

# How to Find Duplicate Records in SQL

## What are Duplicates in SQL?

Duplicates in SQL are mostly the data points that exist more than once in our data store.

### **For example:**

If we consider a customer table and in it, we store the details of customers of any shop. Then here, duplicate records would be the entry of a single customer more than once.

In such cases, the system has an increased load of handling large data stores and their processing.

## Why do we need to handle Duplicates in SQL?

Some of the major reasons why we need to remove duplicates from our records are as follows:

1. The size of the data to be stored increases due to the duplicates.
2. When we have duplicates in our data they can give rise to business errors also known as logical errors.
3. Due to the increased use of resources, the overall cost of handling resources rises.
4. Errors arise when analytics are performed on such erroneous data.

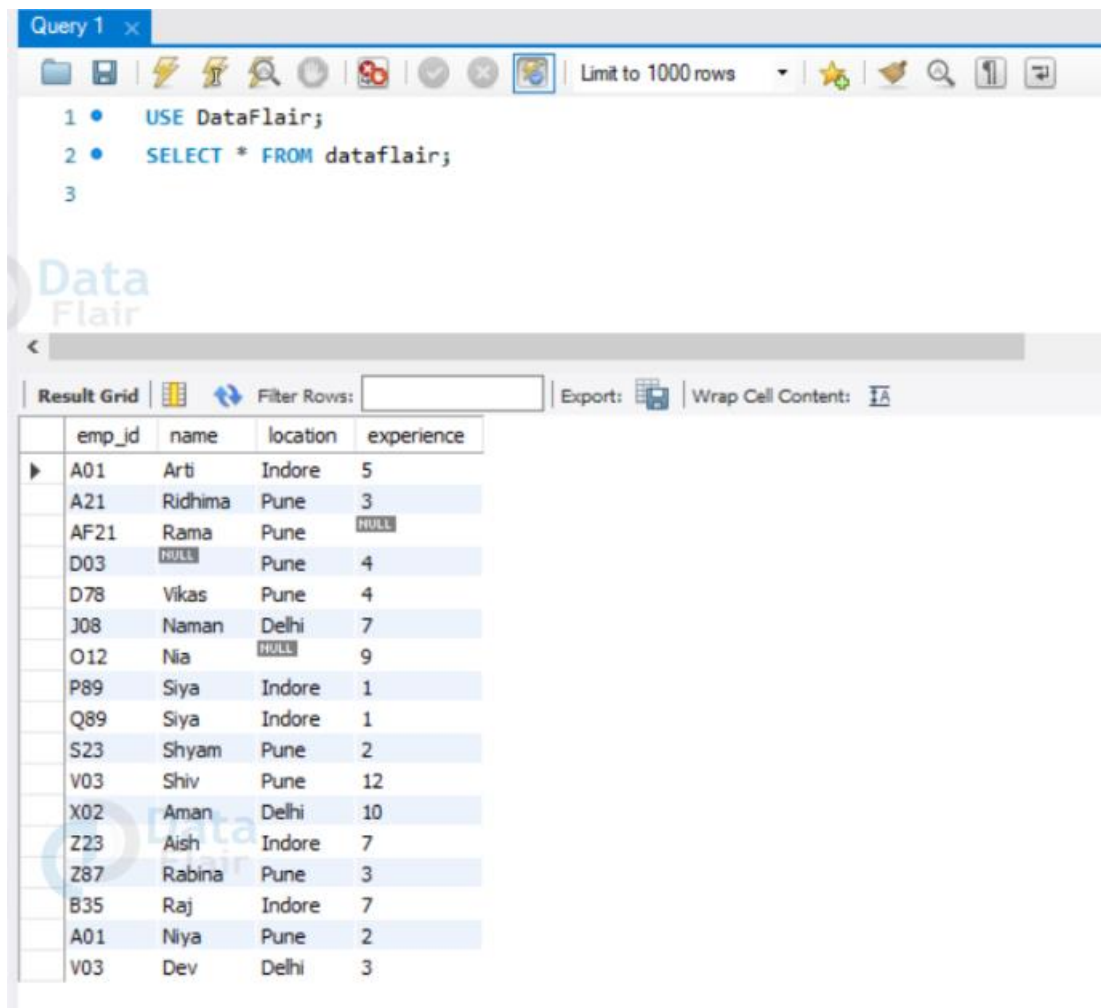
## Demo Database

Let us now have a look at our demo database: DataFlair.

