# **PRACTICE REPORT**

# **WEB PROGRAMMING LAB WORKS**

# **MODULE 4**

# **“SQL (Structured Query Language)”**



**Assembled by:**

# **ONIC AGUSTINO**

# **L200234275**

**X**

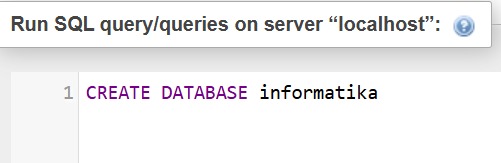
**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**TAHUN 2024/2025**

**PRACTICE**

1. **Experiment 1 (Creating a Database)** 

Picture 1.1 The Code.

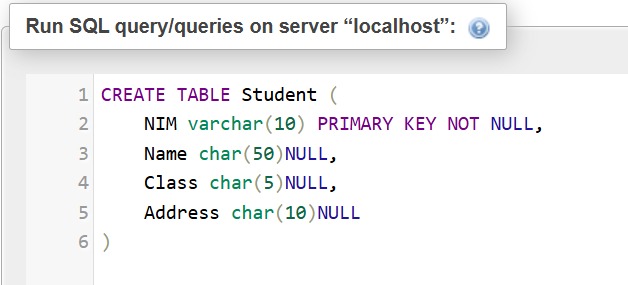
(Explanation)

The **CREATE DATABASE INFORMATIKA;** command in SQL is used to create a new database named **INFORMATIKA** in a database management system (DBMS) such as MySQL or PostgreSQL



Picture 1.2 The Output.

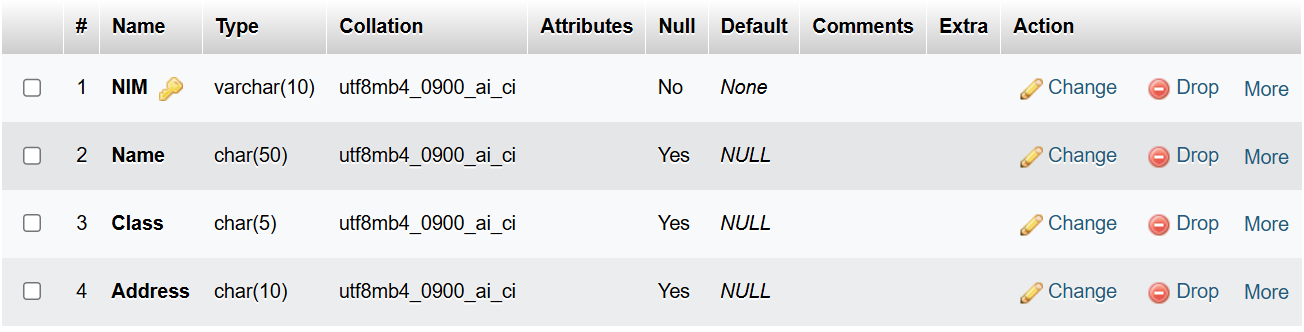
1. **Experiment 2 (Creating a Table)**



Picture 2.1 the code.

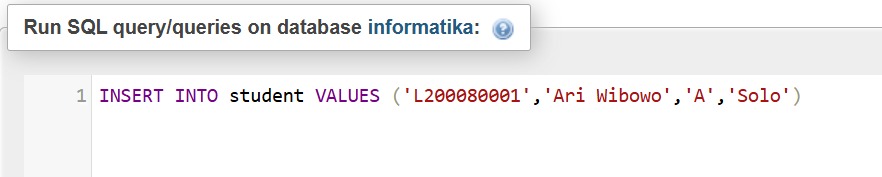
(Explanation)

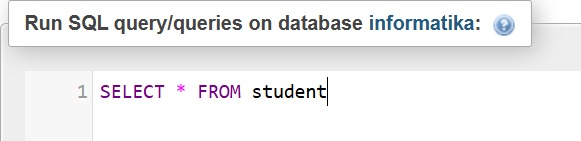
The SQL command **CREATE TABLE Student**creates a table named **student**with four columns: **NIM** (VARCHAR(10) PRIMARY KEY NOT NULL) as the unique identifier for each student, **Name** (CHAR(50) NULL), **Class** (CHAR(5) NULL), and **Address** (CHAR(50) NULL)



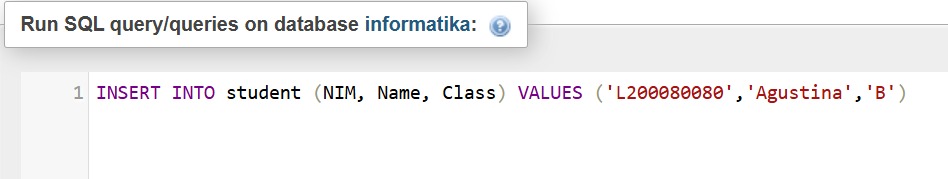
Picture 2.2 the output.

