“Heaven's Light is Our Guide”

Rajshahi University of Engineering & Technology

Department of Electrical & Computer Engineering

Course Title

Data Base Systems Sessional

Course No: ECE 2216

Lab Report No. 01

Date of submission: 16/09/2024

Submitted to:	**Submitted by:**
Oishi Jyoti Assistant Professor, Department of ECE, RUET	Md. Sabbir Hosen Roll: 2110019 Reg. No: 1073/2021-2022 Department of ECE, RUET

Experiment No. : 01

Experiment Name : Introduction to MySQL

Task 1 : Create a database system for class 21 where there are two table for odd and even batch containing following information for 10 students.

- `ID`
- `Name`
- `Contact`
- `Blood Group`
- `Major Subject`
- `Obtained Marks` (Out of 100)

Objectives :

The objective of this lab is to design and implement a database system for managing student information for Class 21. The class is divided into two batches: odd and even, based on student ID. The database will consist of two tables to store relevant information such as student ID, name, contact, blood group, major subject, and marks obtained out of 100.

Database Structure

The database will contain two tables:

1. **Odd Batch**: Contains data for students with odd-numbered IDs.
2. **Even Batch**: Contains data for students with even-numbered IDs.

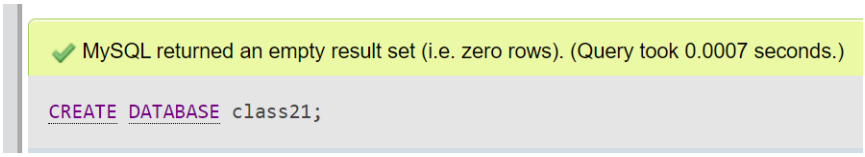
Each table will have the following fields:

- `ID` (Primary Key)
- `Name`
- `Contact`
- `Blood Group`
- `Major Subject`
- `Obtained Marks` (out of 100)

Queries & Output :

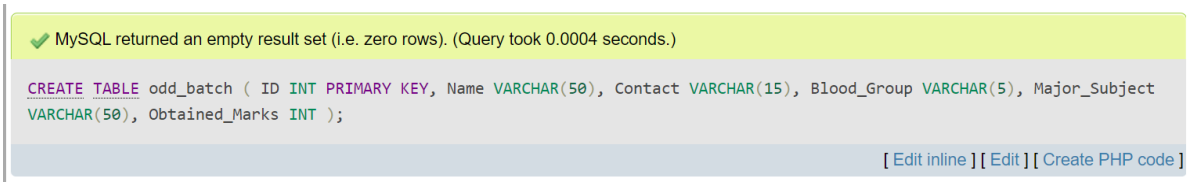
1.Create Database:

```
CREATE DATABASE class21;
```



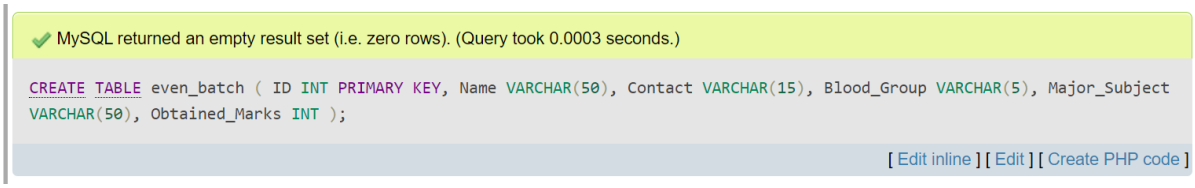
2.Create Odd Batch Table:

```
CREATE TABLE odd_batch (
    ID INT PRIMARY KEY,
    Name VARCHAR(50),
    Contact VARCHAR(15),
    Blood_Group VARCHAR(5),
    Major_Subject VARCHAR(50),
    Obtained_Marks INT
);
```



3.Create Even Batch Table:

```
CREATE TABLE even_batch (
    ID INT PRIMARY KEY,
    Name VARCHAR(50),
    Contact VARCHAR(15),
    Blood_Group VARCHAR(5),
    Major_Subject VARCHAR(50),
    Obtained_Marks INT
);
```



4.Insert into Odd Batch

```
INSERT INTO odd_batch (ID, Name, Contact, Blood_Group, Major_Subject, Obtained_Marks) VALUES
(1, 'Sadik', '1234567890', 'A+', 'ECE 1203', 85),
(3, 'Samia', '0987654321', 'B+', 'ECE 1203', 84),
```

```
(5, 'Hridoy', '9876543210', 'O-', 'ECE 1203', 82),
(7, 'Himel', '5678901234', 'AB+', 'ECE 1203', 95),
(9, 'Prithu', '6789012345', 'A-', 'ECE 1203', 75);
```

