

UNIVERSIDAD NACIONAL DE LA MATANZA

DEPARTAMENTO DE INGENIERÍA  
E INVESTIGACIONES TECNOLÓGICAS

INGENIERIA EN INFORMATICA

## BASE DE DATOS

### **Algunas particularidades para probar en lenguaje SQL**

**Versión 1.1**

*Dadas las siguientes estructuras:*

```
create database dbsql1
go

use dbsql1
go
-- Inicializar las tablas
-- drop table tabla4,tabla2,tabla3,tabla1

create table tabla1 (c1 int not null primary key, c2 varchar(20) null)
create table tabla2 (c1 int not null primary key, c2 varchar(50) null)
create table tabla3 (c3 int not null primary key, c1 int null foreign key
references tabla1(c1))
create table tabla4 (c4 smallint not null primary key, c5 varchar(5) not null)

insert into tabla1 values(1,'a')
insert into tabla1 values(2,'b')
insert into tabla1 values(3,'c')

insert into tabla2 values (2,'b')

insert into tabla4 values (1,'1')
insert into tabla4 values (2,'2')
```

*Probá lo siguiente:*

## Consultas

1) Ejecuta la siguiente sentencia:

```
select count(1) from tabla1
```

❶ *¿Qué realizó el count(1)? ¿Existe alguna diferencia con count(\*)? ¿Existe alguna diferencia con haber usado count(c1)? ¿Sería correcto utilizar count(distinct c1)?*

2) Ejecuta la siguiente sentencia:

```
(SELECT DISTINCT 'Juan' AS Nombre, 'Perez' as Apellido)
UNION ALL
(SELECT DISTINCT 'Juan' AS Nombre, 'Perez' as Apellido)
```

❶ *¿Cuántas filas devolvió? ¿Cómo debería realizar para filtrar duplicados? ¿Qué sucede cuando las columnas no tienen alias?*

## **Constraints**

3) Ejecuta la siguiente sentencia:

```
insert into tabla3 values(1,null)
```

❶ ¿Se pudo ejecutar? ¿Qué conclusión podemos tomar de los nulos en las constraints?

## **Joins**

4) Ejecuta la siguiente sentencia:

```
select * from tabla1,tabla2
```

❶ ¿Qué operación realizó entre *tabla1* y *tabla2*? ¿Cómo se llaman las columnas del listado?

5) Ejecuta la siguiente sentencia:

```
select * from tabla1,tabla4 where tabla1.c1=tabla4.c4
```

❶ ¿Se pudo ejecutar la sentencia? ¿Qué tipo de conversión se ha realizado?

6) Ejecuta la siguiente sentencia:

```
select * from tabla1,tabla4 where tabla1.c1=tabla4.c5
```

❶ ¿Se pudo ejecutar la sentencia? ¿Qué tipo de conversión se ha realizado?

7) Ejecuta la siguiente sentencia:

```
select * from tabla1,tabla2 where tabla1.c2=tabla2.c2
```

❶ ¿Se pudo realizar un join entre distintos tipos de varchar?

## **Triggers**

8) Ejecuta la siguiente sentencia:

```
create trigger tg1 on tabla1,tabla2 after insert  
as insert into tabla1 select c1*10, 'tg-' + c2 from inserted
```

❶ ¿Se pudo crear? ¿Por qué? ¿Qué debería realizar para corregirlo?

9) Ejecuta la siguiente sentencia:

```
create trigger tg3 on tabla1 after select  
as insert into tabla1 select c1*10, 'tg-' + c2 from inserted
```

❗ ¿Se pudo crear? ¿Por qué?

