

# View all customer records (limit 10)

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

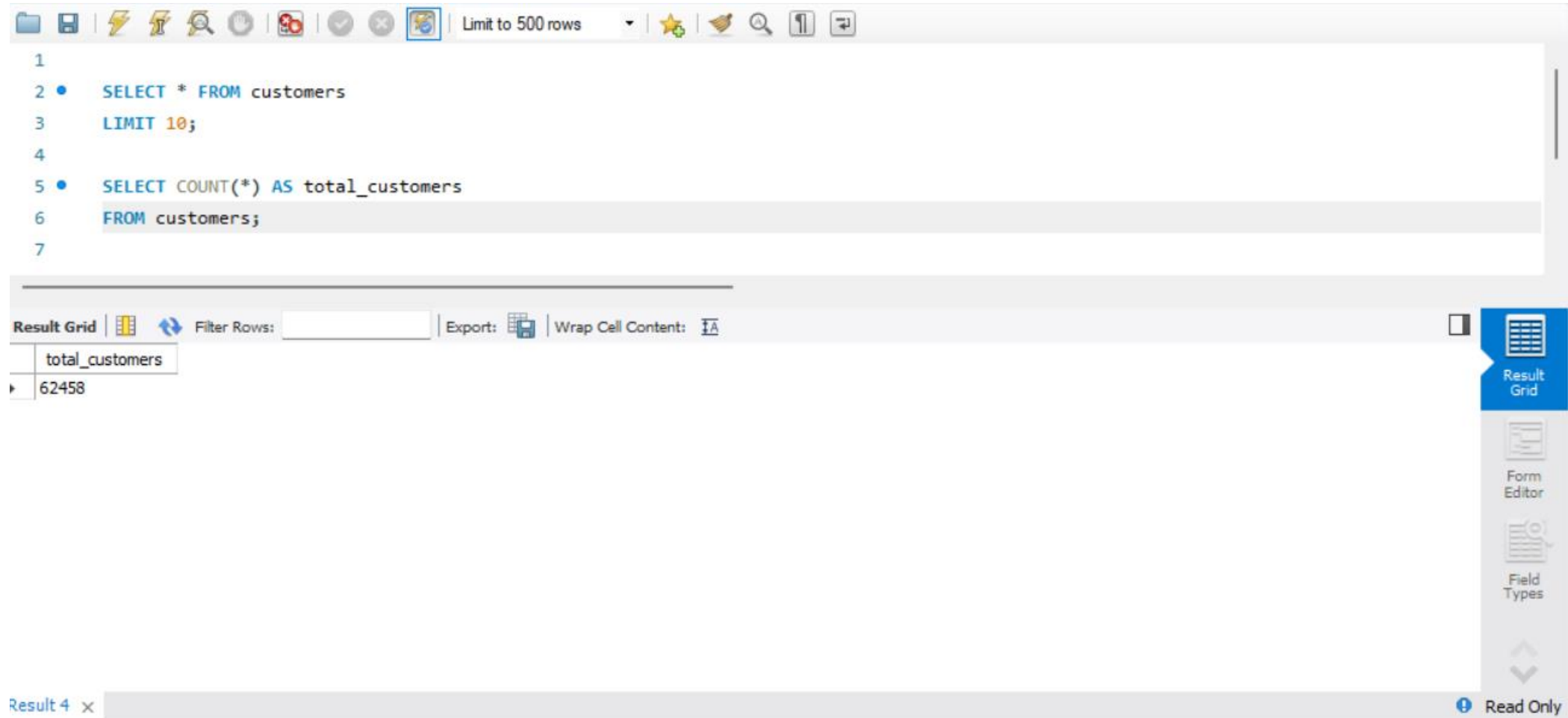
```
1
2 • SELECT * FROM customers
3   LIMIT 10;
4
5
```

