

1. 5 pts. Create and populate the students table

The screenshot shows the phpMyAdmin interface for a database named 'db1'. The 'students' table is selected, and the 'Structure' tab is active. The table has the following structure:

fieldname	lastname	year	gpa
RRWXJ	I	SOPHOMORE	2.01
HUWPNHTFHDWPWBBSOQLX	LIPGDIJB	FRESHMAN	3.50
JCYWOAWJ	GNFTQLBIIQ	SENIOR	3.52
AHBJJJO	AQVEGA	FRESHMAN	2.57
KDJKNXNBPZDHJUSUMJR	UIAPSIBQZKOW	JUNIOR	2.83
ZWFSBYPVHQSW	UOMDS	FRESHMAN	3.39

The 'Console' tab is also active, showing the SQL query: `SELECT * FROM `students``. The query results show 25 rows of data, with the first 6 rows visible in the 'Structure' tab. The 'Console' tab shows the full list of 25 rows.

Recent Favorites

New db1 New student

db2 db3 information mysql performance phpmyadm sys

Server: localhost:3306 » Database: db1 » Table: students

Browse Structure SQL Search Insert Export More

Showing rows 0 - 24 (1082577 total, Query took 0.0009 seconds.)

SELECT * FROM `students`

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

1 > >> | Number of rows: 25 | Filter rows: Search this table

+ Options

fieldname lastname year gpa

RRWXJ I SOPHOMORE 2.01

HUWPNHTFHDWPWBBSOQLX LIPGDIJB FRESHMAN 3.50

JCYWOAWJ GNFTQLBIIQ SENIOR 3.52

AHBJJJO AQVEGA FRESHMAN 2.57

KDJKNXNBPZDHJUSUMJR UIAPSIBQZKOW JUNIOR 2.83

ZWFSBYPVHQSW UOMDS FRESHMAN 3.39

Browse Structure SQL Search Insert Export More

HUWPNHTFHDWPWBBSOQLX LIPGDIJB FRESHMAN 3.50

JCYWOAWJ GNFTQLBIIQ SENIOR 3.52

AHBJJJO AQVEGA FRESHMAN 2.57

KDJKNXNBPZDHJUSUMJR UIAPSIBQZKOW JUNIOR 2.83

WEYBLLWAZWFSBYPVHQSW UOMDS FRESHMAN 3.39

ILBXFJINLQHRDV SQNQNVJLDM SOPHOMORE 2.69

PSVNO EBYZBTVMQJRCG FRESHMAN 2.25

SVJTXWSMJOPFJRMXZABQ BTQRXFKCCKMQTCUJRTJ SENIOR 3.23

QJKMUKMHMNUMTC FIXJZWJM SOPHOMORE 2.05

EOXSDWZXZYG GANDEIZQLFLMNIXHOM SOPHOMORE 3.83

QQU JUWW SENIOR 3.14

SIPRGZEEACUSD ZWJ SOPHOMORE 2.02

AXSUFKMNNDONCNHBM QGYIPWGXGIMUBXFMK SENIOR 3.52

MUOGOTGW RTCXLDYNEANFHLXTX FRESHMAN 2.54

U DHRBCOSEDEQTRLNRPMU SENIOR 3.30

QMHUQZRECSBPA NFD FRESHMAN 3.30

Console

2. 10 pts. Performance measurement using explain (I/O rows) and mysqlslap (time)

a. 2 pts. Execute the following explain SQL command. What does the rows metric represent with the explain command?

```
mysql> explain select count(*) from students where firstname = 'APU'\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: students
   partitions: NULL
         type: ALL
possible_keys: NULL
          key: NULL
        key_len: NULL
         ref: NULL
        rows: 1082577
   filtered: 10.00
   Extra: Using where
1 row in set, 1 warning (0.00 sec)

mysql> █
```

The rows shows the estimate number of rows examined

b. 2 pts. Execute the mysqlslap UNIX command. Show the screen shot displaying the time required to execute.

```
yafet@yafet-VirtualBox:~$ mysqlslap --create-schema=db1 --concurrency=10 --iterations=10 --
user=root --password=██████████ --query="select count(*) from db1.students where firstname = 'AP
U'"
mysqlslap: [Warning] Using a password on the command line interface can be insecure.
Benchmark
  Average number of seconds to run all queries: 13.019 seconds
  Minimum number of seconds to run all queries: 9.763 seconds
  Maximum number of seconds to run all queries: 24.738 seconds
  Number of clients running queries: 10
  Average number of queries per client: 1

yafet@yafet-VirtualBox:~$ █
```

c. 2 pts. Create the index as shown below and repeat the explain and mysqlslap command. Notice the I/O rows and time metrics.

```
mysql> create index fname on students (firstname);
Query OK, 0 rows affected (10.70 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> explain select count(*) from students where firstname = 'APU'\G
***** 1. row *****
      id: 1
    select_type: SIMPLE
        table: students
    partitions: NULL
         type: ref
possible_keys: fname
          key: fname
        key_len: 122
         ref: const
         rows: 4
    filtered: 100.00
      Extra: Using index
1 row in set, 1 warning (0.00 sec)

mysql> █
```

The rows value got way smaller after creating the index

```
yafet@yafet-VirtualBox:~$ mysqlslap --create-schema=db1 --concurrency=10 --iterations=10 --
user=root --password=██████ --query="select count(*) from db1.students where firstname = 'AP
U'"
mysqlslap: [Warning] Using a password on the command line interface can be insecure.
Benchmark
  Average number of seconds to run all queries: 0.014 seconds
  Minimum number of seconds to run all queries: 0.007 seconds
  Maximum number of seconds to run all queries: 0.031 seconds
  Number of clients running queries: 10
  Average number of queries per client: 1

yafet@yafet-VirtualBox:~$ █
```

The timer improved greatly using the index

d. 2 pts. Repeat the above analysis with comparing the gpa values (i.e. use both > and =). In other words, perform both the explain & mysqlslap on the gpa queries (both > and =). This should be 4 screen shots. Then, repeat the same queries after adding an index. Finally, compare the first 4 screen shots with the second 4 screen shots. Note the differences and describe the outcome

>3.5 above 3.5 gpa

```
mysql> explain select count(*) from students where gpa > 3.5\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: students
  partitions: NULL
         type: ALL
possible_keys: NULL
          key: NULL
        key_len: NULL
         ref: NULL
        rows: 1082577
   filtered: 33.33
    Extra: Using where
1 row in set, 1 warning (0.00 sec)

mysql> █
```

```
yafet@yafet-VirtualBox:~$ mysqlslap --create-schema=db1 --concurrency=10 --iterations=10 --
user=root --password=██████ --query="select count(*) from db1.students where gpa > 3.5"
mysqlslap: [Warning] Using a password on the command line interface can be insecure.
Benchmark
  Average number of seconds to run all queries: 10.242 seconds
  Minimum number of seconds to run all queries: 9.291 seconds
  Maximum number of seconds to run all queries: 12.291 seconds
  Number of clients running queries: 10
  Average number of queries per client: 1

yafet@yafet-VirtualBox:~$ █
```

= 4.0 perfect gpa

```
mysql> explain select count(*) from students where gpa = 4.0\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: students
  partitions: NULL
         type: ALL
possible_keys: NULL
          key: NULL
        key_len: NULL
         ref: NULL
         rows: 1082577
   filtered: 10.00
    Extra: Using where
1 row in set, 1 warning (0.00 sec)

mysql> █
```

```
Average number of queries per client: 1

yafet@yafet-VirtualBox:~$ mysqlslap --create-schema=db1 --concurrency=10 --iterations=10 --
user=root --password=██████ --query="select count(*) from db1.students where gpa = 4.0"
mysqlslap: [Warning] Using a password on the command line interface can be insecure.
Benchmark
  Average number of seconds to run all queries: 9.314 seconds
  Minimum number of seconds to run all queries: 8.917 seconds
  Maximum number of seconds to run all queries: 11.034 seconds
  Number of clients running queries: 10
  Average number of queries per client: 1

yafet@yafet-VirtualBox:~$ █
```


Creating index for gpa

```
mysql> create index gpa_index on students(gpa);
Query OK, 0 rows affected (6.23 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> █
```

>3.5 above 3.5 gpa with index

```
mysql> explain select count(*) from students where gpa > 3.5\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: students
   partitions: NULL
         type: range
possible_keys: gpa_index
          key: gpa_index
       key_len: 2
         ref: NULL
        rows: 541288
   filtered: 100.00
      Extra: Using where; Using index
1 row in set, 1 warning (0.00 sec)

mysql> █
```

