LIMIT and OFFSET to Cap Results

by Sophia

WHAT'S COVERED

In this lesson, you will use the LIMIT and OFFSET clauses to cap results, in two parts. Specifically, this lesson will cover:

1. LIMIT Clause

Most of the queries you have run so far have used wildcards to include all the available columns and rows in a table. That is fine for small databases like the one for this course, but many modern databases have billions of records in them. On databases of such volume, you need a way to limit your data request to just a small, relevant cross-section.

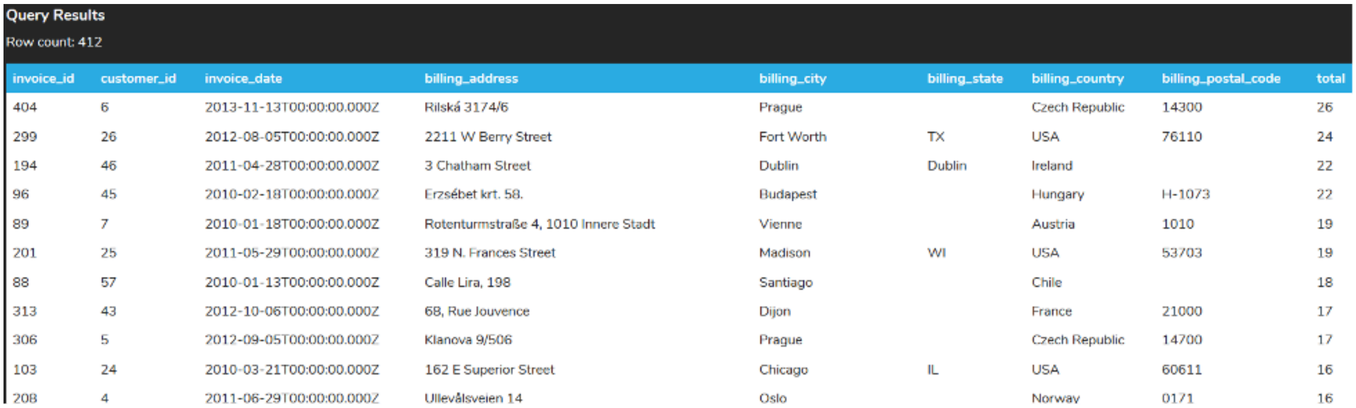
The **LIMIT clause** helps constrain the number of rows returned by a query. It is an optional clause added to the end of the SELECT statement. The format looks like this:

```
SELECT <columns>
FROM <tablename>
LIMIT <rowcount>;
```

For example, in the PostgreSQL database we've been working with throughout this course, let’s take a look at invoices by their total values in descending order:

```
SELECT *
FROM invoice
ORDER BY total DESC;
```

This query ends up returning all of the rows (412, in this case):



