**Mortel, Meg Edelviese B.** **Prof. Bedis**

**BSIT 3-1N** **Data Administration**

# Activity 2: Writing SQL Syntax

**Instructions:**

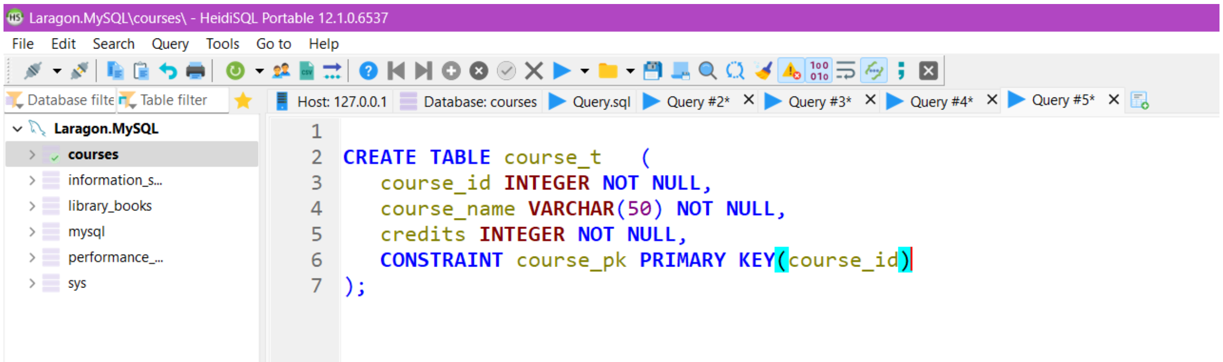
Write the appropriate SQL syntax for the following tasks. Use the examples provided in the lesson as a guide.

1. Create a table named course_t with the following columns:

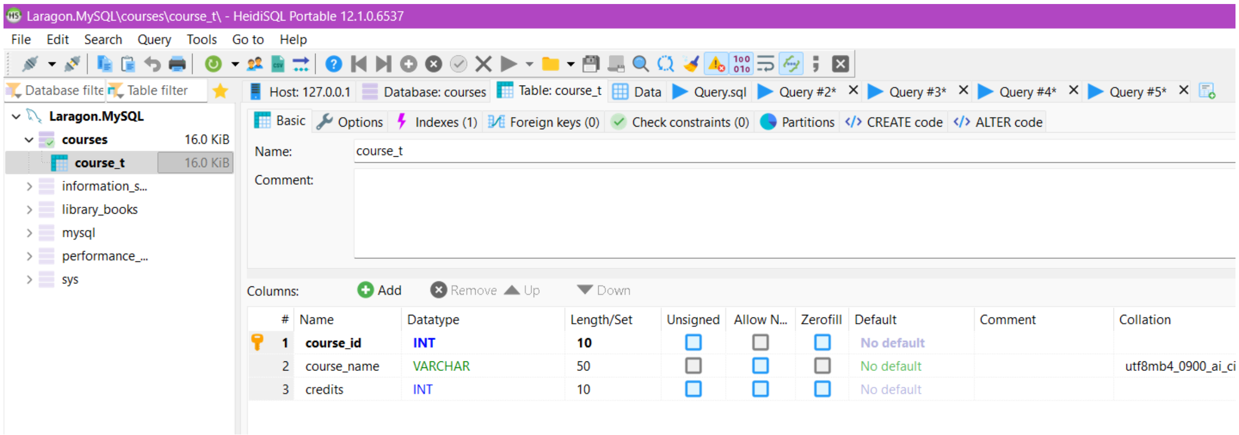
   ○ course_id (INTEGER, primary key, not null)

   ○ course_name (VARCHAR(50), not null)

   ○ credits (INTEGER, not null)


Laragon.MySQL\courses\ - HeidiSQL Portable 12.1.0.6537
File Edit Search Query Tools Go to Help
Database filte Table filter
Host: 127.0.0.1 Database: courses Query.sql Query #2* Query #3* Query #4* Query #5*
Laragon.MySQL
courses
information_s...
library_books
mysql
performance_...
sys

```

CREATE TABLE course_t   (
   course_id INTEGER NOT NULL,
   course_name VARCHAR(50) NOT NULL,
   credits INTEGER NOT NULL,
   CONSTRAINT course_pk PRIMARY KEY(course_id)
);
```

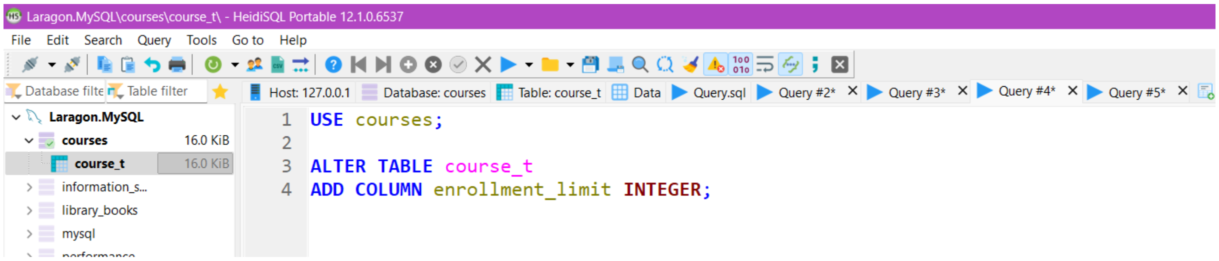

Laragon.MySQL\courses\course_t\ - HeidiSQL Portable 12.1.0.6537
File Edit Search Query Tools Go to Help
Database filte Table filter
Host: 127.0.0.1 Database: courses Table: course_t Data Query.sql Query #2* Query #3* Query #4* Query #5*
Laragon.MySQL
courses 16.0 KiB
course_t 16.0 KiB
information_s...
library_books
mysql
performance_...
sys
Basic Options Indexes (1) Foreign keys (0) Check constraints (0) Partitions CREATE code ALTER code
Columns: Add Remove Up Down

Name: course_t

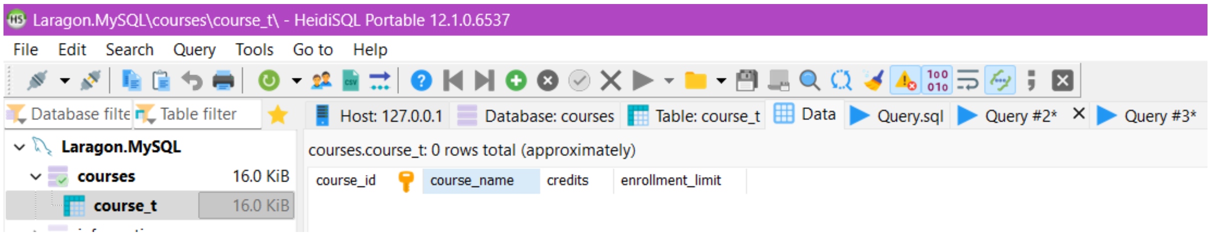
Comment:

| # | Name | Datatype | Length/Set | Unsigned | Allow N... | Zerofill | Default | Comment | Collation |
|---|---|---|---|---|---|---|---|---|---|
| 1 | course_id | INT | 10 | □ | □ | □ | No default | | |
| 2 | course_name | VARCHAR | 50 | □ | □ | □ | No default | | utf8mb4_0900_ai_ci |
| 3 | credits | INT | 10 | □ | □ | □ | No default | | |

