

Lab report

Name : ABID ALI

Student_no : 2019380141

Experiment 1

Experiment No:1.

Create and Manage Database and Table

Goal:

1. Familiar with command line and GUI connection method in MySQL.
2. Master SQL statement to create database and table.
3. Master the update and deletion methods of database and tables.
4. Master the basic methods of backup and restore database.
5. Understand the logical structure and physical structure of MySQL database.

Content of part 1

1. Use GUI to connect the DBMS
2. Use command line to connect the DBMS
3. Create,backup,drop and restore database and tables through GUI.

● Database and table

Database name: SPJ_MNG, four tables in the database: S, P, J, SPJ

S (SNO, SNAME, STATUS, CITY)

P (PNO, PNAME, COLOR, WEIGHT)

J (JNO, JNAME, CITY)

SPJ (SNO, PNO, JNO, QTY)

The supplier table S is composed of supplier code (SNO), supplier name (SNAME), supplier status (STATUS) and supplier city (CITY).

Part list P consists of part code (PNO), part name (PNAME), color (COLOR) and weight (WEIGHT).

Project table J consists of project code (JNO), project name (JNAME) and project city (CITY).

The supply situation table SPJ is composed of supplier code (SNO), part code (PNO), project code (JNO) and supply quantity (QTY). It indicates that the quantity of a certain part supplied by a supplier to an project is QTY.

Now there are several data in the database as follows. The specified operation is completed based on the database table.

Answer of part:1(GUI)

Use GUI to connect the DBMS

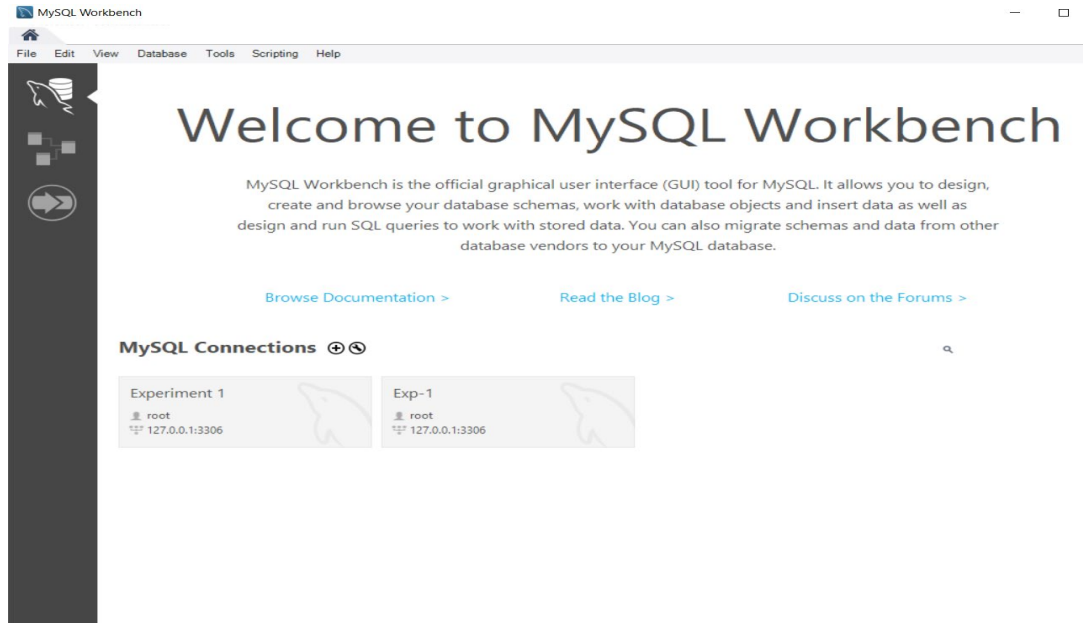


Fig: Connecting with MySQL Workbench

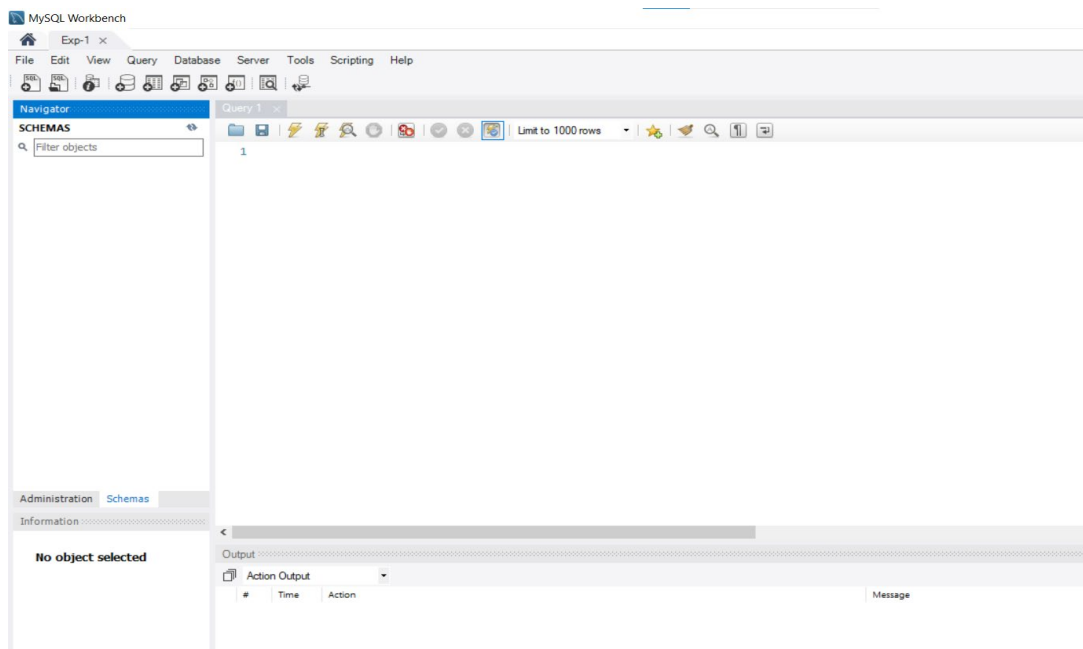


Fig: MySQL Workbench connected to server

Answer of part: 3

(1) Create database SPJ_MNG

Hint: right click **【Create Schema】**

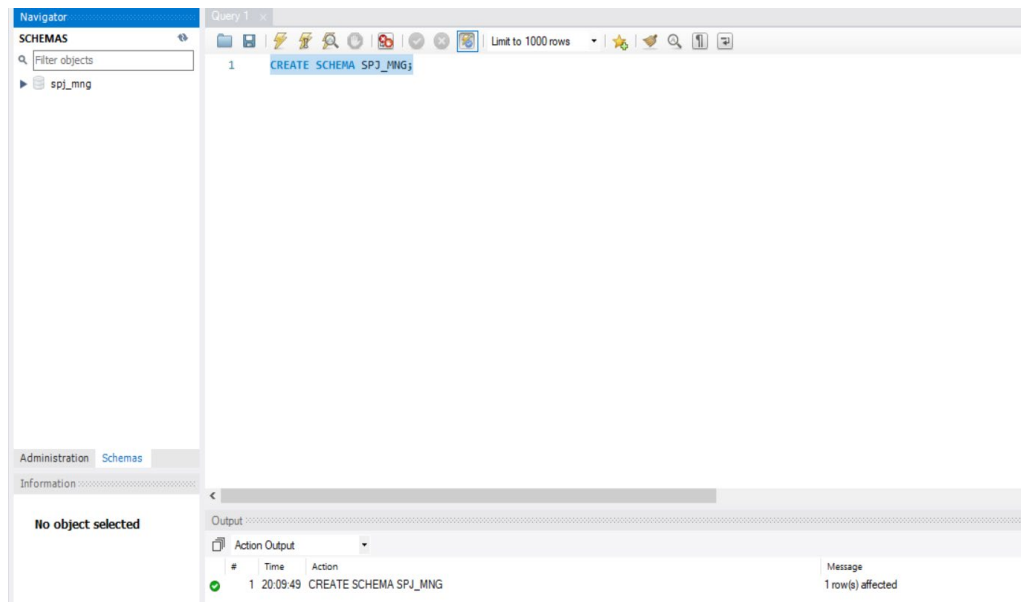


Fig: Creating database SPJ_MNG(By coding)