The result grid displays 10 rows of customer data. The columns are: customer\_id, customer\_unique\_id, customer\_zip\_code\_prefix, customer\_city, and customer\_state. The data is as follows:

customer_id	customer_unique_id	customer_zip_code_prefix	customer_city	customer_state
06b8999e2fba1a1fbc88172c00ba8bc7	861eff4711a542e4b93843c6dd7febb0	14409	franca	SP
18955e83d337fd6b2def6b18a428ac77	290c77bc529b7ac935b93aa66c333dc3	9790	sao bernardo do campo	SP
4e7b3e00288586ebd08712fdd0374a03	060e732b5b29e8181a18229c7b0b2b5e	1151	sao paulo	SP
b2b6027bc5c5109e529d4dc6358b12c3	259dac757896d24d7702b9acbbff3f3c	8775	mogi das cruzeiros	SP
4f2d8ab171c80ec8364f7c12e35b23ad	345ecd01c38d18a9036ed96c73b8d066	13056	campinas	SP
879864dab9bc3047522c92c82e1212b8	4c93744516667ad3b8f1fb645a3116a4	89254	jaragua do sul	SC
fd826e7cf63160e536e0908c76c3f441	addec96d2e059c80c30fe6871d30d177	4534	sao paulo	SP
5e274e7a0c3809e14aba7ad5aae0d407	57b2a98a409812fe9618067b6b8ebe4f	35182	timoteo	MG
5adf08e34b2e993982a47070956c5c65	1175e95fb47ddf9de6b2b06188f7e0d	81560	curitiba	PR
4b7139f34592b3a31687243a302fa75b	9afe194fb833f79e300e37e580171f22	30575	belo horizonte	MG

The interface also includes a toolbar with various icons and a sidebar with buttons for 'Result Grid', 'Form Editor', and 'Field Types'. The status bar at the bottom indicates 'customers 1 x' and 'Read Only'.

# Total number of customers



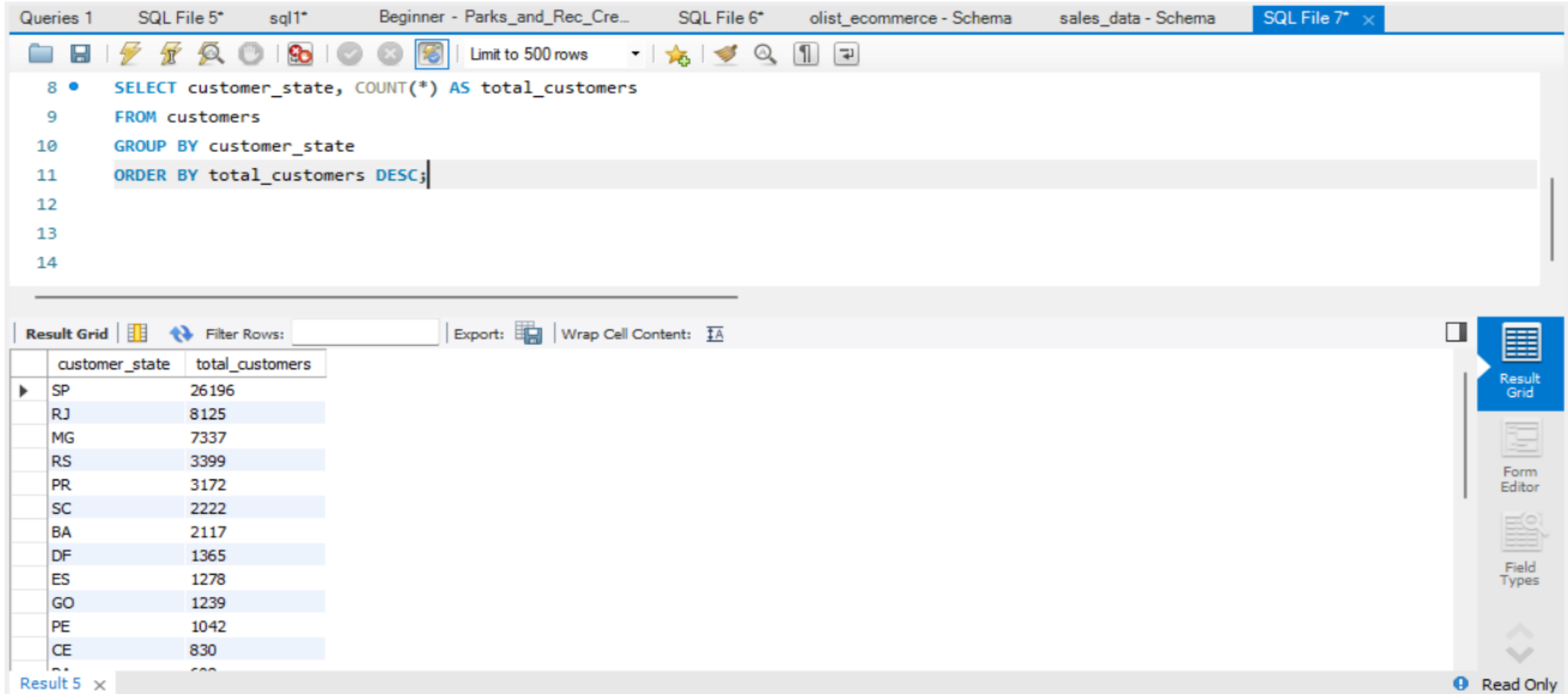
The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 500 rows' dropdown. The SQL editor contains two queries: a full table dump and a count query. The second query is selected and highlighted. Below the editor, the 'Result Grid' tab is active, displaying a single row with the value 62458 for the column 'total\_customers'. The right sidebar contains icons for 'Result Grid', 'Form Editor', and 'Field Types'. The bottom status bar indicates 'Result 4' and 'Read Only'.

```
1
2 • SELECT * FROM customers
3   LIMIT 10;
4
5 • SELECT COUNT(*) AS total_customers
6   FROM customers;
7
```

