



Laboratorio de Bases de Datos (EBB)

Unidad IV – Resumen de datos y Vistas

Departamento de Electricidad, Electrónica y Computación
Facultad de Ciencias Exactas y Tecnología
Universidad Nacional de Tucumán

Primer Cuatrimestre 2017



Introducción [1 | 2]

- ▼ Resumen de datos
 - ▼ Uso de funciones de agrupamiento
 - ▼ Fundamentos de `GROUP BY / HAVING`
 - ▼ Generación de valores
 - ▼ Limitación de la cantidad de filas



Introducción [2 | 2]

- ▼ Implementación de vistas
 - ▼ Concepto / Ventajas
 - ▼ Definición de vistas
 - ▼ Modificación de datos
 - ▼ Consideraciones

Uso de funciones de agrupamiento [1 | 4]

- ▼ **Funciones de agrupamiento:** funciones que calculan promedios, sumas, números de valores en una expresión, etc
- ▼ Cuando se ejecuta una función de agrupamiento se resumen los valores de una tabla completa (o *ResultSet*) en un único valor representativo

Uso de funciones de agrupamiento [2 | 4]

▼ Consideraciones:

- ▼ Se emplean con la sentencia `SELECT` o en combinación con la cláusula `GROUP BY`
- ▼ Salvo `COUNT (*)`, todas las funciones de agrupamiento devuelven `NULL` si ninguna fila satisface la condición del `WHERE`
- ▼ Los valores nulos pueden hacer que estas funciones devuelvan resultados inesperados

- Si se usan funciones de agrupamiento en valores nulos:
 - Salvo `COUNT (*)`, el resto de las funciones de agrupamiento ignoran los valores nulos.
 - La función `COUNT (*)` cuenta las filas del *ResultSet*, aún cuando los valores sean nulos.

Uso de funciones de agrupamiento [3 | 4]

- ▼ Algunas funciones de agrupamiento:
 - ▼ **AVG**: calcula el promedio de los valores en una expresión
 - ▼ **COUNT**: cuenta el número de valores en una expresión
 - ▼ **COUNT(*)**: cuenta el número de filas seleccionadas
 - ▼ **MAX/MIN**: devuelve el valor máximo/mínimo de una expresión
 - ▼ **SUM**: suma el total de los valores de una expresión

Uso de funciones de agrupamiento [4 | 4]

- ▼ Ejemplos:
 - ▼ 1. Contar la cantidad de empleados
 - ▼ 2. Sumar la cantidad de productos para la orden 10248
 - ▼ 3. Contar la cantidad de empleados jefes

Funciones de agrupamiento

Fundamentos de GROUP BY / HAVING [1 | 5]

- ▼ **GROUP BY:** organiza las filas en grupos, resumiendo los mismos con un resultado representativo
- ▼ Ejemplo: Por cada producto, mostrar la cantidad que se ordenó de cada uno

Uso de GROUP BY

Fundamentos de GROUP BY / HAVING [2 | 5]

- ▼ Consideraciones de GROUP BY:
 - ▼ Por cada grupo se produce una columna de valores
 - ▼ Devuelve filas simples por cada grupo que se especifica
 - ▼ Si se especifica la cláusula WHERE sólo se agrupan las filas que la satisfacen

Fundamentos de GROUP BY / HAVING [3 | 5]

- ▼ Consideraciones de GROUP BY (continuación):
 - ▼ La cantidad de bytes que se pueden tener en la lista de GROUP BY depende del SGBDR

- La cantidad de bytes que se pueden tener en la lista de GROUP BY depende del SGBDR. Por ejemplo, en SQL Server se pueden tener hasta 8060 bytes.