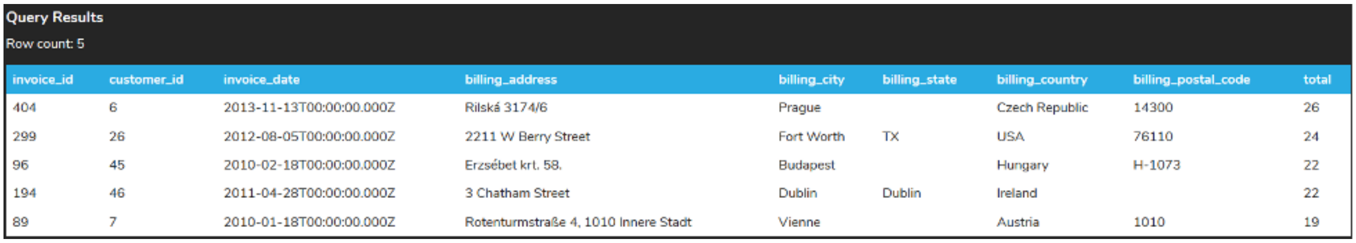
Query Results

Row count: 412

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
404	6	2013-11-13T00:00:00.000Z	Rilská 3174/6	Prague		Czech Republic	14300	26
299	26	2012-08-05T00:00:00.000Z	2211 W Berry Street	Fort Worth	TX	USA	76110	24
194	46	2011-04-28T00:00:00.000Z	3 Chatham Street	Dublin	Dublin	Ireland		22
96	45	2010-02-18T00:00:00.000Z	Erzsébet krt. 58.	Budapest		Hungary	H-1073	22
89	7	2010-01-18T00:00:00.000Z	Rotenturmstraße 4, 1010 Innere Stadt	Vienne		Austria	1010	19
201	25	2011-05-29T00:00:00.000Z	319 N. Frances Street	Madison	WI	USA	53703	19
88	57	2010-01-13T00:00:00.000Z	Calle Lira, 198	Santiago		Chile		18
313	43	2012-10-06T00:00:00.000Z	68, Rue Jouvence	Dijon		France	21000	17
306	5	2012-09-05T00:00:00.000Z	Klanova 9/506	Prague		Czech Republic	14700	17
103	24	2010-03-21T00:00:00.000Z	162 E Superior Street	Chicago	IL	USA	60611	16
208	4	2011-06-29T00:00:00.000Z	Ullevålsveien 14	Oslo		Norway	0171	16

If we were only interested in looking at the top five, we could add the LIMIT clause so that the query only returns the top five rows:

```
SELECT *
FROM invoice
ORDER BY total DESC
LIMIT 5;
```



Query Results

Row count: 5

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
404	6	2013-11-13T00:00:00.000Z	Rilská 3174/6	Prague		Czech Republic	14300	26
299	26	2012-08-05T00:00:00.000Z	2211 W Berry Street	Fort Worth	TX	USA	76110	24
96	45	2010-02-18T00:00:00.000Z	Erzsébet krt. 58.	Budapest		Hungary	H-1073	22
194	46	2011-04-28T00:00:00.000Z	3 Chatham Street	Dublin	Dublin	Ireland		22
89	7	2010-01-18T00:00:00.000Z	Rotenturmstraße 4, 1010 Innere Stadt	Vienne		Austria	1010	19

The five rows that this query returns are the same as the first five rows of the previous query with 412 results.

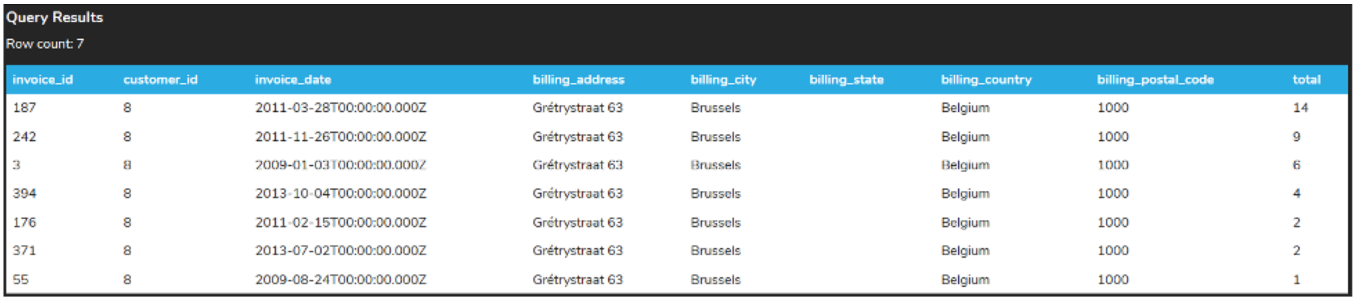
KEY CONCEPT

It is often more efficient to work with smaller result sets. Think about how web search engines return search results. The query you run might produce hundreds of thousands of results, but the search results don't show them all on the same page. That page would take a long time to load, and you would have to scroll thousands of times to see it all. So instead, a search engine shows only a dozen or so results on each page and prompts you to click a button to see the next page, and the next. Search engines use their own form of the LIMIT clause to display the first page of the results, and then the OFFSET clause (which you'll learn about later in this lesson) to display additional pages.

