



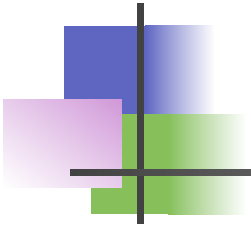
**Departamento de
Ciencias de la Computación y Tecnologías de Información
Universidad del Bío-Bío
Sede Chillán**

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

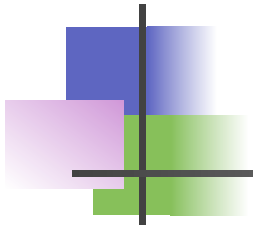
- GROUP BY y HAVING

- “Contar la cantidad de libros de cada clase”
- Una forma de resolverlo sería:

```
SELECT clase, COUNT(*)  
FROM libro  
GROUP BY clase;
```

- Resultado:

	clase integer	count bigint
1	3	2
2	1	1



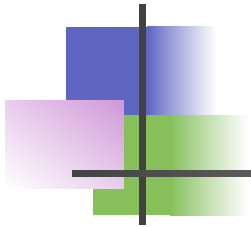
Consultas Complejas en SQL

- GROUP BY y HAVING

- Una consulta general en SQL tiene la siguiente forma:

```
SELECT [DISTINCT] lista-selección  
FROM lista(tablas,vistas)  
WHERE condición  
GROUP BY lista-para-formar-grupos  
HAVING condición-sobre-grupos
```

- La lista-selección en la cláusula SELECT consiste de:
 - 1. Una lista de nombres de atributos
 - 2. Una lista de términos de la forma **OpAgreg**(nombre-columna)
AS nuevo-nombre
- Todos los atributos que aparecen en (1) deben aparecer en lista-para-formar-grupos

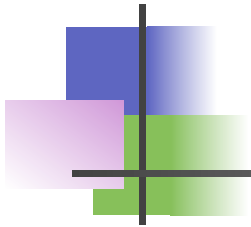


Consultas Complejas en SQL

- GROUP BY y HAVING

```
SELECT [DISTINCT] lista-selección  
FROM lista(tablas,vistas)  
WHERE condición  
GROUP BY lista-para-formar-grupos  
HAVING condición-sobre-grupos
```

- Cada fila del resultado de la consulta se corresponde con un grupo, que es un conjunto de filas que concuerdan con los valores para las columnas de lista-para-formar-grupos
- Las expresiones en condición-sobre-grupos de la cláusula HAVING deben tener un único valor por grupo
- Si se omite GROUP BY, se considera a toda la tabla como un solo grupo



Consultas Complejas en SQL

- Consultas con GROUP BY y HAVING

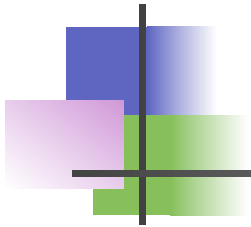
- Dado el esquema: **ALUMNO(ID,NOMBRE,EDAD, CIUDAD)**
- Encontrar la edad del alumno más joven (por sobre los 18 años) para cada ciudad que tenga como mínimo dos alumnos

```
SELECT CIUDAD, MIN(EDAD) AS minEDAD  
FROM ALUMNO  
WHERE EDAD > 18  
GROUP BY CIUDAD  
HAVING COUNT (*) > 1
```

- Los duplicados no se elir
DISTINCT

¿Cuál sería el resultado dada la siguiente tabla?

ID	Nombre	Edad	Cuidad
ICI201039	Juan López	20	Chillán
ICO736822	Juana Tapia	22	Temuco
ICO636254	Igor Torres	18	Chillán
CPA12922	Luis Muñoz	16	San Carlos
ICI103847	Pamela Martin	19	Chillán
ICI273651	Jorge <u>Yañez</u>	18	San Carlos

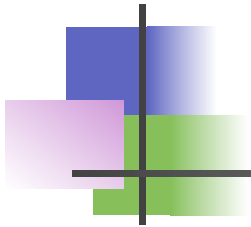


Consultas Complejas en SQL

■ Cláusula HAVING

- No se pueden usar funciones de agrupación en la cláusula WHERE de un SELECT. O sea, no se puede usar el WHERE para, de forma selectiva eliminar datos que no interesan del resultado de una consulta agrupada.
- Por ejemplo, en la tabla
exámenes(id_asignatura, nro_estudiante, nota, fecha)
- Si hacemos:

```
SELECT nro_estudiante, avg(nota)
FROM examenes
WHERE avg (nota) >6
GROUP BY nro_estudiante;
```
- Daría un error por usar una función de agrupamiento en el WHERE



