

# HENRY

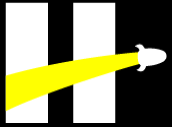


Subconsultas, Vistas y  
Funciones Ventana



## Principales Objetivos de Aprendizaje

- Diferenciar el concepto de Subconsultas y Vistas
- Comprender en qué casos son útiles las Funciones Ventana



# Subconsultas

Una consulta secundaria es una consulta de selección que está contenida dentro de otra consulta.

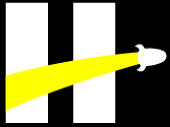
La consulta de selección interna generalmente se usa para determinar los resultados de la consulta de selección externa. Consiste en utilizar los resultados de una consulta dentro de otra, que se considera la principal.

Es una sentencia `SELECT` anidada dentro de una instrucción `SELECT`, `SELECT...INTO`, `INSERT...INTO`, `DELETE`, o `UPDATE` o dentro de otra subconsulta.



# Subconsultas

Una subconsulta tiene la misma sintaxis que una sentencia SELECT exceptuando que aparece encerrada entre paréntesis, no puede contener la cláusula ORDER BY, ni puede ser la UNION de varias sentencias SELECT.



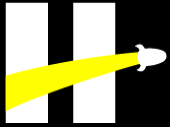
# Subconsultas

Las subconsultas son un proceso de selección interno, y se pueden utilizar en cualquier sentencia que permita una expresión.

- SELECT: Para calcular y crear un nuevo campo virtual a la consulta principal.
- FROM: Para devolver una tabla secundaria calculada o un campo calculado con un contexto diferente.
- WHERE: Para definir filtros compuestos calculados. Si se conociera el valor a calcular con la subconsulta, su utilizaría ese valor.

Listas --> IN , NOT IN.

Valor único --> = , >=, <=, etc.



# Vistas

Es un mecanismo que permite almacenar de forma permanente el resultado de una consulta en SQL. A su vez este resultado almacenado en la vista se puede acceder como si fuera una tabla, denominándose a la vista como una tabla virtual.

Las vistas se componen de campos y filas provenientes del resultado de la consulta, las cuales pueden venir de varias tablas. Al igual que con los otros objetos que forman parte de la base de datos se crean mediante la sentencia CREATE y se eliminan mediante DROP.



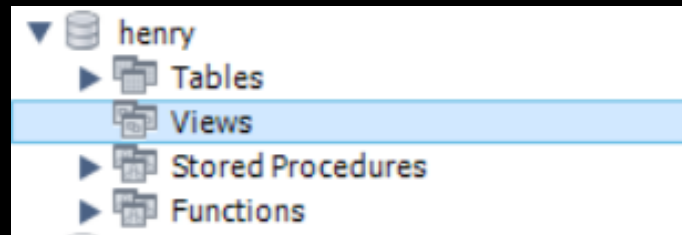
# Vistas

```
-- Crear una vista.  
CREATE VIEW primerosAlumnos AS  
SELECT idAlumno, fechaIngreso  
FROM alumnos  
WHERE fechaIngreso = ( SELECT MIN(fechaIngreso) AS fecha  
                        FROM alumnos)  
  
-- Obtener los resultados de una vista.  
SELECT *  
FROM primerosAlumnos  
  
-- Modificar una vista.  
ALTER VIEW primerosAlumnos AS  
SELECT idAlumno, CONCAT(apellido," ",nombre), fechaIngreso  
FROM alumnos  
WHERE fechaIngreso = ( SELECT MIN(fechaIngreso) AS fecha  
                        FROM alumnos)  
  
-- Eliminar una vista  
DROP VIEW primerosAlumnos
```



# Vistas

Al crear una vista, esta queda alojada en la base de datos correspondiente y se pueden ver en la interfaz del gestor de base de datos.







# Vistas

Una vista actúa como filtro de las tablas subyacentes a las que se hace referencia en ella.

La consulta que define la vista puede provenir de una o de varias tablas, o bien de otras vistas de la base de datos actual u otras bases de datos.