Let’s look at another example from our PostgreSQL database. Suppose you want to see the invoices that have the billing_country as Belgium in descending order:

```
SELECT *
FROM invoice
WHERE billing_country = 'Belgium'
ORDER BY total DESC;
```

This should return seven rows:



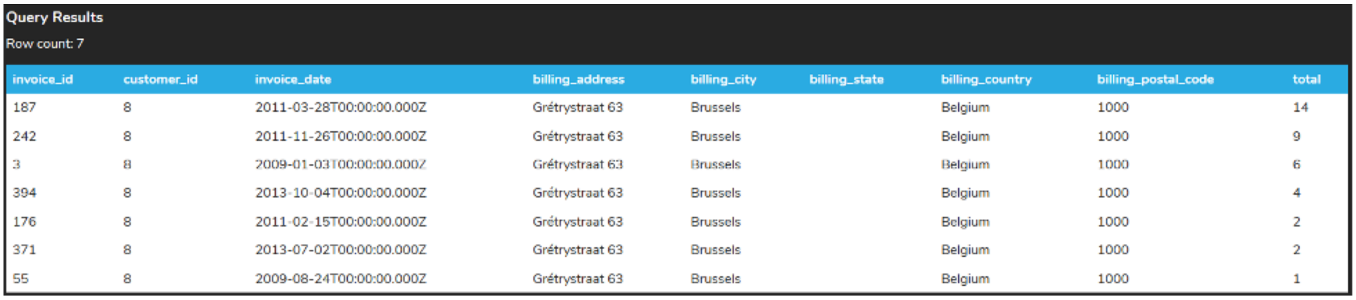
Query Results
Row count: 7

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
187	8	2011-03-28T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	14
242	8	2011-11-26T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	9
3	8	2009-01-03T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	6
394	8	2013-10-04T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	4
176	8	2011-02-15T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	2
371	8	2013-07-02T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	2
55	8	2009-08-24T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	1

If we had set the LIMIT to 10, it would return this:

```
SELECT *
FROM invoice
WHERE billing_country = 'Belgium'
ORDER BY total DESC
LIMIT 10;
```

The result set would stay the same, with seven rows, because there are not 10 rows of data that meet the criteria:



Query Results
Row count: 7

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
187	8	2011-03-28T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	14
242	8	2011-11-26T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	9
3	8	2009-01-03T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	6
394	8	2013-10-04T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	4
176	8	2011-02-15T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	2
371	8	2013-07-02T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	2
55	8	2009-08-24T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	1

TERM TO KNOW

LIMIT Clause

A clause that limits the number of rows returned by a query.

2. OFFSET Clause

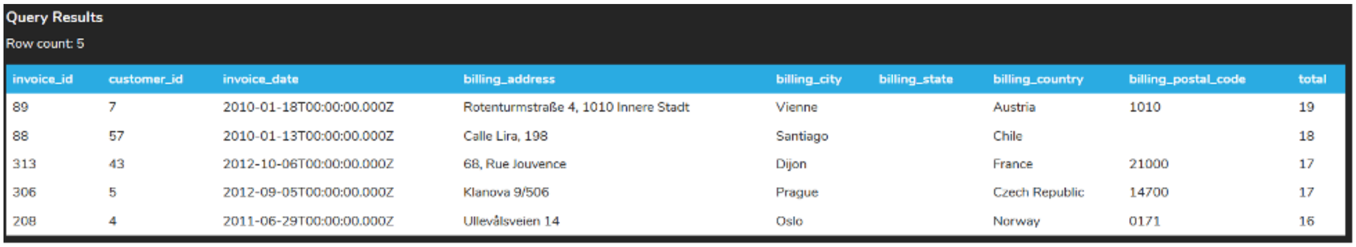
If we want to skip several rows when returning results, we can use the **OFFSET clause**. The OFFSET clause is placed after the LIMIT clause.

```
SELECT <columns>
FROM <tablename>
LIMIT <rowcount>
OFFSET <rowstoskip>;
```

By doing this, the statement will skip the first number of rows in the OFFSET and then return the number of rows based on the LIMIT. This is how a search engine delivering multiple pages of search results displays one page after another. Each time the user advances to the next page, the query is run again with a greater OFFSET value (or its equivalent in the database system that is being used).

