

SELECT:

Casos de Uso: Usada cuando es requerido el realizar una consulta en sí.

Restricciones: Usada únicamente sobre atributos o columnas, pueden seleccionarse en singular o todos los atributos de una tabla usando el carácter *.

Ejemplos:

En una base de datos con registros de alumnos y asesores se consultan los nombres tanto de los alumnos como de los asesores:

```
SELECT alumnos.nombre_Alumno, asesores.nombre_Asesor AS "asesores"
```

FROM:

Casos de Uso: Usada para referencia a una tabla.

Restricciones: Usada únicamente sobre tablas.

Ejemplos:

De la base de datos anterior, es necesario incluir de dónde se tomarán los atributos seleccionados para la consulta:

```
FROM alumnos
```

JOIN:

Casos de Uso: Usada para conjuntar registros de atributos de diversas formas sobre tablas distintas.

Restricciones: Puede conjuntar a los registros coincidentes de dos atributos con el mismo nombre y tipo de dato (INNER JOIN).

Puede igualmente conjuntar a los registros coincidentes de dos atributos con el mismo nombre y tipo de dato, y que, además, conjunte los registros de una de los atributos restantes de una de las dos tablas (LEFT o RIGHT JOIN).

También, puede conjuntar a todos los registros de dos atributos con el mismo nombre y tipo de dato, en este caso no importa que sean coincidentes sus valores (FULL JOIN).

En todos los casos anteriores es necesario incluir la instrucción ON y sus cláusulas para que realice su funcionalidad.

Finalmente, puede conjuntar el mismo nombre y tipo de dato, en este caso no importa que sean coincidentes sus valores y sin necesidad de la instrucción ON (NATURAL JOIN).

Ejemplos:

Bajo la idea de la misma base de datos, se realiza un INNER JOIN para conocer a los alumnos que cuentan con un asesor:

INNER JOIN asesores

ON alumnos.id_Asesor=asesores.id_Asesor;

WHERE:

Casos de Uso: Usada para condicionar una consulta de registros individuales.

Restricciones: Se restringe en base a operadores.

Aritméticos: +, -, *, /, %

Lógicos: AND, OR, NOT

Más lógicos: BETWEEN, LIKE, IN, ALL, ANY/SOME, EXISTS

Ejemplos:

En una base de datos de una librería, [2] se busca conocer el precio de cada libro con un 10% de descuento:

SELECT titulo,precio,

*WHERE precio-(precio*0.1)*

FROM libros;

HAVING:

Casos de Uso: Usada para condicionar una consulta de registros en conjunto.

Restricciones: Usa los mismos operadores que WHERE pero analiza más datos.

Ejemplos:

Siguiendo con el ejemplo de la librería, [2] supongamos que se busca saber la cantidad de libros agrupados por editorial, pero considerando sólo algunos grupos, por ejemplo, los que devuelvan un valor mayor a 2, usamos la siguiente instrucción:

SELECT editorial, count() from libros*

GROUP BY editorial

HAVING count()>2;*

BIBLIOGRAFÍA

- [1] postgresql. "PostgreSQL". PostgreSQL. <https://www.postgresql.org> (Recuperado el 25/05/2023).
- [2] "Tutorial de PostgreSQL - Seleccionar grupos (having)". Tutoriales Programacion Ya. <https://www.tutorialesprogramacionya.com/postgresql/temarios/descripcion.php?cod=191&punto=33&inicio=> (Recuperado el 25/05/2023).