SQL TOP, LIMIT, FETCH FIRST CLAUSE

SQL TOP, LIMIT, and FETCH FIRST clauses are used to retrieve a specific number of records from a table. These clauses are especially useful in large datasets with thousands of records. Each of these SQL clauses performs a similar operation of limiting the results returned by a query, but they are supported by different database management systems:

+ **SQL TOP Clause** is used in SQL Server and Sybase to limit the number of records returned.
+ **SQL LIMIT Clause** is utilized in MySQL, PostgreSQL, and SQLite.
+ **SQL FETCH FIRST Clause** is part of the SQL standard and is supported by Oracle, DB2, PostgreSQL, and SQL Server (as part of OFFSET-FETCH).

Depending on the database management system (DBMS) being used, you can utilize the respective clause to efficiently manage data retrieval. This article will provide examples and guidance on how to use the SQL TOP, LIMIT, and FETCH FIRST clauses in different SQL environments.

SQL SELECT TOP Clause

The **SELECT TOP clause in SQL** only returns the specified number of rows from the table. It is valuable on enormous tables with a large number of records. Returning countless records can affect execution.

Note: Not all database systems support the SELECT TOP clause.

The SQL TOP keyword is utilized with these database systems:

+ **SQL Server**

+ **MS Access**

Syntax

```
SELECT [column1, column2, …] TOP [count]
FROM [table_name]
{WHERE [conditions]}
{ORDER BY [expression] [ ASC | DESC ]};
```

Here,

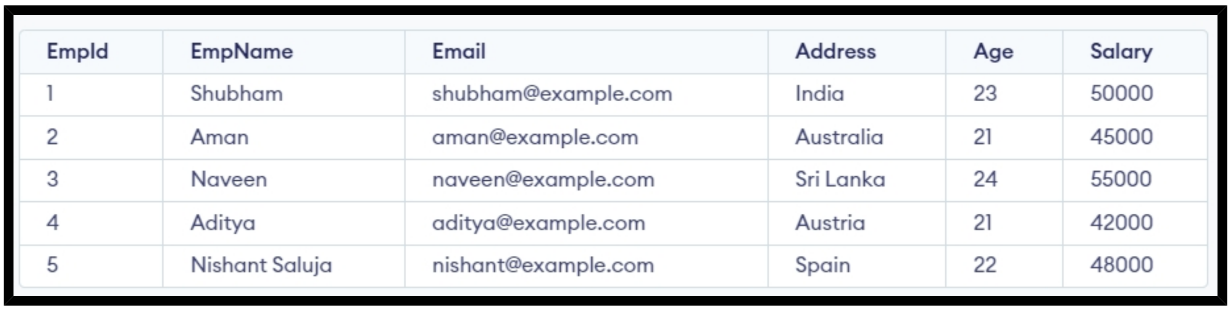
+ **column1, column2** = names of columns

+ **count** = number of records to be fetched

SQL SELECT TOP Clause Example

Let’s understand this using an example of SQL SELECT TOP statement.

We will use the following table for this example:



EmpId	EmpName	Email	Address	Age	Salary
1	Shubham	shubham@example.com	India	23	50000
2	Aman	aman@example.com	Australia	21	45000
3	Naveen	naveen@example.com	Sri Lanka	24	55000
4	Aditya	aditya@example.com	Austria	21	42000
5	Nishant Saluja	nishant@example.com	Spain	22	48000

Write the following SQL queries to create this table

```
CREATE TABLE Employee (
   EmpId INTEGER PRIMARY KEY,
   EmpName VARCHAR(225) NOT NULL,
   Email VARCHAR(225) NOT NULL,
   Address VARCHAR(225) NOT NULL,
   Age INT NOT NULL,
   Salary MONEY NOT NULL
);
INSERT INTO Employee (EmpId, EmpName, Email, Address, Age,
Salary)
VALUES (1, 'Shubham', 'shubham@example.com', 'India', 23,
50000.00),
       (2, 'Aman', 'aman@example.com', 'Australia', 21,
45000.00),
       (3, 'Naveen', 'naveen@example.com', 'Sri Lanka', 24,
55000.00),
       (4, 'Aditya', 'aditya@example.com', 'Austria', 21,
42000.00),
       (5, 'Nishant Saluja', 'nishant@example.com',
'Spain', 22, 48000.00);
```

