FUNDAMENTOS DE BASES DE DATOS

Algebra Relacional

Allan Murillo
Marlen Treviño
Yessenia Calvo

Agenda

- Algebra relacional
- Seguridad

ALGEBRA RELACIONAL

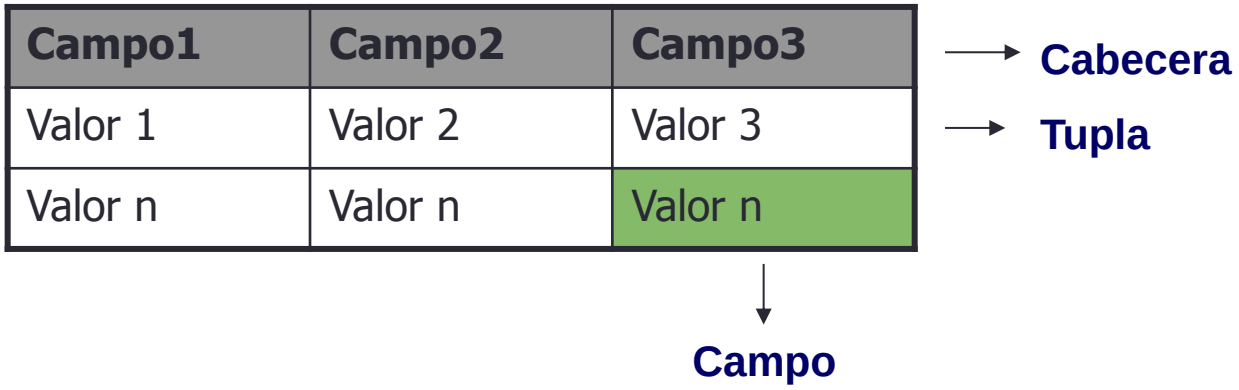
Algebra Relacional

- Se denomina algebra relacional a un conjunto de operaciones encargadas de la manipulación de datos agrupados (relaciones).
- Estas operaciones describen la manipulación de datos. Son en si, una representación intermedia de una consulta a una base de datos.
- Debido a sus propiedades algebraicas, estas operaciones sirven para obtener una versión más optimizada y eficiente de dicha consulta.

Algebra Relacional

Términos necesarios de entender.

- Relación: Se le denomina relación a un conjuntos de datos ordenados en forma de filas y columnas, los cuales están relacionados por algún contexto. Estas relaciones contienen dentro de sus partes a la cabecera, tuplas y campos.



Algebra Relacional

- Tupla: Es una parte de una relación, un conjunto de datos que entregan una información relacionada. Enfocado a las bases de datos puede llamarse “registro” ó “filas de una tabla”.

Campo1	Campo2	Campo3
Valor 1	Valor 2	Valor 3
Valor n	Valor n	Valor n

→ Cabecera

→ Tupla

- Aridad: Se le denomina aridad al numero de atributos que contiene una relación.
- Unión compatible: Se le denomina así a la posibilidad que tengan dos relaciones de tener la misma aridad.

Operadores

- Según su procedencia

De la teoría de conjuntos		Propios relacionales	
∪	Unión	σ	Selección
∩	Intersección	π [ ]	Proyección
~ -	Diferencia	∞ ⋈	Concatenación (Join)
×	Producto cartesiano	÷	División

Operadores

- Según su complejidad

Básicos o primitivos		Derivados	
∪	Unión	∩	Intersección
~ -	Diferencia	∞ ⋈	Concatenación (Join)
×	Producto cartesiano	÷	División
σ	Selección		
π [ ]	Proyección		

Operador: Selección

- Permite seleccionar un subconjunto de tuplas de una relación (R), todas aquellas que cumplan la(s) condición(es) P:
- $\sigma_{P}(R)$
- Por Ej: $\sigma_{Apellido=Gomez}(Alumno)$

Operador: Selección

- Ejemplo – Tabla Préstamos

número-préstamo	*nombre-sucursal*	*importe*
P-15	Navacerrada	1.500
P-16	Navacerrada	1.300

  - Seleccionar las tuplas de la relación *préstamo en* que la sucursal es «Navacerrada»
    - $\sigma_{\textit{nombre-sucursal} = \textit{«Navacerrada»}}(\textit{préstamo})$
  - Las tuplas en las que el importe prestado sea mayor que 1.200 €
    - $\sigma_{\textit{importe}>1200}(\textit{préstamo})$

