

Lenguaje SQL: Parte II

Base de Datos

Mónica Caniupán
mcaniupan@ubiobio.cl

Universidad del Bío-Bío

2020

Operadores de Agregación

- SQL soporta cinco operaciones de agregación:
 - `COUNT ([DISTINCT] A)`: computa el número de valores (únicos) de la columna A
 - `SUM ([DISTINCT] A)`: computa la suma de todos los valores (únicos) de la columna A
 - `AVG ([DISTINCT] A)`: computa el promedio de todos los valores (únicos) de la columna A
 - `MAX (A)`: computa el valor máximo de la columna A
 - `MIN (A)`: computa el valor mínimo de la columna A

Ejemplo: Consultas de Agregación

- Encontrar el promedio de EDAD de todos los navegantes

```
SELECT AVG (EDAD)  
FROM NAVEGANTES
```

- La respuesta es: $\langle 38,5 \rangle$

NAVEGANTES			
IDN	NOMBRE	CATEGORIA	EDAD
22	<i>pedro</i>	7	45
23	<i>andres</i>	1	35
24	<i>juan</i>	10	30
33	<i>loreto</i>	8	31
29	<i>natalia</i>	7	40
30	<i>esteban</i>	9	50

Ejemplo: Consultas de Agregación

- Encontrar el promedio de EDAD de los navegantes con categoría=7

```
SELECT AVG (N.EDAD)  
FROM NAVEGANTES N  
WHERE N.CATEGORIA = 7
```

- La respuesta es: $\langle 42,5 \rangle$

NAVEGANTES			
IDN	NOMBRE	CATEGORIA	EDAD
22	pedro	7	45
23	andres	1	35
24	juan	10	30
33	loreto	8	31
29	natalia	7	40
30	esteban	9	50

Ejemplo: Consultas de Agregación

- Encontrar el nombre y edad del navegante con mayor edad

```
SELECT N.NOMBRE, N.EDAD  
FROM NAVEGANTES N  
WHERE N.EDAD = (SELECT MAX (N2.EDAD)  
                FROM NAVEGANTES N2)
```

- La respuesta es: $\langle \text{esteban}, 50 \rangle$

NAVEGANTES			
IDN	NOMBRE	CATEGORIA	EDAD
22	<i>pedro</i>	7	45
23	<i>andres</i>	1	35
24	<i>juan</i>	10	30
33	<i>loreto</i>	8	31
29	<i>natalia</i>	7	40
30	<i>esteban</i>	9	50

- La siguiente no es una consulta legal en SQL:

```
SELECT NOMBRE, MAX(EDAD)  
FROM NAVEGANTES
```

Ejemplo: Consultas de Agregación

- Contar el número total de botes

```
SELECT COUNT(*)  
FROM BOTES
```

- Contar el número de nombres (distintos) de navegantes

```
SELECT COUNT(DISTINCT NOMBRE)  
FROM NAVEGANTES
```

Ejemplo: Consultas de Agregación

- Encontrar el nombre de los navegantes de más edad que el marinero con más edad de categoría 10

```
SELECT N.NOMBRE  
FROM NAVEGANTES N  
WHERE N.EDAD > (SELECT MAX(N2.EDAD)  
                 FROM NAVEGANTES N2  
                 WHERE N2.CATEGORIA=10)
```

GROUP BY y HAVING

- Hasta ahora se ha aplicado las operaciones de agregación a todas las tuplas de la relación
- Sin embargo, a menudo se desea aplicar operaciones de agregación a cada uno de los **grupos** de filas de una relación
- **Ejemplo:** Consideremos la consulta: “Encontrar la edad del navegante más joven de cada categoría”
- Una forma de resolverlo sería:

```
SELECT MIN(EDAD)
FROM NAVEGANTES
WHERE CATEGORIA =i
```

con $i = 1, 2, \dots$

- Necesitaríamos tantas consultas como categorías existen

GROUP BY y HAVING

- Necesitamos otra forma de hacerlo

```
SELECT CATEGORIA,
MIN(EDAD) AS MIN_EDAD
FROM NAVEGANTES
GROUP BY CATEGORIA
```

NAVEGANTES			
IDN	NOMBRE	CATEGORIA	EDAD
22	<i>pedro</i>	7	45
23	<i>andres</i>	1	35
24	<i>juan</i>	10	30
33	<i>loreto</i>	8	31
29	<i>natalia</i>	7	40
30	<i>esteban</i>	9	50

- La respuesta es:

CATEGORIA	MIN_EDAD
7	40
1	35
10	30
8	31
9	50

Consultas Generales

- 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 **OPAGR**(*nombre-columna*) **AS** *nuevo-nombre*
- Todos los atributos que aparecen en (1) deben aparecer en *lista-para-formar-grupos*

Consultas Generales

```
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

Ejemplo: Consultas con GROUP BY y HAVING

- Encontrar la edad del navegante más joven que tiene derecho a voto (tiene como mínimo 18 años) para cada categoría que tenga como mínimo dos navegantes con derecho a voto

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

- Los duplicados no se eliminan a menos que se especifique con DISTINCT

Ejemplo: Consultas con GROUP BY y HAVING

- Para cada bote rojo, averiguar el número de reservas realizadas

```
SELECT B.IDB, COUNT (*) AS numRESERVAS  
FROM BOTES B, Reserva R  
WHERE B.IDB = R.IDB AND B.COLOR = 'rojo'  
GROUP BY B.IDB
```

Ejemplo: Consultas con GROUP BY y HAVING

- ¿Por qué es incorrecta la siguiente consulta?

```
SELECT B.IDB, COUNT (*) AS numRESERVAS  
FROM BOTES B, Reserva R  
WHERE B.IDB = R.IDB  
GROUP BY B.IDB  
HAVING B.COLOR ='rojo'
```

Ejemplo: Consultas con GROUP BY y HAVING

- Averiguar el promedio de edad de los navegantes de cada categoría que tenga como mínimo dos navegantes

```
SELECT CATEGORIA, AVG (EDAD) AS PromEdad  
FROM NAVEGANTES  
GROUP BY CATEGORIA  
HAVING COUNT (*) > 1
```