**Name:Vaishali Jaware Roll No:516**

Cyber Forensics and Laws

**Mini Project**

**Aim**: Write a program to take backup of a MySQL database.

# Used software:

* Python 3.10.5 (64 bit)
* XAMPP 3.3.0 (64 bit)
  + Apache 2.4.53 (64 bit)
  + MariaDB 10.2.24 (based on MySQL 15.1) (64 bit)
  + PHP 8.1.6 (64 bit)
  + phpMyAdmin 5.2.0
* Chromium 103.0.5060.114 (64 bit)

**Note:** Apache and PHP are internal dependencies for running phpMyAdmin and are not explicitly used in this exercise.

# Description:

* **Backup**: In [information technology](https://en.wikipedia.org/wiki/Information_technology), a **backup**, or **data backup** is a copy of [computer data](https://en.wikipedia.org/wiki/Computer_data) taken and stored elsewhere so that it may be used to restore the original after a [data loss](https://en.wikipedia.org/wiki/Data_loss) event. Backups can be used to recover data after its loss from [data](https://en.wikipedia.org/wiki/File_deletion)

[deletion](https://en.wikipedia.org/wiki/File_deletion) or [corruption](https://en.wikipedia.org/wiki/Data_corruption), or to recover data from an earlier time.

* **Database**: In [computing](https://en.wikipedia.org/wiki/Computing), a **database** is an organized collection of [data](https://en.wikipedia.org/wiki/Data_(computing)) stored and accessed electronically. Small databases can be

stored on a [file system](https://en.wikipedia.org/wiki/File_system), while large databases are hosted on [computer clusters](https://en.wikipedia.org/wiki/Computer_clusters) or [cloud storage](https://en.wikipedia.org/wiki/Cloud_storage).

* + **Relational Database**: A **relational database** is a (most commonly digital) [database](https://en.wikipedia.org/wiki/Database) based on the [relational model](https://en.wikipedia.org/wiki/Relational_model) of data. A relational model organizes data into one or more [tables](https://en.wikipedia.org/wiki/Table_(database))

(or "relations") of [columns](https://en.wikipedia.org/wiki/Column_(database)) and [rows](https://en.wikipedia.org/wiki/Row_(database)), with a unique key identifying each row. Rows are also called [records](https://en.wikipedia.org/wiki/Record_(computer_science)) or [tuples](https://en.wikipedia.org/wiki/Tuple). Columns are also called attributes. Generally, each table/relation represents one "entity type" (such as customer or product). The rows represent instances of that type of entity (such as "Lee" or "chair") and the columns representing values attributed to that instance (such as address or price).

* **Python**: **Python** is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [interpreted](https://en.wikipedia.org/wiki/Interpreter_(computing)), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language) [programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Its design philosophy emphasizes [code](https://en.wikipedia.org/wiki/Code_readability) [readability](https://en.wikipedia.org/wiki/Code_readability) with the use of [significant indentation](https://en.wikipedia.org/wiki/Off-side_rule). Python is

[dynamically-typed](https://en.wikipedia.org/wiki/Type_system#DYNAMIC) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). It is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

