SQL DATA MANIPULATION COMMANDS

- ❑ SELECT statement

  + "WHERE" clause
- ❑ SELECT statement

  + Aggregate Functions

PREPARED BY:
SIR LORENZ

SELECT STATEMENT + "WHERE" CLAUSE

WHERE clause

- is used to filter records in SELECT, UPDATE, DELETE statements
- is used to set conditions (works like “IF statements” in programming) for your queries
- uses logical operators such as:

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal

- [OPTIONAL] can also use relational operators for combination of conditions, such as AND , OR

EXAMPLE

Database name: DB_COLLEGE

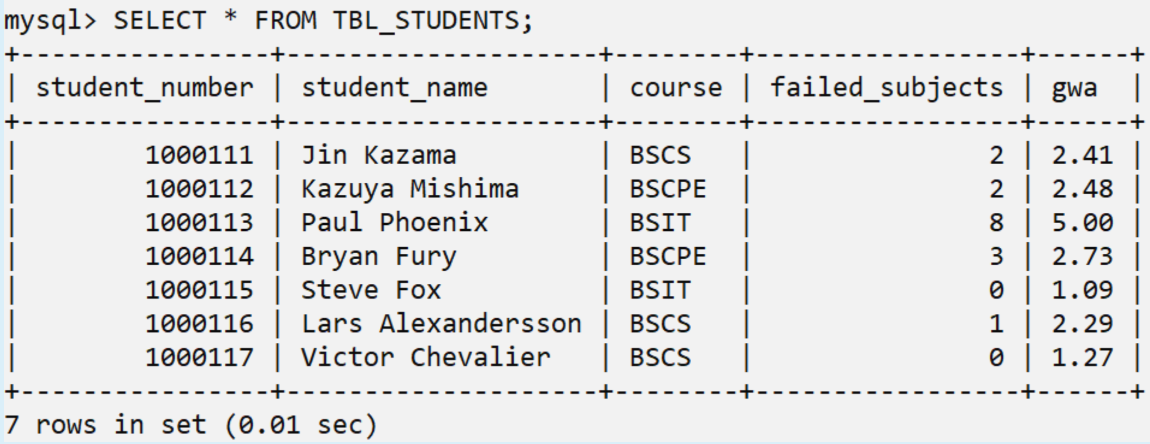
Table name: TBL_STUDENTS

student_number	student_name	course	failed_subjects	gwa
(INT, PK)	(VARCHAR 50)	(VARCHAR 10)	(INT)	(DECIMAL 5,2)
1000111	Jin Kazama	BSCS	2	2.41
1000112	Kazuya Mishima	BSCPE	2	2.48
1000113	Paul Phoenix	BSIT	8	5.00
1000114	Bryan Fury	BSCPE	3	2.73
1000115	Steve Fox	BSIT	0	1.09
1000116	Lars Alexandersson	BSCS	1	2.29
1000117	Victor Chevalier	BSCS	0	1.27

EXAMPLE

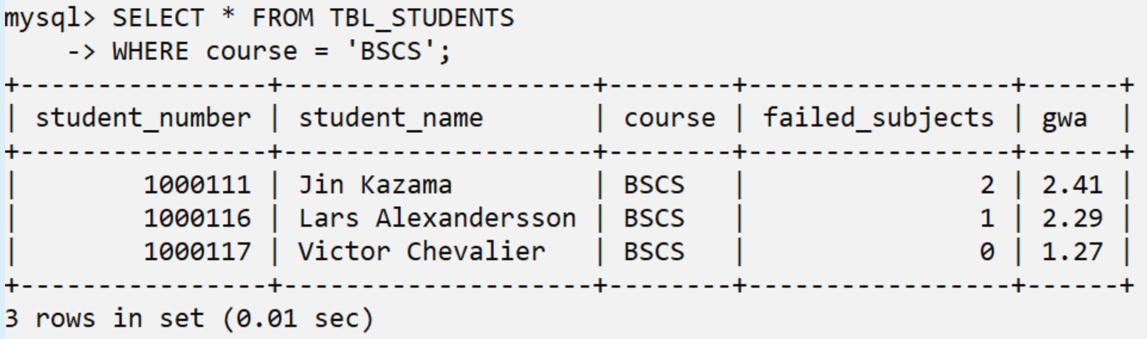
student_number (INT, PK)	student_name (VARCHAR 50)	course (VARCHAR 10)	failed_subjects (INT)	gwa (DECIMAL 5,2)
1000111	Jin Kazama	BSCS	2	2.41
1000112	Kazuya Mishima	BSCPE	2	2.48
1000113	Paul Phoenix	BSIT	8	5.00
1000114	Bryan Fury	BSCPE	3	2.73
1000115	Steve Fox	BSIT	0	1.09
1000116	Lars Alexandersson	BSCS	1	2.29
1000117	Victor Chevalier	BSCS	0	1.27

- SELECT * FROM TBL_STUDENTS;



```
mysql> SELECT * FROM TBL_STUDENTS;
+----------------+--------------------+--------+-----------------+------+
| student_number | student_name       | course | failed_subjects | gwa  |
+----------------+--------------------+--------+-----------------+------+
|        1000111 | Jin Kazama         | BSCS   |               2 | 2.41 |
|        1000112 | Kazuya Mishima     | BSCPE  |               2 | 2.48 |
|        1000113 | Paul Phoenix       | BSIT   |               8 | 5.00 |
|        1000114 | Bryan Fury         | BSCPE  |               3 | 2.73 |
|        1000115 | Steve Fox          | BSIT   |               0 | 1.09 |
|        1000116 | Lars Alexandersson | BSCS   |               1 | 2.29 |
|        1000117 | Victor Chevalier   | BSCS   |               0 | 1.27 |
+----------------+--------------------+--------+-----------------+------+
7 rows in set (0.01 sec)
```

- SELECT * FROM TBL_STUDENTS WHERE course = 'BSCS';

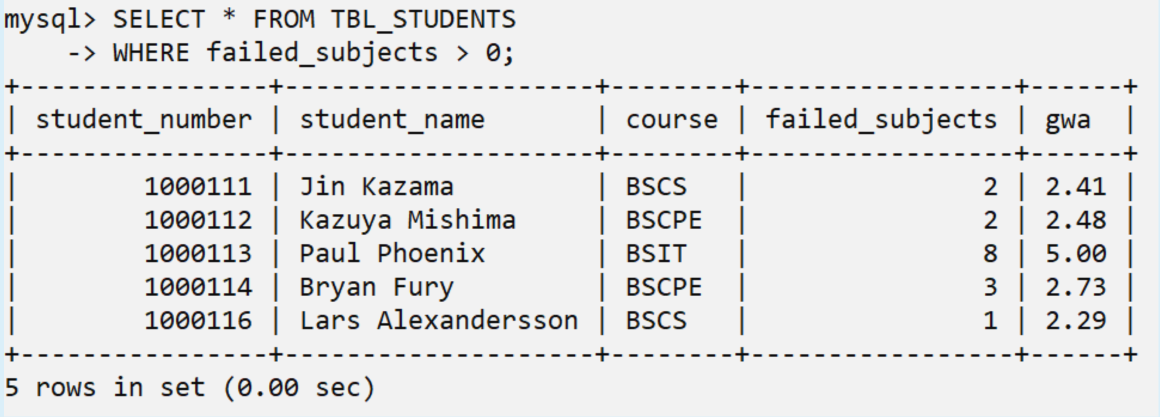


```
mysql> SELECT * FROM TBL_STUDENTS
    -> WHERE course = 'BSCS';
+----------------+--------------------+--------+-----------------+------+
| student_number | student_name       | course | failed_subjects | gwa  |
+----------------+--------------------+--------+-----------------+------+
|        1000111 | Jin Kazama         | BSCS   |               2 | 2.41 |
|        1000116 | Lars Alexandersson | BSCS   |               1 | 2.29 |
|        1000117 | Victor Chevalier   | BSCS   |               0 | 1.27 |
+----------------+--------------------+--------+-----------------+------+
3 rows in set (0.01 sec)
```

