



# Introduction to MySQL Shell for VS Code

February 2, 2023  
preFOSDEM MySQL Days 2023  
Brussels, Belgium

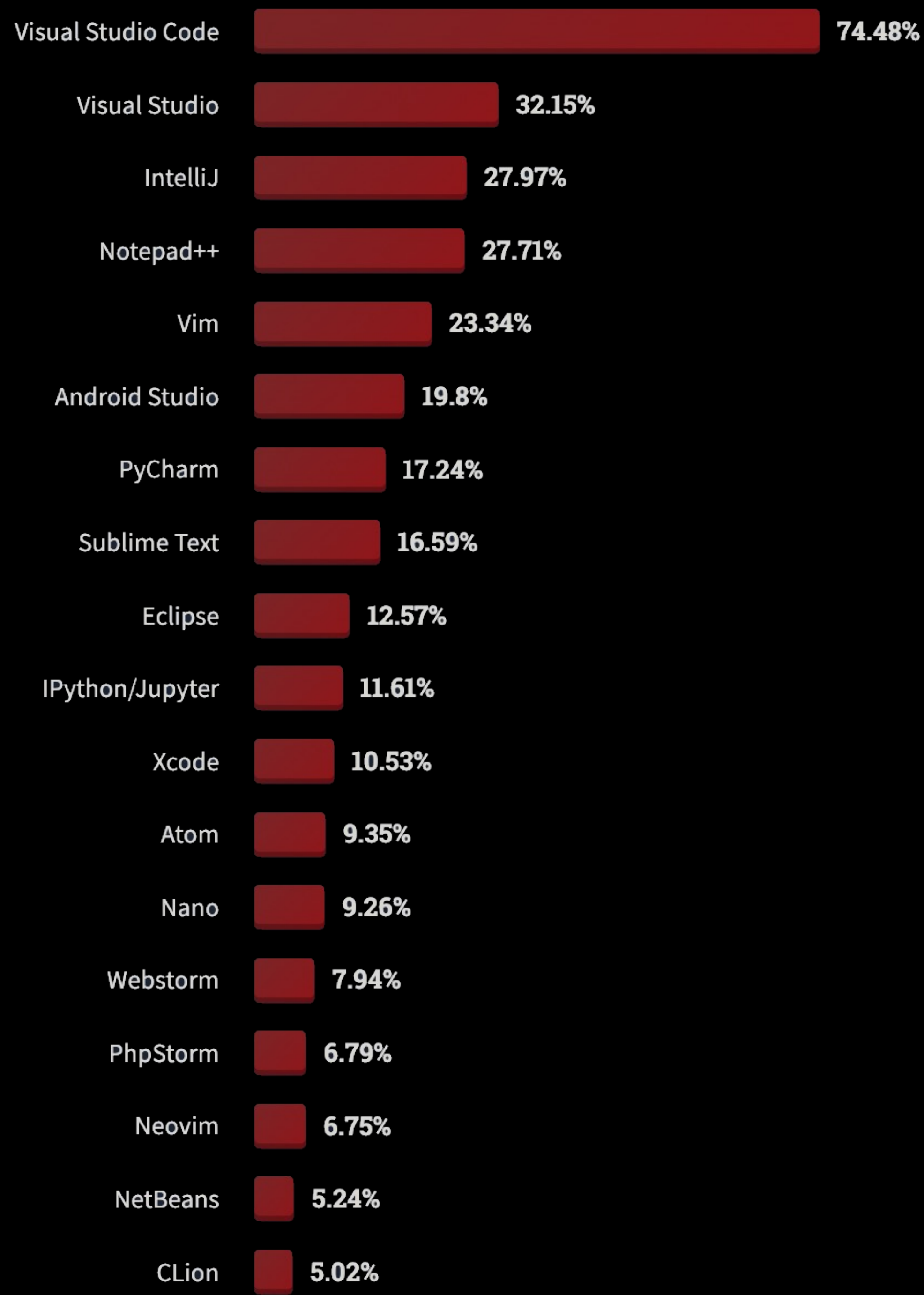




## Kenny Gryp

MySQL Product Manager  
Oracle





“In the Stack Overflow 2022 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 74% of 71,000 respondents reporting that they use it.”



