLACION	PROVINCIA	TELEFONO	FAX	EMAIL
1	LABORATORIO HERMANOS ROS	SAN DEIAS,14	GETAFE	MADRID	914544474	914544475	hrosales@gmail.com
2	LABORATORIO REGIONAL ARGALIST	CTRA. SALINOS	MELILLA	MELILLA	954544474	954544475	lne@rcomail.com
3	LABORATORIO REGIONAL HERNANDEZ NACIONAL	425 KM 54	ALCAZAR DE SAN JUAN	CIUDAD REAL	926544474	926544475	lahernan@lab.es
4	LABORATORIO REGIONAL MENCIANO PCL.	IND. LARACHE	MERICIA	MERICIA	(null)	(null)	(null)
5	FARMAREAL	AVENIDA UNIVERSIDAD	ALCAZAR DE SAN JUAN	ALCAZAR DE SAN JUAN	926212121	926222222	info@farmareal.com

### Sentencia SQL

```
INSERT INTO m_laboratorios (ID_LAB, NOMBRE_LAB, DIRECCION, POBLACION, PROVINCIA,
TELEFONO, FAX, EMAIL)
VALUES (
5,
'FARMAREAL',
'AVENIDA UNIVERSIDAD',
(SELECT poblacion from m_laboratorios where id_lab = 3 ),
(SELECT poblacion from m_laboratorios where id_lab = 3 ),
926212121,
926222222,
'info@farmareal.com'
);
```

## EJERCICIO 4

### Enunciado

Elimina de la tabla **M\_FAMILIAS** el registro cuyo ID\_FAM es 2 ¿te deja eliminarlo? Razona la respuesta.

Elimina de la tabla **M\_PRESENTACIONES** los registros de presentaciones que no estén en ningún medicamento.

Debes realizarlo usando una sola sentencia SQL. No vale averiguar los valores y ponerlos después en la sentencia de eliminación.

Elimina de la tabla **M\_FAMILIAS** los registros de las familias que no estén en ningún medicamento. Debes realizarlo usando una sola sentencia SQL. No vale averiguar los valores y ponerlos después en la sentencia de eliminación.

### Respuesta

Para intentar borrar el registro de M\_FAMILIAS he utilizado:

```
DELETE FROM m_familias WHERE m_familias.id_fam = 2;
```

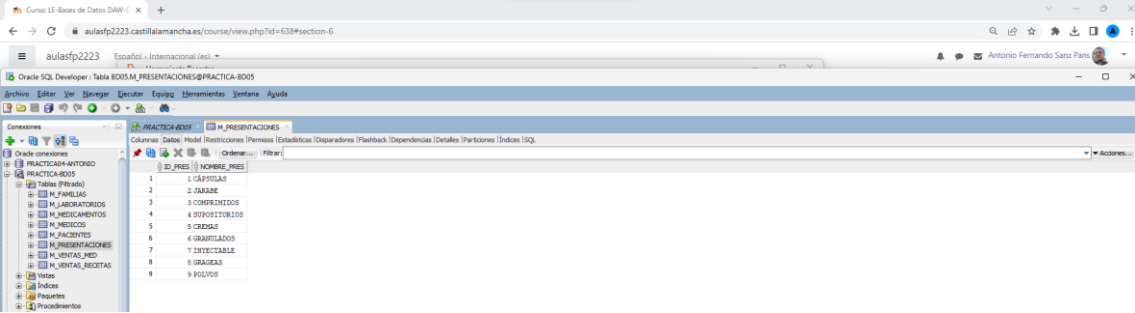
Restricción de integridad violada: No deja borrar el registro ya que hay campos asociados en otras tablas mediante claves ajenas.

```
Error que empieza en la línea: 32 del comando :  
DELETE FROM m_familias WHERE m_familias.id_fam = 2  
Informe de error -  
ORA-02292: restricción de integridad (BD05.SYS_C007740) violada - registro secundario encontrado
```

