

**UNIVERSIDAD NACIONAL DE LA MATANZA**

DEPARTAMENTO DE INGENIERÍA  
E INVESTIGACIONES TECNOLÓGICAS

INGENIERIA EN INFORMATICA

## **BASE DE DATOS**

### **Ejercicios Propuestos y Resueltos ANSI SQL**

*Jefe de Cátedra: Ing. Verónica Ichazo*

*Docentes a cargo de curso:*

*Ing. Alfonso Palomares*

*Ing. Natalia Crespo*

*Ing. Guillermo Giannotti*

*Docentes a cargo de práctica:*

*Ing. Matías López*

*Ing. Juan Carlos Bordachar*

*Ayudantes:*

*Ezequiel Brizuela*

*Ing. Javier Rebagliatti*

*Ing. Sebastián Deuteris*

*Ing. Hernán Jalil*

**2016**

## Ejercicio 1

Dado el siguiente esquema de relación:

ALUMNO(**dni**, apellido, nombre, escuela)  
 HERMANO\_DE(**dniAlum**, **dniHno**)  
 ESCUELA (**cod**, nombre, direccion)  
 ALIMENTO(**id**, descripcion, marca)  
 ALMUERZA\_EN(**dniAlum**, **idAlimento**, **codEscuela**)

- Listar a todos los alumnos que asisten a escuelas donde no sirven alimentos y almuerzan en otro establecimiento
- Mostrar todas las escuelas que sirven alimentos a todos sus alumnos que no tienen más de dos hermanos

### Creación de Tablas

```
create table Alumno(DNI int not null primary key, Apellido varchar(50), Nombre
varchar(50), CodEscuela int);
create table Hermano_De(DniAlumno int not null, DniHermano int not null, constraint
PK_Hermano_De primary key(DniAlumno, DniHermano));
create table Escuela(CodEscuela int not null primary key, Nombre varchar(50), Direccion
varchar(255));
create table Alimento(IdAlimento int not null primary key, Descripcion varchar(50),
Marca varchar(50));
create table Almuerzo_En(DniAlumno int not null, IdAlimento int not null, CodEscuela
int, constraint PK_Almuerzo_En primary key(DniAlumno, IdAlimento));
```

### Inserción de Datos

```
INSERT INTO escuela (CodEscuela, Nombre, Direccion) VALUES ('1', 'Escuela 1', 'Famosos');
INSERT INTO escuela (CodEscuela, Nombre, Direccion) VALUES ('2', 'Escuela 2', 'Oficialistas');
INSERT INTO escuela (CodEscuela, Nombre, Direccion) VALUES ('3', 'Escuela 3', 'Opositores');
INSERT INTO escuela (CodEscuela, Nombre, Direccion) VALUES ('4', 'Escuela 4', 'Hermanos');
```

```
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('1', 'Fort', 'Ricardo', '1');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('2', 'Marcelo', 'Tinelli', '1');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('3', 'Moria', 'Casan', '1');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('4', 'Cristina', 'Fernandez',
'2');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('5', 'Anibal', 'Fernandez',
'2');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('6', 'Amado', 'Boudou', '2');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('7', 'Ricardo', 'Alfonsin',
'3');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('8', 'Elisa', 'Carrio', '3');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('9', 'Hermes', 'Binner', '3');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('10', 'Guido', 'Tinelli', '4');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('11', 'Hugo', 'Tinelli', '4');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('12', 'Alberto', 'Fernandez',
'4');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('13', 'Silvia', 'Fernandez',
'4');
INSERT INTO alumno(DNI, Apellido, Nombre, CodEscuela) VALUES ('14', 'Ricardo', 'Tinelli',
'4');
```

```
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('2', '10');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('2', '11');
```

```

INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('2', '14');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('5', '12');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('4', '13');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('10', '2');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('10', '11');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('10', '14');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('11', '2');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('11', '10');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('11', '14');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('14', '2');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('14', '10');
INSERT INTO hermano_de (DniAlumno, DniHermano) VALUES ('14', '11');

INSERT INTO alimento (IdAlimento, Descripcion, Marca) VALUES ('1', 'Hamburguesa', 'Patty');
INSERT INTO alimento (IdAlimento, Descripcion, Marca) VALUES ('2', 'Milanesa', 'Granja del Sol');
INSERT INTO alimento (IdAlimento, Descripcion, Marca) VALUES ('3', 'Salchicha', 'Vienisima');

INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('4', '1', '1');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('5', '1', '3');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('4', '2', '4');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('1', '3', '1');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('1', '1', '4');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('2', '1', '1');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('3', '1', '1');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('12', '2', '4');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('13', '2', '4');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('10', '1', '3');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('7', '1', '3');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('8', '2', '3');
INSERT INTO almuerzo_en (DniAlumno, IdAlimento, CodEscuela) VALUES ('9', '3', '3');