MYSQL SHELL FOR VSCODE

- ▼ DATABASE CONNECTIONS
  - ▼ Localhost 8.0
    - MySQL Administration
    - information\_schema
    - performance\_schema
    - sakila
      - ▼ Tables
        - actor
        - address
        - category
        - city
        - country
        - customer
        - film
        - film\_actor
        - film\_category
        - film\_text
        - inventory
        - language
        - payment
        - rental
        - staff

▼ ORACLE CLOUD INFRASTRUCTURE

> MYSQL SHELL CONSOLES

▼ MYSQL SHELL TASKS

Localhost 8.0 X

Editor: DB Notebook

```

performance_schema
sakila
sys

```

OK, 5 records retrieved in 2.549ms

```

sql> SELECT * FROM `sakila`.`actor`
      WHERE actor_id |

```

actor_id ▲	first_name ▲	last_name ▲	last_update ▲
1	PENELOPE	GUINESSS	2021-09-28 22:18:53
2	NICK	WAHLBERG	2006-02-15 04:34:33
3	ED	CHASE	2006-02-15 04:34:33
4	JENNIFER	DAVIS	2006-02-15 04:34:33
5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
6	BETTE	NICHOLSON	2006-02-15 04:34:33
7	GRACE	MOSTEL	2006-02-15 04:34:33
8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
9	JOE	SWANK	2006-02-15 04:34:33
10	CHRISTIAN	GABLE	2006-02-15 04:34:33
11	7500	CAGE	2006-02-15 04:34:33

OK, 200 records retrieved in 5.924ms

```

sql> SELECT * FROM `sakila`.`address`

```

address_id ▲	address ▲	address2 ▲	district ▲	city_id ▲	postal_code ▲	p
1	47 MySakila Drive	NULL	Alberta	300		
2	28 MySQL Boulevard	NULL	QLD	576		
3	23 Workhaven Lane	NULL	Alberta	300		1
4	1411 Lillydale Drive	NULL	QLD	576		6
5	1913 Hanoi Way		Nagasaki	463	35200	2
6	1121 Loja Avenue		California	449	17886	8
7	692 Joliet Street		Attika	38	83579	4
8	1566 Inegl Manor		Mandalay	349	53561	7
9	53 Idfu Parkway		Nantou	361	42399	1
10	1795 Santiago de Compostela Way		Texas	295	18743	8

OK, 603 records retrieved in 1.533ms



# Extension Installation

Install  
MySQL Shell for VSCode  
directly in Marketplace

A screenshot of the VS Code Marketplace search results for 'mysql shell'. The search bar at the top contains the text 'mysql shell'. Below the search bar, a list of extensions is displayed. The extension 'MySQL Shell for VS Code' by Oracle Corporation is highlighted with a red box. It shows a rating of 3.5 stars and 10K downloads. Other extensions listed include 'MySQL management tool', 'Database manager for MySQL', 'Shell', 'shell-format', 'MySQL Syntax', 'SQLTools MySQL/MariaDB', 'ShellHauncher', 'mysql-inline-d...', 'Shell Syntax', 'AutoComplete s...', and 'ShellCheck'.



## MySQL Shell for VS Code v1.4.1

Oracle Corporation | 10,182 | ★★★★★ (10)

The power of MySQL Shell as part of your VS Code workflow.

Install

[Details](#) [Feature Contributions](#) [Changelog](#)

## MySQL Shell for VS Code 1.4.1+8.0.30 Preview

This extension enables interactive editing and execution of SQL for MySQL Databases and the MySQL Database Service. It integrates the MySQL Shell directly into VS Code development workflows.

