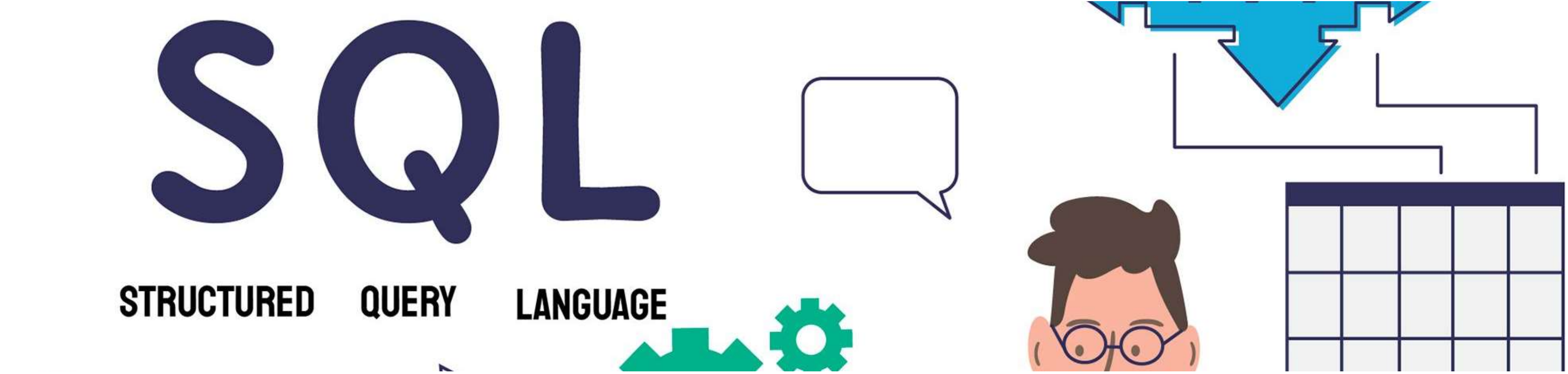


Data transformation In SQL

> *The data transformation process involves updating the year in the TRANSACTIONS table.*

The SQL commands provided in the document demonstrate how to convert specific years to new values. For example, the command `UPDATE TRANSACTIONS SET DATE = DATEADD(YEAR,1,DATE) WHERE YEAR(DATE) = 2021;` *converts all transactions from the year 2021 to 2022.*

Additionally, the document addresses the issue of null values in the BANK column. The SQL command `UPDATE TRANSACTIONS SET Bank = CASE WHEN YEAR(DATE) = 2018 THEN 'Southern Bank' WHEN YEAR(DATE) = 2019 THEN 'Northern Bank' WHEN YEAR(DATE) = 2020 THEN 'Central Bank' WHEN YEAR(DATE) = 2021 THEN 'DBS Bank' WHEN YEAR(DATE) = 2022 THEN 'Sky Bank' ELSE Bank END WHERE Bank IS NULL;` *updates the null values with corresponding bank names based on the transaction year.*

Furthermore, the document includes instructions for adding a new column, "Age," to the CLIENT table. The command `ALTER TABLE CLIENT ADD COLUMN Age INT;` *adds the new column, and the subsequent command* `UPDATE CLIENT SET AGE = DATEDIFF('YEAR',BIRTH_DATE,'2022-12-19');` *populates the column with the age of each client based on their birth date.*

Overall, the document provides a comprehensive guide for data transformation in the banking system, covering various aspects such as updating years, handling null values, adding columns, and updating data in tables.

