Faster Application Development for MySQL

Mike Frank, MySQL Product Management Director | Oracle
Streamlining

Software development can be complicated

Developers strive to deliver high-quality products faster and more efficiently

Streamlining for Creating, Designing and Testing software requires
- Simplification
- Automation
- Optimization

In this talk we will present new techniques for "Streamlining” AppDev with MySQL”
Streamlining development for MySQL

1 - Integration with VS Code for MySQL Devs and DBAs

- SQL IDE for MySQL
- MySQL Shell Extension for VS Code
- Support SQL, JavaScript, Python
- Admin APIs
- Connections
- MySQL HeatWave and OCI Integration

2 - Leverage Progressive Web Apps using REST

- MySQL Restful Service within the MySQL Router
- Automate using create from a schema
- Integration to develop to RestAPIs within VS Code

3 - Supporting JavaScript dev inside the MySQL server

- Write MySQL Stored Procedures
- Develop, debug and and deploy within VS Code

4 - MySQL Kubernetes Operator

- Automates Deployment
- Configuration
- Availability
- Consistent environment for development, testing, and production
- Allows developers to focus on code
1 - Integration with VS Code for MySQL Devs and DBAs

VS Code IDE
- Extensions
- Multi-language
- Multi-platform
- Source Code Control Integration

With MySQL Shell Extension
- Schema Navigation
- SQL Editor
- SQL Worksheets
- Data/Results Grids
- OCI integration

Copyright © 2024 Oracle and/or its affiliates
Leverage Progressive Web Apps using REST

Advantages
• PWAs are platform agnostic – same code across multiple platforms
• Leverages Web Dev Skills
• Faster Development
• Discoverable
• Security Standards supported
• Easy to update

“Progressive Web Apps are just web applications. Using progressive enhancement, new capabilities are enabled in modern browsers. Using service workers and a web app manifest, your web application becomes reliable and installable. If the new capabilities aren't available, users still get the core experience”

https://web.dev/articles/what-are-pwas
3 - Supporting JavaScript dev inside the MySQL server

Execute JavaScript Stored Programs and Stored Functions via GraalVM

Just like SQL Stored Programs, but now with
• Improved Developer Experience
• Security at its core
• State-of-the-art optimizations
• Designed for both Cloud Service and on-premise

Available Now!
• MySQL Heatwave Database Service for OCI, AWS and Azure
• Preview of MySQL Enterprise Edition on Oracle Technology Network (OTN).
## 4 - MySQL Kubernetes Operator

<table>
<thead>
<tr>
<th>Orgs have</th>
<th>Many machines</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Many workloads</td>
</tr>
<tr>
<td>Need to manage those efficiently</td>
<td>So developers can just focus on high value appdev</td>
</tr>
</tbody>
</table>

| Need to hide complexity | Developers just want MySQL servers accessible via some host and port (IP/port) |

| Kubernetes solution | A portable, extensible platform for managing containerized workloads and services |
|                     | Facilitates both declarative configuration and automation |
Details
1 - Integration with VS Code for MySQL Devs and DBAs
VS Code for MySQL Shell

Next generation UI/Dev platform

VS Code Extension - MySQL Shell

Successor to MySQL Workbench

IDE for MySQL DBAs and Developers
Key Features

- SQL Editor
- SQL Worksheets
- Schema Navigation
- Data/Results Grids
- JavaScript and TypeScript Editors
- Query, manipulate, and visualize your data.
- Integrates seamlessly into your development workflow.
- OCI MySQL Heatwave Service Integration
Install VS Code

Download

Run installer

Start VS Code

https://code.visualstudio.com/download
Add the MySQL Shell Extension
As Developer, the easiest way to connect to your MySQL DB System if you don’t have any VPN to your OCI tenancy, is to use the Bastion Service:
The right tools for AppDevs and DBAs
Connection Configuration Wizard

• Easily create, test, and save connections

• Store passwords securely to key vaults/rings

• Supports SSH tunneling and OCI Bastion service
Database Object Browser

Quick Find
OCI HeatWave Integrated

Migrate to HeatWave
Run scripts

Intellisense
MySQL Notebooks

- Allows devs to integrate code, text, equations, and visualizations in a single document known as a “notebook.”

