**1.Merge into query**

**Syntax :**

MERGE INTO <target\_table> USING <source> ON <join\_expr> { matchedClause | notMatchedClause } [ ... ]

**matchedClause ::=**

**WHEN** **MATCHED** [ **AND** <case\_predicate> ] **THEN** { **UPDATE** **SET** <col\_name> = <expr> [ , <col\_name2> = <expr2> ... ] | **DELETE** } [ ... ]

**notMatchedClause** ::=

**WHEN** **NOT** **MATCHED** [ **AND** <case\_predicate> ] **THEN** **INSERT** [ ( <col\_name> [ , ... ] ) ] **VALUES** ( <expr> [ , ... ] )

**Examples :**

TRUNCATE TABLE source\_table;

TRUNCATE TABLE target\_table;

INSERT INTO source\_table (ID, description) VALUES

(50, 'This is a duplicate in the source and has no match in target'),

(50, 'This is a duplicate in the source and has no match in target')

;

**Execute the MERGE statement:**

**MERGE** **INTO** target\_table **USING** source\_table

**ON** target\_table**.**id **=** source\_table**.**id

**WHEN** **MATCHED** **THEN**

**UPDATE** **SET** target\_table**.**description **=** source\_table**.**description

**WHEN** **NOT** **MATCHED** **THEN**

**INSERT** **(**ID**,** description**)** **VALUES** **(**source\_table**.**id**,** source\_table**.**description**);**

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

| number of rows inserted | number of rows updated |

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

| 2 | 0 |

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

**Display the new value in the target table:**

**SELECT** ID **FROM** target\_table**;**

**+**----+

| ID |

|----|

| 50 |

| 50 |

**+**----+

**2.with query**

**Syntax :**

[ WITH

<cte\_name1> [ ( <cte\_column\_list> ) ] AS ( SELECT ... )

[ , <cte\_name2> [ ( <cte\_column\_list> ) ] AS ( SELECT ... ) ]

[ , <cte\_nameN> [ ( <cte\_column\_list> ) ] AS ( SELECT ... ) ]

]

SELECT ...

**Examples 1 :**

with

albums\_1976 as (select \* from music\_albums where album\_year = 1976)

select album\_name from albums\_1976 order by album\_name;

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

| ALBUM\_NAME |

|----------------------|

| Amigos |

| Look Into The Future |

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

**Examples 1 :**

**with**

album\_info\_1976 **as** **(select** m**.**album\_ID**,** m**.**album\_name**,** b**.**band\_name

**from** music\_albums **as** m **inner** **join** music\_bands **as** b

**where** m**.**band\_id **=** b**.**band\_id **and** album\_year **=** 1976**),**

Journey\_album\_info\_1976 **as** **(select** **\***

**from** album\_info\_1976

**where** band\_name **=** 'Journey'**)**

**select** album\_name**,** band\_name

**from** Journey\_album\_info\_1976**;**

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

| ALBUM\_NAME | BAND\_NAME |

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

| Look Into The Future | Journey |

**3.how to join two tables in snowflake**

A join combines rows from two tables to create a new combined row that can be used in the query.

**Types of Joins :**

1.Inner join

2.Outer join

3.Cross join

4.Natural join

5.[ASOF JOIN](https://docs.snowflake.com/en/sql-reference/constructs/asof-join)

**1.Inner join :**

An inner join pairs each row in one table with the matching row(s) in the other table.

**Syntax :**

SELECT p.project\_ID, project\_name, employee\_ID, employee\_name, e.project\_ID

FROM projects AS p INNER JOIN employees AS e

ON e.project\_id = p.project\_id

ORDER BY p.project\_ID, e.employee\_ID;

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

| PROJECT\_ID | PROJECT\_NAME | EMPLOYEE\_ID | EMPLOYEE\_NAME | PROJECT\_ID |

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

| 1000 | COVID-19 Vaccine | 10000001 | Terry Smith | 1000 |

| 1000 | COVID-19 Vaccine | 10000002 | Maria Inverness | 1000 |

| 1001 | Malaria Vaccine | 10000003 | Pat Wang | 1001 |

**2.Outer join :**

**Syntax :**

**SELECT** p**.**project\_name**,** e**.**employee\_name

**FROM** projects **AS** p **LEFT** **OUTER** **JOIN** employees **AS** e

**ON** e**.**project\_ID **=** p**.**project\_ID

**ORDER** **BY** p**.**project\_name**,** e**.**employee\_name**;**

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

| PROJECT\_NAME | EMPLOYEE\_NAME |

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

| COVID-19 Vaccine | Maria Inverness |

| COVID-19 Vaccine | Terry Smith |

| Malaria Vaccine | Pat Wang |

| NewProject | NULL |

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

**3.Cross join :**

**Syntax :**

**SELECT** p**.**project\_name**,** e**.**employee\_name

**FROM** projects **AS** p **CROSS** **JOIN** employees **AS** e

**ORDER** **BY** p**.**project\_ID**,** e**.**employee\_ID**;**

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

| PROJECT\_NAME | EMPLOYEE\_NAME |

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

| COVID-19 Vaccine | Terry Smith |

| COVID-19 Vaccine | Maria Inverness |

| COVID-19 Vaccine | Pat Wang |

| COVID-19 Vaccine | NewEmployee |

| Malaria Vaccine | Terry Smith |

| Malaria Vaccine | Maria Inverness |

| Malaria Vaccine | Pat Wang |

| Malaria Vaccine | NewEmployee |

| NewProject | Terry Smith |

| NewProject | Maria Inverness |

| NewProject | Pat Wang |

| NewProject | NewEmployee |

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

**4.Natural join :**

**Syntax :**

**SELECT** **\***

**FROM** projects **NATURAL** **JOIN** employees

**ORDER** **BY** employee\_ID**;**

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

| PROJECT\_ID | PROJECT\_NAME | EMPLOYEE\_ID | EMPLOYEE\_NAME |

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

| 1000 | COVID-19 Vaccine | 10000001 | Terry Smith |

| 1000 | COVID-19 Vaccine | 10000002 | Maria Inverness |

| 1001 | Malaria Vaccine | 10000003 | Pat Wang |

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

**5.**[**ASOF JOIN**](https://docs.snowflake.com/en/sql-reference/constructs/asof-join) **:**

**Syntax :**

FROM <left\_table> ASOF JOIN <right\_table>

MATCH\_CONDITION ( <left\_table.timecol> <comparison\_operator> <right\_table.timecol> )

[ ON <table.col> = <table.col> [ AND ... ] | USING ( <column\_list> ) ]

**Examples 1 :**

**SELECT** **\***

**FROM** left\_table l **ASOF** **JOIN** right\_table r

**MATCH\_CONDITION(**l**.**c3**>=**r**.**c3**)**

**ORDER** **BY** l**.**c1**,** l**.**c2**;**

**Output :**

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

| C1 | C2 | C3 | C4 | C1 | C2 | C3 | C4 |

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

| A | 1 | 09:15:00 | 3.21 | A | 1 | 09:14:00 | 3.19 |

| A | 2 | 09:16:00 | 3.22 | B | 1 | 09:16:00 | 3.04 |

| B | 1 | 09:17:00 | 3.23 | B | 1 | 09:16:00 | 3.04 |

| B | 2 | 09:18:00 | 4.23 | B | 1 | 09:16:00 | 3.04 |

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

**4.rename table names**

**Syntax :**

--the syntax

alter table old\_table\_name rename to new\_table\_name;

--rename users\_marketing to users table

alter table sessions\_db1 rename to sessions\_db\_1;