The number of rows decrease from 1082577 rows to 541288 rows with index

```
yafet@yafet-VirtualBox:~$ mysqlslap --create-schema=db1 --concurrency=10 --iterations=10 --
user=root --password=██████ --query="select count(*) from db1.students where gpa > 3.5"
mysqlslap: [Warning] Using a password on the command line interface can be insecure.
Benchmark
  Average number of seconds to run all queries: 2.924 seconds
  Minimum number of seconds to run all queries: 2.752 seconds
  Maximum number of seconds to run all queries: 3.232 seconds
  Number of clients running queries: 10
  Average number of queries per client: 1

yafet@yafet-VirtualBox:~$ █
```

The average number of second also decrease from 10.242 to 2.924 seconds with index

= 4.0 perfect gpa with index

```
mysql> explain select count(*) from students where gpa = 4.0\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: students
  partitions: NULL
         type: ref
possible_keys: gpa_index
          key: gpa_index
        key_len: 2
         ref: const
         rows: 5481
    filtered: 100.00
      Extra: Using index
1 row in set, 1 warning (0.00 sec)

mysql> █
```

The number of rows decrease from 1082577 rows to 5481 rows with index

```
yafet@yafet-VirtualBox:~$ mysqlslap --create-schema=db1 --concurrency=10 --iterations=10 --
user=root --password=██████ --query="select count(*) from db1.students where gpa = 4.0"
mysqlslap: [Warning] Using a password on the command line interface can be insecure.
Benchmark
  Average number of seconds to run all queries: 0.097 seconds
  Minimum number of seconds to run all queries: 0.046 seconds
  Maximum number of seconds to run all queries: 0.170 seconds
  Number of clients running queries: 10
  Average number of queries per client: 1

yafet@yafet-VirtualBox:~$ █
```

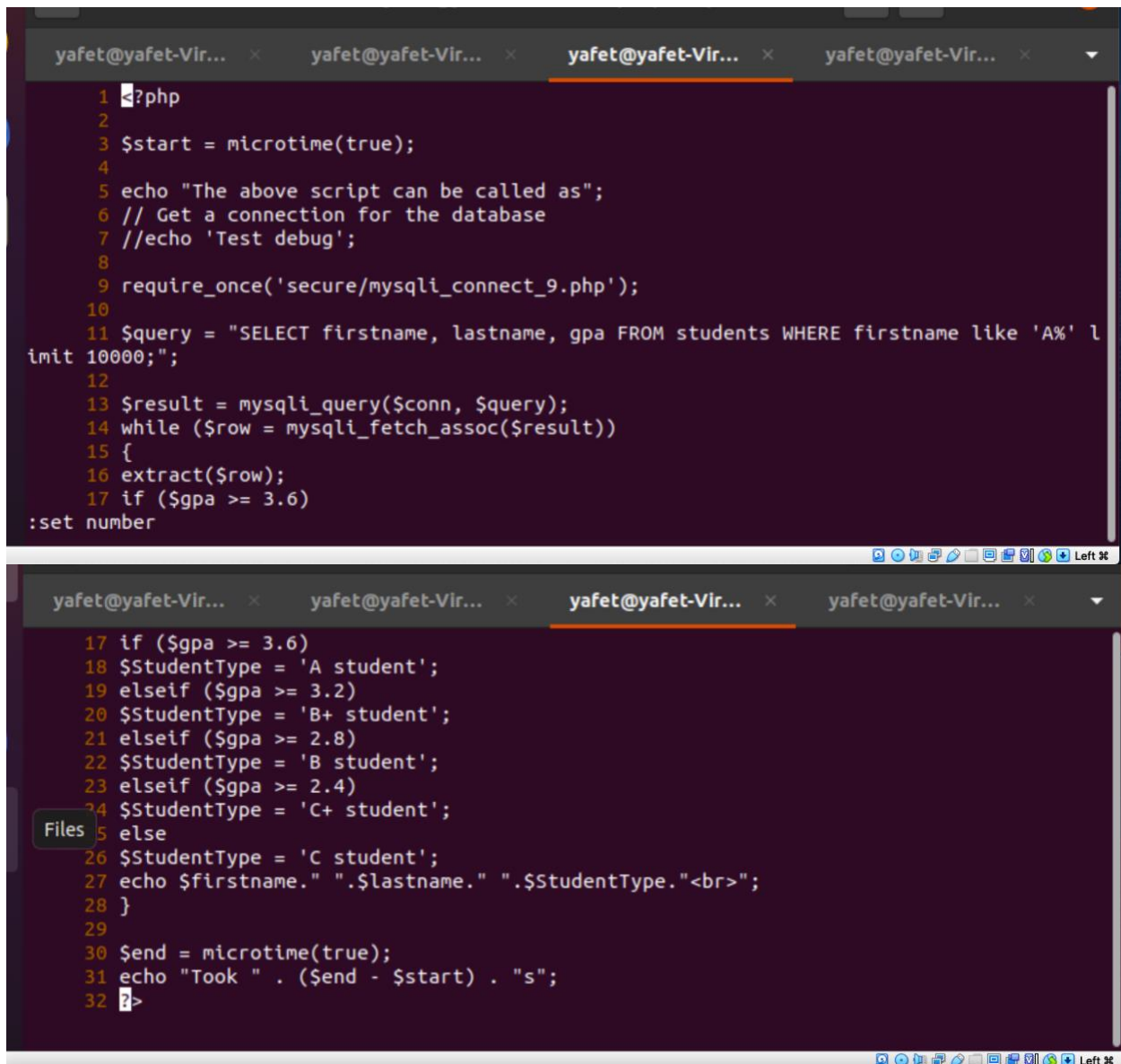
The average number of second also decrease from 9.314 to 0.097 seconds with index