**IMPORTANT: Please note that this is a PREVIEW release which is not meant to be used in production.**

A screenshot of the MySQL Shell for VS Code interface. The interface shows a SQL Notebook with a query executed: `sql> SELECT * FROM 'sakila'.actor`. The results are displayed in a table with columns: actor\_id, first\_name, last\_name, and last\_update. The table contains 10 rows of data. Below the table, the status bar indicates 'OK, 200 records retrieved in 2.379ms'. The interface also shows a file explorer on the left with a tree view of the database schema.

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINNESS	2021-09-28 22:18:53
2	NICK	WAHLBERG	2006-02-15 04:34:33
3	ED	CHASE	2006-02-15 04:34:33
4	JENNIFER	DAVIS	2006-02-15 04:34:33
5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
6	BETTE	NICHOLSON	2006-02-15 04:34:33
7	GRACE	MOSTEL	2006-02-15 04:34:33
8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
9	JOE	SWANK	2006-02-15 04:34:33
10	CHRISTIAN	GABLE	2006-02-15 04:34:33





# Adding DB Connections

Database Editor  
Connections View

Create Remote  
MySQL Connections

The screenshot displays the 'DB Connections' view in the Database Editor. The left sidebar shows a tree view under 'DATABASE CONNECTIONS' with 'test' selected. The main area shows a 'Database Connection Configuration' dialog box. The dialog has the following fields and options:

- Database Type:** MySQL (dropdown)
- Caption:** Localhost 8.0
- Description:** A new Database Connection
- Connection Details:**
  - Basic:** Host Name or IP Address: localhost; Protocol: mysql; Port: 3306
  - User Name:** root
  - Default Schema:** (empty)
  - Connect using SSH Tunnel
  - Connect to OCI DB server using MDS settings

Buttons for 'Store Password', 'Clear Password', 'Cancel', and 'OK' are also visible.





# SQL Notebook Interface

Write, Execute, Edit, Execute  
within Notebook

The screenshot shows the SQL Notebook interface with two queries and their results. The first query is a SELECT statement on the 'actor' table, and the second is a SELECT statement on the 'address' table. The interface includes a toolbar with various icons and a status bar at the bottom.

```
sql> SELECT * FROM `sakila`.`actor`  
WHERE actor_id > 35
```

actor_id	first_name	last_name	last_update
36	BURT	DUKAKIS	2006-02-15 04:34:33
37	VAL	BOLGER	2006-02-15 04:34:33
38	TOM	MCKELLEN	2006-02-15 04:34:33
39	GOLDIE	BRODY	2006-02-15 04:34:33
40	JOHNNY	CAGE	2006-02-15 04:34:33
41	JODIE	DEGENERES	2006-02-15 04:34:33
42	TOM	MIRANDA	2006-02-15 04:34:33
43	KIRK	JOVOVICH	2006-02-15 04:34:33
44	NICK	STALLONE	2006-02-15 04:34:33
45	REESE	KILMER	2006-02-15 04:34:33
46	PARKER	COLBERT	2006-02-15 04:34:33

```
sql> SELECT * FROM `sakila`.`address`
```

address_id	address	address2	district	city_id	postal_code	p
1	47 MySakila Drive	NULL	Alberta	300		
2	28 MySQL Boulevard	NULL	QLD	576		
3	23 Workhaven Lane	NULL	Alberta	300		1
4	1411 Lillydale Drive	NULL	QLD	576		6
5	1913 Hanoi Way		Nagasaki	463	35200	2
6	1121 Loja Avenue		California	449	17886	8
7	692 Joliet Street		Attika	38	83579	4
8	1566 Inegl Manor		Mandalay	349	53561	7
9	53 Idfu Parkway		Nantou	361	42399	1
10	1795 Santiago de Compostela Way		Texas	295	18743	8

```
sql> \ts  
Switched to TypeScript mode  
runSql("SELECT * FROM sakila.actor", (res) => {})
```

TypeScript  
(JS & py Support)



# OCI MySQL HeatWave

Connect Directly to DBSystem via OCI Bastion Service

Access OCI MySQL Infrastructure

The screenshot displays the VS Code DB Connections interface. On the left, a tree view shows the project structure under 'ORACLE CLOUD INFRASTRUCTURE', with 'MDS-Development (Default)' expanded to show 'MySQL Shell HW Dev', 'MDS DB System', 'REST Endpoint AD1', 'REST Endpoint AD2', and 'Bastion2208171158'. A red box highlights this tree view. The main editor area shows the 'Database Connection Configuration' dialog box, also highlighted with a red border. The dialog includes the following fields and options:

- Database Type:** MySQL (selected)
- Caption:** MDS DB System
- Description:** DB System to test MRS.
- Connection Details:**
  - Host Name or IP Address:** 10.0.1.35
  - Protocol:** mysql
  - Port:** 3306
  - User Name:** dba
  - Default Schema:** (empty)
  - Connect using SSH Tunnel
  - Connect to OCI DB server using MDS settings

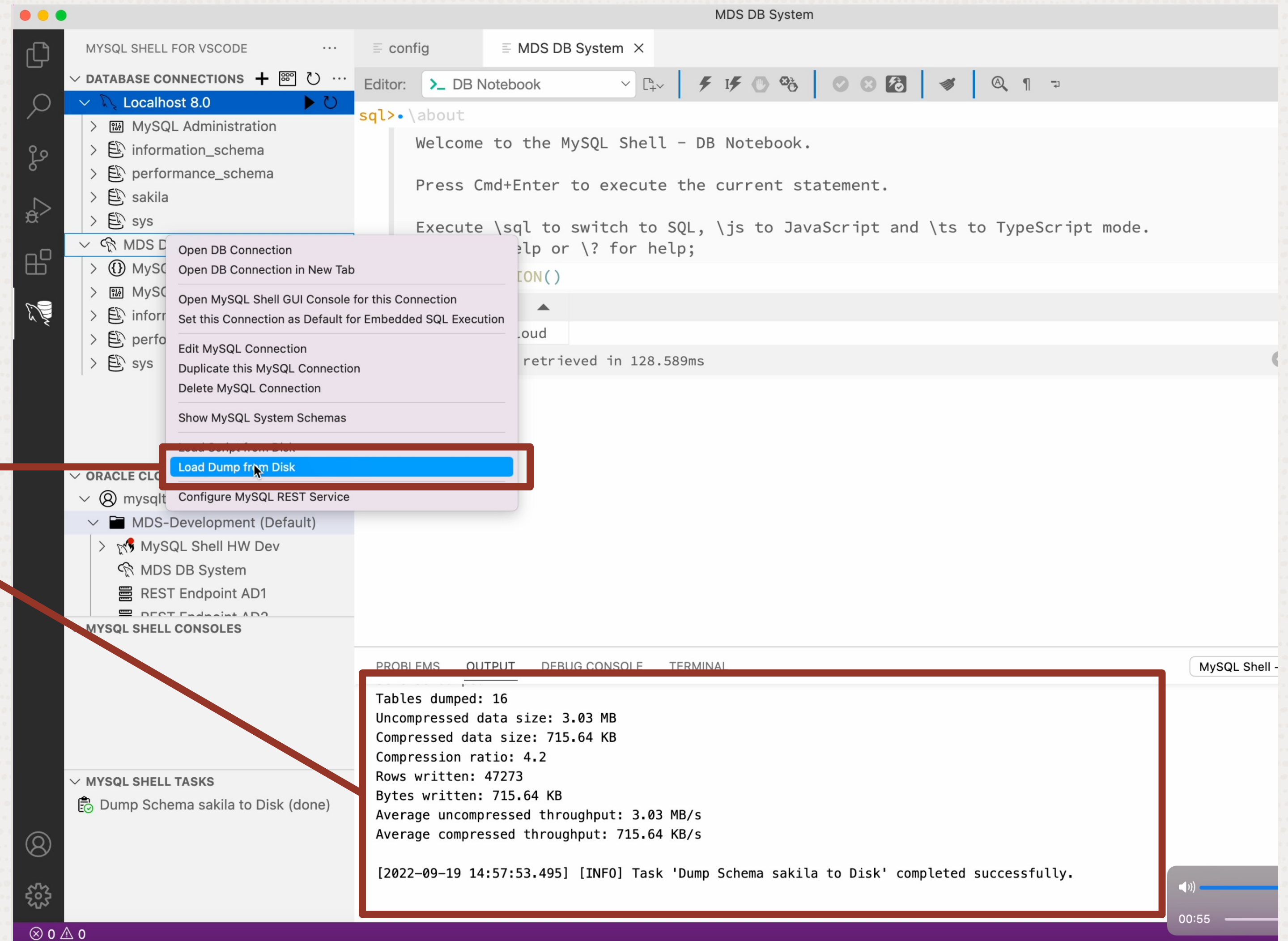
Buttons for 'Store Password', 'Clear Password', 'Cancel', and 'OK' are visible at the bottom of the dialog.





