



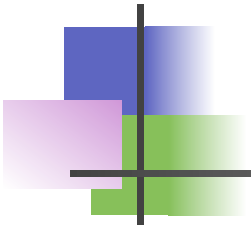
**Departamento de
Ciencias de la Computación y Tecnologías de Información
Universidad del Bío-Bío
Sede Chillán**

Bases de Datos

Lenguaje SQL

M^a Angélica Caro Gutiérrez

<http://www.face.ubiobio.cl/~mcaro/>
mcaro@ubiobio.cl



Lenguaje SQL

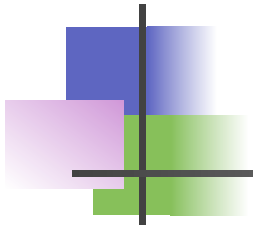
- Introducción
- Conceptos básicos
- Consultas básicas en SQL
- ➔ ■ Consultas complejas en SQL
- Vistas (tablas virtuales) en SQL



Consultas Complejas en SQL

- SELECT: Consulta para recuperar datos de la BD
- La sintaxis de la orden SELECT consta básicamente de las cláusulas SELECT y FROM como obligatorias y de otras varias cláusulas opcionales:

<cláusula SELECT> <cláusula FROM>
[<cláusula WHERE>]
[<cláusula GROUP BY> [<cláusula HAVING>]]
[<cláusula ORDER BY>]



Consultas Complejas en SQL

■ CONSULTAS ANIDADAS CORRELACIONADAS:

- En las consultas anidadas vistas hasta el momento la subconsulta interior ha sido completamente independiente de la consulta exterior
- En general, la subconsulta interior puede depender de la fila (tupla) que se está examinando en cada momento en la consulta exterior
- Ejemplo:

```
SELECT E.Nombre, E.Apellido1 FROM EMPLEADO AS E  
WHERE E.Dni IN (SELECT DniEmpleado FROM Subordinado  
WHERE E.Nombre = NombreSubordinado AND E.sexo = sexo);
```



Consultas Complejas en SQL

- CONSULTAS ANIDADAS CORRELACIONADAS:

- Ambigüedades de los nombres de los atributos:

Obtener el nombre y apellido de cada empleado que tenga un familiar con el mismo nombre de pila y sexo que el empleado.

```
SELECT E.Nombre, E.Apellido1 FROM EMPLEADO AS E
WHERE E.Dni IN (SELECT DniEmpleado FROM Subordinado
WHERE E.Nombre = NombreSubordinado AND E.sexo = sexo);
```

- La regla es: la referencia a un atributo no calificado se refiere a la relación declarada en la consulta anidada más interior

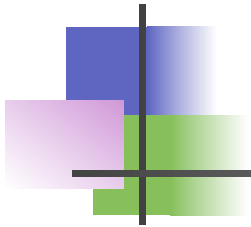


Consultas Complejas en SQL

■ CONSULTAS ANIDADAS CORRELACIONADAS:

- Siempre que una cláusula WHERE de una consulta anidada hace referencia a un atributo de una relación declarada en una consulta externa, **la consulta anidada se evalúa una vez por cada tupla (o combinación de tuplas) en la consulta externa.**
- En la consulta anterior: Por cada tupla empleado, se evalúa la consulta anidada.
- En general una consulta anidada y que emplee los operadores = o IN siempre puede expresarse como una consulta de un sólo bloque Ejemplo:

```
SELECT E.Nombre, E.Apellido1 FROM Empleado AS E, Subordinado  
as D WHERE E.Dni = D.DniEmpleado AND  
E.Nombre = D.NombreSubordinado AND E.sexo = D.sexo);
```



Consultas Complejas en SQL

■ CONSULTAS ANIDADAS CORRELACIONADAS:

- Cláusula **EXIST (NOT EXIST)**, sirve para comprobar si el resultado de una consulta anidada correlacionada es o no vacío.

```
SELECT nombre FROM usuario A WHERE  
EXISTS (SELECT * FROM prestamo WHERE  
        A.carnet = carnet AND fecha_inicio > '31-3-2011');
```

- La subconsulta anterior depende claramente de la fila actual A de USUARIO y se debe volver a evaluar para cada fila de USUARIO



Consultas Complejas en SQL

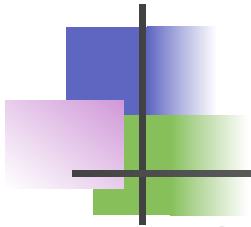
■ CONSULTAS ANIDADAS CORRELACIONADAS:

- Obtener el nombre de las personas que no han pedido nunca un libro

```
SELECT nombre FROM usuario A WHERE NOT EXISTS  
(SELECT * FROM prestamo WHERE A.carnet = carnet);
```

- Obtener el nombre de las personas que han pedido libros y que estos son solo de clase 1

```
SELECT nombre FROM usuario A WHERE EXISTS  
(SELECT * FROM prestamo B  
WHERE A.carnet = B.carnet AND  
NOT EXISTS (SELECT * FROM libro C  
WHERE B.codigo = C.codigo AND clase <> 1));
```

Consultas Complejas en SQL

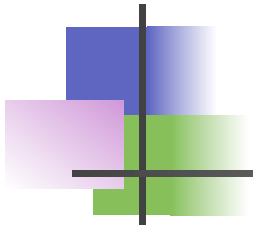
- CONJUNTOS EXPLICITOS:

- Obtener el nombre de los libros que pertenecen a las clases 1, 2 y 3

```
SELECT titulo FROM libro  
WHERE clase IN (1,2,3);
```