Going back to our invoice example, we may want to look through each set of invoices five lines at a time. To look at the next five rows, we would run the query using the following:

```
SELECT *
FROM invoice
ORDER BY total DESC
LIMIT 5
OFFSET 5;
```



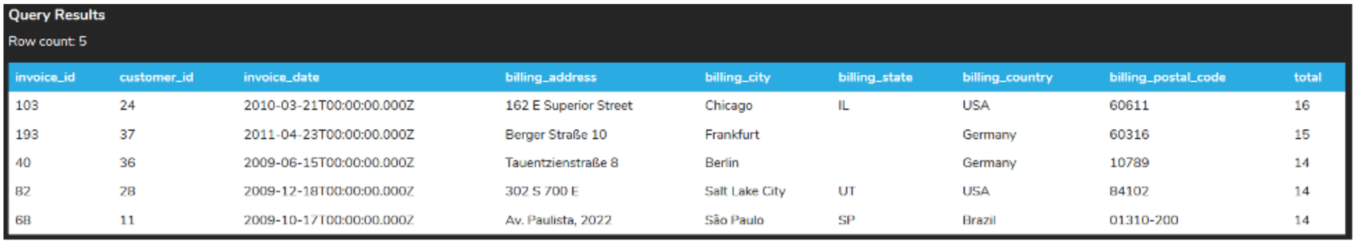
Query Results

Row count: 5

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
89	7	2010-01-18T00:00:00.000Z	Rotenturmstraße 4, 1010 Innere Stadt	Vienne		Austria	1010	19
88	57	2010-01-13T00:00:00.000Z	Calle Lira, 198	Santiago		Chile		18
313	43	2012-10-06T00:00:00.000Z	68, Rue Jouvence	Dijon		France	21000	17
306	5	2012-09-05T00:00:00.000Z	Klanova 9/506	Prague		Czech Republic	14700	17
208	4	2011-06-29T00:00:00.000Z	Ullevålsveien 14	Oslo		Norway	0171	16

If we wanted to look at the next five rows after that, we would run the query using the following:

```
SELECT *
FROM invoice
ORDER BY total DESC
LIMIT 5
OFFSET 10;
```



Query Results

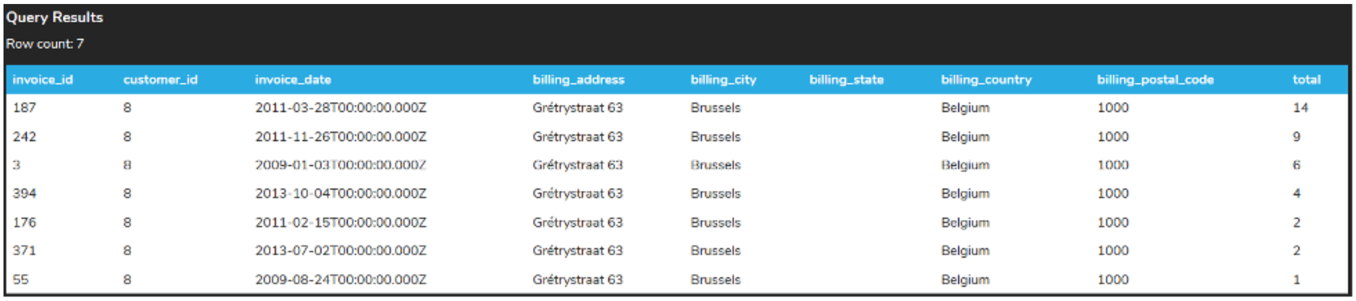
Row count: 5

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
103	24	2010-03-21T00:00:00.000Z	162 E Superior Street	Chicago	IL	USA	60611	16
193	37	2011-04-23T00:00:00.000Z	Berger Straße 10	Frankfurt		Germany	60316	15
40	36	2009-06-15T00:00:00.000Z	Tauentzienstraße 8	Berlin		Germany	10789	14
82	28	2009-12-18T00:00:00.000Z	302 S 700 E	Salt Lake City	UT	USA	84102	14
68	11	2009-10-17T00:00:00.000Z	Av. Paulista, 2022	São Paulo	SP	Brazil	01310-200	14

We can jump to any subset of rows being returned within the SELECT statement using the OFFSET and LIMIT clauses.

