BASES DE DATOS

FES Aragón

ICO

MTI. Omar Mendoza González

Producto Cartesiano

- El producto cartesiano de dos tablas son todas las combinaciones de todas las filas de las dos tablas.

- La salida consiste en todas las combinaciones de todas las tuplas de ambas tablas usadas en el producto.

Producto Cartesiano

```
SELECT a.clave_alu, ap_paterno, ap_materno,
nombre, b.id_curso, b.id_salon, nl
FROM alumnos a, alumno_salon b
WHERE a.clave_alu = b.clave_alu;
```

JOIN

table_references:

table_reference, ***table_reference***

| ***table_reference*** [INNER | CROSS] JOIN ***table_reference*** [***join_condition***]

| ***table_reference*** STRAIGHT_JOIN ***table_reference***

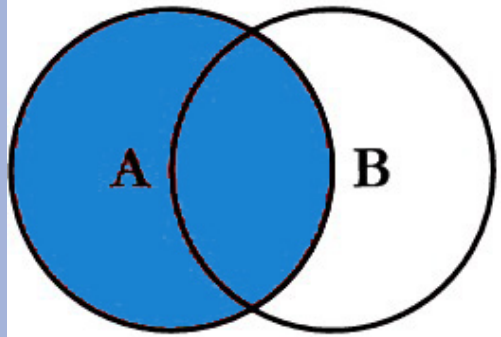
| ***table_reference*** {LEFT|RIGHT} [OUTER] JOIN ***table_reference join_condition***

| ***table_reference*** NATURAL [{LEFT|RIGHT} [OUTER]] JOIN ***table_reference***

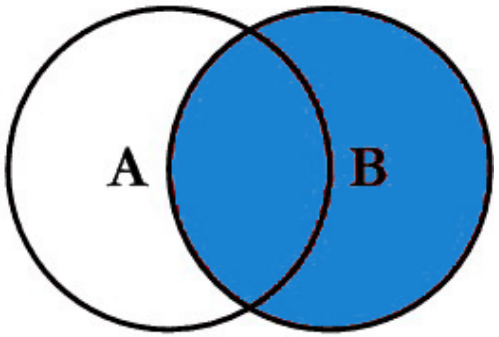
| { OJ ***table_reference*** LEFT OUTER JOIN ***table_reference***

ON ***conditional_expr*** }

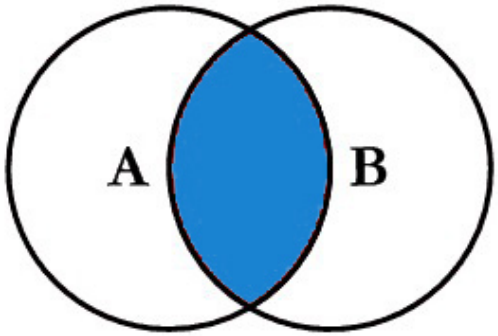
SQL JOINS



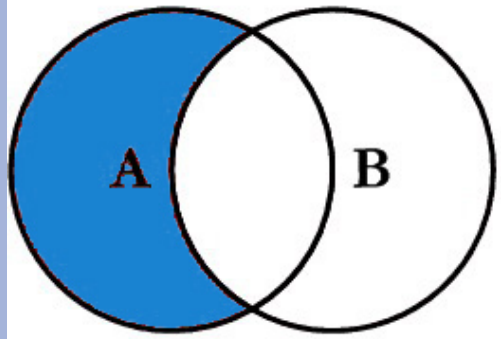

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



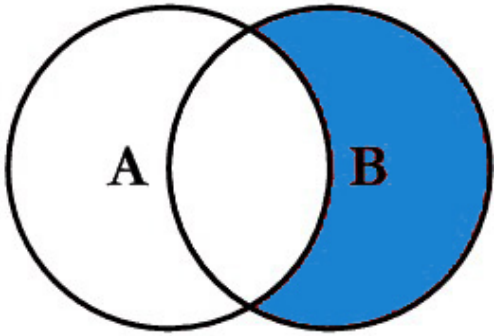

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



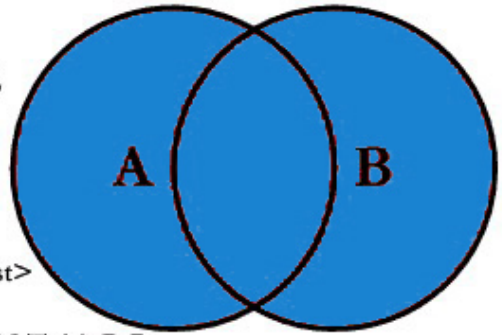

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



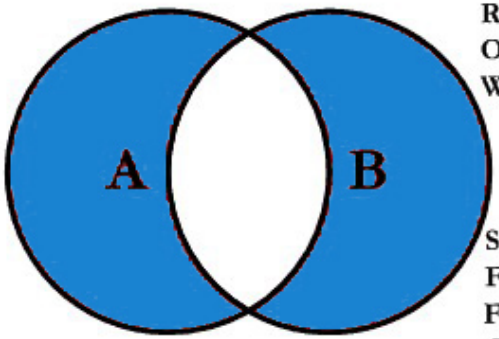

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

JOIN

- Seleccionar todo de 2 tablas

```
SELECT * FROM alumnos
JOIN alumno_salon
```

(1,359,864 renglones)

