

DBMS Project

Title:-Subject Management System

Aditya Tomar (202051012)

Aniket Yadav (202051025)

Avnit Kumar Anand (202051040)

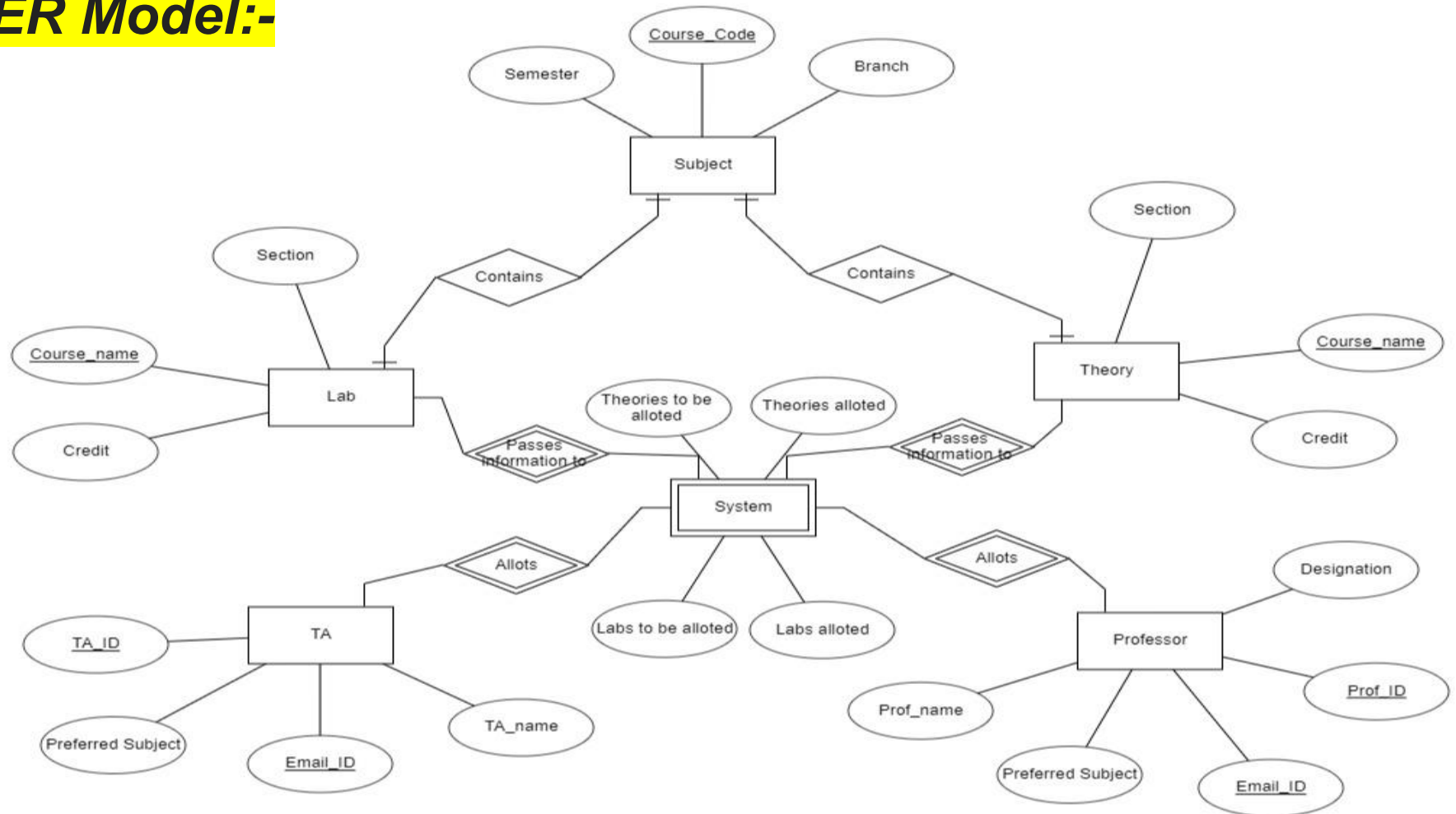
Kewal Delhiwala (202051060)

Vikram Kumar (202052346)

TA:-Amit Dwivedi



ER Model:-



EER Model Concepts Used:-

1- Sub Class and Super Class

2- Specialization and Generalization

3- Attribute and relationship inheritance

4-Aggregation



Steps followed to convert ER and EER model concepts to the Relational Model:-

1- Mapping Entity

Mapping Process (Algorithm)

- Create table for each entity.
- Entity's attributes should become fields of tables with their respective data types.
- Declare primary key.



