



Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

Experiment-2

Experiment Name:

Data Definition & Manipulation (DDL & DML)

Objective:

To create tables using DDL commands and perform data insertion, updation, and deletion using DML commands

Prerequisites:

- Basic knowledge of SQL syntax.
- Understanding of database concepts (tables, rows, columns).

Key Terms: SQL Data types

System requirements:

- 4-8 GB RAM, 10-15 disk space, Windows OS
- MySQL Workbench 8.0 CE

Theory and Application:

SQL Database Data Types

DATA TYPES represents the type of data an object is holding. Data Types are defined for columns of a table, local/global variables, input/output arguments of procedures etc..

Each database system (MS SQL Server, MYSQL, DB2, Oracle etc.) have its own long list of data types but several data types are common in most of them. This article will list down common data types across various database systems.

Number Data Type

Few numeric data type has syntax of data_type(x). Here x is meant for precision value.

Date Time Data Type

- **datetime:** This data type stores both date and time values together. It supports dates ranging from **01/01/1753** to **12/31/9999** and requires **8 bytes** of storage. In some database systems, this data type is also referred to as a **timestamp**.
- **date:** This data type is used to store only the date component, without any time information.



Chandigarh Engineering College Jhanjeri

Mohali-140307

Department of Artificial Intelligence (AI) and Data Sciences

- **time:** This data type stores only time-related information, without including the date.
- Some numeric data types follow the syntax **data_type(x)**, where **x** specifies the precision or size of the value.

String Data Type

- **char(x):** This data type stores fixed-length character data and is padded with spaces to match the specified length. Here, **x** defines the exact number of characters to be stored.
- **varchar(x):** Short for *variable characters*, this data type stores character data of variable length and does not add extra spaces. The value of **x** specifies the maximum number of characters allowed.
- **text:** This data type is used to store large or lengthy textual data.
- Some numeric data types use the syntax **data_type(x)**, where **x** represents the precision of the value.

Other Data Type

- **blob:** Binary Large Object. This type is used to store large amount of binary data such as images or other type of files.
- **money:** In few databases also termed as currency. The type is used to storage money/currency information
- **binary:** The data type is used to store information in binary string format.



Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

1. CREATE & USE DATABASE

Syntax

CREATE DATABASE Db3;

USE Db3;

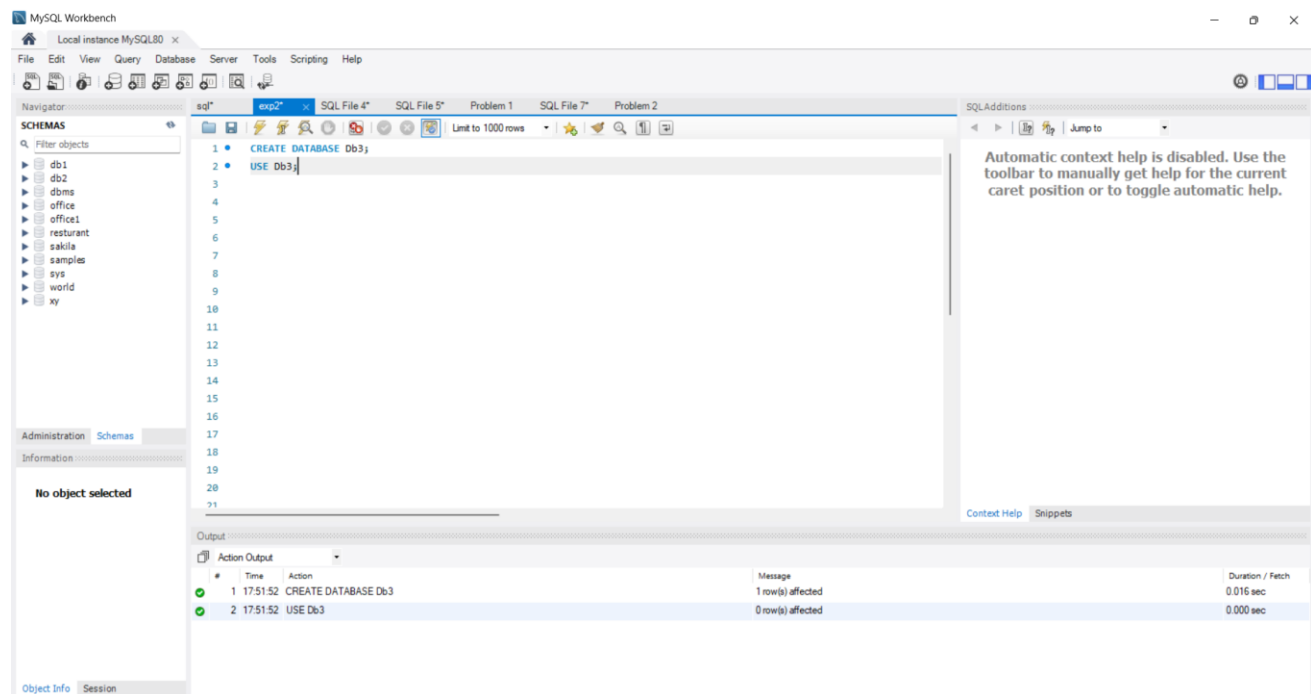


Figure 1: CREATE & USE Database



Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

2. CREATE TABLE

Syntax

```
CREATE TABLE building (  
    building_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(50) NOT NULL UNIQUE,  
    location VARCHAR(100) DEFAULT 'Unknown'  
);
```

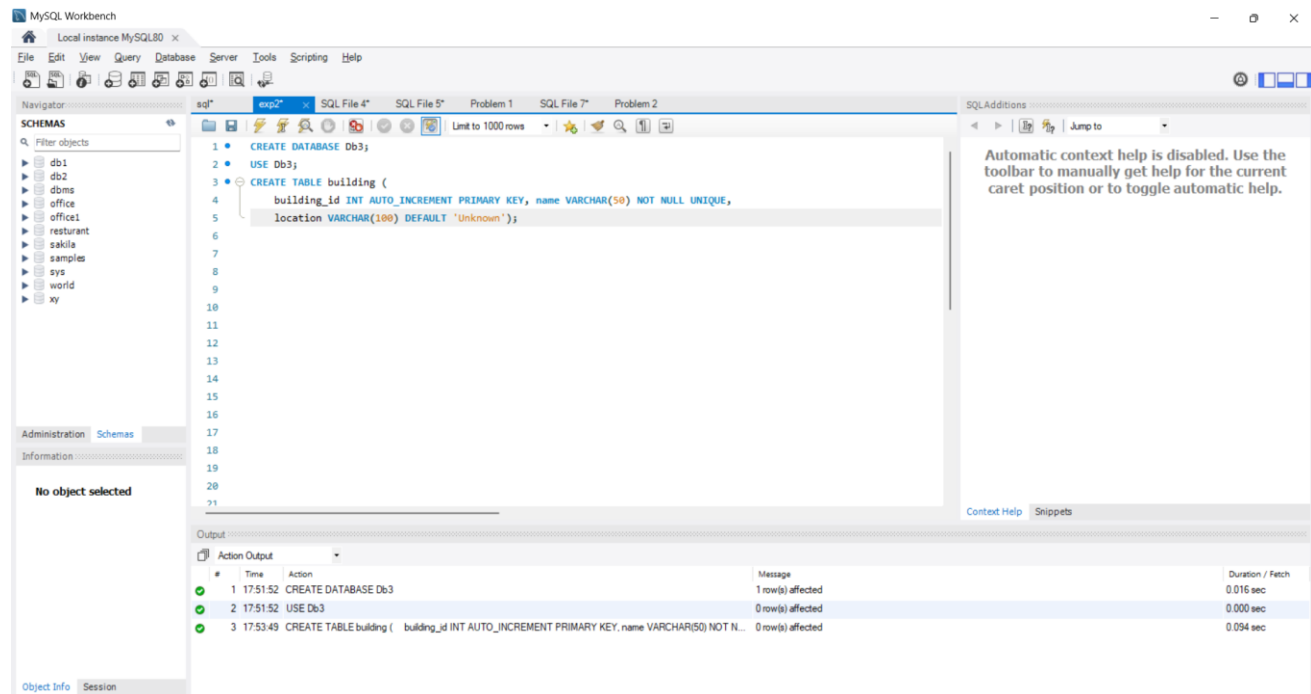


Figure 2: CREATE TABLE

Creates building table with PRIMARY KEY, AUTO_INCREMENT, NOT NULL, UNIQUE, DEFAULT value.



Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

```
CREATE TABLE room1 (  
room_id INT AUTO_INCREMENT PRIMARY KEY,  
chair INT NOT NULL,  
bench INT NOT NULL,  
is_active BOOLEAN DEFAULT TRUE,  
building_id INT,  
FOREIGN KEY (building_id) REFERENCES building(building_id)  
);
```

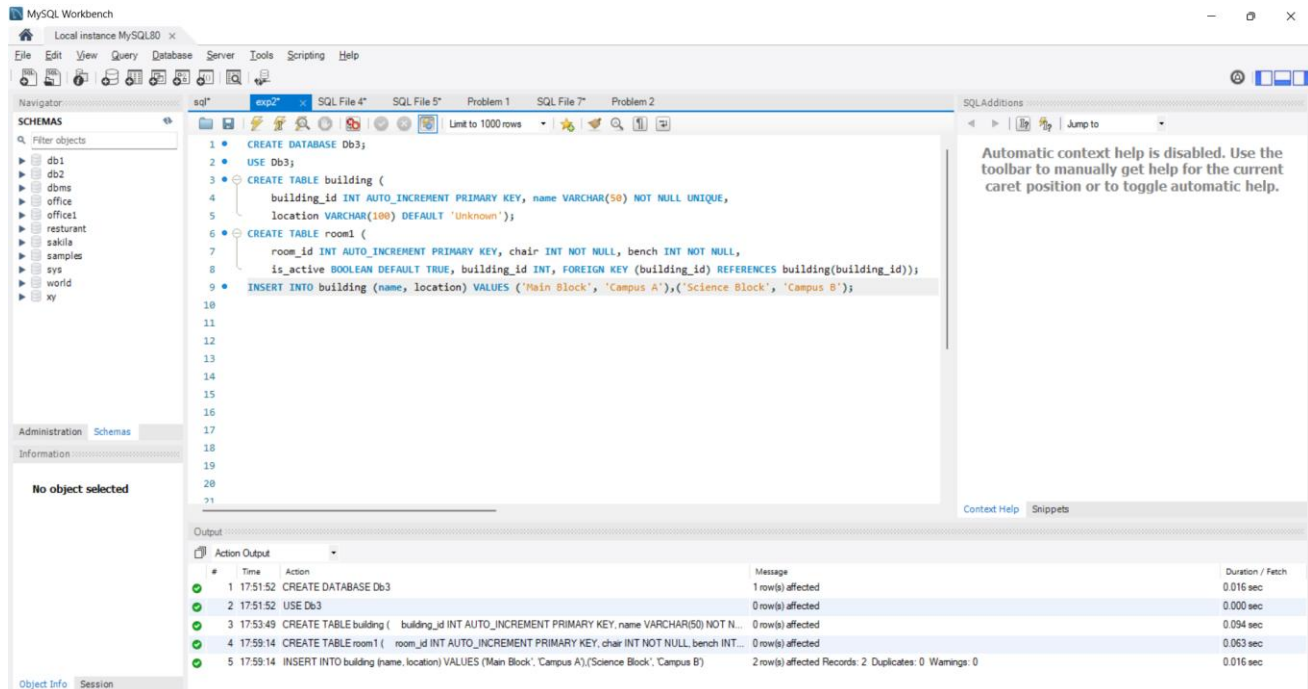


Figure 3: CREATE room1 TABLE

Creates room1 with NOT NULL, DEFAULT and a FOREIGN KEY referencing building.



Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

3. INSERT

Syntax

INSERT INTO building (name, location) VALUES

('Main Block', 'Campus A'),

('Science Block', 'Campus B');

INSERT INTO room1 (chair, bench, is_active, building_id) VALUES

(20, 10, TRUE, 1),

(15, 7, FALSE, 2);

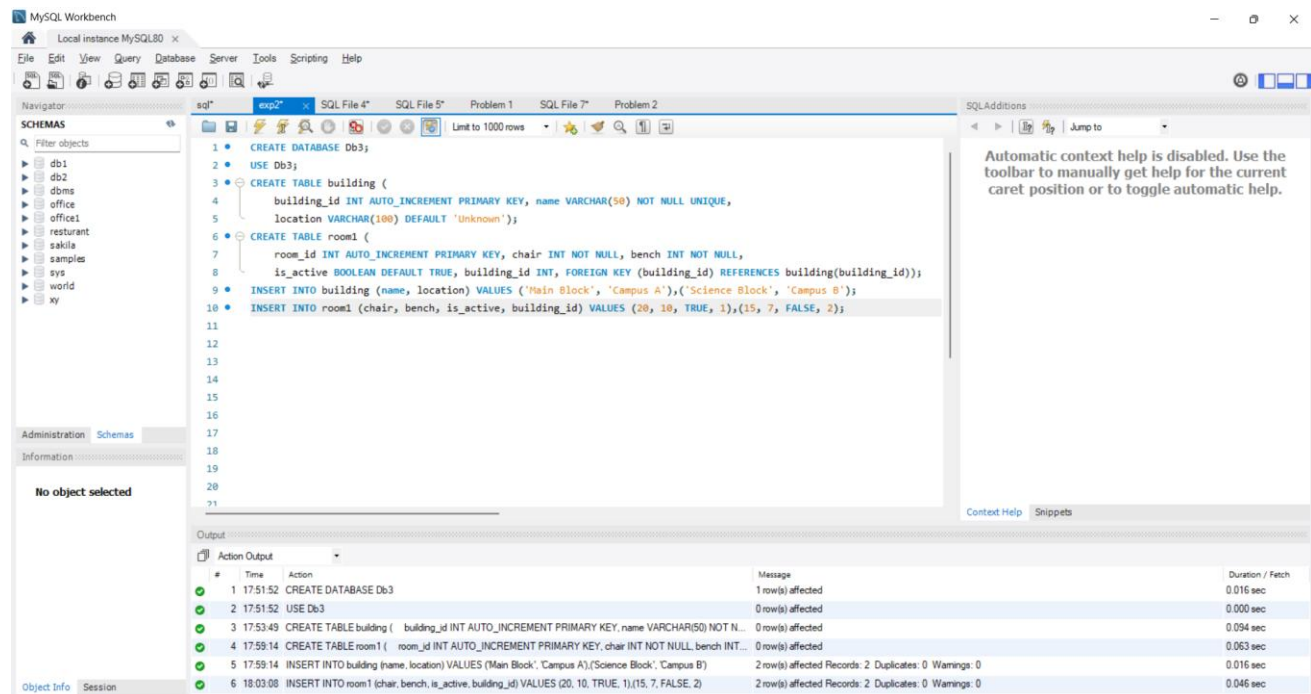


Figure 4: INSERT



Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

4. SELECT

Syntax

SELECT * FROM building;

SELECT * FROM room1;

The screenshot displays the MySQL Workbench interface. The SQL editor contains the following queries:

```
4 building_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(50) NOT NULL UNIQUE,  
5 location VARCHAR(100) DEFAULT 'Unknown');  
6 CREATE TABLE room1 (  
7 room_id INT AUTO_INCREMENT PRIMARY KEY, chair INT NOT NULL, bench INT NOT NULL,  
8 is_active BOOLEAN DEFAULT TRUE, building_id INT, FOREIGN KEY (building_id) REFERENCES building(building_id));  
9 INSERT INTO building (name, location) VALUES ('Main Block', 'Campus A'),('Science Block', 'Campus B');  
10 INSERT INTO room1 (chair, bench, is_active, building_id) VALUES (20, 10, TRUE, 1),(15, 7, FALSE, 2);  
11 SELECT * FROM building;  
12 SELECT * FROM room1;
```

The Results Grid shows the output of the SELECT queries:

room_id	chair	bench	is_active	building_id
1	20	10	1	1
2	15	7	0	2

The Action Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
3	17:53:49	CREATE TABLE building (building_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(50) NOT...	0 row(s) affected	0.094 sec
4	17:59:14	CREATE TABLE room1 (room_id INT AUTO_INCREMENT PRIMARY KEY, chair INT NOT NULL, bench I...	0 row(s) affected	0.063 sec
5	17:59:14	INSERT INTO building (name, location) VALUES ('Main Block', 'Campus A'),('Science Block', 'Campus B')	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.016 sec
6	18:03:08	INSERT INTO room1 (chair, bench, is_active, building_id) VALUES (20, 10, TRUE, 1),(15, 7, FALSE, 2)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.046 sec
7	18:06:10	SELECT * FROM building LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
8	18:06:10	SELECT * FROM room1 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Figure 5: SELECT



Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

5. UPDATE

Syntax

UPDATE room1 SET chair = 25 WHERE room_id = 1;

The screenshot displays the MySQL Workbench interface. The SQL editor contains the following script:

```
5 location VARCHAR(100) DEFAULT 'Unknown');
6 CREATE TABLE room1 (
7   room_id INT AUTO_INCREMENT PRIMARY KEY, chair INT NOT NULL, bench INT NOT NULL,
8   is_active BOOLEAN DEFAULT TRUE, building_id INT, FOREIGN KEY (building_id) REFERENCES building(building_id));
9 INSERT INTO building (name, location) VALUES ('Main Block', 'Campus A'),('Science Block', 'Campus B');
10 INSERT INTO room1 (chair, bench, is_active, building_id) VALUES (20, 10, TRUE, 1),(15, 7, FALSE, 2);
11 SELECT * FROM building;
12 SELECT * FROM room1;
13 UPDATE room1 SET chair = 25 WHERE room_id = 1;
```

The Result Grid shows the data after the update:

room_id	chair	bench	is_active	building_id
1	25	10	1	1
2	15	7	0	2

The Action Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
5	17:59:14	INSERT INTO building (name, location) VALUES ('Main Block', 'Campus A'),('Science Block', 'Campus B');	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.016 sec
6	18:03:08	INSERT INTO room1 (chair, bench, is_active, building_id) VALUES (20, 10, TRUE, 1),(15, 7, FALSE, 2);	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.046 sec
7	18:06:10	SELECT * FROM building LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
8	18:06:10	SELECT * FROM room1 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
9	18:07:26	UPDATE room1 SET chair = 25 WHERE room_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
10	18:07:36	SELECT * FROM room1 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Figure 6: UPDATE



Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

6. DELETE

Syntax:

DELETE FROM room1 WHERE room_id = 2;

The screenshot displays the MySQL Workbench interface. The SQL editor contains the following queries:

```
6 CREATE TABLE room1 (  
7   room_id INT AUTO_INCREMENT PRIMARY KEY, chair INT NOT NULL, bench INT NOT NULL,  
8   is_active BOOLEAN DEFAULT TRUE, building_id INT, FOREIGN KEY (building_id) REFERENCES building(building_id));  
9 INSERT INTO building (name, location) VALUES ('Main Block', 'Campus A'),('Science Block', 'Campus B');  
10 INSERT INTO room1 (chair, bench, is_active, building_id) VALUES (20, 10, TRUE, 1),(15, 7, FALSE, 2);  
11 SELECT * FROM building;  
12 SELECT * FROM room1;  
13 UPDATE room1 SET chair = 25 WHERE room_id = 1;  
14 DELETE FROM room1 WHERE room_id = 2;
```

The 'Result Grid' shows the data for the 'room1' table:

room_id	chair	bench	is_active	building_id
1	25	10	1	1

The 'Output' pane shows the execution results of the queries:

#	Time	Action	Message	Duration / Fetch
7	18:06:10	SELECT * FROM building LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
8	18:06:10	SELECT * FROM room1 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
9	18:07:26	UPDATE room1 SET chair = 25 WHERE room_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
10	18:07:36	SELECT * FROM room1 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
11	18:10:09	DELETE FROM room1 WHERE room_id = 2	1 row(s) affected	0.016 sec
12	18:10:15	SELECT * FROM room1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Figure 7: DELETE



Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

7. ALTER

Syntax:

ALTER TABLE room1 ADD COLUMN projector BOOLEAN DEFAULT FALSE;

ALTER TABLE room1 MODIFY COLUMN bench INT NOT NULL;

ALTER TABLE room1 ADD CONSTRAINT unique_room UNIQUE (chair, bench);

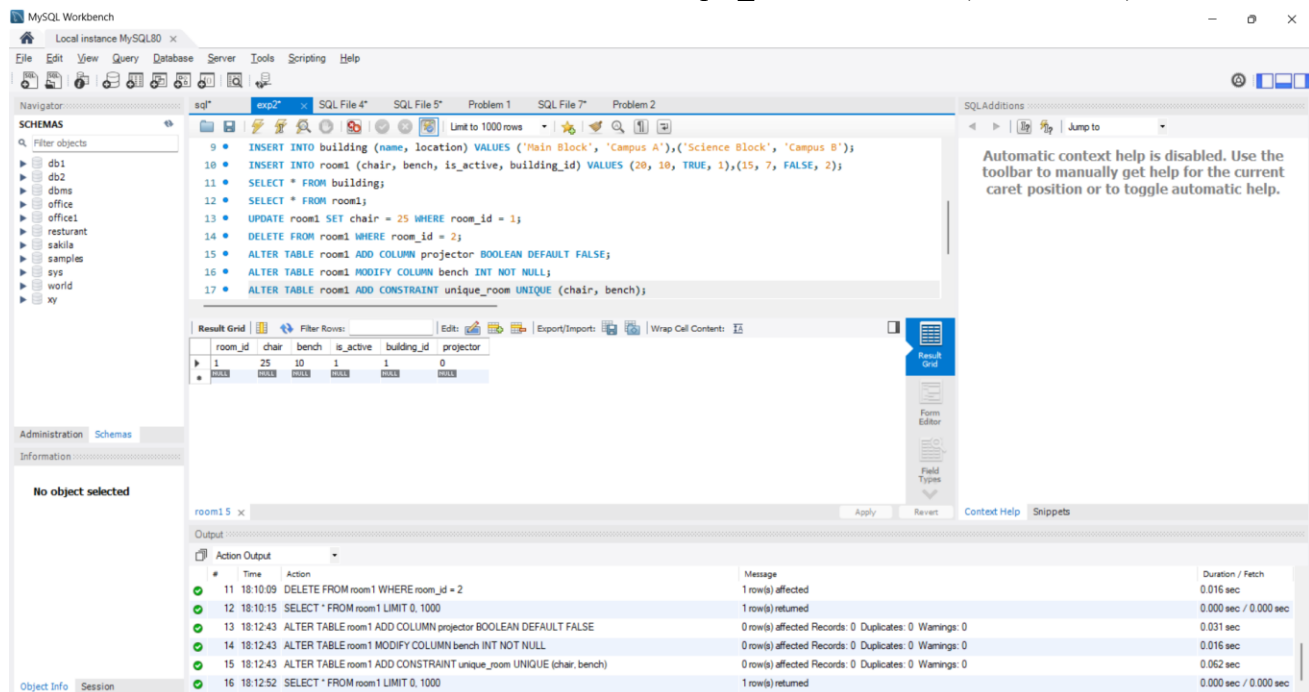


Figure 8: ALTER

8. RENAME

Syntax:

RENAME TABLE room1 TO classroom;



Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

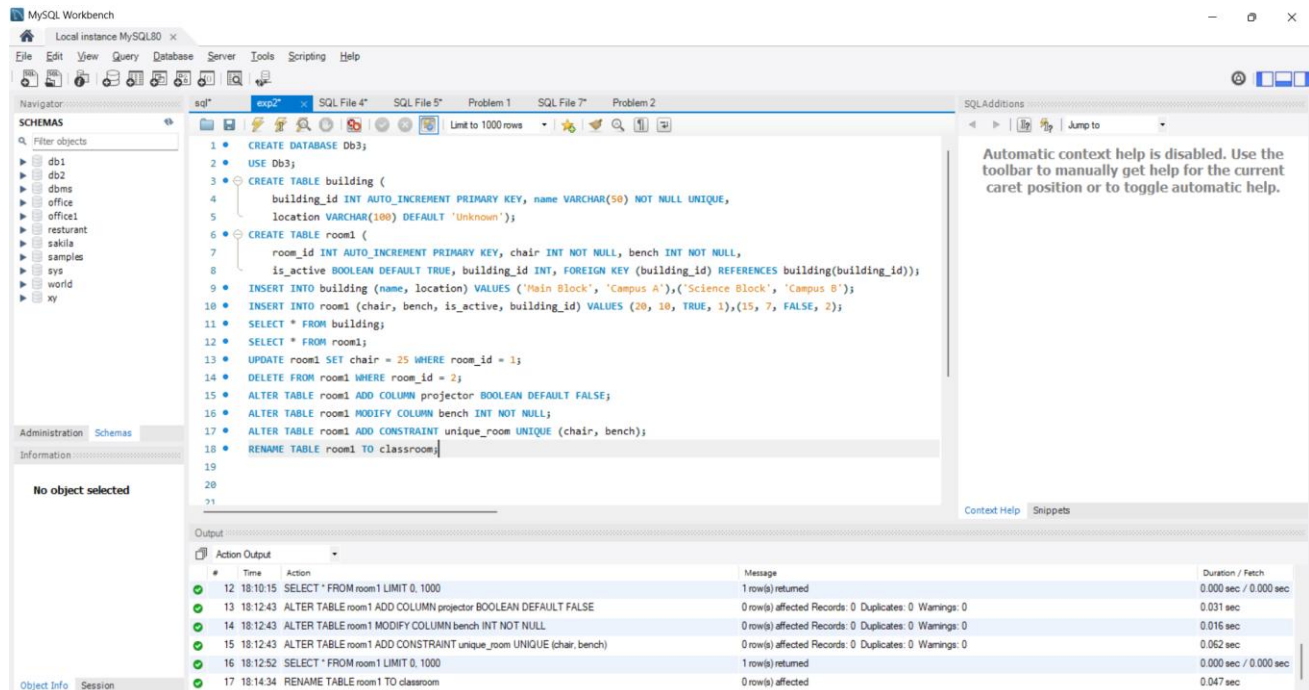


Figure 9: RENAME

9. TRUNCATE

Syntax:

TRUNCATE TABLE classroom;

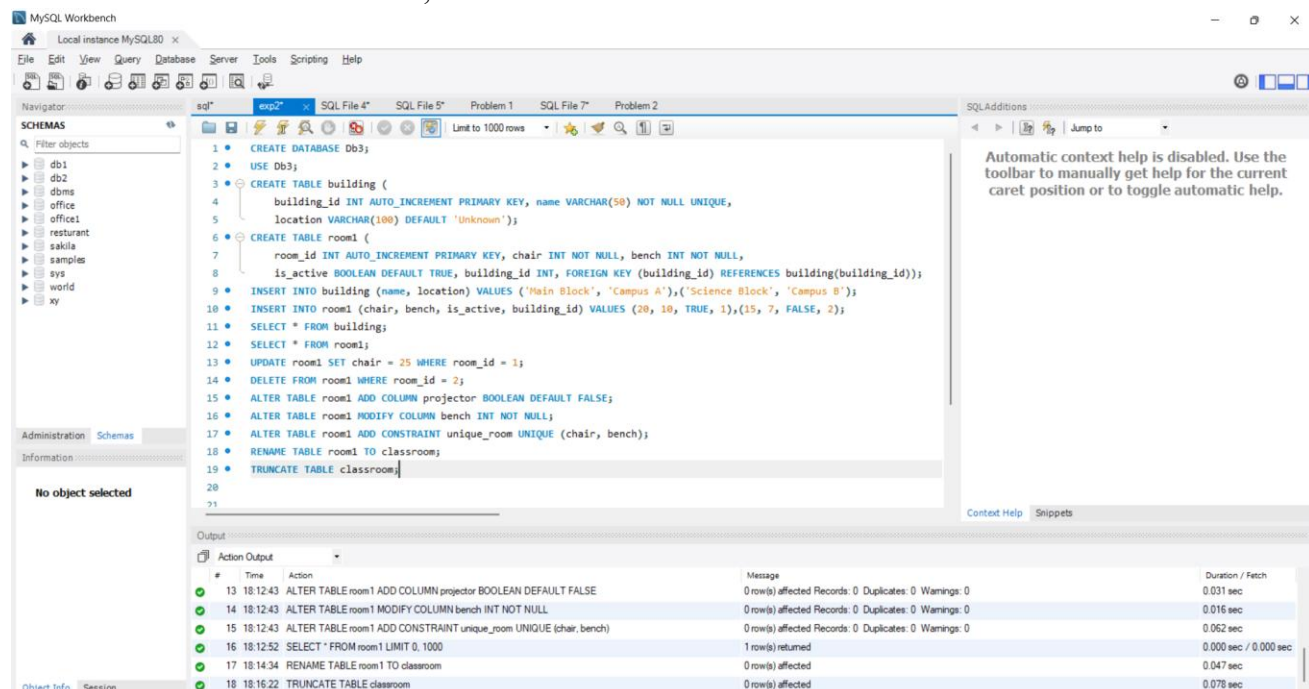


Figure 10: TRUNCATE