

111mil Programadores</>

Base de Datos

Apunte 4



Prof. Germán C. Basisty

basisty.german@gmail.com



FUNDACIÓN DII EJ

Desarrollo para la Inclusión e
Igualdad en el Empleo Juvenil

Índice de Contenidos

Índice de Contenidos	1
Funciones Agregadas	2
MAX	2
MIN	2
SUM	2
COUNT	2
AVG	2
Consultas avanzadas	3
Ordenamiento	3
Cláusula DISTINCT	3
Cláusula DISTINCT ON	4
Consulta simultánea sobre múltiples tablas	4
JOIN	4
Combinaciones Internas	5
Combinaciones externas	5
Combinación externa izquierda	6
Combinación externa derecha	6
Combinación externa completa	7
Combinaciones cruzadas	7
Limitar la cantidad de resultados	7
Ejercitación	9

Funciones Agregadas

Son funciones que se utilizan para determinar estadísticas relacionadas con un conjunto de valores.

MAX

Devuelve el valor más alto de un campo seleccionado. La sintaxis es la siguiente:

```
SELECT MAX(campo) FROM nombre_tabla;
```

MIN

Devuelve el valor más pequeño de un campo seleccionado. La sintaxis es la siguiente:

```
SELECT MIN(campo) FROM nombre_tabla;
```

SUM

Devuelve la suma de un conjunto de valores de un campo específico de la tabla. La sintaxis es la siguiente:

```
SELECT SUM(campo/expresión) FROM nombre_tabla;
```

COUNT

Devuelve el número de filas que devuelve una consulta. La sintaxis es la siguiente:

```
SELECT COUNT(campo) FROM nombre_tabla;
```

AVG

Devuelve el promedio de un campo. La sintaxis es la siguiente:

```
SELECT AVG(campo) FROM nombre_tabla;
```

Consultas avanzadas

Ordenamiento

Se puede ordenar el conjunto de tuplas resultante de una consulta en función de uno o más campos de la misma, y determinar el sentido del ordenamiento que puede ser descendente o ascendente.

```
SELECT * FROM producto ORDER BY precio DESC;
```

Retornará todos los productos ordenados por su precio de forma descendente (el mayor primero).

```
SELECT * FROM producto ORDER BY nombre ASC;
```

Retornará todos los productos ordenados por su nombre de forma ascendente (como es un campo alfanumérico, los ordena alfabéticamente).

Cláusula DISTINCT

Con la cláusula **DISTINCT** se especifica que los registros con ciertos datos duplicados sean obviadas en el resultado. Por ejemplo, queremos conocer todos los autores de los cuales tenemos libros, si utilizamos esta sentencia:

```
SELECT autor FROM libros;
```

Aparecen todos los autores, repetidos según la cantidad de libros que hayan escrito. Para obtener la lista de autores sin repetición usamos:

```
SELECT DISTINCT autor FROM libros;
```

Para contar los distintos autores, sin considerar el valor "null" usamos:

```
SELECT count(DISTINCT autor) FROM libros WHERE autor IS NOT NULL;
```

La cláusula "distinct" afecta a todos los campos presentados. Para mostrar los títulos y editoriales de los libros sin repetir títulos ni editoriales, usamos:

```
SELECT DISTINCT título, editorial FROM libros;
```

Entonces, **DISTINCT** elimina registros duplicados.

Cláusula DISTINCT ON

DISTINCT ON(atributo) retornará la “primera” fila de cada set de tuplas en donde el atributo tenga el mismo valor. La consulta a continuación retorna el nombre y el año de la tabla ceos del conjunto de tuplas donde el nombre es distinto:

```
apple=# select distinct on (name) name, year from ceos;
```

name	year
-----+-----	
Gil Amelio	1996
John Sculley	1983
Michael Spindler	1993
Mike Markkula	1977
Mike Scott	1978
Steve Jobs	1976
(6 rows)	

No siempre es obvio cuál es la primera tupla de un conjunto de registros, por este motivo es una buena práctica combinar **DISTINCT ON** con **ORDER BY** para hacerla explícita.

Consulta simultánea sobre múltiples tablas

SQL permite combinar los atributos de tablas relacionadas, por ejemplo:

```
SELECT
    a.nombre_libro,
    a.autor,
    b.nombre_editorial
FROM
    libro a,
    editorial b
WHERE
    a.editorial_id = b.id;
```

La anterior consulta retorna los campos nombre_libro y autor de la tabla libro, más nombre_editorial de la tabla editorial, siempre que el campo editorial_id (FK) de la tabla libro y id (PK) de la tabla editorial sean iguales.

JOIN

Un join es una operación que relaciona dos o más tablas para obtener un resultado que incluya datos (campos y registros) de ambas; las tablas participantes se combinan según los campos comunes a ambas tablas. Hay tres tipos de combinaciones.