2- Mapping Relationship

Mapping Process

- Create table for a relationship.
- Add the primary keys of all participating Entities as fields of table with their respective data types.
- If relationship has any attribute, add each attribute as field of table.
- Declare a primary key composing all the primary keys of participating entities.
- Declare all foreign key constraints.

4- Mapping Hierarchical Entities

Mapping Process

- Create tables for all higher-level entities.
- Create tables for lower-level entities.
- Add primary keys of higher-level entities in the table of lower-level entities.
- In lower-level tables, add all other attributes of lower-level entities.
- Declare primary key of higher-level table and the primary key for lower-level table.
- Declare foreign key constraints.

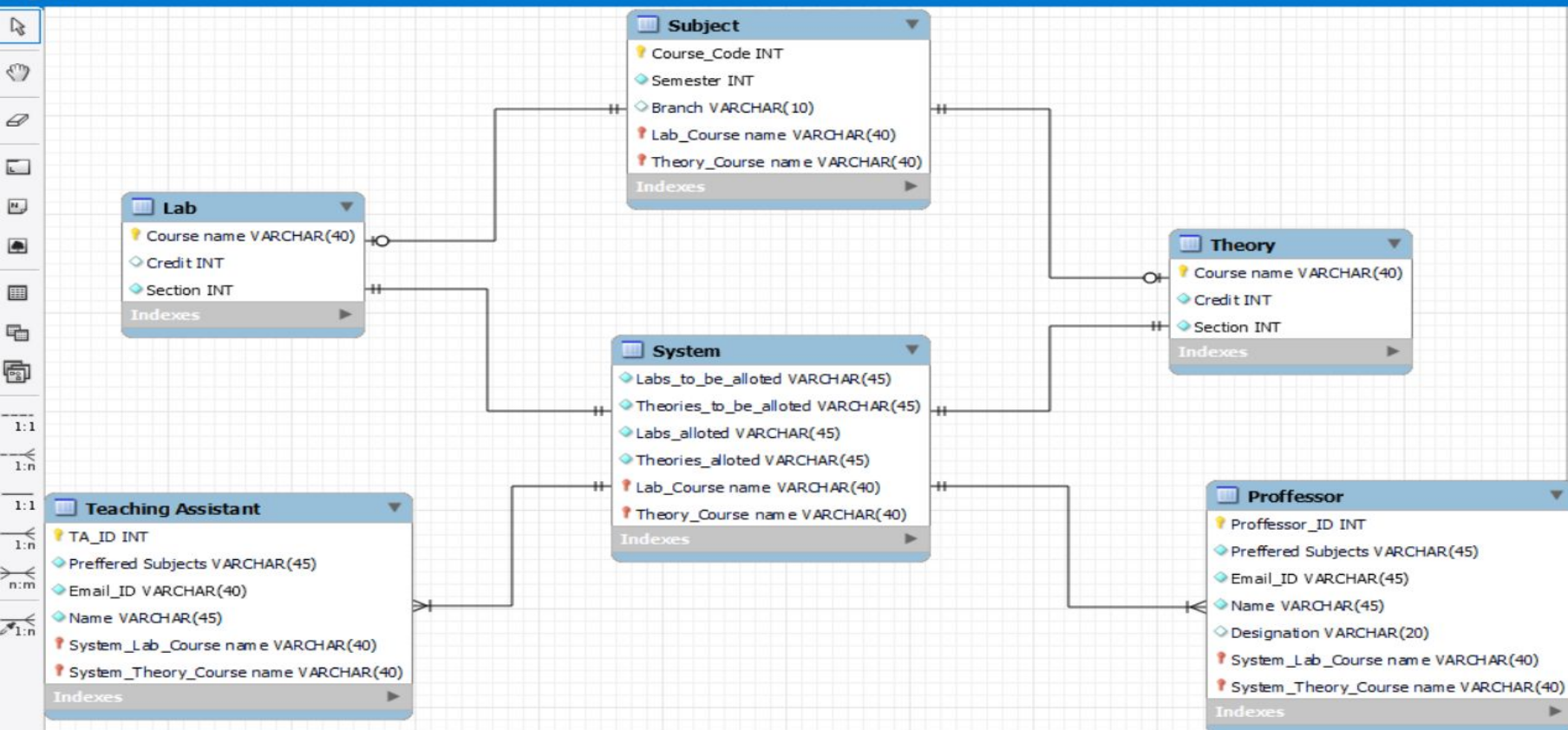
3-Mapping Weak Entity Sets

Mapping Process

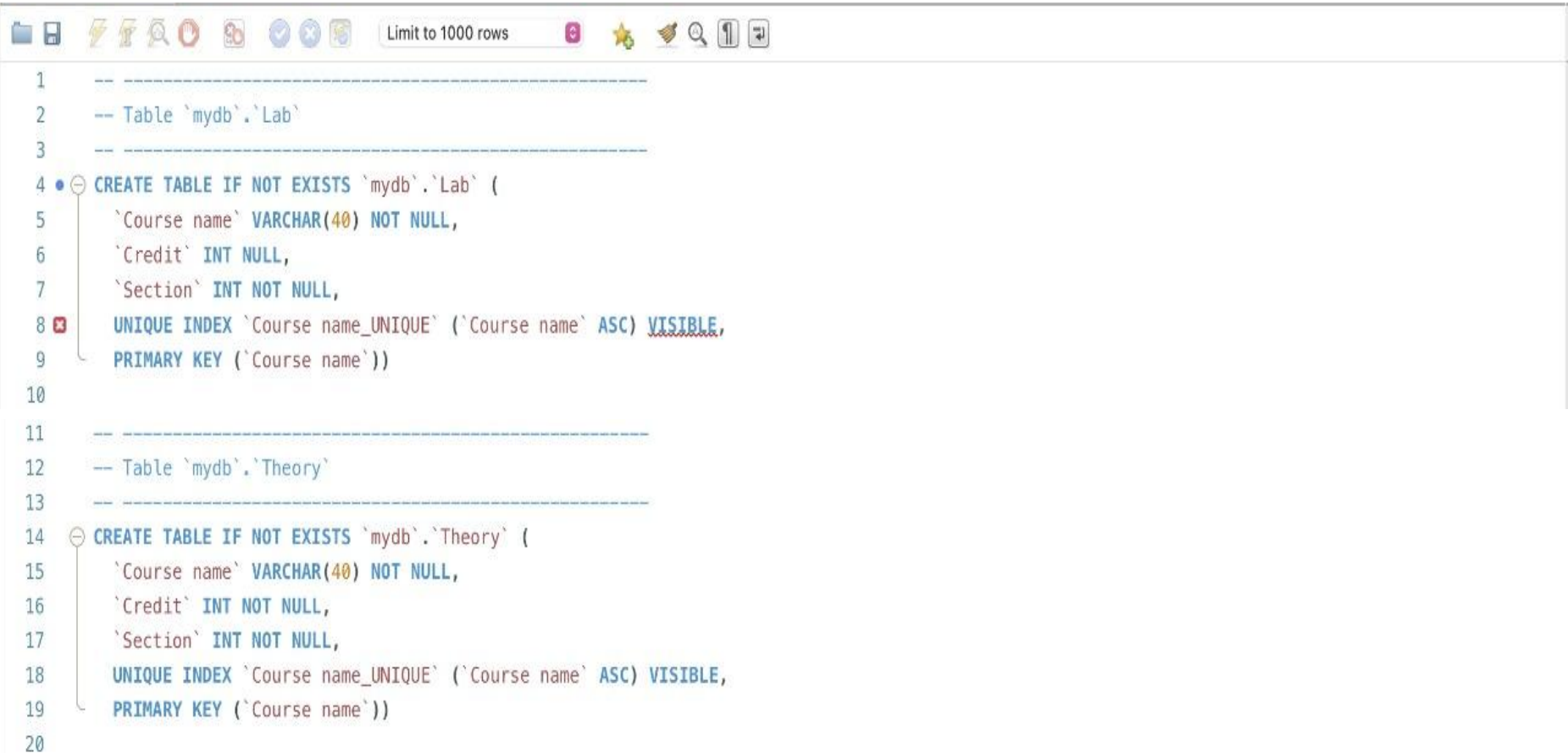
- **Create table for weak entity set.**
- **Add all its attributes to table as field.**
- **Add the primary key of identifying entity set.**
- **Declare all foreign key constraints.**

