

HOMEWORK 1.2

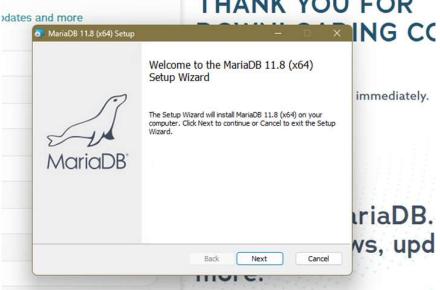
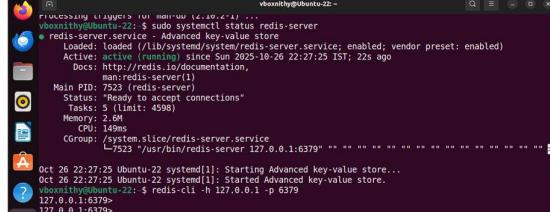
Installing and Creating Databases

Homework 1.2

Objective: Installing MySQL & Creating Databases

- Tasks:
 - Install MySQL using the Developer Default setup and configure the server with a password.
 - Open MySQL Workbench and run the provided SQL script to create a sample database and tables.
 - Verify your database by querying tables in MySQL Workbench.
- Submit:
 - Installation steps summary
 - SQL commands used
 - Screenshot of your created database & table data
 - Short note (3-4 lines) on the benefits of MySQL Workbench
- Resource Link: [Installing MySQL and Creating Databases](#)

The databases that I have worked with: SQL (SQLite, PostgreSQL, MySQL), NoSQL (MongoDB, Firebase). So, I explored MariaDB (SQL) and Redis (NoSQL).

Activity	MariaDB	Redis
What?	Community's enhancement of MySQL	High-speed in-memory key-value data structure server, publish & subscribe model
Type	SQL	NoSQL
Installation	 <p>Download OS specific installer, and follow the steps of the setup wizard (locate the data directory, setup password, service name) and install.</p>	Installation: <pre>sudo apt-get install redis-server sudo apt install redis-tools</pre> Connecting Client: <pre>redis-cli -h 127.0.0.1 -p 6379</pre> 
Create Sample DB and tables	SQL Commands: <ol style="list-style-type: none"> 1. <pre>mariadb -u root -p CREATE DATABASE Company; USE Company;</pre> 2. <pre>CREATE TABLE Employee(e_id INT PRIMARY KEY AUTO_INCREMENT, e_fname VARCHAR(100) NOT NULL, e_lname VARCHAR(100), e_ibanc VARCHAR(34) UNIQUE, e_dob DATE);</pre> 3. <pre>CREATE TABLE Incomes (i_id INT PRIMARY KEY AUTO_INCREMENT, emp_id INT, i_quarter INT NOT NULL,</pre> 	Commands: String <ol style="list-style-type: none"> 1. SET key value – to set an atomic value(string) to a key 2. GET key List: <ol style="list-style-type: none"> 3. RPUSH key val – to set a list of values to a key by inserting at the end 4. LPUSH key val – inserts new element at the beginning of the list 5. LRANGE key start stop – display elements in the range

