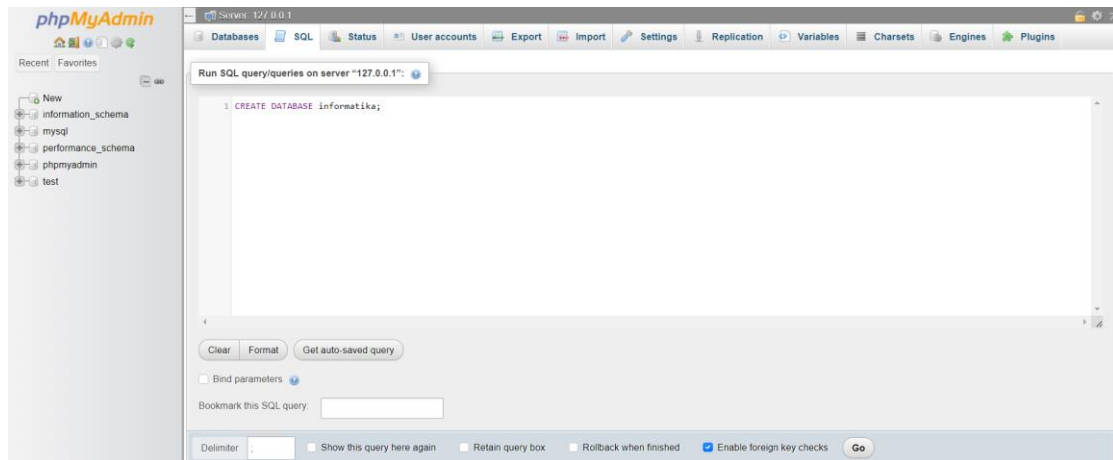**TASK REPORT**

WEEK PROGRAMING

WEEK 4

Created by:

Cesario Hafidh Arifa Noorcholish

L200224217

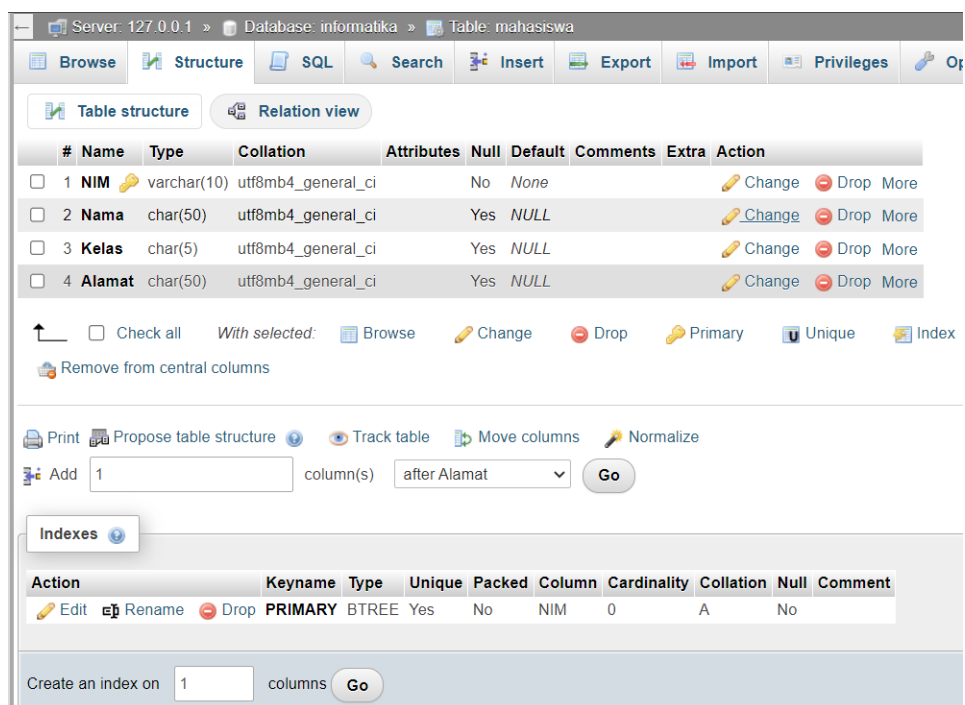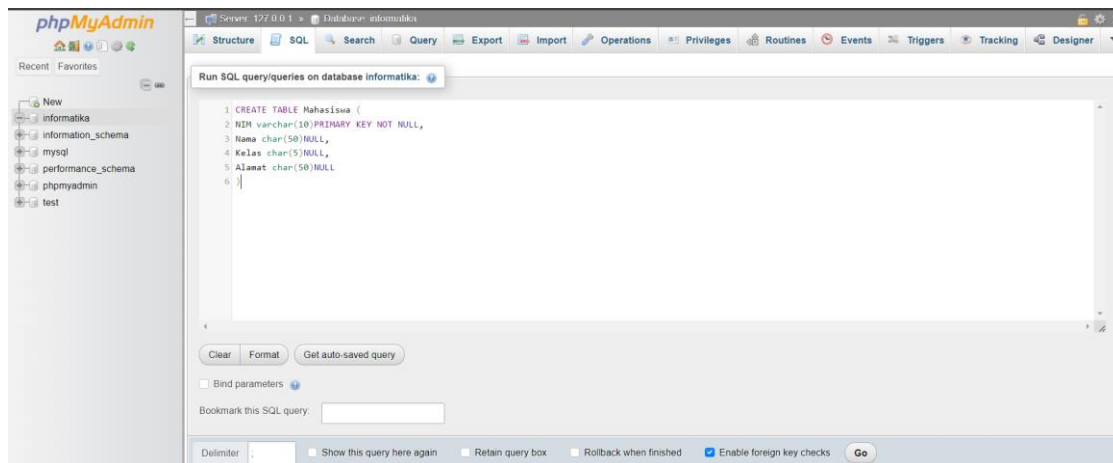**X**