# Lift and Shift

## MySQL Shell Dump & Load Within VS Code





# Integration MySQL HeatWave for OLAP

Execute Directly with HeatWave

Connect Directly from Customers On-Premise Network via Bastion

MDS/Heatwave Management

HeatWave Cluster Tuning

The screenshot displays the MySQL Shell for VS Code interface. On the left, the 'DATABASE CONNECTIONS' tree shows 'MySQL Heatwave MDS' selected. Below it, 'ORACLE CLOUD INFRASTRUCTURE' shows 'mysqltooling (us-ashburn-1)' with 'MySQL Heatwave MDS' selected. At the bottom, 'MYSQL SHELL TASKS' shows 'Rescale HeatWave Cluster (done)'. On the right, the 'MySQL Heatwave MDS' terminal shows SQL commands and their results. A tooltip indicates 'Execute selection or full block on HeatWave'.

```
sql> \about
Welcome to the MySQL Shell - DB Notel...

Press Cmd+Enter to execute the current statement.

Execute \sql to switch to SQL, \js to JavaScript and \ts to TypeScript
Execute \help or \? for help;

sql> SELECT * FROM `hw`.`test`
+----+-----+-----+-----+
| id  | name  | age  | profession |
+----+-----+-----+-----+
| 1182652 |      | 0    | xXyvIfzhpZezgixKmauVwszrkceEQtsHkdLxbC |
| 1184603 |      | 0    | zOvIhIZUBYoyyYyswcWDdDHDWWRxKjPyVkmEp |
| 1185585 |      | 0    | IXPismkmGvh cLCBbCJNLuKuSeSdKq DqTyIV |
| 1187806 |      | 0    | dIhMmYiSPySYlluqtXFkinNdaUvrwjDtervaT |
| 1191323 |      | 0    | NLR BgcTnGuekOobqzxQLhECYjaACjoRrhKgZ |
| 1193768 |      | 0    | Rjr iPCrbfzfdaTqYUBbAzUbynRwFNaMLsETGL |
| 1194169 |      | 0    | pmqRmLUpMTiirjVZlgWrQhnRxIbgBqWLQYsRk |
| 1195602 |      | 0    | Vc rlcFUHEbsFcxXzCrbfxWsVDeLvPPFGsSk> |
| 1196063 |      | 0    | tdlWZgG0aJyLLYHrMM mHCPtvX JUwybMHcnf |
| 1196554 |      | 0    | UjkwBusEVPnUKQZBIKCHGhQKftAzR0tzsorQe |
| 1197375 |      | 0    | 7UD77-META...GACE1...DAishM...Ab-1 |
+----+-----+-----+-----+
OK, 1000 records retrieved in 211.101ms

sql> SELECT COUNT(*) FROM hw.test;
+-----+
| COUNT(*) |
+-----+
| 2000792  |
+-----+
OK, 1 record retrieved in 144.559ms

sql> SELECT age % 7, COUNT(age) FROM `hw`.`test`
GROUP BY age % 7 ORDER BY age % 7;

sql>
```