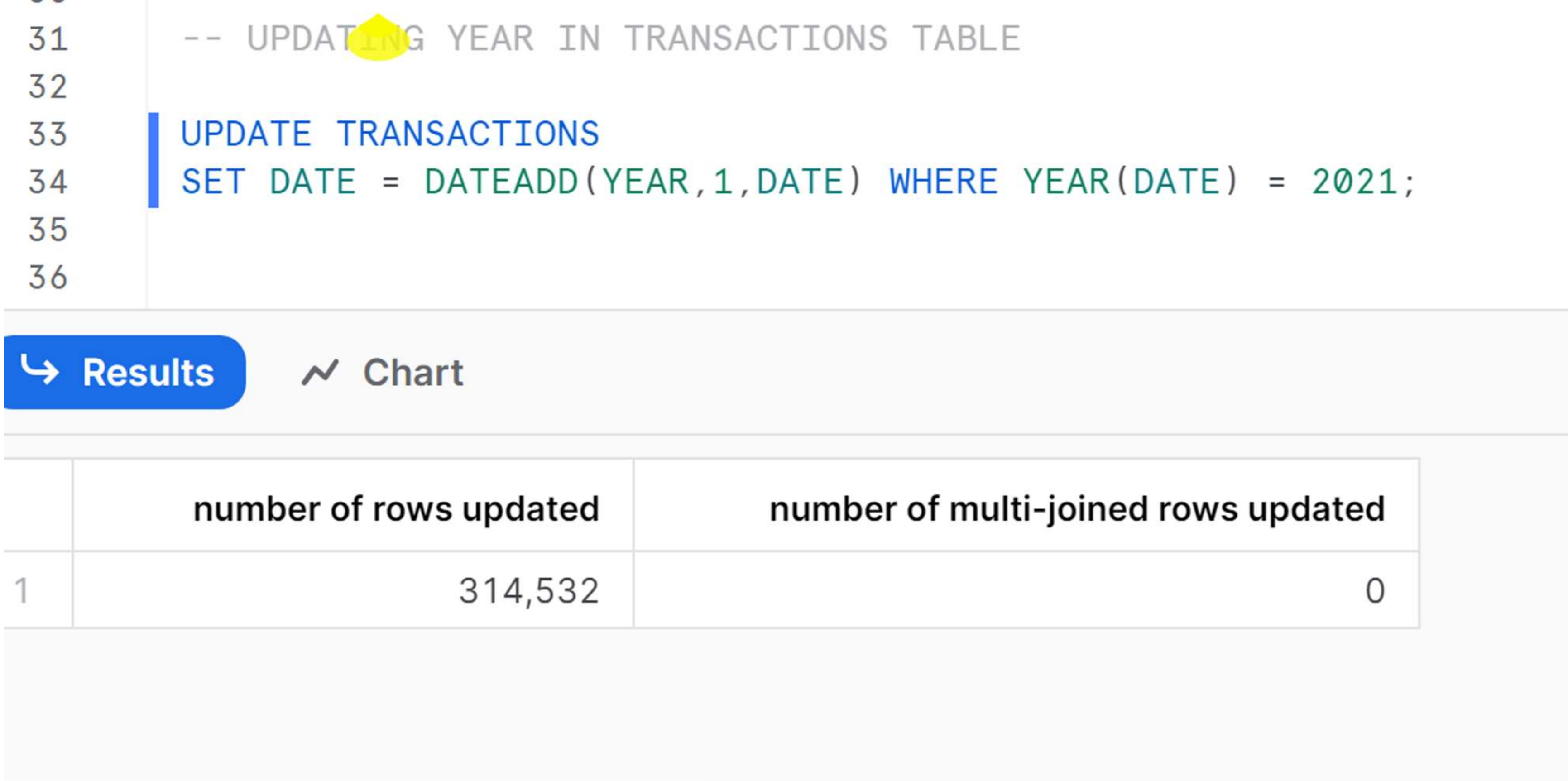
Press 'space' for AI, '/' for commands…

In a data transformation , the following changes were made to the 'TXN_YEAR'

- Changed '2021' to '2022'.
- Updated '2020' to '2021'.
- Transformed '2018' to '2020'.
- Modified '2017' to '2019'.
- Altered '2016' to '2018'.

These changes reflect a shift in the transaction year values, updating them to the specified target years.

```
--DATA TRANSFORMATION /* CONVERT 2021 TXN_YEAR TO 2022 CONVERT 2020 TXN_YEAR TO 2021 CONVERT 2018 TXN_YEAR TO
2020 CONVERT 2017 TXN_YEAR TO 2019 CONVERT 2016 TXN_YEAR TO 2018 WE HAVE TO CHANGE YEAR IN THESE TABLES ➡CARD
➡LOAN ➡TRANSACTIONS */ -- CHECKING YEAR IN AACCOUNT TABLE SELECT DISTINCT YEAR(DATE) FROM ACCOUNT; --
CHECKING YEAR IN TRANSACTIONS TABLE SELECT DISTINCT YEAR(DATE) FROM TRANSACTIONS; -- CHECKING YEAR IN CARD
TABLE SELECT DISTINCT YEAR(ISSUED) FROM CARD; -- CHECKING YEAR IN LOAN TABLE SELECT DISTINCT YEAR(DATE) FROM
LOAN; -- UPDATING YEAR IN TRANSACTIONS TABLE -- CONVERT 2021 TXN_YEAR TO 2022 UPDATE TRANSACTIONS SET DATE =
DATEADD(YEAR,1,DATE) WHERE YEAR(DATE) = 2021;
```

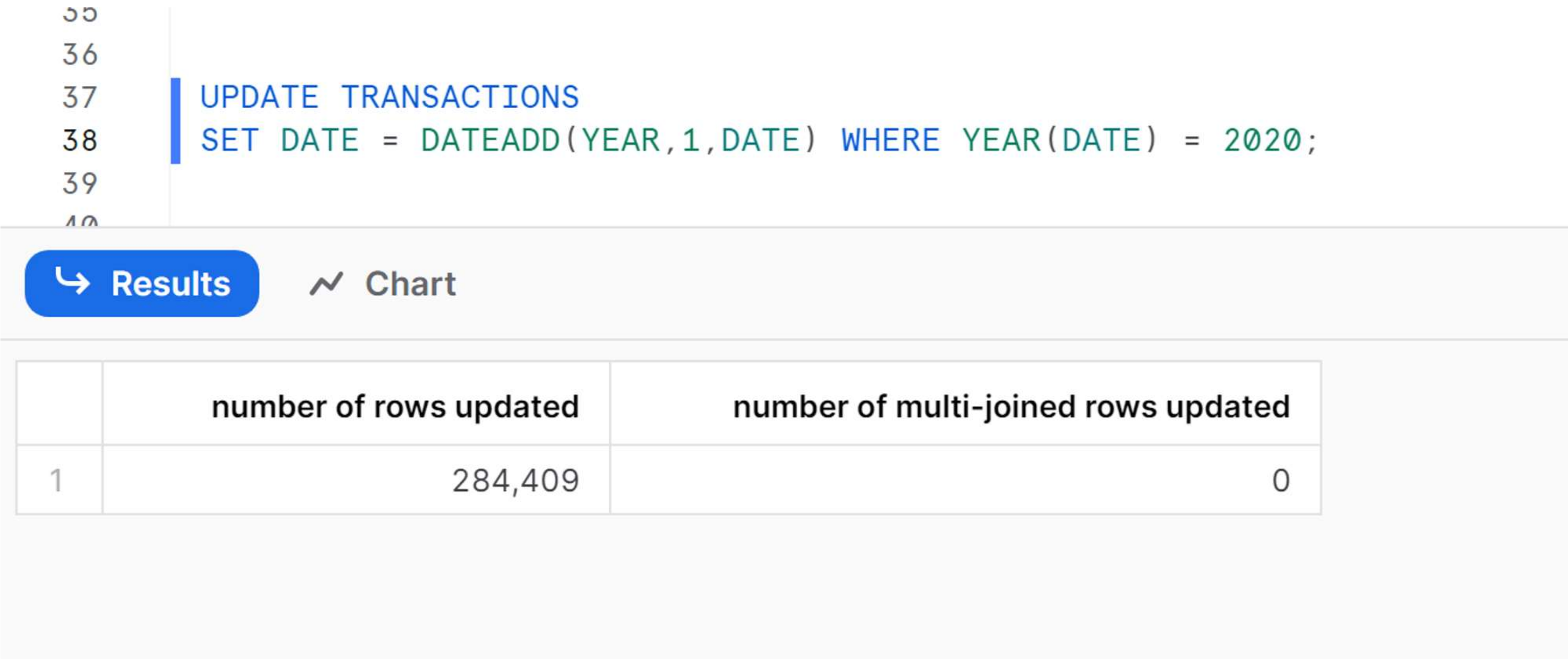


```
-- UPDATING YEAR IN TRANSACTIONS TABLE

UPDATE TRANSACTIONS
SET DATE = DATEADD(YEAR,1,DATE) WHERE YEAR(DATE) = 2021;
```

	number of rows updated	number of multi-joined rows updated
1	314,532	0

CONVERT 2020 TXN_YEAR TO 2021

```sql
UPDATE TRANSACTIONS SET DATE = DATEADD(YEAR,1,DATE) WHERE YEAR(DATE) = 2020;
```

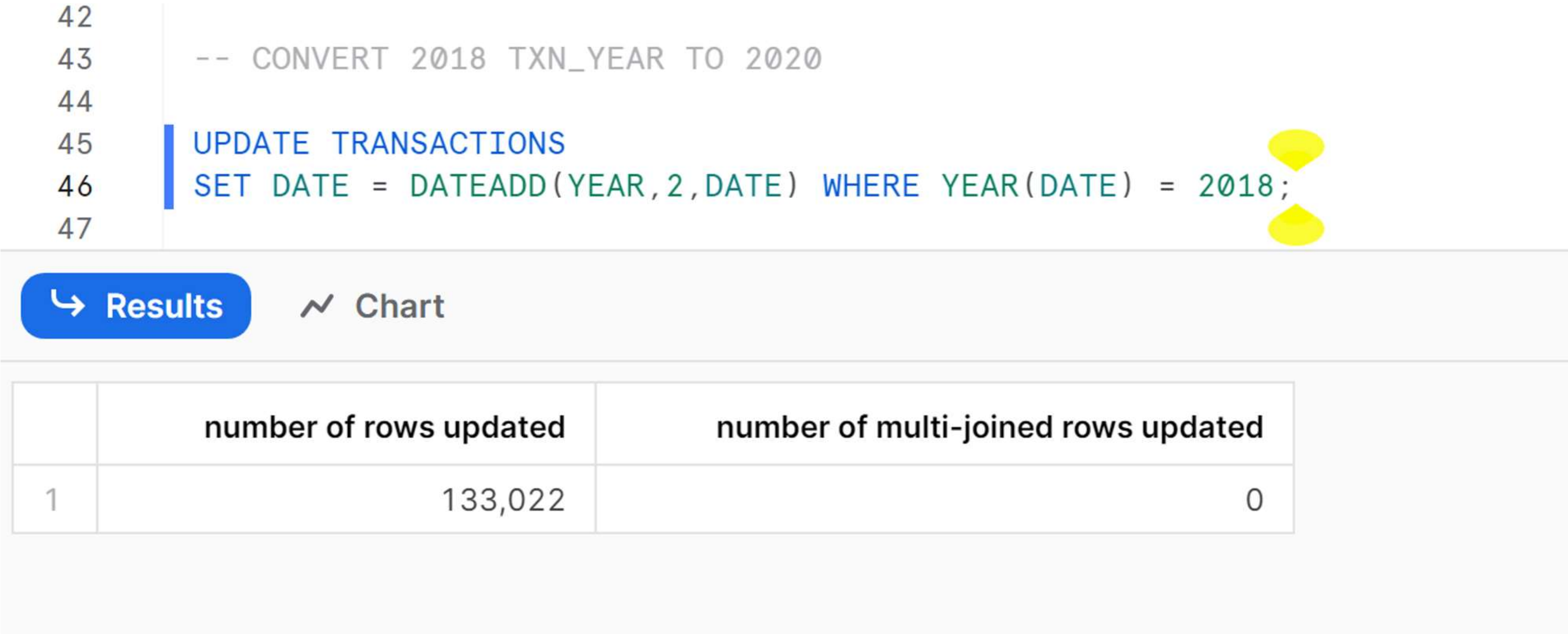


```sql
UPDATE TRANSACTIONS
SET DATE = DATEADD(YEAR,1,DATE) WHERE YEAR(DATE) = 2020;
```

	number of rows updated	number of multi-joined rows updated
1	284,409	0

CONVERT 2018 TXN_YEAR TO 2020

```
UPDATE TRANSACTIONS SET DATE = DATEADD(YEAR,2,DATE) WHERE YEAR(DATE) = 2018;
```

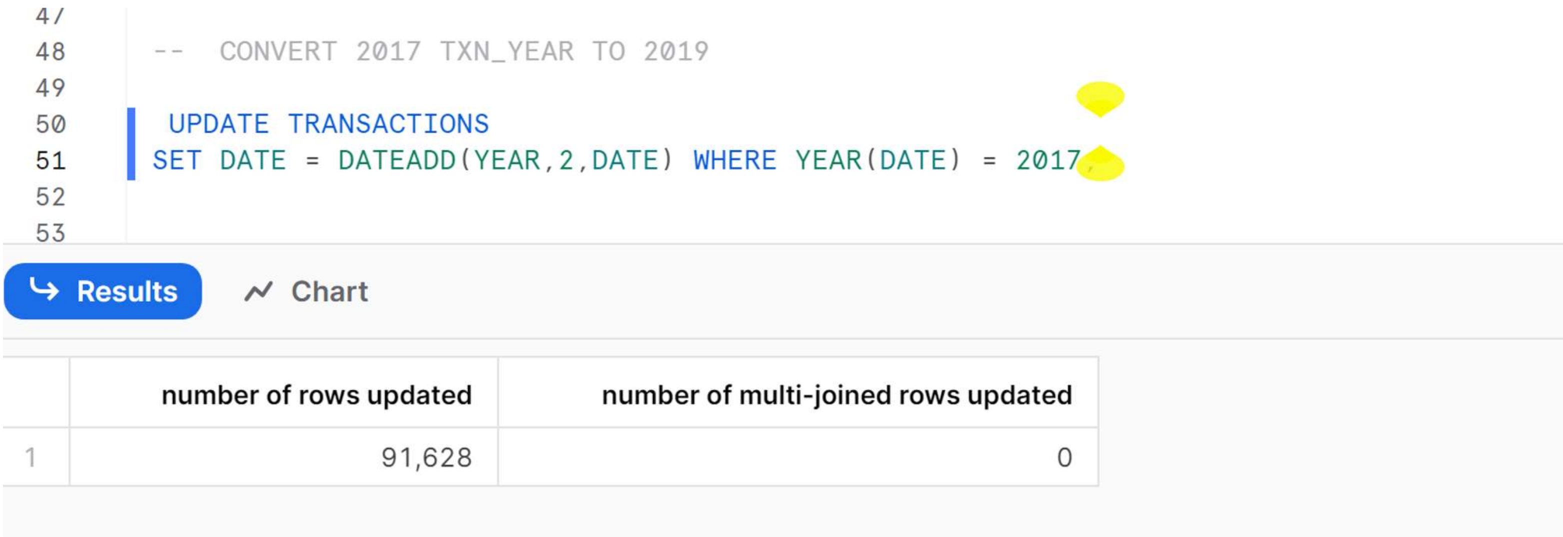


```

-- CONVERT 2018 TXN_YEAR TO 2020

UPDATE TRANSACTIONS
SET DATE = DATEADD(YEAR,2,DATE) WHERE YEAR(DATE) = 2018;