Relational Schema Diagram

Diagram



Screenshots of all the queries executed to implement your Relational Model



The screenshot shows a SQL IDE interface with a toolbar at the top containing icons for file operations, execution, and search. A status bar indicates 'Limit to 1000 rows'. The main editor area displays two SQL queries. The first query, on lines 4-9, creates a table named 'Lab' with columns 'Course name' (VARCHAR(40), NOT NULL), 'Credit' (INT, NULL), and 'Section' (INT, NOT NULL). It also includes a UNIQUE INDEX 'Course name_UNIQUE' on the 'Course name' column, which is marked as VISIBLE and underlined. The second query, on lines 14-19, creates a table named 'Theory' with the same column definitions and index structure as the 'Lab' table. Both queries are preceded by a comment line indicating the table name. The IDE interface includes a sidebar on the left with a tree view showing the database structure.

```
1  -----
2  -- Table `mydb`.`Lab`
3  -----
4  CREATE TABLE IF NOT EXISTS `mydb`.`Lab` (
5      `Course name` VARCHAR(40) NOT NULL,
6      `Credit` INT NULL,
7      `Section` INT NOT NULL,
8      UNIQUE INDEX `Course name_UNIQUE` (`Course name` ASC) VISIBLE,
9      PRIMARY KEY (`Course name`))
10
11  -----
12  -- Table `mydb`.`Theory`
13  -----
14  CREATE TABLE IF NOT EXISTS `mydb`.`Theory` (
15      `Course name` VARCHAR(40) NOT NULL,
16      `Credit` INT NOT NULL,
17      `Section` INT NOT NULL,
18      UNIQUE INDEX `Course name_UNIQUE` (`Course name` ASC) VISIBLE,
19      PRIMARY KEY (`Course name`))
20
```

```

22  -----
23  -- Table `mydb`.`System`
24  -----
25  CREATE TABLE IF NOT EXISTS `mydb`.`System` (
26      `Labs_to_be_alloted` VARCHAR(45) NOT NULL,
27      `Theories_to_be_alloted` VARCHAR(45) NOT NULL,
28      `Labs_alloted` VARCHAR(45) NOT NULL,
29      `Theories_alloted` VARCHAR(45) NOT NULL,
30      `Lab_Course name` VARCHAR(40) NOT NULL,
31      `Theory_Course name` VARCHAR(40) NOT NULL,
32      UNIQUE INDEX `Subjects_to_allot_UNIQUE` (`Labs_to_be_alloted` ASC) VISIBLE,
33      UNIQUE INDEX `Subjects_alloted_UNIQUE` (`Theories_to_be_alloted` ASC) VISIBLE,
34      UNIQUE INDEX `Labs_alloted_UNIQUE` (`Labs_alloted` ASC) VISIBLE,
35      UNIQUE INDEX `Theories_alloted_UNIQUE` (`Theories_alloted` ASC) VISIBLE,
36      PRIMARY KEY (`Lab_Course name`, `Theory_Course name`),
37      INDEX `fk_System_Theory1_idx` (`Theory_Course name` ASC) VISIBLE,
38      CONSTRAINT `fk_System_Lab1`
39          FOREIGN KEY (`Lab_Course name`)
40          REFERENCES `mydb`.`Lab` (`Course name`)
41          ON DELETE NO ACTION
42          ON UPDATE NO ACTION,
43      CONSTRAINT `fk_System_Theory1`
44          FOREIGN KEY (`Theory_Course name`)
45          REFERENCES `mydb`.`Theory` (`Course name`)
46          ON DELETE NO ACTION
47          ON UPDATE NO ACTION)