total_customers
62458

Result 4 x Read Only

# Customer count by state (GROUP BY)



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

```
8 • SELECT customer_state, COUNT(*) AS total_customers
9 FROM customers
10 GROUP BY customer_state
11 ORDER BY total_customers DESC;
12
13
14
```

The result grid displays the following data:

customer_state	total_customers
SP	26196
RJ	8125
MG	7337
RS	3399
PR	3172
SC	2222
BA	2117
DF	1365
ES	1278
GO	1239
PE	1042
CE	830

The interface includes a toolbar with various icons, a "Limit to 500 rows" dropdown, and a "Result Grid" button. The bottom status bar indicates "Result 5" and "Read Only".

# Top 5 cities with most customers

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

```
16
17
18 • SELECT customer_city, COUNT(*) AS city_customers
19 FROM customers
20 GROUP BY customer_city
21 ORDER BY city_customers DESC
22 LIMIT 5;
```

The result grid displays the following data:

customer_city	city_customers
sao paulo	9775
rio de janeiro	4344
belo horizonte	1751
brasil	1358
curitiba	954

The IDE interface includes a top toolbar with icons for file operations, a toolbar with a 'Limit to 500 rows' dropdown, and a bottom toolbar with options for 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. The bottom status bar shows 'Result 6' and 'Read Only'.

# Unique vs. duplicate customers

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

```
27  
28  
29 • SELECT COUNT(DISTINCT customer_unique_id) AS unique_customers,  
30        COUNT(*) AS total_rows,  
31        COUNT(*) - COUNT(DISTINCT customer_unique_id) AS duplicate_entries  
32 FROM customers;  
33
```

The result grid displays the following data:

	unique_customers	total_rows	duplicate_entries
▶	61105	62458	1353

The interface includes a toolbar with various icons, a "Limit to 500 rows" dropdown, and a sidebar with options like "Result Grid", "Form Editor", and "Field Types". The bottom status bar shows "Result 7" and "Read Only".