VS Code Notebook API allowed the MySQL Visual extension to:
- Open files as notebooks,
- Execute notebook code cells,
- Render notebook outputs in a variety of rich and interactive formats

https://code.visualstudio.com/blogs/2021/11/08/custom-notebooks
Creating a DB Connection

Rescale HA

Load Data

Trace Execution Errors
2 - Leverage Progressive Web Apps using REST
Rapidly develop PWAs using RESTful Web Services

<table>
<thead>
<tr>
<th>Restful Web Services</th>
<th>MySQL Shell for VS Code</th>
<th>Built in User Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Automates creation of REST for tables, views, and procedures</td>
<td>• GUI frontend for MySQL Rest Service management</td>
<td>• Support for popular OAuth2 services</td>
</tr>
<tr>
<td>• {JSON} responses</td>
<td>• Interactive documentation</td>
<td>• User Role, Group &amp; Hierarchy Management</td>
</tr>
<tr>
<td>• Developer support (GUI, CLI, API)</td>
<td>• CLI and scripting support</td>
<td>• User Management UI</td>
</tr>
</tbody>
</table>
High Level Architecture

Fast and secure HTTPS access to your MySQL data.

Implemented as a MySQL Router feature

Simple to configuration

Full UI integration with VS Code for MySQL

Ideally suited for Progressive Web Apps

Highly available and scalable
MySQL Restful Service Overview

Works with all popular MySQL Deployment Models

- Cloud - MySQL HeatWave
- On premises - MySQL InnoDB Cluster Set, Replica Set, …

MySQL Router

- Serves JSON data via RESTful Web Services
Steps

1. Configure MySQL Instance for Rest
2. Add schema to Rest
3. Add database objects to the service
   • Auto REST for tables, views and procedures
4. Manage Rest Objects
Mapping – JSON/Relational Duality Via Rest Service
Manage

• Start/Stop
• Enable/Disable
• Router Service Management
• Bootstrap
• Configure
• Start
• Stop
• Kill

Integrated documentation
Rest Shortcuts

- Copy
- Open
- Dump
- Remove
Browse via Rest
TypeScript integration

**TypeScript**
- JavaScript With Syntax For Types
- VS code for MySQL can report errors when the types don't match

Integration allows interactive execution of TypeScript code inside a DB Notebook.
- Makes working with the MySQL REST Service easier
- Is available within the DB Notebooks

**RESTful development specific**
- TypeScript SDK is updated in real time as REST service involves
- Allows instant prototyping of REST queries using the Client API inside VS Code
Data driven application development

Query builder support to read and write data

Loved by developers

- Intuitive
- Automated
- Type safe
- Autocomplete
- View results
REST APIs for

• create
• createMany
• findFirst
• findUnique
• findUniqueOrThrow
• findMany
• delete
• deleteMany
• update
• updateMany

Learn by example
Includes PWA demo app

Showcases features
• Of the MySQL REST Service

Deployed directly
• From with VS Code

Upload and serve the application
• Using MySQL Routers
3 - Supporting JavaScript dev inside the MySQL server
JavaScript applications with MySQL

JavaScript Applications are popular
• Powerful for light weight front-end and server-side applications
Handles data-intensive use cases
• Data Validation
• JSON & String processing / Formatting
• Data Cleansing / Transformation
• Minimize data movement between server and clients
Streamline using Procedural programs inside Database

<table>
<thead>
<tr>
<th>Handle</th>
<th>Data-intensive app functionality inside the database server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimize</td>
<td>Data movement</td>
</tr>
<tr>
<td>Reduce</td>
<td>Cost</td>
</tr>
<tr>
<td>Improve</td>
<td>Security</td>
</tr>
<tr>
<td>Simplify</td>
<td>ETL (Extract, Transform, and Load) to simpler ELT (Extract, Load, Transform) data pipelines - the modern data warehouse approach</td>
</tr>
</tbody>
</table>
## MySQL Stored Programs - SQL vs JavaScript