EXAMPLE

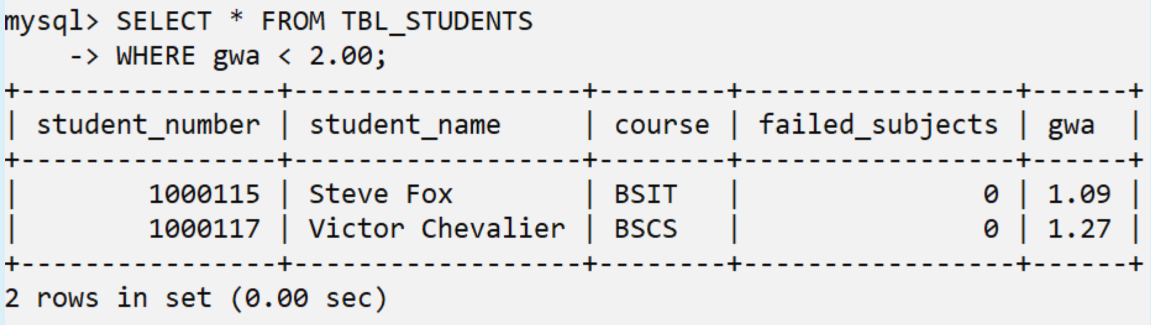
student_number (INT, PK)	student_name (VARCHAR 50)	course (VARCHAR 10)	failed_subjects (INT)	gwa (DECIMAL 5,2)
1000111	Jin Kazama	BSCS	2	2.41
1000112	Kazuya Mishima	BSCPE	2	2.48
1000113	Paul Phoenix	BSIT	8	5.00
1000114	Bryan Fury	BSCPE	3	2.73
1000115	Steve Fox	BSIT	0	1.09
1000116	Lars Alexandersson	BSCS	1	2.29
1000117	Victor Chevalier	BSCS	0	1.27

- SELECT * FROM TBL_STUDENTS WHERE failed_subjects > 0;



```
mysql> SELECT * FROM TBL_STUDENTS
    -> WHERE failed_subjects > 0;
+----------------+--------------------+--------+-----------------+------+
| student_number | student_name       | course | failed_subjects | gwa  |
+----------------+--------------------+--------+-----------------+------+
|        1000111 | Jin Kazama         | BSCS   |               2 | 2.41 |
|        1000112 | Kazuya Mishima     | BSCPE  |               2 | 2.48 |
|        1000113 | Paul Phoenix       | BSIT   |               8 | 5.00 |
|        1000114 | Bryan Fury         | BSCPE  |               3 | 2.73 |
|        1000116 | Lars Alexandersson | BSCS   |               1 | 2.29 |
+----------------+--------------------+--------+-----------------+------+
5 rows in set (0.00 sec)
```

- SELECT * FROM TBL_STUDENTS WHERE gwa < 2.00;

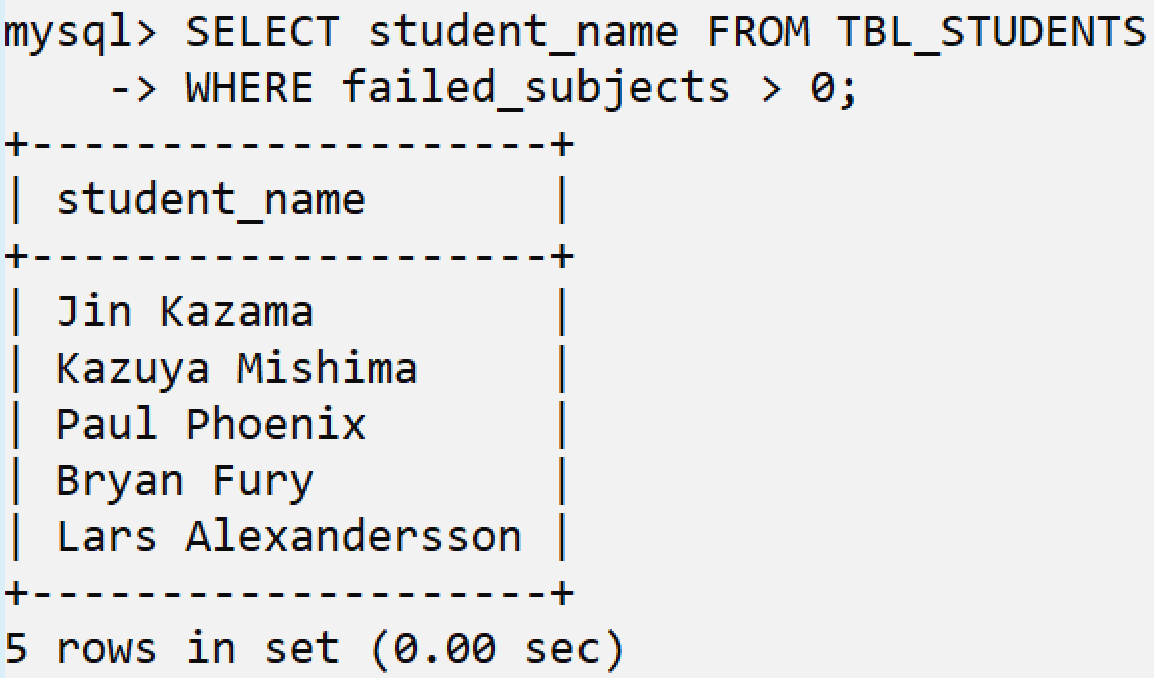


```
mysql> SELECT * FROM TBL_STUDENTS
    -> WHERE gwa < 2.00;
+----------------+------------------+--------+-----------------+------+
| student_number | student_name     | course | failed_subjects | gwa  |
+----------------+------------------+--------+-----------------+------+
|        1000115 | Steve Fox        | BSIT   |               0 | 1.09 |
|        1000117 | Victor Chevalier | BSCS   |               0 | 1.27 |
+----------------+------------------+--------+-----------------+------+
2 rows in set (0.00 sec)
```

EXAMPLE

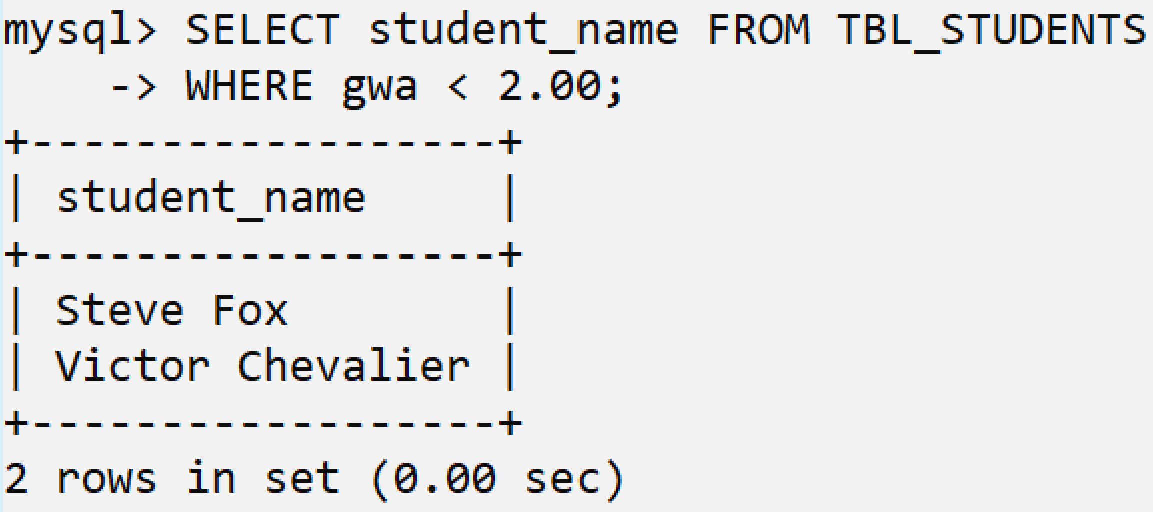
student_number	student_name	course	failed_subjects	gwa
(INT, PK)	(VARCHAR 50)	(VARCHAR 10)	(INT)	(DECIMAL 5,2)
1000111	Jin Kazama	BSCS	2	2.41
1000112	Kazuya Mishima	BSCPE	2	2.48
1000113	Paul Phoenix	BSIT	8	5.00
1000114	Bryan Fury	BSCPE	3	2.73
1000115	Steve Fox	BSIT	0	1.09
1000116	Lars Alexandersson	BSCS	1	2.29
1000117	Victor Chevalier	BSCS	0	1.27

- SELECT student_name
  FROM TBL_STUDENTS
  WHERE failed_subjects > 0;



```
mysql> SELECT student_name FROM TBL_STUDENTS
    -> WHERE failed_subjects > 0;
+--------------------+
| student_name       |
+--------------------+
| Jin Kazama         |
| Kazuya Mishima     |
| Paul Phoenix       |
| Bryan Fury         |
| Lars Alexandersson |
+--------------------+
5 rows in set (0.00 sec)
```

- SELECT student_name
  FROM TBL_STUDENTS
  WHERE gwa < 2.00;