Operador: Proyección

- Permite extraer columnas (atributos) de una relación, dando como resultado un subconjunto vertical de atributos de la relación
  - $\Pi_{A1, A2,\ldots, An}(R)$
- Por Ej:
  - $\Pi_{\text{Apellido, Nombre, Numero\_Registro}}(\text{Alumno})$
  - $\Pi_{\text{(num\_cliente, nombre)}} (\sigma_{\textit{(num\_cliente=2)\&\&(lim\_credito=50000)}} \textit{(Cliente)})$

Operador: Unión

- Retorna el conjunto de tuplas que están en R, o en S, o en ambas. R y S deben ser relaciones compatibles:
  - R U S

- Por Ej:
  - Alumno U Profesor
  - Π (num_carne) (Matricula) **U** Π (num_carne) (Becas)

- Condiciones a cumplir
  - Las relaciones de r y s deben tener la misma aridad

Operador: Unión

- Se desea averiguar todos los clientes que tienen una cuenta, un préstamo o ambos:

Cuenta

N_Cliente	#Cuenta
Santos	C-101
Gómez	C-215
López	C-102
Abril	C-305
González	C-201
Santos	C-217
Rodríguez	C-222

Prestamo

N_Cliente	#Prestamo
Santos	P-17
Gómez	P-23
López	P-15
Soto	P-14
Pérez	P-93
Gómez	P-11
Fernández	P-16

Operador: Unión

- Se desea averiguar todos los clientes que tienen una cuenta, un préstamo o ambos:

$$\pi^{N\_Cliente}(Cuenta) \cup \pi^{N\_Cliente}(Prestamo)$$

N_Cliente
González
Santos
Rodríguez
López
Abril
Soto
Pérez
Gómez
Fernández

Operador: Diferencia

- Entrega todas aquellas tuplas que están en R, pero no en S. R y S deben ser relaciones compatibles:
  - R – S

- Por ejemplo:
  - Π (num_carne) (Matricula) **-** Π (num_carne) (Becas)

- Condiciones a cumplir
  - Las relaciones de r y s deben tener la misma aridad

Operador: Diferencia

- Se desea averiguar todos los clientes que tienen abierta una cuenta, pero que no tienen concedido ningún préstamo:

Cuenta

N_Cliente	#Cuenta
Santos	C-101
Gómez	C-215
López	C-102
Abril	C-305
González	C-201
Santos	C-217
Rodríguez	C-222

Prestamo

N_Cliente	#Prestamo
Santos	P-17
Gómez	P-23
López	P-15
Soto	P-14
Pérez	P-93
Gómez	P-11
Fernández	P-16

Operador: Diferencia

- Se desea averiguar todos los clientes que tienen abierta una cuenta, pero que no tienen concedido ningún préstamo:

$$\pi_{N\_Cliente}(Cuenta) - \pi_{N\_Cliente}(Prestamo)$$

N_Cliente
González
Rodríguez
Abril

Operador: Intersección

- La intersección, como en Teoría de conjuntos, corresponde al conjunto de todas las tuplas que están en R y en S, siendo R y S relaciones compatibles:
  - R ∩ S
  - r ∩ s = r – (r-s)

- Por ejemplo
  - π(num_carne) (Matricula) ∩ π(num_carne) (Becas)

- Condiciones a cumplir
  - Las relaciones de *r* y *s* deben tener la misma aridad

Operador: Intersección

- Se desea averiguar todos los clientes que tienen abierta una cuenta y tienen concedido un préstamo:

Cuenta

N_Cliente	#Cuenta
Santos	C-101
Gómez	C-215
López	C-102
Abril	C-305
González	C-201
Santos	C-217
Rodríguez	C-222

Prestamo

N_Cliente	#Prestamo
Santos	P-17
Gómez	P-23
López	P-15
Soto	P-14
Pérez	P-93
Gómez	P-11
Fernández	P-16

Operador: Intersección

- Se desea averiguar todos los clientes que tienen abierta una cuenta y tienen concedido un préstamo:

$$\pi_{N\_Cliente}(Cuenta) \cap \pi_{N\_Cliente}(Prestamo)$$

N_Cliente
Santos
Gómez
López

Operador: Producto cartesiano

- Entrega una relación, cuyo esquema corresponde a una combinación de todas las tuplas de R con cada una de las tuplas de S, y sus atributos corresponden a los de R seguidos por los de S:
  - R x S

- Por ejemplo:
  - Cliente x Direcciones
  - **Esquema de la Relación**
    - (cliente.num_cliente, cliente.saldo, cliente.descuento, cliente.lim_saldo, direcciones.Nº, direcciones.Calle, Comunidad, direcciones.Ciudad)