	<pre>i_income DECIMAL(13, 2) NOT NULL, i_recorded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, FOREIGN KEY (emp_id) REFERENCES Employee(e_id)); INSERT INTO Employee (e_fname, e_lname, e_ibann, e_dob) VALUES ('Rohan', 'Sharma', 'IN1001A', '1990-05-15'), ('Priya', 'Gupta', 'IN1002B', '1992-11-20'), ('Amit', 'Patel', 'IN1003C', '1988-01-30'), ('Sneha', 'Singh', 'IN1004D', '1995-07-22'); INSERT INTO Incomes (emp_id, i_quarter, i_income) VALUES (1, 1, 50000.00), (1, 2, 52000.00), (1, 3, 51000.00), (2, 1, 60000.00), (2, 2, 61000.00), (3, 1, 45000.00), (3, 2, 45000.00), (4, 1, 55000.00), (4, 2, 56000.00);</pre>
Query DB <ul style="list-style-type: none"> Retrieve all employees with the details: - Full name (fname + lname) - Total taxable income (sum of income in all quarters of employee) - Tax on the taxable income (10%) <pre>SELECT e.e_id, CONCAT_WS(' ', e.e_fname, e.e_lname) AS 'fullname', SUM(i.i_income) AS 'total_taxable_income', ROUND(SUM(i.i_income) * 0.10) AS 'tax_amount' FROM Employee AS e JOIN Incomes AS i ON e.e_id = i.emp_id GROUP BY e.e_id;</pre>	<p>Initially</p> <pre>127.0.0.1:6379> 127.0.0.1:6379> keys * (empty array) 127.0.0.1:6379> config get databases 1) "databases" 2) "16" 127.0.0.1:6379> info keyspace # Keyspace</pre> <p>After data insertion:</p> <pre>127.0.0.1:6379> 127.0.0.1:6379> keys * 1) "online:users" 2) "my_string" 3) "my_queue" 127.0.0.1:6379> config get databases 1) "databases" 2) "16" 127.0.0.1:6379> info keyspace # Keyspace db0:keys=3,expires=0,avg_ttl=0 127.0.0.1:6379> 127.0.0.1:6379> 127.0.0.1:6379></pre>
Screen-shots <p>1. Creating tables and inserting data</p> <pre>MariaDB [Company]> CREATE TABLE Employee(-> e_id INT PRIMARY KEY AUTO_INCREMENT, -> e_fname VARCHAR(100) NOT NULL, -> e_lname VARCHAR(100), -> e_ibann VARCHAR(34) UNIQUE, -> e_dob DATE ->); Query OK, 0 rows affected (0.085 sec) MariaDB [Company]> CREATE TABLE Incomes (-> i_id INT PRIMARY KEY AUTO_INCREMENT, -> emp_id INT, -> i_quarter INT NOT NULL, -> i_income DECIMAL(13, 2) NOT NULL, -> i_recorded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -> FOREIGN KEY (emp_id) REFERENCES Employee(e_id) ->); Query OK, 0 rows affected (0.057 sec) MariaDB [Company]> INSERT INTO Employee (e_fname, e_lname, e_ibann, e_dob) -> VALUES -> ('Rohan', 'Sharma', 'IN1001A', '1990-05-15'), -> ('Priya', 'Gupta', 'IN1002B', '1992-11-20'), -> ('Amit', 'Patel', 'IN1003C', '1988-01-30'), -> ('Sneha', 'Singh', 'IN1004D', '1995-07-22'); Query OK, 4 rows affected (0.065 sec) Records: 4 Duplicates: 0 Warnings: 0 MariaDB [Company]> INSERT INTO Incomes (emp_id, i_quarter, i_income) VALUES -> (1, 1, 50000.00), -> (1, 2, 52000.00), -> (1, 3, 51000.00), -> (2, 1, 60000.00), -> (2, 2, 61000.00), -> (3, 1, 45000.00), -> (3, 2, 45000.00), -> (4, 1, 55000.00), -> (4, 2, 56000.00); Query OK, 9 rows affected (0.012 sec) Records: 9 Duplicates: 0 Warnings: 0</pre>	<p>1. String</p> <pre>127.0.0.1:6379> 127.0.0.1:6379> SET my_string "Nithya" OK 127.0.0.1:6379> GET my_string "Nithya" 127.0.0.1:6379> DEL my_string (integer) 1 127.0.0.1:6379></pre> <p>2. List</p> <pre>127.0.0.1:6379> LPUSH my_queue "A" (integer) 1 127.0.0.1:6379> RPUSH my_queue "B" (integer) 2 127.0.0.1:6379> RPUSH my_queue "C" (integer) 3 127.0.0.1:6379> GET my_queue (error) WRONGTYPE Operation against a key holding the wrong kind of value 127.0.0.1:6379> LRANGE my_queue 0 -1 1) "A" 2) "B" 3) "C" 127.0.0.1:6379> LLEN my_queue (integer) 3 127.0.0.1:6379> LPOP my_queue "A" 127.0.0.1:6379> LRANGE my_queue 0 -1 1) "B" 2) "C"</pre>