### Query:

```
USE DataFlair;  
SELECT * FROM dataflair;
```

### Output:



Query 1 x



Limit to 1000 rows

```
1 • USE DataFlair;  
2 • SELECT * FROM dataflair;  
3
```

Data  
Flair

Result Grid

Filter Rows:

Export:  Wrap Cell Content: 

|   | emp_id | name    | location | experience |
|---|--------|---------|----------|------------|
| ▶ | A01    | Arti    | Indore   | 5          |
|   | A21    | Ridhima | Pune     | 3          |
|   | AF21   | Rama    | Pune     | NULL       |
|   | D03    | NULL    | Pune     | 4          |
|   | D78    | Vikas   | Pune     | 4          |
|   | J08    | Naman   | Delhi    | 7          |
|   | O12    | Nia     | NULL     | 9          |
|   | P89    | Siya    | Indore   | 1          |
|   | Q89    | Siya    | Indore   | 1          |
|   | S23    | Shyam   | Pune     | 2          |
|   | V03    | Shiv    | Pune     | 12         |
|   | X02    | Aman    | Delhi    | 10         |
|   | Z23    | Aish    | Indore   | 7          |
|   | Z87    | Rabina  | Pune     | 3          |
|   | B35    | Raj     | Indore   | 7          |
|   | A01    | Niya    | Pune     | 2          |
|   | V03    | Dev     | Delhi    | 3          |

## How to handle Duplicates in SQL?

We have various solutions to handle the duplicates in our database. Some of them are as follows:

# 1. Using the Distinct Keyword to eliminate duplicate values and count their occurrences from the Query results.

We can use the Distinct keyword to fetch the unique records from our database. This way we can view the unique results from our database.

## Syntax:

```
SELECT col1, col2, COUNT(DISTINCT(col3)),.....  
FROM tableName;
```

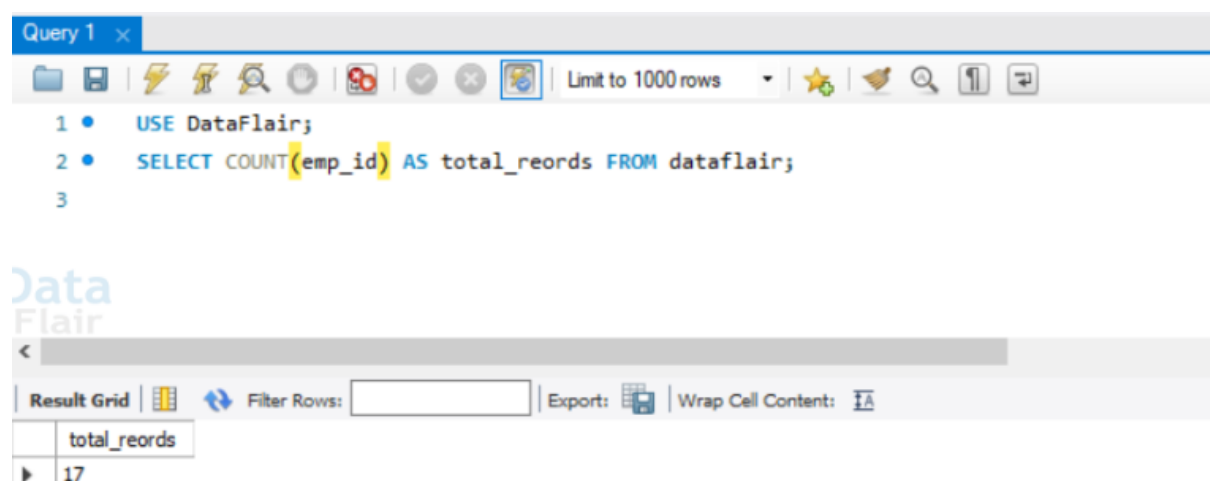
**Example:** Let us first try to view the count of unique id's in our records by finding the total records and then the number of unique records.

