

Mario Rubio Avila BD Tarea 5.1

APARTADO A

1ºAñadiendo 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			

2º 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

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

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

APARTADO B

3º 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

4º Insertar un pedido al cliente anterior

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

☐ Autocommit Rows  



```
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

5º Insertar primera línea al pedido



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

☐ Autocommit Rows  

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

6º Insertar segunda línea al pedido

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

☐ Autocommit Rows  

```
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

7º 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

```
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

8º 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		

9º 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

10º Todos los pedidos sin gasto de envío poner 0



<input type="checkbox"/> Autocommit	Rows	10			Save	Run
update <u>pedidos</u> set <u>gastos envio</u> = 0 where <u>gastos envio</u> 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

11º Subir el precio de todos los productos un 10 %

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



☐ Autocommit Rows  

```
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

12º 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  


```
rollback
```

Results Explain Describe Saved SQL History

Statement processed.

13º Borra todos los pedidos nunca seleccionados

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

☐ Autocommit Rows  

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

Results Explain Describe Saved SQL History

4 row(s) deleted.

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

14º 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 líneas. 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.

15º Sentencias para borrar 10001

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

☐ Autocommit
 Rows
Save Run

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

3 row(s) deleted.

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

☐ Autocommit
 Rows
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