2. Add a column enrollment_limit (INTEGER) to the course_t table.


Laragon.MySQL\courses\course_t\ - HeidiSQL Portable 12.1.0.6537
File Edit Search Query Tools Go to Help
Database filte Table filter
Host: 127.0.0.1 Database: courses Table: course_t Data Query.sql Query #2* Query #3* Query #4* Query #5*
Laragon.MySQL
courses 16.0 KiB
course_t 16.0 KiB
information_s...
library_books
mysql
performance

```
USE courses;

ALTER TABLE course_t
ADD COLUMN enrollment_limit INTEGER;
```

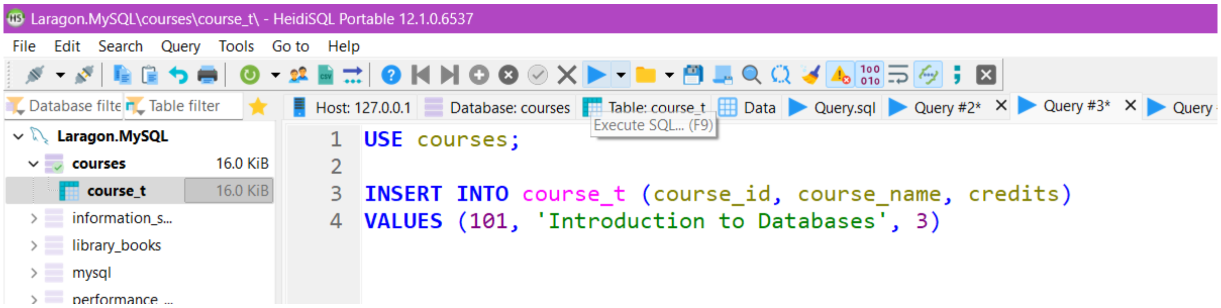


courses.course_t: 0 rows total (approximately)

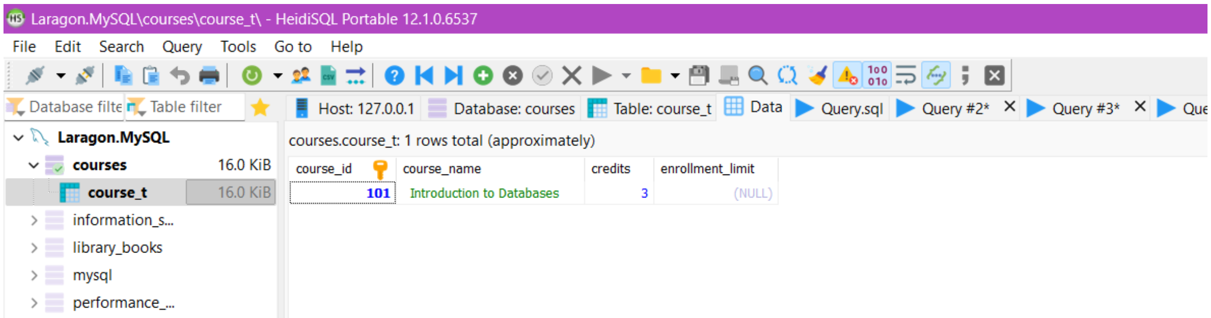
| course_id | course_name | credits | enrollment_limit |
|---|---|---|---|

3. Insert the following data into the course_t table:

   - course_id: 101
   - course_name: "Introduction to Databases"
   - credits: 3



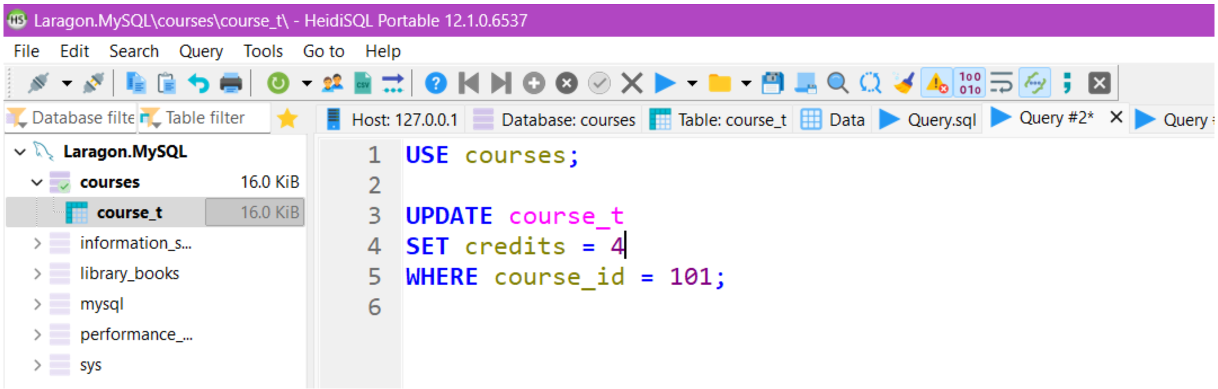
```sql
USE courses;

INSERT INTO course_t (course_id, course_name, credits)
VALUES (101, 'Introduction to Databases', 3)
```



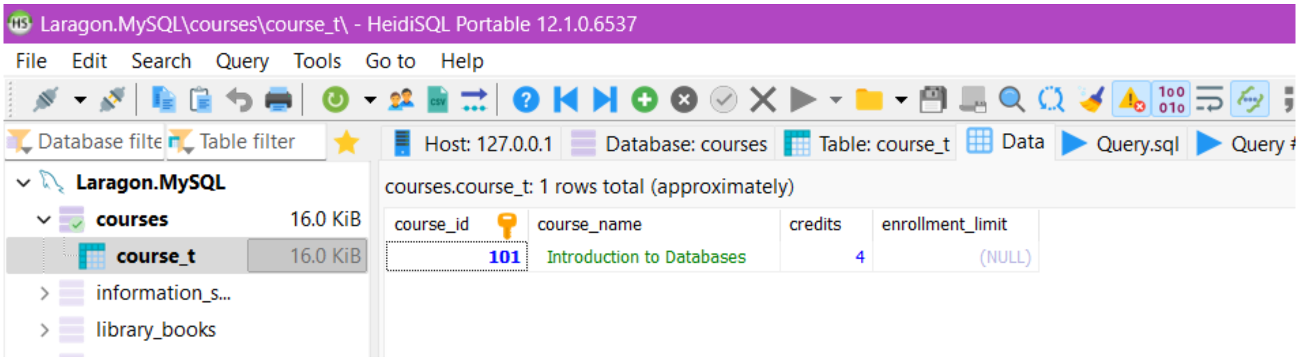
courses.course_t: 1 rows total (approximately)

| course_id | course_name | credits | enrollment_limit |
|---|---|---|---|
| 101 | Introduction to Databases | 3 | (NULL) |

