1. Sentencias SQL

- **DML (DATA MANIPULATION LANGUAGE)**
  - **Creación de consultas de datos**

SELECT [TOP (<valor>)] [ALL | DISTINCT] <nomb_columna> {, <nomb_columna>} | *

FROM <nombre_tabla> {, <nombre_tabla>}

[WHERE <condición> | partícula_where]

[GROUP BY <nombre_columna> {, <nombre_columna>}]

[HAVING <condicion_simple_o_compuesta>]

[ORDER BY <nombre_columna> [ASC | DESC] {, <nombre_columna> [ASC | DESC]} | <ordinal> [ASC | DESC] {, <ordinal> [ASC | DESC]}]

- **WHERE**: Sirve para restringir las filas de la tabla que se va a visualizar. Utilizaremos condiciones para que la consulta nos devuelva solo aquellas filas que cumplen la condición. (Operadores: <> = …(simple) and not or (compuesta))
  [WHERE <condición>] → simple <nom_col> <operador> <nom_col2>

  [WHERE <condición>] → compuesta <c_simple> <operador> <c_simple2>

  - **Partícula IN**
    Sirve para establecer una relación de permanencia entre un campo y un conjunto de valores.

    WHERE <nombre_columna> [NOT] IN (<valor> {, <valor>})

  - **Partícula BETWEEN**
    Sirve para expresar un rango de valores (incluidos los de la expresión)

    WHERE <nombre_columna> [NOT] BETWEEN <valor1> AND <valor2>

  - **Partícula LIKE**
    Determina si una cadena de caracteres específica coincide con un patrón especificado, que puede contener caracteres normales y caracteres comodín (meta símbolos).

    WHERE <nombre_columna> [NOT] LIKE 'Expresión_con_metasimbolos'

METASÍMBOLOS

%: Cualquier cadena de cero o más caracteres.

Ej: WHERE title LIKE '%computer%' Títulos de libros que contengan la palabra 'computer' en el título.

_: Un único carácter.

Ej.: WHERE name LIKE '_ean' busca todos los nombres de cuatro letras que terminen en 'ean' (Dean, Sean, etc.)

[ ]: Cualquier carácter individual del intervalo ([a-f]) o del conjunto ([abcdef]) que se ha especificado.

Ej.: WHERE name LIKE '[C-P]arsen' busca apellidos de autores que terminen en arsen y empiecen por cualquier carácter individual entre C y P, como Carsen, Larsen, Karsen, etc. En las búsquedas de intervalos, los caracteres incluidos en el intervalo pueden variar, dependiendo de las reglas de ordenación de la intercalación.

[^]: Cualquier carácter individual que no se encuentre en el intervalo ([^a-f]) o el conjunto ([^abcdef]) que se ha especificado.

Ej.: WHERE name LIKE 'ma[^r]%' busca todos los apellidos de autores que empiecen por 'ma' y en los que la siguiente letra no sea 'r'.

- **Partículas ANY y ALL**

  Se usan en la cláusula WHERE, al utilizar operadores relacionales, para que no falle al comparar con varios valores.

  - **ANY**: Se cumple cuando coincide con alguno de los valores que devuelve una subconsulta (Ver apartado SUBCONSULTAS).

  - **ALL**: Se cumple cuando coincide con TODOS los valores que devuelve una subconsulta (Ver apartado SUBCONSULTAS).

- **GROUP BY**
  Sirve para agrupar filas con el valor de una columna en común.

  [GROUP BY <nombre_columna> {, <nombre_columna>}]

- **HAVING**
  Sirve para discriminar las agrupaciones del GROUP BY mediante una condición.

  [HAVING <condicion_simple_o_compuesta>]

- **ORDER BY**
  Sirve para ordenar las columnas.

