**Snowflake**

1.Create tables in snowflake :

With the database selected, you can now create a table using the `CREATE TABLE` statement.

**Syntax : 1**

CREATE OR REPLACE TABLE my\_table (  
id INTEGER AUTOINCREMENT,  
name VARCHAR(100) NOT NULL,  
preferences VARIANT,  
created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP()  
);

**Syntax : 2**

The basic syntax for creating a table using Snowflake CREATE TABLE command is:

CREATE TABLE table\_name (

column1 datatype,

column2 datatype,

...

...

);

**Example :**

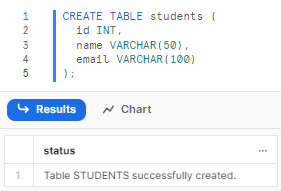
CREATE TABLE students (

id INT,

name VARCHAR(50),

email VARCHAR(100)

);



**2.select queries**

**Selecting all columns in the table**

**Syntax:**

**SELECT description, retail\_price, wholesale\_cost FROM ftable1;**

**+-------------+--------------+----------------+**

**| DESCRIPTION | RETAIL\_PRICE | WHOLESALE\_COST |**

**|-------------+--------------+----------------|**

**| bling | 14 | 6 |**

**+-------------+--------------+----------------+**

1.This example shows how to select all columns in employee\_table:

**SELECT** **\*** **FROM** employee\_table**;**

+-------------+------------+------------+---------------+

| EMPLOYEE\_ID | LAST\_NAME | FIRST\_NAME | DEPARTMENT\_ID |

|-------------+------------+------------+---------------|

| 101 | Montgomery | Pat | 1 |

| 102 | Levine | Terry | 2 |

| 103 | Comstock | Dana | 2 |

+-------------+------------+------------+---------------+

2.Selecting all columns with names that match a pattern

This example shows how to select all columns in employee\_table with names that contain id:

Syntax:

SELECT \* ILIKE '%id%' FROM employee\_table;

Output:

+-------------+---------------+

| EMPLOYEE\_ID | DEPARTMENT\_ID |

|-------------+---------------|

| 101 | 1 |

| 102 | 2 |

| 103 | 2 |

+-------------+---------------+

**3.Selecting all columns except one column**

This example shows how to select all columns in employee\_table except for the department\_id column:

**SELECT** **\*** **EXCLUDE** department\_id **FROM** employee\_table**;**

+-------------+------------+------------+

| EMPLOYEE\_ID | LAST\_NAME | FIRST\_NAME |

|-------------+------------+------------|

| 101 | Montgomery | Pat |

| 102 | Levine | Terry |

| 103 | Comstock | Dana |

|-------------+------------+------------|

4.**Selecting all columns and renaming one column**

This example shows how to select all columns in employee\_table and rename the department\_id column:

**SELECT** **\*** **RENAME** department\_id **AS** department **FROM** employee\_table**;**

+-------------+------------+------------+------------+

| EMPLOYEE\_ID | LAST\_NAME | FIRST\_NAME | DEPARTMENT |

|-------------+------------+------------+------------|

| 101 | Montgomery | Pat | 1 |

| 102 | Levine | Terry | 2 |

| 103 | Comstock | Dana | 2 |

+-------------+------------+------------+------------+

**3.how to insert values one table to another table**

**Example:**

**Syntax:**

 INSERT INTO table2 (id, name, age) SELECT id, name, age FROM table1; This query will copy all rows from table1 into table2

Example:

**CREATE** **OR** **REPLACE** **TABLE** some\_data **(**id **INTEGER,** **name** **VARCHAR);**

**INSERT** **INTO** some\_data **(**id**,** **name)** **VALUES**

**(**1**,** 'a'**),**

**(**2**,** 'b'**);**

**4.replace query**

**Syntax :**

**CREATE** **OR** **REPLACE** **TABLE** replace\_example**(**

subject **VARCHAR(**10**),**

**pattern** **VARCHAR(**10**),**

replacement **VARCHAR(**10**));**

**INSERT** **INTO** replace\_example **VALUES**

**(**'old car'**,** 'old car'**,** 'new car'**),**

**(**'sad face'**,** 'sad'**,** 'happy'**),**

**(**'snowman'**,** 'snow'**,** 'fire'**);**

**Example :**

**SELECT** subject**,**

**pattern,**

replacement**,**

**REPLACE(**subject**,** **pattern,** replacement**)** **AS** new

**FROM** replace\_example

**ORDER** **BY** subject**;**

**Output :**

+----------+---------+-------------+------------+

| SUBJECT | PATTERN | REPLACEMENT | NEW |

|----------+---------+-------------+------------|

| old car | old car | new car | new car |

| sad face | sad | happy | happy face |

| snowman | snow | fire | fireman |

+----------+---------+-------------+------------+

**5.update query**

**Syntax :**

UPDATE <target\_table>

SET <col\_name> = <value> [ , <col\_name> = <value> , ... ]

[ FROM <additional\_tables> ]

[ WHERE <condition> ]

**Example :**

**select** **\*** **from** target**;**

**+**---+----+

| K | V |

|---+----|

| 0 | 10 |

**+**---+----+

**Select** **\*** **from** src**;**

**+**---+----+

| K | V |

|---+----|

| 0 | 11 |

| 0 | 12 |

| 0 | 13 |

**+**---+----+

-- Following statement joins all three rows in src against the single row in target

**UPDATE** target

**SET** v **=** src**.**v

**FROM** src

**WHERE** target**.**k **=** src**.**k**;**

**+**------------------------+-------------------------------------+

| number of rows updated | number of multi-joined rows updated |

|------------------------+-------------------------------------|

| 1 | 1 |

**+**------------------------+-------------------------------------+