# Subquery: Cities with above average customer count

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains a SQL query that selects cities with a customer count greater than the average. The result grid displays the output of the query, showing 15 cities and their corresponding customer counts.

```
39 • SELECT customer_city, COUNT(*) AS city_customers
40 FROM customers
41 GROUP BY customer_city
42 HAVING city_customers > (
43     SELECT AVG(city_count) FROM (
44         SELECT COUNT(*) AS city_count
45         FROM customers
46         GROUP BY customer_city
47     ) AS city_stats
48 );
49
```

Result Grid

customer_city	city_customers
franca	102
sao bernardo do campo	600
sao paulo	9775
mogi das cruzeiras	228
campinas	913
jaragua do sul	57
timoteo	33
curitiba	954
belo horizonte	1751
montes claros	140
rio de janeiro	4344
lencois paulista	27
caxias do sul	143
piracicaba	219
guarulhos	717

Result 8 x

Read Only

# Zip Code Prefix Summary (Top 10)

The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 500 rows". Below the toolbar, a SQL query is entered in a text area:

```
52 • SELECT customer_zip_code_prefix, COUNT(*) AS zip_count
53 FROM customers
54 GROUP BY customer_zip_code_prefix
55 ORDER BY zip_count DESC
56 LIMIT 10;
57
58
59
60
61
62
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, a "Wrap Cell Content:" checkbox, and a "Fetch rows:" button. The result grid displays the following data:

customer_zip_code_prefix	zip_count
22790	89
22775	82
24230	80
22793	77
24220	74
35162	69
13212	62
35500	62
22631	59
29101	59

On the right side of the interface, there is a vertical toolbar with icons for "Result Grid", "Form Editor", "Field Types", and "Query Stats". At the bottom left, it says "Result 9 x". At the bottom right, there is a "Read Only" status indicator.

# Creating a View

The screenshot shows a SQL IDE interface with a query editor and an output window. The query editor contains the following SQL code:

```
62  
63  
64  
65  
66  
67  
68 • CREATE OR REPLACE VIEW view_customer_by_state AS  
69 SELECT customer_state, COUNT(*) AS total_customers  
70 FROM customers  
71 GROUP BY customer_state;  
72  
73  
74  
75  
76  
77  
78
```

The output window displays the results of the SQL execution, including the creation of the view. The output is as follows:

#	Time	Action	Message	Duration / Fetch
✓ 24	21:27:43	SELECT customer_state, COUNT(*) AS total_customers FROM customers GROUP BY customer_state ORDE...	27 row(s) returned	0.109 sec / 0.000 sec
✓ 25	21:28:57	SELECT customer_city, COUNT(*) AS city_customers FROM customers GROUP BY customer_city ORDER B...	5 row(s) returned	0.156 sec / 0.000 sec
✓ 26	21:29:31	SELECT COUNT(DISTINCT customer_unique_id) AS unique_customers, COUNT(*) AS total_rows, C...	1 row(s) returned	0.375 sec / 0.000 sec
✓ 27	21:30:09	SELECT customer_city, COUNT(*) AS city_customers FROM customers GROUP BY customer_city HAVING ci...	432 row(s) returned	0.297 sec / 0.000 sec
✓ 28	21:31:27	SELECT customer_zip_code_prefix, COUNT(*) AS zip_count FROM customers GROUP BY customer_zip_cod...	10 row(s) returned	0.125 sec / 0.000 sec
✓ 29	21:34:22	CREATE VIEW view_customer_by_state AS SELECT customer_state, COUNT(*) AS total_customers FROM c...	0 row(s) affected	0.062 sec
✗ 30	21:34:34	CREATE VIEW view_customer_by_state AS SELECT customer_state, COUNT(*) AS total_customers FROM c...	Error Code: 1050. Table 'view_customer_by_state' already exists	0.000 sec
✓ 31	21:36:34	CREATE OR REPLACE VIEW view_customer_by_state AS SELECT customer_state, COUNT(*) AS total_cust...	0 row(s) affected	0.063 sec



# Optimizing indexes

The screenshot shows a database IDE interface. The main editor displays a list of line numbers from 74 to 90. Line 86 contains the SQL command: `CREATE INDEX idx_customer_state ON customers(customer_state);`. The right sidebar contains a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." Below this is a "Context Help" tab and a "Snippets" tab. The bottom panel shows the "Output" window with a table of execution results.

#	Time	Action	Message	Duration / Fetch
25	21:28:57	SELECT customer_city, COUNT(*) AS city_customers FROM customers GROUP BY customer_city ORDER B...	5 row(s) returned	0.156 sec / 0.000 sec
26	21:29:31	SELECT COUNT(DISTINCT customer_unique_id) AS unique_customers, COUNT(*) AS total_rows, C...	1 row(s) returned	0.375 sec / 0.000 sec
27	21:30:09	SELECT customer_city, COUNT(*) AS city_customers FROM customers GROUP BY customer_city HAVING ci...	432 row(s) returned	0.297 sec / 0.000 sec
28	21:31:27	SELECT customer_zip_code_prefix, COUNT(*) AS zip_count FROM customers GROUP BY customer_zip_cod...	10 row(s) returned	0.125 sec / 0.000 sec
29	21:34:22	CREATE VIEW view_customer_by_state AS SELECT customer_state, COUNT(*) AS total_customers FROM c...	0 row(s) affected	0.062 sec
30	21:34:34	CREATE VIEW view_customer_by_state AS SELECT customer_state, COUNT(*) AS total_customers FROM c...	Error Code: 1050. Table 'view_customer_by_state' already exists	0.000 sec
31	21:36:34	CREATE OR REPLACE VIEW view_customer_by_state AS SELECT customer_state, COUNT(*) AS total_cust...	0 row(s) affected	0.063 sec
32	21:37:22	CREATE INDEX idx_customer_state ON customers(customer_state)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.453 sec

# Inner Join

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