# Executing Embedded SQL

Execute SQL Snippets in MySQL Shell

DB Notebook interface

Update SQL Snippets into source

The screenshot displays a VS Code editor with a file explorer on the left showing a project structure for 'MRS\_PLUGIN'. The main editor window shows a Python file 'db\_objects.py' with a code snippet for querying a database. A red box highlights the SQL query within the Python code:

```
else:  
    sql = '''  
        SELECT ROUTINE_NAME AS OBJECT_NAME  
        FROM INFORMATION_SCHEMA.ROUTINES  
        WHERE ROUTINE_SCHEMA = ? /*=sakila*/  
        AND ROUTINE_TYPE = 'PROCEDURE'  
        ORDER BY ROUTINE_NAME  
    '''  
  
    res = session.run_sql(  
        sql, [schema.get("name")]  
    )  
    rows = res.fetch_all()  
    db_objects = \  
        [row.get_field("OBJECT_NAME") for row in rows]  
  
    if len(db_objects) == 0:  
        raise ValueError(  
            f"No database objects of type {db_object_type} in the "  
            f"database schema {schema.get('name')}")
```

Below the code editor, the 'DB Notebook' interface is visible, showing the same SQL query in a 'sql>' prompt. A red box highlights the query in the notebook:

```
sql> SELECT ROUTINE_NAME AS OBJECT_NAME  
FROM INFORMATION_SCHEMA.ROUTINES  
WHERE ROUTINE_SCHEMA = /*=sakila*/  
AND (ROUTINE_TYPE = 'PROCEDURE'  
OR ROUTINE_TYPE = 'FUNCTION')  
ORDER BY ROUTINE_NAME
```

Below the notebook, a table of results is shown:

OBJECT_NAME
film_in_stock
film_not_in_stock
get_customer_balance
inventory_held_by_customer
inventory_in_stock
rewards_report

On the right side of the notebook, a context menu is open, listing various actions. A red box highlights the 'Update SQL in Original Source File' option, which has the keyboard shortcut `⌘U`.