INFORMATICS ENGINEERING

FACULTY OF COMMUNICATION AND INFORMATICS

UNIVERSITAS MUHAMMADIYAH SURAKARTA

2023/2024

TASK

1. **Experiment 1 (Create Database)**

2. **Experiment 2 (Create Table)**

3. **Experiment 3 (Insert Data)**

Server: 127.0.0.1 » Database: informatika

Structure | SQL | Search | Query | Export | Import

Run SQL query/queries on database informatika:

```
1  INSERT INTO Mahasiswa VALUES('L20080001','Ari Wibowo','A','Solo')
```

Server: 127.0.0.1 » Database: informatika » Table: Mahasiswa

Browse | Structure | SQL | Search | Insert | Export | Import

Show query box

✔ Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

```
SELECT * FROM Mahasiswa;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ∨ | Filter rows: Search this table

Extra options

← T →		NIM	Nama	Kelas	Alamat
☐ 🖉 Edit 📋 Copy ⊖ Delete		L20080001	Ari Wibowo	A	Solo

↰ ☐ Check all | With selected: 🖉 Edit 📋 Copy ⊖ Delete 📤 Export

☐ Show all | Number of rows: 25 ∨ | Filter rows: Search this table

Server: 127.0.0.1 » Database: informatika

Structure | SQL | Search | Query | Export | Import | Operations

Run SQL query/queries on database informatika:

```
1  INSERT INTO Mahasiswa (NIM, NAMA, Kelas) VALUES('L20080080','Agustina','B');
```

```
SELECT* FROM mahasiswa;
```

[Edit inline] [Edit] [Create PHP code]

☐ Show all | Number of rows: 25 ∨ | Filter rows: Search this table | Sort by key:

Extra options

← T →		NIM	Nama	Kelas	Alamat
☐ 🖉 Edit 📋 Copy ⊖ Delete		L20080001	Ari Wibowo	A	Solo
☐ 🖉 Edit 📋 Copy ⊖ Delete		L20080080	Agustina	B	NULL

↰ ☐ Check all | With selected: 🖉 Edit 📋 Copy ⊖ Delete 📤 Export

4. **Experiment 4 (Changing Data)**

5. **Experiment 5 (Join)**

1. Table Nilai

2. Experiment


```
1  SELECT Mahasiswa.NIM, Mahasiswa.Nama, Nilai.Nama_MK, Nilai.Nilai_Angka, Nilai.Nilai_Huruf FROM (Mahasiswa
2  JOIN Nilai ON Mahasiswa.NIM = Nilai.NIM);
```

SELECT Mahasiswa.NIM, Mahasiswa.Nama, Nilai.Nama_MK, Nilai.Nilai_Angka, Nilai.Nilai_Huruf FROM (Mahasiswa JOIN Nilai ON Mahasiswa.NIM = Nilai.NIM);

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ▾ | Filter rows: Search this table

Extra options

NIM	Nama	Nama_MK	Nilai_Angka	Nilai_Huruf
L200080001	Ari Wibowo	Pemrograman Web	87	A
L200080080	Agustina Anggraini	Pemrograman Web	90	A

☐ Show all | Number of rows: 25 ▾ | Filter rows: Search this table

Query results operations

6. **Experiment 6 (Left Join)**


