

Mario Rubio Avila BD Tarea 5.1

Oracle database me da algunos problemas por eso hay un añadido la principio de todo lo que he tenido que ir haciendo para repararlos lo incluyo aquí como añadido. Aun así, la tarea comienza en la pagina 6. Este inicio es como recordatorio si me vuelven a surgir estos problemas.

SOLUCION DE PROBLEMAS SURGIDOS EN LA PRACTICA

Cambiando El puerto Oracle

```
SQL> SELECT DBMS_XDB.GETHTTPPORT() FROM DUAL
2 ;

DBMS_XDB.GETHTTPPORT()
-----
8080

SQL> EXEC DBMS_XDB.SETHTTPPORT(9090);
PL/SQL procedure successfully completed.

SQL> SELECT DBMS_XDB.GETHTTPPORT() FROM DUAL
2 ;

DBMS_XDB.GETHTTPPORT()
-----
9090

SQL>
```

Creando Tablas

En el ejercicio anterior lo realice todo mediante consola ya que en el PC host solo me funciona mediante consola, aunque cambien el puerto por algún motivo posiblemente por la configuración del firewall del karpesky no puedo conectar por el servidor web. Por eso tengo que repetir parte de la practica 4 en una maquina virtual.

☐ Autocommit Rows 10 Save Run

```
CREATE TABLE CLIENTES
(
  CODIGO NUMBER(4,0) PRIMARY KEY
, NOMBRE VARCHAR2(30) NOT NULL
, APELLIDOS VARCHAR2(30) NOT NULL
, EDAD NUMBER(2,0) NOT NULL
);
```

Results Explain Describe Saved SQL History

Table created.

0.58 seconds

☐ Autocommit Rows 10 Save Run

```
CREATE TABLE PEDIDOS
(
  NUM NUMBER(5,0) PRIMARY KEY
, FECHA DATE NOT NULL
, GASTOS_ENVIO NUMBER(5,2)
, FECHA_PREVISTA DATE NOT NULL
, TOTAL NUMBER(10,2)
, CLIENTE NUMBER(4,0)
, CONSTRAINT CLIENTES_FK FOREIGN KEY (CLIENTE) REFERENCES CLIENTES (CODIGO)
);
```

Results Explain Describe Saved SQL History

Table created.

0.18 seconds

☐ Autocommit
 Rows
Save Run

```

CREATE TABLE PRODUCTOS
(
    CODIGO NUMBER(5,0) PRIMARY KEY
    , NOMBRE VARCHAR2(30) NOT NULL
    , PRECIO NUMBER(7,2) NOT NULL
);
    
```

Results Explain Describe Saved SQL History

Table created.

0.01 seconds

☐ Autocommit
 Rows
Save Run

```

CREATE TABLE LINEAS
(
    NUM          NUMBER(2,0)
    , NUM_PEDIDO NUMBER(5,0)
    , PRODUCTO   NUMBER(5,0) NOT NULL
    , CANTIDAD   NUMBER(8,0) NOT NULL
    , IMPORTE    NUMBER(6,2)
    , CONSTRAINT DETALLE_PK PRIMARY KEY (NUM, NUM_PEDIDO)
    , CONSTRAINT PEDIDO_FK FOREIGN KEY (NUM_PEDIDO) REFERENCES PEDIDOS (NUM)
    , CONSTRAINT PRODUCTO_FK FOREIGN KEY (PRODUCTO) REFERENCES PRODUCTOS (CODIGO)
);
    
```

Results Explain Describe Saved SQL History

Table created.

0.01 seconds

```

SQL> select TABLE_NAME from user_tables;
TABLE_NAME
-----
DEPT
EMP
DEMO_USERS
DEMO_CUSTOMERS
DEMO_ORDERS
DEMO_PRODUCT_INFO
DEMO_ORDER_ITEMS
DEMO_STATES
APEX$_ACL
APEX$_WS_WEBPG_SECTIONS
APEX$_WS_ROWS
TABLE_NAME
-----
APEX$_WS_HISTORY
APEX$_WS_NOTES
APEX$_WS_LINKS
APEX$_WS_TAGS
APEX$_WS_FILES
APEX$_WS_WEBPG_SECTION_HISTORY
HTMLDB_PLAN_TABLE
CLIENTES
PEDIDOS
PRODUCTOS
LINEAS
22 rows selected.
SQL>
    
```

Insertando datos iniciales

☒ Autocommit
 Rows
Save Run

```

INSERT INTO CLIENTES VALUES ('0001', 'Luis', 'Garcia Perez', 30);
    
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

Los acabo metiendo por consola que es mucho más rapido

```
SQL> INSERT INTO CLIENTES VALUES ('0003', 'Javier', 'Gamez Ualiente', 30);
1 row created.