4. Update the credits column for the course with course_id 101 to 4.



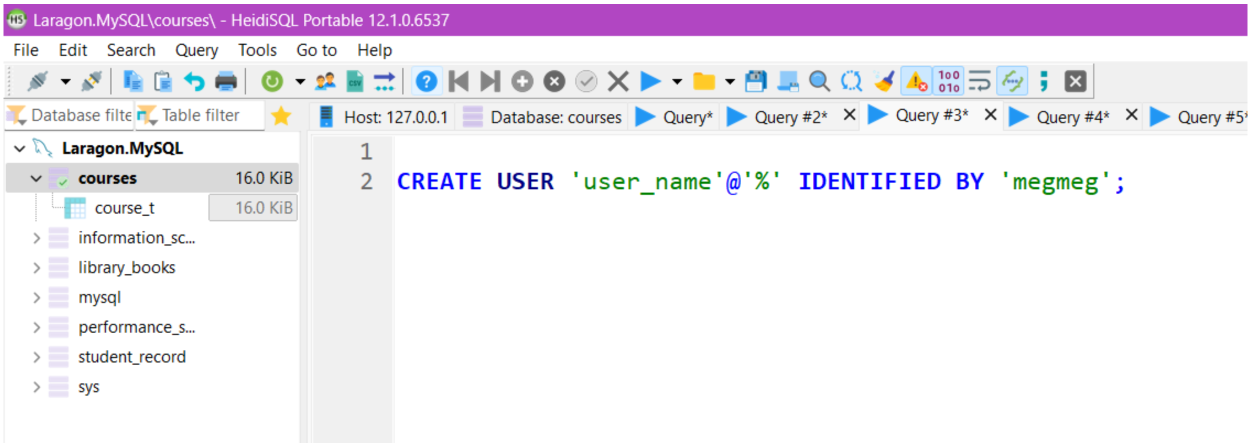
```sql
USE courses;

UPDATE course_t
SET credits = 4
WHERE course_id = 101;
```


Laragon.MySQL\courses\course_t\ - HeidiSQL Portable 12.1.0.6537
File Edit Search Query Tools Go to Help
Database filte Table filter
Host: 127.0.0.1 Database: courses Table: course_t Data Query.sql Query
Laragon.MySQL
courses 16.0 KiB
course_t 16.0 KiB
information_s...
library_books

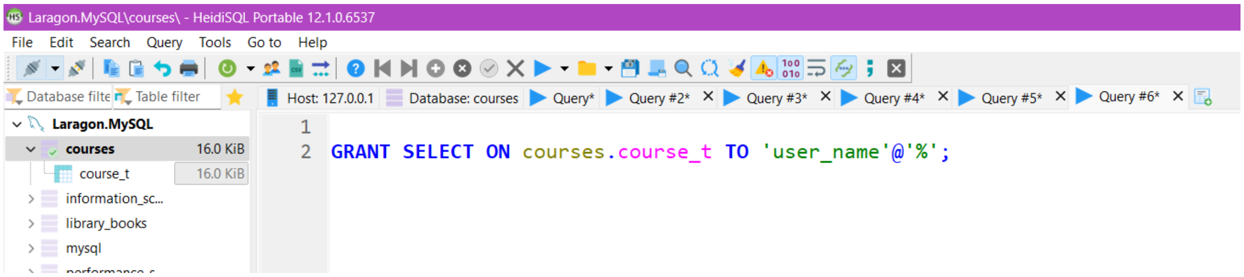
courses.course_t: 1 rows total (approximately)

| course_id | course_name | credits | enrollment_limit |
|---|---|---|---|
| 101 | Introduction to Databases | 4 | (NULL) |

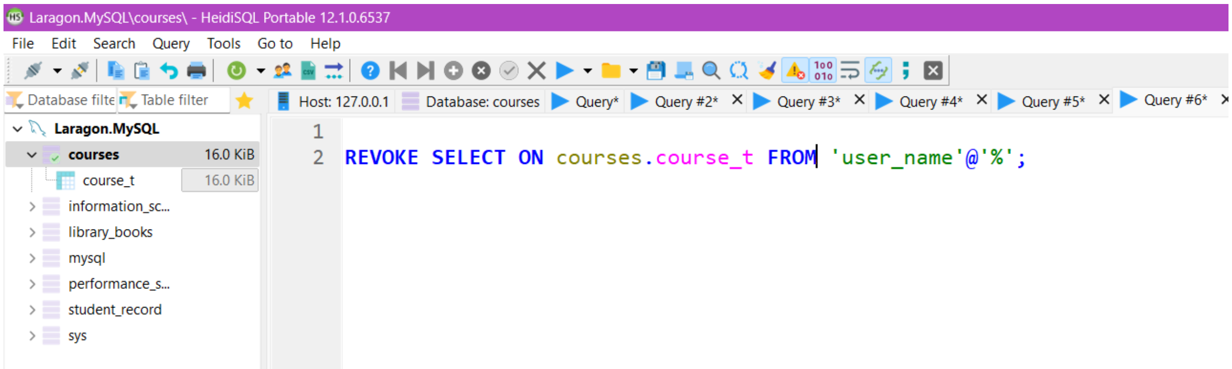
5. Revoke permission for a user to query the course_t table.


Laragon.MySQL\courses\ - HeidiSQL Portable 12.1.0.6537
File Edit Search Query Tools Go to Help
Database filte Table filter
Host: 127.0.0.1 Database: courses Query* Query #2* Query #3* Query #4* Query #5*
Laragon.MySQL
courses 16.0 KiB
course_t 16.0 KiB
information_sc...
library_books
mysql
performance_s...
student_record
sys

```sql

CREATE USER 'user_name'@'%' IDENTIFIED BY 'megmeg';
```


Laragon.MySQL\courses\ - HeidiSQL Portable 12.1.0.6537
File Edit Search Query Tools Go to Help
Database filte Table filter
Host: 127.0.0.1 Database: courses Query* Query #2* Query #3* Query #4* Query #5* Query #6*
Laragon.MySQL
courses 16.0 KiB
course_t 16.0 KiB
information_sc...
library_books
mysql

```sql

GRANT SELECT ON courses.course_t TO 'user_name'@'%';
```