```

```
48  -----
49  -- Table `mydb`.`Teaching Assistant`
50  -----
51  CREATE TABLE IF NOT EXISTS `mydb`.`Teaching Assistant` (
52    `TA_ID` INT NOT NULL,
53    `Preferred Subjects` VARCHAR(45) NOT NULL,
54    `Email_ID` VARCHAR(40) NOT NULL,
55    `Name` VARCHAR(45) NOT NULL,
56    `System_Lab_Course name` VARCHAR(40) NOT NULL,
57    `System_Theory_Course name` VARCHAR(40) NOT NULL,
58    PRIMARY KEY (`TA_ID`, `System_Lab_Course name`, `System_Theory_Course name`),
59    UNIQUE INDEX `TA_ID_UNIQUE` (`TA_ID` ASC) VISIBLE,
60    UNIQUE INDEX `Email_ID_UNIQUE` (`Email_ID` ASC) VISIBLE,
61    INDEX `fk_Teaching Assistant_System1_idx` (`System_Lab_Course name` ASC, `System_Theory_Course name` ASC) VISIBLE,
62    CONSTRAINT `fk_Teaching Assistant_System1`
63      FOREIGN KEY (`System_Lab_Course name`, `System_Theory_Course name`)
64      REFERENCES `mydb`.`System` (`Lab_Course name`, `Theory_Course name`)
65      ON DELETE NO ACTION
66      ON UPDATE NO ACTION)
67
```

```

68  -----
69  -- Table `mydb`.`Proffessor`
70  -----
71  CREATE TABLE IF NOT EXISTS `mydb`.`Proffessor` (
72      `Proffessor_ID` INT NOT NULL,
73      `Preffered Subjects` VARCHAR(45) NOT NULL,
74      `Email_ID` VARCHAR(45) NOT NULL,
75      `Name` VARCHAR(45) NOT NULL,
76      `Designation` VARCHAR(20) NULL,
77      `System_Lab_Course name` VARCHAR(40) NOT NULL,
78      `System_Theory_Course name` VARCHAR(40) NOT NULL,
79      PRIMARY KEY (`Proffessor_ID`, `System_Lab_Course name`, `System_Theory_Course name`),
80      UNIQUE INDEX `Proffessor_ID_UNIQUE` (`Proffessor_ID` ASC) VISIBLE,
81      UNIQUE INDEX `Email_ID_UNIQUE` (`Email_ID` ASC) VISIBLE,
82      INDEX `fk_Proffessor_System1_idx` (`System_Lab_Course name` ASC, `System_Theory_Course name` ASC) VISIBLE,
83      CONSTRAINT `fk_Proffessor_System1`
84          FOREIGN KEY (`System_Lab_Course name`, `System_Theory_Course name`)
85          REFERENCES `mydb`.`System` (`Lab_Course name`, `Theory_Course name`)
86          ON DELETE NO ACTION
87          ON UPDATE NO ACTION)

```

```
90  -----
91  -- Table `mydb`.`Subject`
92  -----
93  CREATE TABLE IF NOT EXISTS `mydb`.`Subject` (
94      `Course_Code` INT NOT NULL,
95      `Semester` INT NOT NULL,
96      `Branch` VARCHAR(10) NOT NULL,
97      `Lab_Course name` VARCHAR(40) NOT NULL,
98      `Theory_Course name` VARCHAR(40) NOT NULL,
99      PRIMARY KEY (`Course_Code`, `Lab_Course name`, `Theory_Course name`),
100     UNIQUE INDEX `Course_Code_UNIQUE` (`Course_Code` ASC) VISIBLE,
101     INDEX `fk_Subject_Lab_idx` (`Lab_Course name` ASC) VISIBLE,
102     INDEX `fk_Subject_Theory1_idx` (`Theory_Course name` ASC) VISIBLE,
103     CONSTRAINT `fk_Subject_Lab`
104         FOREIGN KEY (`Lab_Course name`)
105         REFERENCES `mydb`.`Lab` (`Course name`)
106         ON DELETE NO ACTION
107         ON UPDATE NO ACTION,
108     CONSTRAINT `fk_Subject_Theory1`
109         FOREIGN KEY (`Theory_Course name`)
110         REFERENCES `mydb`.`Theory` (`Course name`)
111         ON DELETE NO ACTION
112         ON UPDATE NO ACTION)
```

Insert the queries



```
Query 1 x
1 • USE mydb;
2 • INSERT INTO lab (`Course name`, `Credit`, `Section`)
3 VALUES ('CS204', 4, 1);
4 |
```

Select the queries

The screenshot shows a database query editor window titled "Query 1". The editor contains two queries:

- 1 • `USE mydb;`
- 2 • `SELECT * FROM lab;`

The toolbar includes icons for file operations, execution, and a "Limit to 1000 rows" dropdown. Below the queries, the "Result Grid" is visible, showing a single row of data with three columns: "Course name", "Credit", and "Section". The data in the row is "NULL", "NULL", and "NULL" respectively.