**Step 1:** View the count of all records in our database.

## Query:

```
USE DataFlair;  
SELECT COUNT(emp_id) AS total_reords FROM dataflair;
```

## Output:



The screenshot shows a SQL query editor window titled 'Query 1'. The query is: `USE DataFlair;` followed by `SELECT COUNT(emp_id) AS total_reords FROM dataflair;`. Below the query, the 'Result Grid' is displayed, showing a single row with the column name 'total\_reords' and the value '17'. The interface includes various icons for file operations, search, and execution, as well as a 'Limit to 1000 rows' dropdown.

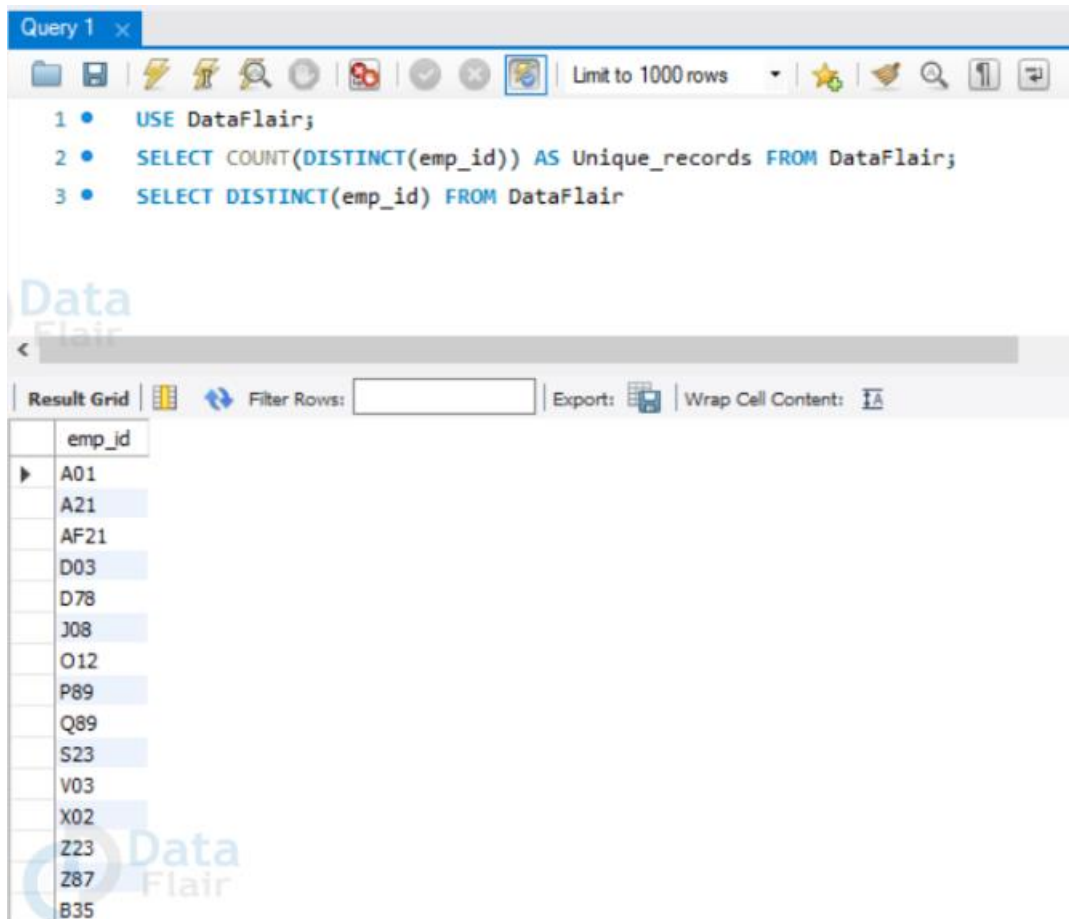
| total_reords |
|--------------|
| 17           |

**Step 2:** View the count of unique records in our database.

**Query:**

```
USE DataFlair;  
SELECT COUNT(DISTINCT(emp_id)) AS Unique_records FROM DataFlair;  
SELECT DISTINCT(emp_id) FROM DataFlair;
```

**Output:**



Query 1 x

Limit to 1000 rows

```
1 • USE DataFlair;  
2 • SELECT COUNT(DISTINCT(emp_id)) AS Unique_records FROM DataFlair;  
3 • SELECT DISTINCT(emp_id) FROM DataFlair
```