5 rows inserted. (Query took 0.0028 seconds.)

```
INSERT INTO odd_batch (ID, Name, Contact, Blood_Group, Major_Subject, Obtained_Marks) VALUES (1, 'Sadik', '1234567890',
'A+', 'ECE 1203', 85), (3, 'Samia', '0987654321', 'B+', 'ECE 1203', 84), (5, 'Hridoy', '9876543210', 'O-', 'ECE 1203',
82), (7, 'Himel', '5678901234', 'AB+', 'ECE 1203', 95), (9, 'Prithu', '6789012345', 'A-', 'ECE 1203', 75);
```

[Edit inline] [Edit] [Create PHP code]

4.Insert into Even Batch

```
INSERT INTO even_batch (ID, Name, Contact, Blood_Group, Major_Subject,
Obtained_Marks) VALUES
(2, 'Nahid', '1122334455', 'O+', 'ECE 1203', 89),
(4, 'Radia', '2233445566', 'A-', 'ECE 1203', 74),
(6, 'Jahin', '3344556677', 'B+', 'ECE 1203', 80),
(8, 'Zanifa', '4455667788', 'AB-', 'ECE 1203', 82),
(10, 'Rubaid', '5566778899', 'O+', 'ECE 1203', 90);
```

5 rows inserted. (Query took 0.0004 seconds.)

```
INSERT INTO even_batch (ID, Name, Contact, Blood_Group, Major_Subject, Obtained_Marks) VALUES (2, 'Nahid', '1122334455',
'O+', 'ECE 1203', 89), (4, 'Radia', '2233445566', 'A-', 'ECE 1203', 74), (6, 'Jahin', '3344556677', 'B+', 'ECE 1203', 80),
(8, 'Zanifa', '4455667788', 'AB-', 'ECE 1203', 82), (10, 'Rubaid', '5566778899', 'O+', 'ECE 1203', 90);
```

[Edit inline] [Edit] [Create PHP code]

5. DELETE Operation : Delete a record from the odd_batch table.

```
DELETE FROM odd_batch WHERE ID = 5;
```

Before Deletion

	ID	Name	Contact	Blood_Group	Major_Subject	Obtained_Marks
Edit Copy Delete	1	Sadik	1234567890	A+	ECE 1203	85
Edit Copy Delete	3	Samia	0987654321	B+	ECE 1203	84
Edit Copy Delete	5	Hridoy	9876543210	O-	ECE 1203	82
Edit Copy Delete	7	Himel	5678901234	AB+	ECE 1203	95
Edit Copy Delete	9	Prithu	6789012345	A-	ECE 1203	75

After Deletion

ID	Name	Contact	Blood_Group	Major_Subject	Obtained_Marks
1	Sadik	1234567890	A+	ECE 1203	85
3	Samia	0987654321	B+	ECE 1203	84
7	Himel	5678901234	AB+	ECE 1203	95
9	Prithu	6789012345	A-	ECE 1203	75

5. UPDATE Operation : Update the contact number and marks for a student in the even_batch.

```
UPDATE even_batch
SET Contact = '9988776655', Obtained_Marks = 95
WHERE ID = 2;
```

Before Updating

ID	Name	Contact	Blood_Group	Major_Subject	Obtained_Marks
2	Nahid	1122334455	O+	ECE 1203	89
4	Radia	2233445566	A-	ECE 1203	74
6	Jahin	3344556677	B+	ECE 1203	80
8	Zanifa	4455667788	AB-	ECE 1203	82
10	Rubaid	5566778899	O+	ECE 1203	90

After Updating

ID	Name	Contact	Blood_Group	Major_Subject	Obtained_Marks
2	Nahid	9988776655	O+	ECE 1203	95
4	Radia	2233445566	A-	ECE 1203	74
6	Jahin	3344556677	B+	ECE 1203	80
8	Zanifa	4455667788	AB-	ECE 1203	82
10	Rubaid	5566778899	O+	ECE 1203	90

6. ALTER Operation: Add a new column Email to the `odd_batch` table.

```
ALTER TABLE odd_batch
ADD Email VARCHAR(50);
```

Before Alteration:

	ID	Name	Contact	Blood_Group	Major_Subject	Obtained_Marks
Edit Copy Delete	1	Sadik	1234567890	A+	ECE 1203	85
Edit Copy Delete	3	Samia	0987654321	B+	ECE 1203	84
Edit Copy Delete	7	Himel	5678901234	AB+	ECE 1203	95
Edit Copy Delete	9	Prithu	6789012345	A-	ECE 1203	75

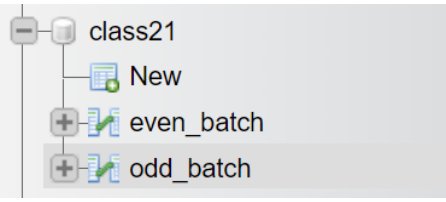
After Alteration:

	ID	Name	Contact	Blood_Group	Major_Subject	Obtained_Marks	Email
Edit Copy Delete	1	Sadik	1234567890	A+	ECE 1203	85	*NULL*
Edit Copy Delete	3	Samia	0987654321	B+	ECE 1203	84	*NULL*
Edit Copy Delete	7	Himel	5678901234	AB+	ECE 1203	95	*NULL*
Edit Copy Delete	9	Prithu	6789012345	A-	ECE 1203	75	*NULL*

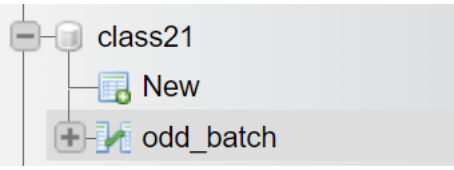
7. DROP Operation: Drop the `even_batch` table completely from the database.

```
DROP TABLE even_batch;
```

Before Dropping :



After Dropping :



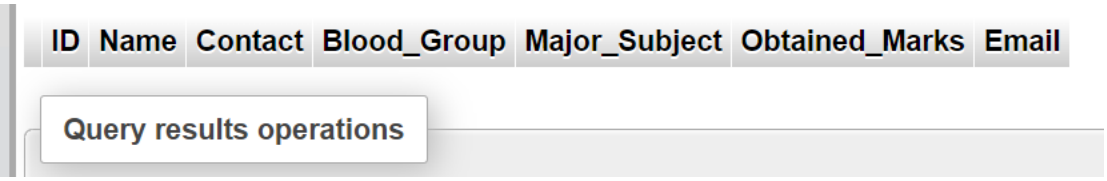
8. TRUNCATE Operation: Remove all data from the `odd_batch` table without deleting the table structure.

```
TRUNCATE TABLE odd_batch;
```

Before Truncating :

	ID	Name	Contact	Blood_Group	Major_Subject	Obtained_Marks	Email
Edit Copy Delete	1	Sadik	1234567890	A+	ECE 1203	85	NULL
Edit Copy Delete	3	Samia	0987654321	B+	ECE 1203	84	NULL
Edit Copy Delete	7	Himel	5678901234	AB+	ECE 1203	95	NULL
Edit Copy Delete	9	Prithu	6789012345	A-	ECE 1203	75	NULL

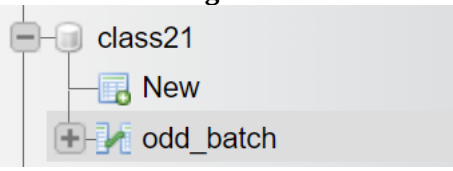
After Truncating :



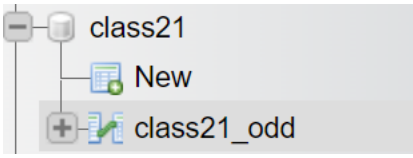
9. RENAME Operation: Rename the `odd_batch` table to `class21_odd`.

```
RENAME TABLE odd_batch TO class21_odd;
```

Before Renaming :



After Renaming :



Discussion:

In this lab, a database system was successfully designed and implemented for Class 21, segregating the students into odd and even batches. The database tables were created with fields such as ID, Name, Contact, Blood Group, Major Subject, and Obtained Marks. Fundamental SQL operations including inserting, updating, deleting, altering, truncating, and renaming were applied effectively to manage the data.

The use of SQL for structuring and manipulating data demonstrated the flexibility and power of relational databases in handling student records. The **CREATE**, **INSERT**, **DELETE**, **UPDATE**, and other operations showed how specific data management needs could be addressed efficiently. For instance, when records were updated or deleted based on specific criteria, the queries performed these operations smoothly without requiring changes to the entire structure of the database.

Through this exercise, it was evident that:

- The **ALTER** command provides a convenient way to extend the database schema by adding columns like `Email` without having to recreate the table.
- The **DELETE** command allowed targeted removal of records without disturbing the table structure, while **TRUNCATE** was used to clear all records, maintaining the schema intact for future insertions.
- The **RENAME** command proved useful in renaming tables for better organization and clarity.
- The **DROP** command is effective for permanent deletion of the entire table when no longer needed.

By designing and implementing a system with these operations, a strong understanding of database management has been achieved. Such systems are essential in academic institutions for handling student data, and the SQL commands utilized can be scaled for larger datasets and more complex requirements. The database system created is highly modular and scalable, which provides the flexibility to adapt to future needs, such as adding more batches, fields, or students.

References

[1] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed., Pearson, 2016.

[2] H. Garcia-Molina, J. D. Ullman, and J. Widom, *Database Systems: The Complete Book*, 2nd ed., Pearson, 2008.

[3] "MySQL Introduction," W3Schools, [Online]. Available: https://www.w3schools.com/mysql/mysql_intro.asp. [Accessed: 16-Sep-2024].

[4] "MySQL Tutorial," JavaTpoint, [Online]. Available: https://www.javatpoint.com/mysql-tutorial. [Accessed: 16-Sep-2024].