SQL> INSERT INTO CLIENTES VALUES ('0004', 'Luis Ma', 'Rico Martin', 17);
1 row created.

SQL> INSERT INTO CLIENTES VALUES ('0005', 'Ana Belen', 'Dimas Marco', 15);
1 row created.

SQL> INSERT INTO CLIENTES VALUES ('0006', 'Jose Luis', 'Garcia Sanchez', 50);
1 row created.

SQL> INSERT INTO CLIENTES VALUES ('0007', 'Ma Pilar', 'Perez Bermejo', 45);
1 row created.

SQL> _
```

Errores al introducir la fecha en la tabla pedidos

☒ Autocommit Rows

INSERT INTO PEDIDOS VALUES (4, '25/03/2016', NULL, '20/04/2016', 100, '0002');

Results Explain Describe Saved SQL History

ORA-01843: not a valid month

Motivo ahora trabajo bajo una máquina virtual en configuración inglesa los parámetros de sesión son diferentes a la de host que está configurada en región española, pero la maquina host. Cuando compruebo el formato de fecha es diferente esta configurado a mes/día/año.

```
select * from nls_session_parameters;
```

<input checked="" type="checkbox"/> Autocommit	Rows	100			Save	Run
--	------	-----	--	--	------	-----

```
select * from nls_session_parameters;
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

PARAMETER	VALUE
NLS_LANGUAGE	AMERICAN
NLS_TERRITORY	AMERICA
NLS_CURRENCY	\$
NLS_ISO_CURRENCY	AMERICA
NLS_NUMERIC_CHARACTERS	.,
NLS_CALENDAR	GREGORIAN
NLS_DATE_FORMAT	mm/dd/yyyy
NLS_DATE_LANGUAGE	AMERICAN
NLS_SORT	BINARY
NLS_TIME_FORMAT	HH.MI.SSXF AM
NLS_TIMESTAMP_FORMAT	DD-MON-RR HH.MI.SSXF AM
NLS_TIME_TZ_FORMAT	HH.MI.SSXF AM TZR
NLS_TIMESTAMP_TZ_FORMAT	DD-MON-RR HH.MI.SSXF AM TZR
NLS_DUAL_CURRENCY	\$
NLS_COMP	BINARY
NLS_LENGTH_SEMANTICS	BYTE
NLS_NCHAR_CONV_EXCP	FALSE

17 rows returned in 0.00 seconds [Download](#)

<input checked="" type="checkbox"/> Autocommit	Rows	100			Save	Run
--	------	-----	--	--	------	-----

```
select sysdate from dual;
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

SYSDATE
02/19/2019

Soluciones o nos ajustamos a formato que tenemos establecido modificando las sentencias entregadas en la tarea 4.1 o realizamos una modificación en NLS_DATE_FORMAT. Vamos a modificar la sentencia correspondiente para que no se tengan que modificar las sentencias.