```
1  SELECT Mahasiswa.NIM, Mahasiswa.Nama, Nilai.Nama_MK, Nilai.Nilai_Angka, Nilai.Nilai_Huruf
2  FROM (Mahasiswa LEFT JOIN Nilai ON Mahasiswa.NIM = Nilai.NIM;
```

SELECT Mahasiswa.NIM, Mahasiswa.Nama, Nilai.Nama_MK, Nilai.Nilai_Angka, Nilai.Nilai_Huruf FROM (Mahasiswa LEFT JOIN Nilai ON Mahasiswa.NIM = Nilai.NIM);

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ▾ | Filter rows: Search this table

Extra options

NIM	Nama	Nama_MK	Nilai_Angka	Nilai_Huruf
L200080001	Ari Wibowo	Pemrograman Web	87	A
L200080080	Agustina Anggraini	Pemrograman Web	90	A

☐ Show all | Number of rows: 25 ▾ | Filter rows: Search this table

7. **Experiment 7 (Right Join)**


```
1  SELECT Mahasiswa.NIM, Mahasiswa.Nama, Nilai.Nama_MK, Nilai.Nilai_Anka, Nilai.Nilai_Huruf
2  FROM (Mahasiswa RIGHT JOIN Nilai ON Mahasiswa.NIM = Nilai.NIM)
```

```
SELECT Mahasiswa.NIM, Mahasiswa.Nama, Nilai.Nama_MK, Nilai.Nilai_Angka, Nilai.Nilai_Huruf FROM (Mahasiswa RIGHT JOIN Nilai ON Mahasiswa.NIM = Nilai.NIM);
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ▾ | Filter rows: Search this table

Extra options

NIM	Nama	Nama_MK	Nilai_Angka	Nilai_Huruf
L200080001	Ari Wibowo	Pemrograman Web	87	A
NULL	NULL	Kapita Selekta	60	BC
L200080080	Agustina Anggraini	Pemrograman Web	90	A

☐ Show all | Number of rows: 25 ▾ | Filter rows: Search this table

8. Experiment 8 (AVG Function)

Server: 127.0.0.1 » Database: informatika

Structure | SQL | Search | Query | Export | Import

Run SQL query/queries on database informatika: ❓

```
1 SELECT AVG(Nilai_Angka) AS "Rata-Rata Nilai" FROM Nilai;
```

```
SELECT AVG(Nilai_Angka) AS "Rata-Rata Nilai" FROM Nilai;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ▾ | Filter rows: Search this table

Extra options

Rata-Rata Nilai
79.0000

9. Experiment 9 (SUM Function)

Server: 127.0.0.1 » Database: informatika

Structure | SQL | Search | Query | Export

Run SQL query/queries on database informatika: ❓

```
1 SELECT SUM(Nilai_Angka) Total_Nilai FROM Nilai;
```

```
SELECT SUM(Nilai_Angka) Total_Nilai FROM Nilai;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ▾ | Filter rows: Search this table

Extra options

Total_Nilai
237

10. **Experiment 10 (View)**

ASSIGNMENT

1. **What is the function of some of the following:**

   1) SELECT

      The SELECT statement retrieves specific data from one or more database tables. It allows users to specify the columns they want to retrieve and can include additional clauses for filtering, sorting, grouping, and limiting results

   2) JOIN

      The JOIN operation combines rows from two or more tables based on a related column between them. It retrieves records that have matching values in the specified columns from both tables.

   3) LEFT JOIN

      A LEFT JOIN retrieves all records from the table specified on the left side of the join operation and matching records from the table on the right side. If there is no

match in the right table, NULL values are returned for the columns of the right table.

4) RIGHT JOIN

A RIGHT JOIN is similar to a LEFT JOIN, but it retrieves all records from the table specified on the right side of the join operation and matching records from the left table. If there is no match in the left table, NULL values are returned for the columns of the left table.

5) AVG

The AVG function is an aggregate function used to compute the average value of a numeric column in a result set. It excludes NULL values from the calculation.

6) SUM

The SUM function is an aggregate function used to calculate the total sum of values in a numeric column in a result set. It adds up all the values in the specified column.

2. **Write SQL Syntax to populate Address data "Sragen" in the Students table (experiment 3) in NIM L20008080.**