AUTO_INCREMENT like cmd in SNOWFLAKE - INDENTITY

BACKGROUD ALGORITHM ON WHICH IDENTITY AND AUTOINCREMENT WORK.

IT USES `SEQUENCES` UNDER THE HOOD.

In Snowflake, you can set the default value for a column, which is typically used to set an autoincrement or identity as the default value, so that each time a new row is inserted a unique id for that row is generated and stored and can be used as a primary key.

you can specify the default value for a column using create table or alter table

However, if you try to alter a table to add an autoincrement column that already has data in it, we will get an error in snowflake.

It’s not as easy as altering the existing table, but there are two ways we can add an identity or autoincrement column to an existing table.

IDENTITY FUNCTION

The SNOWFLAKE uses the `IDENTITY` keyword to perform an auto-increment feature.

In the example above, the starting value for `IDENTITY` is 1, and it will increment by 1 for each new record.

Tip: To specify that the "ID" column should start at value 10 and increment by 5, change it to `IDENTITY(10,5)`.

```
CREATE TABLE Persons (
    Personid int IDENTITY(1,1) PRIMARY KEY,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int
);
```

METHOD 1: Using auto_increment or identity as defaut value.

- First we are going to create a simple table that we want to add an identity/autoincrement field to:

```
CREATE OR REPLACE TABLE AD_COLORS AS
  SELECT NAME
  FROM (VALUES ('BLUE'), ('RED'), ('GREEN')) COLORS(NAME);
```

- Next we create a new table with the same structure as existing table and add an identity column.

```
CREATE OR REPLACE TABLE AD_COLORS_COPY LIKE AD_COLORS;
-- THIS WILL ONLY COPY THE STRUCTURE OF THE TABLE
```

- Now we alter table, make sure your table is empty when working with table.

```
ALTER TABLE AD_COLORS_COPY
ADD COLUMN ID INT IDENTITY(1,1);
-- DEFAULT VALUE IS 1 AND IT WILL INCREMENT BY 1

INSERT INTO AD_COLORS_COPY (NAME)
```

```
  SELECT NAME FROM AD_COLORS;

-- THIS WILL PROVIDE UNIQUE ID TO EVERY COLOR NAME
1 - BLUE
2 - RED
3 - GREEN
```

METHOD 2: Generating Sequences

```
CREATE OR REPLACE SEQUENCE AD_SEQ
start = 1
INCREMENT = 5
comment = 'THIS SEQ WILL BE USED TO GENERATE EMPLOYEE IDs';
```

- Let use an existing data table to why we can’t just alter the existing table:

```
-- CREATING THE COPY OF A MAIN TABLE
CREATE OR REPLACE TABLE COPY_TABLE LIKE MAIN_TABLE_NAME;
-- THIS WILL HAVE SAME STRUCTURE AS MAIN TABLE

-- NOW TO GET ALL DATA FROM MAIN TABLE TO COPY TABLE
INSERT INTO COPY_TABLE
  SELECT * FROM MAIN_TABLE_NAME;
-- THIS WILL INSERT ALL DATA OF MAIN TABLE INTO COPY TABLE.
```

- Let use identity column like first method, let see what will happen

```
ALTER TABLE COPY_TABLE
ADD COLUMN ID INT IDENTITY(1,1);

-- WHEN WE EXECUTE THIS, IT WILL THROW ERROR BECAUSE WE CAN'T USE
-- IDENTITY ON TABLE WHICH ALREADY HAVE THE DATA.
```

- SO WHAT NOW, HOW DO WE ADD ID COLUMN IN COPY_TABLE

- WE CAN DO IT BY USING `SEQUENCE` , USE THESE STEPS

STEP 1 : CREATE A COPY OF A MAIN TABLE

```
CREATE OR REPLACE TABLE COPY_TABLE LIKE MAIN_TABLE_NAME;
```

STEP 2 : GENERATING SEQUENCES

```
CREATE OR REPLACE SEQUENCE AD_SEQ -- AD_SEQ IS THE NAME OF SEQUENCE
start = 1
INCREMENT = 1
comment = 'THIS SEQ WILL BE USED TO GENERATE EMPLOYEE IDs';
```

STEP 3 : ADD ID COLUMN IN TABLE

```
ALTER TABLE COPY_TABLE
ADD COLUMN ID INT IDENTITY(1,1);
```

STEP 4 : FILL THE COLUMNS

```
INSERT INTO COPY_TABLE
SELECT *, ROW_NUMBER() OVER (ORDER BY NULL)
FROM MAIN_TABLE_NAME
```

NOW INSERT VALUE

```
INSERT INTO COPY_TABLE(<give name of all column except ID>)
values(<give value for all the column except id>);

-- LET SAY THERE ARE 150 RECORD IN TABLE SO WHEN WE EXECUTE THE ABOVE
-- CODE WHAT ID WE WILL GET
-- 151 NO
-- WE WILL GET 1.
-- WHICH IS WRONG, SO TO CORRECT THIS FOLLOW THESE STEP.
```

- IN THIS CASE, WE HAVE TO USE A SEQUENCE.
- WHEN WE CREATE OUR OWN SEQUENCE WE HAVE ACCESS TO THE NEXT CALUE IN THE SEQUENCE
- THIS ALLOWS US TO ADD THE NEXT INCREMENTAL VALUE WHEN WE BACKFILL THE NEW TABLE WITH THE OLD TABLE.

FIRST WE CREATE A SEQUENCE THAT STARTS AT 1 AND INCREMENTS BY 1 AND NAME IT SEQ1:

```
STEP - 1
CREATE OR REPLACE SEQUENCE SEQ1 START=1 INCREMENT=1;
```

```
STEP - 2
CREATE OR REPLACE TABLE COPY_TABLE LIKE MAIN_TABLE;
```

```
STEP - 3
ALTER TABLE COPY_TABLE
ADD COLUMN ID INT DEFAULT SEQ1.NEXTVAL;

-- YOU WANT TO START FROM 100 NOT FROM 1
CREATE OR REPLACE SEQUENCE SEQ2 START=100 INCREMENT=1;

ALTER TABLE COPY_TABLE
MODIFY COLUMN ID DEFAULT SEQ2.NEXTVAL;
```

- The NEXTVAL function is used in Oracle SQL to retrieve the next value in a sequence.

```
STEP - 4
INSERT INTO COPY_TABLE
SELECT *, SEQ2.NEXTVAL
FROM MAIN_TABLE_NAME
```

```
INSERT INTO COPY_TABLE(<give name of all column except ID>)
values(<give value for all the column except id>);

-- NOW THIS WILL GIVE YOU WRITE ID WHICH IS UNIQUE. 151
```