**Consultas**

Lenguajes de Consulta 🡪 Utilizados para operar con la BD.

* Procedurales: Instrucciones para realizar secuencia de operaciones (qué y cómo)
* No Procedurales: Solicita directamente la información deseada (qué)
* Las consultas representan el 80% de las operaciones registradas sobre una BD, su resolución dependerá del tipo de respuesta que deseo.

**Álgebra Relacional:**

* Lenguaje de consultas procedimental
* Operaciones de una o dos relaciones (tablas) de entrada que generan una nueva relación (tabla) temporal como resultado.
* Operaciones fundamentales:
  + Unitarias (selección, proyección, renombre)
  + Binarias (producto cartesiano, unión, diferencia)

Operaciones:

* Selección 🡪
  + Selecciona tuplas que satisfacen un predicado dado
  + Ejemplo:
  + Devuelve todas las tuplas para las cuales la sentencia anterior sea verdadera.
  + Condición puede ser =, <,>, <=, >=.
  + Operadores lógicos: and(^), or(v)
* Proyección 🡪
  + Nos da la tabla argumento pero con columnas OMITIDAS. Sirve para cuando queremos un dato ignorando los demás. Se puede especificar uno o más atributos, separándolos con “,”.
  + Nos importa el orden en el cuál practicamos las operaciones, primero se empieza por el paréntesis.
  + Ejemplo:
* Producto Cartesiano 🡪
  + Conecta dos entidades de acuerdo a la definición matemática de la operación.
  + Concatena cada tupla de una tabla con cada una de las tuplas de la otra tabla.
* Renombre 🡪
  + Permite utilizar la misma tabla en una operación de consulta.
* Unión 🡪
  + Tuplas comunes a dos relaciones, equivalente a la unión matemática.
  + Debe efectuarse entre relaciones con sentido.
* Diferencia 🡪
  + Diferencia de conjuntos.
  + Debe efectuarse entre relaciones con sentido.

Definición formal de Álgebra Relacional:

* Una expresión básica en AR consta de:
  + Una relación de una Base de Datos.
  + Relación constante.
* Una expresión general se construye a partir de sub-expresiones.

Operaciones Adicionales:

* Intersección 🡪 = A – (A – B)
  + Equivalente a la intersección matemática.
* Producto Natural 🡪
  + Hace el producto cartesiano con una selección de tuplas “con sentido” eliminando las columnas (atributos) repetidas.
  + Las relaciones (tablas) deben tener un y solo un atributo en común (igual nombre e igual dominio).
* Asignación Temporal 🡪
  + Expresión que asigna a una variable temporal el resultado de una operación.

Operaciones de Updates:

* r relación
* E nueva tupla 🡪 (a1, a2, …, an), a1 \*1,5, …
* Agregar tuplas:
* Eliminar tuplas:
* Actualización de datos:

**SQL**

* Un poco de historia:
  + En 1986 se vuelve un estándar ANSI.
  + En 1992 se amplía el estándar agregando nuevas versiones (SQL2 o SQL92) 🡪 vamos a trabajar con éste estándar.
  + En 1999 se crea SQL2000 incorporando expresiones regulares, consultas recursivas y características de Orientación a Objetos (demanda de la época por el auge del paradigma OO)
  + En 2003 surge SQL 3 agregando características de XML
  + En 2006 se definen características que lo acercan al mundo de la internet.
  + La versión ’92 es el núcleo de todas.
  + Se relaciona al modelo relacional.
  + SQL se divide en dos partes, DLL (Data definition language) y DML(Data manipulation language)

***Lenguaje de definición de datos (DDL):***

* Resulta muy amplio, nos ayuda a crear la BD, crear tablas, atributos, integridades referenciales, índices, etc. Todo para ser manipulado después, DDL resulta ser la mitad menos usada de SQL, el enfoque está en DML.
  + Solo veremos las siguientes operaciones comunes:
    - CREATE DATABASE (crear una base de datos en base a un nombre especificado a continuación de la sentencia)
    - DROP DATABASE (eliminar una base de datos en base a un nombre especificado a continuación de la sentencia)
    - CREATE TABLE (crea una tabla con un nombre dado a continuación de la sentencia)
    - ALTER TABLE (altera una tabla con un nombre dado a continuación de la sentencia)
    - DROP TABLE (eliminar una tabla con un nombre dado a continuación de la sentencia)

**ACLARATORIA**: las operaciones de consulta en SQL generan tablas TEMPORALES, como en álgebra.

Estructura:

* SELECT atr1, atr2, atr3, … 🡪 Obligatorio (Se mostrarán en la tabla en el orden que se pongan los atributos)
  + \* 🡪 Incluye todos los atributos de las tablas que aparecen en el From.
  + DISTINCT (lista\_atrib) 🡪 Elimina tuplas duplicadas.
  + ALL 🡪 valor por defecto, aparecen todas las tuplas
  + Se pueden realizar operaciones válidas a los atributos, sin cambiar su valor siempre y cuando el atributo sea un número.
  + CONCAT 🡪(at1, “,”, at2)
  + RTRIM 🡪 Elimina todos los blancos luego del String
* FROM tabla1, tabla2, … 🡪 Obligatorio
  + , 🡪Producto Cartesiano: Concatena cada tupla de una tabla con cada una de las tuplas de la otra tabla (DEBO poner en el WHERE qué atributos deben ser iguales para que no queden tuplas incorrectas).
  + tab1 INNER JOIN tab2 ON (tab1.at1 = tab2.at2) 🡪 Producto Natural: Concatena las tuplas de tab1 con las tuplas de tab2 cuando los atributos especificados son iguales.
  + tabla t 🡪 renombre de tabla
  + atributo AS at 🡪 renombre de atributo
* WHERE condición/es
  + AND / OR
  + Atributo BETWEEN val1 and val2🡪 Muestra los valores del atributo que se encuentran entre el rango de val1 y val2.
  + Like:
    - % 🡪 Reemplaza una cadena de caracteres
    - \_ 🡪 Reemplaza un carácter dentro de una cadena
  + atrib IS NULL / IS NOT NULL 🡪 Retorna TRUE o FALSE
  + atrib IN subconsulta 🡪 Retorna TRUE o FALSE si pertenece a un conjunto
  + atrib op SOME subconsulta 🡪 op = (>, <, =, >=, <=, <>)
  + atrib op ALL subconsulta 🡪 op = (>, <, =, >=, <=, <>)
  + EXIST / NOT EXIST subconsulta 🡪 Devuelve TRUE si la subconsulta NO está vacía.
* ORDER BY at1, at2, …
  + atributo DESC 🡪 Ordena en forma descendiente
  + ASC 🡪 Valor por defecto, ordena en forma ascendente
* Funciones de fechas:
  + YEAR
  + MONTH
  + DAY
  + DATE TIME

Operaciones:

* UNION 🡪 Agrupa las tuplas resultantes de dos subconsultas (elimina duplicados).
* UNION ALL 🡪 Conserva duplicados.
* INTERSECT 🡪 Agrupa las tuplas que están en ambas subconsultas.
* EXCEPT 🡪 Diferencia. Agrupa las tuplas que están en la primer subconsulta y no están en la segunda.

Funciones de Agregación: Se aplican a un conjunto de tuplas y sobre ese conjunto brindan un resultado.

* Promedio (AVG) 🡪 Aplicable a atributos numéricos, retorna el promedio de la cuenta.
* Mínimo (MIN) 🡪 Retorna el elemento más chico dentro de las tuplas para ese atributo.
* Máximo (MAX) 🡪 Retorna el elemento más grande dentro de las tuplas para ese atributo.
* Total (SUM) 🡪 Aplicable a atributos numéricos, realiza la suma matemática.
* Cuenta (COUNT) 🡪 Cuenta las tuplas resultantes.
* Limitaciones:
  + DEBEN aparecer en el SELECT o en el HAVING
  + NO pueden aparecer en el WHERE
  + Devuelve un solo valor, por ende, no puede haber otro atributo en el SELECT si hay una función de agregación.
    - Excepción: El atributo que se agrega DEBE ser el mismo por el cual se agrupa (GROUP BY).

Operación de Agrupamientos:

* GROUP BY atrib 🡪 Permite agrupar un conjunto de tuplas por algún criterio.
  + - DEBO agrupar por Clave Candidata (o Primaria).
  + HAVING func\_agreg op valor🡪 Permite aplicar condiciones a los grupos.
  + 1ro INNER JOIN, 2do GROUP BY, 3ro HAVING

Variantes del Producto Natural:

* Left Outer Join: LEFT JOIN
  + Primero se calcula el inner join
  + Luego cada tupla t perteneciente a la relación de la izquierda que no encontró un par igual en la tupla de la derecha, aparece igualmente en el resultado final con valores nulos en los atributos del correspondiente a la derecha.
* Right Outer Join: RIGHT JOIN
  + Ídem anterior pero aparecen las tuplas t de la relación de la derecha.
* Full Outer Join: FULL JOIN
  + Aparecen las tuplas colgadas de ambos lados.

ABM (Altas/Bajas/Modificaciones) sobre tablas:

* Altas:
  + INSERT INTO table (a1, a2, …, an)
* Bajas:
  + DELETE FROM table

WHERE at = valor

* + DELETE FROM table

Borrado en Cascada

WHERE at = subconsulta

* Modificaciones:
  + UPDATE table

SET at = valor / operación

WHERE condición

\***Nota**: puedo filtrar a nivel tupla (where) o nivel grupo (having)

VISTA:

* Tabla temporal que tiene nombre.
* Almacena temporalmente el resultado de una consulta.
* Sirve si quiero usar una tabla en otra consulta.
* Cada vez que la vista es utilizada, la consulta almacenada es “recalculada”.
* Sintaxis:
* Create View nombre as