```

a)

**Resultado esperado:**

DNI	Apellido	Nombre	CodEscuela
4	Cristina	Fernandez	2
5	Anibal	Fernandez	2

**Solución:**

```

/* (1) Escuelas que no sirven alimentos */
select *
from Escuela
where CodEscuela not in (
    select CodEscuela
    from almuerzo_en
);

/* (1b) Alumnos que asisten a las escuelas de (1) */
select *
from Alumno A
where CodEscuela not in (
    select CodEscuela
    from almuerzo_en
);

/* (2) Alumnos que almuerzan en otro establecimiento */
select *
from Alumno A
join Almuerzo_En AE on A.DNI = AE.DniAlumno

```

```
where A.CodEscuela <> AE.CodEscuela;

/* Respuesta - (1b) y (2) */
select distinct A.*
from Alumno A
join Almuerza_En AE on A.DNI = AE.DniAlumno
where A.CodEscuela <> AE.CodEscuela
and A.CodEscuela not in (
    select CodEscuela
    from almuerza_en
);
```

b) Resultado esperado:

CodEscuela
3

Solución:

```
/* (1) Alumnos que tienen mas de dos hermanos */
select DniAlumno
from Hermano_De
group by DniAlumno
having count(1) > 2;

/* (2) Alumnos que no tienen mas de dos hermanos */
select *
from Alumno
where DNI not in (
    select DniAlumno
    from Hermano_De
    group by DniAlumno
    having count(1) > 2
);

/* (3) Escuelas que no le dan alimento a alguno de sus alumnos de (2) */
select *
from Alumno E1
where not exists(
    select 1
    from Almuerza_En E2
    join Alumno A on A.DNI = E2.DniAlumno
    where DNI not in (
        select DniAlumno
        from Hermano_De
        group by DniAlumno
        having count(1) > 2
    )
    and E1.DNI = E2.DniAlumno
    and E1.CodEscuela = E2.CodEscuela
);

/* (4) Escuelas que dan alimento a algún alumno */
select distinct CodEscuela
from almuerza_en;

/* Resultado: (4) - (3) */
```

```
select distinct CodEscuela
from almuerza_en
where CodEscuela not in (
    select distinct E1.CodEscuela
    from Alumno E1
    where not exists(
        select 1
        from Almuerza_En E2
        join Alumno A on A.DNI = E2.DniAlumno
        where DNI not in (
            select DniAlumno
            from Hermano_De
            group by DniAlumno
            having count(1) > 2
        )
        and E1.DNI = E2.DniAlumno
        and E1.CodEscuela = E2.CodEscuela
    )
);
```

## Ejercicio 2

---

GaleríaDeArte(id, nombre, disponible, calle, nro, localidad)

Obra(id, nombre, material, idTipo, idAutor)

TipoDeObra(id, descripcion)

Temática(id, descripcion)

Exposición(idGaleria, idObra, idTematica, fecha, sala)

Autor(id, nya, fechaNacimiento)

### -- Creación de estructuras.

```
CREATE TABLE GaleriaDeArte
(id int primary key, nombre varchar(50), disponible varchar(50), calle varchar(50), nro
varchar(50), localidad varchar(50));
```

```
CREATE TABLE Autor
(id INT PRIMARY KEY, nya varchar(50), fech_nacimiento DATE );
```

```
CREATE TABLE TipoDeObra
(id int primary key, descripcion varchar(50));
```

```
CREATE TABLE Obra
(id int primary key, nombre varchar(50), material varchar(50),
idTipo int, idAutor int,
FOREIGN KEY (idTipo ) REFERENCES TipoDeObra(id),
FOREIGN KEY (idAutor ) REFERENCES Autor(id) );
```

```
CREATE TABLE Tematica
(id int primary key, descripción varchar(50));
```

```
CREATE TABLE Exposicion
(idGaleria int , idObra int , idTematica int , fecha date, sala int,
PRIMARY KEY(idGaleria, idObra, idTematica, fecha),
FOREIGN KEY (idGaleria ) REFERENCES GaleriaDeArte (id),
FOREIGN KEY (idObra ) REFERENCES TipoDeObra(id),
FOREIGN KEY (idTematica ) REFERENCES Tematica(id)
);
```