e. 2 pts. The rows value from the explain command will probably not equal to 1000000. State why this is the case.

Since it stats the estimate number of rows examined

3. 10 pts. Client / Server Performance Analysis: Given the code snippets below, write the complete code. Limit the number of rows retrieved to ~10,000 rows (i.e. add a “like clause” that selects names starting with letter “A”).

1st one



The image shows two screenshots of a code editor with a dark theme. The top screenshot displays a PHP script starting with a file icon, followed by comments and a database query. The query is: `$query = "SELECT firstname, lastname, gpa FROM students WHERE firstname like 'A%' limit 10000;";`. The bottom screenshot continues the script with a series of `elseif` statements to classify students based on their GPA, and ends with a `microtime` call to measure execution time.

```
1 ?php
2
3 $start = microtime(true);
4
5 echo "The above script can be called as";
6 // Get a connection for the database
7 //echo 'Test debug';
8
9 require_once('secure/mysql_connect_9.php');
10
11 $query = "SELECT firstname, lastname, gpa FROM students WHERE firstname like 'A%' l
imit 10000;";
12
13 $result = mysqli_query($conn, $query);
14 while ($row = mysqli_fetch_assoc($result))
15 {
16     extract($row);
17     if ($gpa >= 3.6)
18         $StudentType = 'A student';
19     elseif ($gpa >= 3.2)
20         $StudentType = 'B+ student';
21     elseif ($gpa >= 2.8)
22         $StudentType = 'B student';
23     elseif ($gpa >= 2.4)
24         $StudentType = 'C+ student';
25     else
26         $StudentType = 'C student';
27     echo $firstname." ".$lastname." ".$StudentType."<br>";
28 }
29
30 $end = microtime(true);
31 echo "Took " . ($end - $start) . "s";
32 ?>
```


2nd one

```
yafet@yafet-Vir... x yafet@yafet-Vir... x yafet@yafet-Vir... x yafet@yafet-Vir...
1 <?php
2 $start = microtime(true);
3 //echo "The above script can be called as";
4 // Get a connection for the database
5 //echo 'Test debug';
6
7 require_once('secure/mysqli_connect_9.php');
8
9 $query = "SELECT CONCAT(firstname, ' ', lastname) as name,
10 CASE
11 WHEN gpa >= 3.6 THEN 'A student'
12 WHEN gpa >= 3.2 THEN 'B+ student'
13 WHEN gpa >= 2.8 THEN 'B student'
14 WHEN gpa >= 2.4 THEN 'C+ student'
15 ELSE 'C student'
16 END AS StudentType
:set number
```

```
15 ELSE 'C student'
16 END AS StudentType
17 FROM students WHERE firstname like 'A%' limit 10000";
18 $result = mysqli_query($conn, $query);
19 while ($row = mysqli_fetch_assoc($result))
20 {
21 extract($row);
22 echo $name." ".$StudentType."<br>";
23 }
24 $end = microtime(true);
25 echo "Took " . ($end - $start) . "s";
26 ?>
```

4. 5 pts. For each of the 2 different techniques shown in Step #3 above, include the results from the EXPLAIN and MYSQLSLAP tools. Describe how the processing is done to determine the performance difference.

