MariaDB

CLARUSWAY
WAY TO REINVENT YOURSELF

Today’s Takeaways

- Install and configure MariaDB Server in EC2
- Connect to database
  - locally
  - using another EC2 instance
  - using MySQL WorkBench (time permitting)

1

MariaDB

What is MariaDB

- MariaDB is an open source relational database management system (DBMS) that is a compatible drop-in replacement for the widely used MySQL database technology.
- It was created as a software fork (taking the source code from an open source software program and develop an entirely new one) of MySQL by developers who played key roles in building the original database.
- They devised MariaDB in 2009 in response to Oracle Corp.'s acquisition of MySQL, a commitment to stay open source.

MariaDB

Background

- “Fork” from MySQL in 2009

- Intended to maintain high compatibility with MySQL
    - Some new features have caused divergence

- Commitment to stay Open Source
    - MySQL ⇒ Sun MicroSystems (2008) ⇒ Oracle (2009)

- MariaDB meets the same standard enterprise requirements as MySQL, often with additional features, capabilities and options, and by implementing the MySQL protocol and maintaining compatibility with common MySQL data types and SQL syntax, it’s easy to migrate from MySQL to MariaDB without modifying applications and/or dropping requirements.

MariaDB

Comparison Amazon Aurora vs. MariaDB vs. Microsoft SQL Server

Name	Amazon Aurora	MariaDB	Microsoft SQL Server
Description	MySQL and PostgreSQL compatible cloud service by Amazon	MySQL application compatible open source Relational DBMS, enhanced with high availability, security, interoperability and performance capabilities.	Microsoft’s flagship relational DBMS
Primary database model	Relational DBMS	Relational DBMS	Relational DBMS
Secondary database models	Document store	Document store Graph DBMS Spatial DBMS	Document store Graph DBMS Spatial DBMS
License	commercial	Open Source	commercial
Cloud-based only	yes	no	no
Server operating systems	hosted	FreeBSD Linux Solaris Windows	Linux Windows

SQL Basics

- What is SQL?

  Structured query language (SQL) is a programming language for storing and processing information in a relational database.

- What is a Relational Database?

  A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values.

- What is a Non Relational Database?

  NoSQL databases (aka "not only SQL") are non-tabular databases and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph.

- What is a Query?

  A query is a request for data or information from a database table or combination of tables. In the context of queries in a database, it can be either a select query or an action query.

SQL Join

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

a) Let's look at a selection from the "**Orders**" table,

OrderID	CustomerID	OrderDate
10308	2	2022-01-09
10309	3	2021-12-25
10310	5	2021-10-05

b) Then, look at a selection from the "**Customers**" table,

CustomerID	CustomerName	Country
1	Smitha Sajin	Ireland
2	Anu Daniel	India
3	Mary B	USA

Notice that the "**CustomerID**" column in the "**Orders**" table refers to the "**CustomerID**" in the "**Customers**" table. The relationship between the two tables above is the "CustomerID" column. Then, we can create the following SQL statement (that contains an INNER JOIN), that selects records that have matching values in both tables,

Example SQL Query:

```
SELECT Orders.OrderID, Orders.OrderDate, Customers.CustomerName
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

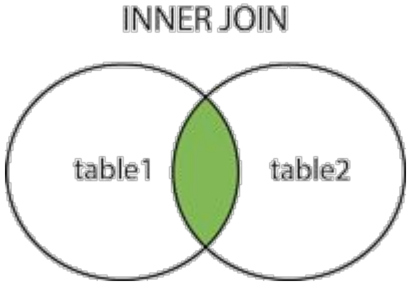
and it will produce something like this:

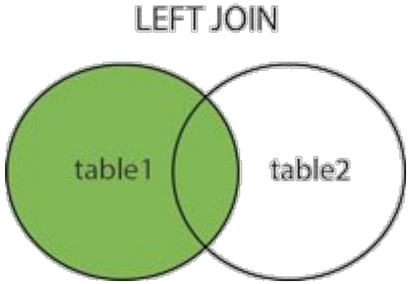
OrderId	OrderDate	CustomerName
10308	2022-01-09	Anu Daniel
10309	2021-12-25	Mary B

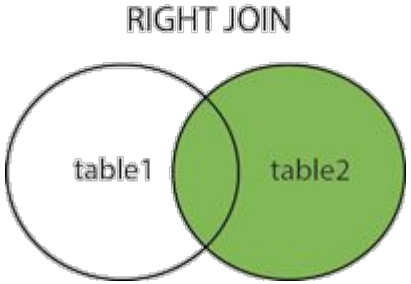
Different Types of SQL JOINS

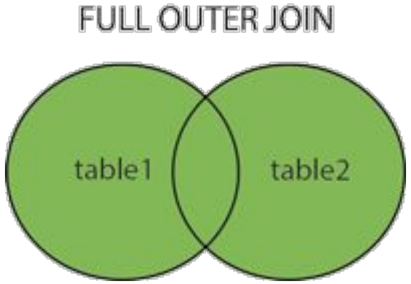
Here are the different types of the JOINs in SQL:

- (INNER) JOIN: Returns records that have matching values in both tables
- LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table









Primary & Foreign Keys

Primary and foreign keys are fundamental concepts in relational databases, and they play a crucial role in maintaining data integrity and relationships between tables.

Primary Key

- **Definition**: A primary key is a unique identifier for each record in a table. It ensures that each record can be uniquely identified and retrieved.
- **Characteristics**:
  - **Uniqueness**: No two rows can have the same primary key value.
  - **Non-null**: Every row must have a value for the primary key.
  - **Immutability**: The primary key value should not change over time.

Example:
sql

