



Diseño de Bases de Datos

Clase 4

Prof. Pablo Thomas

Rodolfo Bertone

Agenda

Lenguaje de Consultas Estructurado (SQL)

- Definiciones
- DDL / DML
- Consultas/Updates
- Ejemplos

Lenguaje de Consultas Estructurado (SQL)

Historia

1986 Es un estándar ANSI

1992 se amplia el estándar (SQL2 o SQL 92)

1999 Se crea SQL 2000 incorporando expresiones regulares, consultas recursivas y características de OO

2003 Surge SQL 3 agrega características de XML

2006 se definen características que lo acercan al mundo W3C

SQL → DDL

Lenguaje de definición de datos

- Es muy amplio
- Solo veremos las operaciones mas comunes
- CREATE DATABASE
- DROP DATABASE
- CREATE TABLE
- ALTER TABLE
- DROP TABLE

SQL → DDL

- ▶ Ejemplos de Creación de Tablas

```
CREATE TABLE empresas ( idempresa INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
                      empresa VARCHAR(100) NOT NULL,  
                      abreviatura VARCHAR(10) NULL,  
                      cuit VARCHAR(13) NULL,  
                      direccion VARCHAR(100) NULL,  
                      observaciones TEXT NULL,  
                      PRIMARY KEY(idempresa),  
                      UNIQUE INDEX empresas_index19108(empresa));
```

SQL → DDL

- ▶ Ejemplos de Creación de Tablas

```
CREATE TABLE pacienteempresas
( idpacienteempresa INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  dpaciente_os INTEGER UNSIGNED NOT NULL,
  idempresa INTEGER UNSIGNED NOT NULL,
  fecha_desde DATE NOT NULL,
  fecha_hasta DATE NULL,
  PRIMARY KEY(idpacienteempresa),
  INDEX empleadosempresas_FKIndex1 (idempresa),
  INDEX pacienteempresas_FKIndex2 (dpaciente_os),
  FOREIGN KEY(idempresa) REFERENCES empresas(idempresa)
    ON DELETE RESTRICT
    ON UPDATE NO ACTION,
  FOREIGN KEY(dpaciente_os) REFERENCES pacientes_os(idpaciente_os)
    ON DELETE RESTRICT
    ON UPDATE NO ACTION);
```

SQL → DDL

- ▶ Ejemplo de modificación de tablas

```
ALTER TABLE empresas (
    Add column razon_social VARCHAR(100) NOT NULL,
    Drop column cuit,
    Alter column direccion VARCHAR(50) NULL);
```

SQL → DML

► Estructura básica

```
SELECT lista_de_atributos  
FROM lista_de_tablas  
WHERE condición
```

```
SELECT atr1, atr2, atr3  
FROM tabla1, tabla2  
WHERE atr4 = "Valor"
```

Equivale a

$$\pi_{\text{atr1, atr2, atr3}} (\sigma_{\text{atr4} = \text{'valor'}} (\text{tabla1} \times \text{tabla2}))$$

SQL → DML

Tablas sobre las cuales se resolverán las consultas

- 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)

SQL → DML

- ▶ Ejemplo 1: nombre de todos los asociados

```
SELECT nombre  
FROM asociados
```

- ▶ La Cláusula Select puede contener
 - ▶ * (incluye todos los atributos de las tablas que aparecen en el from)
 - ▶ Distinct (eliminan tuplas duplicadas)
 - ▶ All (valor por defecto, aparecen todas las tuplas)
- ▶ Ejemplo 2: mostrar todos los datos de los asociados
- ▶ Ejemplo 3: mostrar todos los id de localidades donde viven los asociados, sin repetir valores

SQL → DML

- ▶ Ejemplo 4: mostrar de cuanto sería la cuota de cada deporte si se incrementara en un 25%

```
SELECT montocuota*1.25  
FROM deportes
```

SQL → DML

► Cláusula WHERE

- Ejemplo 5: **Operadores lógicos**: mostrar los asociados varones casados.
- Ejemplo 6: **Operador BETWEEN**: mostrar los deportes cuya cuota esté entre 400 y 600 pesos mensuales

```
SELECT nombre  
FROM deporte  
WHERE montocuota >= 400 AND montocuota <= 600
```

```
SELECT nombre  
FROM deporte  
WHERE montocuota BETWEEN 400 and 600
```

SQL → DML

► Cláusula FROM

- Ejemplo 7: **Producto Cartesiano**: mostrar para cada asociado su nombre y la localidad de residencia
- Ejemplo 8: **Producto Natural** mostrar para cada asociado su nombre y la localidad de residencia

```
SELECT asociado.nombre, localidad.nombre  
FROM asociado, localidad  
WHERE asociado.idlocalidad = localidad.idlocalidad
```

```
SELECT asociado.nombre, localidad.nombre  
FROM asociado INNER JOIN localidad ON (asociado.idlocalidad=localidad.idlocalidad)
```