Fundamentos de GROUP BY / HAVING [4 | 5]

- ▼ HAVING: se usa en columnas o expresiones de agrupamiento incluidas en el *ResultSet* para determinar condiciones de filtro
- ▼ Ejemplo: Por cada producto, mostrar la cantidad que se ordenó de cada uno pero sólo para aquellos productos donde la cantidad supere las 300 unidades

Uso de GROUP BY con HAVING

Resumen de datos
Implementación de vistas

Uso de funciones de agrupamiento
Fundamentos de GROUP BY / HAVING
Generación de valores
Limitación de la cantidad de filas

Fundamentos de GROUP BY / HAVING [5 | 5]

- ▼ Consideraciones de HAVING:
 - ▼ HAVING es a GROUP BY lo que WHERE es a SELECT
 - ▼ La cantidad de condiciones que se pueden especificar en la cláusula HAVING depende del SGBDR
 - ▼ Se pueden referenciar cualquiera de las columnas que aparecen en la lista de selección

Lab. Bases de Datos (EBB) | Unidad IV - 201712

- Cuando se usa GROUP BY no se puede especificar WHERE.
- La cantidad de condiciones que se pueden especificar en la cláusula HAVING depende del SGBDR. Por ejemplo, en SQL Server se pueden tener hasta 128 condiciones.

Generación de valores [1 | 9]

- ▼ **Modificador ROLLUP:** genera que se incluyan filas extras que representan operaciones de agrupamiento de un nivel más alto (super agrupamiento)
- ▼ Ejemplos:
 - ▼ 1. Por cada producto, mostrar su cantidad ordenada y luego la cantidad total de productos
 - ▼ 2. Por cada producto, mostrar la cantidad ordenada por cada orden, y luego la cantidad total de productos

Modificador ROLLUP en MySQL y SQL Server

Generación de valores [2 | 9]

- ▼ Consideraciones de ROLLUP:
 - ▼ Se procesan los datos de derecha a izquierda en la lista de columnas del GROUP BY. Luego se aplica la función de agrupamiento a cada grupo
 - ▼ Se agregan filas al resultado que muestran los acumulados agrupados, y se indican con valores nulos en el *ResultSet*

Generación de valores [3 | 9]

- ▼ Consideraciones de `ROLLUP` (continuación):
 - ▼ La cantidad de expresiones de agrupamiento depende del SGBDR
 - ▼ En MySQL `ROLLUP` y `ORDER BY` no se pueden usar juntos. Sin embargo, en MySQL `GROUP BY` ordena los resultados, pudiéndose especificar `ASC` o `DESC` en las columnas especificadas por `GROUP BY`

- La cantidad de expresiones agrupadas con `ROLLUP` depende del SGBDR. Por ejemplo, en SQL Server se pueden tener hasta 10 expresiones agrupadas.

Generación de valores [4 | 9]

- ▼ Consideraciones de ROLLUP (continuación):
 - ▼ Las columnas del ROLLUP deben tener algún significado en las reglas de negocio del dominio

Generación de valores [5 | 9]

- ▼ **Modificador CUBE:** crea y resume todas las posibles combinaciones de los grupos basados en la cláusula GROUP BY
- ▼ Ejemplo:
 - ▼ Por cada producto, mostrar la cantidad ordenada por cada orden, la cantidad ordenada por producto y luego la cantidad total de productos

Modificador CUBE en SQL Server

Generación de valores [6 | 9]

- ▼ Consideraciones de CUBE:
 - ▼ Si hay N columnas en el GROUP BY, se devuelven $2^N - 1$ posibles combinaciones en el *ResultSet*
 - ▼ Los valores nulos indican aquellas filas particulares creadas como resultado del operador CUBE
 - ▼ Se pueden incluir hasta 10 expresiones de agrupamiento

Generación de valores [7 | 9]

- ▼ Consideraciones de CUBE (continuación):
 - ▼ MySQL no soporta este operador
 - ▼ Las columnas que se agrupan deben tener significado dentro del dominio del problema

Generación de valores [8 | 9]

- ▼ **Función GROUPING:** se usa con CUBE y ROLLUP para distinguir entre los valores de resumen y de detalle en el *ResultSet*
- ▼ Devuelve 1 para representar resúmenes de ROLLUP o CUBE y 0 para valores de detalle en el *ResultSet*

Función GROUPING en SQL Server

Generación de valores [9 | 9]