### Capturas de pantalla

Eliminar de la tabla M\_PRESTACIONES

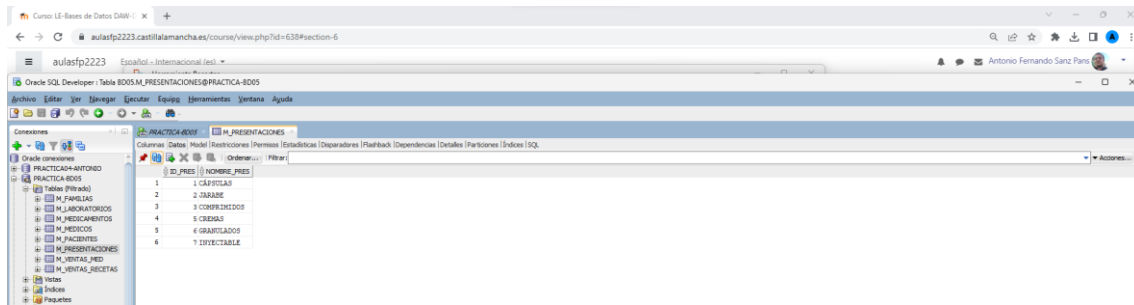
Antes



The screenshot shows the Oracle SQL Developer interface. On the left, a tree view displays the database structure, including tables like M\_FAMILIAS, M\_MEDICAMENTOS, M\_PRESENTACIONES, and M\_PRESTACIONES. The main window shows the 'M\_PRESENTACIONES' table with the following data:

ID_PRES	NOMBRE_PRES
1	1. CAPSULAS
2	2. JARABE
3	3. COMPRENDIDOS
4	4. SUPPOSITORIOS
5	5. CREMA
6	6. GRANULADOS
7	7. INYECTABLE
8	8. GRADEAS
9	9. POLVOS

Después

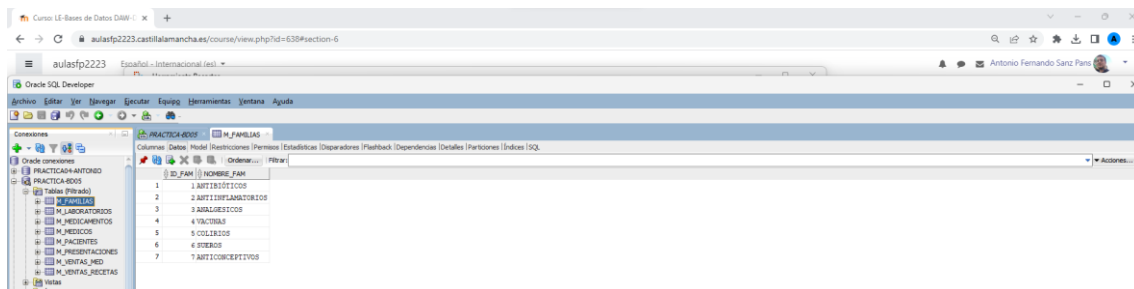


The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with the 'M\_PRESENTACIONES' table selected. The main pane shows the table's data in a grid view.

ID_PRES	NOMBRE_PRES
1	1 CASIPOLAS
2	2 JABASE
3	3 CONFORTINDOS
4	5 CRODAS
5	6 GRANTILLADOS
6	7 INJECTABLE

## Eliminar de la tabla M\_FAMILIAS

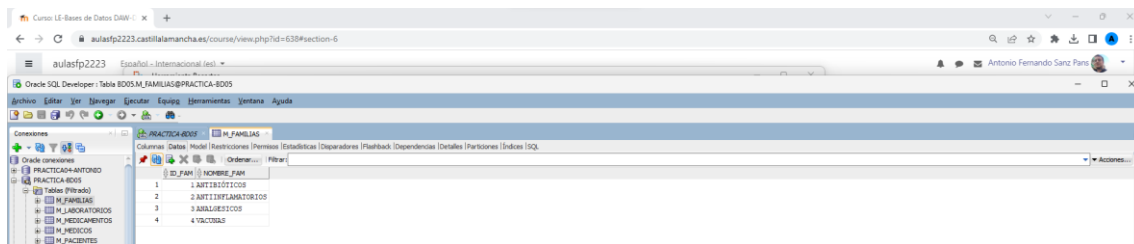
Antes



The screenshot shows the Oracle SQL Developer interface with the 'M\_FAMILIAS' table selected. The table contains 7 records.

ID_FAM	NOMBRE_FAM
1	1 ANTIBIOTICOS
2	2 ANTIINFLAMATORIOS
3	3 ANALGESICOS
4	4 VACUNAS
5	5 COLIROS
6	6 SIROPOS
7	7 ANTICONCEPTIVOS

Después



The screenshot shows the Oracle SQL Developer interface with the 'M\_FAMILIAS' table selected. The table now contains 4 records, as the record with ID\_FAM = 2 has been deleted.

