SQL CMDs: Constraint & Alter - how to add primary key on table filled with data & get_ddl cmd

SQL Constraints

- SQL constraints are used to specify rules for the data in a table.
- Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.
- Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

- `NOT NULL` - Ensures that a column cannot have a NULL value

  ```
  CREATE TABLE Persons (
      ID int NOT NULL,
      LastName varchar(255) NOT NULL,
      FirstName varchar(255) NOT NULL,
      Age int
  );
  ```

- `UNIQUE` - Ensures that all values in a column are different

  ```
  CREATE TABLE Persons (
      ID int NOT NULL UNIQUE,
      LastName varchar(255) NOT NULL,
      FirstName varchar(255),
      Age int
  );
  ```

- `PRIMARY KEY` - A combination of a `NOT NULL` and `UNIQUE`. Uniquely identifies each row in a table
    - A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

```
-- MySQL
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    PRIMARY KEY (ID)
);
```

```
-- In the example above there is only ONE PRIMARY KEY (PK_Person).
-- However, the VALUE of the primary key is made up of TWO COLUMNS
-- (ID + LastName).
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    CONSTRAINT PK_Person PRIMARY KEY (ID,LastName)
);
```

- `FOREIGN KEY` - Prevents actions that would destroy links between tables
    - A `FOREIGN KEY` is a field (or collection of fields) in one table, that refers to the `PRIMARY KEY` in another table.

      The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

```
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)
);
```

SUPPORTED CONSTRAINT TYPES IN SNOWFLAKE.

Snowflake supports the following constraint types from the ANSI SQL standard:

- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- NOT NULL

`Note: Snowflake supports defining and maintaining constraints, but does not enforce them, except for NOT NULL constraints, which are always enforced.`

For Snowflake Time Travel, when previous versions of a table are copied, the current version of the constraints on the table are used because Snowflake does not store previous versions of constraints in table metadata.

ALTER TABLE

The `ALTER TABLE` statement is used to add, delete, or modify columns in an existing table.

The `ALTER TABLE` statement is also used to add and drop various constraints on an existing table.

```
-- ALTER TABLE - ADD Column
ALTER TABLE table_name
ADD COLUMN column_name <datatype> <any_constraint>;

-- ALTER TABLE - DROP COLUMN
ALTER TABLE table_name
DROP COLUMN column_name;

-- ALTER TABLE - RENAME COLUMN
ALTER TABLE table_name
RENAME COLUMN old_name to new_name;

-- ALTER TABLE - ALTER/MODIFY DATATYPE
```

```
ALTER TABLE table_name
MODIFY COLUMN column_name datatype;
```

- If you have a table filed with data and you want to add an extra column of name age in table with no null values.

  ```
  ALTER TABLE table_name
  ADD COLUMN age int not null;

  -- OUTPUT
  -- SQL compilation error: Non-nullable column 'age' cannot
  -- be added to non-empty table 'table_name' unless it has
  -- a non-null default value.
  ```

  - Basically I can’t add column with null values in a table where every column is filled with data.
  - if table is empty then this alter table can execute properly with not null constraint

  ```
  ALTER TABLE table_name
  ADD COLUMN column_name <datatype>
  ```

ALTER A PRIMARY KEY & DROP A PRIMARY KEY.

```
-- If created a table and it have data and want to set primary key
ALTER TABLE table_name
ADD PRIMARY KEY (column1, column2 ...);

-- Create a column id and set it to primary it will only work if table contain no data.
ALTER TABLE Persons
ADD COLUMN id int PRIMARY KEY;

-- If there is a primary key in table -- how to drop
ALTER TABLE Persons
DROP PRIMARY KEY;
```

- IF u have table with already filled with data and you want to add a new column in it and want to be unique

```
-- STEP 0 -- CREATE A REPLICA OF TABLE YOU WANT TO INSERT PRIMARY KEY
CREATE OR REPLACE TABLE replica_table_name LIKE table_name;

-- STEP 1 -- DROP THE PRIMARY KEY FROM THE WHOLE TABLE
ALTER TABLE replica_table_name
DROP PRIMARY KEY;

-- STEP 2 -- ADD A COLUMN MAKE IT PRIMARY KEY WORKS ONLY WHEN TABLE
          -- IS EMPTY
ALTER TABLE replica_table_name
ADD COLUMN column_name datatype;

-- STEP 3 -- NOW MAKE THAT COLUMN A PRIMARY KEY
ALTER TABLE replica_table_name
ADD PRIMARY KEY (column_name);
```

DEFAULT constraint

The `DEFAULT` constraint is used to set a default value for a column.

The default value will be added to all new records, if no other value is specified.

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    City varchar(255) DEFAULT 'Sandnes'
);
```

- To create a `DEFAULT` constraint on the "City" column when the table is already created

```
ALTER TABLE Persons
ALTER COLUMN City SET DEFAULT 'Sandnes';
```

Get_DDL - How data is defined.

Returns a DDL statement that can be used to recreate the specified object. For databases and schemas, GET_DDL is recursive

```
SELECT get_ddl('table', 'table_name');
```