  [ORDER BY <nombre_columna> [ASC | DESC] {, <nombre_columna> [ASC | DESC]} | <ordinal> [ASC | DESC] {, <ordinal> [ASC | DESC]}]

EJEMPLOS

```
- SELECT DESTINO, MAX(DISTANCIA) DistanciaMAxima
  FROM VUELOS
  WHERE TIPO_AVION IN (SELECT TIPO
                       FROM AVIONES
                       WHERE LONGITUD > (SELECT AVG(LONGITUD)
                                         FROM AVIONES))
  GROUP BY DESTINO
  ORDER BY DESTINO DESC
```

```
- SELECT Origen, sum(distancia) Distancia
  FROM vuelos
  WHERE origen IN (SELECT origen
                   FROM vuelos
                   WHERE destino = 'LONDRES')
  GROUP BY origen
  HAVING avg(distancia)>800
  ORDER BY 2, origen DESC
```

- **OPERACIONES ENTRE CONSULTAS**

UNION (Operador **unión** de Álgebra Relacional)
Sirve para recuperar mediante una sola consulta la información de varios SELECT, que tengan tipos de datos equivalentes y el mismo número de columnas. Su sintaxis es:

<sentencia_SELECT> **UNION** [ALL] <sentencia_SELECT>

{ UNION [ALL] <sentencia_SELECT>}

[ORDER BY <nombre_columna> [ASC | DESC] {, <nombre_columna> [ASC | DESC]} | <ordinal> [ASC | DESC] {, <ordinal> [ASC | DESC]}]

```
EJEMPLOS
  select origen from vuelos
  UNION
  select destino from vuelos
  UNION
  select num_vuelo from vuelos
```

```
select origen,destino from vuelos
UNION ALL
select origen,destino from vuelos
ORDER BY 1
```

EXCEPT (Operador ***diferencia*** de Álgebra Relacional)
Sirve para recuperar mediante una sola consulta la información diferente entre dos sentencias SELECT; deben tener tipos de datos equivalentes y el mismo número de columnas. Su sintaxis es:

<sentencia_SELECT> **EXCEPT** <sentencia_SELECT>

```
EJEMPLO
        select destino from vuelos
        EXCEPT
        select destino from vuelos where destino = 'BARCELONA'
```

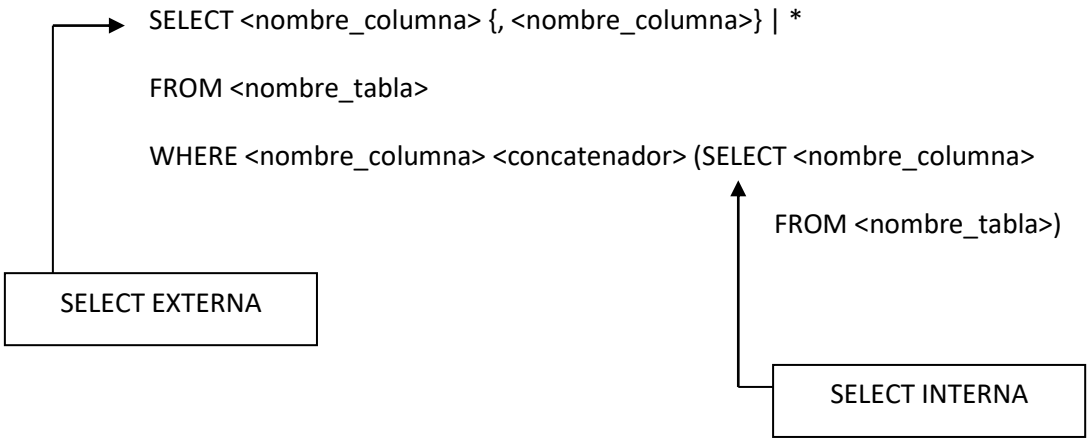
INTERSECT (Operador ***intersección*** de Álgebra Relacional)
Sirve para recuperar mediante una sola consulta la información común entre dos sentencias SELECT; deben tener tipos de datos equivalentes y el mismo número de columnas. Su sintaxis es:

<sentencia_SELECT> **INTERSECT** <sentencia_SELECT>

```
EJEMPLO
        select origen from vuelos
        INTERSECT
        select destino from vuelos where destino = 'MADRID'
```

SUBCONSULTAS

Sirven para recuperar información de una tabla a partir de otra tabla. Es una sentencia SELECT que forma parte del WHERE de otra sentencia SELECT. Lo primero en ejecutarse es la SELECT interna.