ID_FAM	NOMBRE_FAM
1	1 ANTIBIOTICOS
2	2 ANTIINFLAMATORIOS
3	3 ANALGESICOS
4	4 VACUNAS

## Sentencias SQL

```
/* Elimina de la tabla M_FAMILIAS el registro cuyo ID_FAM es 2 */  
DELETE FROM m_familias WHERE m_familias.id_fam = 2;  
  
/* Elimina de la tabla M_PRESENTACIONES los registros de presentaciones que no estén en  
ningún medicamento. */  
DELETE FROM M_PRESENTACIONES  
WHERE ID_PRES NOT IN  
    (SELECT DISTINCT ID_PRES FROM M_MEDICAMENTOS);  
  
/* Elimina de la tabla M_FAMILIAS los registros de las familias que no estén en ningún  
medicamento. */  
DELETE FROM M_FAMILIAS  
WHERE ID_FAM NOT IN (SELECT DISTINCT ID_FAM FROM M_MEDICAMENTOS);
```

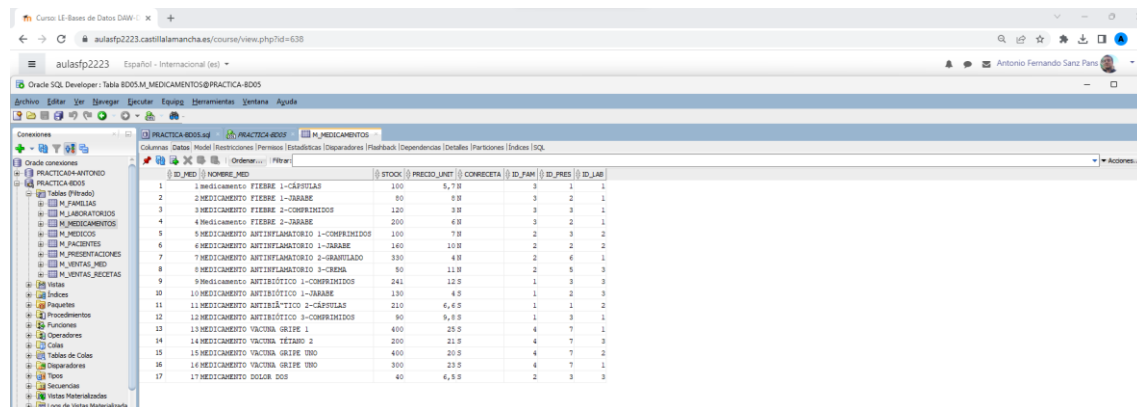
## EJERCICIO 5

### Enunciado

A todos los medicamentos de las familias de Antiinflamatorios y Vacunas, se les aumentará el stock en 10 unidades. Debes realizarlo usando una sola sentencia SQL. No vale averiguar los valores y ponerlos después en la sentencia de actualización.

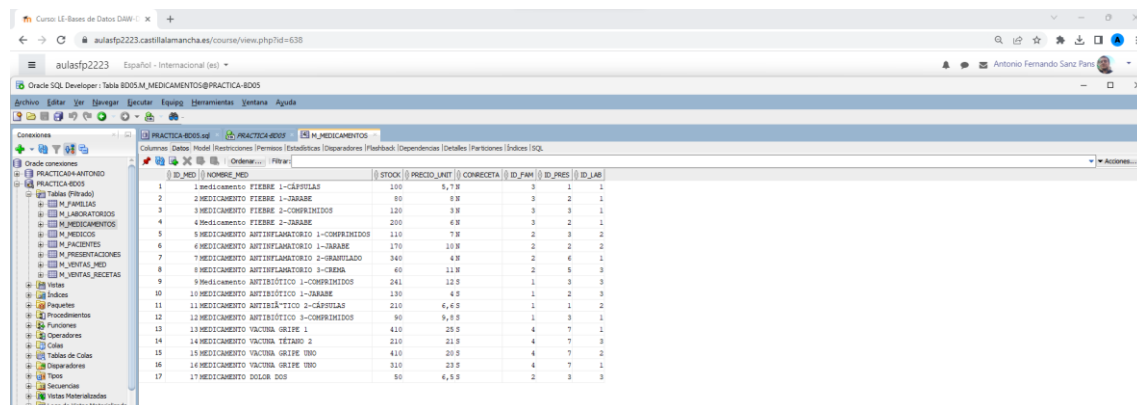
### Capturas de pantalla

Antes:



ID_MED	NOMBRE_MED	STOCK	PRECIO_UNIT	COMERCETA	ID_FAM	ID_PRES	ID_LAB
1	Medicamento FIEBRE 1-CÁPSULAS	100	5,70	3	1	1	
2	Medicamento FIEBRE 1-DABARE	80	8,0	3	2	1	
3	Medicamento FIEBRE 2-COMPRIMIDOS	120	3,0	3	3	1	
4	Medicamento FIEBRE 2-DABARE	200	6,0	3	2	1	
5	Medicamento ANTIINFLAMATORIO 1-COMPRIMIDOS	100	7,0	2	3	2	
6	Medicamento ANTIINFLAMATORIO 1-DABARE	160	10,0	2	2	2	
7	Medicamento ANTIINFLAMATORIO 2-GRANULADO	300	6,0	2	6	1	
8	Medicamento ANTIINFLAMATORIO 3-CREMA	50	11,0	2	5	3	
9	Medicamento ANTIBIOTICO 1-COMPRIMIDOS	240	12,0	1	3	3	
10	Medicamento ANTIBIOTICO 1-DABARE	100	6,0	1	3	3	
11	Medicamento ANTIBIOTICO 2-CÁPSULAS	210	6,60	1	1	2	
12	Medicamento ANTIBIOTICO 3-COMPRIMIDOS	90	9,90	1	3	1	
13	Medicamento VACUNA GRIPE 1	400	20,0	4	7	1	
14	Medicamento VACUNA TETANO 2	200	21,0	4	7	3	
15	Medicamento VACUNA GRIPE TMO	400	20,0	4	7	2	
16	Medicamento VACUNA GRIPE TMO	300	20,0	4	7	1	
17	Medicamento DOLOR DOS	40	6,90	2	3	3	

Después



ID_MED	NOMBRE_MED	STOCK	PRECIO_UNIT	COMERCETA	ID_FAM	ID_PRES	ID_LAB
1	Medicamento FIEBRE 1-CÁPSULAS	100	5,70	3	1	1	
2	Medicamento FIEBRE 1-DABARE	90	8,0	3	2	1	
3	Medicamento FIEBRE 2-COMPRIMIDOS	120	3,0	3	3	1	
4	Medicamento FIEBRE 2-DABARE	200	6,0	3	2	1	
5	Medicamento ANTIINFLAMATORIO 1-COMPRIMIDOS	110	7,0	2	3	2	
6	Medicamento ANTIINFLAMATORIO 1-DABARE	170	10,0	2	2	2	
7	Medicamento ANTIINFLAMATORIO 2-GRANULADO	310	6,0	2	6	1	
8	Medicamento ANTIINFLAMATORIO 3-CREMA	60	11,0	2	5	3	
9	Medicamento ANTIBIOTICO 1-COMPRIMIDOS	240	12,0	1	3	3	
10	Medicamento ANTIBIOTICO 1-DABARE	100	6,0	1	3	3	
11	Medicamento ANTIBIOTICO 2-CÁPSULAS	210	6,60	1	1	2	
12	Medicamento ANTIBIOTICO 3-COMPRIMIDOS	90	9,90	1	3	1	
13	Medicamento VACUNA GRIPE 1	410	20,0	4	7	1	
14	Medicamento VACUNA TETANO 2	210	21,0	4	7	3	
15	Medicamento VACUNA GRIPE TMO	410	20,0	4	7	2	
16	Medicamento VACUNA GRIPE TMO	310	20,0	4	7	1	
17	Medicamento DOLOR DOS	50	6,90	2	3	3	

### Sentencia SQL

```
/*A todos los medicamentos de las familias de Antiinflamatorios y Vacunas, se les aumentará el stock en 10 unidades.*/  
UPDATE M_MEDICAMENTOS SET STOCK = (STOCK + 10) WHERE ID_FAM IN (2, 4);
```