SQL → DML

- ▶ **Operación de Renombrar**
 - ▶ Ejemplo 9: mostrar todos los deportes salvo el mas costoso.
 - ▶ Ejemplo 10: presentar todos los asociados con nombre, dirección y localidad. El listado debe figurar con la leyenda DIRECCIÓN LEGAL.
- ▶ **Operaciones sobre cadenas**
 - ▶ **Operador Like:** %, _
 - ▶ “Alfa%”: cualquier cadena que empiece con Alfa
 - ▶ "%casa%": cualquier cadena que tenga casa en su interior
 - ▶ “___”: cualquier cadena con tres caracteres
 - ▶ “__%”: cualquier cadena con al menos tres caracteres.
 - ▶ Ejemplo 11: mostrar aquellos asociados cuyo nombre empiece con RA
 - ▶ Ejemplo 12: mostrar los deportes que incluyan la palabra BALL en su definición

SQL → DML

- ▶ 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.

- ▶ Ejemplo 14 : presentar los asociados que nacieron el mes de agosto ordenados por fecha desde el 31 al 1 de agosto.

SQL → DML

16

Otras operaciones del álgebra, Operaciones sobre conjuntos

- ▶ Unión: agrupa las tuplas resultantes de dos subconsultas.
- ▶ Union all conserva duplicados
 - ▶ Ejemplo 15: asociados que practican futbol o voley
- ▶ Intersección:
 - ▶ Ejemplo 16: asociados que practican futbol y voley
- ▶ Diferencia: (except)
 - ▶ Ejemplo 17: asociados que p

```
(SELECT 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 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" )
```

SQL → DML

- ▶ 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.
- ▶ Ejemplos:
 - ▶ Ejemplo 18: mostrar la cantidad de asociados del club
 - ▶ Ejemplo 19: mostrar el asociado que aparece primero en una lista alfabética
 - ▶ Ejemplo 20: mostrar cual es la cuota promedio que se cobra
 - ▶ Ejemplo 21: mostrar cual será la recaudación mensual de futbol.
 - ▶ Ejemplo 22: mostrar el deporte que cobra la cuota mas alta

SQL → DML

- ▶ **Operación de Agrupamientos (**GROUP BY**):**
 - ▶ Permite agrupar un conjunto de tuplas por algún criterio
 - ▶ Ejemplo 23: obtener la cantidad de asociados que practica cada deporte
 - ▶ Ejemplo 24: obtener la recaudación mensual de cada deporte
 - ▶ Ejemplo 25: informar para cada asociado cuantos deportes practica y el valor en cuotas que abona
 - ▶ Ejemplo 26: que modificaría al problema anterior para mostrar solamente los asociados de La Plata
 - ▶ **HAVING:** permite aplicar condiciones a los grupos
 - ▶ Ejemplo 27: mostrar todos los deportes (Id es suficiente) y la cantidad de asociados que lo practican siempre y cuando haya mas de 100 miembros

SQL → DML

▶ Valores nulos:

- ▶ Los atributos con valores nulos deben ser tratados de manera especial
- ▶ Ejemplo 28: Mostrar aquellos deportes que no tengan valor ingresado en su cuota

```
SELECT nombre  
FROM deportes  
WHERE montocuota IS NOT NULL
```

SQL → DML

- ▶ Subconsultas anidadas

- ▶ Pertenencia a conjuntos: **IN**

- ▶ Ejemplo 29: mostrar aquellos asociados que practiquen Basquet.

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

- ▶ Comparación de Conjuntos

- ▶ > **some** (<, =, >=, <=, <>)

- ▶ Ejemplo 31: mostrar todos los deportes, menos el mas económico

Select nombre

From deporte

Where monto_cuota > some (select monto_cuota from deporte)

- ▶ > **all** (<, =, >=, <=, <>)

- ▶ Ejemplo 32: presentar el deporte mas oneroso

SQL → DML

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

- ▶ Ejemplo 33: mostrar los asociados que practican futbol y voley (una tercera variante)

Select a.nombre

From asociados a, deporte d, practica p

**Where a.idasociado=p.idasociado and. p.iddeporte=d.iddeporte and
d.nombre="futbol" and exist (select * from... where And
a.idasociado=a2.idasociado)**

- ▶ Ejemplo 34: obtener los asociados que practiquen todos los deportes.

- ▶ **Creación de vistas** (tienen la estructura de una tabla y almacena temporalmente una consulta). Cada vez que la vista es utilizada, la consulta almacenada es “recalculada”

- ▶ Create View nombre as <expresion>

- ▶ Ejemplo 35: crea una vista con todos los asociados que practican yudo

- ▶ CreateView Judo as (select nombre from....)