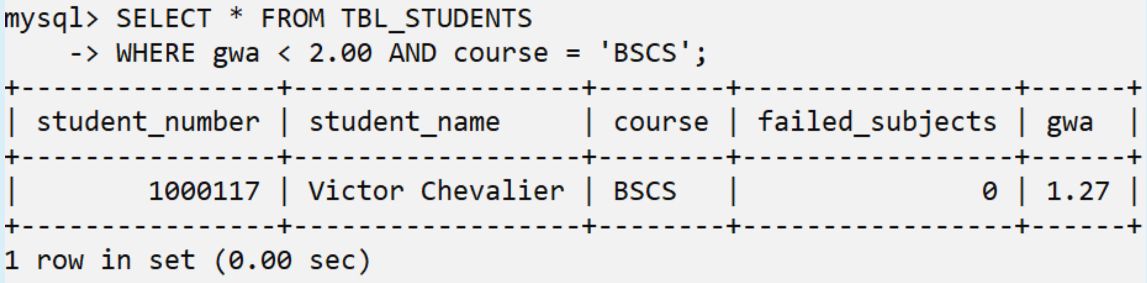


```
mysql> SELECT student_name FROM TBL_STUDENTS
    -> WHERE gwa < 2.00;
+------------------+
| student_name     |
+------------------+
| Steve Fox        |
| Victor Chevalier |
+------------------+
2 rows in set (0.00 sec)
```

EXAMPLE

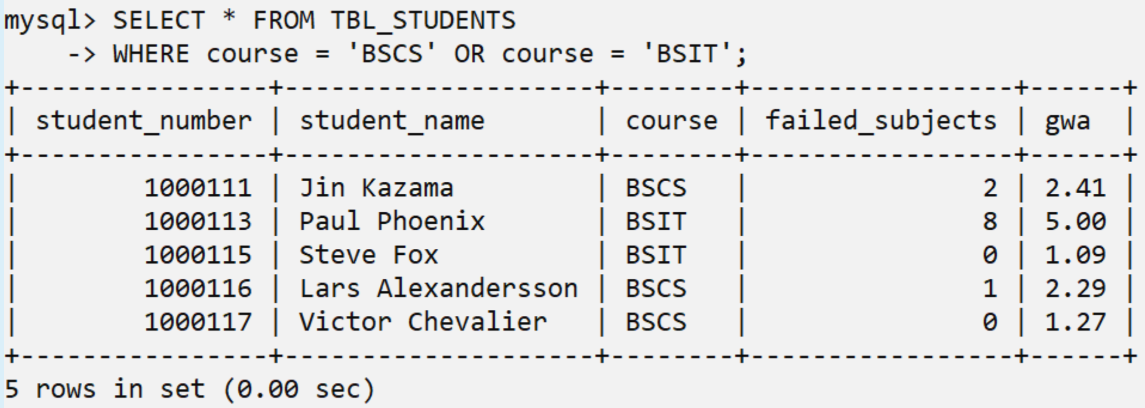
student_number (INT, PK)	student_name (VARCHAR 50)	course (VARCHAR 10)	failed_subjects (INT)	gwa (DECIMAL 5,2)
1000111	Jin Kazama	BSCS	2	2.41
1000112	Kazuya Mishima	BSCPE	2	2.48
1000113	Paul Phoenix	BSIT	8	5.00
1000114	Bryan Fury	BSCPE	3	2.73
1000115	Steve Fox	BSIT	0	1.09
1000116	Lars Alexandersson	BSCS	1	2.29
1000117	Victor Chevalier	BSCS	0	1.27

- SELECT * FROM TBL_STUDENTS
  WHERE gwa < 2.00
  AND course = 'BSCS';



```
mysql> SELECT * FROM TBL_STUDENTS
    -> WHERE gwa < 2.00 AND course = 'BSCS';
+----------------+------------------+--------+-----------------+------+
| student_number | student_name     | course | failed_subjects | gwa  |
+----------------+------------------+--------+-----------------+------+
|        1000117 | Victor Chevalier | BSCS   |               0 | 1.27 |
+----------------+------------------+--------+-----------------+------+
1 row in set (0.00 sec)
```

- SELECT * FROM TBL_STUDENTS
  WHERE course = 'BSCS'
  OR course = 'BSIT';



```
mysql> SELECT * FROM TBL_STUDENTS
    -> WHERE course = 'BSCS' OR course = 'BSIT';
+----------------+--------------------+--------+-----------------+------+
| student_number | student_name       | course | failed_subjects | gwa  |
+----------------+--------------------+--------+-----------------+------+
|        1000111 | Jin Kazama         | BSCS   |               2 | 2.41 |
|        1000113 | Paul Phoenix       | BSIT   |               8 | 5.00 |
|        1000115 | Steve Fox          | BSIT   |               0 | 1.09 |
|        1000116 | Lars Alexandersson | BSCS   |               1 | 2.29 |
|        1000117 | Victor Chevalier   | BSCS   |               0 | 1.27 |
+----------------+--------------------+--------+-----------------+------+
5 rows in set (0.00 sec)
```

QUESTION!

What is the output?

student_number (INT, PK)	student_name (VARCHAR 50)	course (VARCHAR 10)	failed_subjects (INT)	gwa (DECIMAL 5,2)
1000111	Jin Kazama	BSCS	2	2.41
1000112	Kazuya Mishima	BSCPE	2	2.48
1000113	Paul Phoenix	BSIT	8	5.00
1000114	Bryan Fury	BSCPE	3	2.73
1000115	Steve Fox	BSIT	0	1.09
1000116	Lars Alexandersson	BSCS	1	2.29
1000117	Victor Chevalier	BSCS	0	1.27

```
SELECT failed_subjects FROM TBL_STUDENTS
WHERE gwa > 2.00;
```

QUESTION!

What is the output?

student_number (INT, PK)	student_name (VARCHAR 50)	course (VARCHAR 10)	failed_subjects (INT)	gwa (DECIMAL 5,2)
1000111	Jin Kazama	BSCS	2	2.41
1000112	Kazuya Mishima	BSCPE	2	2.48
1000113	Paul Phoenix	BSIT	8	5.00
1000114	Bryan Fury	BSCPE	3	2.73
1000115	Steve Fox	BSIT	0	1.09
1000116	Lars Alexandersson	BSCS	1	2.29
1000117	Victor Chevalier	BSCS	0	1.27

-
  ```
  SELECT failed_subjects FROM TBL_STUDENTS
  WHERE course <> BSCPE;
  ```

QUESTION!

What is the output?

student_number (INT, PK)	student_name (VARCHAR 50)	course (VARCHAR 10)	failed_subjects (INT)	gwa (DECIMAL 5,2)
1000111	Jin Kazama	BSCS	2	2.41
1000112	Kazuya Mishima	BSCPE	2	2.48
1000113	Paul Phoenix	BSIT	8	5.00
1000114	Bryan Fury	BSCPE	3	2.73
1000115	Steve Fox	BSIT	0	1.09
1000116	Lars Alexandersson	BSCS	1	2.29
1000117	Victor Chevalier	BSCS	0	1.27

```
SELECT student_name FROM TBL_STUDENTS
WHERE failed_subjects < 1;
```

QUESTION!

What is the output?

student_number (INT, PK)	student_name (VARCHAR 50)	course (VARCHAR 10)	failed_subjects (INT)	gwa (DECIMAL 5,2)
1000111	Jin Kazama	BSCS	2	2.41
1000112	Kazuya Mishima	BSCPE	2	2.48
1000113	Paul Phoenix	BSIT	8	5.00
1000114	Bryan Fury	BSCPE	3	2.73
1000115	Steve Fox	BSIT	0	1.09
1000116	Lars Alexandersson	BSCS	1	2.29
1000117	Victor Chevalier	BSCS	0	1.27

```
SELECT student_name FROM TBL_STUDENTS
WHERE course = 'BSIT' AND failed_subjects = 0;
```

QUESTION!

What is the output?

student_number	student_name	course	failed_subjects	gwa
(INT, PK)	(VARCHAR 50)	(VARCHAR 10)	(INT)	(DECIMAL 5,2)
1000111	Jin Kazama	BSCS	2	2.41
1000112	Kazuya Mishima	BSCPE	2	2.48
1000113	Paul Phoenix	BSIT	8	5.00
1000114	Bryan Fury	BSCPE	3	2.73
1000115	Steve Fox	BSIT	0	1.09
1000116	Lars Alexandersson	BSCS	1	2.29
1000117	Victor Chevalier	BSCS	0	1.27

```
SELECT gwa FROM TBL_STUDENTS
WHERE student_name = 'Steve Fox' AND course = 'BSIT'
```

SELECT STATEMENT
+ AGGREGATE FUNCTIONS

WHAT IS AGGREGATE FUNCTION?