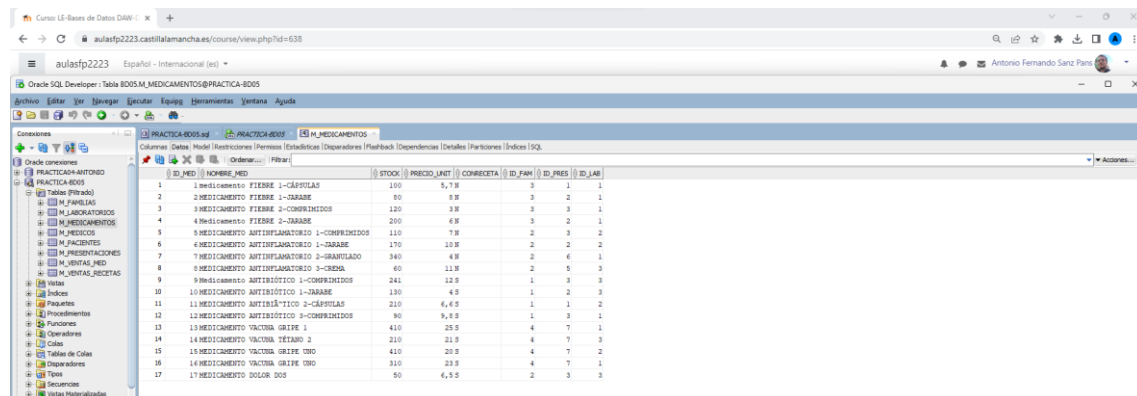
## EJERCICIO 6

### Enunciado

En la tabla **M\_MEDICAMENTOS**, actualiza el campo **PRECIO\_UNIT** incrementándolo un 10% a los medicamentos que hayan vendido más de 10 unidades en el año 2020. Debes realizarlo usando una sola sentencia SQL. No vale averiguar los valores y ponerlos después en la sentencia de actualización.

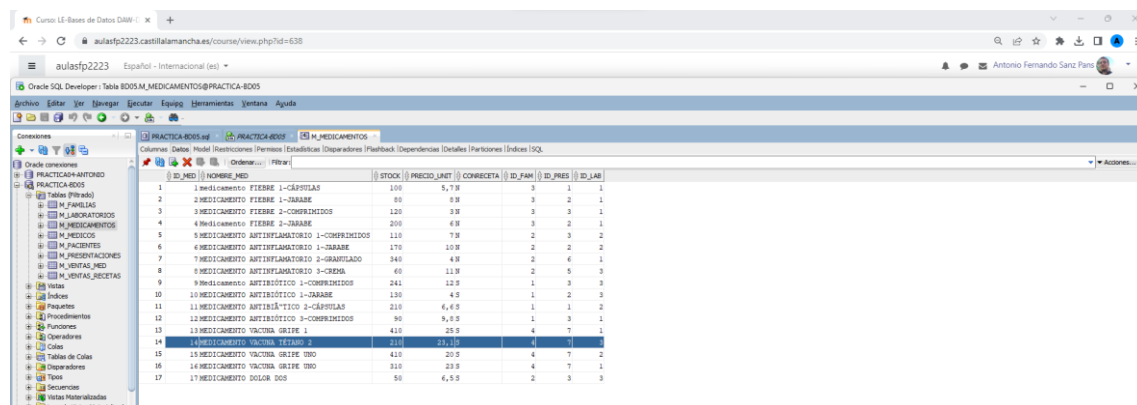
### Capturas de pantalla

Antes



ID_MED	NOMBRE_MED	STOCK	PRECIO_UNIT	CORRECTA	ID_FAM	ID_PRES	ID_LAB
1	medicamento FIEBRE 1-CÁPSULAS	100	5,70	3	1	1	
2	medicamento FIEBRE 1-GRABAR	80	9,9	3	2	1	
3	medicamento FIEBRE 2-COMPRIMIDOS	120	3,9	3	3	1	
4	medicamento FIEBRE 2-GRABAR	200	6,9	3	2	1	
5	medicamento ANTIFILANATORIO 1-COMPRIMIDOS	110	7,9	2	3	2	
6	medicamento ANTIFILANATORIO 1-GRABAR	170	10,9	2	2	2	
7	medicamento ANTIFILANATORIO 2-GRABAR	340	6,9	2	6	1	
8	medicamento ANTIFILANATORIO 3-GRABAR	60	11,9	2	5	3	
9	medicamento ANTIBIOTICO 1-COMPRIMIDOS	241	12,9	1	3	3	
10	medicamento ANTIBIOTICO 1-GRABAR	130	6,9	1	2	3	
11	medicamento ANTIBIOTICO 2-CÁPSULAS	210	6,6,9	1	1	2	
12	medicamento ANTIBIOTICO 3-COMPRIMIDOS	90	9,6,9	1	3	1	
13	medicamento VACUNA GRIPE 1	410	25,9	4	7	1	
14	medicamento VACUNA TETANO 2	210	21,9	4	7	3	
15	medicamento VACUNA GRIPE SHO	410	20,9	4	7	2	
16	medicamento VACUNA GRIPE SHO	310	23,9	4	7	1	
17	medicamento DOLOR DOS	50	6,5,9	2	3	3	