```

	number of rows updated	number of multi-joined rows updated
1	133,022	0

CONVERT 2017 TXN_YEAR TO 2019

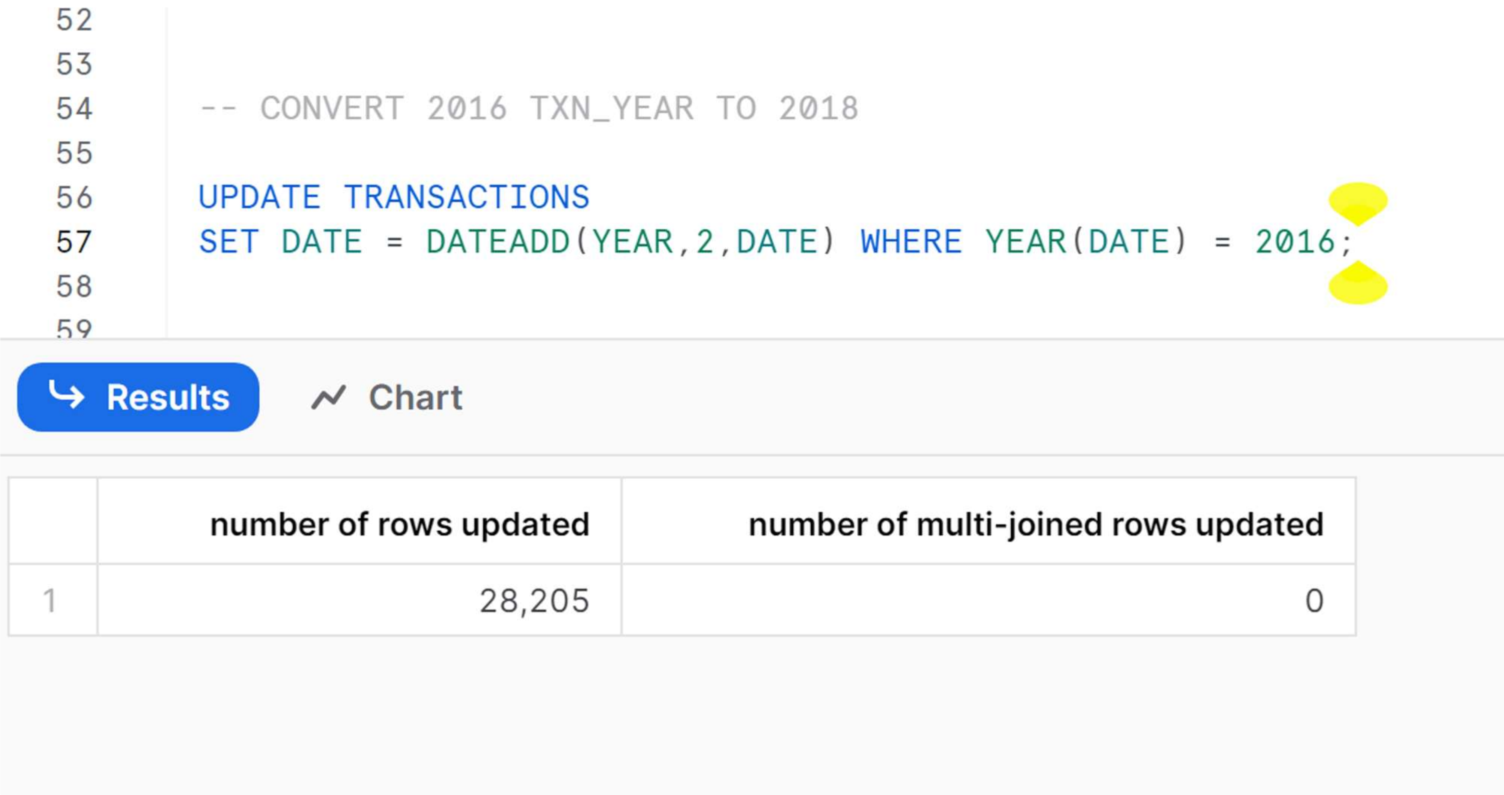
```
UPDATE TRANSACTIONS SET DATE = DATEADD(YEAR,2,DATE) WHERE YEAR(DATE) = 2017;
```



```
-- CONVERT 2017 TXN_YEAR TO 2019

 UPDATE TRANSACTIONS
SET DATE = DATEADD(YEAR,2,DATE) WHERE YEAR(DATE) = 2017;
```