DataFlair

Result Grid

| emp_id |
|--------|
| A01    |
| A21    |
| AF21   |
| D03    |
| D78    |
| J08    |
| O12    |
| P89    |
| Q89    |
| S23    |
| V03    |
| X02    |
| Z23    |
| Z87    |
| B35    |

## 2. Using Distinct keywords to delete Duplicate records from the database.

**Syntax:**

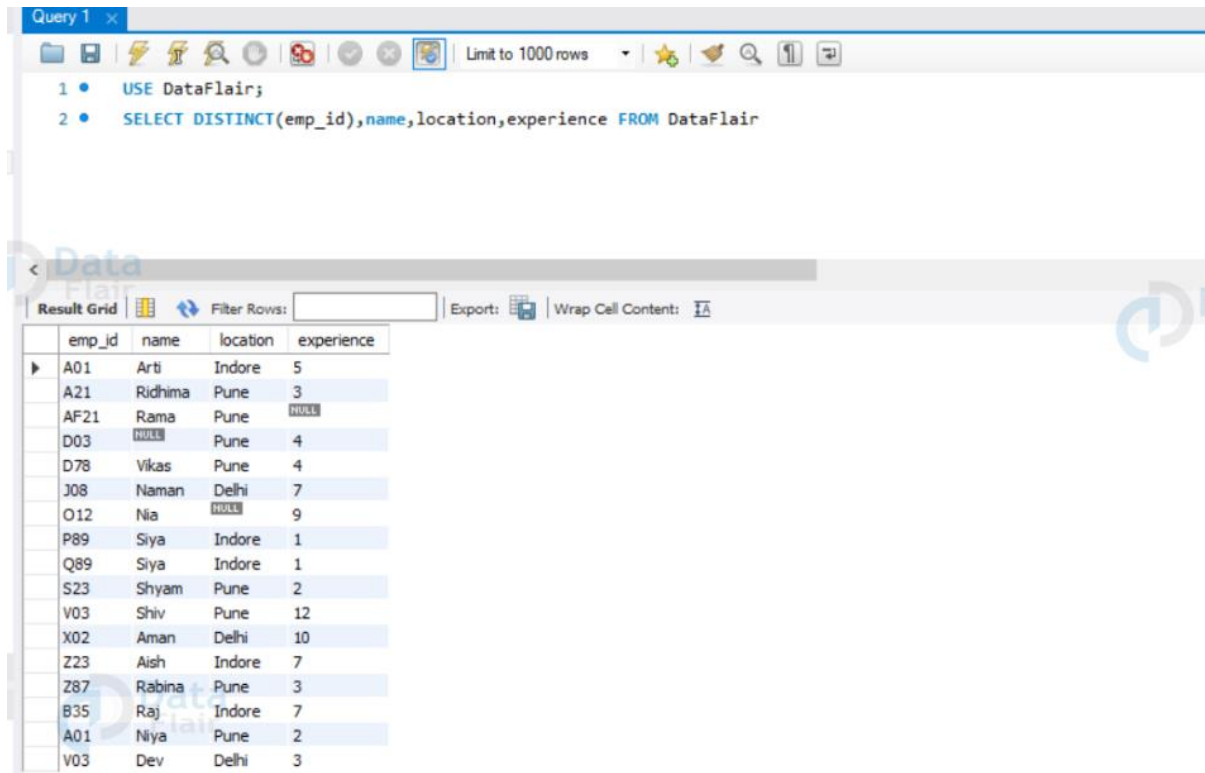
```
SELECT col1, col2, DISTINCT(col3),.....  
FROM tableName;
```