### -- Insertando datos.

```
INSERT INTO GaleriaDeArte VALUES
(1, 'Galeria barcelona', '', '', '', ''),
(2, 'Galeria buenos aires', '', '', '', ''),
(3, 'Galeria Florencia', '', '', '', ''),
(4, 'Galeria Recoleta', '', '', '', ''),
(5, 'Galeria Orfeo', '', '', '', '');
```

```
INSERT INTO Autor VALUES
(1, 'dali', '1904-05-11'),
(2, 'picasso', '1881-10-25'),
```

```
(3, 'Joan Miro', '1893-04-20'),
(4, 'Max Ernst', '1891-04-02'),
(5, 'Man Ray', '1890-08-27');
```

```
INSERT INTO TipoDeObra VALUES
(1,'dadaísmo'),
(2,'surrealismo'),
(3,'pop art'),
(4,'Art Deco'),
(5,'Minimalismo');
```

```
INSERT INTO OBRA VALUES
(),
(),
(),
(),
();
```

```
INSERT INTO Tematica VALUES
(),
(),
(),
(),
();
```

```
INSERT INTO Exposicion VALUES
(),
(),
(),
(),
();
```

a - Obtener el nombre de la galería de arte, la descripción de la temática presentada y la fecha de realización, cuando la exposición tuvo la mayor cantidad de obras en expuestas. Sólo se mostrarán los resultados siempre y cuando la galería de arte haya presentado todas las temáticas disponibles o haya expuesto distintas obras a tal punto de haber presentado todos los tipos de obra disponibles.

```
create view exposicion_cantidadObras
as
select distinct e.id_galeria, e.id_tematica, e.fecha, count(id_obra) as cantObras
from exposicion e
group by e.id_galeria, e.id_tematica, e.fecha
```

```
--
```

```
create view exposicion_obrasMax
as
select e.id_galeria, e.id_tematica, e.fecha
from exposicion_cantidadObras
where cantObras = (select max(cantObras) from exposicion_cantidadObras)
--
```

```
create view galaria_todasTematicas
as
select id_galeria, count(id_tematica) tematicasPresentadas
from exposicion
group by id_galeria
having tematicasPresentadas = (select count(*) from tematicas)

--

create view galaria_todosTiposObra
as
select id_galeria, count(o.id_tipo) cantidadTiposObra
from exposicion e
join obra o on e.id_obra = o.id
group by id_galeria
having cantidadTiposObra = (select count(*) from tipoDeObra)

--

select *
from exposicion_obrasMax
where id_galeria in (select id_galeria from galaria_todasTematicas)
or id_galeria in (select id_galeria from galaria_todosTiposObra)
```

b - Se requiere crear un procedimiento almacenados q o función (PostgreSQL) para generar una nueva exposición, por lo tanto se desea recibir por parámetro, el id de la galería de arte, id de la temática, id de la obra a participar y la fecha. Si la exposición no existe se deberá asignar el número de sala "1", pero si la exposición ya existiera deberá utilizarse el número de sala previamente cargado para la misma.

Aclaración: Deberá validar que los id recibidos por parámetros existan en las tablas correspondientes.

#### -- LENGUAJE: SQL SERVER

```
CREATE PROCEDURE nueva_exposicion
@id_galeria as int,
@id_tematica as int,
@id_obra as int
@fecha as datetime
AS

declare @sala as int;

IF ( NOT EXISTS (SELECT 1 FROM GaleriaDeArte WHERE id = @id_galera)
    OR NOT EXISTS (SELECT 1 FROM Tematica WHERE id = @id_tematica)
    OR NOT EXISTS (SELECT 1 FROM Obra WHERE id = @id_obra)
    OR EXISTS (SELECT 1 FROM Exposicion WHERE id_obra = @id_obra AND id_tematica =
@id_tematica AND id_obra = @id_obra) )
BEGIN
```



```
        RETURN;
END

set @sala = SELECT sala from FROM Exposicion WHERE id_galeria = @id_galeria AND
id_tematica = @id_tematica AND fecha = @fecha;

IF @sala is null
BEGIN
    set @sala = 1;
END

INSERT INTO Exposicion (id_galeria, id_tematica, id_obra, fecha, sala) values
(@id_galeria, @id_tematica, @id_obra, @fecha, @sala)

GO
```