JOIN

- Usar cláusula ON

```
SELECT * FROM alumnos a
JOIN alumno_salon b
ON (a.clave_alu = b.clave_alu)
```

JOIN

- Usar cláusula USING

```
SELECT * FROM alumnos a
JOIN alumno_salon b
USING (clave_alu)
```

JOIN

- Proyectar los campos deseados

```
SELECT a.clave_alu, ap_paterno, ap_materno,
nombre, b.id_curso, b.id_salon, nl
FROM alumnos a
JOIN alumno_salon b
ON (a.clave_alu = b.clave_alu);
```

JOIN

- Seleccionar registros deseados

```
SELECT a.clave_alu, ap_paterno, ap_materno,
nombre, b.id_curso, b.id_salon, nl
FROM alumnos a
JOIN alumno_salon b
ON (a.clave_alu = b.clave_alu)
WHERE b.id_curso = 'C001';
```

Natural JOIN

```
SELECT a.clave_alu, ap_paterno, ap_materno,
nombre, b.id_curso, b.id_salon, nl
FROM alumnos a
NATURAL JOIN alumno_salon b
WHERE b.id_curso = 'C001';
```

LEFT JOIN

```
SELECT a.clave_alu, ap_paterno, ap_materno,
nombre, b.*
FROM alumnos a
JOIN f_alumno_pagos b
ON (a.clave_alu = b.clave_alu)
ORDER BY pago;
```

LEFT JOIN

```
SELECT a.clave_alu, ap_paterno, ap_materno,
nombre, b.*
FROM alumnos a
LEFT JOIN f_alumno_pagos b
ON (a.clave_alu = b.clave_alu)
ORDER BY pago;
```

RIGHT JOIN

```
SELECT a.clave_alu, ap_paterno, ap_materno,
nombre, b.*
FROM alumnos a
RIGHT JOIN f_alumno_pagos b
ON (a.clave_alu = b.clave_alu)
ORDER BY pago;
```

NATURAL LEFT JOIN

```
SELECT a.clave_alu, ap_paterno, ap_materno,
nombre, b.*
FROM alumnos a
NATURAL LEFT JOIN f_alumno_pagos b
ON (a.clave_alu = b.clave_alu)
ORDER BY pago;
```

UNION

SELECT ...
UNION [ALL | DISTINCT]
SELECT ...
[UNION [ALL | DISTINCT]
SELECT ...]

UNION

```
SELECT clave_alu, ap_paterno, ap_materno,
nombre, 'alumno' as tipo
FROM alumnos
UNION ALL
SELECT clave_prof, apellido_p, apellido_m,
nombres, 'profesor' as tipo
FROM profesor
ORDER BY 2, 3, 4
```

Subconsultas

```
SELECT * FROM
(SELECT clave_alu, ap_paterno, ap_materno,
nombre FROM alumnos WHERE ap_paterno
like 'A%') a
JOIN f_alumno_pagos b
ON (a.clave_alu = b.clave_alu);
```

Subconsultas

```
SELECT a.clave_alu, ap_paterno, ap_materno,
nombre, sum(pago) as tpago, count(pago) as npago
FROM
(SELECT clave_alu, ap_paterno, ap_materno, nombre
FROM alumnos WHERE ap_paterno like 'A%') a
JOIN f_alumno_pagos b ON (a.clave_alu = b.clave_alu)
GROUP BY a.clave_alu, ap_paterno, ap_materno,
nombre;
```

Subconsultas

```
SELECT * FROM
(SELECT a.clave_alu, ap_paterno, ap_materno,
nombre, sum(pago) as tpago, count(pago) as npago
FROM
(SELECT clave_alu, ap_paterno, ap_materno, nombre
FROM alumnos WHERE ap_paterno like 'A%') a
JOIN f_alumno_pagos b ON (a.clave_alu = b.clave_alu)
GROUP BY a.clave_alu, ap_paterno, ap_materno,
nombre ) x
WHERE npago < 10;
```

Subconsultas

```
SELECT * FROM f_alumno_pagos
WHERE clave_alu IN (select clave_alu from alumnos
where sexo = 'M');
```

Ejercicio

- Listar las materias de los alumnos de 5º grado
- Listar el total de pagos por salón
- Listar las materias de cada profesor por salón
- Listar el promedio por curso de cada alumno, mostrando el nombre del alum, la clave, el curso y el promedio
- Listar los alumnos y profesores en una sola tabla y mostrar solo los que en su apellido paterno contengan una letra Z