Consultas Básicas SQL

Segunda parte

CLAVE PRIMARIA

Una clave primaria es un campo (o varios) que identifica un solo registro (fila) en una tabla.
Para un valor del campo clave existe solamente un registro.
Al menos hay una clave, la formada por todos los atributos.

Por ejemplo: En la entidad CLIENTES podríamos encontrar las siguientes claves: él código de cliente, el conjunto apellidos + nombre, el DNI, el nombre, apellidos y la dirección, etc.

Clave implica que sólo aparecerá un objeto para cada valor de la clave.

Sólo una vez y nunca más

El sistema se encargará de impedirnos que podamos meter dos valores de clave iguales

De un campo

```
create table NOMBRETABLA
(
  CAMPO TIPO primary key,
  CAMPO TIPO,
  ...
);
```

Con varios campos

```
create table NOMBRETABLA
(
  CAMPO TIPO,
  ...
  primary key (NOMBRECAMPO, NOMBREDECAMPO,…)
);
```

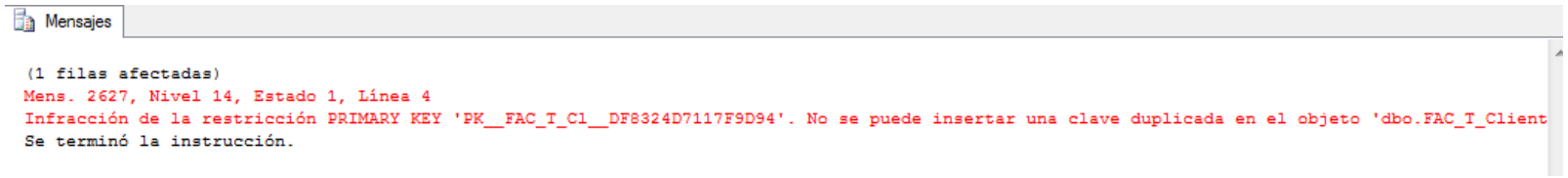
Ejemplo

```
use facturasbasicas
go
-- crear tabla FAC_T_Articulo
if object_id('FAC_T_Articulo') is not null
  drop table FAC_T_Articulo;
  go
create table FAC_T_Articulo
(
   CodArticulo    integer primary key,
   NombreArticulo varchar(50),
   PrecioActual   numeric(10,2)
);
go
```

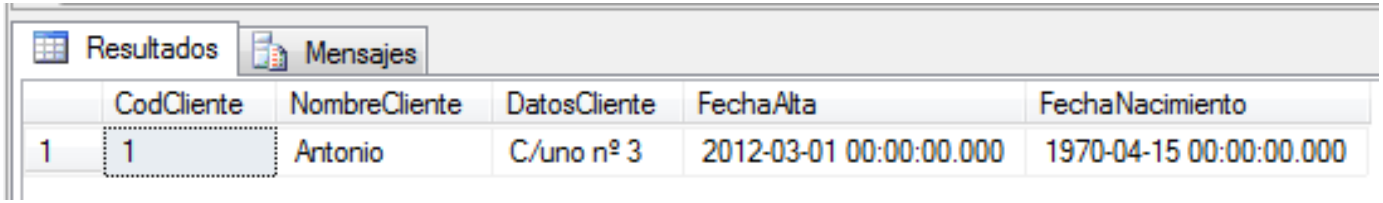
```
-- crear tabla FAC_T_Cliente
if object_id('FAC_T_Cliente') is not null
  drop table FAC_T_Cliente;
  go
create table FAC_T_Cliente
(
   CodCliente      integer primary key,
   NombreCliente   varchar(60),
   DatosCliente    varchar(60),
   FechaAlta       datetime ,
   FechaNacimiento   datetime
);
go
```

Si intentamos entrar dos valores con la misma clave primaria …

```
insert into FAC_T_Cliente
( CodCliente,NombreCliente,DatosCliente,FechaAlta,FechaNacimiento )
values ( 1,'Antonio','C/uno nº 3','01/03/2012','15/04/1970')
insert into FAC_T_Cliente
( CodCliente,NombreCliente,DatosCliente,FechaAlta,FechaNacimiento )
values ( 1,'Juan','C/la hornera nº 7' ,'22/05/2012','29/06/1982' )
go
select CodCliente,NombreCliente,DatosCliente,FechaAlta,FechaNacimiento
from FAC_T_Cliente
```



Sólo se almacenará el primer insert



Ejercicio:

En la Base de datos MoviesBasicas

- Cambiar las instrucciones que crean la tabla Peliculas añadiéndole como clave principal id.
- Igualmente con la tabla socios y la columna NIFNIE.
- Insertar dos registros a cada tabla con el mismo valor de la clave y anotar lo que ocurre.
- Crear otra tabla Socios2 con la misma estructura que socios y clave principal los campos Apellidos y nombre
- Insertar registros de manera que permita ver el comportamiento. (Nombres iguales pero apellidos distintos, apellidos iguales y nombres distintos y finalizando con apellidos y nombre iguales).

