

TEMA 5. ÁLGEBRA RELACIONAL SQL

A pesar de que SQL utiliza operadores algebraicos matemáticos, éstos operan de forma diferente a como lo hacían en la teoría de conjuntos ya que se aplican sobre tablas relaciones en lugar de sobre conjuntos o relaciones. SQL es el lenguaje estandarizado y de uso universal utilizado por los SGBD basados en el álgebra relacional. Como cualquier lenguaje de un SGBD, SQL se encarga de la descripción, manipulación, control y seguridad de los datos.

Definiremos como compatibles a 2 relaciones R1 y R2 si ambas tienen el mismo grado y el atributo enésimo de R1 está definido en el mismo dominio que el atributo enésimo de la relación R2, si bien el nombre de los atributos puede ser diferente.

La unión de 2 relaciones compatibles R1 y R2 es una nueva relación R3, también compatible, cuyo esquema es igual al esquema de R1 y R2 y cuya extensión está formada por la agrupación sin repetición de las extensiones de R1 y R2.

La diferencia (minus) de 2 relaciones compatibles R1 y R2 es una nueva relación R3, también compatible, cuyo esquema es igual al esquema de R1 y R2 y cuya extensión está formada por aquellas tuplas de la relación R1 que no se encuentran en la relación R2.

La selección (select) sobre una relación R1 mediante una cualificación Q es una nueva relación R2, cuyo esquema es igual a R1 y cuya extensión está formada por todas aquellas tuplas de R1 que satisfacen la cualificación Q.

La proyección (project) sobre una relación R1 con esquema R1.ai, R1.aj, ..., R1.az mediante un subesquema $S(R1) \equiv R1.am, \dots, R1.ap$, donde $am \geq ai$ y $ap \leq az$, es una nueva relación R2 cuyo esquema es igual al subesquema S(R1) y cuya extensión es igual a todas las tuplas de R1 sin repetición sobre el subesquema S(R1).

El producto cartesiano de 2 relaciones R1 y R2 no necesariamente compatibles es una nueva relación R3, cuyo esquema es igual a la concatenación de los esquemas de R1 y R2 y cuya extensión está formado por el conjunto de tuplas que se obtiene de concatenar cada una de las tuplas R1 con todas y cada una de las tuplas de R2.

La intersección (intersect) de 2 relaciones compatibles R1 y R2 es una relación R3, también compatible, cuyo esquema es igual al esquema de R1 y R2 y cuya extensión está formada por el conjunto de tuplas que son comunes a R1 y R2.

La reunión (join) de 2 relaciones R1 y R2, compatibles o no, con al menos un atributo con el dominio común y sobre una cualificación Q da lugar a una nueva relación R3, cuya intención es la concatenación de R1 y R2 y cuya extensión está formada por las tuplas que resultan del producto cartesiano $R1 * R2$ que satisfacen la cualificación Q. Existirá distintos tipos de reunión como:

- Equireunión: Cualificación Q es igual.
- Thetareunión: Cualificación Q no es igual.
- Reunión natural: Se representa con el signo $|x|$, y es una equireunión sobre todos los atributos comunes existentes de las relaciones R1 y R2.

- Autoreunión: Es una reunión natural sobre una misma relación ($R1=R2$).
- Semireunión: Esta operación se realiza entre 2 relaciones $R1$ y $R2$ sobre una cualificación multiatributo Q en la que puede aparecer cualquier operador de comparación y el resultado es una relación $R3$, de esquema igual a $R1$ y cuya extensión está formada por todas aquellas tuplas de $R1$ que intervienen en la reunión de $R1$ y $R2$.

La división de una relación $R1$ con esquema $R1.a_i, R1.a_j, \dots, R1.a_z$ entre una relación $R2$ con esquema $R1.a_m, R1.a_n, \dots, R1.a_p$, donde $a_m \geq a_i$ y $a_p \leq a_z$, es una relación $R3$ cuyo esquema es igual a la diferencia del esquema de $R1$ menos el esquema de $R2$ y cuya extensión es igual a todas las tuplas de $R1$ sin repetición para las cuales está presente toda la extensión de la relación $R2$.

En SQL, se podrá definir tablas del esquema relacional mediante create table, incluyendo los distintos atributos a partir de los cuales se forman. También podremos consultar la información mediante select. Por otro lado, podremos manipular los datos mediante insert, actualizar la tabla mediante update o borrar elementos de las tablas mediante delete.

```
CREATE TABLE NombreTabla (
    nombre-atributo-1: DOMINIO {opción},
    nombre-atributo-2: DOMINIO {opción},
    .....
    nombre-atributo-n: DOMINIO {opción},
    PRIMARY KEY (lista-atributos),
    FOREIGN KEY (lista-atributos),
    REFERENCES NombreTabla{clave-principal}
    [ON DELETE CASCADE] );

SELECT [DISTINCT | ALL] (lista-atributos-1 | *)
[INTO cursor]
FROM lista-tablas
[WHERE condición-1]
[GROUP BY lista-atributos-2]
[HAVING condición-2]]
[ORDER BY lista-atributos-3 [ASC | DESC]];

INSERT [INTO] (lista-atributos)
NombreTabla [lista-tablas]
VALUES (lista-valores)

UPDATE NombreTabla
SET atributo-i = valor
[WHERE Condición];

DELETE [FROM] NombreTabla
[WHERE Condición];
```