- ▼ Consideraciones de `GROUPING`:
 - ▼ Se producen nuevas columnas en el *ResultSet* que son especificadas en la función `GROUPING`
 - ▼ Se la usa frecuentemente para hacer referencias a estas filas (por ejemplo, poner en negrita esas filas)
 - ▼ MySQL no soporta esta función
 - ▼ Usar `GROUPING` sólo en columnas con `GROUP BY`

Limitación de la cantidad de filas [1 | 4]

- ▼ En un *ResultSet* se puede especificar la cantidad de filas a devolver:
 - ▼ SQL Server:
 - ▼ Cláusula TOP (con o sin la opción TIES)
 - ▼ MySQL:
 - ▼ Cláusula LIMIT

Limitación de la cantidad de filas [2 | 4]

▼ Ejemplo:

- ▼ Mostrar los primeros 10 productos con más órdenes (mostrar el identificador del producto y la cantidad ordenada del mismo)

Ejemplo TOP/LIMIT

- Al usar la cláusula TOP con la opción WITH TIES, también se incluyen en las filas devueltas los valores “cola”, o sea aquellos que tienen el mismo valor que la última fila del TOP, con lo cual la cantidad de filas obtenidas es incierta.

Limitación de la cantidad de filas [3 | 4]

- ▼ Consideraciones:
 - ▼ Se debe usar la cláusula `ORDER BY`, caso contrario el resultado se devuelve en cualquier orden
 - ▼ Usar siempre un entero positivo luego de `TOP/LIMIT`
 - ▼ `TOP` también permite especificar un porcentaje (si este valor resulta en un número fraccionario, `SQL Server` devuelve el siguiente entero próximo)

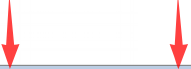
Limitación de la cantidad de filas [4 | 4]

▼ Consideraciones:

- ▼ LIMIT también permite especificar un intervalo: LIMIT 10, 15 (ente caso, se ignoran las 10 primeras filas, y se muestran desde la fila 11 a la 15 inclusive)

Concepto / Ventajas [1 | 7]

Employees (table)				
EmployeeID	LastName	FirstName	Title	...
287	Mensa-Annan	Tete	Mr.	...
288	Abbas	Syed	Mr.	...
289	Valdez	Rachel	NULL	...



vEmployees (view)	
LastName	FirstName
Mensa-Annan	Tete
Abbas	Syed
Valdez	Rachel

- Una vista es una tabla virtual cuyo contenido se define mediante una consulta (permite guardar una consulta predefinida como un objeto en la BD para su uso posterior).
- Salvo pocas excepciones, en una vista se puede usar cualquier sentencia `SELECT`.

Concepto / Ventajas [2 | 7]

- ▼ Salvo que la vista esté indexada, la misma no existe como un conjunto de datos guardados en una BD
- ▼ Las filas y columnas de datos provienen de las tablas referenciadas en la consulta que define la vista, y se generan dinámicamente cuando se la selecciona
- ▼ A las tablas referenciadas en una vista se las llama tablas base

Concepto / Ventajas [3 | 7]

- ▼ Ejemplos de uso de vistas:
 - ▼ Subconjunto de filas o columnas de una tabla
 - ▼ Unión de 2 o más tablas
 - ▼ Join de 2 o más tablas
 - ▼ Subconjunto de otra vista o combinación de vistas de tablas

Concepto / Ventajas [4 | 7]

- ▼ Ventajas:
 - ▼ **Focalizan los datos para los usuarios:**
 - ▼ Se crean vistas para que los usuarios tengan acceso sólo a ciertos datos: los que usa, los importantes
 - ▼ Los datos innecesarios (o inapropiados) quedan fuera de las vistas, creando un nivel de seguridad

Concepto / Ventajas [5 | 7]

- ▼ Ventajas (continuación):
 - ▼ **Enmascaran la complejidad subyacente:**
 - ▼ Crean una capa de abstracción que hace de puente entre el usuario y la complejidad de los datos de la BD
 - ▼ Sirven para simplificar consultas complejas en un número finito de vistas simples

Concepto / Ventajas [6 | 7]

