10. Lectura SQL y Álgebra Relacional

SQL (Structured Query Language) es el lenguaje estándar utilizado por los Sistemas de Gestión de Bases de Datos (DBMS) para definir, manipular y consultar datos en una base de datos relacional. Las operaciones de consulta en SQL están estrechamente relacionadas con las operaciones del **álgebra relacional**, una teoría matemática que proporciona un conjunto de operaciones para manipular relaciones (tablas) en una base de datos.

Todas las consultas que pueden plantearse con álgebra relacional pueden expresarse mediante sentencias SQL.

Sintaxis Básica de una Consulta SQL:

Esta estructura se compone de tres cláusulas principales:

Sintaxis.sql

```sql
/* SELECT especifica las columnas que se desean recuperar.*/
SELECT columnas
/*FROM indica las tablas de las cuales se obtendrán los datos.*/
/*WHERE filtra las filas según una condición dada*/
FROM tablas WHERE condición;
```

Esquema de Referencia Utilizado

Para ilustrar la equivalencia entre las operaciones del álgebra relacional y las consultas SQL, se utiliza el siguiente esquema de referencia:

- **Materiales(Clave, Descripción, Precio)**
- **Proveedores(RFC, RazónSocial)**
- **Proyectos(Número, Denominación)**
- **Entregan(Clave, RFC, Número, Fecha, Cantidad)**

Notación Utilizada para el Álgebra Relacional

Para facilitar la comprensión, se emplea la siguiente notación en lugar de las letras griegas tradicionales del álgebra relacional:

- **SL{condición}**: Selección con el criterio especificado.
- **PR{lista de columnas}**: Proyección de las columnas indicadas.
- **JN**: Reunión natural (natural join).
- **JN{condición}**: Reunión con el criterio especificado (theta join).
- **UN**: Unión.
- **IN**: Intersección.
- **-**: Diferencia.

- **X**: Producto cartesiano.

Equivalencias entre Operaciones del Álgebra Relacional y Consultas SQL

A continuación, se presentan ejemplos que ilustran la equivalencia entre las operaciones del álgebra relacional y las consultas SQL correspondientes:

4.1. Consulta de una Relación

Álgebra Relacional:
materiales

SQL:

Consulta.sql

```sql
SELECT * FROM materiales;
```

4.2. Selección(σ)
Afecta cardinalidad por que es el numero de duplas

- **Álgebra Relacional:**
- SL{clave=1000}(materiales)

SQL:

Selección.sql

```sql
SELECT * FROM materiales
WHERE clave = 1000;
```

4.3. Proyección (π)

- **Álgebra Relacional:**
  PR{clave, rfc, fecha}(entregan)

Proyección.sql

```sql
SELECT clave, rfc, fecha FROM entregan;
```

Reunión Natural (Natural Join)(><)

Álgebra Relacional:
entregan JN proveedores

Natural_Join.sql

```sql
SELECT * FROM entregan, proveedores
```

```
2   WHERE entregan.rfc = proveedores.rfc;
```

Reunión con Criterio Específico (Theta Join)(><condición)

Álgebra Relacional:
entregan JN{entregan.numero <= proyectos.numero} proyectos

SQL:

Theta_Join.sql

```
1   SELECT * FROM entregan, proyectos
2   WHERE entregan.numero <= proyectos.numero;
```

Unión (∪)

Álgebra Relacional:
SL{clave=1000}(entregan) UN SL{clave=2000}(entregan)

SQL:

Unión.sql

```
1   (SELECT * FROM entregan WHERE clave = 1000)
2   UNION
3   (SELECT * FROM entregan WHERE clave = 2000);
```

4.7. Intersección(∩)
Álgebra Relacional: PR{clave}(SL{numero=5001}(entregan)) IN PR{clave}(SL{numero=5018}(entregan))

SQL:

Intersección.sql

```
1   (SELECT clave FROM entregan WHERE numero = 5001)
2   INTERSECT
3   (SELECT clave FROM entregan WHERE numero = 5018);
```

La operación **INTERSECT** está disponible en sistemas como Oracle, pero no en SQL Server o Access. En estos casos, se pueden utilizar subconsultas para lograr el mismo resultado.

Diferencia (−)
Álgebra Relacional: entregan - SL{clave=1000}(entregan)

SQL:

Diferencia

```
1     (SELECT * FROM entregan)
2     MINUS
```

```
3     (SELECT * FROM entregan WHERE clave = 1000);
```

La operación **MINUS** está disponible en Oracle, pero no en SQL Server o Access. Se pueden utilizar subconsultas para obtener resultados equivalentes en estos sistemas.

Producto Cartesiano en SQL (`×`)

El **producto cartesiano** ocurre cuando se combinan todas las tuplas de dos relaciones, formando una nueva relación con todas las combinaciones posibles.

Álgebra Relacional: entregan X materiales

SQL:

Producto_Cartesiano.sql

```
1   SELECT * FROM entregan, materiales;
2
```

Aunque el **producto cartesiano** es una operación fundamental en el álgebra relacional, en SQL rara vez se usa directamente, ya que generalmente se combina con condiciones de unión (JOIN) para evitar combinaciones no deseadas.

Conclusión

El **álgebra relacional** y **SQL** están estrechamente relacionadas, ya que ambos permiten manipular bases de datos relacionales mediante un conjunto de operaciones bien definidas.

- **SQL** y el **álgebra relacional** están estrechamente relacionadas, ya que SQL se basa en los principios del álgebra relacional para manipular datos.
- Conocer el **álgebra relacional** ayuda a optimizar consultas SQL y comprender su procesamiento interno.
- Algunas operaciones de SQL como `INTERSECT` o `MINUS` pueden no estar disponibles en todos los SGBD, pero pueden implementarse con subconsultas.
- **SQL** es más fácil de usar y está optimizado para ser ejecutado en **Sistemas de Gestión de Bases de Datos (DBMS)**.
- **El álgebra relacional** proporciona la base teórica para SQL, permitiendo comprender cómo se procesan las consultas internamente.
- Muchas operaciones de SQL tienen equivalentes directos en el álgebra relacional, lo que ayuda a los diseñadores de bases de datos a escribir consultas más eficientes.

El estudio de **SQL y Álgebra Relacional** es clave para comprender cómo funcionan las bases de datos y cómo optimizar el rendimiento de las consultas. Aunque SQL es más práctico y fácil de usar, conocer los principios del álgebra relacional permite construir consultas más eficientes y entender cómo se ejecutan a nivel interno.

Dudas