**SQL**

**Breve historia de SQL**

**SQL** (*Structured Query Language* - Lenguaje de consulta estructurada) es un lenguaje que se basó en 2 principios fundamentales:

1. Teoría de conjuntos.
2. Álgebra relacional de Edgar Codd (científico informático inglés que evoluciono la teoría de conjuntos).

⠀  
**SQL** fue creada en 1974 por IBM. Originalmente fue llamado **SEQUEL**, pero posteriormente se cambió el nombre por problemas de derechos de autor con otra compañía. Mas adelante **SQL** se convertiría en un lenguaje estándar que unificaría todo dentro de las bases de datos relacionales, se convierte en una norma ANSI o ISO; por lo que, la gran mayoría de sub-lenguajes o manejadores de bases de datos relacionales tratan de la misma manera la sintaxis del lenguaje SQL (Oracle, MySQL, PostgreSQL,…).

**Álgebra relacional**

El **álgebra relacional** estudia básicamente las operaciones que se pueden realizar entre diversos conjuntos de datos.

⠀  
No confundir las **relaciones del álgebra relacional** con las **relaciones de una base de datos relacional**.

* Las relaciones en una base de datos se refieren a cuando, básicamente, unes dos tablas.
* Las relaciones en álgebra relacional se refieren a una tabla; o, para ser más precisos, a la interacción que haya con los conjuntos de datos dentro de una misma tabla.

La diferencia es conceptual: Las tablas pueden tener filas o tuplas repetidas pero en el álgebra relacional cada relación no tiene un cuerpo, no tiene un primer ni último *row* (esto debe abordarse de manera más detallada, posteriormente, para su mejor entendimiento).

**Tipos de operadores en Álgebra relacional:**

* **Operadores unarios.-** Requieren una tabla o relación para funcionar o correr. *Ejemplos*:

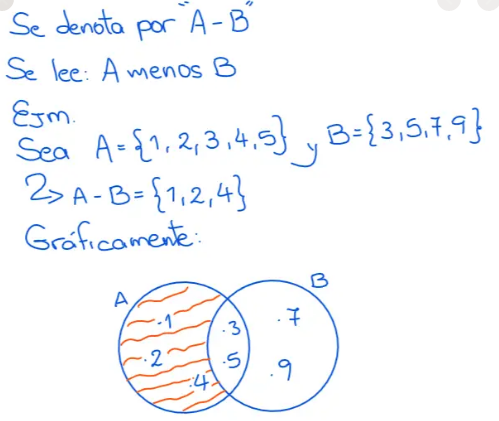
- **Proyección (π):** Haciendo un símil con SQL equivale al comando ***Select***. Saca o extrae toda la información de un número de columnas o atributos. Lo dicho, esto sin necesidad de hacer una unión con una segunda tabla o relación. *Simbología de la proyección:* ***π***.  
  
**π**<Nombre, Apellido, Email>(Tabla\_Alumno).  
⠀  
- **Selección (σ):** Haciendo un símil con SQL equivale al comando ***Where***. Consiste en el filtrado de tuplas o filas que cumplan con cierta condición o valor buscado. *Simbología de la la selección:* **σ.**  
  
**σ**<Suscripción=Expert>(Tabla\_Alumno).

⠀

* **Operadores binarios.-** Requieren más de una tabla para operar (específicamente dos). *Ejemplos*:

- **Producto cartesiano (x):** Toma todos los elementos de una primera tabla, uno por uno, y los combina con todos los elementos de una segunda tabla.   
  
Docentes\_Quinto\_A **x** Alumnos\_Quinto\_A  
⠀  
- **Unión (∪):** Obtiene los elementos que existen, así sea una vez, en cualquiera de las dos tablas; es decir, todo elemento existente, ya sea en tabla A o en tabla B, los une. Une los elementos de tabla A con los de tabla B.  
  
Alumnos\_Quinto\_A **x** Alumnos\_Quinto\_B  
⠀  
- **Diferencia (-):** Obtiene los elementos que existen en una primera tabla seleccionada, pero que no existen en la segunda tabla evaluada en cuestión.  
  
Alumnos\_planExpertPlus **-** Alumnos\_planFree.

***Diferencia al detalle***La diferencia entre dos conjuntos A y B es el conjunto A \ B que contiene todos los elementos de A que no pertenecen a B.



Como ni 7 ni 9 están en el conjunto A, no podemos restar en A esos equivalentes que tenemos en B, pues no los tenemos (7, 9); es decir, se omiten 7 y 9 simplemente en dicha resta. En el conjunto de la “diferencia” se colocan solamente, entonces, los números que no se pudieron restar en A porque no tuvieron su equivalente en B; es decir: 1, 2 y 4.