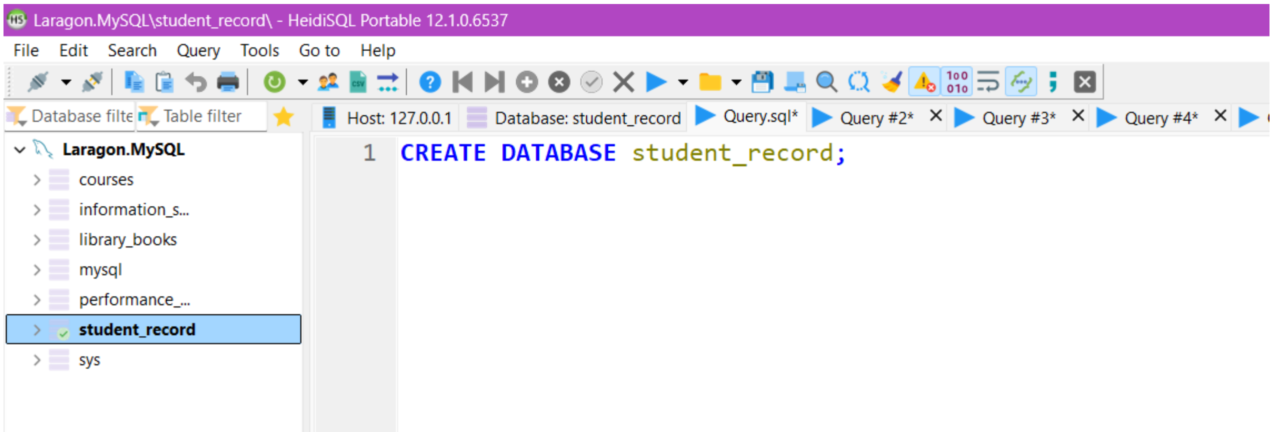

Laragon.MySQL\courses\ - HeidiSQL Portable 12.1.0.6537
File Edit Search Query Tools Go to Help
Database filte Table filter
Host: 127.0.0.1 Database: courses Query* Query #2* Query #3* Query #4* Query #5* Query #6*
Laragon.MySQL
courses 16.0 KiB
course_t 16.0 KiB
information_sc...
library_books
mysql
performance_s...
student_record
sys

```sql

REVOKE SELECT ON courses.course_t FROM 'user_name'@'%';
```

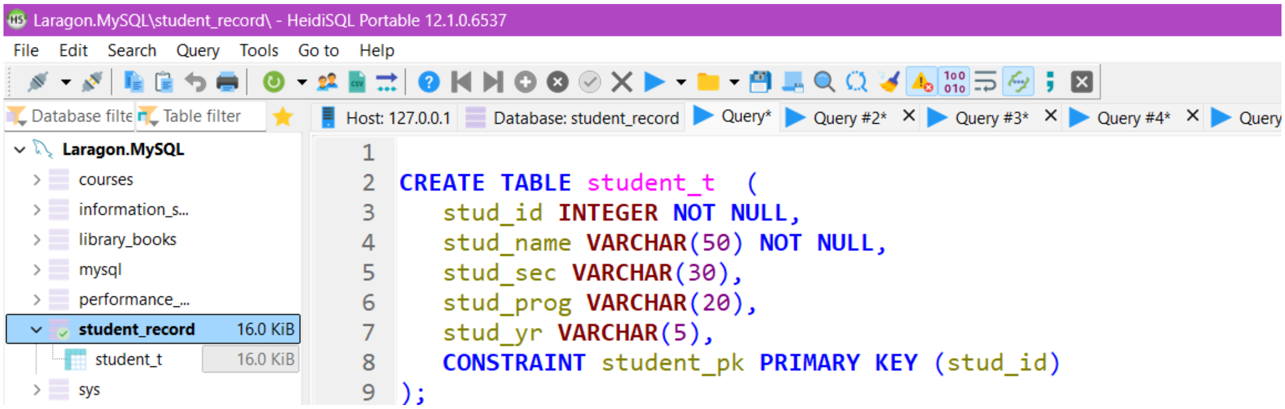
**PART #2**

**Instructions:** Write the appropriate SQL syntax for the following tasks.

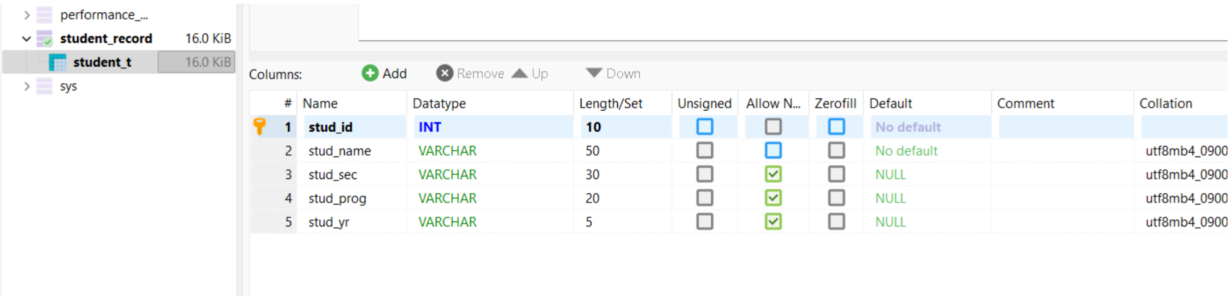
1. Create a table student_t with the following columns:
   - stud_id (INTEGER, Primary Key, NOT NULL)
   - stud_name (VARCHAR(50), NOT NULL)
   - stud_sec (VARCHAR(30))
   - stud_prog (VARCHAR(20))
   - stud_yr (VARCHAR(5))



```sql
CREATE DATABASE student_record;
```

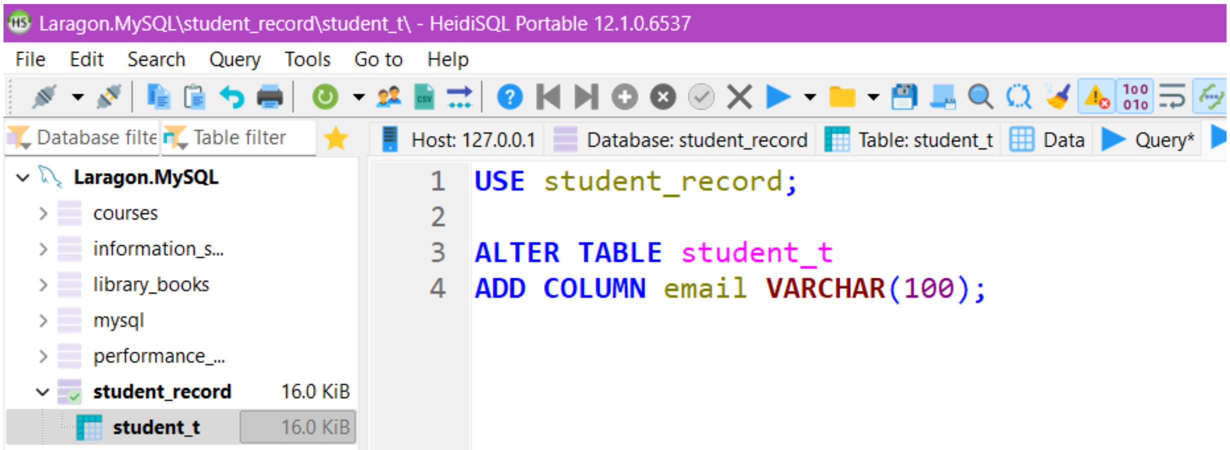


```sql

CREATE TABLE student_t  (
    stud_id INTEGER NOT NULL,
    stud_name VARCHAR(50) NOT NULL,
    stud_sec VARCHAR(30),
    stud_prog VARCHAR(20),
    stud_yr VARCHAR(5),
    CONSTRAINT student_pk PRIMARY KEY (stud_id)
);
```