Después



ID_MED	NOMBRE_MED	STOCK	PRECIO_UNIT	CORRECTA	ID_FAM	ID_PRES	ID_LAB
1	medicamento FIEBRE 1-CÁPSULAS	100	5,70	3	1	1	
2	medicamento FIEBRE 1-GRABAR	80	9,9	3	2	1	
3	medicamento FIEBRE 2-COMPRIMIDOS	120	3,9	3	3	1	
4	medicamento FIEBRE 2-GRABAR	200	6,9	3	2	1	
5	medicamento ANTIFILANATORIO 1-COMPRIMIDOS	110	7,9	2	3	2	
6	medicamento ANTIFILANATORIO 1-GRABAR	170	10,9	2	2	2	
7	medicamento ANTIFILANATORIO 2-GRABAR	340	6,9	2	6	1	
8	medicamento ANTIFILANATORIO 3-GRABAR	60	11,9	2	5	3	
9	medicamento ANTIBIOTICO 1-COMPRIMIDOS	241	12,9	1	3	3	
10	medicamento ANTIBIOTICO 1-GRABAR	130	6,9	1	2	3	
11	medicamento ANTIBIOTICO 2-CÁPSULAS	210	6,6,9	1	1	2	
12	medicamento ANTIBIOTICO 3-COMPRIMIDOS	90	9,6,9	1	3	1	
13	medicamento VACUNA GRIPE 1	410	28,49	4	7	1	
14	medicamento VACUNA TETANO 2	210	21,9	4	7	3	
15	medicamento VACUNA GRIPE SHO	410	20,9	4	7	2	
16	medicamento VACUNA GRIPE SHO	310	23,9	4	7	1	
17	medicamento DOLOR DOS	50	6,5,9	2	3	3	

### Sentencia SQL

/\*En la tabla M\_MEDICAMENTOS, actualiza el campo PRECIO\_UNIT incrementándolo un 10% a los medicamentos que hayan vendido mas de 10 unidades en el año 2020\*/

```
UPDATE m_medicamentos
SET precio_unit = precio_unit * 1.1
WHERE id_med IN (
  SELECT id_med
  FROM m_ventas_med
  WHERE fecha_venta >= TO_DATE('01/01/2020', 'DD/MM/YYYY')
  AND fecha_venta < TO_DATE('01/01/2021', 'DD/MM/YYYY')
  GROUP BY id_med
  HAVING SUM(unidades) > 10
);
```

## EJERCICIO 7

### Enunciado

Crea una tabla llamada **M\_TOTAL\_VENTAS** que tenga las siguientes columnas:

ID\_MED: Number(4), UNIDADES VENDIDAS: Number(6), TOTAL\_VENTAS : Number(9,2)

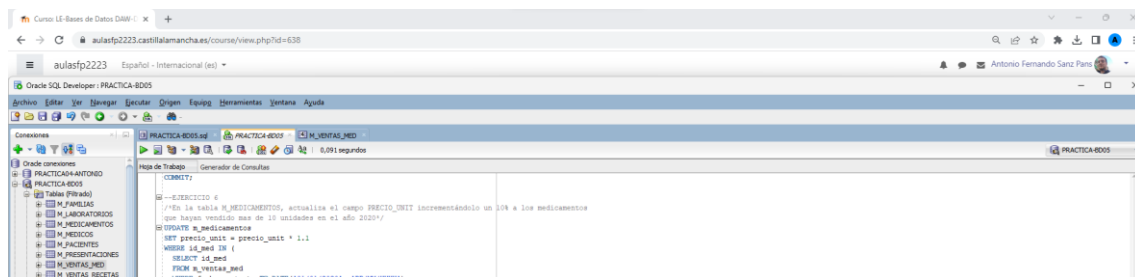
Inserta en la tabla **M\_TOTAL\_VENTAS** un registro por cada medicamento, almacenando:

ID\_MED, el número de unidades totales vendidas de ese medicamento y total de todas las ventas (se calcula multiplicando las unidades por el precio unitario). Los registros que hay que insertar son:

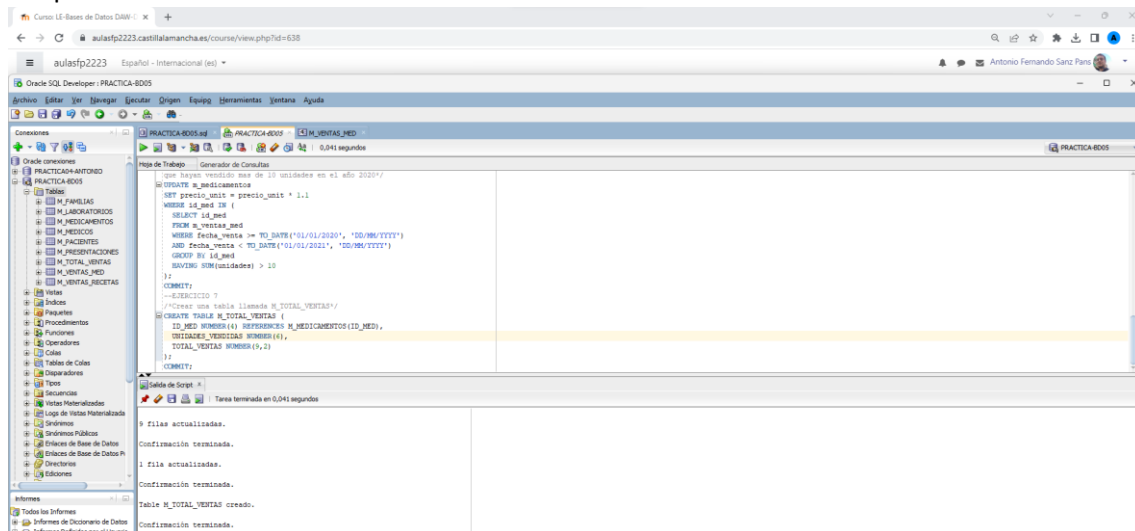
ID_MED	UNIDADES	TOTAL_VENTAS
1	13	74,1
2	7	56
3	2	6
4	1	6
5	4	28
6	6	60
8	7	77
9	4	48
12	6	58,8
14	11	231
16	3	69