1. **Experiment 3 (Entering Data)**





Picture 3.1 the code 1.



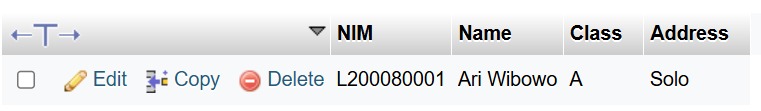


Picture 3.2 the code 2.

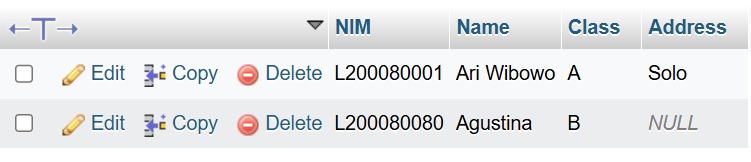
(Explanation)

**Picture 3.1** : The SQL statement **INSERT INTO Student VALUES ('L200080001', 'Ari Wibowo', 'A', 'Solo');** inserts a new record into the **student** table with **NIM** as 'L200080001', **Name** as 'Ari Wibowo', **Class** as 'A', and **Address** as 'Solo'

**Picture 3.2 :** The SQL command **INSERT INTO student** **(NIM, Name, Class) VALUES ('L200080080', 'Agustina', 'B');** inserts a new record into the student table within the informatika database, assigning 'L200080080' to NIM, 'Agustina' to Name, and 'B' to Class, but since Name and Class are reserved keywords in SQL

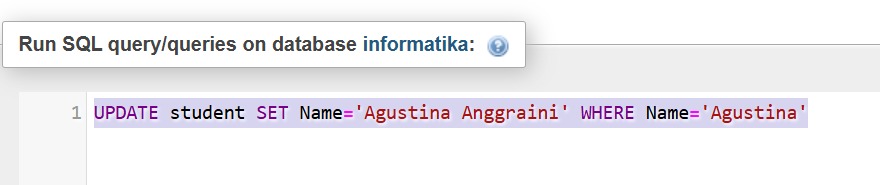


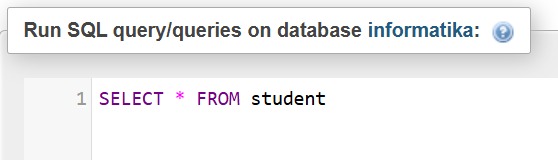
Picture 3.3 the output 1.



Picture 3.4 the output 2.

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

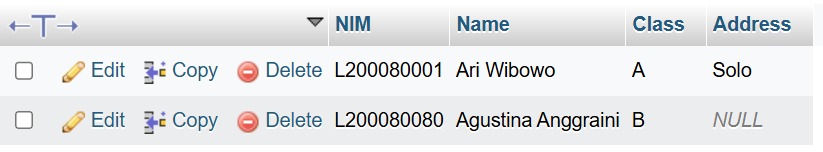




Picture 4.1 the code.

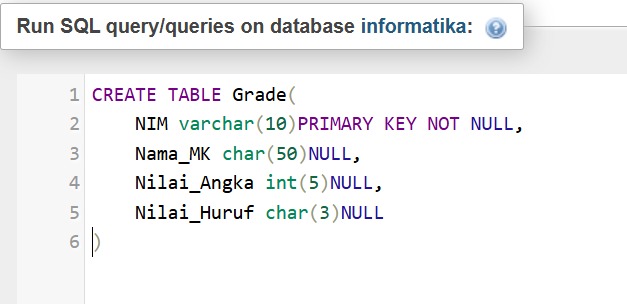
(Explanation)

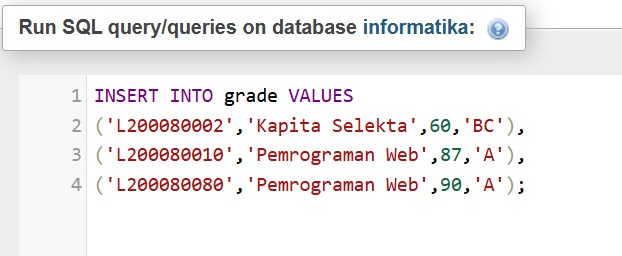
he SQL command **UPDATE student SET Name='Agustina Anggraini' WHERE Name='Agustina';** updates the **Name** column in the **student** table within the **informatika** database, changing the value **'Agustina'** to **'Agustina Anggraini'.**