Let's look at our previous example of invoices in Belgium:

```
SELECT *
FROM invoice
WHERE billing_country = 'Belgium'
ORDER BY total DESC
LIMIT 10;
```



Query Results

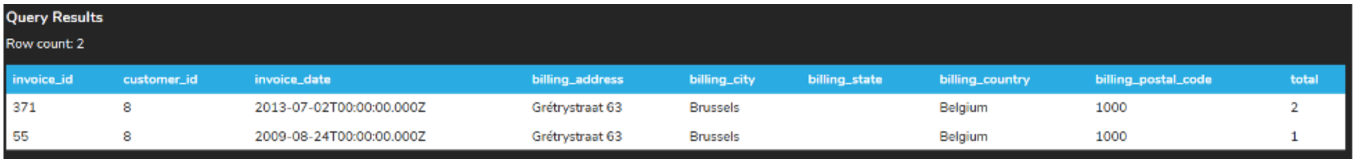
Row count: 7

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
187	8	2011-03-28T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	14
242	8	2011-11-26T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	9
3	8	2009-01-03T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	6
394	8	2013-10-04T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	4
176	8	2011-02-15T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	2
371	8	2013-07-02T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	2
55	8	2009-08-24T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	1

Recall that the result set returned seven rows. But look at what happens if we add the OFFSET to 5.

```
SELECT *
FROM invoice
WHERE billing_country = 'Belgium'
ORDER BY total DESC
LIMIT 10
OFFSET 5;
```

Again, this would skip the first five rows and return the number of rows remaining up to the LIMIT of 10:



Query Results

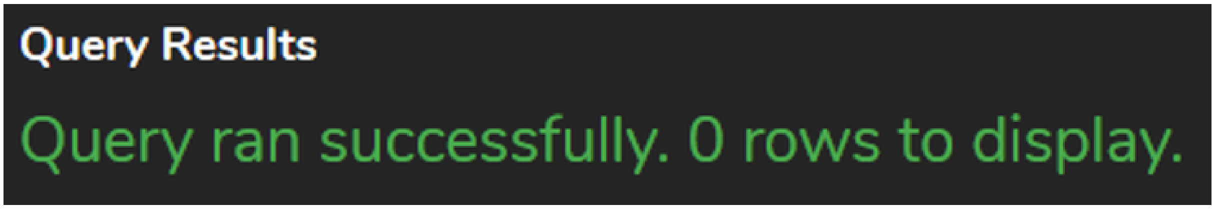
Row count: 2

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
371	8	2013-07-02T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	2
55	8	2009-08-24T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	1

If we had added an OFFSET of 10, we would exceed the number of rows available to return:

```
SELECT *
FROM invoice
WHERE billing_country = 'Belgium'
ORDER BY total DESC
LIMIT 10
OFFSET 10;
```

So, the query would simply return 0 rows:



Query Results

Query ran successfully. 0 rows to display.

WATCH

TRY IT

Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then, enter in one of the examples above and see how it works. Next, try your own choices for which columns you want the query to provide.

TERM TO KNOW

OFFSET Clause

This clause specifies a maximum number of rows to return from a query to restrict the number of rows returned.

SUMMARY

In this lesson, you learned that a query result in PostgreSQL can be limited or offset using the **LIMIT clauses** and **OFFSET clauses**. The LIMIT clause specifies a maximum number of rows to return from a query to restrict the number of rows returned. LIMIT lets you limit the results set to a specified number of rows after you specify a numeric value. You also learned that with the OFFSET clause, you are able to skip a specified number of rows from the beginning. Combining OFFSET and LIMIT allows you to retrieve different subsets of data and implement pagination.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND Faithe Wempen (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR **TERMS OF USE**.

TERMS TO KNOW

LIMIT Clause

A clause that limits the number of rows returned by a query.

OFFSET Clause

This clause specifies a maximum number of rows to return from a query to restrict the number of rows returned.