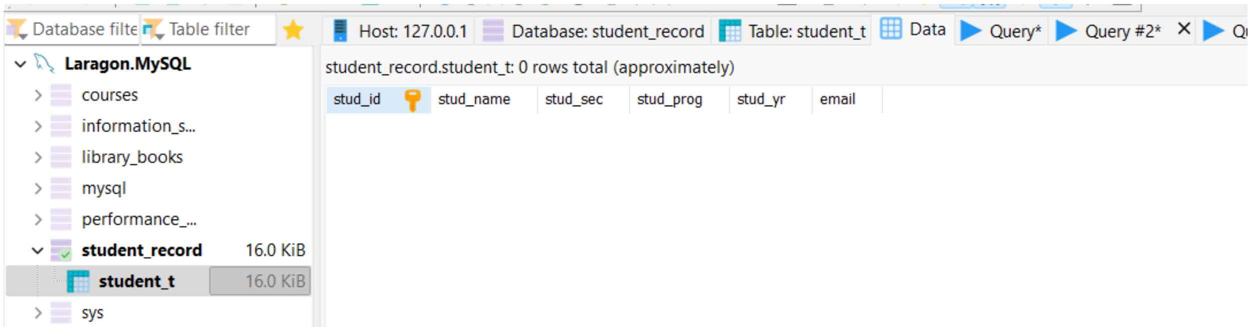


| # | Name | Datatype | Length/Set | Unsigned | Allow N... | Zerofill | Default | Comment | Collation |
|---|---|---|---|---|---|---|---|---|---|
| 1 | stud_id | INT | 10 | ☐ | ☐ | ☐ | No default | | |
| 2 | stud_name | VARCHAR | 50 | ☐ | ☐ | ☐ | No default | | utf8mb4_0900 |
| 3 | stud_sec | VARCHAR | 30 | ☐ | ☑ | ☐ | NULL | | utf8mb4_0900 |
| 4 | stud_prog | VARCHAR | 20 | ☐ | ☑ | ☐ | NULL | | utf8mb4_0900 |
| 5 | stud_yr | VARCHAR | 5 | ☐ | ☑ | ☐ | NULL | | utf8mb4_0900 |

2. Add a column email (VARCHAR(100)) to the student_t table.



```sql
USE student_record;

ALTER TABLE student_t
ADD COLUMN email VARCHAR(100);
```

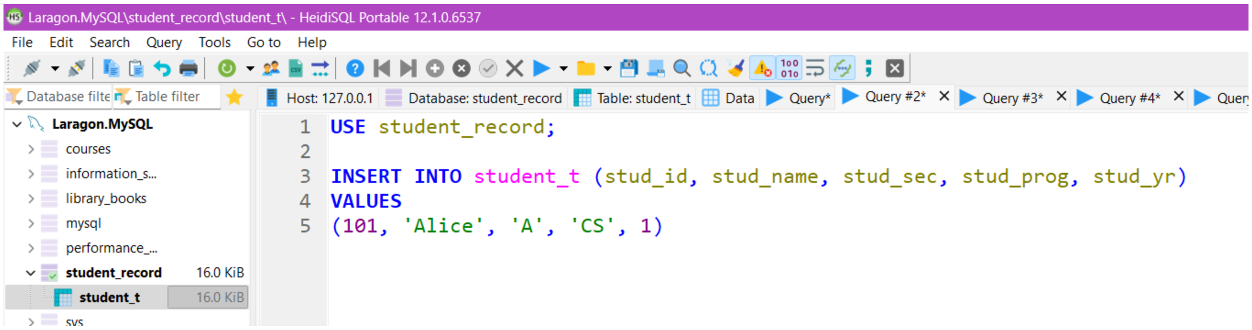


student_record.student_t: 0 rows total (approximately)

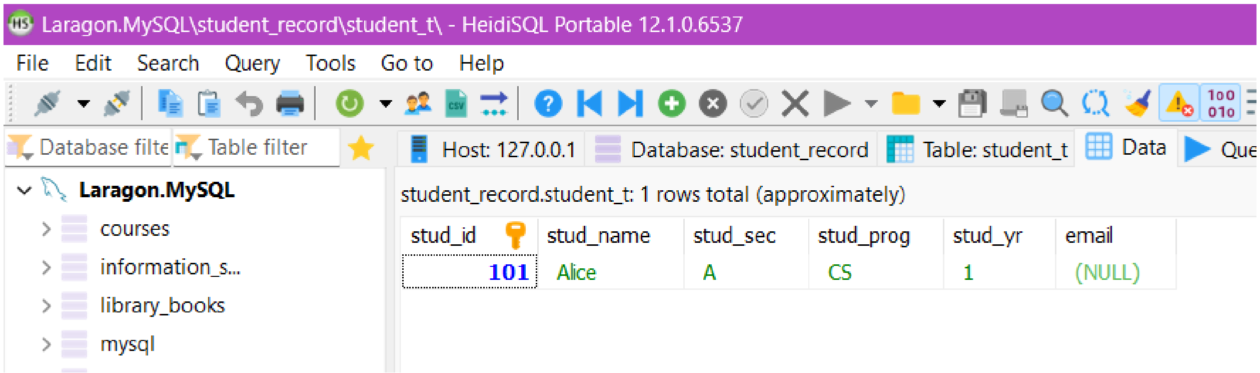
| stud_id | stud_name | stud_sec | stud_prog | stud_yr | email |
|---|---|---|---|---|---|

3. Insert the following record into the student_t table:

  ○ stud_id: 101, stud_name: "Alice", stud_sec: "A", stud_prog: "CS", stud_yr: "1"