- Resultado:

	titulo character varying(30)
1	BASES DE DATOS
2	HIJO DE LADRON
3	VASO DE LECHE



Consultas Complejas en SQL

- Valores null (**IS NULL** y **IS NOT NULL**):

- Obtener el titulo de los libros sin editor

```
SELECT titulo FROM libro  
WHERE editor IS NULL;
```

- Obtener el nombre y dirección de todos los usuarios que posean dirección:

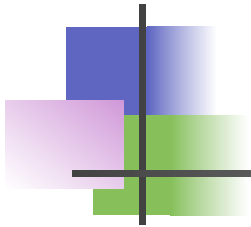
```
SELECT nombre, direccion FROM usuario  
WHERE direccion IS NOT NULL;
```



Consultas Complejas en SQL

- Tablas combinadas (JOIN)
 - Recuperar los nombres y dirección de los empleados que trabajan en el departamento de Investigación

```
SELECT NOMBRE, APELLIDO1, DIRECCION FROM  
(EMPLEADO JOIN DEPARTAMENTO ON DNO =  
NUMERODPTO) WHERE NOMBREDPTO=  
'Investigacion';
```



Consultas Complejas en SQL

- Tablas combinadas (NATURAL JOIN)
 - Recuperar los nombres y dirección de los empleados que trabajan en el departamento de Investigación

```
SELECT NOMBRE, APELLIDO1, DIRECCION FROM  
(EMPLEADO NATURAL JOIN DEPARTAMENTO AS  
DEPTO(NOMBRED, DNO, DNIS, FECHAING)) WHERE  
NOMBRED = 'Investigacion';
```

Consultas Complejas en SQL

- Variantes de la Cláusula JOIN

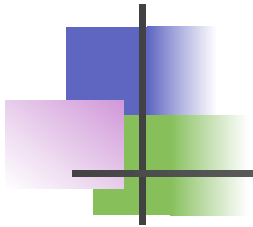
- SQL soporta algunas variedades interesantes de la operación join que aprovechan los valores nulos, las que se denominan Outer Joins
- Considere la sgte operación: Navegantes $\triangleleft \triangleright_{idn=idn}$ Reservas

Navegantes			
idn	nombre	edad	categoría
22	Pedro	45	4
23	Andres	35	6
33	Loreto	31	6
29	Natalia	40	7
30	Esteban	50	8

Reservas		
idn	idb	fecha
23	102	10.11.00
22	102	10.11.00
33	101	05.01.02

- El resultado es:

idn	nombre	edad	categoría	idn	idb	fecha
22	Pedro	45	4	22	102	10.11.00
23	andres	35	6	23	102	10.11.00
33	loreto	31	6	33	101	05.01.02



Consultas Complejas en SQL

■ Cláusula JOIN

- Sin embargo, podría ser interesante mantener las tuplas de Navegantes que no tienen reservas en el resultado. Para esto usamos el Outer Join
- Con Outer Join, las tuplas que no tienen reservas aparecen en el resultado del join y los atributos correspondientes a reservas toman valores nulos
- Existen tres variantes de Outer Join:
 - Left outer join
 - Right outer join
 - Full outer join



Consultas Complejas en SQL

- Cláusula LEFT OUTER JOIN

- Consideremos la consulta:

```
SELECT *
```

```
FROM Navegantes NATURAL LEFT OUTER JOIN Reservas
```

- El resultado es:

idn	nombre	edad	categoria	idn	idb	fecha
22	Pedro	45	4	22	102	10.11.00
23	andres	35	6	23	102	10.11.00
33	loreto	31	6	33	101	05.01.02
29	natalia	40	7	NULL	NULL	NULL
30	esteban	50	8	NULL	NULL	NULL



Consultas Complejas en SQL

- Cláusula RIGHT OUTER JOIN
 - Consideremos las siguientes relaciones:

<i>Reservas</i>		
idn	idb	fecha
23	102	10.11.00
22	102	10.11.00
33	101	05.01.02

<i>Botes</i>		
idb	nombreb	color
101	<i>marino</i>	azul
102	<i>inter-lagos</i>	rojo
103	<i>clipper</i>	verde
104	<i>inter-lagos</i>	rojo

- SELECT * FROM Reservas NATURAL RIGHT OUTER JOIN Botes

idn	idb	fecha	idb	nombreb	color
23	102	10.11.00	102	<i>inter-lagos</i>	rojo
22	102	10.11.00	102	<i>inter-lagos</i>	rojo
33	101	05.01.02	101	<i>marino</i>	azul
NULL	NULL	NULL	103	<i>clipper</i>	verde
NULL	NULL	NULL	104	<i>inter-lagos</i>	rojo



Consultas Complejas en SQL

- Cláusula FULL OUTER JOIN
 - Consideremos las siguientes relaciones:

<i>Reservas</i>		
idn	idb	fecha
23	102	10.11.00
22	102	10.11.00
33	101	05.01.02
33	106	06.01.02

<i>Botes</i>		
idb	nombreb	color
101	<i>marino</i>	azul
102	<i>inter-lagos</i>	rojo
103	<i>clipper</i>	verde
104	<i>inter-lagos</i>	rojo

- SELECT * FROM Reservas NATURAL FULL OUTER JOIN Botes

idn	idb	fecha	idb	nombreb	color
23	102	10.11.00	102	<i>inter-lagos</i>	rojo
22	102	10.11.00	102	<i>inter-lagos</i>	rojo
33	101	05.01.02	101	<i>marino</i>	azul
NULL	NULL	NULL	103	<i>clipper</i>	verde
NULL	NULL	NULL	104	<i>inter-lagos</i>	rojo
33	106	06.01.02	NULL	NULL	NULL