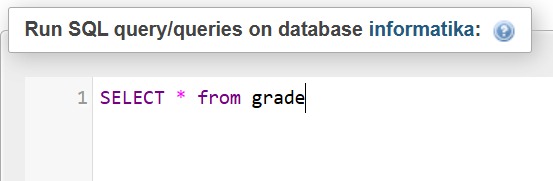


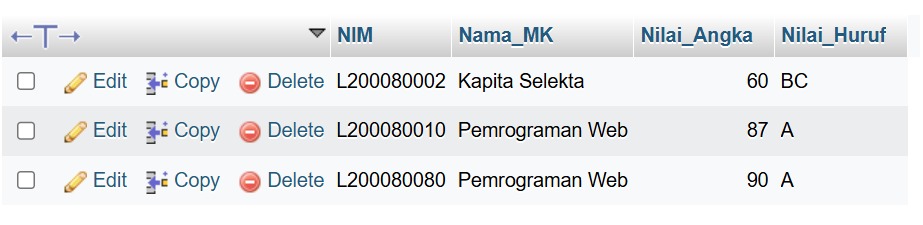
Picture 4.2 the output.

**JOIN**









Create table grade, then content the table, and show results.

1. **Attempt 5 (Join)**



Picture 5.1 the code.

(Explanation)

The SQL query **SELECT student.NIM, student.Name, grade.Nama\_MK, grade.Nilai\_Angka, grade.Nilai\_Huruf FROM student JOIN grade ON student.NIM = grade.NIM;** retrieves student data from the **student** and **grade** tables within the **informatika** database, linking them using **NIM** as a common key.



Picture 5.2 the output.

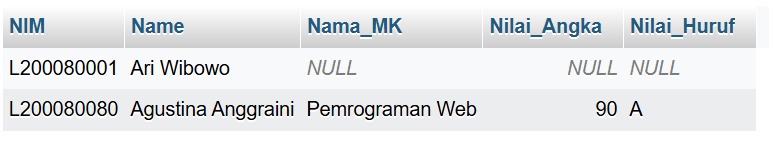
1. **Percobaan 6 (Left Join)**



Picture 6.1 the code.

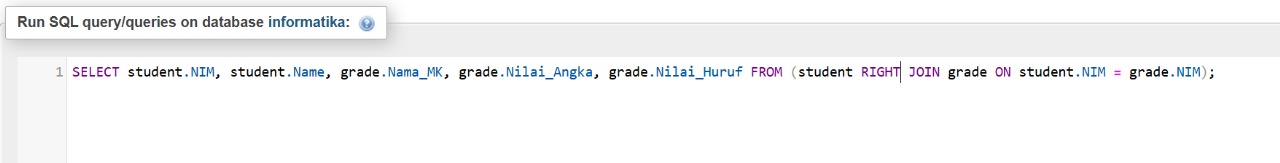
(Explanation)

The SQL query **SELECT student.NIM, student.Name, grade.Nama\_MK, grade.Nilai\_Angka, grade.Nilai\_Huruf FROM student LEFT JOIN grade ON student.NIM = grade.NIM;** retrieves all students from the **student** table and any matching records from the **grade** table, ensuring students without grades are still included.



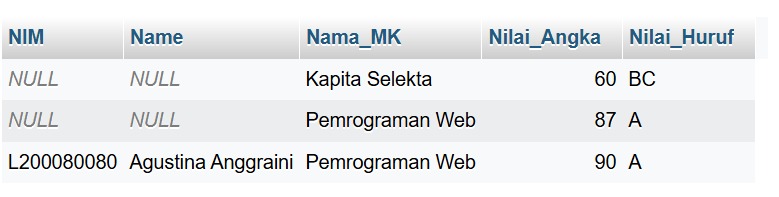
Picture 6.2 the output.

1. **Percobaan 7 (Right Join)**



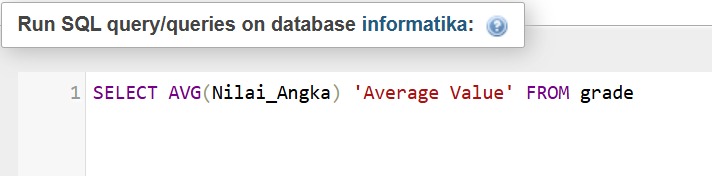
Picture 7.1 the code.

