Resumen para el parcial práctico de DBD

Tema 1: Modelado Conceptual

- Todas las entidades van a tener al menos un identificador. Hay que encontrarlo. Puede ser interno, compuesto o externo.
- Hay que tener cuidado con los atributos que no son (1,1) . Tratar a los teléfonos como (1,N) . Tratar a las direcciones como atributos compuestos: calle, nro, piso (0,1), dpto (0,1).

Tema 2: Modelado Lógico y Físico

Pasaje a modelo Lógico:

- Resolver las Jerarquías: dejamos todas las entidades y las conectamos con relaciones del tipo “es un”
- Resolver los Atributos Compuestos: dejar sólo los atributos individuales.
- Resolver los Atributos Polivalentes: convertimos el atributo polivalente en una entidad y lo conectamos con una relación.

Pasaje a modelo Físico:

- Cada entidad se transforma en una tabla
- Las relaciones N a N se convierten en tablas
- Las relaciones (0,1) a (0,1) se convierten en tablas.
- Las relaciones (0,1) a N se convierten en tablas.
- Tener cuidado con las relaciones N a N recursivas. Se convierten en tabla.

Tema 3: Álgebra Relacional

- Selección: retorna una tabla con las rows que cumplan con una condición sobre T tabla.

Tabla persona

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Juan	Perez
Carlos	Bilardo
Gustavo	Lopez

$\sigma_{\text{nombre='Carlos'}}(\text{persona})$

Nombre	Apellido
Carlos	Griguol
Carlos	Bilardo

- Proyección: retorna una tabla que tiene un subconjunto de atributos de T tabla. Elimina las rows duplicadas.

Tabla persona

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Juan	Perez
Carlos	Bilardo
Gustavo	Lopez

$\pi_{\text{nombre}}(\text{persona})$

Nombre
Pedro
Carlos
Juan
Gustavo

- Unión: retorna una tabla que contiene las rows de T1 más todas las rows de T2. Elimina las rows duplicadas. T1 y T2 deben ser compatibles, es decir, ambas tablas deben tener la misma cantidad, posición y dominio de atributos (los nombres de los atributos si pueden ser distintos).

Tabla futbol

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Gustavo	Lopez

Tabla tenis

Nombre	Apellido
Rafael	Nadal
Roger	Federer
Carlos	Griguol

futbol ∪ tenis

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Gustavo	Lopez
Rafael	Nadal
Roger	Federer

- Intersección: retorna una tabla que contiene las rows que se encuentren en T1 y en T2. Ambas tablas deben ser compatibles.

Tabla futbol

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Gustavo	Lopez

Tabla tenis

Nombre	Apellido
Rafael	Nadal
Gustavo	Lopez

futbol ∩ tenis

Nombre	Apellido
Gustavo	Lopez

- Producto Cartesiano: retorna una tabla concatenando cada row de T1 con todas las rows de T2. Tener en cuenta que esto puede retornar una tabla gigante con rows inútiles.

Tabla futbol

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Facu	Oreja

Tabla tenis

Nombre	Apellido
Rafael	Nadal
Gustavo	Lopez

futbol x tenis

Futbol.nombre	Futbol.apellido	Tenis.nombre	Tenis.apellido
Pedro	Troglio	Rafael	Nadal
Pedro	Troglio	Gustavo	Lopez
Carlos	Griguol	Rafael	Nadal
Carlos	Griguol	Gustavo	Lopez
Facu	Oreja	Rafael	Nadal
Facu	Oreja	Gustavo	Lopez

- Producto Natural: Retorna una tabla concatenando cada row de T1 con todas las rows de T2 donde coincidan los valores de todos los atributos con igual nombre.

Tabla futbol

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Facu	Oreja

Tabla tenis

Nombre	Apellido
Pedro	Troglio
Pedro	Gonzalez

futbol |×| tenis

Futbol.Nombre	Futbol.Apellido
Pedro	Troglio

Tabla tenis

Nombre	Apellido1
Pedro	Gonzalez
Pedro	Troglio

futbol |×| tenis

Futbol.Nombre	Futbol.Apellido	Tenis.apellido1
Pedro	Troglio	Gonzalez
Pedro	Troglio	Troglio

- Diferencia: retorna una tabla que contiene las rows de T1 que no se encuentren en T2. Ambas tablas deben ser compatibles.

Tabla futbol

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Facu	Oreja

Tabla tenis

Nombre	Apellido
Pedro	Gonzalez
Pedro	Troglio

futbol – tenis

Nombre	Apellido
Carlos	Griguol
Facu	Oreja

- División: retorna una tabla con los atributos de T1 que no estén en T2, donde los valores en esos atributos de T1 se correspondan con todas las rows de T2. Es necesario que el esquema de T2 esté incluido en T1. Elimina rows duplicados.

Tabla futbol

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Pedro	Gonzalez

Tabla tenis

Apellido
Gonzalez
Troglio

futbol ÷ tenis

Nombre
Pedro

- Renombre: retorna la misma tabla que entró con un nombre distinto.

$$\rho T\ (Table)$$

- Asignación: guarda en una variable el resultado de una consulta para luego poder utilizar la variable.

$$A \Leftarrow Consulta$$

Para agregar rows a una tabla solamente necesitamos unir la tabla en cuestión con los datos entrantes y asignarlos a la misma tabla.

$$Producto \Leftarrow Producto \cup \{(1235, \text{“tuerca de 9 mm”}, 10, 50, \$10)\}$$

Para eliminar una row de una tabla tenemos que hacer una diferencia con la tabla en cuestión y la selección de la row que queremos eliminar, y luego asignar la consulta a la tabla en cuestión.