Asimismo, es posible utilizar las consultas distribuidas para definir vistas que utilicen datos de orígenes heterogéneos. Esto puede resultar de utilidad, por ejemplo, si desea combinar datos de estructura similar que proceden de distintos servidores, cada uno de los cuales almacena los datos para una región distinta de la organización.



# Vistas

## Ventajas

- Permite centrar, simplificar y personalizar la forma de mostrar la información a cada usuario.
- Se usa como mecanismo de seguridad, el cual permite a los usuarios obtener acceso a la información proveniente de la vista sin acceder a otras opciones.
- Proporciona una sintaxis simple para acceder a los resultados de la vista.



# Función Ventana

Puede entenderse como un conjunto de registros y una función que se ejecuta sobre los mismos y cumple determinadas condiciones. Para cada registro se debe ejecutar una función en esta ventana.

Las funciones ventana evita los JOIN de una tabla consigo misma:

```
-- Promedio de ventas por Fecha:
SELECT Fecha,
        AVG(Precio * Cantidad) AS Promedio_Ventas
FROM venta
GROUP BY Fecha;

-- Unimos el promedio de ventas por fecha con las ventas por fecha:
SELECT v.Fecha,
        v.Precio * v.Cantidad AS Venta,
        v2.Promedio_Ventas
FROM   venta v JOIN ( SELECT Fecha,
                             AVG(Precio * Cantidad) AS Promedio_Ventas
                       FROM venta
                       GROUP BY Fecha) v2
ON (v.Fecha = v2.Fecha);
```



# Función Ventana

¿Cómo podría una función ventana ayudar en este caso?

```
SELECT v.Fecha,  
       v.Precio * v.Cantidad AS Venta,  
       AVG(v.Precio * v.Cantidad) OVER (PARTITION BY v.Fecha) AS Promedio_Ventas  
FROM venta v;
```

Fecha	Venta	Promedio_Ventas
2015-01-01	2355.000	6014.4992308
2015-01-01	4710.000	6014.4992308
2015-01-01	321.000	6014.4992308
2015-01-01	321.000	6014.4992308
2015-01-01	642.000	6014.4992308
2015-01-01	1638.120	6014.4992308
2015-01-01	1638.120	6014.4992308
2015-01-01	1638.120	6014.4992308
2015-01-01	1638.120	6014.4992308
2015-01-01	2364.000	6014.4992308
2015-01-01	3546.000	6014.4992308
2015-01-02	356.000	1705.6317647
2015-01-02	178.000	1705.6317647
2015-01-02	534.000	1705.6317647
2015-01-02	356.000	1705.6317647
2015-01-02	1030.000	1705.6317647
2015-01-02	1030.000	1705.6317647
2015-01-02	1030.000	1705.6317647
2015-01-02	1545.000	1705.6317647
2015-01-02	1030.000	1705.6317647
2015-01-02	515.000	1705.6317647
2015-01-02	515.000	1705.6317647



# Función Ventana

La función ventana se puede descomponer en las siguientes partes:

```
SELECT v.Fecha,  
       v.Precio * v.Cantidad AS Venta,  
       SUM(v.Precio * v.Cantidad) OVER (PARTITION BY v.Fecha ORDER BY v.IdVenta) AS Total_Ventas  
FROM venta v;
```

Fecha	Venta	Total_Ventas
2015-01-10	1637.680	161342.300
2015-01-10	1637.680	162979.980
2015-01-10	1637.680	164617.660
2015-01-10	17407.280	182024.940
2015-01-12	2475.000	2475.000
2015-01-12	3712.500	6187.500
2015-01-12	3712.500	9900.000
2015-01-12	1024.000	10924.000
2015-01-12	1024.000	11948.000
2015-01-12	2509.760	14457.760
2015-01-12	3764.640	18222.400
2015-01-12	2509.760	20732.160
2015-01-12	706.640	21438.800
2015-01-12	387684.000	409122.800
2015-01-13	2978.000	2978.000
2015-01-13	2978.000	5956.000
2015-01-13	1062.000	7018.000
2015-01-13	708.000	7726.000
2015-01-13	7672.500	15398.500
2015-01-13	7672.500	23071.000
2015-01-13	5115.000	28186.000
2015-01-13	5115.000	33301.000
2015-01-13	2557.500	35858.500