	Course name	Credit	Section
*	NULL	NULL	NULL

Select the queries

The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons for file operations, execution, and editing. Below the toolbar, two queries are listed:

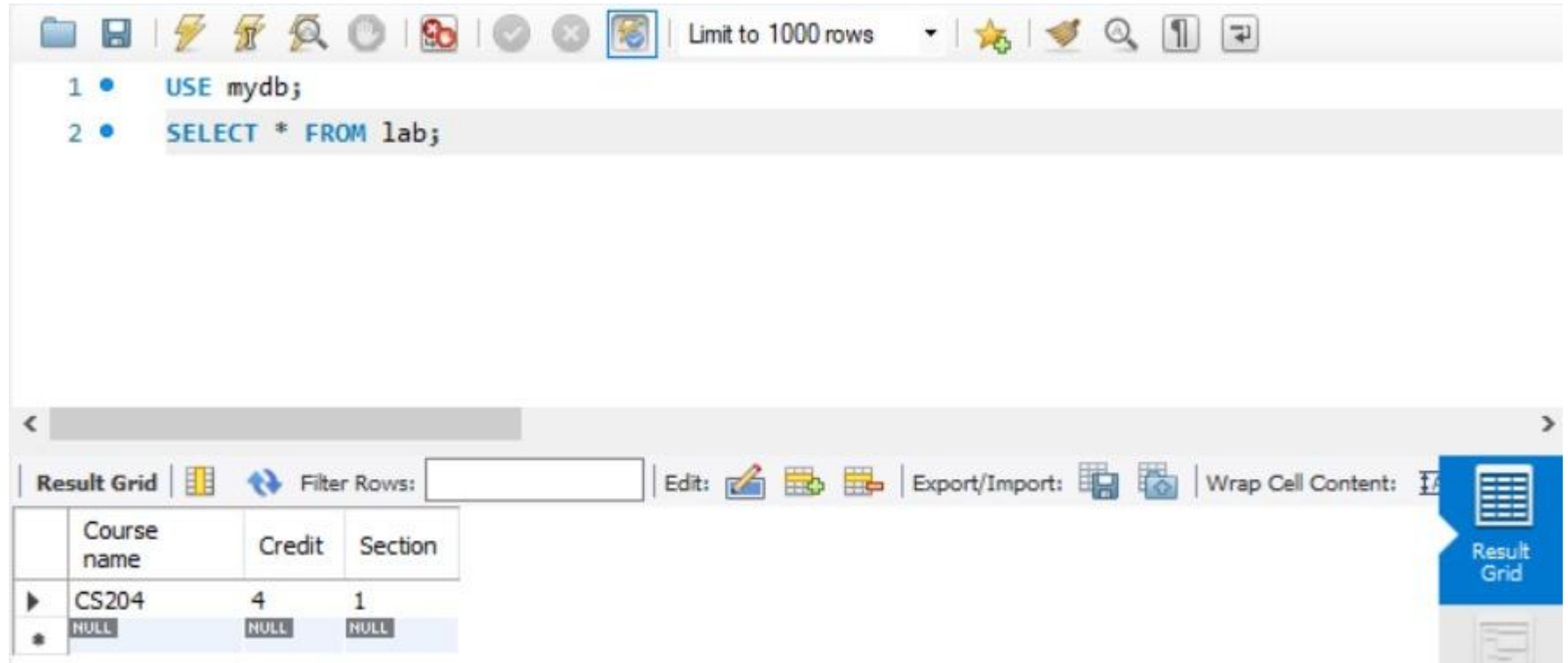
- 1 • `USE mydb;`
- 2 • `SELECT * FROM lab;`

Below the queries, there is a horizontal scrollbar. At the bottom, there is a toolbar with options for the result grid, filtering, editing, and exporting. The result grid is displayed below the toolbar, showing the following data:

	Course name	Credit	Section
▶	CS204	3	1
★	NULL	NULL	NULL

On the right side of the interface, there is a vertical toolbar with two buttons: "Result Grid" (highlighted in blue) and a button with a document icon.

