

Modificación de la Base de Datos

- Hasta ahora hemos visto como obtener información desde la Base de Datos.
- Es necesario entregar herramientas para poder alterar el estado de la Base de Datos.
- Eliminación, Inserción, Actualización son algunas de las tareas necesarias, usaremos el álgebra relacional.

Eliminación

- Usamos el operador de diferencia ($-$) y asignación (\leftarrow)

$$r \leftarrow r - E$$

donde r es una relación y E una expresión del álgebra relacional.

- Eliminar los datos de Soto

$$empleado \leftarrow empleado - \sigma_{nombre=Soto}(empleado)$$

- Eliminar todos los empleados del departamento de Marketing

$$r1 \leftarrow \sigma_{dep.nombre=Marketing}(empleado \bowtie departamento)$$

$$r2 \leftarrow \pi_{emp.nombre, sueldo, emp.cod_depto, fecha_ing}(r1)$$

$$empleado \leftarrow empleado - r2$$

<u>nombre</u>	<u>sueldo</u>	<u>cod_dept</u>	<u>fecha_ing</u>
Torres	\$ 1.200.000	A1	01/01/2004
Figueroa	\$ 600.000	A1	01/03/2002
Salas	\$ 1.500.000	A1	01/01/2002
Ríos	\$ 2.000.000	A3	01/06/2002
Venegas	\$ 600.000	A1	01/06/2002
Gonzalez	\$ 2.000.000	A3	01/10/2002

Modificación de la Base de Datos (cont.)

Inserción

- Usamos el operador de unión

$$r \leftarrow r \cup E$$

donde r es una relación y E una expresión del álgebra relacional.

- Agregar el departamento de Finanzas con código A5 y fecha de creación 5/8/2004

$$departamento \leftarrow departamento \cup \{(Finanzas, A5, 5/8/2004)\}$$

<i>nombre</i>	<i>cod_dept</i>	<i>fecha_crea</i>
Informática	A1	01/03/2002
Marketing	A2	01/01/2002
Ventas	A3	01/01/2001
Recursos Humanos	A4	01/01/2003
Finanzas	A5	05/08/2004

Actualización

- A veces queremos cambiar el valor de algún atributo de una tupla sin cambiarla entera. Podría hacerse una eliminación e inserción, pero resulta engorrosos muchas veces.
- Usamos un nuevo operador δ (no reasignamos!):

$$\delta A \leftarrow E(r)$$

que representa el hecho de cambiar el atributo A por la expresión matemática E en la relación r .

- Subir el sueldo de todos los empleados en %10 si su sueldo actual es mayor que \$1.000.000 y en %20 si es menor

$$\begin{aligned}\delta_{\text{sueldo}} &\leftarrow \text{sueldo} * 1,1(\sigma_{\text{sueldo} \geq 1000000}(\text{empleado})) \\ \delta_{\text{sueldo}} &\leftarrow \text{sueldo} * 1,2(\sigma_{\text{sueldo} < 1000000}(\text{empleado}))\end{aligned}$$

■ Cuidado con el orden!!!

<u>nombre</u>	<u>sueldo</u>	<u>cod_dept</u>	<u>fecha_ing</u>
Torres	\$ 1.320.000	A1	01/01/2004
Soto	\$ 600.000	A2	01/01/2003
Pérez	\$ 360.000	A2	01/10/2003
Figueroa	\$ 720.000	A1	01/03/2002
Salas	\$ 1.650.000	A1	01/01/2002
Ríos	\$ 2.200.000	A3	01/06/2002
Campos	\$ 960.000	A2	01/11/2003
Venegas	\$ 720.000	A1	01/06/2002
Carcamo	\$ 600.000	A2	01/04/2003
Gonzalez	\$ 2.200.000	A3	01/10/2002

Vistas

- Hasta ahora hemos operado al nivel conceptual, directamente con las tablas de la base de datos. Muchas veces es mejor que ciertos usuarios tenga acceso a ciertos datos o vistas de los datos.
- Por ejemplo en una aplicación para fijar reuniones, los empleados deben poder ver los nombres de los empleados y los departamentos a los que pertenecen pero no sus sueldos.

$\pi_{emp.nombre, emp.cod_dept, dept.nombre}(empleado \bowtie departamento)$

- Para crear una vista usaremos conceptualmente la sentencia create view de la siguiente forma:

create view v as E

dónde v es el nombre de la vista a crear y E es una expresión de consulta.

- Una vista puede usarse como cualquier relación efectiva de la base de datos, pero cuidado una vista NO es una nueva relación, es simplemente una forma abreviada de referirse a una consulta. La vista se *instancia* cada vez que nos referimos a ella.
- Ejemplo:

create view *importantes* as

$\pi_{nombre, cod_dept}(\sigma_{sueldo \geq 1000000}(empleado))$

importantes

<u>nombre</u>	<u>cod_dept</u>
Torres	A1
Salas	A1
Ríos	A3
Gonzalez	A3

Actualización de Vistas y Valores Nulos

- Las vistas son útiles para consultar datos, pero presentan problemas para actualizaciones, inserciones y borrados.

La dificultad radica en que las modificaciones de la base de datos expresadas en términos de vistas deben traducirse en modificaciones de las relaciones reales en el modelo lógico de la base de datos.

- Por ejemplo, si escribimos

$importantes \leftarrow importantes \cup \{(Gomez, A4)\}$

implica una actualización de la relación *empleado* pero sin especificar ni sueldo ni fecha de ingreso. Existen entonces dos posibilidades:

- Rechazar la inserción y lanzar un error.
 - Insertar la tupla (Gomez, null, A4, null) en la relación *empleado*.
- El valor 'null' representa un valor nulo, desconocido o que no existe. Todas las comparaciones que implican 'null' son por definición falsas (esto último tendrá implicancias prácticas importantes).
 - Un problema importante de la anterior actualización es que, a pesar de que la relación *empleado* tiene una nueva tupla

<u>nombre</u>	<u>sueldo</u>	<u>cod_dept</u>	<u>fecha_ing</u>
Torres	\$ 1.200.000	A1	01/01/2004
Soto	\$ 500.000	A2	01/01/2003
Pérez	\$ 300.000	A2	01/10/2003
Figueroa	\$ 600.000	A1	01/03/2002
Salas	\$ 1.500.000	A1	01/01/2002
Ríos	\$ 2.000.000	A3	01/06/2002
Campos	\$ 800.000	A2	01/11/2003
Venegas	\$ 600.000	A1	01/06/2002
Carcamo	\$ 500.000	A2	01/04/2003
Gonzalez	\$ 2.000.000	A3	01/10/2002
Gomez	null	A4	null

La vista *importantes* no cambia

Actualización de Vistas (cont.)

- El siguiente es el resultado de la vista *importantes* antes y después de la inserción si se ingresan valores nulos para los atributos no especificados.

importantes

<u>nombre</u>	<u>cod_dept</u>
Torres	A1
Salas	A1
Ríos	A3
Gonzalez	A3

$$importantes \leftarrow importantes \cup \{(Gomez, A4)\}$$

importantes

<u>nombre</u>	<u>cod_dept</u>
Torres	A1
Salas	A1
Ríos	A3
Gonzalez	A3

- Una posible solución al problema sería haber insertado datos (mínimos) adicionales al ingresar la nueva tupla en la vista de manera de obtener el resultado deseado, o sea, haber insertado la tupla con atributo (Gomez, 1000000, A4, null) pero ¿cómo podemos tomar esta decisión automáticamente? Un tema complejo que vendrá determinado por cada relación concreta.