  - **Esquema de la Relación Simplificada**
    - (num_cliente, saldo, descuento, lim_saldo, Nº, Calle, Comunidad,Ciudad)

Operador: Concatenación

- Outer Join
  - Es una variante del Join en la que se intenta mantener toda la información de los operandos, incluso para aquellas filas que no participan en el Join
  - Se “rellenan con nulos” las tuplas que no tienen correspondencia en el Join
  - Tres variantes
    - Left
      - se tienen en cuenta todas las filas del primer operando
    - Right
      - se tienen en cuenta todas las filas del segundo operando
    - Full
      - se tienen en cuenta todas las filas de ambos operandos

Operador: Concatenación

- Hace un producto cartesiano de sus dos argumentos y realiza una selección forzando la igualdad de atributos que aparecen en ambas relaciones, eliminando repetidos:
  - R ⋈ S o R * S
- También se conoce como JOIN

R1

E#	Nombre	D#
320	José	D1
322	Rosa	D3
•	María	D3
•	José	D5

R2

D#	Descrip
D1	Central
D3	I+D
D4	Ventas

R1 * R2

E#	Nombre	D#	Descrip
320	José	D1	Central
322	Rosa	D3	I+D
•	María	D3	I+D

Operador: Concatenación

R1

E#	Nombre	D#
320	José	D1
322	Rosa	D3
•	María	D3
•	José	D5

R2

D#	Descrip
D1	Central
D3	I+D
D4	Ventas

R1 * R2

E#	Nombre	D#	Descrip
320	José	D1	Central
322	Rosa	D3	I+D
•	María	D3	I+D

R1 * LEFT R2

E#	Nombre	D#	Descrip
320	José	D1	Central
322	Rosa	D3	I+D
•	María	D3	I+D
•	José	D5	null

R1 * RIGHT R2

E#	Nombre	D#	Descrip
•	José	D1	Central
322	Rosa	D3	I+D
•	María	D3	I+D
null	null	D4	Ventas

R1 * FULL R2

E#	Nombre	D#	Descrip
320	José	D1	Central
322	Rosa	D3	I+D
•	María	D3	I+D
•	José	D5	null
null	null	D4	Ventas

SQL y Algebra relacional

SQL	Algebra relacional
SELECT	Proyección π
FROM	Definición de Relaciones (renombramiento, reunión natural, etc.)
WHERE	Selección σ

SQL y Algebra relacional

Operador	Algebra relacional	SQL
Selección	R DONDE F	SELECT ... FROM R WHERE F
Proyección	R [A_i , A_j ..., A_k]	SELECT A_i , A_j ..., A_k FROM R
Producto Cartesiano	R_1 x R_2, ... x R_n	SELECT ... FROM R_1, R_2, ..., R_n, o SELECT...FROM R_1 JOIN R_2, ..., JOIN R_n
Concatenación	$R_1 \bowtie R_2$	SELECT... FROM R_1 NATURAL JOIN R_2
Unión	$R_1 \cup R_2$	SELECT * FROM R_1 UNION SELECT * FROM R_2
Diferencia	$R_1 - R_2$	SELECT * FROM R_1 EXCEPT SELECT * FROM R_2
Intersección	$R_1 \cap R_2$	SELECT * FROM R_1 INTERSECT SELECT * FROM R_2

Ejemplos

- Si tenemos las siguientes tablas
  - Tabla personas, con la clave primaria "per“

per	nombre	apellido1	apellido2	dep
1	ANTONIO	PEREZ	GOMEZ	1
2	ANTONIO	GARCIA	RODRIGUEZ	2
3	PEDRO	RUIZ	GONZALEZ	2

  - Tabla "departamentos", con la clave primaria "dep"

dep	departamento
1	ADMINISTRACION
2	INFORMATICA
3	COMERCIAL

Ejemplos

- Si queremos saber los nombres de las personas que trabajan en INFORMATICA
  - Inner join
    - SELECT nombre, apellido1, departamento FROM personas INNER JOIN departamentos ON personas.dep = departamentos.dep
  - Left join
    - SELECT nombre, apellido1, departamento FROM personas LEFT JOIN departamentos ON personas.dep = departamentos.dep
  - Right join
    - SELECT nombre, apellido1, departamento FROM personas RIGHT JOIN departamentos ON personas.dep = departamentos.dep

Ejemplos

- Tabla "personas_empresa1“

per	nombre	apellido1	apellido2
1	ANTONIO	PEREZ	GOMEZ
2	ANTONIO	GARCIA	RODRIGUEZ
3	PEDRO	RUIZ	GONZALEZ