```
76 • SELECT
77     c.customer_id,
78     c.customer_state,
79     v.total_customers
80 FROM customers c
81 INNER JOIN view_customer_by_state v
82     ON c.customer_state = v.customer_state
83 LIMIT 10;
84
85
86
87
```

The result grid displays the following data:

customer_id	customer_state	total_customers
06b8999e2fba1a1fbc88172c00ba8bc7	SP	26196
18955e83d337fd6b2def6b18a428ac77	SP	26196
4e7b3e00288586ebd08712fdd0374a03	SP	26196
b2b6027bc5c5109e529d4dc6358b12c3	SP	26196
4f2d8ab171c80ec8364f7c12e35b23ad	SP	26196
879864dab9bc3047522c92c82e1212b8	SC	2222
fd826e7cf63160e536e0908c76c3f441	SP	26196
5e274e7a0c3809e14aba7ad5aae0d407	MG	7337
5adf08e34b2e993982a47070956c5c65	PR	3172
4b7139f34592b3a31687243a302fa75b	MG	7337

The interface includes a toolbar with icons for file operations, a 'Limit to 500 rows' dropdown, and a 'Result Grid' button. The result grid also has a 'Filter Rows' input field, an 'Export' button, a 'Wrap Cell Content' checkbox, and a 'Fetch rows' button. A sidebar on the right contains buttons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'.

# Left Join

The screenshot shows a SQL IDE interface with a query editor and a results grid. The query editor contains the following SQL code:

```
95
96
97 • SELECT
98     v.customer_state,
99     v.total_customers,
100     c.customer_id
101 FROM view_customer_by_state v
102 LEFT JOIN customers c
103     ON v.customer_state = c.customer_state
104 LIMIT 10;
105
106
```

The results grid displays the following data:

	customer_state	total_customers	customer_id
▶	AC	56	94906d652f7d1c9e4c4af604ede7784b
	AC	56	ca077dd95290191565abb393a580bbfc
	AC	56	5a5f70fead43dc16ec4d5afed5281239
	AC	56	c5de7db594b246f428940ced283125bd
	AC	56	16aba5570b45f6b4c9204407132e2cae
	AC	56	6b04ea93ce93cef2a5fbdfe302e21331
	AC	56	2201362e68992f654942dc0067c1b716
	AC	56	31dbc13addc753e210692eacaea065e4
	AC	56	5a0dc6805a195218a4e448653dee4740
	AC	56	5c495510e9f861b83aa5fb7c308e8d76

The interface includes a toolbar with various icons, a "Limit to 500 rows" dropdown, and a "Result Grid" button on the right. The bottom status bar shows "Result 13" and "Read Only".