Code:

```
CREATE SCHEMA SPJ_MNG;
```

Alternate way to create database SPJ_MNG

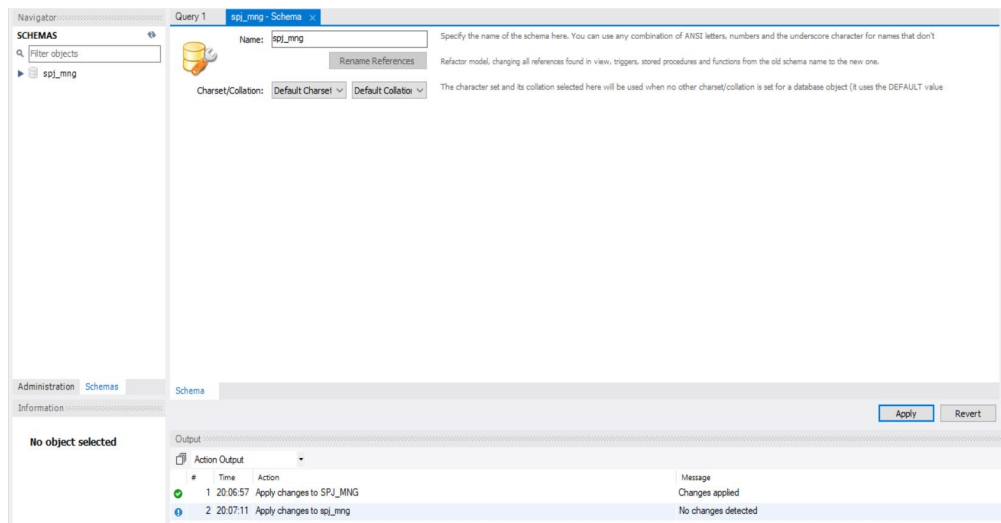


Fig: Creating database SPJ_MNG(By using the options provided by MySQL workbench)

(2) Create four tables in database SPJ_MNG (add some tuples in each table)。

Alternate way to create table named “S”

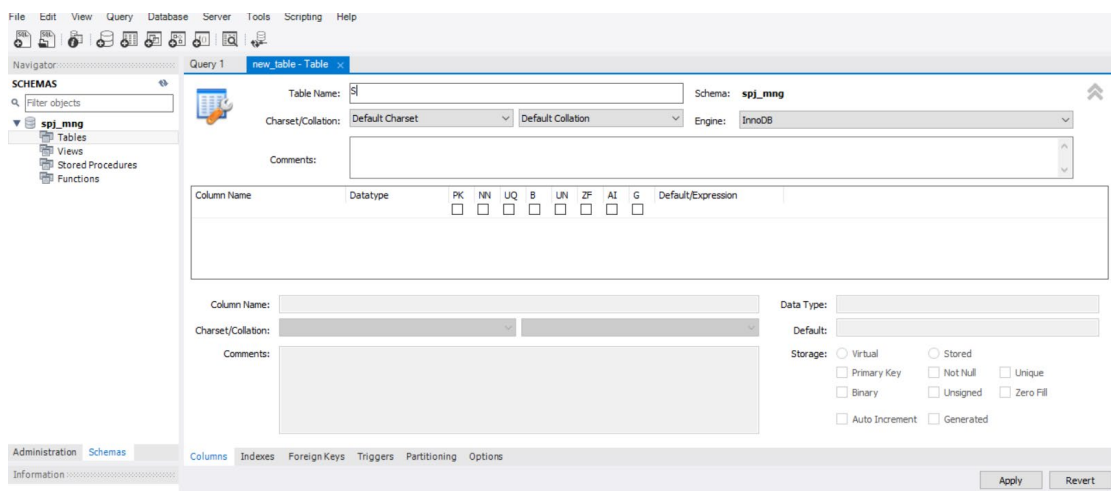


Fig: Creating table S(By using the options provided by MySQL workbench)

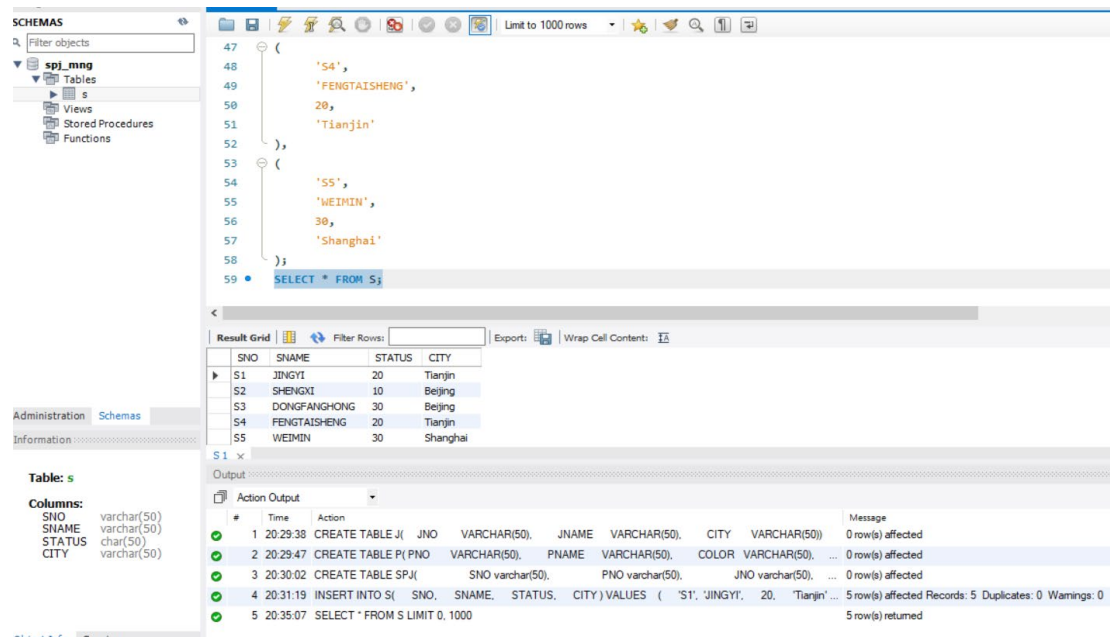


Fig: Creating table S

Code:

-- Creating table S

```

CREATE TABLE S(
    SNO      VARCHAR(50),
    SNAME    VARCHAR(50),
    STATUS   CHAR(50),
    CITY     VARCHAR(50));

```

-- Inserting value in table S

```

INSERT INTO S(
    SNO,
    SNAME,
    STATUS,
    CITY
)
VALUES
(
    'S1',
    'JINGYI',

```

20,
'Tianjin'
,
(
'S2',
'SHENGXI',
10,
'Beijing'
,
(
'S3',
'DONGFANGHONG',
30,
'Beijing'
,
(
'S4',
'FENGTAISHENG',
20,
'Tianjin'
,
(
'S5',
'WEIMIN',
30,
'Shanghai'
);

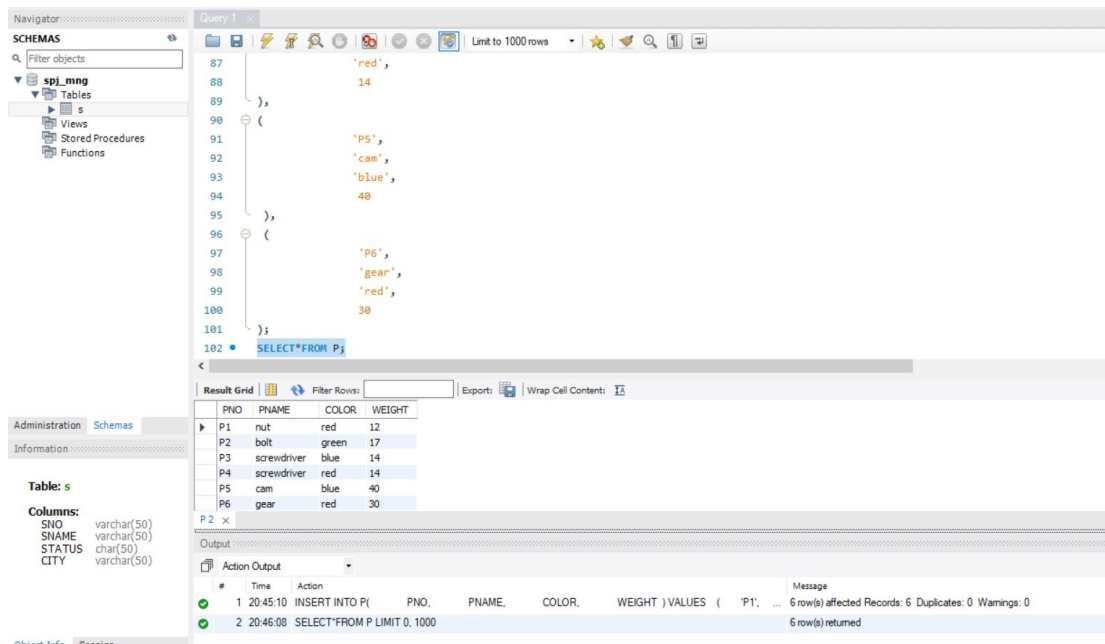


Fig: Creating table P

Code:

-- Creating table P

```

CREATE TABLE P(
    PNO          VARCHAR(50),
    PNAME        VARCHAR(50),
    COLOR        VARCHAR(50),
    WEIGHT       CHAR(50));

```

-- Inserting value in table P

```

INSERT INTO P(
    PNO,
    PNAME,
    COLOR,
    WEIGHT
)
VALUES
(
    'P1',
    'nut',
    'red',

```



```

12
),
(
    'P2',
    'bolt',
    'green',
17
),
(
    'P3',
    'screwdriver',
    'blue',
14
),
(
    'P4',
    'screwdriver',
    'red',
14
),
(
    'P5',
    'cam',
    'blue',
40
),
(
    'P6',
    'gear',
    'red',
30
);

```

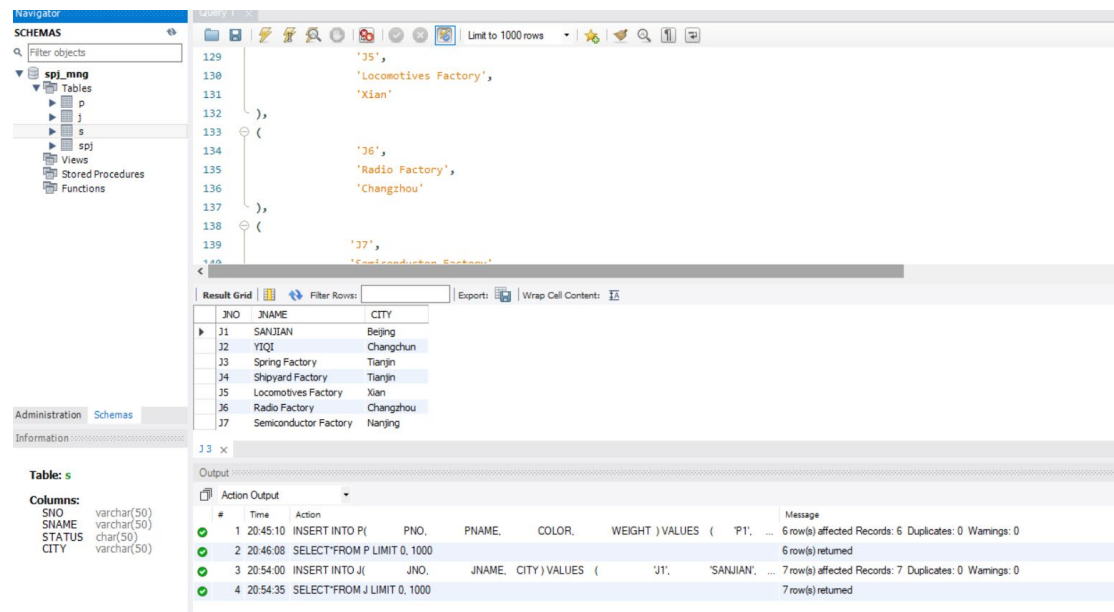


Fig: Creating table J

Code:

-- Creating table J

```
CREATE TABLE J(
    JNO          VARCHAR(50),
    JNAME        VARCHAR(50),
    CITY         VARCHAR(50));
```

-- Inserting value in table P

```
INSERT INTO J(
    JNO,
    JNAME,
    CITY
)
VALUES
    (
        'J1',
        'SANJIAN',
        'Beijing'
    ),
    (
        'J2',
```

'YIQI',
 'Changchun'
,
(
 'J3',
 'Spring Factory',
 'Tianjin'
,
(
 'J4',
 'Shipyard Factory',
 'Tianjin'
,
(
 'J5',
 'Locomotives Factory',
 'Xian'
,
(
 'J6',
 'Radio Factory',
 'Changzhou'
,
(
 'J7',
 'Semiconductor Factory',
 'Nanjing'
);

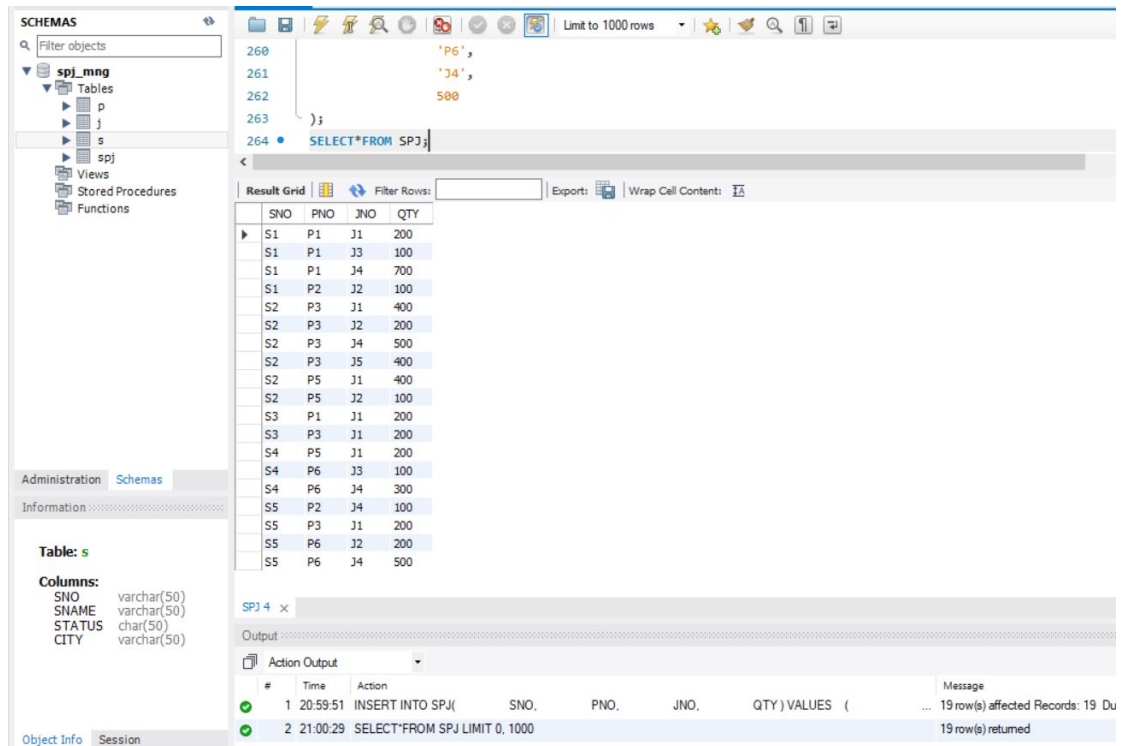


Fig: Creating table SPJ

Code:

-- Creating table SPJ

```
CREATE TABLE SPJ(
    SNO varchar(50),
    PNO varchar(50),
    JNO varchar(50),
    QTY varchar(50));
```

-- Inserting value in table SPJ

```
INSERT INTO SPJ(
    SNO,
    PNO,
    JNO,
    QTY
)
VALUES
(
```

```

        'S1',
        'P1',
        'J1',
        200
    ),
    (
        'S1',
        'P1',
        'J3',
        100
    ),
    (
        'S1',
        'P1',
        'J4',
        700
    ),
    (
        'S1',
        'P2',
        'J2',
        100
    ),
    (
        'S2',
        'P3',
        'J1',
        400
    ),
    (
        'S2',
        'P3',
        'J2',
        200
    ),
    (
        'S2',

```

```
        'P3',
        'J4',
        500
    ),
    (
        'S2',
        'P3',
        'J5',
        400
    ),
    (
        'S2',
        'P5',
        'J1',
        400
    ),
    (
        'S2',
        'P5',
        'J2',
        100
    ),
    (
        'S3',
        'P1',
        'J1',
        200
    ),
    (
        'S3',
        'P3',
        'J1',
        200
    ),
    (
        'S4',
        'P5',
```

'J1',
200
,
(
'S4',
'P6',
'J3',
100
,
(
'S4',
'P6',
'J4',
300
,
(
'S5',
'P2',
'J4',
100
,
(
'S5',
'P3',
'J1',
200
,
(
'S5',
'P6',
'J2',
200
,
(
'S5',
'P6',
'J4',

500

);

(3) Export the database SPJ_MNG as a *.SQLfile

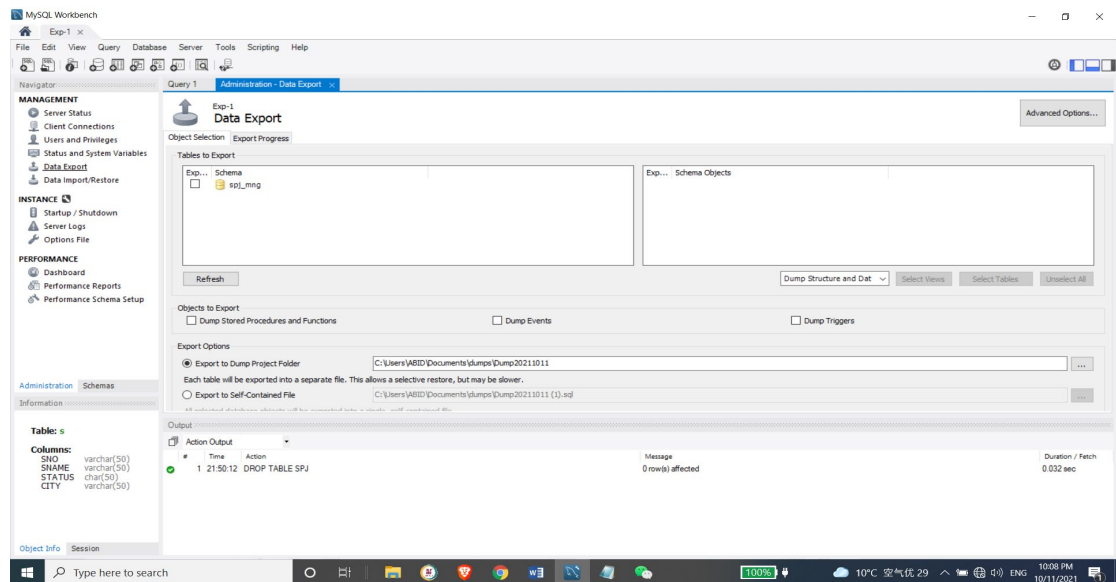


Fig: Choosing Administration → Data Export

(4) Delete the table of supplier(table SPJ)

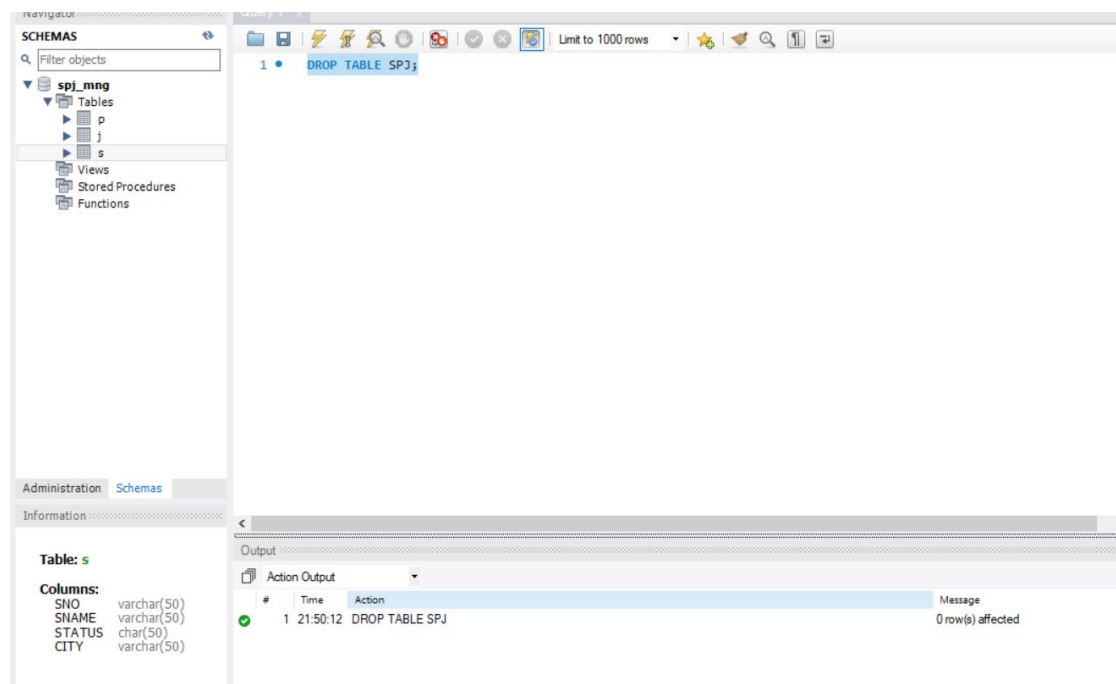


Fig: Deleting the table of supplier(table SPJ)

Code:

-- Deleting table SPJ

```
DROP TABLE SPJ;
```

(5) Delete database SPJ_MNG

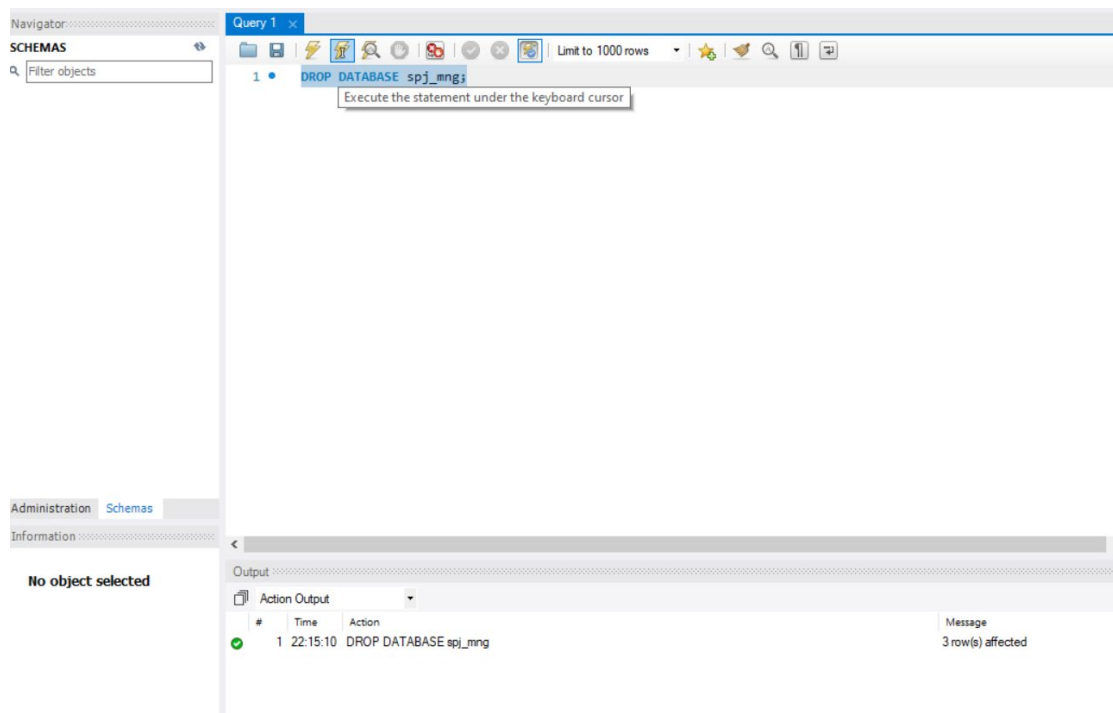


Fig:Deleting database SPJ_MNG

Code:

-- Deleting database SPJ_MNG

```
DROP DATABASE SPJ_MNG;
```

- (6) Restore the database SPJ_MNG with the file I have backed up in step (3)

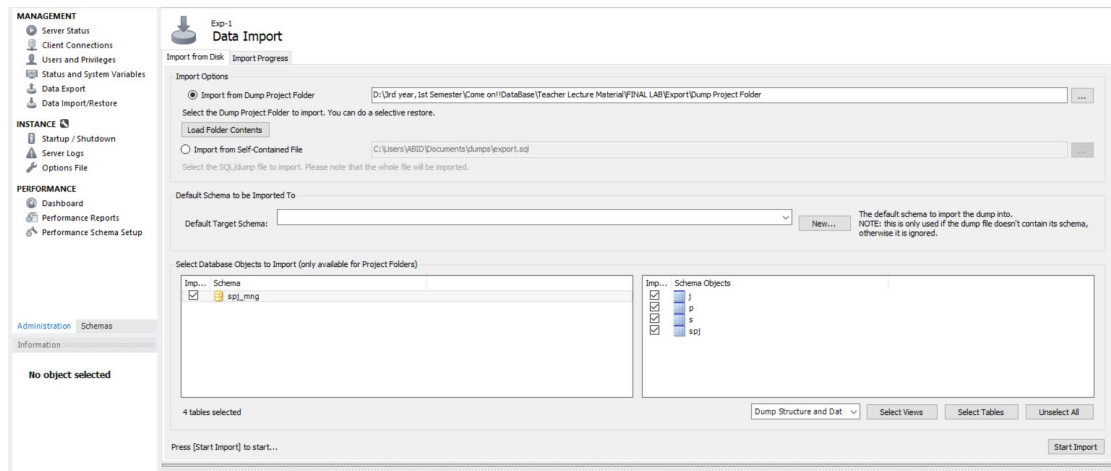


Fig 1: Restore the database SPJ_MNG

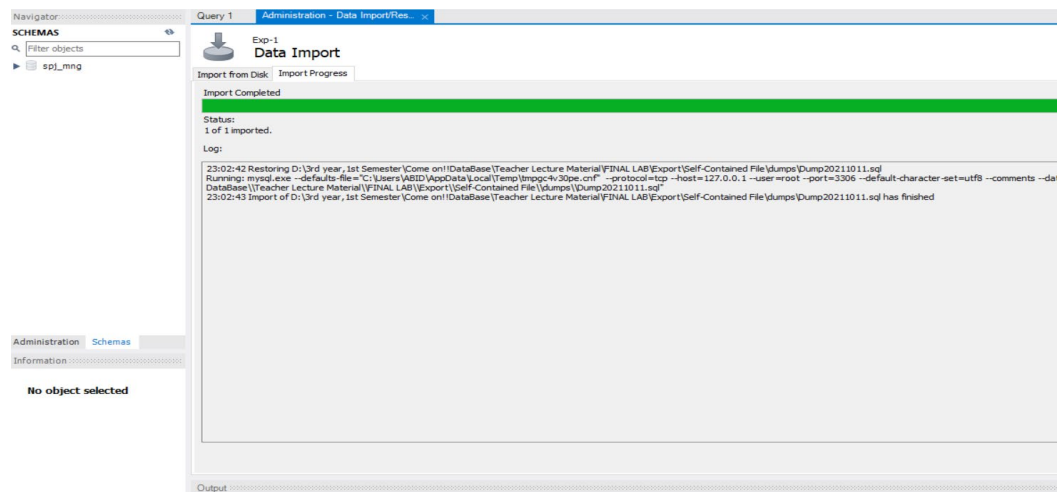


Fig 2: Restore the database SPJ_MNG

- (7) Update table S, Add an attribute of contact phone number STEL, the data type is string , and modify the maximum string length allowed by SNO in table S.

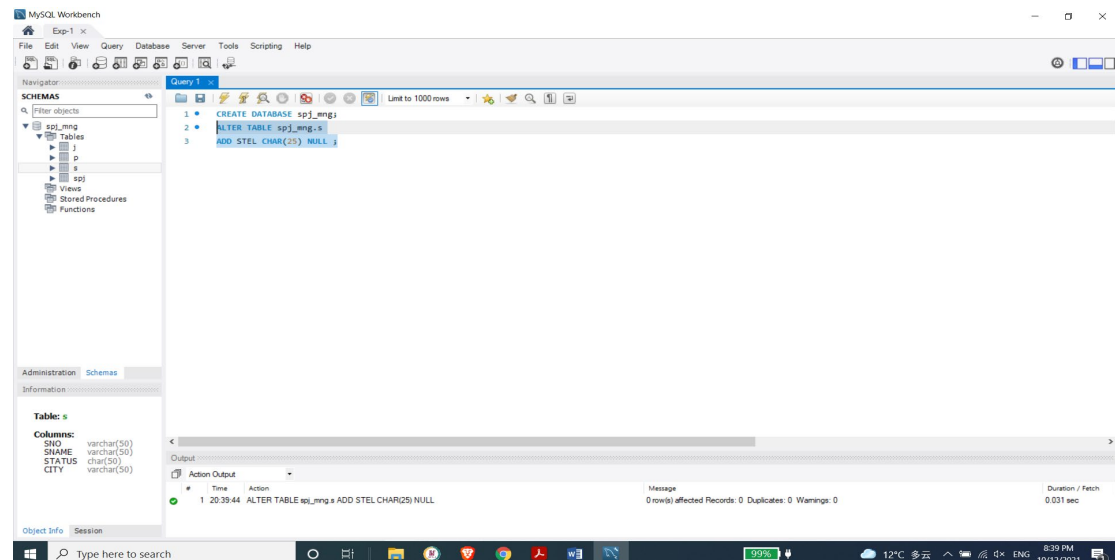


Fig: Adding an attribute of contact phone number STEL

Code:

-- Adding an attribute of contact phone number STEL

ALTER TABLE spj_mng.s

ADD STEL CHAR(25) NULL ;

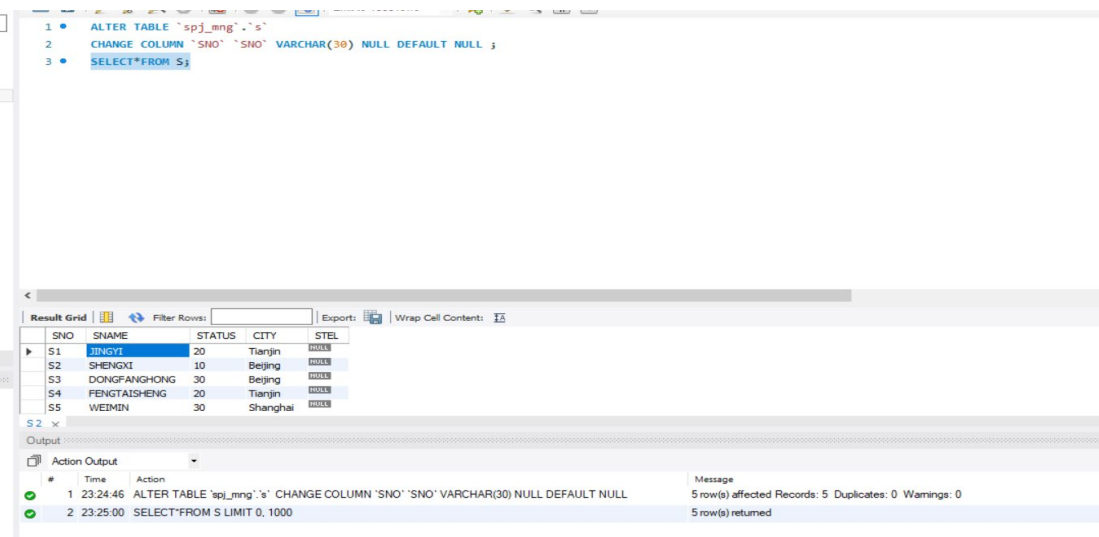


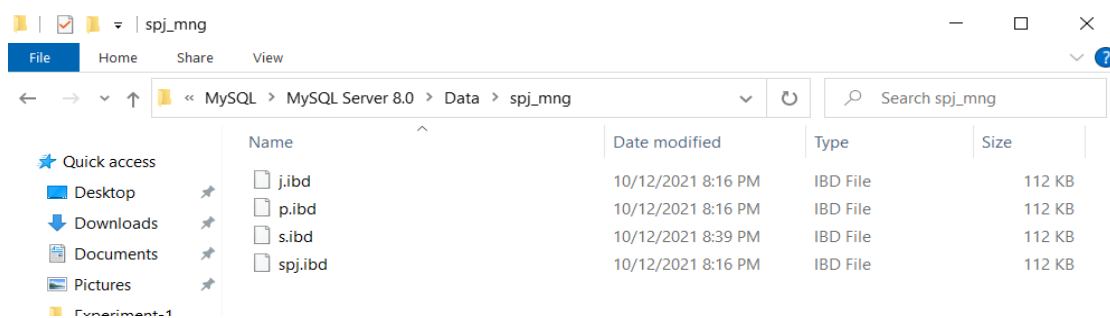
Fig: Modifying the maximum string length allowed by SNO in table S.

- (8) Understand the physical storage files of MySQL, and check the data files under the local MySQL service installation directory (such as the default installation directory: C:\programdata\MySQL\MySQL server 8.0\data). Try to create tables according to different storage engines of InnoDB and MyISAM, observe and explain the differences of physical storage files.

```
mysql> show global variables like "%datadir%";
```

Variable_name	Value
datadir	C:\ProgramData\MySQL\MySQL Server 8.0\Data\

1 row in set (0.02 sec)



.ibd(InnoDB) is the default extension of MySQL

```
mysql> show engines;
```

Engine	Support	Comment	Transactions	XA	Savepoints
MEMORY	YES	Hash based, stored in memory, useful for temporary tables	NO	NO	NO
MRG_MYISAM	YES	Collection of identical MyISAM tables	NO	NO	NO
CSV	YES	CSV storage engine	NO	NO	NO
FEDERATED	NO	Federated MySQL storage engine	NULL	NULL	NULL
PERFORMANCE_SCHEMA	YES	Performance Schema	NO	NO	NO
MyISAM	YES	MyISAM storage engine	NO	NO	NO
InnoDB	DEFAULT	Supports transactions, row-level locking, and foreign keys	YES	YES	YES
BLACKHOLE	YES	/dev/null storage engine (anything you write to it disappears)	NO	NO	NO
ARCHIVE	YES	Archive storage engine	NO	NO	NO

9 rows in set (0.00 sec)

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| spj_mng |
+-----+
4 rows in set (0.00 sec)
```

```
mysql> use spj_mng;
Database changed
mysql>
```

```
mysql> show tables;
+-----+
| Tables_in_spj_mng |
+-----+
| j |
| p |
| s |
| spj |
+-----+
4 rows in set (0.00 sec)
```

```
+-----+
| Tables_in_spj_mng |
+-----+
| j |
| p |
| s |
| spj |
+-----+
4 rows in set (0.00 sec)

mysql> create table x (id int);
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> show table status like 'x';
+-----+
| Name | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_data_length | Index_length | Data_free | Auto_increment | Create_time | Update_time | Check_time | Collation |
+-----+
| x | InnoDB | 10 | Dynamic | 0 | 0 | 16384 | 0 | 0 | 0 | 0 | NULL | 2021-10-12 22:42:04 | NULL | NULL | utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.02 sec)

mysql> alter table x engine=MyISAM;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

We can see InnoDB engine is default in My SQL

```
mysql> alter table x engine=MyISAM;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> show table status like '%x';
```

Name	Engine	Version	Row_format	Rows	Avg_row_length	Data_length	Max_data_length	Index_length	Data_free	Auto_increment	Create_time	Update_time	Check_time	Collation
x	MyISAM	10	Fixed	0	0	16384	0	0	0	NULL	2021-10-12 22:47:10	NULL	NULL	utf8mb4_0900_ai_ci

1 row in set (0.00 sec)

We alter the table into a new MyISAM engine

```
mysql> create table y (id int);
Query OK, 0 rows affected (0.04 sec)

mysql>
```

j.ibd	10/12/2021 8:16 PM	IBD File	112 KB
p.ibd	10/12/2021 8:16 PM	IBD File	112 KB
s.ibd	10/12/2021 8:39 PM	IBD File	112 KB
spj.ibd	10/12/2021 8:16 PM	IBD File	112 KB
x.MYD	10/12/2021 10:47 PM	MYD File	0 KB
x.MYI	10/12/2021 10:47 PM	MYI File	1 KB
x_406.sdi	10/12/2021 10:47 PM	SDI File	2 KB
y.ibd	10/12/2021 10:53 PM	IBD File	112 KB

Then, we create 2 new files using MyISAM and InnoDB engine to see the difference.

I used orange color to represent x.MYI and yellow colour to represent y.ibd which is the default engine of MySQL.

The main difference between MyISAM and INNODB are :

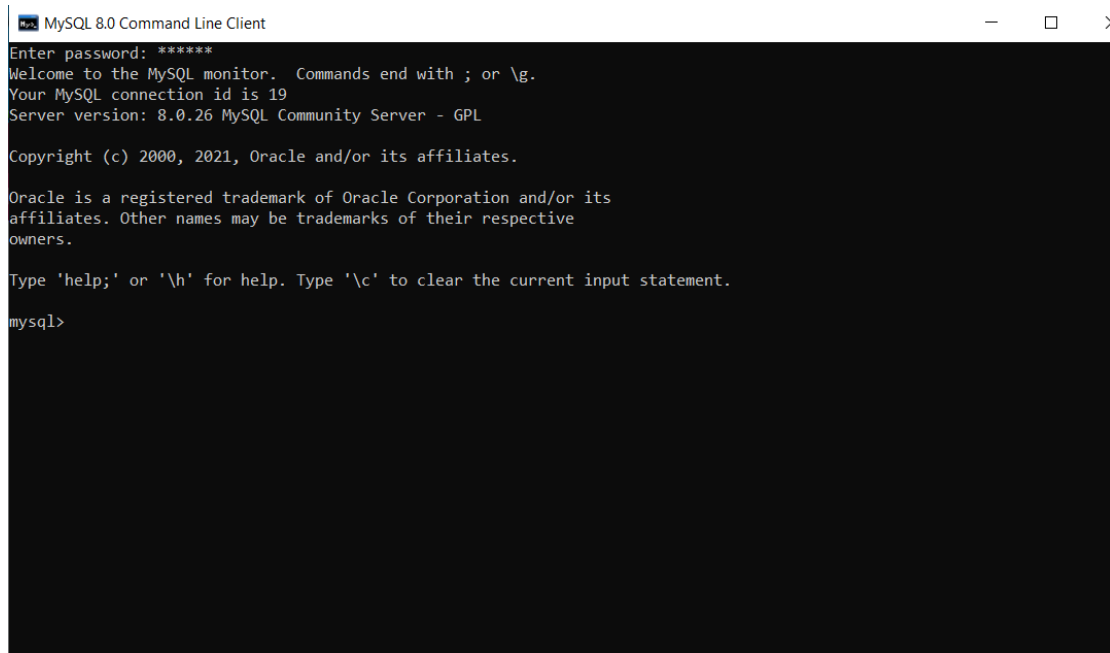
- MyISAM does not support transactions by tables while InnoDB supports.

- There are no possibility of row-level locking, relational integrity in MyISAM but with InnoDB this is possible. MyISAM has table-level locking.
- InnoDB does not support FULLTEXT index while MyISAM supports.
- Performance speed of MyISAM table is much higher as compared with tables in InnoDB.
- InnoDB is better option while you are dealing with larger database because it supports transactions, volume while MyISAM is suitable for small project.
- As InnoDB supports row-level locking which means inserting and updating is much faster as compared with MyISAM.
- InnoDB supports ACID (Atomicity, Consistency, Isolation and Durability) property while MyISAM does not support.
- In InnoDB table, AUTO_INCREMENT field is a part of index.
- Once table in InnoDB is deleted then it can not re-establish.
- InnoDB does not save data as table level so while implementation of select count(*) from table will again scan the whole table to calculate the number of rows while MyISAM save data as table level so you can easily read out the saved row number.
- MyISAM does not support FOREIGN-KEY referential-integrity constraints while InnoDB supports.

Contents of part 2

1. Use SQL statements to create a database of university.
2. Use SQL to create 3 tables: student, course, takes, define the data type and primary key, ignore the other constraints, add some tuples if you like.
3. Backup the database of university.
4. Delete created tables with SQL statement.
5. Delete created database with SQL statement.
6. Restore the database with the backed up files you've got in the operation of step (3).
7. Use SQL statement to add a new column STEL to store phone number into the table of student, and modify the max length of the attribute ID.
8. Check the SQL scripts that define the database or table in the SQL file generated by [mysqldump](#), and compare the similarities and differences between the automatically generated scripts and the SQL statements written by yourself.

Answer of part:2(MySQL command line)



```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Fig: Using command line to connect the DBMS

1. Use SQL statements to create a database of university.

```
mysql> CREATE DATABASE university;
Query OK, 1 row affected (0.02 sec)

mysql> _
```

Code:-

```
CREATE DATABASE university;
```

2. Use SQL to create 3 tables: student, course, takes, define the data type and primary key, ignore the other constraints, add some tuples if you like.

Code:-

```
CREATE TABLE student ( ID VARCHAR(5) NOT NULL,name VARCHAR(20) NULL,dept_name  
VARCHAR(20) NULL, tot_cred NUMERIC(3,0) NULL, PRIMARY KEY (ID));
```

```
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('00128', 'Zhang', 'Comp. Sci', '102');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('12345', 'Shankar', 'Comp. Sci', '32');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('19991', 'Brandt', 'History', '80');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('23121', 'Chavez', 'Finance', '110');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('44553', 'Peltier', 'Physics', '56');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('45678', 'Levy', 'Physics', '46');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('54321', 'Williams', 'Comp. Sci.', '54');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('55739', 'Sanchez', 'Music', '38');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('70557', 'Snow', 'Physics', '0');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('76543', 'Brown', 'Comp. Sci.', '58');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('76653', 'Aoi', 'Elec. Eng.', '60');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('98765', 'Bourikas', 'Elec. Eng.', '98');  
INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('98988', 'Tanaka', 'Biology', '120');
```

```
CREATE TABLE course (course_id VARCHAR(8) NOT NULL,title VARCHAR(50) NULL,dept_name  
VARCHAR(20) NULL,credits NUMERIC(2,0) NULL,PRIMARY KEY (course_id));
```

```
INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('BIO-101', 'Intro. to Biology', 'Biology', '4');  
INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('BIO-301', 'Genetics', 'Biology', '4');  
INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('BIO-399', 'Computational Biology', 'Biology', '3');  
INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('CS-101', 'Intro. to Computer Science', 'Comp.Sci.', '4');
```

INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('CS-190', 'Game Design', 'Comp.Sci.', '4');

INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('CS-315', 'Robotics', 'Comp.Sci.', '3');

INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('CS-319', 'Image Processing', 'Comp.Sci.', '3');

INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('CS-347', 'Database System Concept', 'Comp.Sci.', '3');

INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('EE-181', 'Intro. to Digital Systems', 'Elec.Eng.', '3');

INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('FIN-201', 'Investment Banking', 'Finance', '3');

INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('HIS-351', 'World History', 'History', '3');

INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('MU-199', 'Music Video Production', 'Music', '3');

INSERT INTO course ('course_id', 'title', 'dept_name', 'credits') VALUES ('Phy-101', 'Physical Principles', 'Physics', '4');

CREATE TABLE takes(CREATE TABLE takes (ID VARCHAR(5) NOT NULL,course_id VARCHAR(8) NOT NULL,sec_id VARCHAR(8)
NOT NULL,semester VARCHAR(6) NOT NULL,year NUMERIC(4,0) NOT NULL,grade VARCHAR(2) NULL,PRIMARY KEY (ID, course_id,
sec_id, semester, year));

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('00128', 'CS-101', '1', 'Fall', '2009', 'A');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('00128', 'CS-347', '1', 'Fall', '2009', 'A-');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('12345', 'CS-101', '1', 'Fall', '2009', 'C');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('12345', 'CS-190', '2', 'Spring', '2009', 'A');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('12345', 'CS-315', '1', 'Spring', '2010', 'A');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('12345', 'CS-347', '1', 'Fall', '2009', 'A');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('19991', 'HIS-351', '1', 'Spring', '2010', 'B');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('23121', 'FIN-201', '1', 'Spring', '2010', 'C+');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('44553', 'PHY-101', '1', 'Fall', '2009', 'B-');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('45678', 'CS-101', '1', 'Fall', '2009', 'F');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('45678', 'CS-101', '1', 'Spring', '2010', 'B+');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('45678', 'CS-319', '1', 'Spring', '2010', 'B');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('54321', 'CS-101', '1', 'Fall', '2009', 'A-');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('54321', 'CS-190', '2', 'Spring', '2009', 'B+');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('55739', 'MU-199', '1', 'Spring', '2010', 'A-');

```

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('76543', 'CS-101', '1', 'Fall', '2009', 'A');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('76543', 'CS-319', '2', 'Spring', '2010', 'A');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('76653', 'EE-181', '1', 'Spring', '2009', 'C');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('98765', 'CS-101', '1', 'Fall', '2009', 'C-');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('98765', 'CS-315', '1', 'Spring', '2010', 'B');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year', 'grade') VALUES ('98988', 'BIO-101', '1', 'Summer', '2009', 'A');

INSERT INTO takes ('ID', 'course_id', 'sec_id', 'semester', 'year') VALUES ('98988', 'BIO-101', '1', 'Summer', '2010');

```

```

mysql> USE university;
Database changed
mysql> CREATE TABLE student ( ID VARCHAR(5) NOT NULL, name VARCHAR(20) NULL, dept_name VARCHAR(20) NULL, tot_cred NUMERIC(3,0) NULL, PRIMARY KEY (ID));
Query OK, 0 rows affected (0.05 sec)

mysql> INSERT INTO student ('ID', 'name', 'dept_name', 'tot_cred') VALUES ('00128', 'Zhang', 'Comp. Sci', '102');
Query OK, 1 row affected (0.02 sec)

mysql>

```

Fig:Creating table and inputting value using MySQL command line

```

mysql> Show tables;
+-----+
| Tables_in_university |
+-----+
| course                |
| student               |
| takes                 |
+-----+
3 rows in set (0.01 sec)

```

Fig:3 tables created

```
mysql> Select*From course;
```

course_id	title	dept_name	credits
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp.Sci.	4
CS-190	Game Design	Comp.Sci.	4
CS-315	Robotics	Comp.Sci.	3
CS-319	Image Processing	Comp.Sci.	3
CS-347	Database System Concept	Comp.Sci.	3
EE-181	Intro. to Digital Systems	Elec.Eng	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
Phy-101	Physical Principles	Physics	4

```
13 rows in set (0.01 sec)
```

Fig:Course Table

```
mysql> Select*From student;
```

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci	102
12345	Shankar	Comp. Sci	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

```
13 rows in set (0.01 sec)
```

Fig:Student Table

```
mysql> Select*From takes;
```

ID	course_id	sec_id	semester	year	grade
00128	CS-101	1	Fall	2009	A
00128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B
23121	FIN-201	1	Spring	2010	C+
44553	PHY-101	1	Fall	2009	B-
45678	CS-101	1	Fall	2009	F
45678	CS-101	1	Spring	2010	B+
45678	CS-319	1	Spring	2010	B
54321	CS-101	1	Fall	2009	A-
76653	EE-181	1	Spring	2009	C
98765	CS-101	1	Fall	2009	C-
98765	CS-315	1	Spring	2010	B
98988	BIO-101	1	Summer	2009	A
98988	BIO-101	1	Summer	2010	NULL

```
18 rows in set (0.00 sec)
```

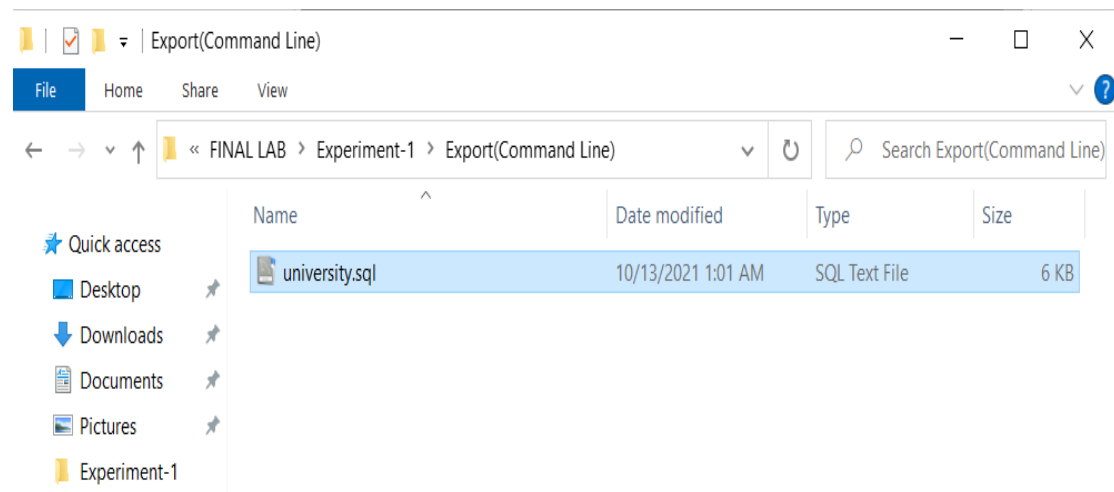
Fig:Takes Table

3. Backup the database of university.

```
8 Dir(s) 41,663,586,304 bytes free

C:\Program Files\MySQL\MySQL Server 8.0>cd bin

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -u root -p university >d:\university.sql
Enter password: *****
```



4. Delete created tables with SQL statement.

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| spj_mng |
| university |
+-----+
5 rows in set (0.01 sec)

mysql> use university;
Database changed
mysql> show tables;
+-----+
| Tables_in_university |
+-----+
| course |
| student |
| takes |
+-----+
3 rows in set (0.00 sec)

mysql> drop table course;
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+-----+
| Tables_in_university |
+-----+
| student |
| takes |
+-----+
2 rows in set (0.00 sec)

```

5. Delete created database with SQL statement.

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| spj_mng |
| university |
+-----+
5 rows in set (0.01 sec)

mysql> use university;
Database changed
mysql> show tables;
+-----+
| Tables_in_university |
+-----+
| course |
| student |
| takes |
+-----+
3 rows in set (0.00 sec)

mysql> drop table course;
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+-----+
| Tables_in_university |
+-----+
| student |
| takes |
+-----+
2 rows in set (0.00 sec)

mysql> drop database university;
Query OK, 2 rows affected (0.05 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| spj_mng |
+-----+
4 rows in set (0.01 sec)

```

6. Restore the database with the backed up files you've got in the operation of step (3).


```
C:\Program Files\MySQL\MySQL Server 8.0>cd bin  
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -u root -p university >d:\university.sql  
Enter password: *****
```

Fig: Backing up the database university

mysqldump -u root -p university > university_db_backup.sql

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p university < university_db_backup.sql  
Enter password: *****  
C:\Program Files\MySQL\MySQL Server 8.0\bin>
```

Fig: Restoring the database university

```
mysql> drop database university;  
Query OK, 2 rows affected (0.05 sec)  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| spj_mng |  
+-----+  
4 rows in set (0.01 sec)  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| spj_mng |  
| university |  
+-----+  
5 rows in set (0.00 sec)
```

Fig: Restored the database university

mysql -u root -p university < university_db_backup.sql

```
mysql> select*from student;
```

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci	102
12345	Shankar	Comp. Sci	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

```
13 rows in set (0.00 sec)

mysql> ALTER TABLE university.student add STEL INT(20) NULL;
Query OK, 0 rows affected, 1 warning (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 1

mysql> select*from student;
```

ID	name	dept_name	tot_cred	STEL
00128	Zhang	Comp. Sci	102	NULL
12345	Shankar	Comp. Sci	32	NULL
19991	Brandt	History	80	NULL
23121	Chavez	Finance	110	NULL
44553	Peltier	Physics	56	NULL
45678	Levy	Physics	46	NULL
54321	Williams	Comp. Sci.	54	NULL
55739	Sanchez	Music	38	NULL
70557	Snow	Physics	0	NULL
76543	Brown	Comp. Sci.	58	NULL
76653	Aoi	Elec. Eng.	60	NULL
98765	Bourikas	Elec. Eng.	98	NULL
98988	Tanaka	Biology	120	NULL

```
13 rows in set (0.00 sec)
```

ALTER TABLE university.student add STEL INT(20) NULL;

9. Check the SQL scripts that define the database or table in the SQL file generated by [mysqldump](#), and compare the similarities and differences between the automatically generated scripts and the SQL statements written by yourself.

```
-- MySQL dump 10.13 Distrib 8.0.26, for Win64 (x86_64)
--
-- Host: localhost Database: university
--
-- Server version 8.0.26

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `course`
--

DROP TABLE IF EXISTS `course`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `course` (
  `course_id` varchar(8) NOT NULL,
  `title` varchar(50) DEFAULT NULL,
  `dept_name` varchar(20) DEFAULT NULL,
  `credits` decimal(2,0) DEFAULT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @@character_set_client */;

-- MySQL dump 10.13 Distrib 8.0.26, for Win64 (x86_64)
--
-- Host: localhost Database: university
--
-- Server version 8.0.26

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `course`
--

DROP TABLE IF EXISTS `course`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `course` (
  `course_id` varchar(8) NOT NULL,
  `title` varchar(50) DEFAULT NULL,
  `dept_name` varchar(20) DEFAULT NULL,
  `credits` decimal(2,0) DEFAULT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @@character_set_client */;

-- Dumping data for table `course`
--

-- Table structure for table `takes`
--

DROP TABLE IF EXISTS `takes`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `takes` (
  `id` varchar(8) NOT NULL,
  `course_id` varchar(8) NOT NULL,
  `sec_id` varchar(8) NOT NULL,
  `section` varchar(8) NOT NULL,
  `year` decimal(4,0) NOT NULL,
  `grade` decimal(2,0) DEFAULT NULL,
  PRIMARY KEY (`id`,`course_id`,`sec_id`,`section`,`year`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @@character_set_client */;

-- Dumping data for table `takes`
--

-- Table structure for table `takes`
--

DROP TABLE IF EXISTS `takes`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `takes` (
  `id` varchar(8) NOT NULL,
  `course_id` varchar(8) NOT NULL,
  `sec_id` varchar(8) NOT NULL,
  `section` varchar(8) NOT NULL,
  `year` decimal(4,0) NOT NULL,
  `grade` decimal(2,0) DEFAULT NULL,
  PRIMARY KEY (`id`,`course_id`,`sec_id`,`section`,`year`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @@character_set_client */;

-- Dump completed on 2021-10-13 1:41:11
```

DROP TABLE IF EXISTS

LOCK TABLES

UNLOCK TABLES

DROP TABLE IF EXISTS

This are the keyword found in mysqldump which is not similar like mysql statement also the syntax structure is little bit different.

Problems

Simple syntax error.

Solutions

To solve these problems I looked for information in internet. In order to understand some questions and procedure I also asked the teacher to help me understand them. And provided instructions helped to solve some of my errors during the experiment.

Summary

From this experiment I have learned SQL statement to Create and Manage Database and Table. I have learned how to update and delete methods of database and table. Have become familiar with SQL statements of data insertion, modification and deletion of basic tables. Learned all kinds of data operation about basic table in GUI and command line and GUI connection method in MySQL.

Reference:

- 1) <https://www.w3schools.com/sql/>
- 2) https://www.w3schools.com/sql/sql_syntax.asp
- 3) https://www.youtube.com/watch?v=7S_tz1z_5bA
- 4) <https://www.youtube.com/watch?v=ER8oKX5myE0>
- 5) <https://www.sqlshack.com/how-to-backup-and-restore-mysql-databases-using-the-mysqldump-command/>
- 6) <https://dev.mysql.com/doc/refman/8.0/en/command-line-options.html>

Attachments

- 1) DB1_2019380141_ABID ALI.docx
- 2) DB1_2019380141_ABID ALI.pdf