```
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    name VARCHAR(100),
    position VARCHAR(50)
);
```

- In this example, `employee_id` is the primary key for the `employees` table.

Primary & Foreign Keys

Foreign Key

- **Definition**: A foreign key is a column or a set of columns in one table that references the primary key columns of another table. It establishes and enforces a link between the data in two tables.
- **Characteristics**:
  - **Referential Integrity**: Ensures that the value in the foreign key column exists in the referenced primary key column.
  - **Consistency**: Helps maintain data consistency between related tables.
  - **Cascading Actions**: Can define actions (e.g., `ON DELETE CASCADE`) to take when the referenced record is updated or deleted.

Example:

sql

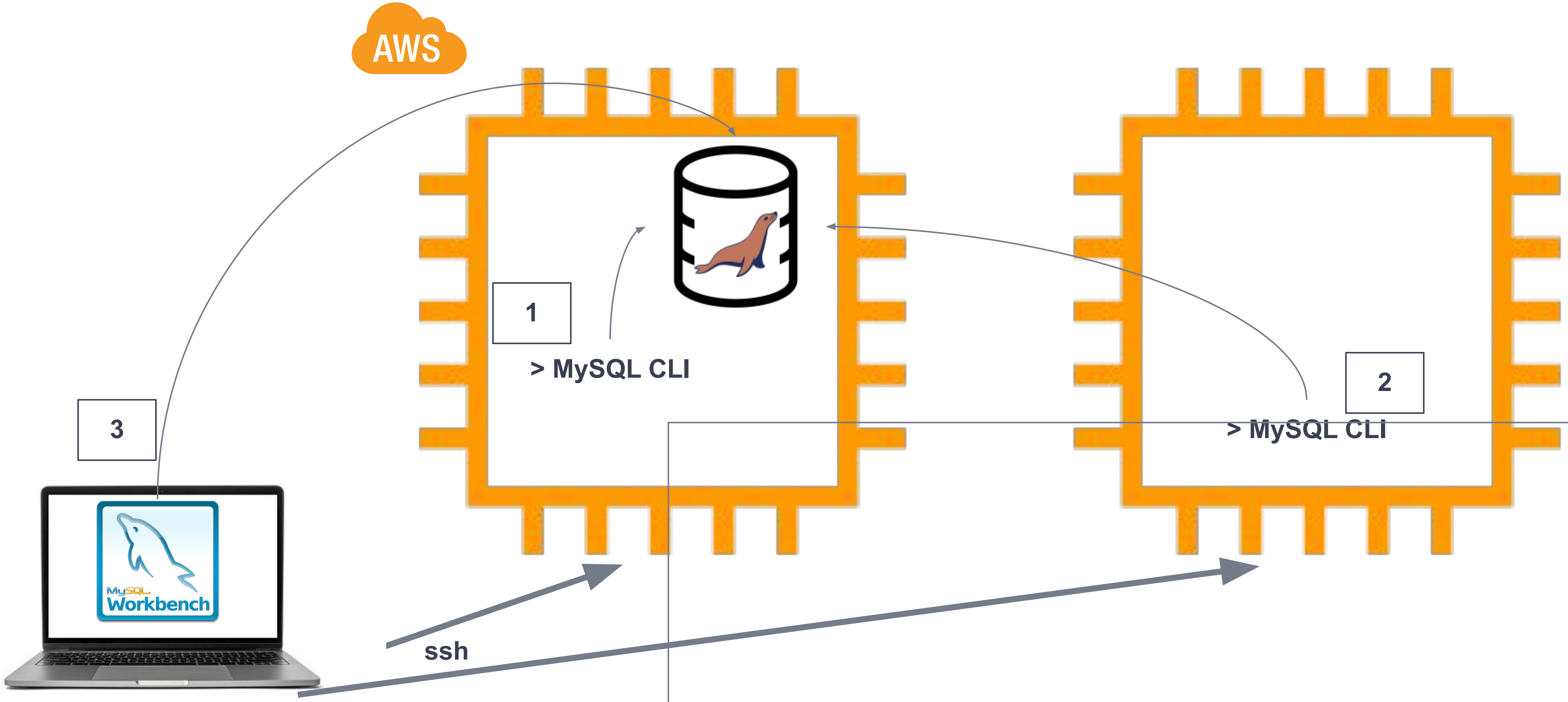
```
CREATE TABLE departments (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(100)
);

CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    name VARCHAR(100),
    position VARCHAR(50),
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES departments(department_id)
);
```

- In this example, `department_id` in the `employees` table is a foreign key that references the `department_id` in the `departments` table.

MariaDB

Database Architecture / Access



** compare to RDS **

Lifecycle Management & Bucket Replication

Let's get our hands dirty!

- Installing and configuring MariaDB

THANKS!

Any questions?

You can find me at:

- @sumod
- sumod@clarusway.com