

SQL

Consulta de Datos

L. en C.C. Odín M. Escorza Soria

UNAM



Las consultas tienen 4 componentes principales:

- `SELECT <lista de atributos>`
- `FROM <relación objetivo>`
- `WHERE <condición de selección>`
- `[<Modificadores de resultado>]`

Listas de atributos



Una lista de atributos es una lista separada por comas. Cada elemento de la lista...

- puede ser:
 - ★ una constante
 - ★ el nombre de alguno de los atributos de la relación objetivo
 - ★ una operación aritmética, lógica, de texto o definida para un tipo en particular (por ejemplo: concatenación, descomposición de fechas, etc.)
 - ★ una consulta que regrese solamente una columna y un renglón.
 - ★ el resultado de evaluar una función
- puede ser *renombrado* utilizando la palabra reservada AS seguida del nombre correspondiente

Ejemplos



- Cálculos de valores:

```
select pow(4*5+9-10/22-9*5,2)
```

- Selección de constantes:

```
select 'a',1,'b',2
```

- Renombrado de columnas:

```
select 'a' as atr1,7+9 as atr2
```

- Selección de columnas particulares:

```
select idEstado,nombre from estado
```

- Selección de todos los atributos de una tabla:

```
select * from ocupacion
```

- Selección con sub-consulta:

```
select claveEstado,  
       nombre,  
       (select idMunicipio  
        from municipio  
        limit 1)  
from estado
```

Relación Objetivo



Es la relación sobre la cuál se seleccionarán las columnas y renglones correspondientes.

- Puede tratarse de:
 - ★ el nombre de una relación definida en la bd (relvar)
 - ★ una operación (producto cartesiano, unión, join, etc.) entre una o más relaciones
 - ★ cualquier expresión cuyo resultado sea una relación
- también se puede renombrar con AS

Ejemplos



- producto cartesiano:

```
select a.nombre as est,  
       b.nombre as mun  
from estado as a,municipio as b
```

- diferencia:

```
select nombre  
from ((select * from estado)  
      except (select * from estado where nombre like 'A%')) as a
```

- unión

```
select nombre  
from ((select * from estadocivil) union (select * from ocupacion)) as n
```

- intersección

```
select *  
from ((select nombre from asentamiento where idMunicipio = 1618)  
      intersect (SELECT nombre from asentamiento where idMunicipio = 7)) as ms
```

Condición de Selección



Una condición de selección puede ser cualquier expresión SQL que se evalúe a falso o verdadero:

Comparación de columnas contra literales.

```
select * from estado where claveEstado = 'DF'
```

Operadores lógicos

```
select * from estado where (poblacion > 1000000 and nombre like 'C%') or nombre like 'O%'
```

Modificadores de resultados



Una vez satisfecha la condición de selección, se puede manipular un poco más el resultado:

- Ordenando según criterios con:

`ORDER BY [exp [ASC|DESC]][,...].`

- Limitando el número de resultados deseados con:

`LIMIT N` donde *N* es el número de tupas deseadas.

- Agrupando los resultados según ciertos criterios con:

`GROUP BY [exp]+ [HAVING cond]`

Operadores infijos



- Aritméticos
 - *: Multiplicación
 - +: Suma numérica
 - /: División
 - −: Resta
 - ?: Residuo
- Cadenas
 - ||: Concatenación
- Lógicos
 - AND: Conjunción
 - OR: Disyunción
 - NOT: Negación
- Comparación
 - =: Igualdad
 - <>: Diferencia
 - <=: Menor o igual
 - >=: Mayor o igual
 - LIKE: Comparación de cadenas por aproximación
 - IS NULL: Verificación de nulos
 - IS NOT NULL: Verificación de nulos
 - BETWEEN .. AND ...: Valor dentro de intervalo
 - .. IN ...: Contención en conjuntos

Funciones escalares para cadenas



- `concat(text,text)→text`: concatenación.
- `left(text,int)→text`: primeros n caracteres.
- `right(text,int)→text`: últimos n caracteres.
- `substring(text,int,int)→text`: substring(texto,posición,longitud).
- `ltrim(text)→text`: elimina espacios al inicio.
- `rtrim(text)→text`: elimina espacios al final.
- `upper(text)→text`: a mayúsculas.
- `lower(text)→text`: a minúsculas.
- `length(text)→int`: longitud en caracteres.

Funciones escalares para fechas



- `now()`→timestamp: regresa la fecha actual en el servidor.
- `extract(.. from timestamp)`→number: extrae partes de una fecha según el patrón:

★ year

★ day

★ second

★ month

★ hour

★ week

★ minute

- `to_char(timestamp,pattern)`→text: le da formato a una fecha según cierto patrón, el patrón depende del SDBD.
- `datediff(timestamp,timestamp)`→int: número de días entre una fecha y otra.