- ▼ Ventajas (continuación):
 - ▼ **Simplifican el manejo de permisos de usuario:**
 - ▼ En vez de dar permisos a los usuarios en las tablas, se da permisos sobre las vistas solamente
 - ▼ **Organizan los datos para exportarlos:**
 - ▼ Se pueden crear vistas basadas en una consulta compleja para “adaptar” los datos a otra aplicación

Resumen de datos Implementación de vistas	Concepto / Ventajas
	Definición de vistas Modificación de datos Consideraciones
Concepto / Ventajas [7 7]	
<ul style="list-style-type: none"> ▼ Tipos de vistas: <ul style="list-style-type: none"> ▼ Estándar (MySQL/SQL Server): combina datos de una o más tablas base. Sólo guarda la definición de la vista ▼ Indexadas (MySQL/SQL Server): vista que se materializó (guarda los datos). Exigen un índice agrupado único ▼ Particionadas (SQL Server): combina datos de una o más tablas base en uno o más servidores 	
Lab. Bases de Datos (EBB) Unidad IV - 2017	32

- Las vistas indexadas mejoran notablemente el rendimiento de ciertas consultas (aquellas donde hay mucho trabajo en cuanto a agrupamiento), pero no resultan muy eficaces cuando las tablas base se modifican frecuentemente.
- Las vistas particionadas pueden ser locales (cuando combinan tablas de la misma instancia) o distribuidas (cuando las tablas base provienen de distintas instancias).

Definición de vistas [1 | 15]

▼ Creación de vistas:

- ▼ Se usa la sentencia `CREATE VIEW`
- ▼ Al crear una vista primero se verifica la existencia de los objetos referenciados
- ▼ Conviene adoptar una convención de nombres para distinguir las tablas de las vistas

Creación de vistas

Definición de vistas [2 | 15]

- ▼ Consideraciones:
 - ▼ Para crear una vista se deben tener ciertos permisos
 - ▼ Se debe tener permiso para seleccionar las tablas subyacentes
 - ▼ A las columnas que forman una vista se les debe proporcionar un nombre si son derivadas de una expresión, función o constante, o cuando se presenten conflictos de nombres por el join de tablas

- Por ejemplo, en SQL Server se debe ser miembro del rol “sysadmin” o “db_owner”, o bien tener el permiso para crear vistas.
- En MySQL se debe tener el permiso “create view”, y el permiso de selección por cada columna que referencia la vista.

Resumen de datos	Concepto / Ventajas
Implementación de vistas	Definición de vistas
	Modificación de datos
	Consideraciones

Definición de vistas [3 | 15]

- ▼ Consideraciones (continuación):
 - ▼ La cantidad de columnas que soporta una vista depende del SGBDR
 - ▼ Las vistas no deben referenciar tablas temporales
 - ▼ En el caso de SQL Server, `CREATE VIEW` debe ser la única sentencia en el *batch*, por lo que no se puede combinar con otra sentencia T-SQL

Lab. Bases de Datos (EBB) | Unidad IV - 2017 35

- Por ejemplo, en SQL Server las vistas no pueden tener más de 1024 columnas.
- Muchos comandos necesitan estar en su propio bloque, o *batch*. En SQL Server, `GO` no es una sentencia SQL, sino un separador de bloques (*batches*). El cliente envía los bloques separados por `GO` al servidor para su procesamiento y espera los resultados. Por lo tanto, `GO` no se envía al servidor: es una palabra reservada del cliente, sólo reconocida por SSMS y la utilidad `osql` (si se usa una herramienta de consulta de terceros, el servidor no reconocerá al `GO` y generará un error).