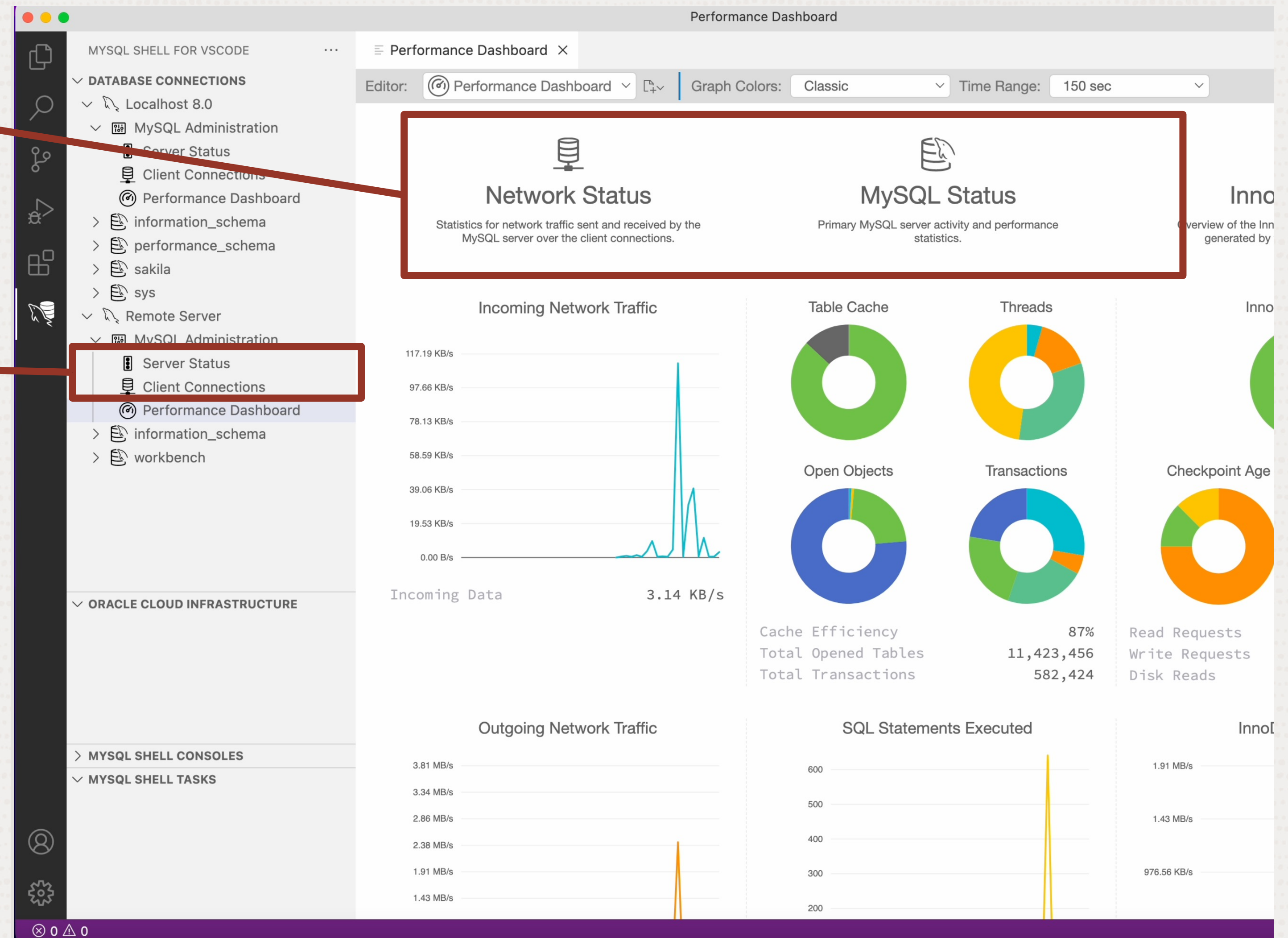




# MySQL Administration

MySQL Performance Metrics

Quick overview of Server Status





# Download MySQL Shell for VSCode now on [VSCode Marketplace](#)

The screenshot displays the MySQL Shell for VS Code interface. On the left, the 'DATABASE CONNECTIONS' sidebar shows 'Localhost 8.0' with a tree view of databases (MySQL Administration, information\_schema, performance\_schema, sakila) and tables (actor, address, category, city, country, customer, film, film\_actor, film\_category, film\_text, inventory, language, payment, rental, staff). The main editor shows a 'DB Notebook' with two SQL queries and their results.

**Query 1:** `SELECT * FROM `sakila`.`actor` WHERE actor_id < 10`

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINESSS	2021-09-28 22:18:53
2	NICK	WAHLBERG	2006-02-15 04:34:33
3	DENNIS	CHASSIN	2006-02-15 04:34:33
4	ENRIQUE	WIS	2006-02-15 04:34:33
5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
6	BETTE	NICHOLSON	2006-02-15 04:34:33
7	GRAE	MANTLE	2006-02-15 04:34:33
8	MATHEW	JOHANSSON	2006-02-15 04:34:33
9	JOE	SWANK	2006-02-15 04:34:33
10	CHRISTIAN	GABLE	2006-02-15 04:34:33

**Query 2:** `SELECT * FROM `sakila`.`address``

address_id	address	address2	district	city_id	postal_code	phone
1	47 MySakila Drive	NULL	Alberta	300		
2	28 MySQL Boulevard	NULL	QLD	576		
3	23 Workhaven Lane	NULL	Alberta	300		1
4	1411 Lillydale Drive	NULL	QLD	576		6
5	1913 Hanoi Way		Nagasaki	463	35200	2
6	1121 Loja Avenue		California	449	17886	8
7	692 Joliet Street		Attika	38	83579	4
8	1566 Inegl Manor		Mandalay	349	53561	7
9	53 Idfu Parkway		Nantou	361	42399	1
10	1795 Santiago de Compostela Way		Texas	295	18743	8







**ORACLE**  
CloudWorld  
Thank you

Introduction to  
MySQL Shell for VS Code

**Kenny Gryp, MySQL Product Manager, Oracle**