- Tabla "personas_empresa2"

per	nombre	apellido1	apellido2
1	JUAN	APARICIO	TENS
2	ANTONIO	GARCIA	RODRIGUEZ
3	LUIS	LOPEZ	VAZQUEZ

Ejemplos

SELECT nombre, apellido1 FROM personas_empresa1 UNION SELECT nombre, apellido1 FROM personas_empresa2

nombre	apellido1
ANTONIO	PEREZ
ANTONIO	GARCIA
PEDRO	RUIZ
JUAN	APARICIO
LUIS	LOPEZ

Ejemplos

Asignatura

CodA	NombreA	Precio
1	Programación	15000
2	Dibujo técnico	20000
3	Inglés	18000

Alumnos

Nmat	Nombre	Apellidos	Dirección
0338	Ana	Pérez Gómez	Florencia
0254	Rosa	López López	Muelle
0168	Juan	García García	La Vega

Notas

Nmat	CodA	Convocatoria	Nota
0338	1	Feb 02	8
0254	2	Feb 02	5
0168	2	Feb 02	3
0338	2	Feb 02	5
0338	3	Jun 02	7
0254	1	Jun 02	6
0168	1	Jun 02	9
0168	3	Jun 02	5

Ejemplos

- Obtener los nombres de todas las asignaturas

  $\pi^{nombreA}$ (Asignatura)

  Select nombreA from Asignatura

- Obtener los apellidos y teléfonos de los alumnos de nombre Rosa

  $\pi^{apellidos,\ telefono}$ ($\sigma^{nombre='Rosa'}$ (Alumnos))

  Select apellidos, telefono from Alumnos where nombre='Rosa'

- Obtener todas las notas del CodA 1 y de la convocatoria de Junio de 2002

  π^{nota} (σ CodA=1 ∧ convocatoria='Jun 02' (Notas))

  Select nota from Notas where CodA=1 and convocatoria = 'Jun 02'

Ejemplos

- Obtener el nombre de las asignaturas impartidas en la convocatoria de Junio de 2002.

  $\pi^{nombreA}(\sigma^{convocatoria='Jun\ 02'}(Notas) * Asignatura)$

  SELECT nombreA FROM Notas **INNER JOIN** Asignatura **ON** Notas.Nmat = Asignatura.Nmat where Asignatura.convocatoria='Jun 02'

- *Obtener el identificador de los alumnos que figuren matriculados en las asignaturas de Inglés y Dibujo*

  $\pi_{Nmat}(\sigma_{nombreA='Ingles'}(Asignatura) * Notas) \cap$

  $\pi_{Nmat}(\sigma_{nombreA='Dibujo'}(Asignatura) * Notas)$

  SELECT Nmat FROM Asignatura INNER JOIN Notas ON Asignatura.codA = Notas.codA where Asignatura.nombreA='Ingles' **UNION** SELECT Nmat FROM Asignatura INNER JOIN Notas ON Asignatura.codA = Notas.codA where Asignatura.nombreA='Dibujo'

- *Obtener el identificador de los alumnos que no han perdido ninguna asignatura*

  $\pi_{Nmat}(\sigma_{nota>=5}(Notas)) - \pi_{Nmat}(\sigma_{nota<5}(Notas))$

  SELECT Nmat FROM Notas where nota>=5 **EXCEPT** SELECT Nmat FROM Notas where nota<5

Secuencia

	Sección	Secuencia
1	SELECT	
2	DISTINCT	
3	TOP	
4	FROM	
5	CROSS	
6	INNER	
7	OUTER	
8	ON	
9	WHERE	
10	GROUP BY	
11	HAVING	
12	ORDER BY	

Secuencia

	Sección	Secuencia
1	SELECT	9
2	DISTINCT	10
3	TOP	11
4	FROM	1
5	CROSS	2
6	INNER	3
7	OUTER	5
8	ON	4
9	WHERE	6
10	GROUP BY	7
11	HAVING	8
12	ORDER BY	12

- Taller

SEGURIDAD

El problema de la seguridad

- **Aspectos del problema de la seguridad**
- El ámbito de la seguridad de las bases de datos es **amplio**, pues abarca aspectos relacionados con hardware, software, personas y datos:
- **Legales, sociales y éticos**
  - ¿El solicitante tiene el derecho legal de obtener determinada información? (saldos de cuentas de clientes)
  - ¿Cómo asegurar que no se revelen datos confidenciales a cambio de sobornos u otros favores?
- **Política gubernamental, institucional o corporativa**
  - ¿Qué información no debe estar disponible al público? (historiales médicos...)
  - ¿Cómo se decide 'quién' puede acceder a 'qué'?
- **Controles físicos**
  - ¿Cómo proteger físicamente contra intrusos las salas en donde están los sistemas informáticos?

El problema de la seguridad

- **Aspectos del problema de la seguridad**
- **Seguridad del Sistema Operativo**
  - ¿Borra el SO el contenido de áreas de almacenamiento y archivos de datos cuando no se necesitan?
  - ¿El SO permite el acceso directo a los ficheros de la base de datos?
  - si se usan contraseñas para el acceso al SO ¿cómo se mantienen en secreto? ¿Con qué frecuencia se cambian?
- **Seguridad de la Red**
  - La seguridad en el nivel de software de red es hoy en día fundamental, tanto en Internet como en las redes privadas de las organizaciones
- **Aspectos específicos del Sistema de Bases de Datos ☆**
  - ¿Dispone el sistema de BD del concepto de propiedad de la información?
  - ¿Cómo evitar que los usuarios puedan acceder a ciertas partes de la BD?
  - ¿Cómo limitar las operaciones que los usuarios pueden realizar sobre la BD?

- **Hay que adoptar medidas de seguridad en todos estos niveles, pero nos centraremos en el último de ellos (☆)**

El problema de la seguridad

- La **protección total y absoluta de la BD** contra el mal uso intencionado es **imposible**

  - El SGBD proporciona técnicas para que (grupos de) usuarios tengan acceso a ciertas partes de la base de datos, sin tener acceso al resto
    - El **Subsistema de Seguridad y Autorización** del SGBD garantiza la seguridad de (partes de) la BD contra accesos no autorizados
- El SGBD tiene definido el **objeto de datos**:
  - Unidad de datos que requiere protección individual
  - Puede ser desde la base de datos completa, o un conjunto de tablas ... hasta una posición (fila, columna) dentro de cierta tabla

Control de acceso

- El **administrador de la base de datos**, ABD, es la autoridad central responsable de la seguridad global del sistema de bases de datos
- Dispone de una cuenta de bases de datos privilegiada o de sistema, con «capacidades extraordinarias», desde la que puede...
  - **Crear** y **eliminar** cuentas de usuario para acceso a la base de datos
  - **Conceder** y **cancelar** privilegios a (cuentas de) usuarios
  - **Asignar** datos a niveles de seguridad
  - **Asignar** cuentas de usuario a niveles de seguridad o acreditación

◂ **Control Global**

◂ **Control Discrecional**

◂ **Control Obligatorio**

Control de acceso global

- Evitar que personal no autorizado acceda al sistema de BD
- Puesto en práctica mediante creación, por parte del ABD, de cuentas de usuario de BD con contraseñas
- **Implementación**
  - **Tabla cifrada** con dos columnas: cuenta y contraseña
  - **Almacenada en** el **INFORMATION_SCHEMA** del catálogo y mantenida por el SGBD
- **Autenticación de usuarios**
  - Para entrar al sistema, el usuario indica al SGBD su cuenta y contraseña
  - Una vez que el SGBD valida esos datos, el usuario puede acceder a la información almacenada en la BD

- ⓘ un programa de aplicación puede 'ser un usuario' (puede exigírsele contraseña)

Control de acceso obligatorio

- Para establecer una **seguridad multinivel s**e suele combinar con el control de acceso discrecional
  - Aunque la mayoría de SGBD sólo ofrecen el discrecional
- **Clasificación de los datos** y **usuarios** en **niveles de seguridad**
  - Cada **objeto** de datos es etiquetado con un **nivel de seguridad**
  - Cada **usuario** se asigna a un **nivel de acreditación**
  - Cada **objeto** de datos puede ser **accedido** sólo **por usuarios con** la **acreditación apropiada**

Control de acceso discrecional

- Soportado por la mayoría de los SGBD comerciales
- Basado en **privilegios**
- Un privilegio es un derecho de acceso o autorización para realizar determinada **operación** sobre ciertos objetos de BD
  - Un **usuario** puede tener **diversos privilegios** sobre **distintos objetos**
    - Privilegio SELECT sobre tablas
  - Diferentes **usuarios** pueden tener **privilegios distintos sobre** un **mismo objeto**
    - U1 con privilegios SELECT, INSERT y DELETE sobre una tabla, mientras que U2 tiene sólo el privilegio SELECT
- **Clases de privilegios**
  - Privilegios de **nivel de cuenta** y
  - Privilegios de **nivel de objeto** de base de datos