**Example:** Let us now view our table after deleting the duplicate records in our database.

**Query:**

```
USE DataFlair;  
SELECT DISTINCT(emp_id),name, location,experience FROM DataFlair
```

**Output:**



| emp_id | name    | location | experience |
|--------|---------|----------|------------|
| A01    | Arti    | Indore   | 5          |
| A21    | Ridhima | Pune     | 3          |
| AF21   | Rama    | Pune     | NULL       |
| D03    | NULL    | Pune     | 4          |
| D78    | Vikas   | Pune     | 4          |
| J08    | Naman   | Delhi    | 7          |
| O12    | Nia     | NULL     | 9          |
| P89    | Siya    | Indore   | 1          |
| Q89    | Siya    | Indore   | 1          |
| S23    | Shyam   | Pune     | 2          |
| V03    | Shiv    | Pune     | 12         |
| X02    | Aman    | Delhi    | 10         |
| Z23    | Aish    | Indore   | 7          |
| Z87    | Rabina  | Pune     | 3          |
| B35    | Raj     | Indore   | 7          |
| A01    | Niya    | Pune     | 2          |
| V03    | Dev     | Delhi    | 3          |

### 3. Using the Count keyword and Group By to eliminate duplicate records.

**Syntax:**

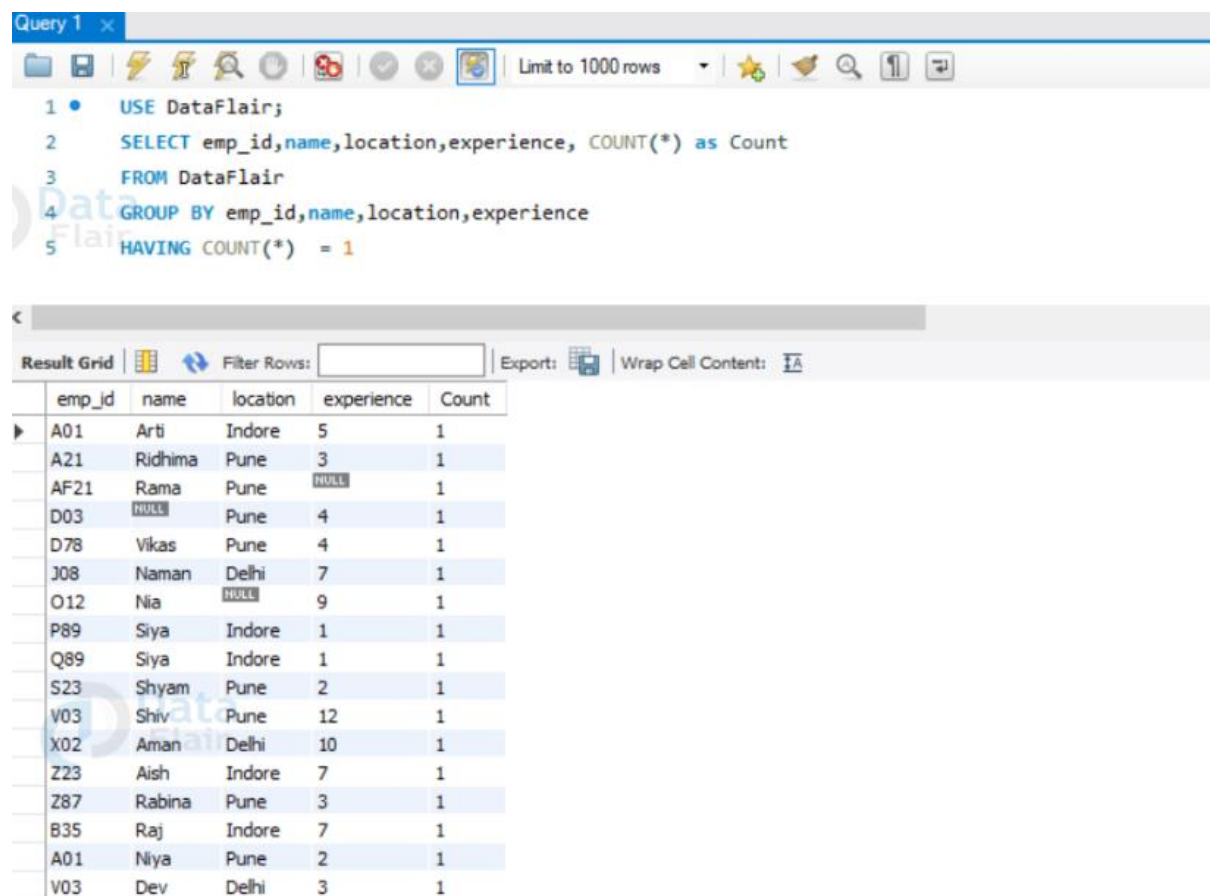
```
SELECT col1,col2,col3,.....  
COUNT(*) AS aliasName  
FROM tableName  
GROUP BY col1,col2,col3,.....  
HAVING COUNT(*) = 1;
```