(Explanation)  
he SQL query **SELECT student.NIM, student.Name, grade.Nama\_MK, grade.Nilai\_Angka, grade.Nilai\_Huruf FROM student RIGHT JOIN grade ON student.NIM = grade.NIM;** retrieves all records from the **grade** table and any matching records from the **student** table, ensuring all grades are displayed even if no corresponding student exists.



Picture 7.2 the output.

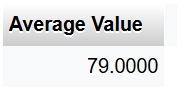
1. **Trial 8 (AVG Function)**



Picture 8.1 the code.

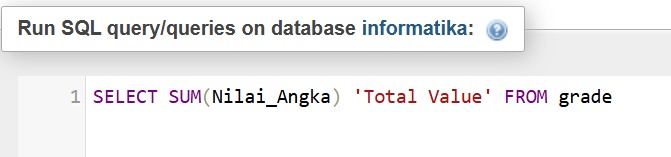
(Explanation)

Calculate the average grade from the grade table



Picture 8.2 the output.

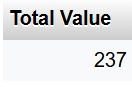
1. **Experiment 9 (SUM Function)**



Picture 9.1 the code.

(Explanation)

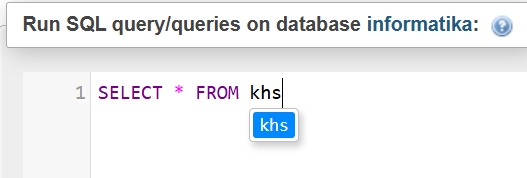
sums up all the number values in the grade table



Picture 9.2 the output.

1. **Experiment 10 (View)**

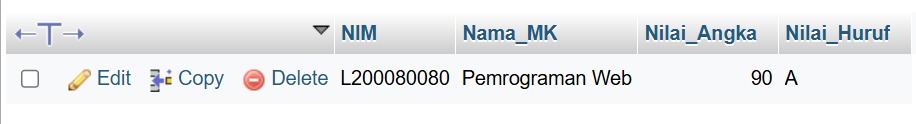




Picture 10.1 the code.

(Explanation)

The SQL query creates a view called KHS, which contains student data and their grades by using an INNER JOIN between the student and grade tables, so that it only displays students who have grades.



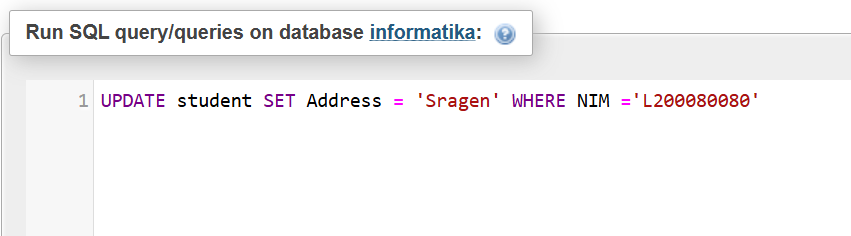
Picture 10.2 the output.

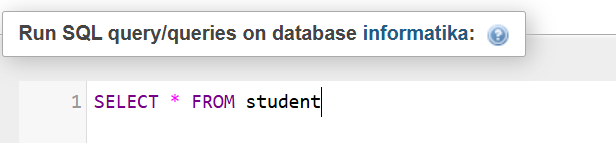
**ASSIGNMENT**

1. **What are the functions of the following things:**

* SELECT = Used to retrieve data from a table in the database.
* JOIN = Used to combine two tables based on a related column.
* LEFT JOIN = Returns all data from the left table and the matching data from the right table. If there is no match, the result from the right table will be NULL.
* RIGHT JOIN = Returns all data from the right table and the matching data from the left table. If there is no match, the result from the left table will be NULL.
* AVG = Calculates the average value of a numeric column.
* SUM = Calculates the total value of a numeric column.

1. **Write SQL syntax to fill in the “Sragen” address data in the Student table (Experiment 3) in L200080080.**

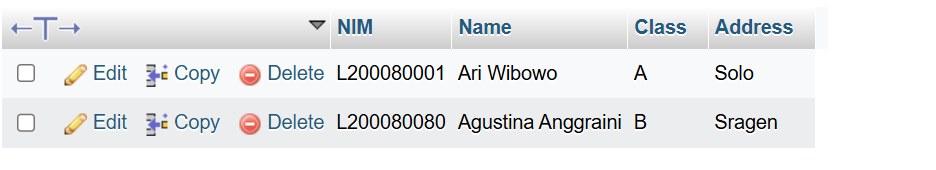




Picture assigment 2.1 the code

(Explanation)

This SQL query updates the address of the student with NIM 'L200080080' in the student table, changing it to 'Sragen'.



Picture Assigments 2.2 the output.