It is impossible to compare php script using explain and mysqlslap since they will only use accept sql queries as shown below

```
mysql> explain SELECT firstname, lastname, gpa FROM students WHERE firstname like 'A%' limit 10000\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: students
   partitions: NULL
         type: range
possible_keys: fname
          key: fname
       key_len: 122
         ref: NULL
        rows: 92694
   filtered: 100.00
   Extra: Using index condition
1 row in set, 1 warning (0.00 sec)

mysql>
```

```
yafet@yafet-VirtualBox:~$ mysqlslap --create-schema=db1 --concurrency=10 --iterations=10 --user=root --password=[REDACTED] --query="SELECT firstname, lastname, gpa FROM students WHERE firstname like 'A%' limit 10000"
mysqlslap: [Warning] Using a password on the command line interface can be insecure.
Benchmark
  Average number of seconds to run all queries: 0.730 seconds
  Minimum number of seconds to run all queries: 0.645 seconds
  Maximum number of seconds to run all queries: 0.949 seconds
  Number of clients running queries: 10
  Average number of queries per client: 1

yafet@yafet-VirtualBox:~$
```

1st one

```
mysql> create table students_grade ( firstname varchar(30) not null, lastname varchar(30) not null, StudentType varchar(30) not null);
```

```
$StudentType = "C Student";
//echo $firstname." ".$lastname." ".$StudentType."<br>";

$query = "INSERT INTO students_grade (firstname,lastname,StudentType)
VALUES ('$firstname','$lastname','$StudentType')";

if($conn->query($query) == TRUE){ echo "Row added successfully: ".$query."<br>"; }
else { echo "Could not add row" . $conn->error; }

} VBOX_GAs_6.1.26
```

Inside the php code where insert into table students_grade happen

```
mysql> explain select count(*) from students_grade\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: students_grade
  partitions: NULL
         type: ALL
possible_keys: NULL
          key: NULL
        key_len: NULL
         ref: NULL
         rows: 9891
    filtered: 100.00
       Extra: NULL
1 row in set, 1 warning (0.01 sec)

mysql> █
```

Created a new table called students_grade and in the php file have another query that insert the firstname , lastname and StudentType inside the students_grade table

```
yafet@yafet-VirtualBox:~$ mysqlslap --create-schema=db1 --concurrency=10 --iterations=10 --
user=root --password=██████ --query="select count(*) from db1.students_grade "
mysqlslap: [Warning] Using a password on the command line interface can be insecure.
Benchmark
  Average number of seconds to run all queries: 0.202 seconds
  Minimum number of seconds to run all queries: 0.166 seconds
  Maximum number of seconds to run all queries: 0.402 seconds
  Number of clients running queries: 10
  Average number of queries per client: 1

yafet@yafet-VirtualBox:~$ █
```

2nd one

```
mysql> create table students_grade_2 ( name varchar(60) not null, StudentType varchar(30) n
ot null);
Query OK, 0 rows affected (0.06 sec)

mysql> █
```

Inside the php code where insert into table students_grade_2 happen
In this case since the first and last name are concat only one name needed

```
extract($row);
//echo $name." ".$StudentType."<br>";

$query = "INSERT INTO students_grade_2 (name,StudentType)
VALUES ('$name','$StudentType')";

if($conn->query($query) == TRUE){ echo "Row added successfully: ".$query."\n"; }
else { echo "Could not add row" . $conn->error; }

}
```

```
mysql> explain select count(*) from students_grade_2\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: students_grade_2
  partitions: NULL
         type: ALL
possible_keys: NULL
          key: NULL
        key_len: NULL
         ref: NULL
         rows: 10052
   filtered: 100.00
    Extra: NULL
1 row in set, 1 warning (0.00 sec)

mysql> █
```

```
yafet@yafet-VirtualBox:~$ mysqlslap --create-schema=db1 --concurrency=10 --iterations=10 --
user=root --password=██████ --query="select count(*) from db1.students_grade_2 "
mysqlslap: [Warning] Using a password on the command line interface can be insecure.
Benchmark
  Average number of seconds to run all queries: 0.186 seconds
  Minimum number of seconds to run all queries: 0.171 seconds
  Maximum number of seconds to run all queries: 0.213 seconds
  Number of clients running queries: 10
  Average number of queries per client: 1
```