**Example:** Let us now eliminate the duplicate records of our database using the Count and the Group By keyword.

**Query:**

```
USE DataFlair;
SELECT emp_id,name,location,experience, COUNT(*) as Count
FROM DataFlair
GROUP BY emp_id,name,location, experience
HAVING COUNT(*) = 1
```

**Output:**



The screenshot shows a SQL query editor with a query window and a result grid. The query window contains the following SQL code:

```
1 • USE DataFlair;
2 SELECT emp_id,name,location,experience, COUNT(*) as Count
3 FROM DataFlair
4 GROUP BY emp_id,name,location,experience
5 HAVING COUNT(*) = 1
```

The result grid displays the output of the query, showing a list of employees with their IDs, names, locations, experiences, and counts. The columns are labeled: emp\_id, name, location, experience, and Count. The data is as follows:

| emp_id | name    | location | experience | Count |
|--------|---------|----------|------------|-------|
| A01    | Arti    | Indore   | 5          | 1     |
| A21    | Ridhima | Pune     | 3          | 1     |
| AF21   | Rama    | Pune     | HULL       | 1     |
| D03    | HULL    | Pune     | 4          | 1     |
| D78    | Vikas   | Pune     | 4          | 1     |
| J08    | Naman   | Delhi    | 7          | 1     |
| O12    | Nia     | HULL     | 9          | 1     |
| P89    | Siya    | Indore   | 1          | 1     |
| Q89    | Siya    | Indore   | 1          | 1     |
| S23    | Shyam   | Pune     | 2          | 1     |
| V03    | Shiv    | Pune     | 12         | 1     |
| X02    | Aman    | Delhi    | 10         | 1     |
| Z23    | Aish    | Indore   | 7          | 1     |
| Z87    | Rabina  | Pune     | 3          | 1     |
| B35    | Raj     | Indore   | 7          | 1     |
| A01    | Niya    | Pune     | 2          | 1     |
| V03    | Dev     | Delhi    | 3          | 1     |

## 4. Using Joins to eliminate Duplicate records.

**Syntax:**

```
SELECT col1,col2,col3,....
FROM tableName alias
WHERE EXISTS(SELECT 1 FROM tableName alias2
WHERE Alias.col1 = Alias1.col1 AND
Alias.col2 = Alias2.col2 AND
Alias.col2 = Alias2.col2 AND
```