- performs a calculation on a set of values, and returns a single value
- is used within the `SELECT` statement
- ignores `null` values, except for `COUNT`

EXAMPLE

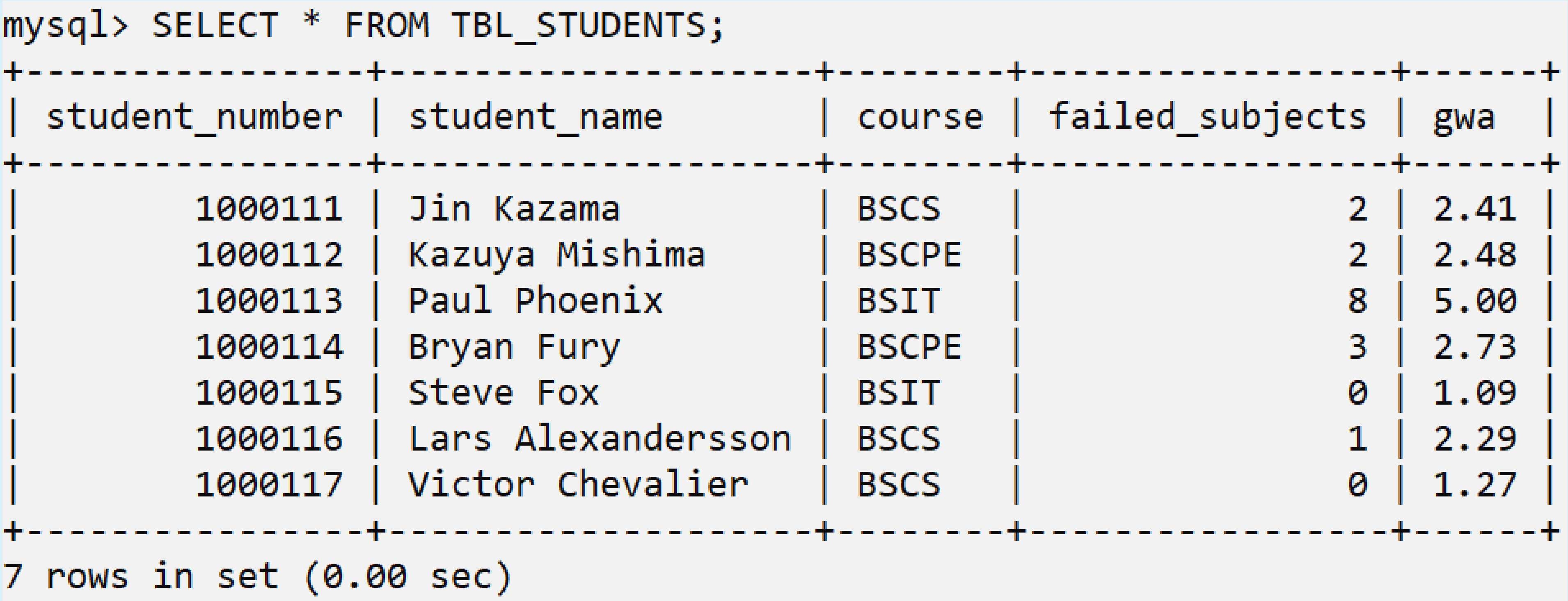
Database name: DB_COLLEGE

Table name: TBL_STUDENTS

student_number	student_name	course	failed_subjects	gwa
(INT, PK)	(VARCHAR 50)	(VARCHAR 10)	(INT)	(DECIMAL 5,2)
1000111	Jin Kazama	BSCS	2	2.41
1000112	Kazuya Mishima	BSCPE	2	2.48
1000113	Paul Phoenix	BSIT	8	5.00
1000114	Bryan Fury	BSCPE	3	2.73
1000115	Steve Fox	BSIT	0	1.09
1000116	Lars Alexandersson	BSCS	1	2.29
1000117	Victor Chevalier	BSCS	0	1.27

What is the output?

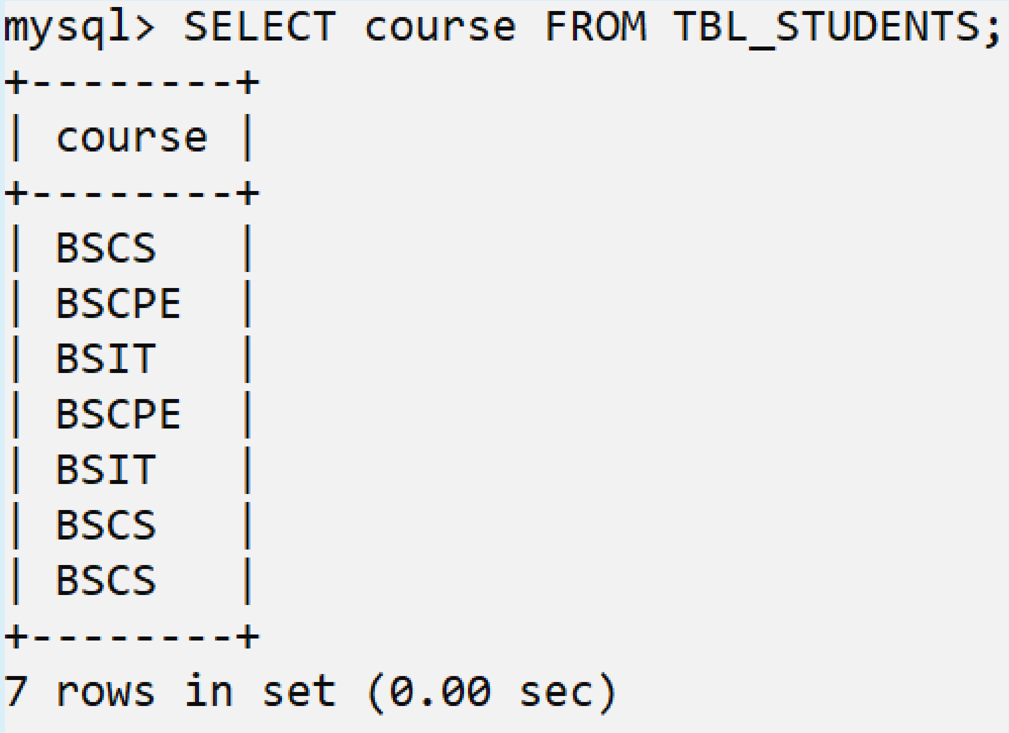
SELECT * FROM TBL_STUDENTS;



```
mysql> SELECT * FROM TBL_STUDENTS;
+----------------+--------------------+--------+-----------------+------+
| student_number | student_name       | course | failed_subjects | gwa  |
+----------------+--------------------+--------+-----------------+------+
|        1000111 | Jin Kazama         | BSCS   |               2 | 2.41 |
|        1000112 | Kazuya Mishima     | BSCPE  |               2 | 2.48 |
|        1000113 | Paul Phoenix       | BSIT   |               8 | 5.00 |
|        1000114 | Bryan Fury         | BSCPE  |               3 | 2.73 |
|        1000115 | Steve Fox          | BSIT   |               0 | 1.09 |
|        1000116 | Lars Alexandersson | BSCS   |               1 | 2.29 |
|        1000117 | Victor Chevalier   | BSCS   |               0 | 1.27 |
+----------------+--------------------+--------+-----------------+------+
7 rows in set (0.00 sec)
```