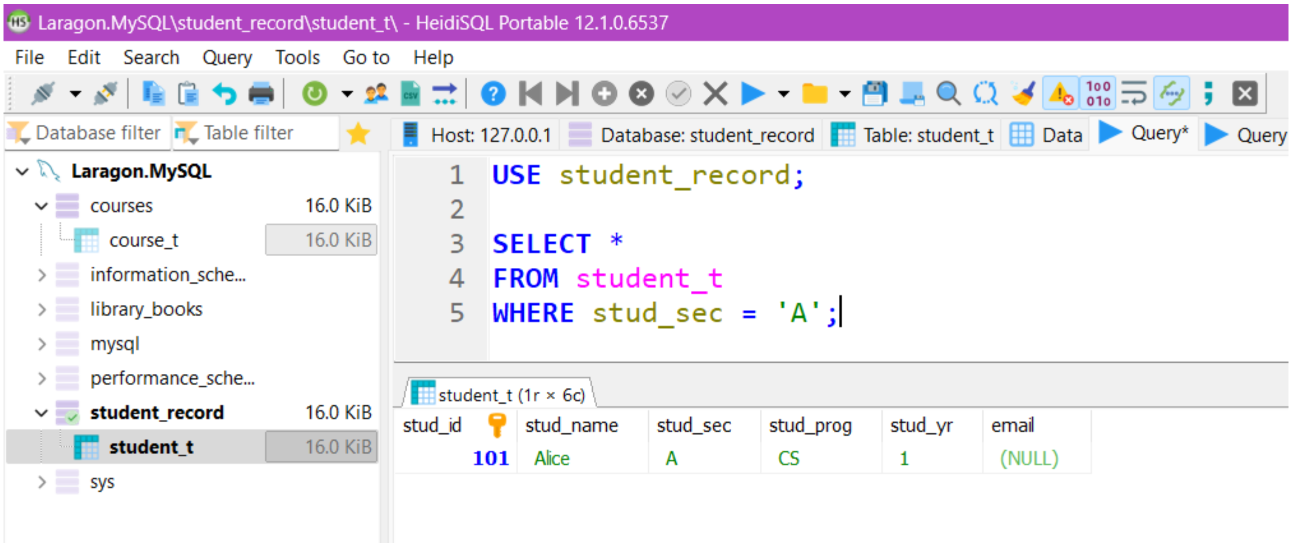
```sql
USE student_record;

INSERT INTO student_t (stud_id, stud_name, stud_sec, stud_prog, stud_yr)
VALUES
(101, 'Alice', 'A', 'CS', 1)
```



student_record.student_t: 1 rows total (approximately)

| stud_id | stud_name | stud_sec | stud_prog | stud_yr | email |
|---|---|---|---|---|---|
| 101 | Alice | A | CS | 1 | (NULL) |

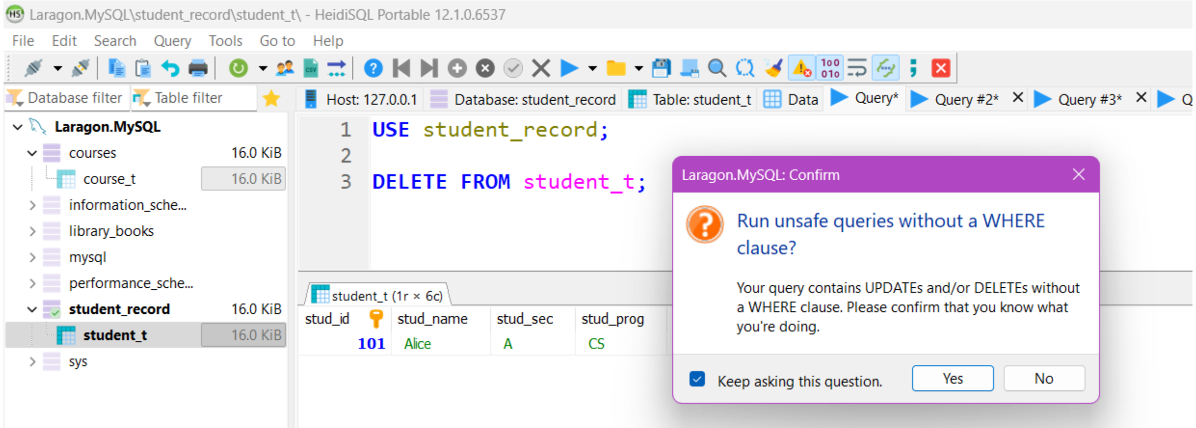
4. Write a query to retrieve all records from the student_t table where stud_sec is "A".



```sql
USE student_record;

SELECT *
FROM student_t
WHERE stud_sec = 'A';
```

| stud_id | stud_name | stud_sec | stud_prog | stud_yr | email |
|---|---|---|---|---|---|
| 101 | Alice | A | CS | 1 | (NULL) |

5. Delete all records from the student_t table but retain the table structure.



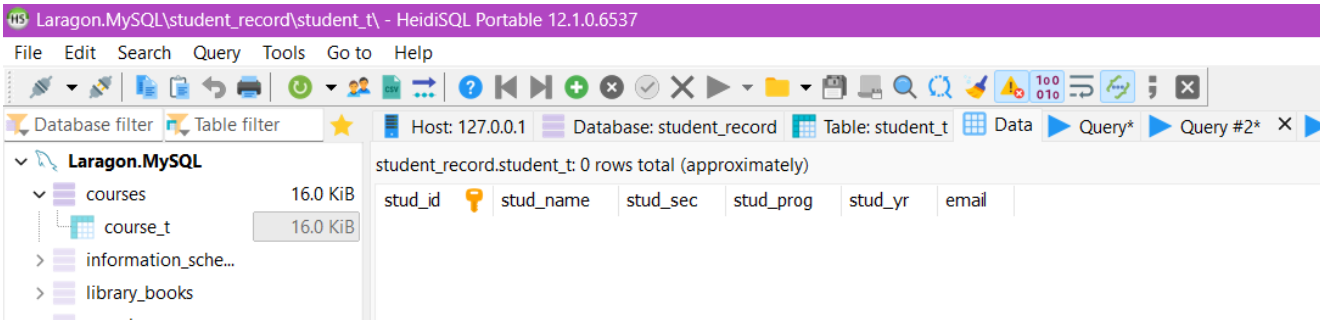
```sql
USE student_record;

DELETE FROM student_t;
```

| stud_id | stud_name | stud_sec | stud_prog |
|---|---|---|---|
| 101 | Alice | A | CS |

Run unsafe queries without a WHERE clause?

Your query contains UPDATEs and/or DELETEs without a WHERE clause. Please confirm that you know what you're doing.

☑ Keep asking this question.



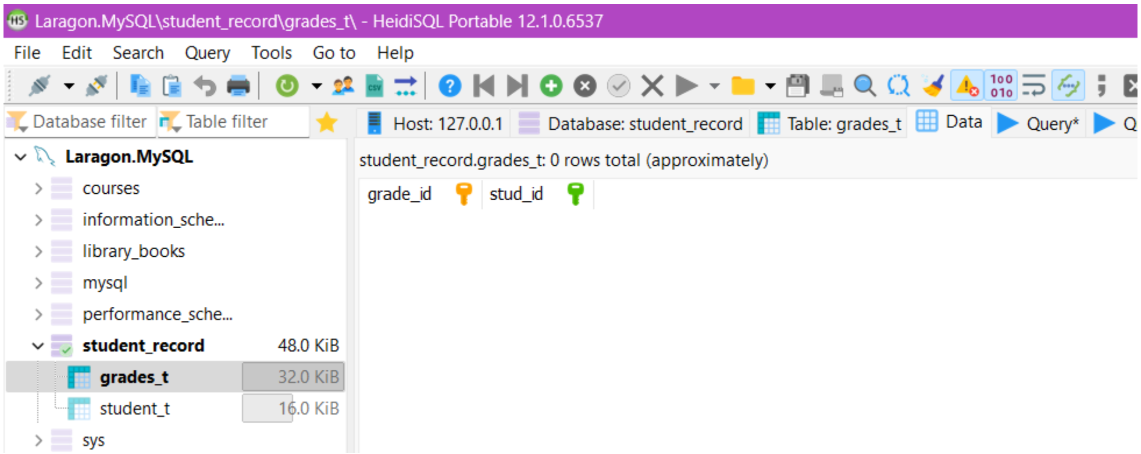
student_record.student_t: 0 rows total (approximately)

| stud_id | stud_name | stud_sec | stud_prog | stud_yr | email |
|---|---|---|---|---|---|