Definición de vistas [4 | 15]

▼ **Modificación de vistas:**

- ▼ Las vistas se cambian en respuesta a requisitos de los usuarios para ver/ocultar información o cambiar tablas subyacentes
- ▼ Al modificar una vista se mantienen los permisos de la misma
- ▼ Se usa la sentencia `ALTER VIEW`

Modificación de vistas

Definición de vistas [5 | 15]

- ▼ Consideraciones:
 - ▼ La modificación de vistas está sujeta a las mismas restricciones que la creación
 - ▼ Al crear/modificar una vista, se puede especificar la opción `WITH CHECK OPTION`, la cual exige que todas las instrucciones de modificación de datos ejecutadas contra la vista se adhieran a los criterios establecidos en el `SELECT` que define la vista (ver más adelante)

Resumen de datos	Concepto / Ventajas
Implementación de vistas	Definición de vistas
	Modificación de datos
	Consideraciones

Definición de vistas [6 | 15]

- ▼ Consideraciones (continuación):
 - ▼ En SQL Server, al crear/modificar una vista, se puede especificar la opción `WITH ENCRYPTION`, la cual oculta el código de la vista (es importante guardar el *script* de creación para recuperar el código)

Encriptación de vistas

Lab. Bases de Datos (EBB) | Unidad IV - 2017 38

- Al crear o modificar una vista en SQL Server, en la vista `sys.syscomments` se guarda el texto con la estructura de la misma, y por lo tanto, un usuario con ciertos permisos podría ver cómo está hecha la misma.
- MySQL no tiene esta opción.

Definición de vistas [7 | 15]

▼ Borrado de vistas:

- ▼ Se usa la sentencia `DROP VIEW`
- ▼ Se borra tanto la definición como los permisos asignados
- ▼ Si se borra una tabla subyacente, la vista no se borra

Borrado de vistas

Definición de vistas [8 | 15]

- ▼ **Cadena de permisos (SQL Server):**
 - ▼ Las vistas dependen de los objetos subyacentes (vistas y tablas)
 - ▼ Estas dependencias forman una cadena de permisos: si el dueño de una vista es también el dueño de los objetos subyacentes, éste sólo debe dar permisos sobre la vista

Definición de vistas [9 | 15]

▼ Cadena de permisos (continuación):

- ▼ Si el dueño de la vista no es también de los objetos subyacentes, se rompe la cadena de permisos y sólo accederá aquel usuario que tenga permisos sobre la vista y los objetos subyacentes

Definición de vistas [10 | 15]

▼ Cadena de permisos (continuación):

▼ Ejemplo:

- ▼ La tabla "**Tabla1**" tiene por dueño a **Lucía**
- ▼ La vista "**Vista1**" está creada sobre la tabla "**Tabla1**".
Lucía también es dueña de "**Vista1**"
- ▼ La vista "**Vista2**" está creada sobre la vista "**Vista1**".
María es dueña de "**Vista2**" (se rompe la cadena)
- ▼ **María** le asigna a **Pedro** permisos de selección sobre "**Vista2**"

Definición de vistas [11 | 15]

- ▼ **Cadena de permisos (continuación):**
 - ▼ Ejemplo (continuación):
 - ▼ Cuando Pedro consulta “Vista2” puede ocurrir:
 - ▼ Si Lucía le dio permisos a Pedro sobre “Vista1”, Pedro puede acceder sin problemas a “Vista2”
 - ▼ Si Lucía no le dio permisos a Pedro sobre “Vista1”, Pedro no puede acceder a “Vista2” por falta de permisos

Definición de vistas [12 | 15]

- ▼ **Cadena de permisos (continuación):**
 - ▼ En conclusión, la cadena de permisos permite al dueño del objeto original retener el control sobre los usuarios que están autorizados a accederlo

Resumen de datos	Concepto / Ventajas
Implementación de vistas	Definición de vistas
	Modificación de datos
	Consideraciones

Definición de vistas [13 | 15]

- ▼ **Información sobre vistas:**
 - ▼ `INFORMATION_SCHEMA.TABLES`
 - ▼ Nombres de vistas
 - ▼ `INFORMATION_SCHEMA.VIEW_TABLE_USAGE`
 - ▼ Nombres de objetos base
 - ▼ `INFORMATION_SCHEMA.VIEWS`
 - ▼ Definición de vistas

Lab. Bases de Datos (EBB) | Unidad IV - 2017 45

- En SQL Server, las vistas `INFORMATION_SCHEMA` permiten obtener metadatos sobre objetos de una BD. Estas vistas están en la BD master, y se pueden consultar desde cualquier BD.
- En el caso de MySQL, `INFORMATION_SCHEMA` es una BD de sistema, la cual también brinda metadatos sobre objetos de la BD. Dentro de esta BD están las vistas citadas (no todas).
- Al encriptar la vista, su definición no se muestra en la vista `INFORMATION_SCHEMA.VIEWS`.