What is the output?

```
SELECT course FROM TBL_STUDENTS;
```



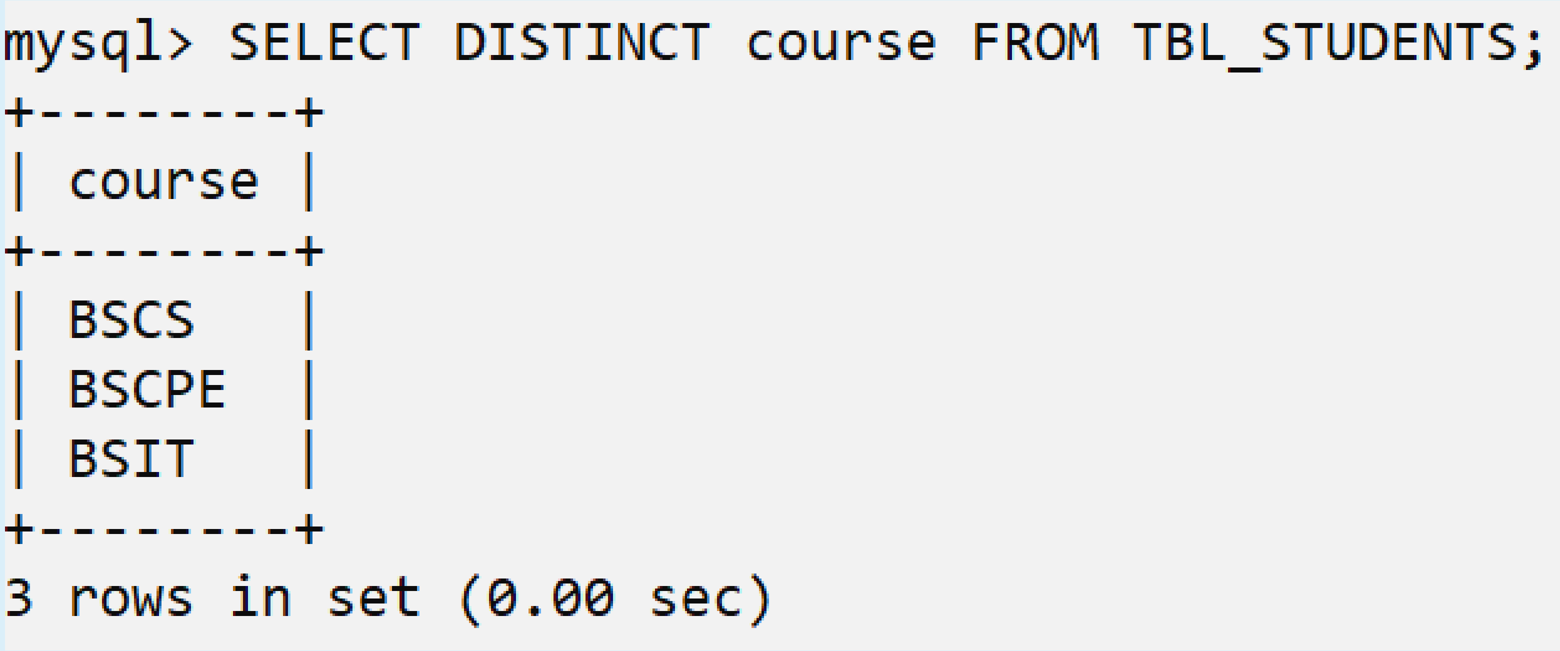
```
mysql> SELECT course FROM TBL_STUDENTS;
+--------+
| course |
+--------+
| BSCS   |
| BSCPE  |
| BSIT   |
| BSCPE  |
| BSIT   |
| BSCS   |
| BSCS   |
+--------+
7 rows in set (0.00 sec)
```

DISTINCT

- retrieves only unique values from a specific column
- Syntax: `SELECT DISTINCT pangalan_ng_column1 FROM pangalan_ng_table;`
- Example:

What is the output?

```
SELECT DISTINCT course FROM TBL_STUDENTS;
```

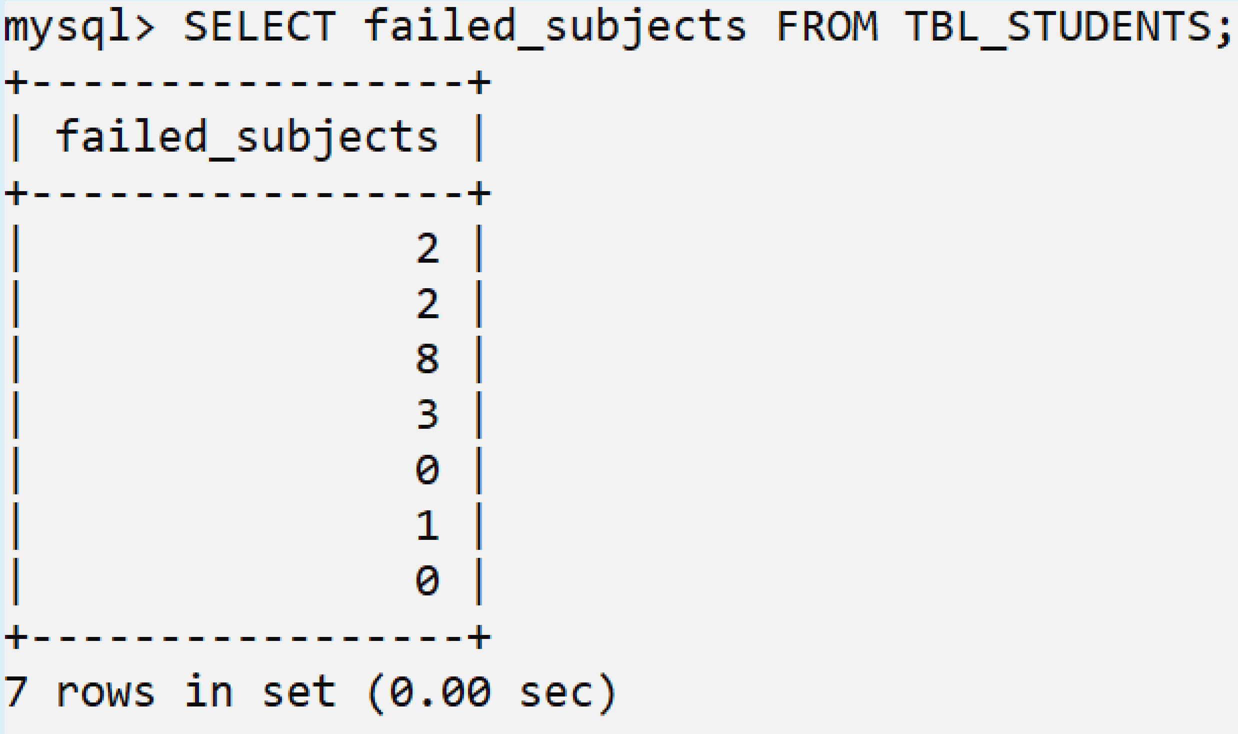


```
mysql> SELECT DISTINCT course FROM TBL_STUDENTS;
+--------+
| course |
+--------+
| BSCS   |
| BSCPE  |
| BSIT   |
+--------+
3 rows in set (0.00 sec)
```

QUESTION!

What is the output?

- `SELECT failed_subjects FROM TBL_STUDENTS;`
- Answer:

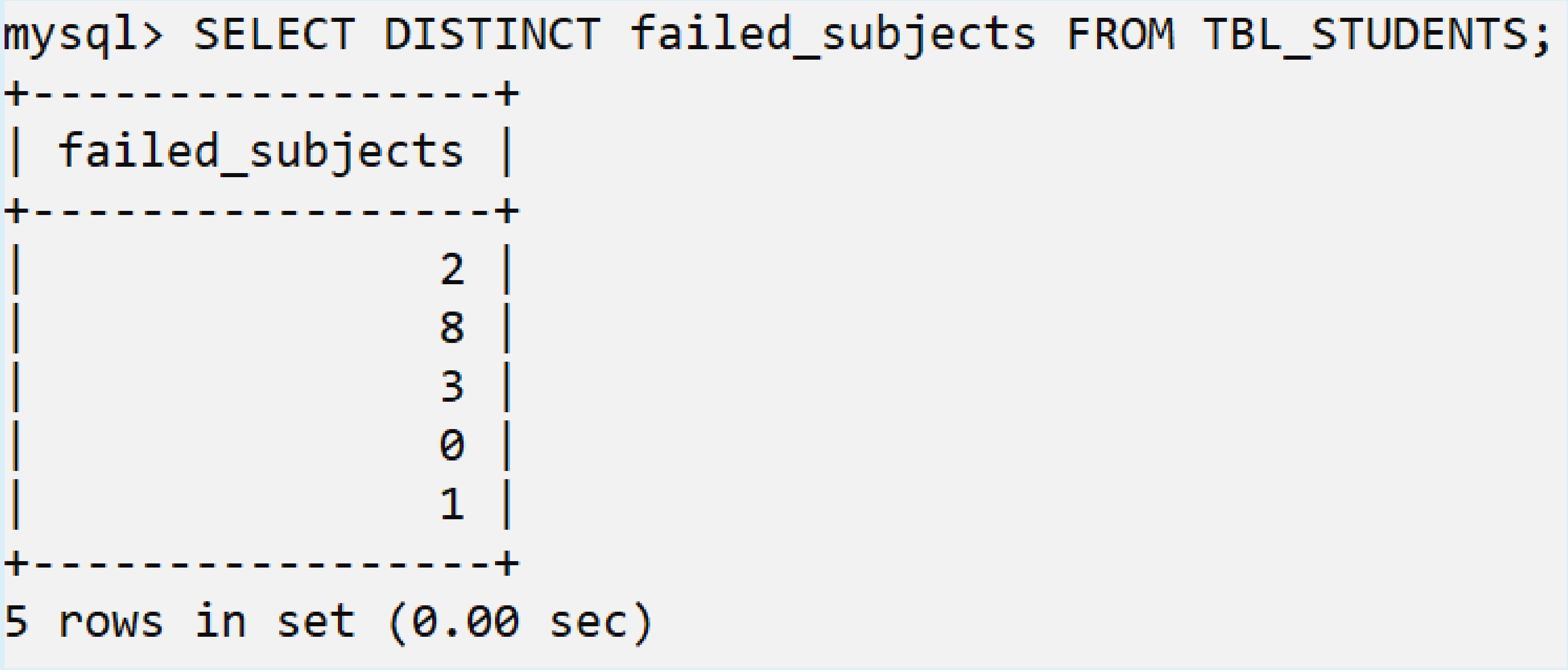


```
mysql> SELECT failed_subjects FROM TBL_STUDENTS;
+-----------------+
| failed_subjects |
+-----------------+
|               2 |
|               2 |
|               8 |
|               3 |
|               0 |
|               1 |
|               0 |
+-----------------+
7 rows in set (0.00 sec)
```

QUESTION!