<table>
<thead>
<tr>
<th></th>
<th>SQL Stored Procedures</th>
<th>JavaScript Stored Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Expressiveness</strong></td>
<td>Hard to use, lacks basic constructs like containers (arrays, maps)</td>
<td>Highly expressive and robust</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>Challenging to optimize due to interpreted code</td>
<td>Many JS code analysis tools. JavaScript apps are fast and optimized by GraalVM</td>
</tr>
<tr>
<td><strong>Ecosystem</strong></td>
<td>Insufficient: Lacks support from IDEs, debuggers, testing frameworks, ...</td>
<td>Massive ecosystem of tools for developers of JavaScript applications</td>
</tr>
<tr>
<td><strong>Availability of developers</strong></td>
<td>Few experienced programmers Especially with MySQL Ecosystem</td>
<td>13.8 M Developers The most popular developer language</td>
</tr>
<tr>
<td><strong>Reusable 3rd Party libraries</strong></td>
<td>Few, mostly code examples</td>
<td>Thousands</td>
</tr>
</tbody>
</table>

**Note:** SQL Stored Procedures are not as expressive and efficient compared to JavaScript Stored Programs. However, JavaScript Stored Programs face challenges in terms of ecosystem, availability of developers, and reusable 3rd party libraries.
JavaScript

Ubiquitous
• One of the most used language by developers*
• > 98% of all web pages use JavaScript**

Multiple Runtimes
• Support in all major web browsers
• Massively used server-side runtimes
• Node.js
• Deno

Development Eco-system
• Npm contains > 2 million free to use JavaScript packages***
• > 10 million users use the npm package manager

* Stack Overflow 2024 survey  
** https://w3techs.com/technologies/details/cp-javascript  
*** https://www.npmjs.com/
Defining JavaScript stored programs

Simple Syntax
• LANGUAGE clause now allows JavaScript
• String quoting mechanism to enclose non-SQL language
  • AS$$...$$
  • AS $JavaScript$ ... $JavaScript$

Function Environment
• No function redefinition in JavaScript required
• SQL argument identifiers directly available in JavaScript

Auto Type-Conversion
Transparent MySQL ↔ JavaScript type conversion
Supports all variations of INT, FLOATS, DATETIME, VARCHAR (utf8mb4)
**JavaScript inside SQL**

**SELECT**
- Use anywhere where SQL stored functions can be used
- Expressions, Projection, WHERE clause, GROUP-BY, JOIN, ORDER BY, HAVING etc.

**DMLs, DDLs, VIEWs**
- Support inside DMLs (INSERT, UPDATE, DELETE, ...)
- DDLs including CREATE TABLE AS SELECT
- Support inside VIEWs

**Interoperability**
- Invoke JavaScript & SQL functions and Programs inside existing SQL stored functions or procedures
- Chain JavaScript & SQL stored functions together using input / output arguments

```sql
SELECT col1, col2, gcd_js(col1,col2) FROM my_table WHERE gcd_js(col1, col2) > 1 ORDER BY gcd_js(col1, col2); CREATE TABLE gcd_table AS SELECT gcd_js(col1,col2) FROM my_table;
```

```sql
CREATE TABLE gcd_table AS SELECT gcd_js(col1,col2) FROM my_table;
```
SQL inside JavaScript

Statement Types
• Simple SQL statements
• Prepared statements with bind parameters

Data Access API
• Execute SQL inside JavaScript using XDevAPI
• Seamless MySQL ↔ JavaScript type conversion for query results

Session State
• Continue transactions inside JavaScript
• Access all session state inside JavaScript such as session variables & temporary tables

CREATE PROCEDURE gen_random_age (IN row_count INT) LANGUAGE JAVASCRIPT AS $
let insertStatement = session.prepare("INSERT INTO my_table(age) VALUES (?)"); for (let j = 0; j < row_count; j++) {
  let random_age = Math.trunc(Math.random() * 100);
  insertStatement.bind(random_age).execute();
}
$

CREATE PROCEDURE average_age (OUT avg_age FLOAT) LANGUAGE JAVASCRIPT AS $
let age_sum = 0, count = 0;
let selectStatement = session.sql("SELECT age FROM my_table");
let result = selectStatement.execute(), row = null;
while (row = result.fetchOne()) {
  age_sum += row[0]; count++;
}
avg_age = age_sum / count;
$

