

# PRÁCTICA 6 BASES DE DATOS

Javier Guzmán Muñoz y Jorge Villarrubia Elvira

**Apartado 1:** Disparador por tabla.

a) Creación de las tablas:

```
create table pedidos(código char(6),
                    fecha char(10),
                    importe number(6,2),
                    cliente char(20),
                    notas char(1024),
                    primary key(código));

create table contiene(pedido char(6),
                    plato char(20),
                    precio number(6,2),
                    unidades number(2,0),
                    primary key(pedido, plato),
                    foreign key (pedido) references
                        pedidos(código) on delete cascade );

create table auditoría(operación char(6),
                    tabla char(50),
                    fecha char(10),
                    hora char(8));
```

Salida por consola:

Table PEDIDOS creado.

Table CONTIENE creado.

Table AUDITORÍA creado.

b) Disparador sobre pedidos:

```
create or replace trigger trigger_pedidos
after insert or update or delete on pedidos
begin
    if deleting then
        insert into auditoría
            values ('DELETE', 'pedidos',
                to_char(sysdate,'dd/mm/yyyy'),
                to_char(sysdate,'hh:mi:ss'));
    elsif inserting then
        insert into auditoría
            values ('INSERT', 'pedidos',
                to_char(sysdate,'dd/mm/yyyy'),
                to_char(sysdate,'hh:mi:ss'));
    elsif updating then
        insert into auditoría
            values ('UPADTE', 'pedidos',
                to_char(sysdate,'dd/mm/yyyy'),
                to_char(sysdate,'hh:mi:ss'));
    end if;
end;
```

Salida por consola:

```
Trigger TRIGGER_PEDIDOS compilado
```

```
insert into pedidos values('01', '01/01/2015', 10.1, 'Pepe',
                        'Jajja lol xd');

update pedidos set cliente = 'Juan';

delete from pedidos where código = '01';
```

Salida por consola:

```
1 fila insertadas.
1 fila actualizadas.
1 fila eliminado
```

```
select * from auditoría;
```

Resultado de la consulta:

	OPERACIÓN	TABLA	FECHA	HORA
1	INSERT	pedidos	23/11/2018	01:06:22
2	UPADTE	pedidos	23/11/2018	01:06:22
3	DELETE	pedidos	23/11/2018	01:06:22

## **Apartado 2:** Disparador por fila

```
create or replace trigger trigger_contiene
after insert or delete or update on contiene
for each row

begin
    if inserting then
        update pedidos set importe = importe + :new.precio *
:new.unidades
        where código = :new.pedido;
    elsif updating then
        update pedidos set importe = importe - :old.precio *
:old.unidades
        where código = :old.pedido;
        update pedidos set importe = importe + :new.precio *
:new.unidades
        where código = :new.pedido;
    elsif deleting then
        update pedidos set importe = importe - :old.precio *
:old.unidades
        where código = :old.pedido;
    end if;

end;
```

Modificaciones en la tabla pedidos:

```
insert into pedidos values ('01', '07/03/2013', 1, 'Javi',
'holaa');

insert into pedidos values ('02', '07/03/2013', 0, 'Javi',
'holaa');

insert into pedidos values ('03', '07/03/2013', 0, 'Javi',
'holaa');

insert into contiene values('01', 'Pasta', 12.54, 3);
update contiene set pedido = '02' where pedido = '01';
insert into contiene values('03', 'Chorizo', 2.34, 65);
delete from contiene where pedido = '03';
```

```
select * from pedidos;
```

	CÓDIGO	FECHA	IMPORTE	CLIENTE	NOTAS
1	01	07/03/2013	1	Javi	holaa
2	02	07/03/2013	37,62	Javi	holaa
3	03	07/03/2013	0	Javi	holaa

```
select * from contiene;
```

	PEDIDO	PLATO	PRECIO	UNIDADES
1	02	Pasta	12,54	3

```
select * from auditoría;
```

	OPERACIÓN	TABLA	FECHA	HORA
1	INSERT	pedidos	23/11/2018	01:41:40
2	UPADTE	pedidos	23/11/2018	01:41:40
3	DELETE	pedidos	23/11/2018	01:41:40
4	INSERT	pedidos	23/11/2018	01:45:41
5	INSERT	pedidos	23/11/2018	01:45:41
6	INSERT	pedidos	23/11/2018	01:45:41
7	UPADTE	pedidos	23/11/2018	01:45:41
8	UPADTE	pedidos	23/11/2018	01:45:41
9	UPADTE	pedidos	23/11/2018	01:45:41
10	UPADTE	pedidos	23/11/2018	01:45:41
11	UPADTE	pedidos	23/11/2018	01:45:41