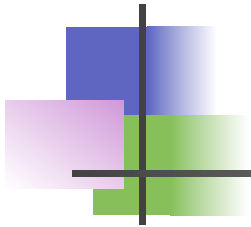
Consultas Complejas en SQL

- Cláusula HAVING

- La cláusula HAVING hace una función parecida a la del WHERE cuando se trabaja con este tipo de funciones. Así, para listar aquellos alumnos cuya media es mayor que 6 sería:

```
SELECT nro_estudiante, avg(nota)
FROM examenes
GROUP BY nro_estudiante
HAVING avg (nota) >6;
```

- El campo referenciado en la cláusula HAVING no puede tener más de un valor por grupo. Esto significa que, en la práctica, HAVING sólo puede referenciar a funciones de agregación y columnas que se están usando en el GROUP by

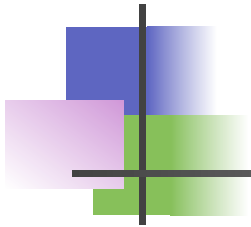


Consultas Complejas en SQL

- Cláusula HAVING

- De cada proyecto, en el que trabajen más de dos empleados, recupere su número, su nombre y el número de empleados que trabajan en él.

```
SELECT NUMPROY, NOMBREPROYECTO, COUNT(*)  
FROM (PROYECTO JOIN TRABAJA_EN  
      ON NUMPROY = NUMPROYECTO)  
GROUP BY NUMPROY, NOMBREPROYECTO  
HAVING COUNT(*) > 2;
```

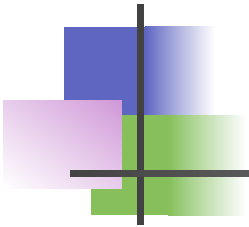


Consultas Complejas en SQL

- Cláusula HAVING

- De cada proyecto, recupere su número, su nombre y el número de empleados del departamento 5 que trabajan en él.

```
SELECT P.NUMPROY, P.NOMBREPROYECTO, COUNT(*)  
FROM ((PROYECTO AS P JOIN TRABAJA_EN  
      ON NUMPROY = NUMPROYECTO)  
      JOIN EMPLEADO ON DNI=DNIEMPLEADO)  
WHERE DNO = 5  
GROUP BY NUMPROY, NOMBREPROYECTO;
```

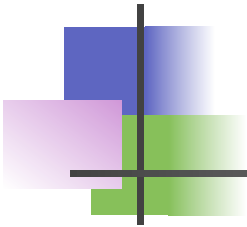


Ejercicios

Dado el esquema:

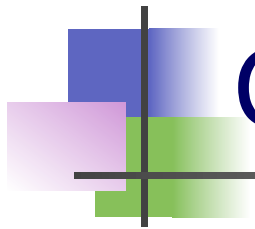
Almacen(**Codigo**, Ubicacion, Capacidad),
Cajas(**Numero**, Contenido, Valor, Almacen), donde Almacen es FK de ALMACEN

1. Obtener el código y número de cajas por cada almacén.
2. Obtener el código y valor promedio de las cajas por cada almacén.
3. Obtener el código de los almacenes donde el valor promedio de las cajas sea mayor a 200.000.
4. Obtener la ubicación de los almacenes que contienen más de 500 cajas



Ejercicios

1. `SELECT Almacen, Count(*) FROM Cajas
GROUP BY Almacen;`
2. `SELECT Almacen, AVG(Valor) FROM Cajas
GROUP BY Almacen;`
3. `SELECT Almacen FROM Cajas GROUP BY Almacen
HAVING AVG(Valor) > 200000;`
4. `SELECT Ubicacion FROM Almacen WHERECodigo IN
(SELECT Almacen FROM Cajas GROUP BY Almacen
HAVING Count (*) > 500);`
- ó `SELECT A.Ubicacion FROM Almacen AS A JOIN
(SELECT Almacen FROM Cajas GROUP BY Almacen
HAVING Count (*) > 500) AS B ON A.Codigo = B.Almacen;`



Creación BD Ejemplo

```
Create Table Almacen(  
Codigo integer,  
Ubicacion varchar(30),  
Capacidad integer,  
PRIMARY KEY (Codigo));
```

```
Create Table Cajas(  
Numero integer,  
Contenido varchar(30),  
Valor integer,  
Almacen integer,  
PRIMARY KEY (Numero),  
FOREIGN KEY (Almacen) REFERENCES Almacen (Codigo));
```