```
alter session set NLS_DATE_FORMAT = 'DD/MM/RR';
```

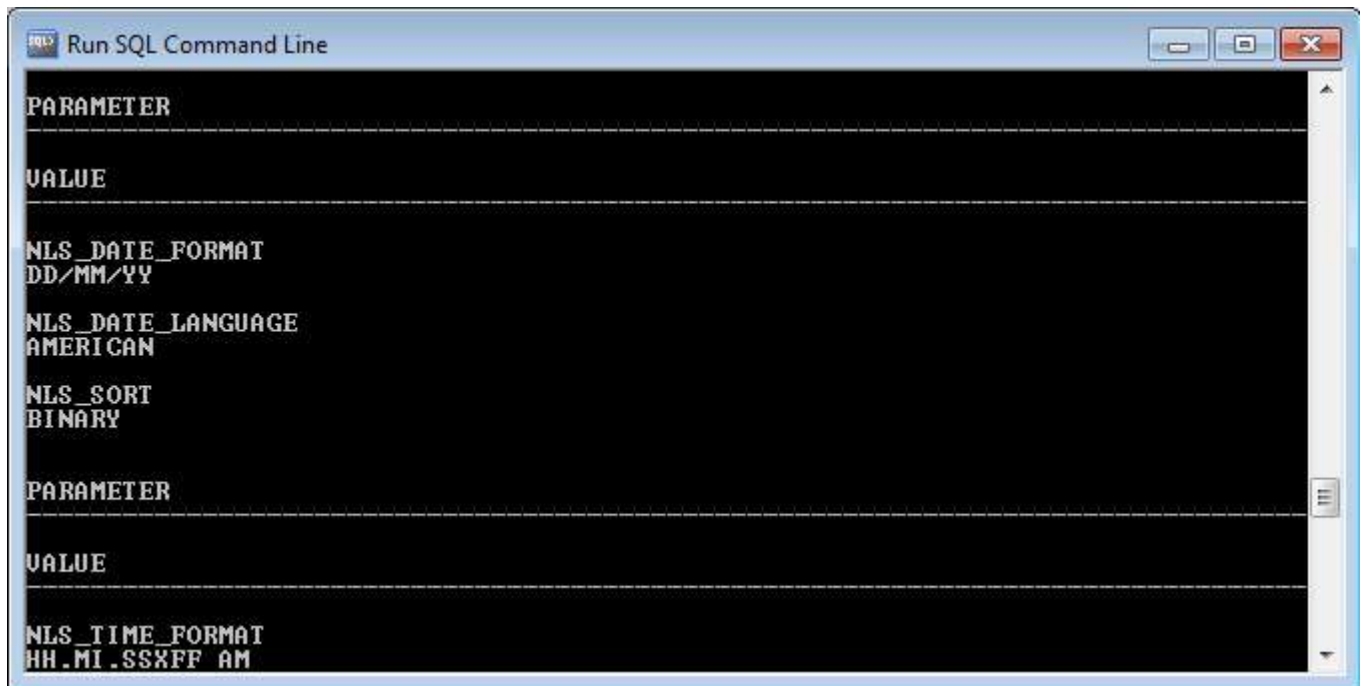
Modificamos la sesión, cada vez que cerremos sesión se vuelve a cambiar

```
SQL> alter session set NLS_DATE_FORMAT = 'DD/MM/YY';
Session altered.
```

Ahora si queremos hacerlo fijo tendremos que conectar con una cuenta con privilegios

```
alter session set NLS_DATE_FORMAT = 'DD/MM/RR' scope = spfile;
```

```
SQL> alter system set NLS_DATE_FORMAT = 'DD/MM/RR' scope = spfile;
System altered.
```



```
SQL> SELECT SYSDATE FROM DUAL;  
SYSDATE  
-----  
19/02/19  
SQL>
```

APARTADO A

Añadiendo nuevo producto

Table

Data

Indexes

Model

Constraints

Grants

Statistics

Query | Count Rows | Insert Row

EDIT	CODIGO	NOMBRE	PRECIO
	50001	GORRO	10
	10001	PANTALON	50
	10002	PANTALON PITILLO	60
	10003	PANTALON CAMPANA	55
	20001	CAMISA M/L	65
	20002	CAMISA M/C	45
	30001	VESTIDO C	80
	30002	VESTIDO L	80
	40001	FALDA LARGA	50
	40002	FALDA CORTA	45
	40003	FALDA MINI	40

row(s) 1 - 11 of 11

Create Row

Table: PRODUCTOS

*Codigo

0012

*Nombre

SOMBRERO

*Precio

5

x

Query | Count Rows | Insert Row

EDIT	CODIGO	NOMBRE	PRECIO
	50001	GORRO	10
	12	SOMBRERO	5
	10001	PANTALON	50
	10002	PANTALON PITILLO	60
	10003	PANTALON CAMPANA	55
	20001	CAMISA M/L	65
	20002	CAMISA M/C	45
	30001	VESTIDO C	80
	30002	VESTIDO L	80
	40001	FALDA LARGA	50
	40002	FALDA CORTA	45
	40003	FALDA MINI	40

row(s) 1 - 12 of 12

Mediante sql commander