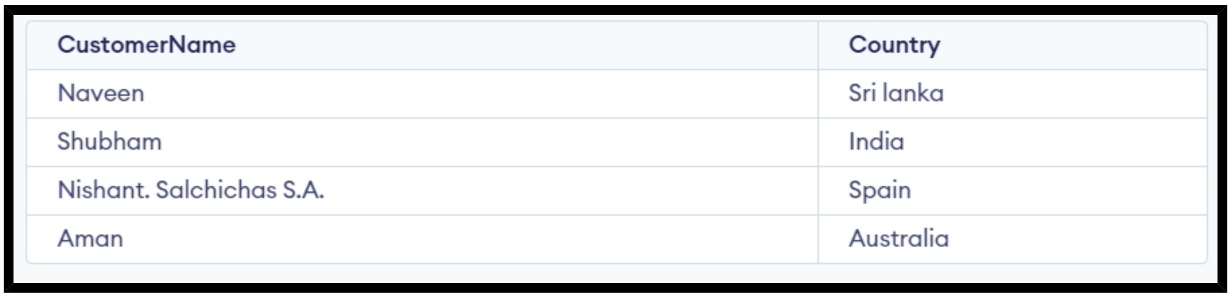
Using SELECT TOP Clause In SQL

In this example, we will fetch the top 4 rows from the table.

Query

```
SELECT TOP 4*
FROM Customer;
```

Output



CustomerName	Country
Naveen	Sri lanka
Shubham	India
Nishant. Salchichas S.A.	Spain
Aman	Australia

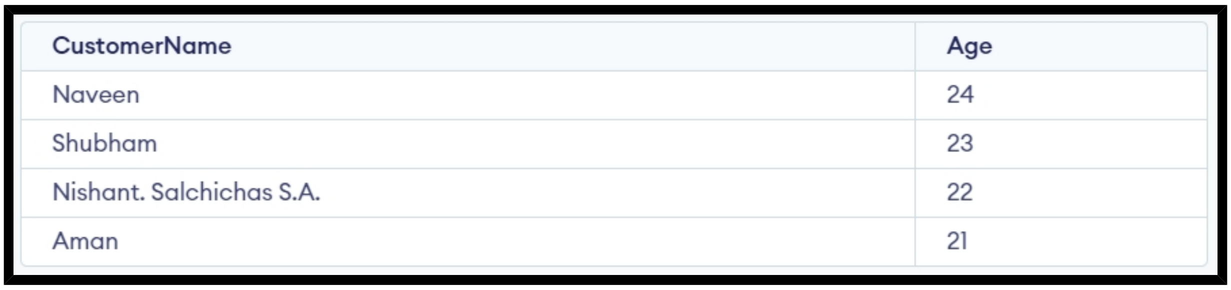
SQL SELECT TOP With ORDER BY Clause Example

In this example, we will use the SQL SELECT TOP clause with ORDER BY clause to sort the data in the results set.

Query

```
SELECT TOP 4*
FROM Customer
ORDER BY Salary DESC;
```

Output



CustomerName	Age
Naveen	24
Shubham	23
Nishant. Salchichas S.A.	22
Aman	21

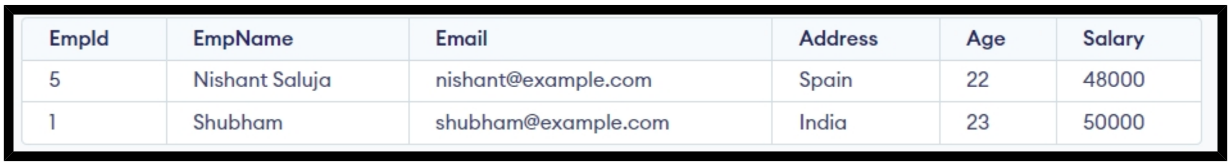
SQL SELECT TOP Clause With WHERE Clause Example

In this example, we will use the SELECT TOP clause with **WHERE clause** to filter data on specific conditions

Query

```
SELECT TOP 2*
FROM Employee
WHERE Salary > 2000
ORDER BY Salary;
```

Output



EmpId	EmpName	Email	Address	Age	Salary
5	Nishant Saluja	nishant@example.com	Spain	22	48000
1	Shubham	shubham@example.com	India	23	50000

The above query will select all the employees according to the given condition (i.e. all Employees except the employee whose salary is less than 2000 will be selected) then the result will be sorted by Salary in ascending order (The ORDER BY keyword sorts the records in ascending order by default). Finally, the first 2 rows would be returned by the above query.

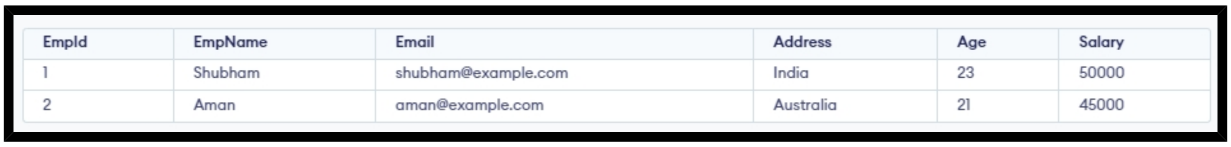
SQL SELECT TOP PERCENT Clause Example

The PERCENT keyword is utilized to select the primary and percent of all-out rows. For example:

Query

```
SELECT TOP 50 PERCENT *
FROM Employee;
```

Output



EmpId	EmpName	Email	Address	Age	Salary
1	Shubham	shubham@example.com	India	23	50000
2	Aman	aman@example.com	Australia	21	45000

Here, the above query will select the first 50% of employee records out of the total number of records(i.e., the first 3 rows will be returned).

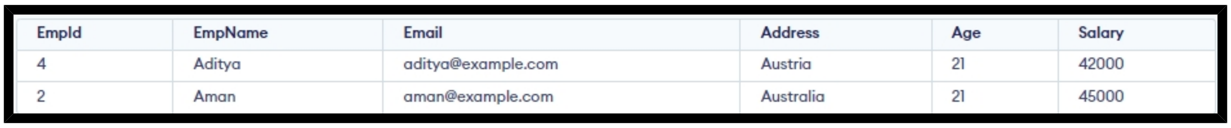
SQL TOP PERCENT with WHERE Clause Example

We can also include some situations using the TOP PERCENT with the WHERE clause in the above query.

Query

```
SELECT TOP 50 PERCENT *
FROM Employee
WHERE Salary < 50000;
```

Output



EmpId	EmpName	Email	Address	Age	Salary
4	Aditya	aditya@example.com	Austria	21	42000
2	Aman	aman@example.com	Australia	21	45000

The above query will select the Top 50% of the records out of the total number of records from the table according to the given condition such that it returns only the Top 50% of the records with the employee whose salary is less than 5000 (i.e, 2 rows will be returned)

SQL LIMIT Clause

SQL LIMIT Clause limits the number of results returned in the results set. The LIMIT Clause is utilized with the accompanying database systems:

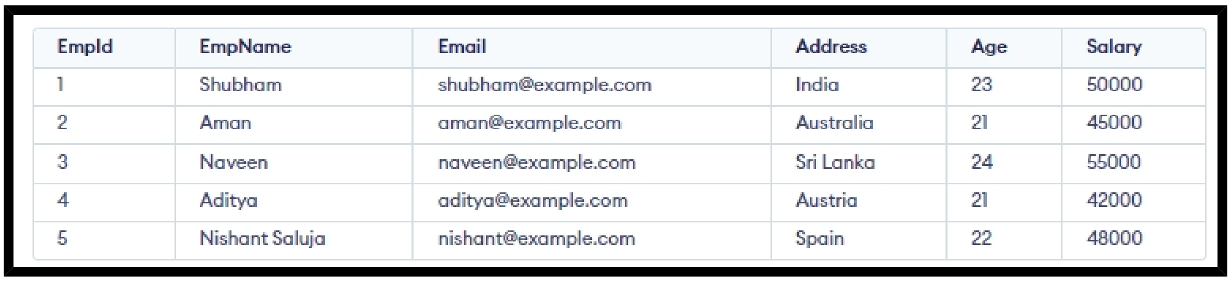
+ MySQL
+ PostgreSQL
+ SQLite

SQL LIMIT Clause Example

Since the LIMIT Clause is not supported in SQL Server we need to create a table in MySQL/PostgreSQL/SQLite. We will use the **LIMIT clause in MySQL**.

```
CREATE TABLE Employee (
   EmpId INTEGER PRIMARY KEY,
   EmpName VARCHAR(225) NOT NULL,
   Email VARCHAR(225) NOT NULL,
   Address VARCHAR(225) NOT NULL,
   Age INT NOT NULL,
   Salary MONEY NOT NULL
);
INSERT INTO Employee (EmpId, EmpName, Email, Address, Age,
Salary)
VALUES (1, 'Shubham', 'shubham@example.com', 'India', 23,
50000.00),
       (2, 'Aman', 'aman@example.com', 'Australia', 21,
45000.00),
       (3, 'Naveen', 'naveen@example.com', 'Sri Lanka', 24,
55000.00),
       (4, 'Aditya', 'aditya@example.com', 'Austria', 21,
42000.00),
       (5, 'Nishant Saluja', 'nishant@example.com',
'Spain', 22, 48000.00);
SELECT * FROM Employee ;
```

Output



EmpId	EmpName	Email	Address	Age	Salary
1	Shubham	shubham@example.com	India	23	50000
2	Aman	aman@example.com	Australia	21	45000
3	Naveen	naveen@example.com	Sri Lanka	24	55000
4	Aditya	aditya@example.com	Austria	21	42000
5	Nishant Saluja	nishant@example.com	Spain	22	48000

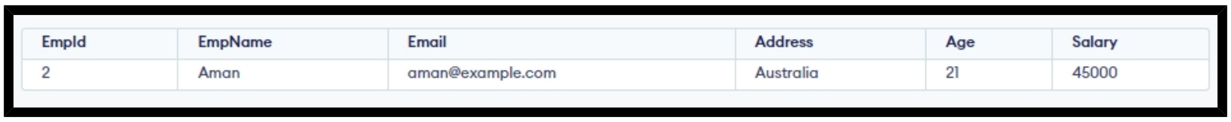
SELECT LIMIT Clause in SQL Example

In this example, we will use the SELECT LIMIT clause to display only 2 results.

Query

```
SELECT * FROM Employee
WHERE Salary = 45000
LIMIT 2;
```

Output



EmpId	EmpName	Email	Address	Age	Salary
2	Aman	aman@example.com	Australia	21	45000

From the above query, the LIMIT operator limits the number of records to be returned. Here, it returns the first 2 rows from the table.

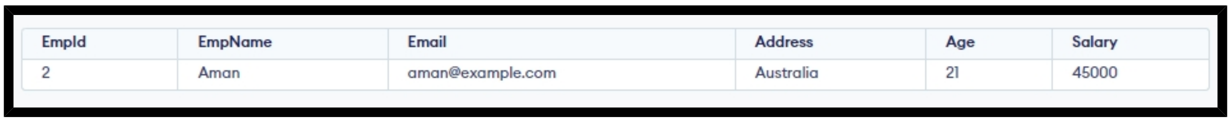
SQL LIMIT With WHERE Clause Example

The accompanying query selects the initial 4 records from the Employee table with a given condition.

Query

```
SELECT * FROM Employee
WHERE Salary = 45000
LIMIT 2;
```

Output



EmpId	EmpName	Email	Address	Age	Salary
2	Aman	aman@example.com	Australia	21	45000

The above query will select all the employees according to the imposed condition (i.e. it selects the limited 2 records from the table where salary is 2000). Finally, the first 2 rows would be returned by the above query.

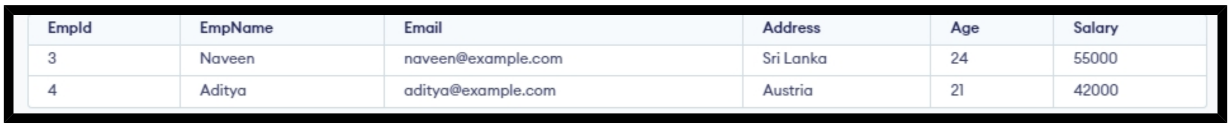
SQL LIMIT With OFFSET Clause Example

The OFFSET keyword is utilized to indicate beginning rows from where to select rows. For instance,

Query

```
SELECT * FROM Employee
LIMIT 2 OFFSET 2;
```

Output



EmpId	EmpName	Email	Address	Age	Salary
3	Naveen	naveen@example.com	Sri Lanka	24	55000
4	Aditya	aditya@example.com	Austria	21	42000

Here, the above query selects 2 rows from the beginning of the third row (i.e., OFFSET 2 means, the initial 2 rows are excluded or avoided).

SQL FETCH FIRST Clause

SQL FETCH FIRST clause fetches the first given number of rows from the table.