Control de acceso discrecional

- **Privilegios de nivel de cuenta**
- **Privilegios** particulares **de cada usuario** (cuenta), independientemente de los objetos de BD existentes
- Tipos de **privilegios**
  - CREATE SCHEMA, DROP SCHEMA,
  - CREATE TABLE, ALTER TABLE, DROP TABLE,
  - CREATE VIEW, DROP VIEW,
  - CREATE TYPE, ALTER TYPE, DROP TYPE,
  - SELECT, INSERT, UPDATE, DELETE (sobre cualquier tabla)
- **Privilegios de nivel de objeto de BD**
  - Permiten especificar **qué usuarios tienen qué privilegios sobre qué objetos** concretos

Seguridad en SQL

- **Propietario de un objeto** de base de datos
  - El **usuario que lo ha creado** (= cuenta de BD en la que fue creado)
  - Posee **todos** los **privilegios** (posibles) sobre el objeto,
  - y capacidad de **conceder** tales privilegios **a otros** usuarios (GRANT)
  - e incluso propagar dicha capacidad de concesión (GRANT OPTION)

- **Concesión de privilegios (de nivel de objeto)**
  - **GRANT <privilegios> ON <objetos> TO <sujetos> [WITH GRANT OPTION]**
  - Especifica **qué operaciones pueden realizar sobre qué objetos ciertos usuarios**
  - GRANT **SELECT** ON Fotografo TO julia;
  - GRANT **UPDATE(cuota)** ON Editorial TO julia, ruben;
  - GRANT **INSERT** ON Reportaje TO cristina WITH GRANT OPTION;
  - GRANT **DELETE** ON Exclusiva TO eva;

Seguridad en SQL

- Ejemplo #1

```
USE erolinea;
GO

CREATE LOGIN profesora
WITH PASSWORD = '123' ,
DEFAULT_DATABASE=aerolinea
GO

/*ALTER LOGIN profesora
WITH PASSWORD = 'profesora'
GO*/

/*DROP LOGIN profesora
GO*/
CREATE USER profesora
FOR LOGIN profesora;
GO
```

Seguridad en SQL

- **Tipos de privilegios** (de nivel de objeto)
  - **SELECT** ver toda columna de cierta tabla, incluso si ha sido añadida después de haber sido creada. NO ES POSIBLE INDICAR SÓLO ALGUNAS COLUMNA.
  - **UPDATE** sobre una tabla concreta, quizá de sólo algunas columnas
  - **INSERT** sobre una tabla concreta, quizá con valores para sólo algunas columnas
  - **DELETE** filas de cierta tabla
  - **REFERENCES** permite hacer referencia a (columnas concretas de) cierta tabla mediante Restricciones de Integridad de cualquier tipo, no sólo RI Referencial
  - **USAGE** uso de ciertos dominios
  - **ALL PRIVILEGES** todos los que tiene sobre el objeto el usuario que concede (que ejecuta GRANT)

Seguridad en SQL

- **Cancelación de privilegios** (nivel de objeto)
  - **REVOKE** [**GRANT OPTION FOR**] <privilegios> **ON** <objetos>
  - **FROM** <sujetos> { **RESTRICT** | **CASCADE** }

  – También ‘revocación’ o ‘denegación’ de privilegios

  - REVOKE **SELECT** ON Fotografo FROM julia;
  - REVOKE **UPDATE** ON Editorial FROM julia, ruben;
  - REVOKE **INSERT** ON Reportaje FROM cristina;
  - REVOKE **DELETE** ON Exclusiva FROM eva;

  - REVOKE **ALL PRIVILEGES** ON <objetos> FROM <sujetos>

  – Cancela todos los privilegios que el usuario que ejecuta la sentencia concedió a los sujetos indicados en el FROM

Ejercicios

- Cree un usuario que tenga acceso de consulta a las tablas piloto y destinos
- Cree un usuario que tenga acceso de insertar, modificar, borrar y consultar la tabla programacion

Seguridad en SQL

- **Uso de vistas como mecanismo de seguridad**
  - Una vista es una alternativa para mostrar datos.
  - Una vista se puede ver como una tabla virtual.
  - Los datos en la vista pueden o no estar almacenados en la BD.
  - Una vista puede ser un subconjunto del total de datos.
  - Permiten ocultar Datos.
  - Simplifican administración de usuarios.
  - Mejoran el rendimiento.