What is the output?

- SELECT DISTINCT failed_subjects FROM TBL_STUDENTS;
- Answer:



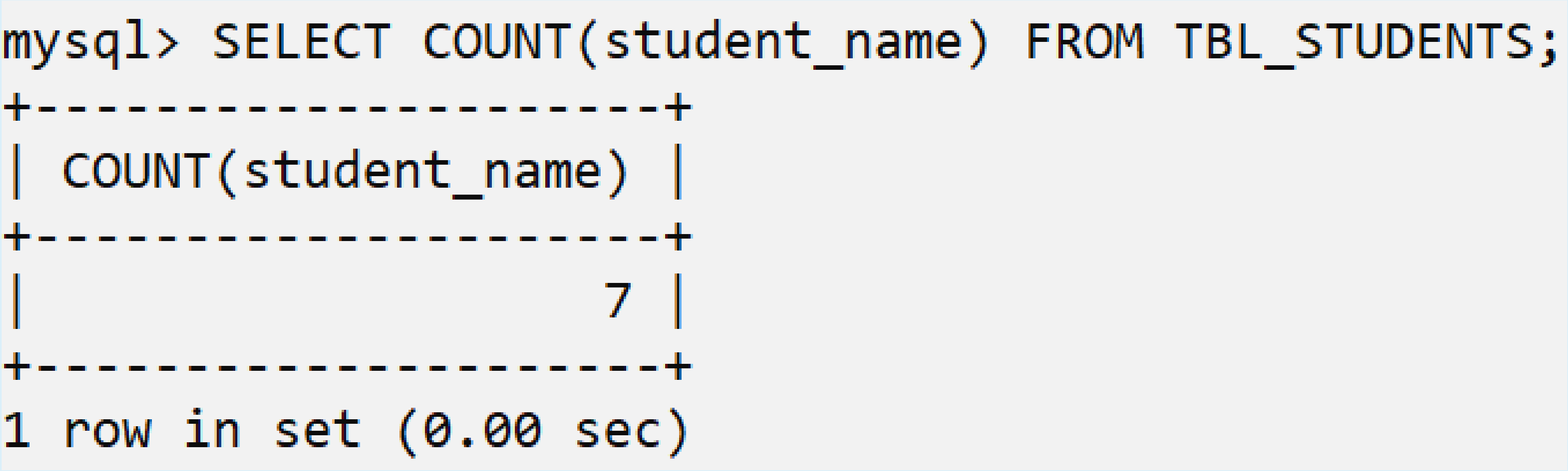
```
mysql> SELECT DISTINCT failed_subjects FROM TBL_STUDENTS;
+-----------------+
| failed_subjects |
+-----------------+
|               2 |
|               8 |
|               3 |
|               0 |
|               1 |
+-----------------+
5 rows in set (0.00 sec)
```

COUNT

- gets the number of rows for a particular group in the table
- Syntax: `SELECT COUNT(pangalan_ng_column) FROM pangalan_ng_table;`
- Example:

What is the output?

```
SELECT COUNT(student_name) FROM TBL_STUDENTS;
```

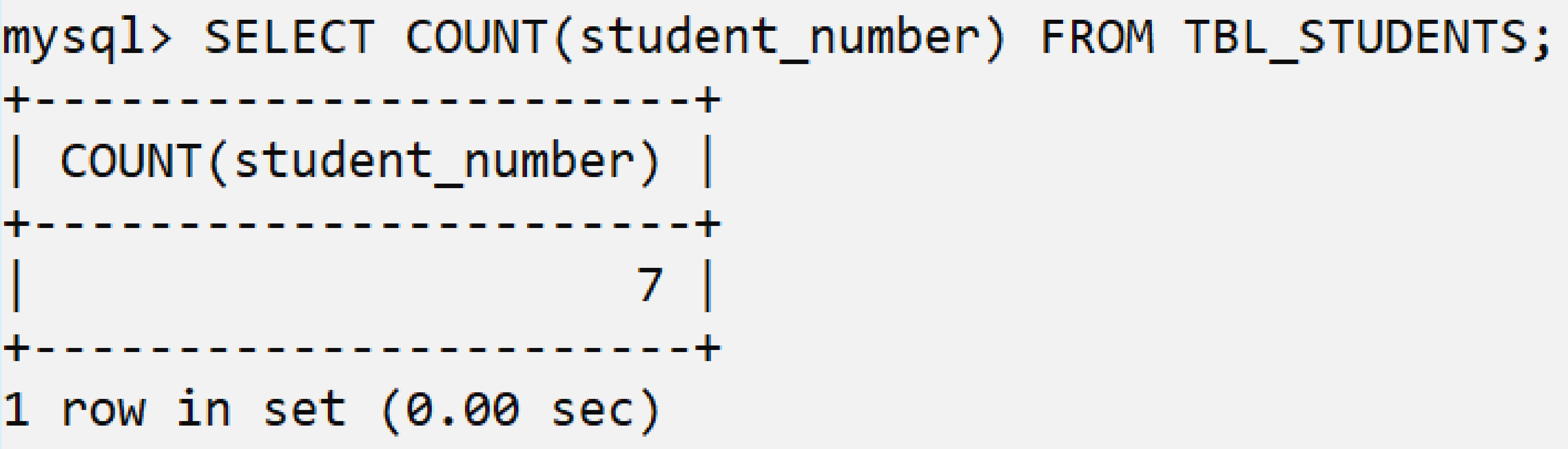


```
mysql> SELECT COUNT(student_name) FROM TBL_STUDENTS;
+---------------------+
| COUNT(student_name) |
+---------------------+
|                   7 |
+---------------------+
1 row in set (0.00 sec)
```

QUESTION!

What is the output?

- SELECT COUNT(student_number) FROM TBL_STUDENTS;
- Answer:



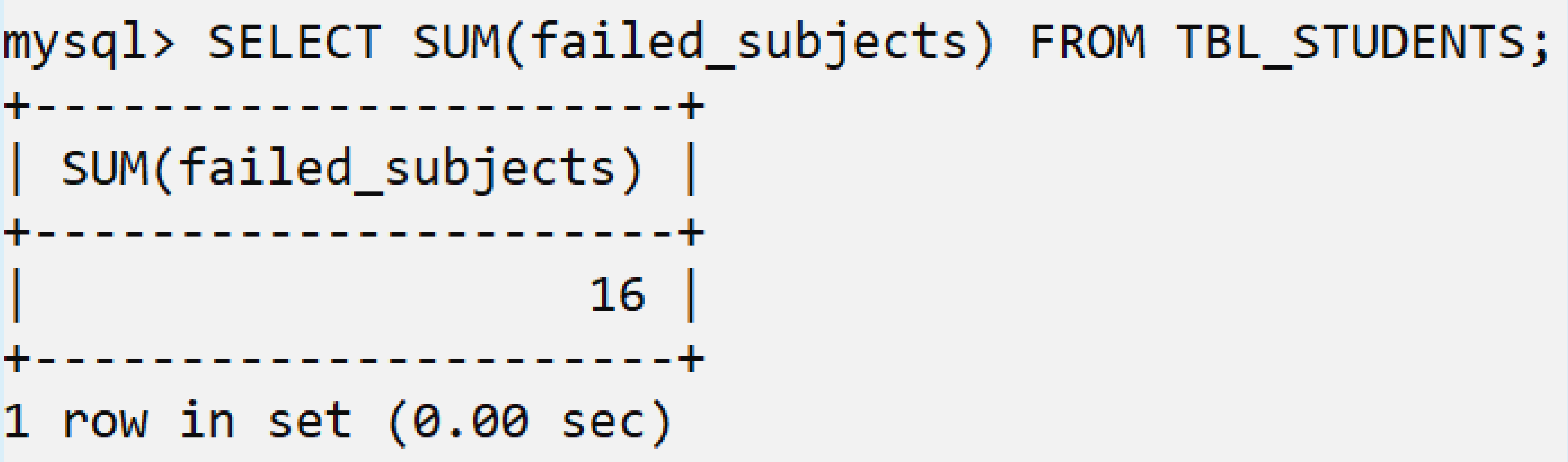
```
mysql> SELECT COUNT(student_number) FROM TBL_STUDENTS;
+-----------------------+
| COUNT(student_number) |
+-----------------------+
|                     7 |
+-----------------------+
1 row in set (0.00 sec)
```