```
INSERT INTO PRODUCTOS VALUES (50001, 'GORRO', 10);
```

☒ Autocommit

Rows

10

Save

Run

INSERT INTO PRODUCTOS VALUES (50001, 'GORRO', 10);

Results | Explain | Describe | Saved SQL | History

CODIGO	NOMBRE	PRECIO
10001	PANTALON	50
10002	PANTALON PITILLO	60
10003	PANTALON CAMPANA	55
20001	CAMISA M/L	65
20002	CAMISA M/C	45
30001	VESTIDO C	80
30002	VESTIDO L	90
40001	FALDA LARGA	50
40002	FALDA CORTA	45
40003	FALDA MINI	40

10 rows returned in 0.00 seconds

Download

Rebajando producto

Query Count Rows Insert Row

EDIT	CODIGO	NOMBRE	PRECIO
	50001	GORRO	10
	12	SOMBRERO	5
	10001	PANTALON	50
	10002	PANTALON PITILLO	60
	10003	PANTALON CAMPANA	55
	20001	CAMISA M/L	65
	20002	CAMISA M/C	45
	30001	VESTIDO C	80
	30002	VESTIDO L	90
	40001	FALDA LARGA	50
	40002	FALDA CORTA	45
	40003	FALDA MINI	40

row(s) 1 - 12 of 12

Download

EDIT	CODIGO	NOMBRE	PRECIO
	50001	GORRO	10
	12	SOMBRERO	5
	10001	PANTALON	50
	10002	PANTALON PITILLO	60
	10003	PANTALON CAMPANA	55
	20001	CAMISA M/L	65
	20002	CAMISA M/C	45
	30001	VESTIDO C	80
	30002	VESTIDO L	80
	40001	FALDA LARGA	50
	40002	FALDA CORTA	45
	40003	FALDA MINI	40

row(s) 1 - 12 of 12

Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL

Cancel Delete Apply Changes

Edit Row

Table: PRODUCTOS

Codigo: 30002

Nombre: VESTIDO L

Precio: 90

Table Information

Mediante sql commander

```
update productos set precio=precio-10 where productos.nombre = (select nombre from productos where precio in (select max(precio) from productos));
```

☒ Autocommit Rows 10 Save Run

```
update productos set precio=precio-10 where productos.nombre = (select nombre from productos where precio in (select max(precio) from productos));
```

Results Explain Describe Saved SQL History

1 row(s) updated.

O manualmente seleccionado uno mismo el producto más caro

```
update productos set precio=80 where nombre='VESTIDO L';
```

☒ Autocommit Rows 10 Save Run

```
update productos set precio=80 where nombre='VESTIDO L';
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.01 seconds

APARTADO B

Insertando un nuevo cliente

```
INSERT INTO CLIENTES VALUES ('0008', 'Maria Pilar', 'Quiles Carrera', 25);
```

Home > SQL Workshop > SQL Commands

☐ Autocommit Rows: 10   Save Run

```
INSERT INTO CLIENTES VALUES ('0008', 'Maria Pilar', 'Quiles Carrera', 25);
```



Select * from clientes

Results Explain Describe Saved SQL History

CODIGO	NOMBRE	APELLIDOS	EDAD
1	Luis	Garcia Perez	30
2	Maria	Lopez Garrido	50
3	Javier	Gamez Valiente	30
4	Luis Mj	Rico Martin	17
5	Ana Belen	Dimas Marco	15
6	Jose Luis	Garcia Sanchez	50
7	Mj Pilar	Perez Bermejo	45
8	Maria Pilar	Quiles Carrera	25

Insertar un pedido al cliente anterior

```
INSERT INTO PEDIDOS VALUES (11, '19/02/2019', 0, '11/03/2019', 0, '0008');
```

☐ Autocommit Rows: 10   Save Run

```
INSERT INTO PEDIDOS VALUES (11, '19/02/2019', 0, '11/03/2019', 0, '0008');
```

NUM	FECHA	GASTOS_ENVIO	FECHA_PREVISTA	TOTAL	CLIENTE
1	10/10/15	-	10/11/15	310	1
2	10/02/16	5	10/03/16	185	1
3	20/02/16	3	20/04/16	180	1
4	25/03/16	-	20/04/16	100	2
5	25/03/16	-	20/05/16	135	3
6	15/04/16	-	20/05/16	45	4
7	15/04/16	-	20/05/16	45	5
8	15/05/16	10	20/06/16	45	6
9	15/07/16	10	20/09/16	85	7
10	15/01/17	10	15/02/17	90	7
11	19/02/19	0	11/03/19	0	8