CAMPO CON ATRIBUTO IDENTITY

Un campo numérico puede tener un atributo extra "**identity**". Los valores de un campo con este atributo generan valores secuenciales que se inician en 1 y se incrementan en 1 automáticamente. Es el Sistema el que se encarga de generarlo.

Pero NO podemos asegurar que no deje huecos en la numeración.

Se utiliza generalmente en campos correspondientes a códigos de identificación para generar valores únicos para cada nuevo registro que se inserta. Siendo por tanto una primary key adecuada para cuando no tenemos otra válida

No se debe mostrar al usuario, debe ser para uso interno del programa. Será difícil de explicar que no reutilice los huecos y que no parta de uno al vaciar la tabla de datos.

```
 create table NOMBRETABLA
(
  CAMPO int identity,
  CAMPO TIPO,
  ...
);
```

Usándolo como clave primaria

```
 create table NOMBRETABLA
(
  CAMPO int identity primary key,
  CAMPO TIPO,
  ...
);
```

Un campo "identity" no es editable, es decir, no se puede ingresar un valor ni actualizarlo.
Sólo se genera automáticamente.
No puede ser NULL por lo mismo.

Suele generar el siguiente en secuencia al último, pero no recuperará huecos anteriores.

Incluso si generamos el 5 y lo borramos no recuperará el 5 sino que saltará al 6.
Si teníamos hasta el 10 y borramos todos los registros pero no la tabla generará valores a partir del 11.

Ejemplo

```
use facturasbasicas
go
create table FAC_T_Cliente2
(
    CodCliente  integer identity primary key,
    NombreCliente   varchar(60),
    DatosCliente    varchar(60),
    FechaAlta   datetime ,
    FechaNacimiento datetime
);
go
```

Insertar registros sin especificar el campo identity

```
insert into FAC_T_Cliente2
( NombreCliente,DatosCliente,FechaAlta,FechaNacimiento )
 values ('Antonio','C/uno nº 3','01/03/2012','15/04/1970')
insert into FAC_T_Cliente2
( NombreCliente,DatosCliente,FechaAlta,FechaNacimiento )
 values ('Juan','C/la hornera nº 7' ,'22/05/2012','29/06/1982' )
insert into FAC_T_Cliente2
( NombreCliente,DatosCliente,FechaAlta,FechaNacimiento )
 values ('María','C/el pino nº 7','22/05/2010','15/06/1960')
 go
```

Viendo el resultado…

```
select CodCliente, NombreCliente
 from Fac_T_Cliente2
```

Resultados | Mensajes

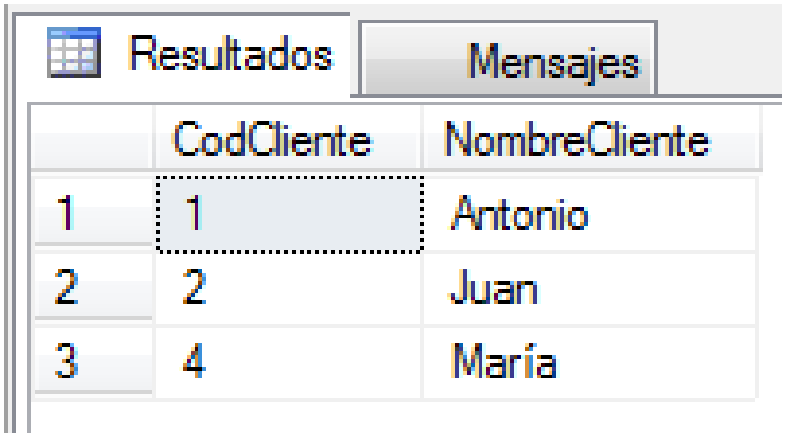
	CodCliente	NombreCliente
1	1	Antonio
2	2	Juan
3	3	María

Generó claves consecutivas

No podemos asegurar esa continuidad de las claves…

```
delete from Fac_T_Cliente2
where CodCliente=3
go
insert into FAC_T_Cliente2
( NombreCliente,DatosCliente,FechaAlta,FechaNacimiento )
 values ('María','C/el pino nº 7','22/05/2010','15/06/1960')
 go

 select CodCliente, NombreCliente
 from Fac_T_Cliente2
go
```

