

## TAREA OPERACIONES DIFERENTES TIPOS DE JOIN, INTERSECT, UNION

Tomando en cuenta el script dado en la plataforma primero se crearán las tablas de granja y mascota.

```
CREATE TABLE mascota (
    id INT PRIMARY KEY,
    nombrAnimal VARCHAR2 ( 30 ) NOT NULL
);
Table MASCOTA creado.
Transcurrido: 00:00:00.049

CREATE TABLE granja (
    id INT PRIMARY KEY,
    nombrAnimal VARCHAR2 ( 30 ) NOT NULL
);
Table GRANJA creado.
Transcurrido: 00:00:00.017
```

Después de crear las tablas se procederá a ingresar los datos correspondientes:

```
INSERT INTO mascota (id, nombrAnimal)
VALUES (1, 'Iguana');
1 fila insertadas.
Transcurrido: 00:00:00.024

INSERT INTO mascota (id, nombrAnimal)
VALUES (2, 'Gato');
1 fila insertadas.
Transcurrido: 00:00:00.003

INSERT INTO mascota (id, nombrAnimal)
VALUES (3, 'Perro');
1 fila insertadas.
Transcurrido: 00:00:00.003

INSERT INTO mascota (id, nombrAnimal)
VALUES (4, 'Pájaro');
1 fila insertadas.
Transcurrido: 00:00:00.012

INSERT INTO granja (id, nombrAnimal)
VALUES (1, 'Vaca');
1 fila insertadas.
Transcurrido: 00:00:00.012

INSERT INTO granja (id, nombrAnimal)
VALUES (2, 'Caballo');
1 fila insertadas.
Transcurrido: 00:00:00.006
```

```
1 fila insertadas.  
INSERT INTO granja (id, nombAnimal) Transcurrido: 00:00:00.003  
VALUES (3, 'Perro');
```

```
1 fila insertadas.  
INSERT INTO granja (id, nombAnimal) Transcurrido: 00:00:00.003  
VALUES (4, 'Gato');
```

Para confirmar la inserción de datos se mostrará el contenido de estos mismos:

```
SELECT * FROM GRANJA;
```

	ID	NOMBANIMAL
1		3 Perro
2		2 Caballo
3		1 Vaca
4		4 Gato

```
SELECT * FROM MASCOTA;
```

	ID	NOMBANIMAL
1		4 Pájaro
2		1 Iguana
3		3 Perro
4		2 Gato

Ahora procederemos a aplicar una unión como se muestra a continuación:

```
SELECT NOMBANIMAL MASCOTA FROM MASCOTA  
UNION SELECT NOMBANIMAL GRANJA FROM GRANJA;
```

	MASCOTA
1	Caballo
2	Gato
3	Iguana
4	Perro
5	Pájaro
6	Vaca

Para después comparar con un SELECT ALL en donde se muestran los valores repetidos de las tablas.

```
SELECT NOMBANIMAL MASCOTA FROM MASCOTA  
UNION ALL SELECT NOMBANIMAL GRANJA FROM GRANJA;
```

	MASCOTA
1	Pájaro
2	Iguana
3	Perro
4	Gato
5	Perro
6	Caballo
7	Vaca
8	Gato

Después de ver todos se mostrarán de manera descendente en forma alfabética.

```
SELECT NOMBANIMAL MASCOTA FROM MASCOTA  
UNION ALL SELECT NOMBANIMAL GRANJA FROM GRANJA  
ORDER BY MASCOTA DESC;
```

MASCOTA	
1	Vaca
2	Perro
3	Perro
4	Pájaro
5	Iguana
6	Gato
7	Gato
8	Caballo

Ahora se mostrará el ejemplo de INTERSECT en donde solo se verán los valores iguales en las tablas.

```
SELECT NOMBANIMAL MASCOTA FROM MASCOTA  
INTERSECT SELECT NOMBANIMAL GRANJA FROM GRANJA;
```

MASCOTA	
1	Gato
2	Perro

Para después mostrarlo de forma ascendente.

```
SELECT NOMBANIMAL MASCOTA FROM MASCOTA  
INTERSECT SELECT NOMBANIMAL GRANJA FROM GRANJA  
ORDER BY MASCOTA ASC;
```