11 rows returned in 0.01 seconds

Insertar primera línea al pedido

```
INSERT INTO LINEAS VALUES (1, 11, 10001, 3, (select precio from productos where codigo  
= '10001') * 3);
```


☐ Autocommit
 Rows 15
✎ 🔍
Save Run

```
INSERT INTO LINEAS VALUES (1,11,10001, 3, (select precio from productos where codigo ='10001')*3);
```

Insertar segunda línea al pedido

```
INSERT INTO LINEAS VALUES (2,11,10002, 2, (select precio from productos where codigo ='10002')*2);
```

☐ Autocommit
 Rows 15
✎ 🔍
Save Run

```
INSERT INTO LINEAS VALUES (2,11,10002, 2, (select precio from productos where codigo ='10002')*2)
```

NUM	NUM_PEDIDO	PRODUCTO	CANTIDAD	IMPORTE
1	1	10001	2	100
2	1	30001	1	80
3	1	20001	2	130
1	2	20001	1	65
2	2	40003	3	120
1	3	40002	2	180
1	4	10001	2	100
1	5	20002	2	90
2	5	40002	1	45
1	6	40002	1	45
1	7	40002	1	45
1	8	40002	1	45
1	9	40003	1	40
2	9	20002	1	45
1	10	20002	2	90
1	11	10001	3	150
2	11	10002	2	120

17 rows returned in 0.01 seconds

Actualización del pedido para aplicar el precio

```
update pedidos set total = pedidos.gastos_envio + (select sum(lineas.importe) from lineas where lineas.num_pedido=11)where num =11;
```

☐ Autocommit
 Rows 30
✎ 🔍
Save Run

```
update pedidos set total = pedidos.gastos_envio + (select sum(lineas.importe) from lineas where lineas.num_pedido=11)where num =11;
```

NUM	FECHA	GASTOS_ENVIO	FECHA_PREVISTA	TOTAL	CLIENTE
1	10/10/15	-	10/11/15	310	1
2	10/02/16	5	10/03/16	185	1
3	20/02/16	3	20/04/16	180	1
4	25/03/16	-	20/04/16	100	2
5	25/03/16	-	20/05/16	135	3
6	15/04/16	-	20/05/16	45	4
7	15/04/16	-	20/05/16	45	5
8	15/05/16	10	20/06/16	45	6
9	15/07/16	10	20/09/16	85	7
10	15/01/17	10	15/02/17	90	7
11	19/02/19	0	11/03/19	270	8

Mostrar líneas del pedido 11

```
select * from lineas where num_pedido = 11;
```

<input checked="" type="checkbox"/> Autocommit	Rows	10			Save	Run
select * from <u>lineas</u> where <u>num_pedido</u> = 11;						
Results Explain Describe Saved SQL History						
NUM	NUM_PEDIDO	PRODUCTO	CANTIDAD	IMPORTE		
1	11	10001	3	150		
2	11	10002	2	120		

Retrasar pedidos de febrero

Con la siguiente sentencia cogemos el registro de mes de febrero del año actual no tiene sentido cambiar la fecha a los de febrero de otros años.

La sentencia de selección

```
select * from pedidos where to_char(fecha,'MM') = '02' and to_char(fecha,'YY') = (select
to_char(sysdate,'YY') from dual);
```



La sentencia de actualización

```
update pedidos set FECHA_PREVISTA = FECHA +30 where to_char(fecha,'MM') = '02' and
to_char(fecha,'YY') = (select to_char(sysdate,'YY') from dual);
```

<input type="checkbox"/> Autocommit	Rows	20			Save	Run
update <u>pedidos</u> set <u>FECHA_PREVISTA</u> = <u>FECHA</u> +30 where <u>to_char(fecha,'MM')</u> = '02' and <u>to_char(fecha,'YY')</u> = (select <u>to_char(sysdate,'YY')</u> from dual);						

NUM	FECHA	GASTOS_ENVIO	FECHA_PREVISTA	TOTAL	CLIENTE
11	19/02/19	0	21/03/19	270	8

Todos los pedidos sin gasto de envío poner 0

☐ Autocommit Rows   Save Run



```
update pedidos set gastos_envio = 0 where gastos_envio is null;
```

5 row(s) updated.