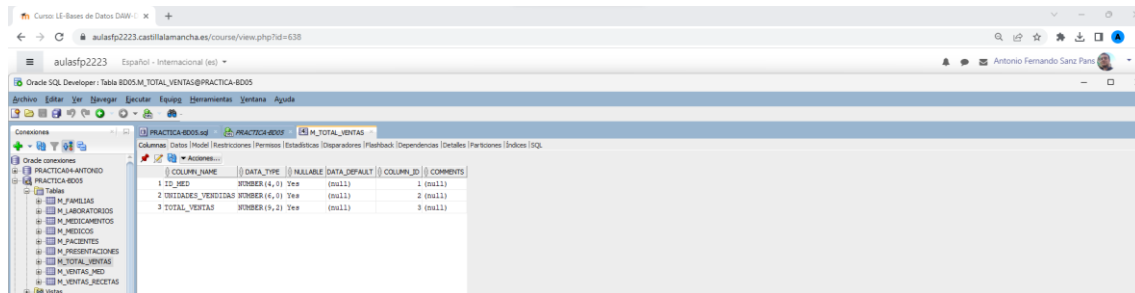
## Capturas de pantalla

## Antes de crear la tabla

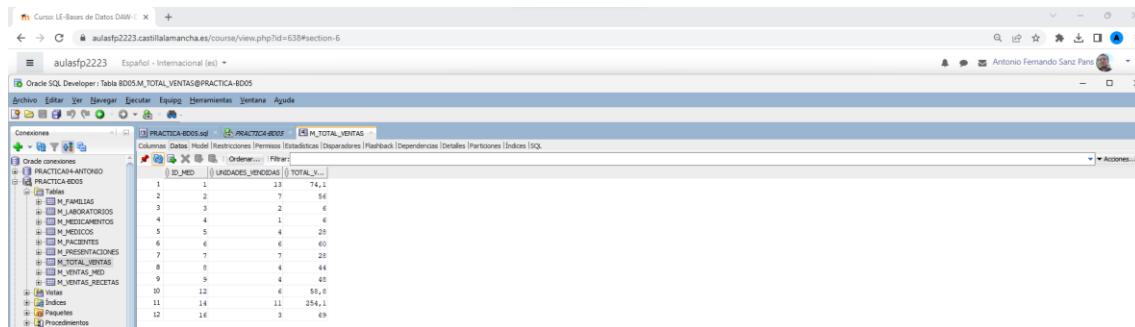


## Después de crear la tabla





## Inserción de valores:



## Sentencia SQL

```
/*Crear una tabla llamada M_TOTAL_VENTAS*/  
CREATE TABLE M_TOTAL_VENTAS (  
    ID_MED NUMBER(4) REFERENCES M_MEDICAMENTOS(ID_MED),  
    UNIDADES_VENDIDAS NUMBER(6),  
    TOTAL_VENTAS NUMBER(9,2)  
);  
COMMIT;  
/*Insertar los datos en la tabla*/  
INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)  
VALUES(1, 13, 13*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =  
1));  
  
INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)  
VALUES(2, 7, 7*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =  
2));  
  
INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)  
VALUES(3, 2, 2*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =  
3));  
  
INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)  
VALUES(4, 1, 1*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =  
4));  
  
INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)  
VALUES(5, 4, 4*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =  
5));  
INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)  
VALUES(6, 6, 6*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =  
6));  
  
INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)  
VALUES(7, 7, 7*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =  
7));  
  
INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)  
VALUES(8, 4, 4*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =  
8));  
  
INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)  
VALUES(9, 4, 4*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =  
9));
```



```
INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)
VALUES(12, 6, 6*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =
12));

INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)
VALUES(14, 11, 11*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =
14));

INSERT INTO m_total_ventas (ID_MED, UNIDADES_VENDIDAS, TOTAL_VENTAS)
VALUES(16, 3, 3*(SELECT PRECIO_UNIT FROM m_medicamentos WHERE m_medicamentos.id_med =
16));

COMMIT;
```

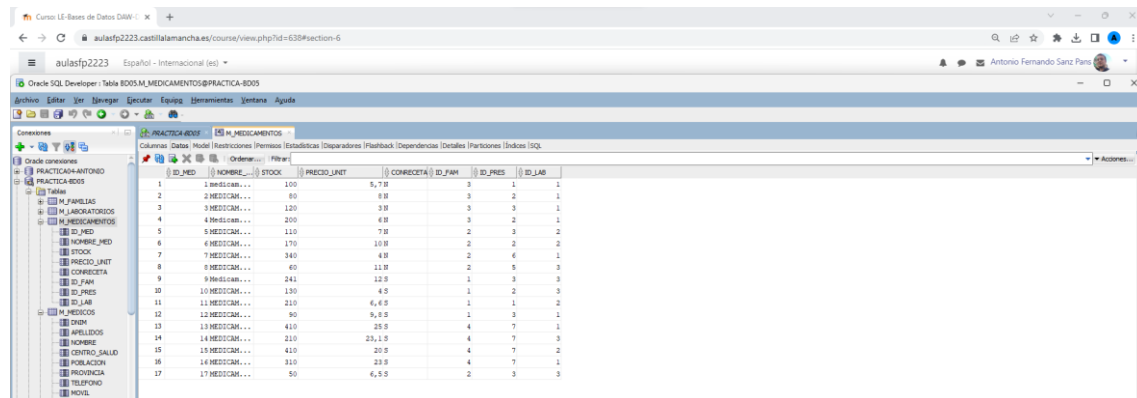
## EJERCICIO 8

### Enunciado

Se quiere actualizar la columna STOCK de la tabla M\_MEDICAMENTOS . Hay que sumar todas las unidades vendidas de ese medicamento y restarlas al STOCK. Debes hacerlo usando una sola sentencia SQL

### Capturas de pantalla

Antes



The screenshot shows the Oracle SQL Developer interface with the M\_MEDICAMENTOS table selected. The table contains 17 rows of data. The columns are: ID\_MED, NOMBRE, STOCK, PRECIO\_UNIT, CORRECTA, ID\_FAM, ID\_PRES, and ID\_LAB.

ID_MED	NOMBRE	STOCK	PRECIO_UNIT	CORRECTA	ID_FAM	ID_PRES	ID_LAB
1	1 Medicam...	100	5,70	3	1	1	1
2	2 Medicam...	80	9,0	3	2	1	1
3	3 Medicam...	120	3,0	3	3	1	1
4	4 Medicam...	200	6,0	3	2	1	1
5	5 Medicam...	110	7,0	2	3	2	1
6	6 Medicam...	170	10,0	2	2	2	1
7	7 Medicam...	340	4,0	2	6	1	1
8	8 Medicam...	60	11,0	2	5	3	1
9	9 Medicam...	241	12,0	1	3	3	1
10	10 Medicam...	150	4,5	1	2	3	1
11	11 Medicam...	210	6,6	1	1	2	1
12	12 Medicam...	90	9,9	1	3	1	1
13	13 Medicam...	410	25,0	4	7	1	1
14	14 Medicam...	210	20,1	4	7	3	1
15	15 Medicam...	410	20,0	4	7	2	1
16	16 Medicam...	310	23,0	4	7	1	1
17	17 Medicam...	50	6,5	2	3	3	1

### Sentencia SQL

```
UPDATE M_MEDICAMENTOS
SET STOCK = STOCK - (
    SELECT UNIDADES_VENDIDAS FROM M_TOTAL_VENTAS WHERE M_TOTAL_VENTAS.ID_MED =
M_MEDICAMENTOS.ID_MED
);
```