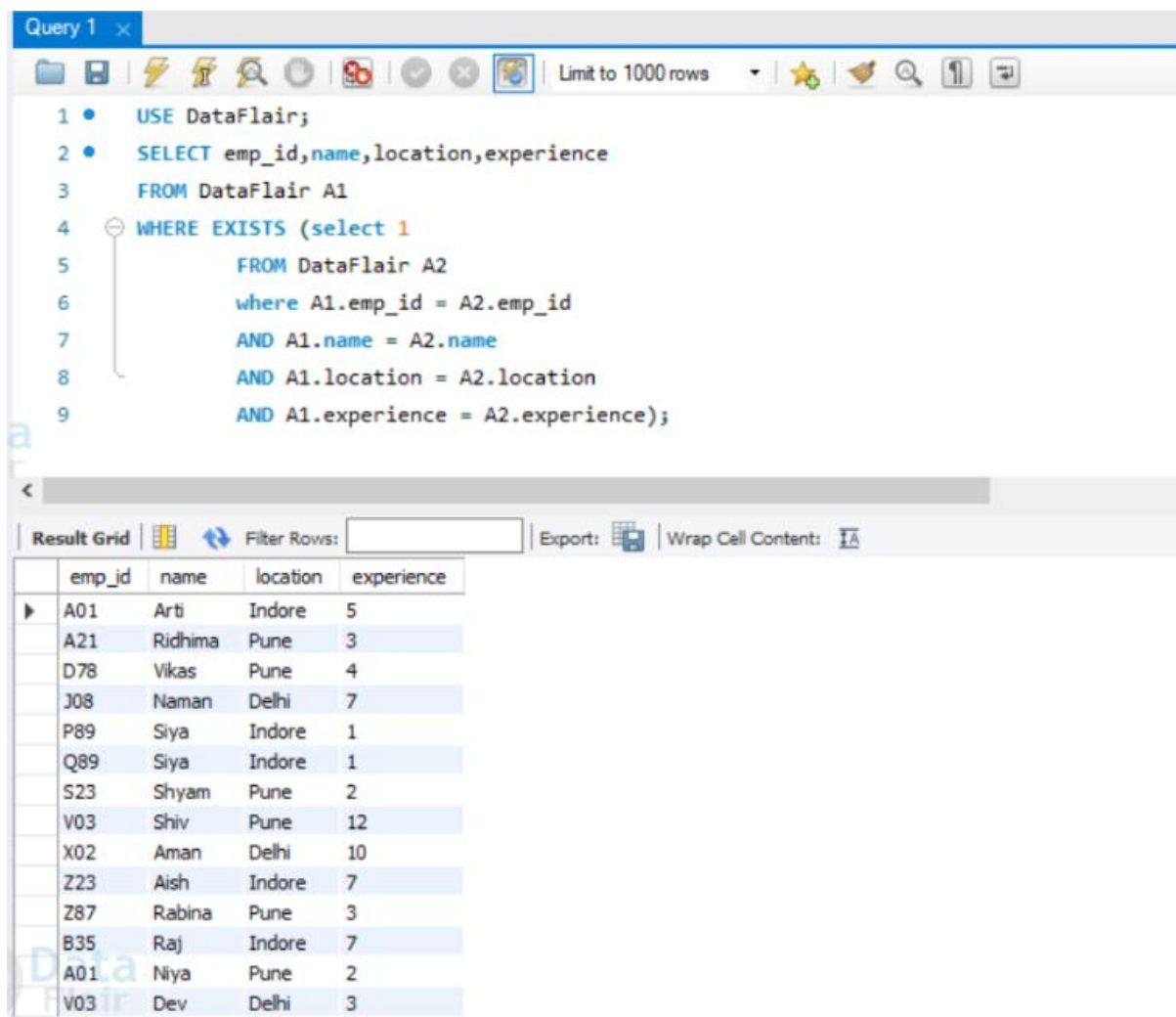
...);

**Example:** Let us now try to remove the duplicates from our data using the Join operation on our data.

### Query:

```
USE DataFlair;
SELECT emp_id,name, location,experience
FROM DataFlair A1
WHERE EXISTS (SELECT 1
              FROM DataFlair A2
              where A1.emp_id = A2.emp_id
              AND A1.name = A2.name
              AND A1.location = A2.location
              AND A1.experience = A2.experience);
```

### Output:



The screenshot shows a SQL query editor window titled 'Query 1'. The query is as follows:

```
1 • USE DataFlair;
2 • SELECT emp_id,name,location,experience
3   FROM DataFlair A1
4 WHERE EXISTS (select 1
5               FROM DataFlair A2
6               where A1.emp_id = A2.emp_id
7               AND A1.name = A2.name
8               AND A1.location = A2.location
9               AND A1.experience = A2.experience);
```

Below the query editor, the 'Result Grid' is displayed, showing the output of the query. The grid has columns for emp\_id, name, location, and experience. The data is as follows:

| emp_id | name    | location | experience |
|--------|---------|----------|------------|
| A01    | Arti    | Indore   | 5          |
| A21    | Ridhima | Pune     | 3          |
| D78    | Vikas   | Pune     | 4          |
| J08    | Naman   | Delhi    | 7          |
| P89    | Siya    | Indore   | 1          |
| Q89    | Siya    | Indore   | 1          |
| S23    | Shyam   | Pune     | 2          |
| V03    | Shiv    | Pune     | 12         |
| X02    | Aman    | Delhi    | 10         |
| Z23    | Aish    | Indore   | 7          |
| Z87    | Rabina  | Pune     | 3          |
| B35    | Raj     | Indore   | 7          |
| A01    | Niya    | Pune     | 2          |
| V03    | Dev     | Delhi    | 3          |

# Summary

we have understood what are duplicates and how do they affect our business. We have then also discussed various methods by which we can either handle or completely remove duplicates from our database.