	number of rows updated	number of multi-joined rows updated
1	91,628	0

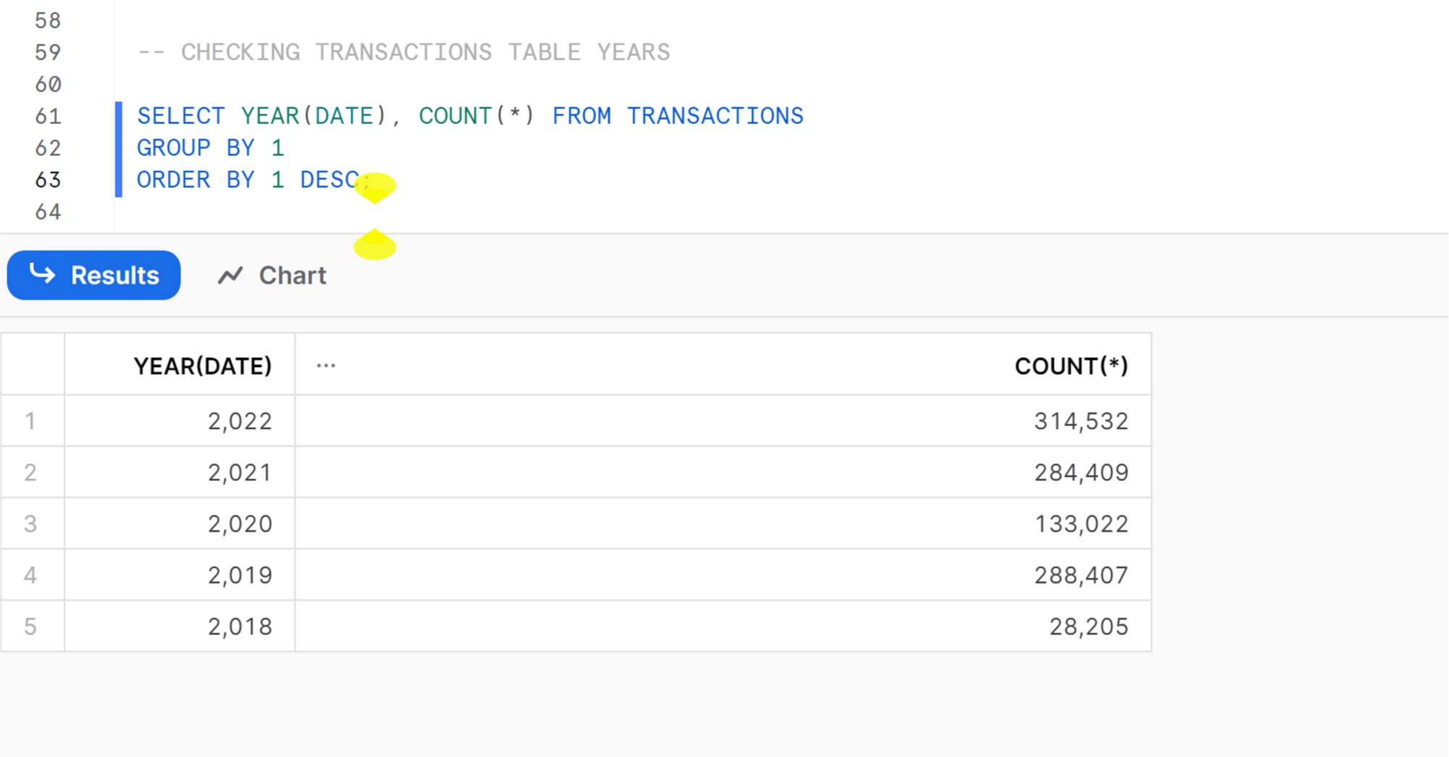
CONVERT 2016 TXN_YEAR TO 2018



```
-- CONVERT 2016 TXN_YEAR TO 2018

UPDATE TRANSACTIONS
SET DATE = DATEADD(YEAR,2,DATE) WHERE YEAR(DATE) = 2016;
```

	number of rows updated	number of multi-joined rows updated
1	28,205	0

CHECKING THE YEAR UPDATE IN THE TRANSACTION TABLE HERE YEAR WILL BE 2018, 2019, 2020, 2021, 2022



```
-- CHECKING TRANSACTIONS TABLE YEARS

SELECT YEAR(DATE), COUNT(*) FROM TRANSACTIONS
GROUP BY 1
ORDER BY 1 DESC;
```

	YEAR(DATE)	COUNT(*)
1	2,022	314,532
2	2,021	284,409
3	2,020	133,022
4	2,019	288,407
5	2,018	28,205

UPDATING YEAR IN TRANSACTIONS TABLE SQL COMMAND

```
-- UPDATING YEAR IN TRANSACTIONS TABLE -- CONVERT 2021 TXN_YEAR TO 2022 UPDATE
TRANSACTIONS SET DATE = DATEADD(YEAR,1,DATE) WHERE YEAR(DATE) = 2021; -- CONVERT 2020
TXN_YEAR TO 2021 UPDATE TRANSACTIONS SET DATE = DATEADD(YEAR,1,DATE) WHERE YEAR(DATE) =
2020; -- CONVERT 2018 TXN_YEAR TO 2020 UPDATE TRANSACTIONS SET DATE =
DATEADD(YEAR,2,DATE) WHERE YEAR(DATE) = 2018; -- CONVERT 2017 TXN_YEAR TO 2019 UPDATE
TRANSACTIONS SET DATE = DATEADD(YEAR,2,DATE) WHERE YEAR(DATE) = 2017; -- CONVERT 2016
TXN_YEAR TO 2018 UPDATE TRANSACTIONS SET DATE = DATEADD(YEAR,2,DATE) WHERE YEAR(DATE) =
2016; -- CHECKING TRANSACTIONS TABLE YEARS SELECT YEAR(DATE), COUNT(*) FROM TRANSACTIONS
GROUP BY 1 ORDER BY 1 DESC;
```

```
/*UPDATE ALL DATA AT ONCE*/ UPDATE TRANSACTIONS SET DATE = CASE WHEN YEAR(DATE) = 2016
THEN DATEADD(YEAR,2,DATE) WHEN YEAR(DATE) = 2017 THEN DATEADD(YEAR,2,DATE) WHEN
YEAR(DATE) = 2018 THEN DATEADD(YEAR,2,DATE) WHEN YEAR(DATE) = 2020 THEN
DATEADD(YEAR,1,DATE) WHEN YEAR(DATE) = 2021 THEN DATEADD(YEAR,1,DATE) ELSE DATE END;
```