$$Producto \Leftarrow Producto - \sigma codProd=893(Producto)$$

Para actualizar un atributo de todas las rows tenemos una operación especial.

$$\delta_{saldo \Leftarrow saldo * 1.05}\ (\text{depósito})$$

Tema 4: SQL

Definición de datos:

- De bases de datos: CREATE | DROP SCHEMA
- De dominios: CREATE | ALTER | DROP DOMAIN
- De tablas: CREATE | ALTER | DROP TABLE
- De vistas: CREATE | DROP VIEW
- De índices (algunos SGBD): CREATE | DROP INDEX

Manipulación de datos:

- SELECT: permite listar contenido de una o varias tablas.
  - Formato Básico:
    Select campo/s
    From tabla/s
  - Operador *: en el select se puede utilizar * y permite mostrar todos los campos de las tablas involucradas sin necesidad de escribir todos los atributos.
  - Función DISTINCT: se aplica a campos del select y elimina rows repetidas.
  - Cláusula WHERE: permite agregar condiciones a una consulta. Los operadores permitidos son: =, >, <, >=, <=, <>, BETWEEN (entre tal y tal. Se

incluyen los extremos), LIKE (permite una gran potencia para manejo de strings), IS NULL, IS NOT NULL.

- Operador LIKE:
    - %: representa cualquier cadena de caracteres. Ej: WHERE (apellido LIKE '%or%').
    - _ (guión bajo): sustituye sólo el carácter del lugar donde aparece. Ej: WHERE (nombre LIKE '_ _ _')
- Los atributos utilizados en el SELECT pueden tener asociados operadores válidos para sus dominios. El: SELECT apellido, dni + 5000.
- Producto Cartesiano: para hacer un producto cartesiano entre dos o más tablas solo basta con poner en la cláusula FROM dos o más tablas separados por coma.
- Definir un alias para una tabla: podemos asignarle otro nombre a una tabla en una consulta. Solo basta con agregar el nombre nuevo seguido de la tabla en el FROM. Ej: FROM alumno a, materia m.
- Renombrar atributos: utilizando el operador AS en el SELECT podemos renombrar atributos. Ej: SELECT a.nombre AS 'Nombre alumno'.
- Producto Natural: el producto natural se realiza en la cláusula FROM indicando las tablas involucradas en dicho producto, y luego de la sentencia ON la condición que debe cumplirse. Ej: FROM alumno a INNER JOIN examen e ON (a.dni = e.dni). Existen 3 formas: INNER JOIN, LEFT JOIN, RIGHT JOIN.
- Unión: Para unir el resultado de dos consultas existen los operadores UNION y UNION ALL, una no retorna tuplas duplicadas y la otra si, respectivamente. Las consultas deben tener esquemas compatibles.
- Diferencia: para hacer la diferencia entre dos consultas existe el operador EXCEPT. Ambas consultas deben tener esquemas compatibles.
- Intersección: para hacer la intersección entre dos consultas existe el operador INTERSECT.
- Ordenar: podemos ordenar las tuplas resultantes de una consulta por el atributo que se quiera utilizando el operador ORDER BY. Por defecto ordena de menor a mayor (ASC). Si se desea ordenar de mayor a menor se utiliza el operador DESC. Dentro de la cláusula se pueden indicar más de un criterio de ordenación. El segundo criterio se aplica en caso de empate en el primero y así sucesivamente. Ej: ORDER BY apellido, dni DESC
- Funciones de agregación: permite realizar una operación sobre un conjunto de tuplas de entrada produciendo un único valor de salida. Se utilizan en el SELECT.
    - AVG: retorna el promedio del atributo indicado para todas las tuplas del conjunto.
    - COUNT: retorna la cantidad de tuplas involucradas en el conjunto de entrada.
    - MAX: retorna el valor más grande dentro del conjunto de tuplas para el atributo indicado.
    - MIN: retorna el valor más pequeño dentro del conjunto de tuplas para el atributo indicado.
    - SUM: retorna la suma del valor del atributo indicado para todas las tuplas del conjunto.

- Agrupar: podemos agrupar las tuplas de una consulta por algún criterio con el objetivo de aplicar alguna función de agregación utilizando la cláusula GROUP BY. Ej:
  SELECT nombre, apellido, AVG(nota) AS promedio
  FROM alumno a INNER JOIN examen e ON (a.dni = e.dni)
  GROUP BY e.dni, nombre, apellido

  Además podemos agregar la cláusula HAVING luego de GROUP BY para restringir los grupos que aparecen en la tabla resultante mediante alguna condición que deben cumplir los grupos. Ej: HAVING AVG(nota) >= 6
- Subconsultas: podemos ubicar una consulta SQL dentro de otra. Existen 5 operaciones de comparación para subconsultas:
  - igualdad =: cuando una subconsulta retorna un único resultado, es posible compararlo contra un valor simple.
  - pertenencia IN: comprueba si un elemento es parte o no de un conjunto. También existe NOT IN
  - =SOME: compara si un elemento es igual a alguno.
  - >ALL: compara si un elemento es mayor que todos.
  - <=SOME: compara si un elemento es menor o igual que alguno.
- Cláusula Exists: permite comprobar si una subconsulta generó o no alguna tupla como respuesta.

- INSERT: permite agregar contenido a una tabla.
  - Estructura básica: INSERT INTO alumno (dni, nombre, apellido) VALUES (2827893, ‘Raul’, ‘Perez’);
- UPDATE: permite modificar el contenido de uno o varios atributos de una tabla.
  - Estructura básica: UPDATE alumno SET nombre=’Lorenzo’ WHERE dni=22232425;
- DELETE: permite borrar una tupla o un conjunto de tuplas de una tabla.
  - Estructura básica: DELETE FROM alumno WHERE dni=22222222;