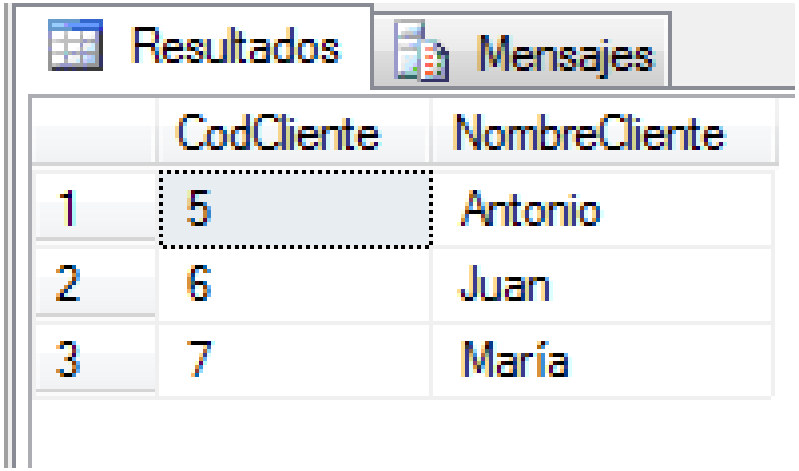


	CodCliente	NombreCliente
1	1	Antonio
2	2	Juan
3	4	María

Borrando todos los registros tampoco conseguimos reiniciar la numeración

```
delete from Fac_T_Cliente2
go
insert into FAC_T_Cliente2
( NombreCliente,DatosCliente,FechaAlta,FechaNacimiento )
 values ('Antonio','C/uno nº 3','01/03/2012','15/04/1970')
insert into FAC_T_Cliente2
( NombreCliente,DatosCliente,FechaAlta,FechaNacimiento )
 values ('Juan','C/la hornera nº 7' ,'22/05/2012','29/06/1982' )
insert into FAC_T_Cliente2
( NombreCliente,DatosCliente,FechaAlta,FechaNacimiento )
 values ('María','C/el pino nº 7','22/05/2010','15/06/1960')
 go

 select CodCliente, NombreCliente
 from Fac_T_Cliente2
go
```



	CodCliente	NombreCliente
1	5	Antonio
2	6	Juan
3	7	María

El atributo "identity" permite indicar el valor de inicio de la secuencia y el incremento, para ello usamos la siguiente sintaxis:

identity (inicial,incremento)

Inicial será el primer número e incremento el salto al siguiente que genere. No se suele usar salvo que se quieran mezclar tablas, aunque para esto hay soluciones mejores.

Hay funciones que nos devuelven valores relacionados con el campo.

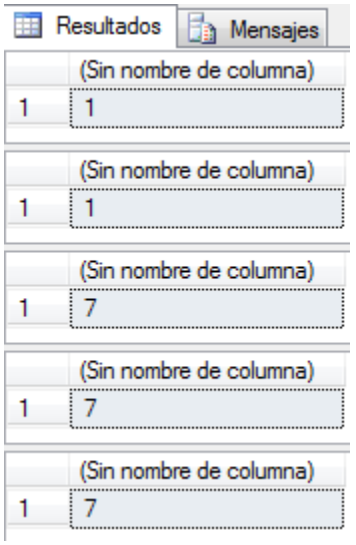
select ident_seed('tabla')
devuelve el valor inicial del generador identity de la tabla.
select ident_incr('tabla')
devuelve el incremento del generador identity de la tabla.

select SCOPE_IDENTITY()
select @@identity
devuelven el último valor generado del anterior insert con alguna diferencia en el ámbito

select IDENT_CURRENT('tabla')
devuelve el último valor generado para la tabla especificada

Ejemplo

```
select ident_seed('Fac_T_Cliente2')
--devuelve el valor inicial del generador identity de la tabla.
select ident_incr('Fac_T_Cliente2')
--devuelve el incremento del generador identity de la tabla.
select SCOPE_IDENTITY()
select @@identity
--devuelven el último valor generado del anterior insert con
--alguna diferencia en el ámbito
select IDENT_CURRENT( 'Fac_T_Cliente2' )
--devuelve el último valor generado para la tabla especificada
```



Hemos visto que en un campo declarado "identity" no puede ingresarse explícitamente un valor.