Seguridad en SQL

- **Uso de vistas como mecanismo de seguridad**
- Creación de Vistas
  - Sintaxis
  - CREATE VIEW NombreVista AS SentenciaSELECT.
- Modificación de Vista
  - ALTER VIEW NombreVista AS SentenciaSELECT.
- Borrado de Vista
  - DROP VIEW NombreVista.

Seguridad en SQL

- **Uso de vistas como mecanismo de seguridad**
- El usuario u1 es propietario de la tabla R(a1,a2,a3,a4,a5)
- Si u1 desea que otro usuario u2 pueda leer **sólo algunas columnas** a1,a2,a3 de R
  - CREATE VIEW **V** AS **SELECT a1, a2, a3 FROM R**;
  - GRANT **SELECT** ON **V** TO **u2**;
- Si u1 desea que u2 lea **sólo algunas filas** de R, las que satisfacen cierta condición Q
  - CREATE VIEW **W** AS **SELECT * FROM R WHERE Q**;
  - GRANT **SELECT** ON **W** TO **u2**;

Para poder crear una vista, el usuario debe poseer el privilegio SELECT sobre cada tabla base de la vista, además del privilegio de cuenta CREATE VIEW

Seguridad en SQL

- **Uso de vistas como mecanismo de seguridad**
  - Ejemplo #2

```
/*SELECT *
FROM SYS.SYSLOGINS*/

GRANT SELECT
ON vuelo
TO profesora
go

--DENY para denegar permisos
DENY CREATE PROCEDURE
TO profesora
```

Seguridad en SQL

- **Uso de vistas como mecanismo de seguridad**
  - Ejemplo #2

```
CREATE VIEW consulta
AS
SELECT nombre, monto_base FROM T_Componente
go
            SELECT * FROM CONSULTA
go

--DROP FUNCTION dbo.fn_NumberPlus10
--GO

CREATE FUNCTION dbo.fn_NumberPlus10 (@Number INT)
RETURNS INT
             --NO MUESTRA EL CODIGO
             WITH ENCRYPTION
AS
BEGIN
            RETURN @Number + 10
END
GO

GRANT SELECT
ON fn_NumberPlus10
TO profesora
go

SELECT dbo.fn_NumberPlus10(25)
```

Seguridad en SQL

- Ejercicios vistas
  - Cree una vista que retorne la licencia, nombre y horas vuelo de un piloto, la vista debe recibir el nombre.
  - Cree una vista que retorne el TOP 5 de los pilotos que tienen menor horas vuelo.
  - Modifique los permisos de esta vista para que solamente la pueda ver un usuario.

Seguridad en SQL

- Roles
  - Son como los *grupos* del sistema operativo Microsoft Windows
  - Tipos
    - Roles a nivel de servidor
    - Roles a nivel de base de datos

Seguridad en SQL

- Roles a nivel de servidor
  - Los roles de nivel de servidor también se denominan *roles fijos de servidor* porque no se pueden crear nuevos roles de nivel de servidor.

Rol	Descripción
sysadmin	Pueden realizar cualquier actividad en el servidor.
serveradmin	Pueden cambiar las opciones de configuración en el servidor y apagarlo.
securityadmin	Administran los inicios de sesión y sus propiedades. Pueden administrar los permisos de servidor GRANT, DENY y REVOKE. También, pueden administrar los permisos de nivel de base de datos GRANT, DENY y REVOKE si tienen acceso a una base de datos. Asimismo, pueden restablecer las contraseñas para los inicios de sesión de SQL Server.
processadmin	Pueden finalizar los procesos que se ejecuten en una instancia de SQL Server.
setupadmin	Pueden agregar y quitar servidores vinculados.
diskadmin	Se usa para administrar archivos de disco.
dbcreator	Pueden crear, modificar, quitar y restaurar cualquier base de datos.
public	Cada inicio de sesión de SQL Server pertenece al rol public de servidor. Cuando a una entidad de seguridad de servidor no se le han concedido ni denegado permisos específicos para un objeto protegible, el usuario hereda los permisos concedidos al rol public en ese objeto.

Seguridad en SQL

- Roles a nivel de base de datos
  - Existen dos tipos de roles de nivel de base de datos en SQL Server: los *roles fijos de base de datos*, que están predefinidos en la base de datos, y los *roles flexibles de base de datos*, que pueden crearse.