2nd one overall performed better have faster time and examine more rows

5. 5 pts. For each of the given 2 different techniques shown in Step #1 above, measure the ELAPSED amount of time each takes using the php microtime function such as

1st one


```
yafet@yafet-Vir... x yafet@yafet-Vir... x yafet@yafet-Vir... x yafet@yafet-Vir... x
VXGZT C+ student<br>AFAQEDFDWIA LSTWKKUAXI B student<br>AFATQEUZEDX IAZ00TPQQ B+ student<br>
>AFAUERKXVGW CNINAPCGBLCEORKKL C student<br>AFAUPTTSXPOSCUCWCQO BNDQ B+ student<br>AFAUZXSQ
LUHLQYCWG DIUKLXBTVERV C student<br>AFAVBTKXMGXAVDY APTYOEJYYLRSQC C+ student<br>AFAW JOYDY
DBMDEZPKXP C+ student<br>AFAXTASKDSENS VKSULUXADM C+ student<br>AFAXWNWTTXHHUGMOBTL TPIGE
B Thunderbird Mail \XX FOJY B+ student<br>AFAYBOUQNLTUZ NCZNPCKQUUNRXILJE B student<br>AFAYNQZ
EJPIZIFRN JIILWYTCYEVGKEGVBBV B student<br>AFAYRWQWDGLX PWJQGX B student<br>AFAZUWKDW GVJOI
IFWQPLGJIC C+ student<br>AFB UYOUPIM C student<br>AFB DDJPEQGIHAWJTMVFGZUH B student<br>AF
BAHGYZGSDWKMHJOIMZ ZAGEBBSGEARVWUZTEE B student<br>AFBAKDYHHLDUYU NNZESWBPOMUU C student<br>
r>AFBCXPQQACPFQ QAKBMWEYRFTYL A student<br>AFBDNBCHMICEM RPYNNVYMB A student<br>AFBECCMJHO
FBXEWZMTHDYWJCMDZPB B+ student<br>AFBEGYXJVBG MHYI C student<br>AFBFUN CDQSXFBBNFGQSLZDPX
C student<br>AFBGAKZITKTHWZMXXQMX CYRDF C+ student<br>AFBGFLMJMWVDRVLTXY AGCQKCHNSPGQLK C+
student<br>AFBHQXEFZLRNSLMPULOO HWFPJRYXYKSUK B student<br>AFBIFZ ANGFKHBEFPOFSGKF C studen
t<br>AFBIQMUEPA KCWNVRCATUXSGRSZ C+ student<br>AFBIYITNHNGS APJGLZZCRCC C+ student<br>AFB
JDCTTP TMLVKMVONY B student<br>AFBJKQVKUKBWCBAJMJM MZNNZ C student<br>AFBJVUUP CCCIHJVDAZF
XYYHD C student<br>AFBJYCOHXBKKM VDVKIBX C+ student<br>AFBKGLTXJTHQGNVRP IFVUAMDLYWBNR B s
tudent<br>AFBLLS QWFODGS C+ student<br>Took 0.9499351978302syafet@yafet-VirtualBox:/var/www
/html$
```

```
Files t@yafet-Vir... x yafet@yafet-Vir... x yafet@yafet-Vir... x yafet@yafet-Vir... x
Row added successfully: INSERT INTO students_grade (firstname,lastname,StudentType)
VALUES ('AFBIQMUEPA','KCWNVRCATUXSGRSZ','C+ student')
Row added successfully: INSERT INTO students_grade (firstname,lastname,StudentType)
VALUES ('AFBIYITNHNGS','APJGLZZCRCC','C+ student')
Row added successfully: INSERT INTO students_grade (firstname,lastname,StudentType)
VALUES ('AFBJDCTTP','TMLVKMVONY','B student')
Row added successfully: INSERT INTO students_grade (firstname,lastname,StudentType)
VALUES ('AFBJKQVKUKBWCBAJMJM','MZNNZ','C student')
Row added successfully: INSERT INTO students_grade (firstname,lastname,StudentType)
VALUES ('AFBJVUUP','CCCIHJVDAZFXYHD','C student')
Row added successfully: INSERT INTO students_grade (firstname,lastname,StudentType)
VALUES ('AFBJYCOHXBKKM','VDVKIBX','C+ student')
Row added successfully: INSERT INTO students_grade (firstname,lastname,StudentType)
VALUES ('AFBKGLTXJTHQGNVRP','IFVUAMDLYWBNR','B student')
Row added successfully: INSERT INTO students_grade (firstname,lastname,StudentType)
VALUES ('AFBLLS','QWFODGS','C+ student')
Took 121.24396300316syafet@yafet-VirtualBox:/var/www/html$
```