It is supported in database systems like:

+ IBM DB2
+ Oracle
+ PostgreSQL

Syntax

The syntax to use the FETCH FIRST clause in SQL is:

```
SELECT [columns] FROM [table] WHERE [condition]
FETCH FIRST [n] ROWS ONLY;
```

Here,

n: desired number of rows

SQL FETCH FIRST Clause Example

> We will use the same “Employee” table as used in previous examples.

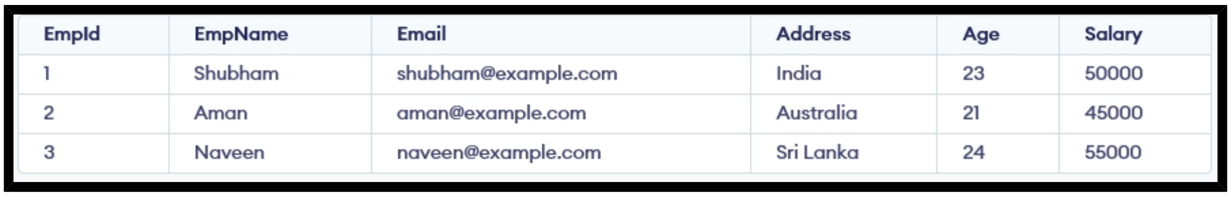
FETCH FIRST clause in SQL Example

In this example, we will fetch the first 3 rows from the table.

Query

```
SELECT * FROM Employee
FETCH FIRST 3 ROWS ONLY;
```

Output



EmpId	EmpName	Email	Address	Age	Salary
1	Shubham	shubham@example.com	India	23	50000
2	Aman	aman@example.com	Australia	21	45000
3	Naveen	naveen@example.com	Sri Lanka	24	55000

Here, the above query will fetch the first 3 rows only from the table. We can also include some situations using the FETCH FIRST PERCENT and WHERE Clause in the above query.

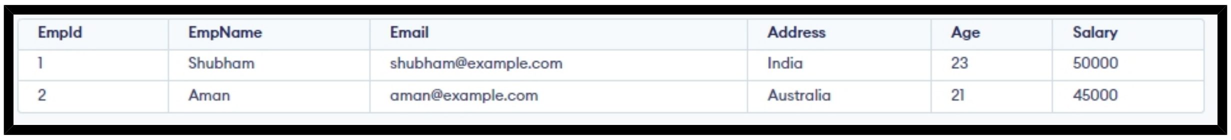
SQL FETCH FIRST PERCENT Example

In this example, we will fetch first 50% of the data from the table

Query

```
SELECT * FROM Employee
FETCH FIRST (SELECT CEIL(COUNT(*) / 2) FROM Employee) ROWS
ONLY;
```

Output



EmpId	EmpName	Email	Address	Age	Salary
1	Shubham	shubham@example.com	India	23	50000
2	Aman	aman@example.com	Australia	21	45000

Here, the above query fetches the first 50% of the total number of rows (i.e., 2 rows) from the table.

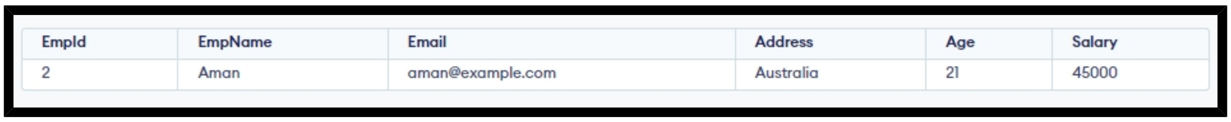
SQL FETCH FIRST with WHERE Clause Example

The “FETCH FIRST” syntax is not supported in MySQL. The correct syntax for limiting the number of rows in MySQL is by using the LIMIT clause.

Query

```sql
SELECT * FROM Employee
WHERE Salary = 45000
FETCH FIRST 1 ROWS ONLY;
```

Output



EmpId	EmpName	Email	Address	Age	Salary
2	Aman	aman@example.com	Australia	21	45000

Here, the above query fetches the first 1 row from the table, with the condition that the salary is 45000 (i.e., it returns 1 row only).

Summary

SQL TOP, **LIMIT** and **FETCH FIRST** clause are used for a same purpose of limiting the data returned in results set. All three of these queries are not supported by all SQL DBMS. Each of them is supported by only some of the DBMS and depending on the DBMS you use, the query can differ.

We have explained all **TOP**, **LIMIT** and **FETCH FIRST clause in SQL with examples**, and also mentioned their supported DBMS to avoid confusion. Users should check if the clause is supported by their DBMS before using them.