SQL → DML

▶ Variantes del producto natural

- ▶ **Left outer Join:** primero se calcula el inner join (ídem anterior) y luego cada tupla t perteneciente a la relación de la izquierda que no encontró par aparece en el resultado con valores nulos en los atributos del segundo lado.
- ▶ **Right outer Join:** ídem anterior pero aparecen las tuplas t de la relación de la derecha
- ▶ **Full outer join:** aparecen las tuplas colgadas de ambos lados.
- ▶ Ejemplos:
 - ▶ Ejemplo 36: Presentar para cada deporte un listado de asociados que lo practican. Pueden haber deportes que aun no tengan asociados y deben aparecer igualmente en el listado
 - ▶ Ejemplo 37: Presentar para cada asociado los deportes que practican. Pueden que haya alguno que no practique deportes, igualmente debe aparecer.

SQL → DML

- ▶ ABM sobre tablas:

- ▶ Inserción:

- INSERT INTO *tab_name* (<column_name>,) VALUES (<valor>,,)**

- ▶ Ejemplo 38: agregar un nuevo asociado a la tabla

- ▶ Borrado

- DELETE FROM *tab_name***

- [WHERE condición];**

- ▶ Ejemplo 39: quitar de la tabla el deporte bochas

- ▶ Ejemplo 40: como haría en el caso anterior si bochas tuviera deportistas y estuviera definida integridad referencial

- ▶ Actualización

- UPDATE *tab_name***

- set *column name* = *valor***

- ▶ Ejemplo 41: incrementar la cuota de cada deporte en un 20%

- ▶ Ejemplo 42: incrementar la cuota de hockey en un 30%

SQL → DML

► Ejercicios para resolver

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)

SQL → DML

1. Cuál es el nombre del cliente 433?
2. Cuál es la ciudad destino del embarque 3244?
3. Cuales son los números de los camiones que han llevado paquetes (embarques) por encima de 100 kg?
4. Dé todos los datos de los embarques de más de 20 kg?
5. Cree una lista por orden alfabético de los clientes con renta anual de más de 10 millones?
6. Cual es el Id del cliente José García?
7. Mostrar los nombres de los clientes que han enviado embarques a las ciudades cuyo nombre empieza con C.
8. Mostrar los nombres de los clientes que han enviado embarques a las ciudades cuyo nombre termina con City.
9. Mostrar los nombres de los clientes que tienen una D como tercera letra del nombre.
10. Mostrar los nombres de los clientes que sean minoristas

SQL → DML

11. Cómo se llaman los clientes que han enviado paquetes a Bariloche?
12. A cuales destinos han hecho envíos las compañías con renta anual menor que 1 millón?
13. Cuales son los nombres y las poblaciones de las ciudades que han recibido embarques que pesen más de 100 kg?
14. Cuales con los clientes que tienen más de 5 millones de renta anual y que han enviado embarques de menos de 1 kg?
15. Quienes son los clientes que tienen más de 5 millones de renta anual y que han enviado embarques de menos de 1kg. O han enviado embarques a Villa La Angostura?
16. Quienes son los choferes que han conducido embarques de clientes que tienen renta anual mayor de 20 millones a ciudades con más de 1 millón de habitantes?
17. Indique los choferes que han transportado embarques a cada una de las ciudades.
18. Indique las ciudades que han recibido embarques de clientes que tienen más de 15 millones de renta anual.
19. Indique el nombre y la renta anual de los clientes que han enviado embarques que pesan más de 100 kg.
20. Indique los clientes que han tenido embarques transportados en cada camión.

SQL → DML

21. Cual es el peso promedio de los embarques?
22. Cual es el peso promedio de los embarques que van a Neuquén?
23. De un alista de los clientes para los que todos sus embarques han pesado más de 25 kg.
24. Cuales ciudades de la BD tienen la menor y la mayor población?
25. Agregue el camión 95 con el chofer García a la BD
26. Borre de la BD todas las ciudades con población de menos de 5000 habitantes. Debe sacar, además los embarques que haya en dicha ciudad.
27. Borre de la BD todas las ciudades con población de menos de 5000 habitantes que no posean embarques enviados.
28. Convierta el peso de cada envío a libras, para ello se sabe que una libra son 2.2 kg. (aproximadamente).

SQL → DML

29. Indique las ciudades que han recibido embarques de todos los clientes
30. Cuantos embarques han sido enviados pro el cliente 433?
31. Para cada cliente ¿cuál es el peso medio de los paquetes enviados por él?
32. Para cada ciudad ¿cuál es el peso máximo de un paquete que haya sido enviado a dicha ciudad?
33. Para cada ciudad con población por encima de un millón de habitantes ¿cuál es el peso menor de un paquete enviado a dicha ciudad?
34. Para cada ciudad que haya recibido al menos diez paquetes, ¿cuál es el peso medio de los paquetes enviados a dicha ciudad?