SUM

- calculates the total sum of a numeric column
- Syntax: `SELECT SUM(pangalan_ng_column) FROM pangalan_ng_table;`
- Example:

What is the output?

```
SELECT SUM(failed_subjects) FROM TBL_STUDENTS;
```

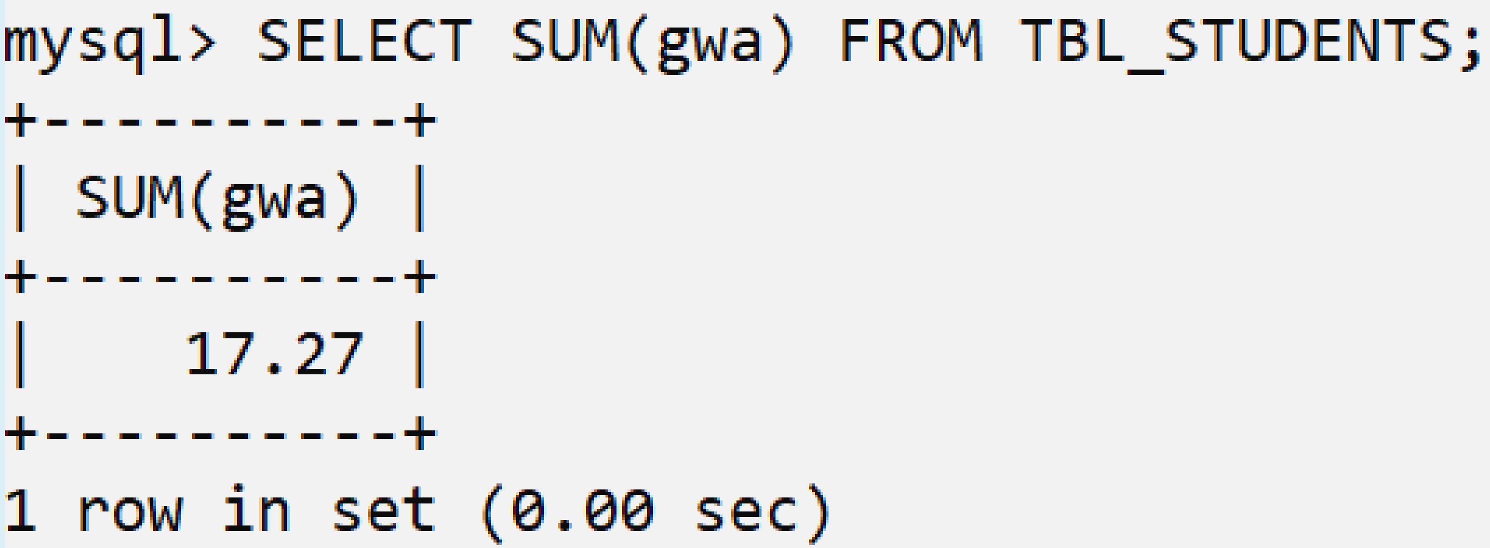


```
mysql> SELECT SUM(failed_subjects) FROM TBL_STUDENTS;
+----------------------+
| SUM(failed_subjects) |
+----------------------+
|                   16 |
+----------------------+
1 row in set (0.00 sec)
```

QUESTION!

What is the output?

- SELECT SUM(gwa) FROM TBL_STUDENTS;
- Answer:



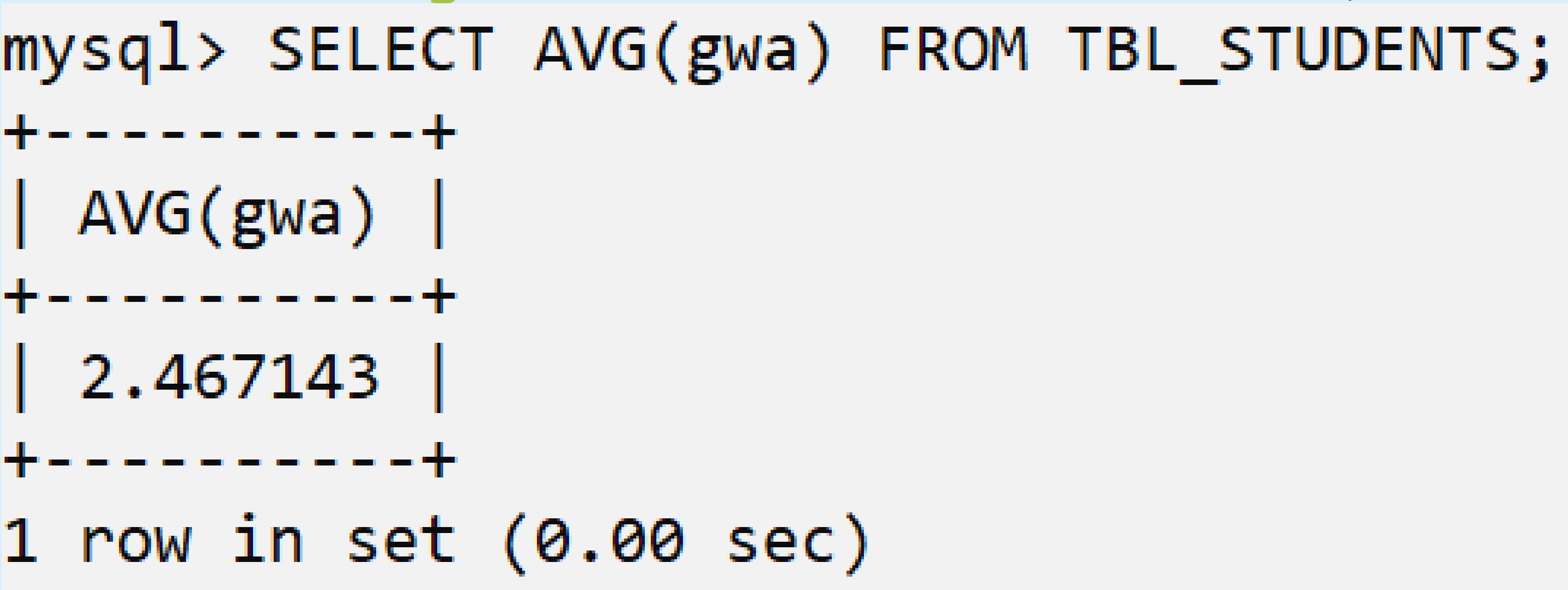
```
mysql> SELECT SUM(gwa) FROM TBL_STUDENTS;
+----------+
| SUM(gwa) |
+----------+
|    17.27 |
+----------+
1 row in set (0.00 sec)
```

AVERAGE

- calculates the average value of a numeric column
- Syntax: `SELECT AVG(pangalan_ng_column) FROM pangalan_ng_table;`
- Example:

What is the output?

```
SELECT AVG(gwa) FROM TBL_STUDENTS;
```

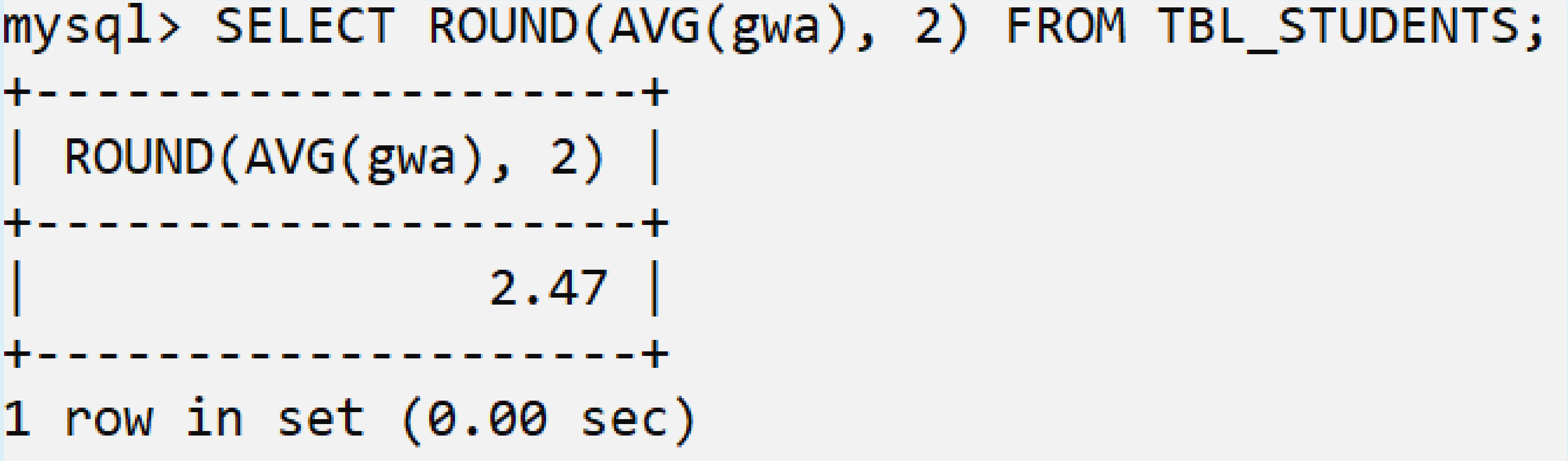


```
mysql> SELECT AVG(gwa) FROM TBL_STUDENTS;
+----------+
| AVG(gwa) |
+----------+
| 2.467143 |
+----------+
1 row in set (0.00 sec)
```

QUESTION!