Funciones escalares para números



- `random()`→float: Número pseudo-aleatorio entre 0 y 1
- `round(float)`→int: Redondea al entero más cercano
- `floor(float)`→int: El entero menor más cercano
- `ceil(float)`→int: El entero mayor más cercano
- `pow(float,float)`→float: Potenciación, `pow(a,b)` es a^b
- `sqrt(float)`→float: Raíz cuadrada: $\text{sqrt}(a) = \text{pow}(a, 0.5)$
- `pi()`→float: Un valor aproximado para el número π

Funciones y operadores generales



- $\text{coalesce}(T[\dots]) \rightarrow T$: Regresa el primer valor de la lista que no es nulo.
- $\text{case [when .. then ..]}^+ \text{ else .. end}$: Evaluación condicional.

Definiciones



- Unión: Todas las tuplas que están en una de las relaciones o la otra.

```
select ... from ... UNION select ... from ...
```

- Unión Total: Todas las tuplas que están en una de las relaciones o la otra sin eliminar duplicados.

```
select ... from ... UNION ALL select ... from ...
```

- Intersección: Las tuplas tanto en la primera relación como en la segunda.

```
select ... from ... INTERSECT select ... from ...
```

- Diferencia: Las tuplas que están en la primera relación, pero no en la segunda.

```
select ... from ... EXCEPT select ... from ...
```

¿Qué rayos es NULL?



- Se trata de una “etiqueta” que representa que ignoramos el valor.
- El resultado de operar escalarmente cualquier cosa con null, es null.
- Para ordenar, dependiendo del SMBD, pueden aparecer al principio o al final.
- Conlleva ciertos problemas que separan a SQL del modelo relacional.

Motivación



En nuestra base de datos...

¿cuántos municipios tiene cada estado?

Para averiguarlo, podríamos intentar algo como:

```
select e.nombre as edo,  
       m.nombre as mun  
  from estado as e,municipio as m  
 where m.idestado = e.idestado  
order by e.nombre;
```

Pero tendríamos que agrupar a mano los 2457 registros resultantes, o hacer 32 consultas, una por cada estado.

Motivación



Las funciones de agregación nos permiten hacer cálculos sobre “grupos de datos”, en su modo más simple, colapsan toda la relación objetivo:

¿Cuántos registros hay en la tabla estado?

```
select count(*) from estado
```

¿Cuántos registros hay en la tabla persona?

```
select count(*) from persona
```

Regresando al problema original:

```
select e.nombre as edo,  
       count(*)  
  from estado as e,municipio as m  
 where m.idestado = e.idestado  
group by e.nombre  
order by e.nombre
```

Estructura



```
select agg1(<1>),agg2(<2>) , ...  
      from ...  
[group by <3>]  
[having <4>]
```

1. Parámetros que recibe la función *agg1*
2. Parámetros que recibe la función *agg2*
3. Lista de criterios de agrupación: una lista separada por comas de expresiones válidas SQL o nombres de atributos en la relación objetivo.
4. Condición de filtrado después post-agrupación.

Resumen



Algunas de las funciones más populares son:

1. `count (..)`
2. `max (..)`
3. `min (..)`
4. `sum (..)`
5. `stddev (..)`

..y entre los paréntesis:

`[DISTINCT] expr [ORDER BY ..] , ...`

Esto es, para calcular cada función de agregación, podemos ignorar duplicaciones y elegir el orden en el que se llevará a cabo el cálculo.

Definición



Le llamamos sub-consulta a cualquier expresión SQL que incluye un SELECT y aparece dentro de otra consulta (en las partes en donde el lenguaje lo permite).

Por ejemplo,
hallar el nombre de todos los estados en donde no hay asentamientos cuyo nombre empiece con “Santiago”

```
select e.nombre as Edo
  from estado as e
 where not exists (select *
                   from asentamiento as a,municipio as m
                  where a.idmunicipio = m.idmunicipio
                        and e.idestado = m.idestado
                        and a.nombre like 'Santiago%')
```

Clasificación



Las subconsultas se distinguen en 3 categorías:

- Cuando aparecen como “componente” de la relación objetivo.
 - ★ Se calcula solamente una vez, a la hora de generar la relación objetivo.
 - ★ No puede tener referencias a atributos de otros componentes de la relación objetivo.
- Cuando aparecen dentro de una lista de columnas.
 - ★ Se calcula una o varias veces dependiendo de la definición.
 - ★ Puede tener referencias a atributos de la relación objetivo que no estén agrupados.
- Cuando aparecen como parte de una condición.
 - ★ Se calcula una o varias veces dependiendo de la definición.
 - ★ Puede tener referencias a atributos de la relación objetivo que no estén agrupados.