** Concatenador: Cualquier operador relacional o partícula valida de WHERE

EJEMPLOS DE SUBCONSULTAS

```
-   select *
    from vuelos
    where distancia >= (select distancia
                        from vuelos
                        where num_vuelo='BA467')
```

```
-   select *
    from vuelos
    where distancia > (select avg(distancia)
                       from vuelos)
```

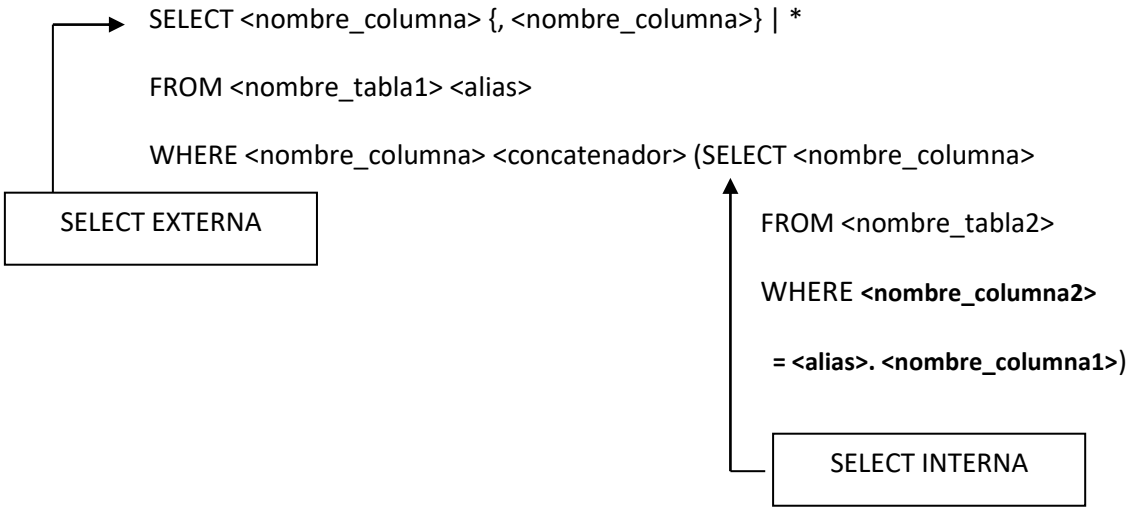
EJEMPLOS DE SUBCONSULTAS CON ANY/ALL

```
-   select *
    from vuelos
    where distancia >= (select distancia
                        from vuelos)
```

- **SUBCONSULTAS CORRELACIONADAS**

Son un tipo especial de subconsultas en la que **la SELECT interna se ejecuta 1 vez por cada fila candidata a ser mostrada en el resultado final de consulta en la SELECT externa**.

A veces son necesarias cuando en una sentencia SQL se necesitan condiciones complejas para seleccionar conjuntos específicos de datos de las tablas.



La consulta interna realiza una función de agregado, tal como una estadística y alimenta esta información a la consulta externa, que la utiliza como la base de una comparación.

Si las tablas son grandes, lleva más tiempo una correlacionada que una normal.

EJEMPLO 1: Vuelos que despegan más pronto para cada origen. (Sin utilizar GROUP BY)

(Con GROUP BY)

```
SELECT origen, min(hora_salida)
FROM vuelos
GROUP BY origen)
```

(Sin GROUP BY)

```
SELECT origen, hora_salida
FROM vuelos v
WHERE hora_salida =
       (SELECT min(hora_salida)
        FROM vuelos
        WHERE origen = v.origen)
```

EJEMPLO 2: Recuperar las reservas cuyo número de plazas libres es mayor que la media para ese mismo vuelo.

```
SELECT *
FROM reservas res
WHERE plazas_libres > (SELECT avg(plazas_libres)
                       FROM reservas
                       WHERE num_vuelo=res.num_vuelo)
```

• SUBCONSULTAS CON EXISTS

Especifica una subconsulta para probar la existencia de filas. Sirve para indicar si existe o no alguna fila que cumpla las condiciones de una subconsulta dentro del WHERE.

