

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. The query editor contains the following SQL code:

```
1 CREATE VIEW active_users AS
2 SELECT id, name, email FROM users WHERE status = 'active';
3
4
5
```

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 are tabs for "Context Help" and "Snippets".

The bottom "Output" window shows the "Action Output" tab with a table of execution results:

#	Time	Action	Message	Duration / Fetch
48	20:27:32	SELECT u.name, o.order_number FROM users u LEFT JOIN orders o ON u.id = o.user_id LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
49	20:27:32	SELECT o.order_number, u.name FROM orders o RIGHT JOIN users u ON o.user_id = u.id LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
50	20:28:14	SELECT title, price FROM products WHERE price > (SELECT AVG(price) FROM products) LIMIT 0, 1000	0 row(s) returned	0.015 sec / 0.000 sec
51	20:28:51	SELECT SUM(total_amount) AS total_revenue FROM orders LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
52	20:28:51	SELECT AVG(price) AS average_price FROM products LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
53	20:29:13	CREATE VIEW active_users AS SELECT id, name, email FROM users WHERE status = 'active'	0 row(s) affected	0.016 sec

Query 1



Limit to 1000 rows

```
1  -- Total revenue from all orders
2  • SELECT SUM(total_amount) AS total_revenue FROM orders;
3
4  -- Average product price
5  • SELECT AVG(price) AS average_price FROM products;
6
7
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content: ☐

average_price

NULL



```
1
2  -- INNER JOIN: Orders with User Details
3  • SELECT o.order_number, u.name, o.total_amount
4  FROM orders o
5  INNER JOIN users u ON o.user_id = u.id;
6
7  -- LEFT JOIN: All users and any orders they placed
8  • SELECT u.name, o.order_number
9  FROM users u
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	order_number	name
▶	NULL	John Doe
	NULL	Jane Doe

Query 1 x



Limit to 1000 rows

```
1  -- Subquery: Products that cost more than average price
2  •  SELECT title, price FROM products
3     WHERE price > (SELECT AVG(price) FROM products);
4
5
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	title	price
--	-------	-------

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	id	order_number	user_id	sub_total	shipping_id	coupon	total_amount	quantity	payment_method	payment_status	status	first_name	last_name	email	phon
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

Field Types

Output					
Action Output					
#	Time	Action	Message	Duration / Fetch	
✓ 16	20:19:45	CREATE TABLE IF NOT EXISTS coupons (id INT AUTO_INCREMENT PRIMARY KEY, code VARCHA...	0 row(s) affected	0.031 sec	
✓ 17	20:19:45	CREATE TABLE IF NOT EXISTS wishlists (id INT AUTO_INCREMENT PRIMARY KEY, product_id INT, ...	0 row(s) affected	0.032 sec	
✓ 18	20:19:45	CREATE TABLE IF NOT EXISTS product_reviews (id INT AUTO_INCREMENT PRIMARY KEY, user_id I...	0 row(s) affected	0.031 sec	
✓ 19	20:19:45	CREATE TABLE IF NOT EXISTS post_comments (id INT AUTO_INCREMENT PRIMARY KEY, user_id I...	0 row(s) affected	0.047 sec	
✓ 20	20:19:45	CREATE TABLE IF NOT EXISTS settings (id INT AUTO_INCREMENT PRIMARY KEY, description TEXT...	0 row(s) affected	0.032 sec	
✓ 21	20:19:45	CREATE TABLE IF NOT EXISTS jobs (id INT AUTO_INCREMENT PRIMARY KEY, queue VARCHAR(10...	0 row(s) affected	0.031 sec	

Query 1

Limit to 1000 rows

1

•

CREATE INDEX idx_user_id ON orders(user_id);

2

•

CREATE INDEX idx_product_id ON carts(product_id);

3

4

5

6

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 50	20:28:14	SELECT title, price FROM products WHERE price > (SELECT AVG(price) FROM products) LIMIT 0, 1000	0 row(s) returned	0.015 sec / 0.000 sec
✓ 51	20:28:51	SELECT SUM(total_amount) AS total_revenue FROM orders LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
✓ 52	20:28:51	SELECT AVG(price) AS average_price FROM products LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
✓ 53	20:29:13	CREATE VIEW active_users AS SELECT id, name, email FROM users WHERE status = 'active'	0 row(s) affected	0.016 sec
✓ 54	20:29:50	CREATE INDEX idx_user_id ON orders(user_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec
✓ 55	20:29:50	CREATE INDEX idx_product_id ON carts(product_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec