Sql 2

jueves, 5 de octubre de 2023 14:38

- Operadores que permite **ORDENAR** las tuplas
  - **ORDER BY** *atributo:*
    - especifica el atributo por el cual las tuplas serán ordenadas
    - Ejemplo 13: presentar todos los asociados ordenados por nombre.

```
SELECT nombre
FROM asociados
ORDER BY nombre
```

    - **Desc, asc**: por defecto ascendente, se puede especificar descendente.

Se puede ordenar por varios criterios

```
(SELECT a.nombre
FROM asociados a INNER JOIN practica p ON a.idsocio = p.idsocio
                 INNER JOIN deportes d ON p.iddeporte = d.iddeporte
WHERE d.nombre = "FUTBOL" )
UNION / UNION ALL / INTERSECT / EXCEPT
(SELECT a.nombre
FROM asociados a INNER JOIN practica p ON a.idsocio = p.idsocio
                 INNER JOIN deportes d ON p.iddeporte = d.iddeporte
WHERE d.nombre = "VOLEY" )
```

- Asociados=(idsocio, nombre, dirección, teléfono, sexo, estadocivil, fechanacimiento, idlocalidad)
- Deportes=(iddeporte, nombre, monto_cuota, idsede)
- Practica = (idsocio, iddeporte)
- Localidad =(idlocalidad, nombre)
- Sedes = (idsede, nombre, dirección, idlocalidad)

- Funciones de agregación:
  - Promedio (avg): aplicable a atributos numéricos, retorna el promedio de la cuenta
  - Mínimo (min): retorna el elemento más chico dentro de las tuplas para ese atributo
  - Máximo (max): retorna el elemento más grande dentro de las tuplas para ese atributo
  - Total (sum): aplicable a atributos numéricos, realiza la suma matemática
  - Cuenta (count): cuenta las tuplas resultantes.

Alumno(legajo, materia, nota)

select legajo, AVG(nota) as promedio
from Alumno
group by legajo
Agrupando las por 2 parámetros

select legajo, count(nota) as cant → cant de veces q' rindió un alumno
from Alumno
group by legajo → siempre va a mostrar lo = q' agrupé o nada (solo la cant)

select count(nota) as cant
from Alumno
group by legajo

Primero se hace group by, después lo otro

select legajo, materia, AVG(nota) as promedio
from Alumno
group by legajo, materia
} Un subgrupo dentro de legajo.

where se hace antes q' group by → atributo

HAVING → condición q' se aplica a tupla después de agrupar
→ siempre después de group by.

SUBCONSULTAS

- Subconsultas anidadas
  - Pertenencia a conjuntos: **IN**
    - Ejemplo 29: mostrar aquellos asociados que practiquen Basquet.

```
SELECT nombre
FROM asociados
WHERE idsocio IN ( SELECT idsocio
                   FROM practica p INNER JOIN deportes d ON (d.iddeporte = p.idpractica )
                   WHERE d.nombre = "basquet" )
```

    - Ejemplo 30: mostrar los asociados de Gonnet que practiquen Handball.

- Subconsultas anidadas
  - Comparación de Conjuntos
    - > **some** (<, =, >=, <=, <>)
    - Ejemplo 31: mostrar todos los deportes, menos el mas económico

```
SELECT nombre
FROM deportes
WHERE montocuota > SOME ( SELECT montocuota
                          FROM deportes )
```

    - > **all** (<, =, >=, <=, <>)
    - Ejemplo 32: presentar el deporte mas oneroso

```
SELECT nombre
FROM deportes
WHERE montocuota >= ALL ( SELECT montocuota
                          FROM deportes )
```

select nombre max( ) X no nos serviría.
solo con group by

Sn → 1 elem el mín en subconjunto

EXIST

- Cláusula **EXIST**: devuelve verdadero si la subconsulta argumento no es vacía.

→ si el conjunto q' devuelve no es vacío

```
SELECT A.NOMBRE
FROM ASOCIADOS A, DEPORTE D, PRACTICA P
WHERE (A.IDSOCIO = P.IDSOCIO) AND (P.IDDEPORTE = D.IDDEPORTE)
      AND (D.NOMBRE = 'FÚTBOL')
      AND
      EXIST (SELECT *
             FROM DEPORTE D2, PRACTICA P2
             WHERE (D2.IDDEPORTE = P2.IDDEPORTE) AND (IDNOMBRE='VOLEY') AND
             P2.IDSOCIO = A.IDSOCIO
```

→ nom de los socios q' hacen algún deporte.

VISTA → Genera una tabla temporal con la consulta.

↳ create view nombre as <consulta>

```
create view judo as (select nombre from ...)
select nombre
from judo
```

} No usar mucho.
Se ejecuta cada vez q' la llamamos.

deportes

IDDEP	NOM
1	fút
2	judo
3	voley

práctica

IDDEP	socio
1	Juan
2	MARÍA

```
Select *
FROM Deporte INNER JOIN
PRACTICA ON (D.IDDEP=P.IDDEP)
```

D.IDDEP	nombre	D.IDDEP	socio
1	fútbol	1	Juan
2	judo	2	MARÍA
3	voley	—	—

Select *
FROM Depto (Left Join)

PRAOTECA on (D.SUDEP=P.SUDEP)

Dadas las siguientes tablas

- Cliente (id_cliente, nombre_cliente, renta_anual, tipo_cliente)
- Embarque (embarque_#, id_cliente, peso, camión_#, destino, fecha)
- Camión (camión_#, nombre_chofer)
- Ciudad (nombre_ciudad, población)

Indique los clientes que han tenido embarques transportados en cada camión.

```
SELECT *
FROM cliente
WHERE NOT EXISTS(
	SELECT *
	FROM CAMION ca
	WHERE NOT EXISTS(
		SELECT *
		FROM embarque e
		WHERE e.id=cliente.id AND e.camion=ca.camion
		)
	)
)
```