Combinaciones Internas

La combinación interna emplea JOIN o INNER JOIN, y utiliza para obtener información de intersección entre dos tablas y combinar dicha información en una salida.

La sintaxis es:

```
SELECT
    a.campo1
    a.campo2
    b.campo1
    b.campo4
FROM
    tabla1 a
    INNER JOIN tabla2 b ON a.campo_fk = b.campo_pk
WHERE
    ....
```

Esta consulta retornará las tuplas de a y de b siempre y cuando haya concordancia. Si para alguna tupla de a no existe una tupla en b, será ignorada. Si los campos que relacionan a y b en ambas tablas se llaman igual, puede reemplazarse el ON por USING:

```
SELECT
    a.campo1
    a.campo2
    b.campo1
    b.campo4
FROM
    tabla1 a
    INNER JOIN tabla2 b USING(campo_combinador)
WHERE
    ....
```

Combinaciones externas

Las combinaciones externas combinan registros de dos tablas que cumplen la condición, más los registros de la segunda tabla que no la cumplen; es decir, muestran todos los registros de las tablas relacionadas, aún cuando no haya valores coincidentes entre ellas. Existen 3 tipos de combinaciones externas:

Combinación externa izquierda

Se emplea una combinación externa izquierda para mostrar todos los registros de la tabla de la izquierda. Si no encuentra coincidencia con la tabla de la derecha, el registro muestra los campos de la segunda tabla como NULL.

La sintaxis es:

```
SELECT  
    campos  
FROM  
    tablaizquierda  
LEFT OUTER JOIN tabladerecha ON condicion;
```

Combinación externa derecha

Se emplea una combinación externa derecha para mostrar todos los registros de la tabla de la derecha. Si no encuentra coincidencia con la tabla de la izquierda, el registro muestra los campos de la segunda tabla como NULL.

La sintaxis es:

```
SELECT  
    campos  
FROM  
    tablaizquierda  
RIGHT OUTER JOIN tabladerecha ON condicion;
```

Combinación externa completa

Se emplea una combinación externa completa para mostrar todos los registros ambas tablas. Si no se encuentra coincidencia, el registro muestra los campos faltantes como NULL.

La sintaxis es:

```
SELECT  
    campos  
FROM  
    tablaizquierda  
    FULL OUTER JOIN tabladerecha ON condicion;
```

Combinaciones cruzadas

Las combinaciones cruzadas (cross join) muestran todas las combinaciones de todos los registros de las tablas combinadas. Para este tipo de join no se incluye una condición de enlace. Se genera el producto cartesiano en el que el número de filas del resultado es igual al número de registros de la primera tabla multiplicado por el número de registros de la segunda tabla, es decir, si hay 5 registros en una tabla y 6 en la otra, retorna 30 filas.

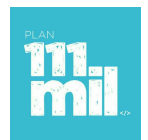
La sintaxis es:

```
SELECT  
    campos  
FROM  
    tablaizquierda  
    CROSS JOIN tabladerecha;
```

Limitar la cantidad de resultados

Es posible limitar la cantidad de tuplas retornadas por una consulta utilizando la cláusula **LIMIT**.

```
SELECT * FROM prueba LIMIT 100;
```

Retornará solo los primeros 100 resultados de la consulta. La cláusula complementaria **OFFSET** permite a **LIMIT** traer un subconjunto de resultados a partir del registro n.

```
SELECT * FROM prueba LIMIT 100 OFFSET 50;
```

Retornará solo los primeros 100 resultados a partir de la tupla 51.

Ejercitación

Sobre la base de datos *inmobiliaria*

https://github.com/gbasisty/111mil/blob/master/baseDatos/inmobiliaria_adt.sql

- 1) Determinar cuál es la propiedad de mayor superficie.
- 2) Determinar cuál es la propiedad de menor superficie.
- 3) Determinar el promedio de superficie de las propiedades.
- 4) Contar la cantidad de propiedades registradas en la base.
- 5) Determinar la sumatoria de los precios de todas las propiedades que están en venta y aún no han sido vendidas.
- 6) Determinar cuál es la propiedad en venta de mayor valor a cargo del vendedor Pedro Iriarte.
- 7) Determinar cuál es el vendedor que más propiedades ha vendido.
- 8) Determinar cuál es el vendedor que menos propiedades ha vendido.
- 9) Determinar cuál es el vendedor que más ganancias ha generado.
- 10) Determinar cuál es el vendedor que menos ganancias ha generado.
- 11) Determinar cuál es la región en donde hay más actividad del vendedor Luisa Estevez.
- 12) Determinar cuál es el vendedor que más ventas realizó en la zona de Barcelona.
- 13) Determinar cuál es la propiedad más cara de Lleida en venta.