- 10) Triggers recursivos: Realizar la siguiente creación de triggers sobre las tablas tabla1 y tabla2. Observar que el trigger de la tabla1 llama implícitamente al trigger de la tabla2.

```
create or alter trigger tg1_tabla1 on tabla1 instead of insert
as
begin
    declare @val1 int, @val2 varchar(20)
    select @val1=c1, @val2=c2 from inserted
    insert into tabla2 values (@val1, @val2)
end

create or alter trigger tg1_tabla2 on tabla2 instead of insert
as
begin
    declare @val1 int, @val2 varchar(20)
    select @val1=c1, @val2=c2 from inserted
    insert into tabla2 values (@val1, @val2 + 'xxxxx')
end

select * from tabla1
select * from tabla2
insert into tabla1 values (4, 'd')
```

❗ ¿Qué sucedió al ejecutarse un trigger que posee una tabla con otro trigger? ¿Podemos decir que un trigger puede llamar a otro trigger implícitamente?

- 11) Triggers cíclicos: Realizar la siguiente creación de triggers sobre las tablas tabla1 y tabla2. En este caso, se observa que existe un ciclo entre las llamadas a triggers, ya que tg1 llama a tabla2 y tg2 llama a tabla1.

```
create or alter trigger tg1_tabla1 on tabla1 instead of insert
as
begin
    declare @val1 int, @val2 varchar(20)
    select @val1=c1, @val2=c2 from inserted
    insert into tabla2 values (@val1, @val2)
end

create or alter trigger tg1_tabla2 on tabla2 instead of insert
as
begin
    declare @val1 int, @val2 varchar(20)
    select @val1=c1, @val2=c2 from inserted
    insert into tabla1 values (@val1, @val2 + 'x')
end

select * from tabla1
select * from tabla2

insert into tabla1 values (4, 'd')
```

❶ ¿Qué sucedió al ejecutarse un trigger que posee una tabla con otro trigger en forma cíclica?  
¿Podemos decir que es posible realizar ciclos a través de los triggers o provoca algún error?

## Vistas

12) Ejecuta la siguiente sentencia:

```
create view vista1
as
select c1 from tabla1 order by 1
```

❶ ¿Se pudo crear? ¿Por qué? ¿Qué debería realizar para corregirlo? Realiza las correcciones necesarias.

13) Ejecuta la siguiente sentencia:

```
create view vista2
as
insert into tabla2 values (2, 'b')
```

❶ ¿Se puede crear una vista con una sentencia de Insert? ¿Qué sentencias pueden existir en una vista?

14) Ejecuta la siguiente sentencia:

```
create view vista2 as select count(*) from tabla2
```

❶ ¿Se pudo crear? ¿Por qué? ¿Qué debería realizar para corregirlo? Realiza las acciones necesarias.

15) Ejecuta la siguiente sentencia:

```
create view vista3
as
select * from tabla1,tabla2
where tabla1.c1=tabla2.c1
```

❶ ¿Se pudo crear? ¿Por qué? ¿Qué debería realizar para corregirlo?

16) Ejecuta la siguiente sentencia:

```
create view vista4
as
select * from tabla1
-----
```

Verificamos qué datos tiene la vista:

```
select * from vista4
```

Intentamos actualizar la vista:

```
update vista4 set c1=c1*10 where c1=1
```

❗ ¿Se pudo realizar?

Cambiamos la vista:

```
alter view vista4  
as  
select top 10 * from tabla1 order by c1
```

Intentamos actualizar la vista:

```
update vista4 set c1=c1*10 where c1=1
```

❗ ¿Se pudo realizar?

Cambiamos la vista:

```
alter view vista4  
as  
select tabla1.*,tabla2.c1 as t2c1, tabla2.c2 t2c2 from tabla1,tabla2  
where tabla1.c1=tabla2.c1
```

Intentamos actualizar la vista:

```
update vista4 set c1=c1*10 where c1=2
```

❗ ¿Se pudo realizar? ¿Cuándo una vista es actualizable?

## **Functions**

17) Ejecuta la siguiente sentencia:

```
create function function1()  
returns int  
as  
begin  
update tabla1 set c1=c1*10  
return (select count(*) from tabla1)  
end
```

❗ ¿Se pudo crear? ¿Qué tipo de sentencias permiten las funciones?