2. Show all tables

```
MariaDB [Company]> SHOW TABLES;
+-----+
| Tables_in_company |
+-----+
| employee      |
| incomes       |
+-----+
2 rows in set (0.027 sec)

MariaDB [Company]> SELECT * FROM employee;
+-----+
| e_id | e_fname | e_lname | e_ibancard | e_dob |
+-----+
| 1    | Rohan   | Sharma  | IN1001A  | 1990-05-15 |
| 2    | Priya   | Gupta   | IN1002B  | 1992-11-20 |
| 3    | Amit    | Patel   | IN1003C  | 1988-01-30 |
| 4    | Sneha   | Singh   | IN1004D  | 1995-07-22 |
+-----+
4 rows in set (0.004 sec)

MariaDB [Company]> SELECT * FROM incomes;
+-----+
| i_id | emp_id | i_quarter | i_income | i_recorded_at |
+-----+
| 1    | 1       | 1          | 50000.00 | 2025-10-26 21:36:27 |
| 2    | 1       | 2          | 52000.00 | 2025-10-26 21:36:27 |
| 3    | 1       | 3          | 51000.00 | 2025-10-26 21:36:27 |
| 4    | 2       | 1          | 60000.00 | 2025-10-26 21:36:27 |
| 5    | 2       | 2          | 61000.00 | 2025-10-26 21:36:27 |
| 6    | 3       | 1          | 45000.00 | 2025-10-26 21:36:27 |
| 7    | 3       | 2          | 45000.00 | 2025-10-26 21:36:27 |
| 8    | 4       | 1          | 55000.00 | 2025-10-26 21:36:27 |
| 9    | 4       | 2          | 56000.00 | 2025-10-26 21:36:27 |
+-----+
9 rows in set (0.003 sec)
```

3. Set

```
127.0.0.1:6379> SADD online:users "user:1" "user:2" "user:3"
(integer) 3
127.0.0.1:6379> SADD online:users "user:3" "user:4"
(integer) 1
127.0.0.1:6379> SMEMBERS online:users
1) "user:2"
2) "user:3"
3) "user:1"
4) "user:4"
127.0.0.1:6379> SISMEMBER online:users "user:2"
(integer) 1
127.0.0.1:6379> SREM online:users "user:2"
(integer) 1
127.0.0.1:6379> SCARD online:users
(integer) 3
127.0.0.1:6379>
```

3. Required query

```
MariaDB [Company]> SELECT
-->     e.e_id,
-->     CONCAT_WS(' ', e.e_fname, e.e_lname) AS 'fullname',
-->     SUM(i.i_income) AS 'total_taxable_income',
-->     ROUND(SUM(i.i_income) * 0.10) AS 'tax_amount'
-->   FROM
-->     Employee AS e
-->   JOIN
-->     Incomes AS i ON e.e_id = i.emp_id
-->   GROUP BY
-->     e.e_id;
+-----+
| e_id | fullname      | total_taxable_income | tax_amount |
+-----+
| 1    | Rohan Sharma  | 153000.00           | 15300      |
| 2    | Priya Gupta   | 121000.00           | 12100      |
| 3    | Amit Patel    | 90000.00            | 9000       |
| 4    | Sneha Singh   | 111000.00           | 11100      |
+-----+
4 rows in set (0.039 sec)
```

PS: The example of DB and query taken was a part of the problem in my Uber's Data Science Interview.

Benefits

```
MariaDB [Company]> EXIT;
Bye
```

More friendly 😊

- Structured Data (ACID)
- Powerful for querying & establishing relationships

```
127.0.0.1:6379> shutdown nosave
not connected
```

- High-speed & versatile
- Sharding for better write access,
- Master slave replication for read access

- Nithya(1RV22CS099)

References:

- <https://mariadb.com/get-started-with-mariadb/>
- <https://mariadb.com/docs/server/mariadb-quickstart-guides/basics-guide>
- <https://www.w3schools.io/nosql/redis-cli/>
- <https://www.w3schools.io/nosql/redis-datatypes/#redis-sets-types>