2nd one


```
VXGZT C+ student<br>AFAQEDFDWIA LSTWKKUAXI B student<br>AFATQEUEZDX IAZOOTPQQ B+ student<br>AFAUERKXVGW CNINAPCGBLCEORKKL C student<br>AFAUPTTSXPOSCUCWCQO BNDQ B+ student<br>AFAUZXSQ LUHLQYCWG DIUKLXBTVERV C student<br>AFAVBTKXMGXAVDY APTYOEJYYLRSQC C+ student<br>AFAW JOYDY DBMDEZPKXP C+ student<br>AFAXTASKDSENS VKSULUXADM C+ student<br>AFAXWNWTHXHHUGMOBTL TPIGE B student<br>AFAXX FOJY B+ student<br>AFAYBOUQNLТУ NCZNPCCCKQUUNRXILJE B student<br>AFAYNQZ EJPZTFRN JIILWYTCYEVGKEGVBVV B student<br>AFAYRWQWDGLX PWJQGX L B student<br>AFAZUWKDW GVJOI IFWQPLGJIC C+ student<br>AFB UYUUPIKM C student<br>AFB DDJPEQGIHAWJTMVFGZUH B student<br>AF BAHGYZGSDWKMHJOIMZ ZAGEBBTSGEARVWUZTEE B student<br>AFBAKDYHHLDUYU NNZESWBPOMUU C student<br>AFBCXPQQACPF QAKBMWEYRFTYL A student<br>AFBDNBCHMICEM RPYNNVYMB A student<br>AFBECCMJHO FBXEWZHMTHDYWJCMDZPB B+ student<br>AFBEGYXJVBG MHYI C student<br>AFBFUN CDQSXFBBNFGQSLZDPX C student<br>AFBGAKZITKTHWZMXXQMX CYRDF C+ student<br>AFBGFLMJMWVDRVLTKY AGCQKCHNSPGQLK C+ student<br>AFBHQXEFZLRNSLMPULOO HWFPJRYXYKSUK B student<br>AFBIFZ ANGFKHBEFPOFSGKF C student<br>AFBIQMUEPA KCWNVRCATUXSGRSZ C+ student<br>AFBIYITNHNGS APJGLZZCRCC C+ student<br>AFB JDCTT TMLVKMVONY B student<br>AFBJKQVKUKBWCBNAMJM MZNNZ C student<br>AFBJVUUP CCCIHJVDAZF XYYHD C student<br>AFBJYCOHXBKMM VDVKIBX C+ student<br>AFBKGLTXJTHQGNVRP IFVUAMDLYWBNR B student<br>AFBLLS QWFODGS C+ student<br>Took 0.48240995407104syafet@yafet-VirtualBox:/var/www/html$
```

```
yafet@yaf... x yafet@yaf... x yafet@yaf... x yafet@yaf... x yafet@yaf...
Row added successfully: INSERT INTO students_grade_2 (name,StudentType)
VALUES ('AFBIQMUEPA KCWNVRCATUXSGRSZ','C+ student')
Row added successfully: INSERT INTO students_grade_2 (name,StudentType)
VALUES ('AFBIYITNHNGS APJGLZZCRCC','C+ student')
Row added successfully: INSERT INTO students_grade_2 (name,StudentType)
VALUES ('AFBJDCTT TMLVKMVONY','B student')
Row added successfully: INSERT INTO students_grade_2 (name,StudentType)
VALUES ('AFBJKQVKUKBWCBNAMJM MZNNZ','C student')
Row added successfully: INSERT INTO students_grade_2 (name,StudentType)
VALUES ('AFBJVUUP CCCIHJVDAZFXYHD','C student')
Row added successfully: INSERT INTO students_grade_2 (name,StudentType)
VALUES ('AFBJYCOHXBKMM VDVKIBX','C+ student')
Row added successfully: INSERT INTO students_grade_2 (name,StudentType)
VALUES ('AFBKGLTXJTHQGNVRP IFVUAMDLYWBNR','B student')
Row added successfully: INSERT INTO students_grade_2 (name,StudentType)
VALUES ('AFBLLS QWFODGS','C+ student')
Took 90.375618934631syafet@yafet-VirtualBox:/var/www/html$
```