

DICCIONARIO DE DATOS

El estándar SQL que define cómo los sistemas de gestión de bases de datos relacionales (RDBMS) deben manejar el **diccionario de datos** es el **SQL/Information Schema**, que forma parte del estándar SQL más amplio definido por **ISO/IEC 9075**. Este estándar establece cómo los sistemas deben proporcionar una vista estructurada de los metadatos de la base de datos, incluyendo tablas, columnas, restricciones, índices, procedimientos almacenados, entre otros.

SQL/Information Schema – Parte del estándar SQL

El **Information Schema** es una parte integral del estándar SQL, y define un conjunto de vistas que permiten acceder a la **metainformación** (o metadatos) de la base de datos, comúnmente conocido como el **diccionario de datos**.

Estas vistas están diseñadas para ofrecer una interfaz uniforme y estandarizada que se pueda utilizar en cualquier sistema compatible con SQL, independientemente del motor de base de datos que esté siendo utilizado. El estándar garantiza que las aplicaciones puedan consultar el diccionario de datos utilizando un conjunto común de comandos SQL.

Principales Vistas

El estándar SQL/Information Schema define una serie de vistas que proporcionan acceso a diferentes aspectos de los metadatos de la base de datos:

1. **INFORMATION_SCHEMA.TABLES**

Proporciona información sobre las tablas de la base de datos.

2. **INFORMATION_SCHEMA.COLUMNS**

Contiene información sobre las columnas de las tablas, como los nombres de las columnas, tipos de datos, restricciones de longitud, etc.

3. **INFORMATION_SCHEMA.TABLE_CONSTRAINTS**

Muestra las restricciones aplicadas a las tablas, como claves primarias, claves foráneas y restricciones únicas.

4. **INFORMATION_SCHEMA.KEY_COLUMN_USAGE**

Contiene información sobre el uso de claves primarias y foráneas, detallando qué columnas están implicadas en cada una de las restricciones.

5. **INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS**

Proporciona detalles sobre las relaciones referenciales entre tablas, como las claves foráneas y las tablas que se están referenciando.

6. **INFORMATION_SCHEMA.VIEWS**

Ofrece información sobre las vistas definidas en la base de datos.

Requisitos del estándar

El estándar ISO/IEC 9075 garantiza que cualquier motor de base de datos compatible con SQL:

- **Implemente** el esquema `INFORMATION_SCHEMA`, permitiendo acceder a los metadatos de la base de datos de una manera uniforme.
- **Proporcione vistas de solo lectura** (es decir, las vistas del esquema no pueden modificarse directamente) que permitan consultar la estructura de la base de datos.
- **Describa** la estructura interna de la base de datos, como nombres de tablas, columnas, tipos de datos, restricciones, relaciones entre tablas y otros aspectos del diccionario de datos.

Aunque el estándar SQL/Information Schema está ampliamente aceptado y es implementado por la mayoría de los sistemas de gestión de bases de datos, en la práctica cada motor puede tener diferencias en la **implementación de los detalles**:

- Algunos sistemas añaden vistas adicionales específicas de su propio funcionamiento (por ejemplo, PostgreSQL tiene las vistas del catálogo `pg_catalog` que contienen más detalles que no están en `INFORMATION_SCHEMA`).
- Otros motores pueden omitir o no implementar ciertas partes del estándar en su totalidad.

Ejemplo:

El siguiente ejemplo muestra cómo el uso del esquema `INFORMATION_SCHEMA` es parte del estándar SQL y debería funcionar en cualquier sistema que lo respete:

```
-- Consultar las tablas de un esquema en cualquier motor compatible con el estándar SQL
SELECT table_name
FROM INFORMATION_SCHEMA.TABLES
WHERE table_schema = 'public';

-- Obtener columnas de una tabla en particular
SELECT column_name, data_type
FROM INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'mi_tabla';
```

El **estándar SQL/Information Schema** es el conjunto de reglas y especificaciones que define cómo los sistemas de bases de datos relacionales deben manejar y exponer su **diccionario de datos**. Este esquema proporciona una manera estandarizada y consistente de acceder a los metadatos de la base de datos, haciendo que sea más sencillo para los desarrolladores trabajar con diferentes sistemas de bases de datos sin tener que aprender una API específica para cada motor. Sin embargo, es importante tener en cuenta que aunque este estándar es ampliamente implementado, los detalles exactos pueden variar entre diferentes RDBMS.

Diccionario de Datos en PostgreSQL

a. Ver las tablas de un esquema

```
SELECT table_schema, table_name
FROM information_schema.tables
WHERE table_schema = 'public';
```

b. Ver las columnas de una tabla

```
SELECT column_name, data_type, character_maximum_length
FROM information_schema.columns
WHERE table_name = 'nombre_tabla';
```

c. Obtener detalles de las restricciones de una tabla

```
SELECT constraint_name, constraint_type
FROM information_schema.table_constraints
WHERE table_name = 'nombre_tabla';
```

d. Ver las claves foráneas

```
SELECT tc.constraint_name, kcu.column_name,
       ccu.table_name AS foreign_table_name, ccu.column_name AS
foreign_column_name
FROM information_schema.table_constraints AS tc
JOIN information_schema.key_column_usage AS kcu
ON tc.constraint_name = kcu.constraint_name
JOIN information_schema.constraint_column_usage AS ccu
ON ccu.constraint_name = tc.constraint_name
WHERE tc.constraint_type = 'FOREIGN KEY' AND tc.table_name = 'nombre_tabla';
```

Diccionario de Datos en SQL Server

a. Ver las tablas de una base de datos

```
SELECT TABLE_SCHEMA, TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE';
```

b. Ver las columnas de una tabla

```
SELECT COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'nombre_tabla';
```

c. Ver restricciones de una tabla

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE
FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
WHERE TABLE_NAME = 'nombre_tabla';
```

d. Ver claves ajenas

```
SELECT FK.TABLE_NAME AS TablaHija, CU.COLUMN_NAME AS ColumnaHija,
       PK.TABLE_NAME AS TablaPadre, PT.COLUMN_NAME AS ColumnaPadre
FROM INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS C
JOIN INFORMATION_SCHEMA.TABLE_CONSTRAINTS FK
  ON C.CONSTRAINT_NAME = FK.CONSTRAINT_NAME
JOIN INFORMATION_SCHEMA.TABLE_CONSTRAINTS PK
  ON C.UNIQUE_CONSTRAINT_NAME = PK.CONSTRAINT_NAME
JOIN INFORMATION_SCHEMA.KEY_COLUMN_USAGE CU
  ON C.CONSTRAINT_NAME = CU.CONSTRAINT_NAME
JOIN (
  SELECT i1.TABLE_NAME, i2.COLUMN_NAME
  FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS i1
  JOIN INFORMATION_SCHEMA.KEY_COLUMN_USAGE i2
    ON i1.CONSTRAINT_NAME = i2.CONSTRAINT_NAME
  WHERE i1.CONSTRAINT_TYPE = 'PRIMARY KEY'
) PT
  ON PT.TABLE_NAME = PK.TABLE_NAME;
```