Seguridad en SQL

- Roles fijos a nivel de base de datos

Rol	Descripción
db_owner	Pueden realizar todas las actividades de configuración y mantenimiento en la base de datos y también pueden quitar la base de datos.
db_securityadmin	Pueden modificar la pertenencia a roles y administrar permisos.
db_accessadmin	Pueden agregar o quitar el acceso a la base de datos para inicios de sesión de Windows, grupos de Windows e inicios de sesión de SQL Server.
db_backupoperator	Pueden crear copias de seguridad de la base de datos.
db_ddladmin	Pueden ejecutar cualquier comando del lenguaje de definición de datos (DDL) en una base de datos.
db_datawriter	Pueden agregar, eliminar o cambiar datos en todas las tablas de usuario.
db_datareader	Pueden leer todos los datos de todas las tablas de usuario.
db_denydatawriter	No pueden agregar, modificar ni eliminar datos de tablas de usuario de una base de datos.
db_denydatareader	No pueden leer datos de las tablas de usuario dentro de una base de datos.

Ejemplo

- Revisar los roles de servidor del usuario sa y del usuario profesora (SELECT * FROM SYS.SYSLOGINS)

```
CREATE LOGIN duenno
WITH PASSWORD = '123' ,
DEFAULT_DATABASE=aerolinea
GO

/*DROP LOGIN duenno
GO*/

USE aerolinea;
GO
CREATE USER duenno
FOR LOGIN duenno;
GO

USE [aerolinea]
GO
ALTER ROLE [db_datawriter] ADD MEMBER [duenno]
GO
USE [aerolinea]
GO
ALTER ROLE [db_denydatareader] ADD MEMBER [duenno]
GO

--insert into T_Materia (nombre, estado) values ('Italiano',1)
--select * from T_Materia
```

Copias de seguridad

- **Copia de seguridad completa**
  - Contiene todos los datos en una base de datos específica o un conjunto de grupos de archivos o archivos, y también suficiente registro para permitir la recuperación de los datos.
- **Copia de seguridad diferencial**
  - Se basa en la última copia de seguridad completa de los datos. Una copia de seguridad diferencial incluye sólo los datos que ha cambiado desde la base diferencial.
- **Copia de seguridad del registro de transacciones**
  - Cada copia de seguridad de registro cubre la parte del registro de transacciones que estaba activa cuando la copia de seguridad fue creada, e incluye todos los registros que no fueron respaldados en una copia de seguridad de registros anterior.

Copias de seguridad

BACKUP DATABASE { nombre_base_datos | @var_nombre_base_datos }
TO <dispositivo_backup> [,...n]
[<MIRROR TO clause>] [next-mirror-to]
[WITH { DIFFERENTIAL | <general_WITH_options> [,...n] }]
[;]

Ver ejemplo

Restauración de una copia de seguridad

- Es un proceso que restaura los datos de una o más copias de seguridad y se recupera la base de datos cuando la última copia de seguridad se restaura.
- Tipos
  - Restaurar una base de datos completa a partir de una copia de seguridad completa de la base de datos (restauración completa).
  - Restaurar parte de una base de datos (restauración parcial).
  - Restaurar archivos o grupos de archivos en una base de datos.
  - Restaurar páginas específicas en una base de datos (restauración de páginas).
  - Restaurar un registro de transacciones en una base de datos (restauración del registro de transacciones).
  - Revertir una base de datos al punto temporal capturado por una instantánea de la base de datos.

Restauración de una copia de seguridad

```
RESTORE DATABASE { nombre_base_datos | @var_nombre_base_datos }
[ FROM <backup_device> [ ,...n ] ]
[ WITH
{
        [ RECOVERY | NORECOVERY | STANDBY =
                {standby_file_name | @standby_file_name_var }
        ]
```

Restauración de una copia de seguridad

Ejemplo #4

```
BACKUP DATABASE [aerolinea]
TO  DISK = N'D:\aerolinea.Bak'
WITH FORMAT, NAME = N'aerolinea-Full Database Backup',
SKIP, NOREWIND, NOUNLOAD,  STATS = 10
GO

RESTORE DATABASE [aerolinea] FROM  DISK = N'D:\aerolinea.Bak'
WITH  FILE = 1,  NOUNLOAD,  REPLACE,  STATS = 10
GO
```

Ejercicios

- Cree una nueva base de datos (debe contener al menos 3 tablas)
- Cree tres nuevos usuarios para dicha bases de datos: uno solamente podrá realizar escritura de datos, otro solamente podrá consultar datos y el otro solamente podrá realizar respaldos
- Utilice el usuario creado para hacer respaldos y cree un respaldo de la base de datos que usted creo y luego realice una restauración de la misma en una nueva base de datos