### Apartado 3: Creación y uso de índices

Creación del índice:

```
create index index_pedidos on pedidos(cliente);
```

Salida por consola:

```
Index INDEX_PEDIDOS creado.
```

```
declare

begin
for i in 1..500000
loop
    insert into pedidos values(i , '06/01/2015', 10.0, 'C' ||
                                i , ' ');
end loop;

end;
```

SALIDA ERRÓNEA:

Error que empieza en la línea: 83 del comando :

```
declare

begin
for i in 1..500000
loop
    insert into pedidos values(i , '06/01/2015', 10.0, 'C' || i
, ' ');
end loop;

end;

Informe de error -

ORA-01653: no se ha podido ampliar la tabla DG09.PEDIDOS con
1024 en el tablespace EMPRESADG09
```

ORA-06512: en línea 6

01653. 00000 - "unable to extend table %s.%s by %s in  
tablespace %s"

\*Cause: Failed to allocate an extent of the required number  
of blocks for

a table segment in the tablespace indicated.

\*Action: Use ALTER TABLESPACE ADD DATAFILE statement to add  
one or more

files to the tablespace indicated.

Nos asignamos más espacio en el servidor:

```
ALTER DATABASE DATAFILE 'D:\oracle\EMPRESADG09' AUTOEXTEND ON  
MAXSIZE 1G;
```

Database alterado.

Aunque aquí aparezca que se nos ha asignado más espacio esto no es real, así que tenemos que reducir el número de tuplas a insertar de 500000 a 50000.

Veamos que se han insertado todas correctamente:

```
SELECT * FROM pedidos order by to_number(código);
```

	CÓDIGO	FECHA	IMPORTE	CLIENTE	NOTAS
1	1	06/01/2015	10	C1	
2	01	07/03/2013	1	Javi	holaa
3	2	06/01/2015	10	C2	
4	02	07/03/2013	37,62	Javi	holaa
5	3	06/01/2015	10	C3	
6	03	07/03/2013	0	Javi	holaa
7	4	06/01/2015	10	C4	
8	5	06/01/2015	10	C5	
9	6	06/01/2015	10	C6	
10	7	06/01/2015	10	C7	
11	8	06/01/2015	10	C8	
12	9	06/01/2015	10	C9	
13	10	06/01/2015	10	C10	
14	11	06/01/2015	10	C11	
15	12	06/01/2015	10	C12	
16	13	06/01/2015	10	C13	
17	14	06/01/2015	10	C14	
18	15	06/01/2015	10	C15	
19	16	06/01/2015	10	C16	
20	17	06/01/2015	10	C17	
21	18	06/01/2015	10	C18	
22	19	06/01/2015	10	C19	
23	20	06/01/2015	10	C20	
24	21	06/01/2015	10	C21	
25	22	06/01/2015	10	C22	
26	23	06/01/2015	10	C23	
27	24	06/01/2015	10	C24	
28	25	06/01/2015	10	C25	
29	26	06/01/2015	10	C26	
30	27	06/01/2015	10	C27	
31	28	06/01/2015	10	C28	
32	29	06/01/2015	10	C29	
33	30	06/01/2015	10	C30	
34	31	06/01/2015	10	C31	

```
SELECT count(*) FROM pedidos;
```

	COUNT(*)
1	50003

Comparemos los tiempos de ejecución de una consulta con y sin índice

```
SET TIMING ON;
```

```
select * from pedidos where cliente = 'C50000';
```

```
>>Query Run In:Resultado de la Consulta 2
```

```
Transcurrido: 00:00:00.352
```

Borramos el índice:

```
drop index index_pedidos;  
Index INDEX_PEDIDOS borrado.
```

```
>>Query Run In:Resultado de la Consulta  
Transcurrido: 00:00:00.346
```

No hemos podido ampliar más espacio, así que no se aprecia una diferencia significativa, es más, observamos que se tarda menos tiempo sin índice, en contra de lo que debería pasar.

#### **Creación de índices sobre vistas:**

**No** se pueden crear índices sobre vistas:

```
create view v1 as select * from contiene;  
  
create index index_contiene on v1(pedido);
```

Salida por consola:

```
View V1 creado.
```

```
Error que empieza en la línea: 106 del comando :
```

```
create index index_contiene on v1(pedido)
```

```
Informe de error -
```

```
ORA-01702: una vista no es apropiada aquí
```

```
01702. 00000 - "a view is not appropriate here"
```

```
*Cause: Among other possible causes, this message will be  
produced if an
```

```
attempt was made to define an Editioning View over a  
view.
```

```
*Action: An Editioning View may only be created over a base  
table.
```



**Sí** se pueden sobre vistas materializadas:

```
create materialized view v2 as select * from contiene;  
  
create index index_contiene on v2(pedido);
```

Salida por consola:

```
Materialized view VISTA2 creado.  
Index "index_pedidos2" creado
```

Sobre la vista normal no se puede, sobre la materializada sí porque esta, a diferencia de la otra, sí que almacena físicamente los datos.