Copyright © 2024 Oracle and/or its affiliates
Debugging simplified

Standard Streams
• Access language standard output and error streams inside MySQL

Error Handling
• Translates unhandled JavaScript exceptions into MySQL errors
• Allow access to JavaScript stack traces in case of unhandled runtime error
• Translates MySQL errors and warnings into JavaScript exceptions while executing SQL statements inside JavaScript

```sql
CREATE PROCEDURE division (IN a INT, IN b INT, OUT result DOUBLE) LANGUAGE JAVASCRIPT AS $$
  function validate(num) {
    console.log("validating input value: ", num);
    if (num === 0) throw ("Division by Zero!");
  }
  validate(b);
  result = a / b;
$$
```

```sql
CALL division( 5, 0, @res);
ERROR 6000 (HY000): JavaScript> Division by Zero!
```

```sql
SELECT mle_session_state("stdout");
validating input value: 0
```

```sql
SELECT mle_session_state("stack_trace"); <js>
validate(division:9:187-214)
<js> division(division:11:222-232)
<js> :anonymous(division:15:256-265)
```
JavaScript inside MySQL Server

Works seamlessly with:
• InnoDB
• HA / Replication
• HeatWave Analytics
• HeatWave AutoML
• HeatWave AutoPilot
• HeatWave Lakehouse
JavaScript Stored Programs: Key Benefits

01 Express complex logic in database using JavaScript

02 Push data-intensive application logic inside the database

03 Reduce data movement cost

04 Includes GraalVM Enterprise Edition optimizations at no additional cost

05 Integrates with MySQL HeatWave
MySQL for Developers License

FREE DOWNLOAD OF MYSQL ENTERPRISE EDITION FROM OTN

Full Access to MySQL Enterprise Edition

- Enterprise Server
- Backup
- Router
- Shell
- Connectors
- JavaScript

Learn, Develop, Prototype

Download Now
4 - MySQL Kubernetes Operator

- Provides a consistent environment across development, testing, and production
- Allows developers to focus on code changes
Kubernetes

A portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.

Kubernetes Operator

Method of automatically deploying and managing a service.

Goals of an operator:
• Deployment
• Configuration
• Self-healing
• Backup & Restore
• Observability
• Using Kubernetes custom resources

Both Kubernetes Operator & MySQL InnoDB Cluster share a common goal to make it easier to deploy, automate and manage a service.


Copyright © 2024, Oracle and/or its affiliates. All rights reserved.
Streamlining with Kubernetes

MySQL maintains container images

- Modern container based development model
- Kubernetes MySQL operator seamlessly streamlines
  - Containerized to Containerized
    - Dev to QA
    - QA to Production
MySQL Operator

- Automated deployment and management
  - Server
  - Router
  - HA/DR
- Self-healing
- Backup & Restore to/from
  - Object Stores
- Scaleup/Scaledown
- Rolling upgrades
  - Minimizes downtime
- Configuration Management
- Database Cloning
- Private container registries
- CNCF cert-manager support
- Enterprise Edition

MySQL Operator for Kubernetes Architecture

Uses **Controllers**
- Manage the life-cycle of containerized workloads
- Workload run as **Pods**

A MySQL **Operator** is
- Software running inside the Kubernetes cluster
- Interacts with the Kubernetes API to observe resources and services to assist with the life-cycle management.

Via Kubernetes controllers the operator configures MySQL
- MySQL Servers
- MySQL Replication (using Group Replication)
- MySQL Router
Install the Kubernetes Extension

Define using YAML
VS Code k8 extension running MySQL InnoDB Cluster
Create complete Environment Stack with Kubernetes

For example
Adding WordPress to MySQL

Defining the appdev ecosystem
Conclusion

Equipped with these next generation tools

Developers can

Increase efficiency and productivity

Automate tasks
Focused App Dev
Integrated documentation

Improve quality

Automated checks
Testing environment
Simplify architecture

Reduce Time to Market

Shortened development time
Quickly iterate
Thank You!

Q&A