MASCOTA	
1	Gato
2	Perro

Ahora un ejemplo de la operación MINUS.

```
SELECT NOMBANIMAL MASCOTA FROM MASCOTA
MINUS SELECT NOMBANIMAL GRANJA FROM GRANJA;
```

MASCOTA	
1	Iguana
2	Pájaro

Después pasamos a los JOIN en donde el primer ejemplo nos muestra los elementos iguales en ambas tablas.

```
SELECT a.id id_m, a.nombAnimal mascota,
b.id id_g, b.nombAnimal granja FROM
MASCOTA a INNER JOIN GRANJA b ON
a.NOMBANIMAL= b.NOMBANIMAL;
```

	ID_M	MASCOTA	ID_G	GRANJA
1		3 Perro		3 Perro
2		2 Gato		4 Gato

Continuamos con un LEFT JOIN en donde de la tabla de la derecha puede haber valores nulos.

```
SELECT a.id id_m, a.nombAnimal mascota,
b.id id_g, b.nombAnimal granja FROM
MASCOTA a LEFT JOIN GRANJA b ON
a.NOMBANIMAL= b.NOMBANIMAL;
```

	ID_M	MASCOTA	ID_G	GRANJA
1		3 Perro		3 Perro
2		2 Gato		4 Gato
3		4 Pájaro		(nulo) (nulo)
4		1 Iguana		(nulo) (nulo)

Ahora solo se muestra aquellos valores en donde hay valores nulos en la tabla derecha.

```
SELECT a.id id_m, a.nombAnimal mascota,
b.id id_g, b.nombAnimal granja FROM
MASCOTA a LEFT JOIN GRANJA b ON
a.NOMBANIMAL= b.NOMBANIMAL
WHERE b.id IS NULL;
```

	ID_M	MASCOTA	ID_G	GRANJA
1		4 Pájaro		(nulo) (nulo)
2		1 Iguana		(nulo) (nulo)

Continuamos con un RIGHT JOIN en donde de la tabla de la izquierda puede haber valores nulos.

```
SELECT a.id id_m, a.nombAnimal mascota,
b.id id_g, b.nombAnimal granja FROM
MASCOTA a RIGHT JOIN GRANJA b ON
a.NOMBANIMAL= b.NOMBANIMAL;
```

	ID_M	MASCOTA	ID_G	GRANJA
1		3 Perro		3 Perro
2		2 Gato		4 Gato
3		(nulo) (nulo)		1 Vaca
4		(nulo) (nulo)		2 Caballo

Ahora solo se muestra aquellos valores en donde hay valores nulos en la tabla izquierda.

```
SELECT a.id id_m, a.nombAnimal mascota,
b.id id_g, b.nombAnimal granja FROM
MASCOTA a RIGHT JOIN GRANJA b ON
a.NOMBANIMAL= b.NOMBANIMAL
WHERE a.id IS NULL;
```

	ID_M	MASCOTA	ID_G	GRANJA
1		(nulo) (nulo)		1 Vaca
2		(nulo) (nulo)		2 Caballo

Un ejemplo de FULL OUTER JOIN en donde se muestran todos los valores de ambas tablas con sus valores nulos al compararlas con la otra tabla ya que estos no existen en ambas tablas.

```
SELECT a.id id_m, a.nombAnimal mascota,
b.id id_g, b.nombAnimal granja FROM
MASCOTA a FULL OUTER JOIN GRANJA b ON
a.NOMBANIMAL= b.NOMBANIMAL;
```

	ID_M	MASCOTA	ID_G	GRANJA
1		3 Perro		3 Perro
2		(nulo) (nulo)		2 Caballo
3		(nulo) (nulo)		1 Vaca
4		2 Gato		4 Gato
5		4 Pájaro		(nulo) (nulo)
6		1 Iguana		(nulo) (nulo)

Ahora solo dejaremos los valores donde hay algún valor nulo.

```
SELECT a.id id_m, a.nombAnimal mascota,
b.id id_g, b.nombAnimal granja FROM
MASCOTA a FULL OUTER JOIN GRANJA b ON
a.NOMBANIMAL= b.NOMBANIMAL
WHERE a.id IS NULL OR b.id IS NULL;
```

	ID_M	MASCOTA	ID_G	GRANJA
1		(nulo) (nulo)		2 Caballo
2		(nulo) (nulo)		1 Vaca
3		4 Pájaro		(nulo) (nulo)
4		1 Iguana		(nulo) (nulo)