6. Create a table grades_t with the following constraints:

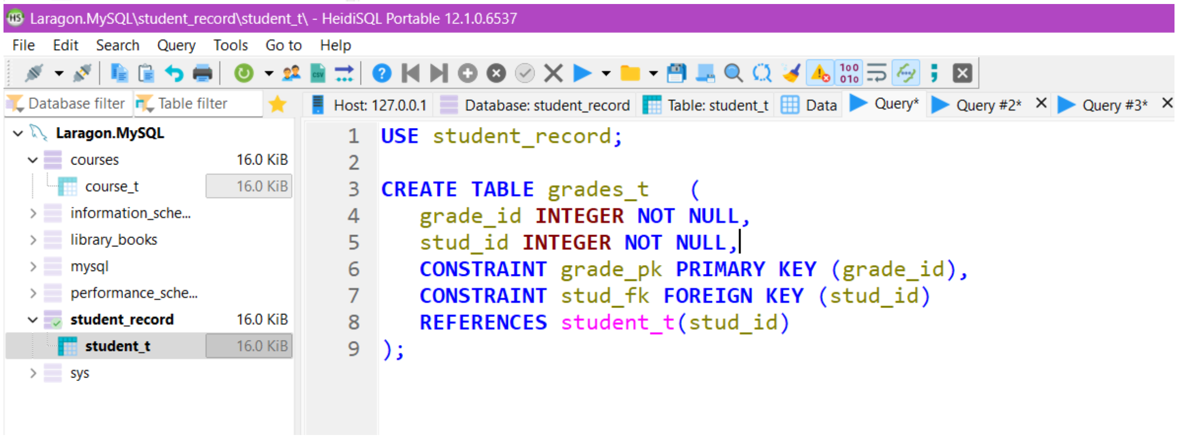
   ○ grade_id as the Primary Key

   ○ stud_id as a Foreign Key referencing student_t



student_record.grades_t: 0 rows total (approximately)

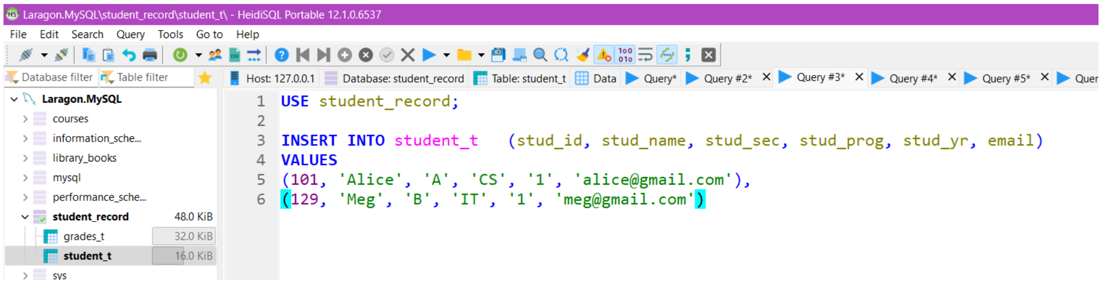
| grade_id | stud_id |
|---|---|



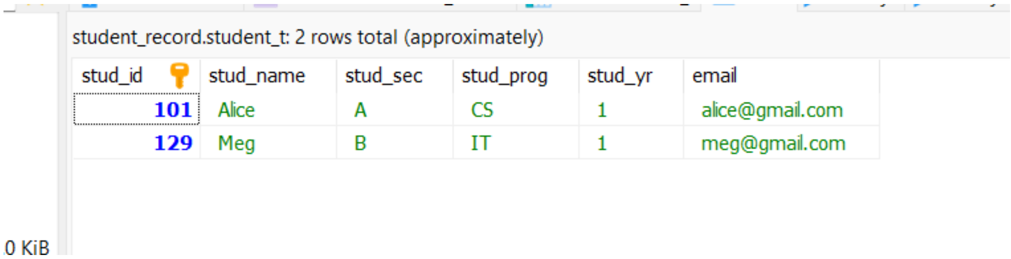
```
USE student_record;

CREATE TABLE grades_t   (
    grade_id INTEGER NOT NULL,
    stud_id INTEGER NOT NULL,
    CONSTRAINT grade_pk PRIMARY KEY (grade_id),
    CONSTRAINT stud_fk FOREIGN KEY (stud_id)
    REFERENCES student_t(stud_id)
);
```

7. Update the stud_yr of the student with stud_id 101 to "2".

    ○ I inserted new values as I deleted the data of the table instructed in #5



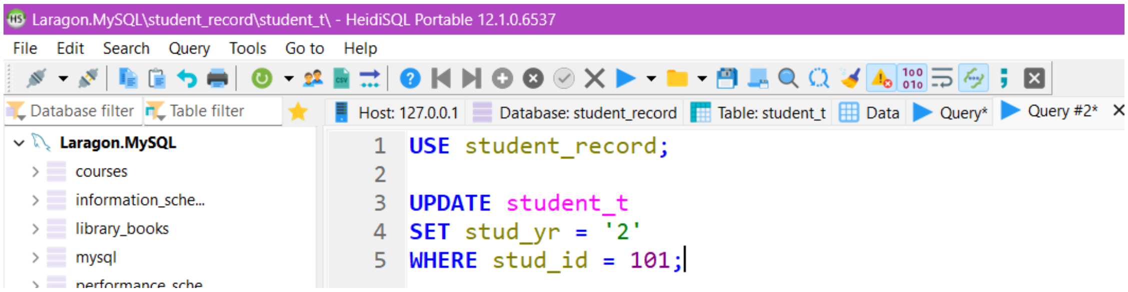
```
USE student_record;

INSERT INTO student_t   (stud_id, stud_name, stud_sec, stud_prog, stud_yr, email)
VALUES
(101, 'Alice', 'A', 'CS', '1', 'alice@gmail.com'),
(129, 'Meg', 'B', 'IT', '1', 'meg@gmail.com')
```