- What if I want the AVG result to be two decimal places only?
- Answer:

```
SELECT ROUND(AVG(gwa), 2) FROM TBL_STUDENTS;
```



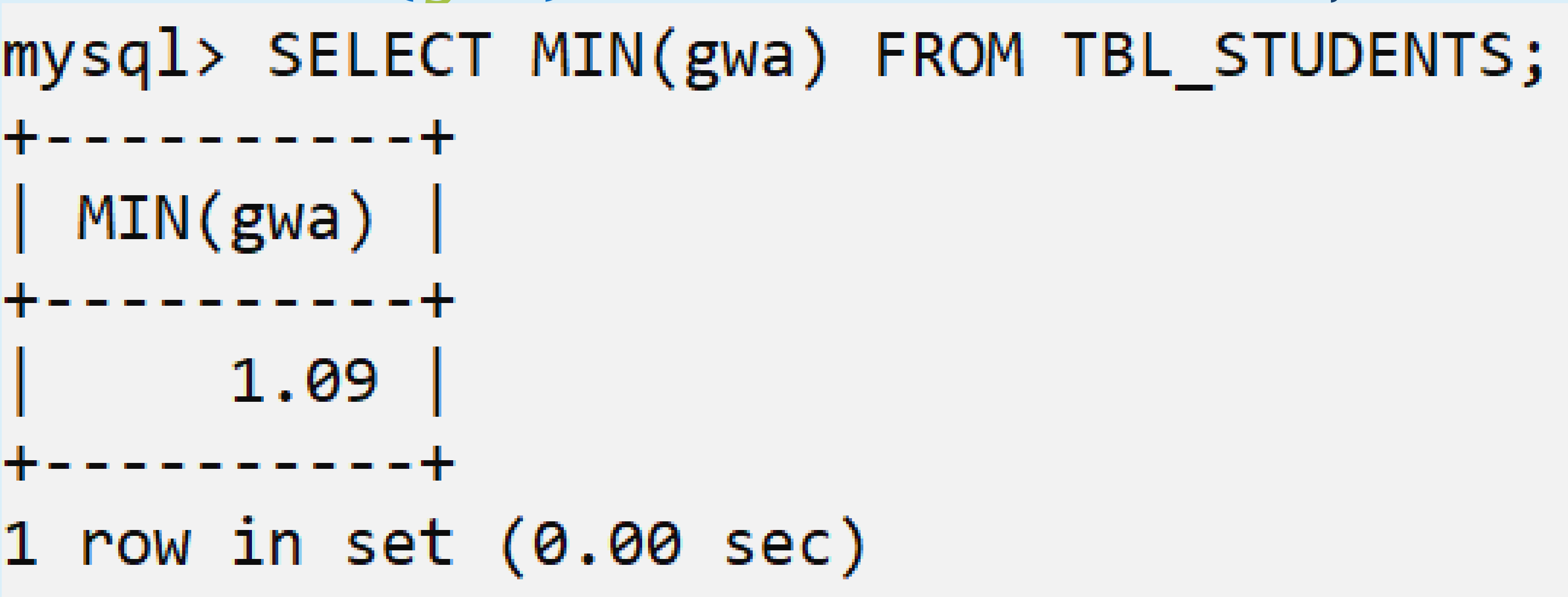
```
mysql> SELECT ROUND(AVG(gwa), 2) FROM TBL_STUDENTS;
+--------------------+
| ROUND(AVG(gwa), 2) |
+--------------------+
|               2.47 |
+--------------------+
1 row in set (0.00 sec)
```

MINIMUM

- returns the smallest value of the selected column
- Syntax: `SELECT MIN(pangalan_ng_column) FROM pangalan_ng_table;`
- Example:

What is the output?

```
SELECT MIN(gwa) FROM TBL_STUDENTS;
```



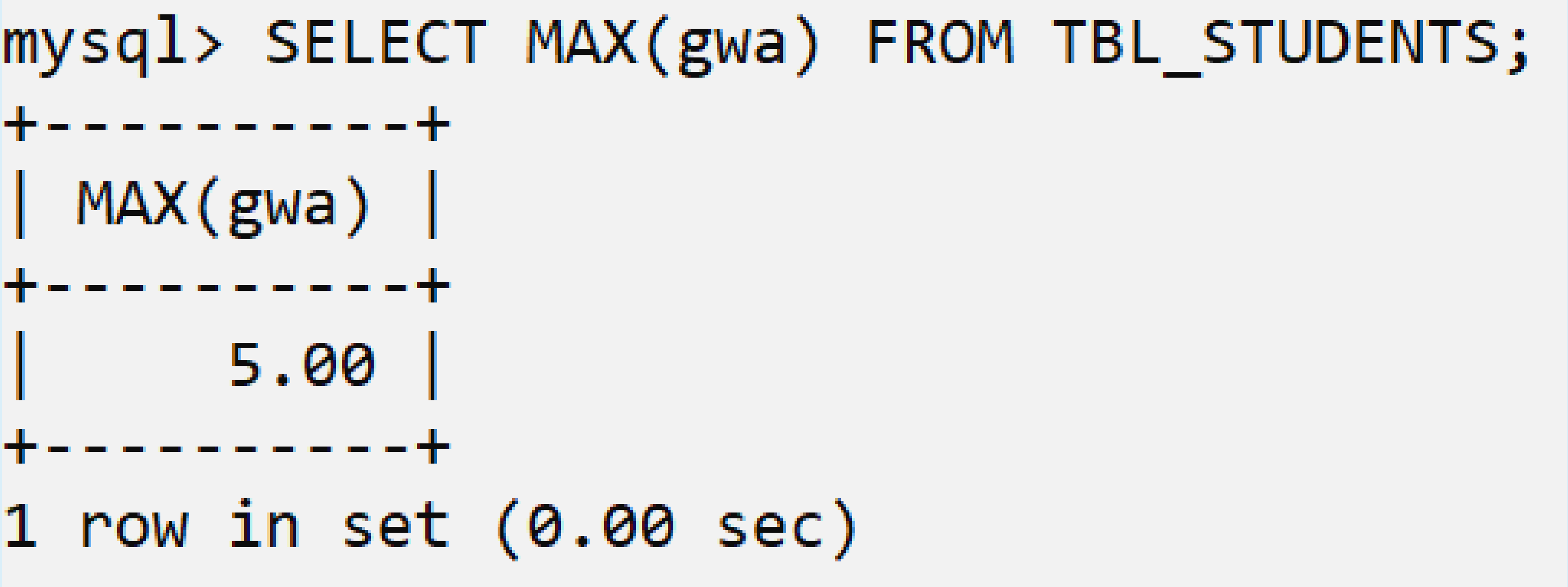
```
mysql> SELECT MIN(gwa) FROM TBL_STUDENTS;
+----------+
| MIN(gwa) |
+----------+
|     1.09 |
+----------+
1 row in set (0.00 sec)
```

MAXIMUM

- returns the largest value of the selected column
- Syntax: `SELECT Max(pangalan_ng_column) FROM pangalan_ng_table;`
- Example:

What is the output?

```
SELECT MAX(gwa) FROM TBL_STUDENTS;
```



```
mysql> SELECT MAX(gwa) FROM TBL_STUDENTS;
+----------+
| MAX(gwa) |
+----------+
|     5.00 |
+----------+
1 row in set (0.00 sec)
```

DISCLAIMER:

You may watch other YouTube tutorials
or read other programming articles
so you can help yourself <u>self-study</u>
about database programming.

Translation sa Tagalog:
Hindi lahat ng dapat mong matututunan
ay nasa loob lamang ng silid-aralan.

ACTIVITY

[INFOMNGT] ACTIVITY

- Answer the eLMS activity:
  - *MIDTERMS - Laboratory Exercise 01*
  - *MIDTERMS - Laboratory Exercise 02*