Explanation

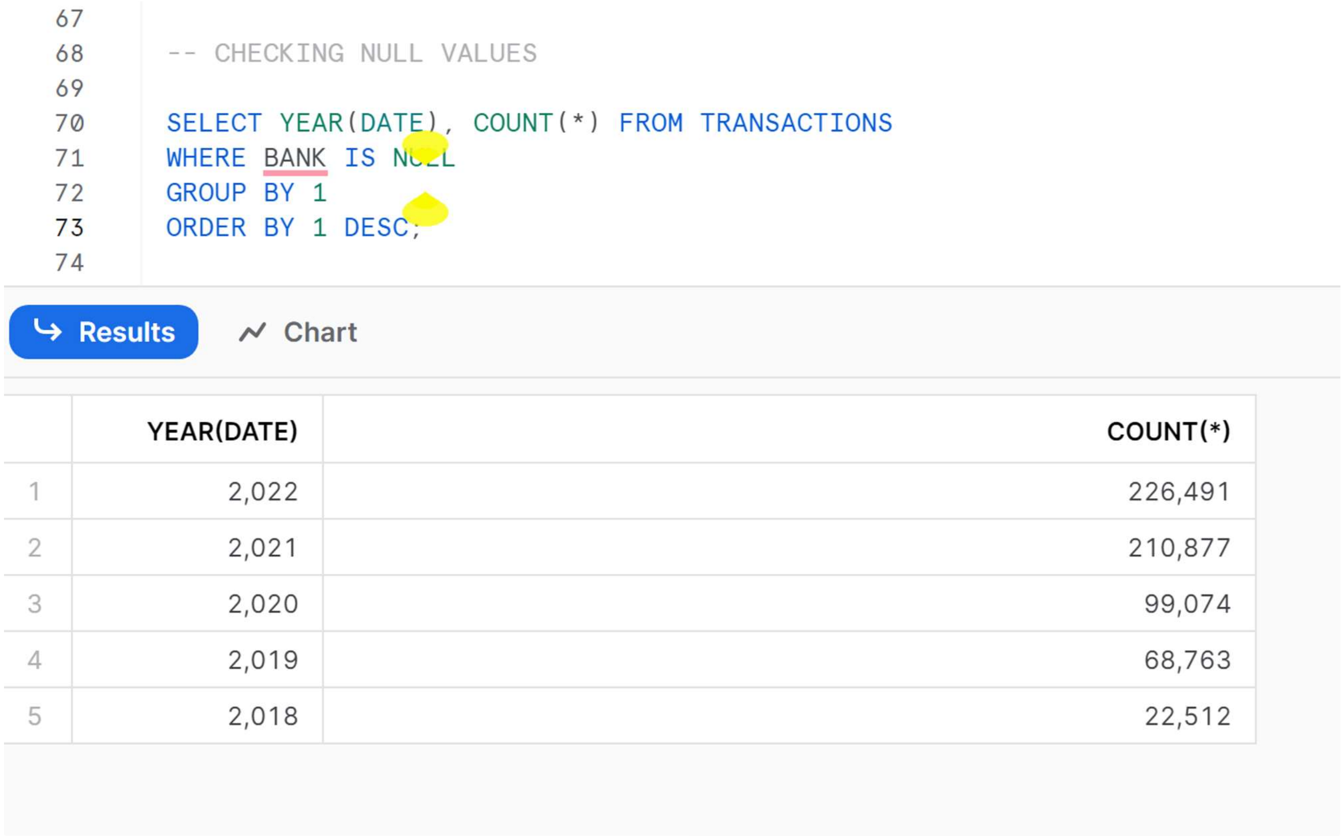
This SQL statement is an `UPDATE` query for a table named "TRANSACTIONS." It updates the values in the "DATE" column of that table based on the year of the existing date values. Here's a step-by-step explanation:

1. `UPDATE TRANSACTIONS`: This part of the SQL statement indicates that you want to modify data in the "TRANSACTIONS" table.
2. `SET DATE = ...`: This part specifies that you want to update the values in the "DATE" column of the "TRANSACTIONS" table.