WHERE [NOT] EXISTS (<subconsulta>)

EJEMPLO 1

```
SELECT *
FROM vuelos
WHERE EXISTS (SELECT *
              FROM vuelos
              WHERE origen='HUELVA')
```

EJEMPLO 2

```
SELECT DESTINO, MAX(DISTANCIA)
FROM VUELOS V
WHERE EXISTS (SELECT TIPO
              FROM AVIONES
              WHERE LONGITUD > (SELECT AVG(LONGITUD) FROM AVIONES)
                    AND V.TIPO_AVION = TIPO)
GROUP BY DESTINO
ORDER BY DESTINO
```

• RESOLUCIÓN DE CONSULTAS CON "JOIN"

Sirve para crear una macro tabla a través de dos tablas que tengan una o más columnas en común.

SELECT *
FROM <nombre_tabla1> [alias] {, < nombre_tablaX> [alias]}
WHERE <nombre_tabla1>.<nombre_columna1> =
<nombre_tablaX>.<nombre_columnaX>
{ [AND|OR]
<nombre_tabla1>.<nombre_columna1> =
<nomObre_tablaX>.<nombre_columnaX> }

EJEMPLOS

```
SELECT *
FROM VUELOS, AVIONES
WHERE TIPO_AVION = TIPO
```

```
SELECT destino, max(distancia)
FROM vuelos, aviones
WHERE tipo_avion = tipo AND longitud >
       (SELECT avg(longitud)
        FROM aviones)
GROUP BY destino
ORDER BY destino
```

- **ALGORITMOS DE FUNCIONAMIENTO:**
  - **CONSULTA BÁSICAS CON SELECT**
    1. Where → Genera una tabla temporal con las filas que cumplen la condición
    2. Group by → Genera tablas temporales, una por cada subgrupo
    3. F(x) → Se aplican las funciones de columnas a cada subgrupo
    4. Having → Elimina grupos que no queremos visualizar mediante la condición.

  - **SUBCONSULTAS CORRELACIONADAS**
    1. Se selecciona la primera fila de la tabla externa que cumple la condición de la WHERE.
    2. Se ejecuta la SELECT interna.
    3. Si la condición de la WHERE externa se cumple, la fila se visualizará.
    4. Si existen más filas candidatas en la tabla externa se repite desde el primer punto.

- **DCL (DATA CONTROL LANGUAGE)**

  Es un lenguaje proporcionado por el Sistema de Gestión de Base de Datos que incluye una serie de comandos SQL que permiten al administrador controlar el acceso a los datos contenidos en la Base de Datos.

  - **Conceder autorizaciones a los usuarios:**

    GRANT <operaciones> ON [TABLE] <nombre_tabla>
    TO <usuario> [WITH GRANT OPTIONS]

    https://docs.microsoft.com/es-es/sql/t-sql/statements/grant-transact-sql?view=sql-server-ver15

  - **Eliminar autorizaciones concedidas:**

    REVOKE <operaciones> ON [TABLE] <nombre_tabla>
    FROM <usuario> [CASCADE]

    <operaciones>: ALL | ALTER | DELETE | INSERT | SELECT | INDEX | UPDATE
    [<nombre_columna> {, <nombre_columna>}]

    <usuario>: PUBLIC | <nombre_usuario> {, <nombre_usuario>}

    https://docs.microsoft.com/es-es/sql/t-sql/statements/revoke-transact-sql?view=sql-server-ver15

AYUDA SOBRE LAS CONVENCIONES DE SINTAXIS DE TRANSACT-SQL (MS-SQL-SERVER 2017)

https://docs.microsoft.com/es-es/sql/t-sql/language-elements/transact-sql-syntax-conventions-transact-sql?view=sql-server-ver15