Results	Explain	Describe	Saved SQL	History	
NUM	FECHA	GASTOS_ENVIO	FECHA_PREVISTA	TOTAL	CLIENTE
1	10/10/2015	0	11/10/2015	310	1
2	02/10/2016	5	03/10/2016	185	1
3	02/20/2016	3	04/20/2016	180	1
4	03/25/2016	0	04/20/2016	100	2
5	03/25/2016	0	05/20/2016	135	3
6	04/15/2016	0	05/20/2016	45	4
7	04/15/2016	0	05/20/2016	45	5
8	05/15/2016	10	06/20/2016	45	6
9	07/15/2016	10	09/20/2016	85	7
10	01/15/2017	10	02/15/2017	90	7
11	02/19/2019	0	03/11/2019	270	8

Subir el precio de todos los productos un 10 %

```
update productos set precio=(precio * 1.10)
```



☐ Autocommit Rows   Save Run

```
update productos set precio=(precio * 1.10)
```

CODIGO	NOMBRE	PRECIO
50001	GORRO	11
10001	PANTALON	55
10002	PANTALON PITILLO	66
10003	PANTALON CAMPANA	60.5
20001	CAMISA M/L	71.5
20002	CAMISA M/C	49.5
30001	VESTIDO C	88
30002	VESTIDO L	88
40001	FALDA LARGA	55
40002	FALDA CORTA	49.5
40003	FALDA MINI	44

Si hago un rollback que cambios se desharian

Se deshace todo hasta el último commit. Es decir, hasta donde se mostró las líneas del pedido 17

☐ Autocommit Rows   Save Run

```
rollback
```

Results Explain Describe Saved SQL History

Statement processed.

Borra todos los pedidos nunca seleccionados

```
delete from productos where NOT (CODIGO IN (SELECT PRODUCTO FROM LINEAS));
```

☐ Autocommit Rows 20   Save Run

```
delete from productos where NOT (CODIGO IN (SELECT PRODUCTO FROM LINEAS));
```

Results Explain Describe Saved SQL History

4 row(s) deleted.

Results Explain Describe Saved SQL History

CODIGO	NOMBRE	PRECIO
10001	PANTALON	50
10002	PANTALON PITILLO	60
20001	CAMISA M/L	65
20002	CAMISA M/C	45
30001	VESTIDO C	80
40002	FALDA CORTA	45
40003	FALDA MINI	40



7 rows returned in 0.00 seconds

Borra el producto 10001

No se puede borrar así tan fácil al ser una forent key en la tabla línea y además esta referenciada varias veces primero tenemos que quitar esos punteros en la tabla línea. Además, no puede ser nulo en la tabla líneas por lo que abría que borrar las líneas y si se observa el pedido 4 solo dispone una línea este pedido quedaría como que se realizo pero no si se busca la línea que lo conforma no existirá porque lo habremos tenido que eliminar.

Sentencias para borrar 10001



```
delete from lineas where producto = '10001';
```

☐ Autocommit Rows 20   Save Run

```
delete from lineas where producto = '10001';|
```

3 row(s) deleted.

```
delete from productos where codigo = '10001';
```

☐ Autocommit Rows 20   Save Run

```
delete from productos where codigo = '10001';|
```

Results Explain Describe Saved SQL History

1 row(s) deleted.

NUM	NUM_PEDIDO	PRODUCTO	CANTIDAD	IMPORTE
2	1	30001	1	80
3	1	20001	2	130
1	2	20001	1	65
2	2	40003	3	120
1	3	40002	2	180
1	5	20002	2	90
2	5	40002	1	45
1	6	40002	1	45
1	7	40002	1	45
1	8	40002	1	45
1	9	40003	1	40
2	9	20002	1	45
1	10	20002	2	90
2	11	10002	2	120

CODIGO	NOMBRE	PRECIO
10002	PANTALON PITILLO	60
20001	CAMISA M/L	65
20002	CAMISA M/C	45
30001	VESTIDO C	80
40002	FALDA CORTA	45
40003	FALDA MINI	40

☐ Autocommit

Rows

20



Save

Run

COMMIT;|

Results

Explain

Describe

Saved SQL

History