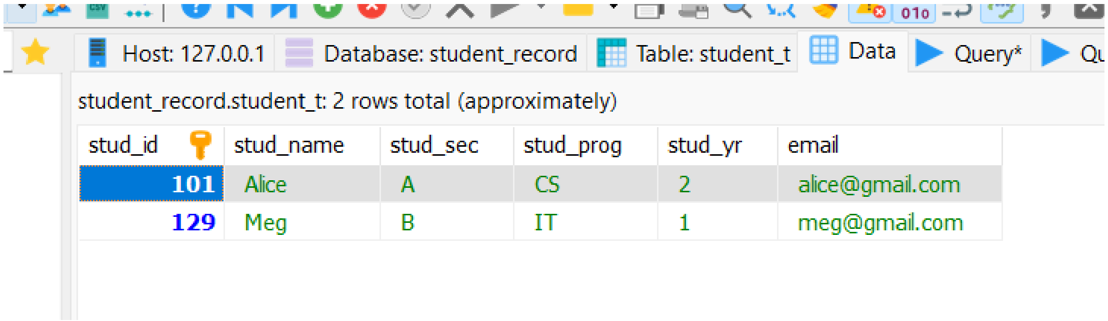
student_record.student_t: 2 rows total (approximately)

| stud_id | stud_name | stud_sec | stud_prog | stud_yr | email |
|---|---|---|---|---|---|
| 101 | Alice | A | CS | 1 | alice@gmail.com |
| 129 | Meg | B | IT | 1 | meg@gmail.com |

    ○ Then updated the year of stud_id 101:



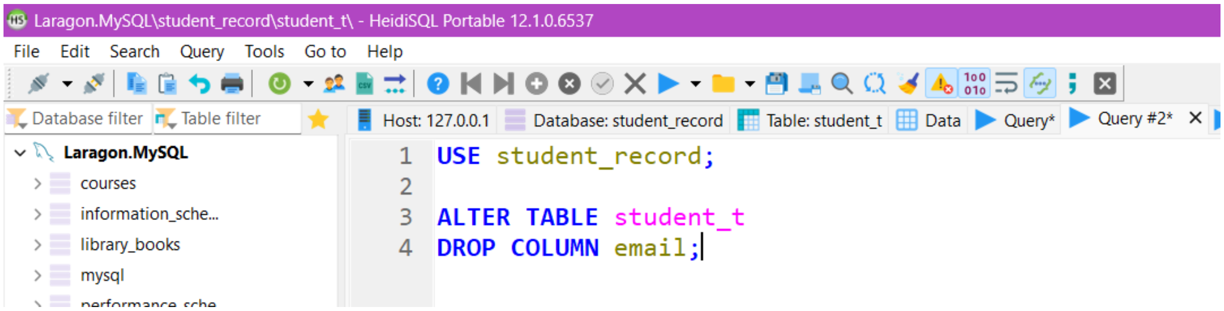
```sql
USE student_record;

UPDATE student_t
SET stud_yr = '2'
WHERE stud_id = 101;
```



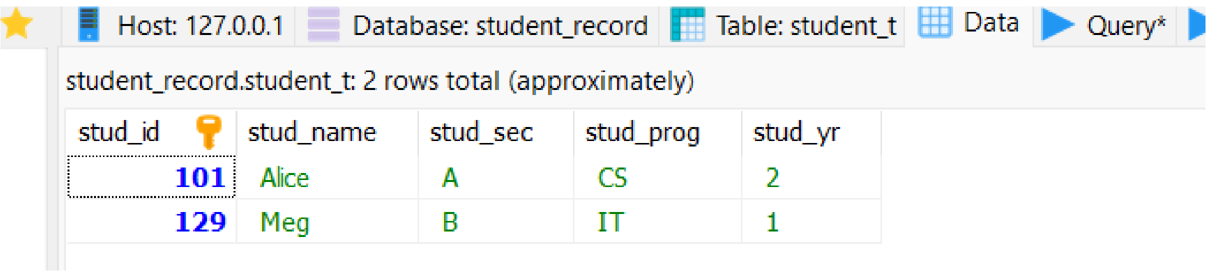
student_record.student_t: 2 rows total (approximately)

| stud_id | stud_name | stud_sec | stud_prog | stud_yr | email |
|---|---|---|---|---|---|
| 101 | Alice | A | CS | 2 | alice@gmail.com |
| 129 | Meg | B | IT | 1 | meg@gmail.com |

8. Write the syntax to remove the email column from the student_t table.



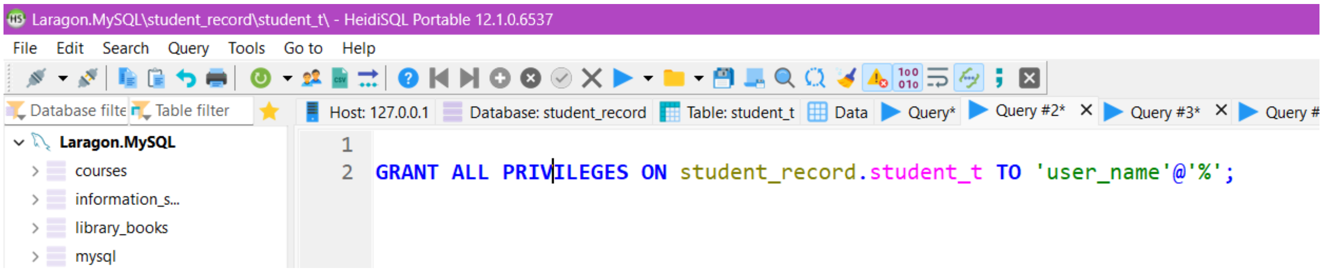
```sql
USE student_record;

ALTER TABLE student_t
DROP COLUMN email;
```



student_record.student_t: 2 rows total (approximately)

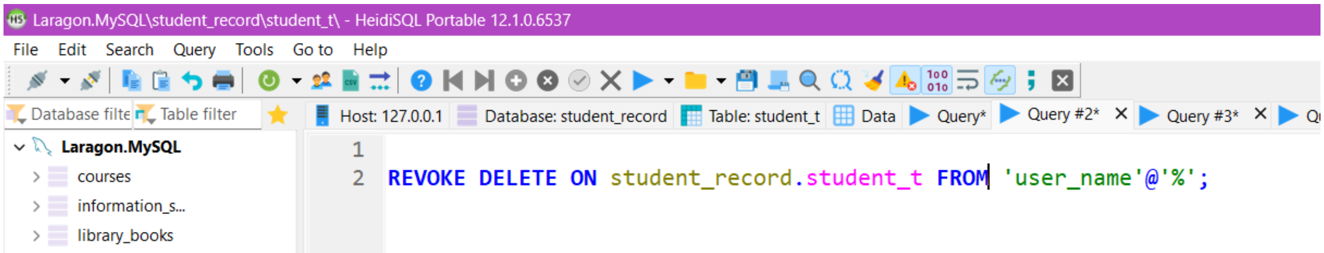
| stud_id | stud_name | stud_sec | stud_prog | stud_yr |
|---|---|---|---|---|
| 101 | Alice | A | CS | 2 |
| 129 | Meg | B | IT | 1 |

9. Grant permission to a user to query the student_t table.



```sql

GRANT ALL PRIVILEGES ON student_record.student_t TO 'user_name'@'%';
```

10. Revoke permission for a user to delete records from the student_t table.



```sql
REVOKE DELETE ON student_record.student_t FROM 'user_name'@'%';
```