Select the queries



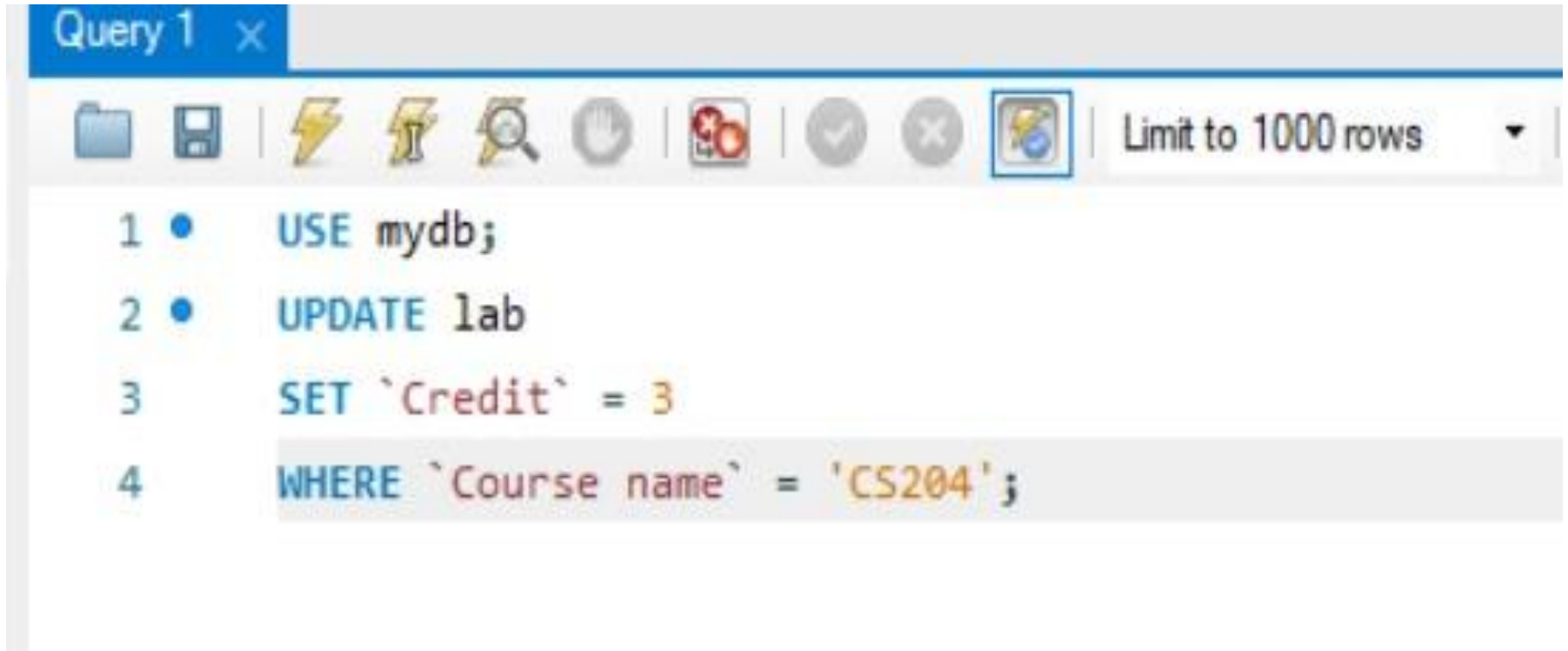
The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons for file operations, execution, and editing. Below the toolbar, two queries are listed:

- 1 • `USE mydb;`
- 2 • `SELECT * FROM lab;`

Below the queries, there is a "Result Grid" section. It includes a "Filter Rows:" input field and a "Result Grid" button. The result grid itself displays the following data:

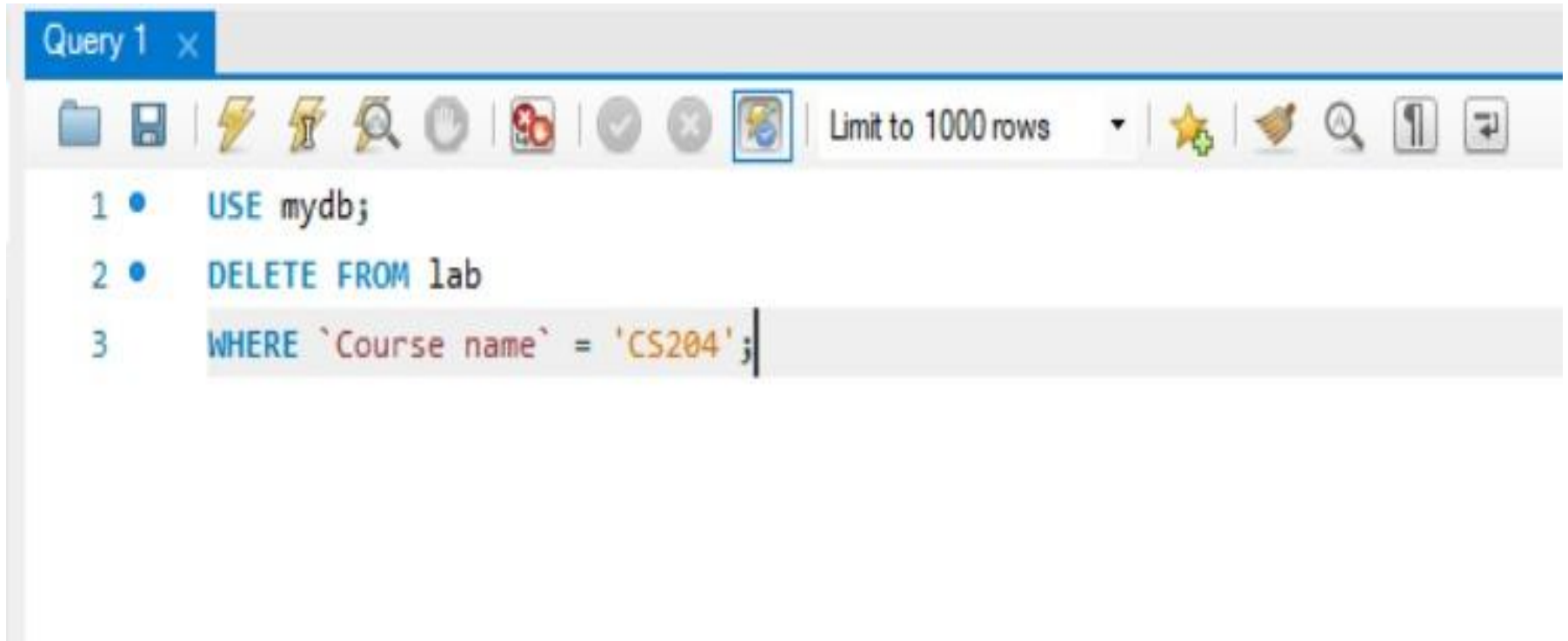
	Course name	Credit	Section
▶	CS204	4	1
*	NULL	NULL	NULL

Update the queries



```
Query 1 x
1 • USE mydb;
2 • UPDATE lab
3   SET `Credit` = 3
4   WHERE `Course name` = 'CS204';
```

Delete the queries



```
Query 1 x
1 • USE mydb;
2 • DELETE FROM lab
3 WHERE 'Course name' = 'CS204';
```

Output

Action Output

#	Time	Action	Message
✓ 1	22:39:45	USE mydb	0 row(s) affected
✓ 2	22:39:45	INSERT INTO lab ('Course name', 'Credit', 'Section') VALUES ('CS204', 4, 1)	1 row(s) affected
✓ 3	22:40:02	USE mydb	0 row(s) affected
✗ 4	22:40:02	INSERT INTO lab ('Course name', 'Credit', 'Section') VALUES ('CS204', 4, 1)	Error Code: 1062. Duplicate entry 'CS204' for key lab.PRIMARY
✓ 5	22:41:44	USE mydb	0 row(s) affected
✓ 6	22:41:44	SELECT * FROM lab LIMIT 0, 1000	1 row(s) returned
✓ 7	22:44:46	USE mydb	0 row(s) affected
✓ 8	22:44:46	UPDATE lab SET 'Credit' = 3 WHERE 'Course name' = 'CS204'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
✓ 9	22:45:30	USE mydb	0 row(s) affected
✓ 10	22:45:30	SELECT * FROM lab LIMIT 0, 1000	1 row(s) returned
✓ 11	22:46:29	USE mydb	0 row(s) affected
✓ 12	22:46:29	DELETE FROM lab WHERE 'Course name' = 'CS204'	1 row(s) affected
✓ 13	22:46:57	USE mydb	0 row(s) affected