3. `CASE ... END`: This is a `CASE` expression, which is used to conditionally determine the new values for the "DATE" column based on the year of the existing dates.
4. Inside the `CASE` expression, there are several `WHEN` clauses. Each `WHEN` clause checks the year of the existing "DATE" value and specifies how to update it based on the year.
    - `WHEN YEAR(DATE) = 2016 THEN DATEADD(YEAR, 2, DATE)`: If the year of the existing date is 2016, it adds 2 years to the existing date using the `DATEADD` function. This effectively shifts the date two years into the future.
    - `WHEN YEAR(DATE) = 2017 THEN DATEADD(YEAR, 2, DATE)`: If the year is 2017, it also adds 2 years to the existing date.
    - `WHEN YEAR(DATE) = 2018 THEN DATEADD(YEAR, 2, DATE)`: The same logic is applied if the year is 2018.
    - `WHEN YEAR(DATE) = 2020 THEN DATEADD(YEAR, 1, DATE)`: If the year is 2020, it adds 1 year to the existing date.
    - `WHEN YEAR(DATE) = 2021 THEN DATEADD(YEAR, 1, DATE)`: Similarly, if the year is 2021, it adds 1 year to the date.
    - `ELSE DATE`: If none of the above conditions are met (i.e., if the year is not 2016, 2017, 2018, 2020, or 2021), it keeps the existing date unchanged.

The `DATEADD` function is used to add years to the existing date. So, this query updates the "DATE" column in the "TRANSACTIONS" table, adjusting the date values based on the specified conditions for different years, while leaving the date unchanged if it doesn't match any of the specified conditions.

▾ CHECKING NULL VALUE IN EVERY YEAR



```
-- CHECKING NULL VALUES

SELECT YEAR(DATE), COUNT(*) FROM TRANSACTIONS
WHERE BANK IS NULL
GROUP BY 1
ORDER BY 1 DESC;
```

	YEAR(DATE)	COUNT(*)
1	2,022	226,491
2	2,021	210,877
3	2,020	99,074
4	2,019	68,763
5	2,018	22,512

Now we have to update the Bank name where Bank name is null

- - UPDATE BANK WHERE IT IS NULL

/*

2022 - Sky Bank

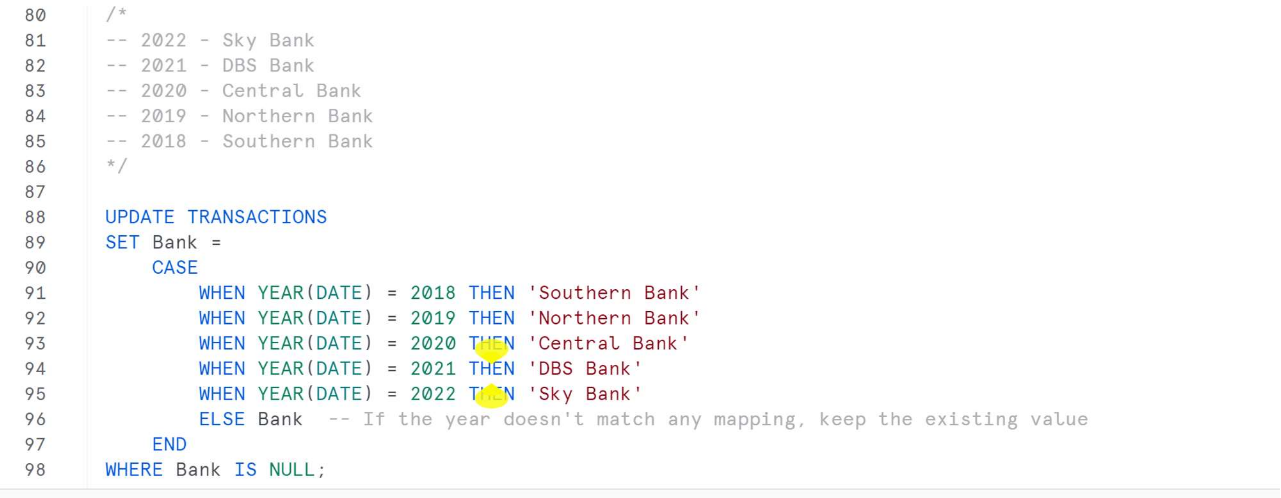
2021 - DBS Bank

2020 - Central Bank

2019 - Northern Bank

2018 - Southern Bank

- /



```
/*
-- 2022 - Sky Bank
-- 2021 - DBS Bank
-- 2020 - Central Bank
-- 2019 - Northern Bank
-- 2018 - Southern Bank
*/

UPDATE TRANSACTIONS
SET Bank =
    CASE
        WHEN YEAR(DATE) = 2018 THEN 'Southern Bank'
        WHEN YEAR(DATE) = 2019 THEN 'Northern Bank'
        WHEN YEAR(DATE) = 2020 THEN 'Central Bank'
        WHEN YEAR(DATE) = 2021 THEN 'DBS Bank'
        WHEN YEAR(DATE) = 2022 THEN 'Sky Bank'
        ELSE Bank  -- If the year doesn't match any mapping, keep the existing value
    END
WHERE Bank IS NULL;
```