Resumen de datos	Concepto / Ventajas
Implementación de vistas	Definición de vistas
	Modificación de datos
	Consideraciones

Definición de vistas [14 | 15]

- ▼ **Información sobre vistas:**
 - ▼ INFORMATION_SCHEMA.VIEW_COLUMN_USAGE
 - ▼ Columnas definidas en vistas
 - ▼ Vista (SQL Server) sys.views
 - ▼ Vistas en la BD

Lab. Bases de Datos (EBB) | Unidad IV - 2017 46

- El procedimiento `sp_helptext` está disponible en SQL Server solamente.

Resumen de datos	Concepto / Ventajas
Implementación de vistas	Definición de vistas
	Modificación de datos
	Consideraciones

Definición de vistas [15 | 15]

- ▼ **Información sobre vistas:**
 - ▼ Procedimiento `sp_helptext <nombre_objeto>`
 - ▼ Brinda información sobre la vista (no disponible si la vista tiene encriptada su definición)
 - ▼ Procedimiento `sp_depends <nombre_objeto>`
 - ▼ Informa sobre las dependencias de una vista (es importante ejecutarlo antes de borrar una tabla para que no queden vistas sin referencia)

Información sobre las vistas

Lab. Bases de Datos (EBB) | Unidad IV - 2017 47

- Estos procedimientos están disponibles en SQL Server solamente.

Modificación de datos [1 | 3]

- ▼ A menos que la vista esté indexada, la misma no mantiene una copia separada de los datos (muestra los resultados de una consulta en una o más tablas)
- ▼ Al modificar datos en una vista, se modifica la tabla base
- ▼ Con algunas limitaciones, se puede insertar, modificar o borrar datos de una tabla desde una vista

Modificación de datos [2 | 3]

▼ Restricciones:

- ▼ Las modificaciones no pueden afectar a más de una tabla subyacente
- ▼ Ciertas columnas no pueden modificarse, como las que son resultados de un cálculo de valores, funciones de agrupamiento o funciones de SQL
- ▼ Las modificaciones causan error si afectan columnas que no son referenciadas en la vista

Resumen de datos	Concepto / Ventajas
Implementación de vistas	Definición de vistas
▼	Modificación de datos
	Consideraciones

Modificación de datos [3 | 3]

- ▼ Restricciones (continuación):
 - ▼ Cuando se modifica, si se habilita la opción `WITH CHECK OPTION`, se verifican las sentencias contra la definición de la vista

Lab. Bases de Datos (EBB) | Unidad IV - 2017

50

- Si por ejemplo se crea una vista con todos los empleados cuyo código postal sea 4000, y luego a través de la vista se los actualiza cambiándoles el código, todos estos empleados desaparecerían de la vista. Si la vista se crea con la opción `WITH CHECK OPTION`, cada modificación o inserción a través de la vista que produzca que desaparezca un registro de la misma genera un error.
- Esta opción está disponible en los 2 motores.

Consideraciones [1 | 1]

- ▼ Cuando las vistas que realizan joins de muchas tablas y evalúan expresiones complejas son anidadas con otras vistas, en caso de surgir un problema puede resultar difícil determinar su origen
- ▼ Es recomendable transformar las vistas anidadas en joins (mientras sea posible) y hacer un mapa de dependencias de las vistas y sus objetos subyacentes
- ▼ No se recomienda más de 3 niveles de anidamiento

Resumen [1 | 1]

- ▼ Uso de funciones de agrupamiento
- ▼ Fundamentos de `GROUP BY` / `HAVING`
- ▼ Generación de valores (`ROLLUP` / `CUBE`)
- ▼ Limitación de la cantidad de filas (`TOP` / `LIMIT`)
- ▼ Concepto / ventajas de las vistas
- ▼ Definición de vistas
- ▼ Modificación de datos desde una vista
- ▼ Consideraciones