Para permitir ingresar un valor en un campo de identidad se debe activar la opción "identity_insert":

```
set identity_insert NombreTabla on;
```

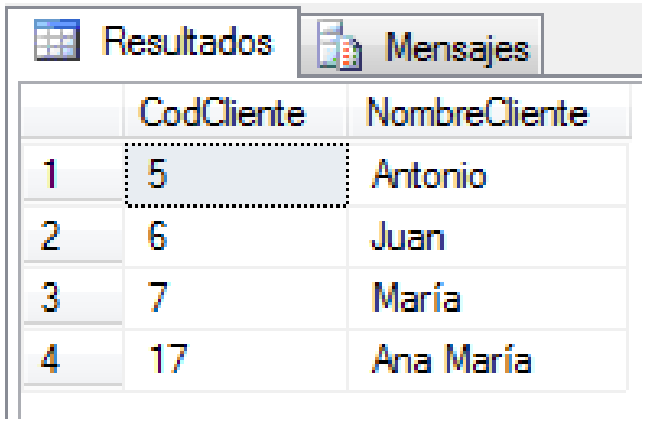
Es decir, podemos ingresar valor en un campo "identity" seteando la opción "identity_insert" en "on".

Cuando "identity_insert" está en ON, las instrucciones "insert" deben especificar un valor.

Ejemplo

```
set identity_insert FAC_T_Cliente2 on;
go
insert into FAC_T_Cliente2
( CodCliente,NombreCliente,DatosCliente,FechaAlta,FechaNacimiento )
 values (17,'Ana María','C/el pino nº 7','22/05/2010','15/06/1960')
 go
 select CodCliente, NombreCliente
 from Fac_T_Cliente2
go
```

Al poner identity_insert a on para la tabla tendremos que especificar el valor del campo identity en los insert. Ya que el Sistema no lo genera automáticamente.



	CodCliente	NombreCliente
1	5	Antonio
2	6	Juan
3	7	María
4	17	Ana María

Ejercicio:

En la base de datos MoviesBasicas.
Crear una tabla Peliculas2 con la misma estructura que Peliculas y colocando el campo Id como clave primaria autogenerada (identity).
Intentar insertar un registro especificando el Id, ¿qué ocurre?
Insertar tres registros sin especificar el Id.
Mirar el contenido de la tabla (id y Titulo)
Eliminar un registro.
Mirar el contenido de la tabla (id y Titulo)
Insertar de nuevo el registro
Mirar el contenido de la tabla (id y Titulo) ¿qué ocurre y por qué?
Desactivar el generado automático campo identity
Insertar nuevo registro sin el Id ¿qué ocurre?
Insertar nuevo registro con el id ¿qué ocurre?
Mostrar el valor del último identity generado

TRUNCATE TABLE

Aprendimos que para borrar todos los registros de una tabla se usa "delete" sin condición "where". También podemos eliminar todos los registros de una tabla con "truncate table".

truncate table nombredelatabla;

Es más eficiente que el delete y tarda menos. El delete elimina registro a registro y el truncate table lo gestiona a través de punteros el sistema.

Ejemplo

```
use facturasbasicas
go
if object_id('FAC_T_Prueba') is not null
  drop table FAC_T_Prueba;
  go


create table FAC_T_Prueba
(
    id integer identity primary key,
    dato varchar(100)
);
go
declare @contador integer =0;
while @contador<=10000
    begin
    insert into FAC_T_Prueba
        values ('Dato'+convert(char,@contador));
    set @contador=@contador+1
    end
go
--select dato from FAC_T_Prueba;
--go
declare @tiempoini datetime=getdate();
delete from FAC_T_Prueba;
declare @tiempofin datetime=getdate();
select DATEDIFF(millisecond,@tiempoini,@tiempofin)
go
```

Creamos y llenamos una tabla, borrando su contenido. Tarda unos 54 milisegundos.

```
use facturasbasicas
go
if object_id('FAC_T_Prueba') is not null
  drop table FAC_T_Prueba;
  go


create table FAC_T_Prueba
(
    id integer identity primary key,
    dato varchar(100)
);
go
declare @contador integer =0;
while @contador<=10000
    begin
    insert into FAC_T_Prueba
        values ('Dato'+convert(char,@contador));
    set @contador=@contador+1
    end
go
--select dato from FAC_T_Prueba;
--go
declare @tiempoini datetime=getdate();
truncate table FAC_T_Prueba;
declare @tiempofin datetime=getdate();
select DATEDIFF(millisecond,@tiempoini,@tiempofin)
go
```

Con el truncate table es más rápido (0 milisegundos)

Ejercicio:

En la Base de datos MoviesBasicas
Crear una tabla con muchos registros
Probar borrarla con delete y con truncate.