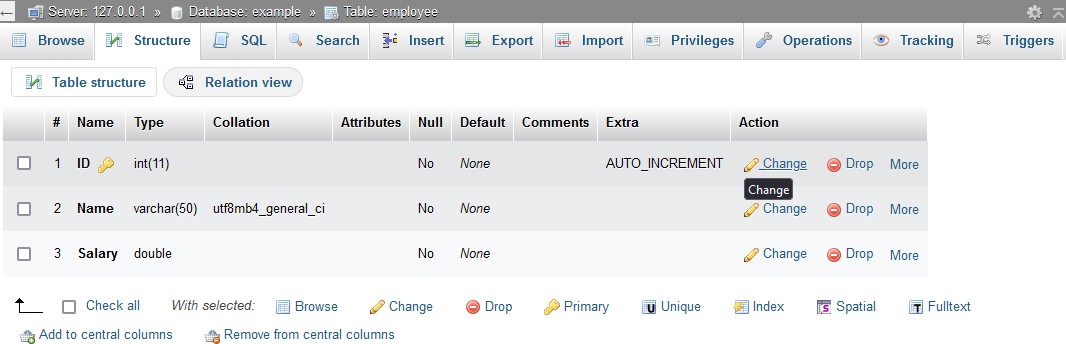
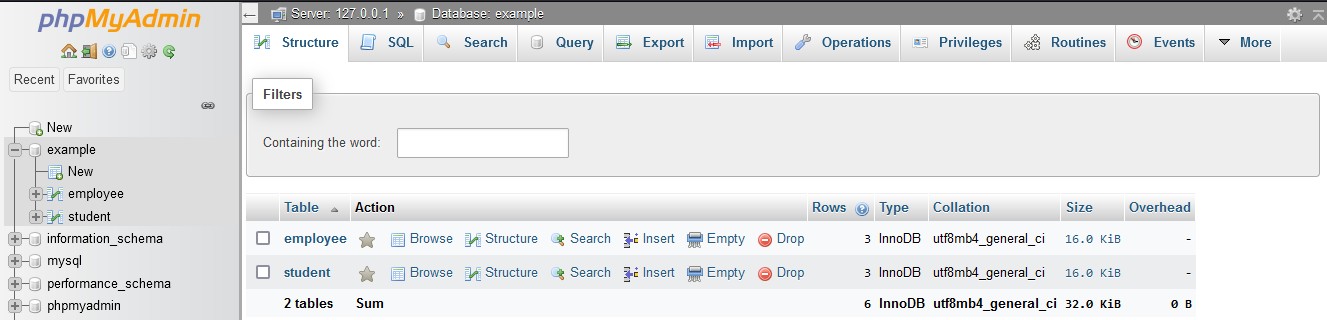
* **MySQL**: **MySQL** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [relational database](https://en.wikipedia.org/wiki/Relational_database_management_system) [management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). Its name is a combination of "My", the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, and

"SQL", the abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). MySQL is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU](https://en.wikipedia.org/wiki/GNU_General_Public_License) [General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), and is also available under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) licenses.

* **MariaDB**: **MariaDB** is a community-developed, commercially supported [fork](https://en.wikipedia.org/wiki/Fork_(software_development)) of the [MySQL](https://en.wikipedia.org/wiki/MySQL) [relational database management](https://en.wikipedia.org/wiki/Relational_database_management_system) [system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), intended to remain [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) [software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License). Development is

led by some of the original developers of MySQL, who forked it due to concerns over its [acquisition](https://en.wikipedia.org/wiki/Takeover) by [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation) in 2009.

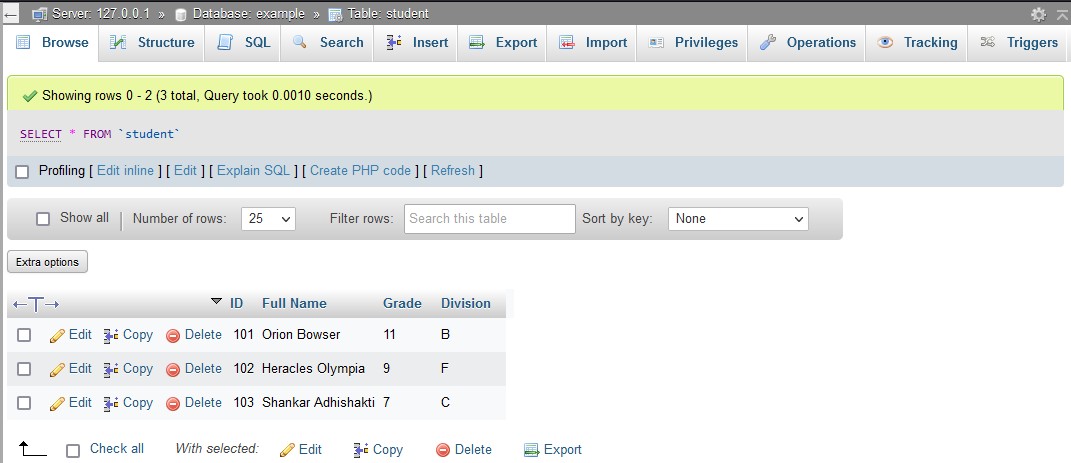
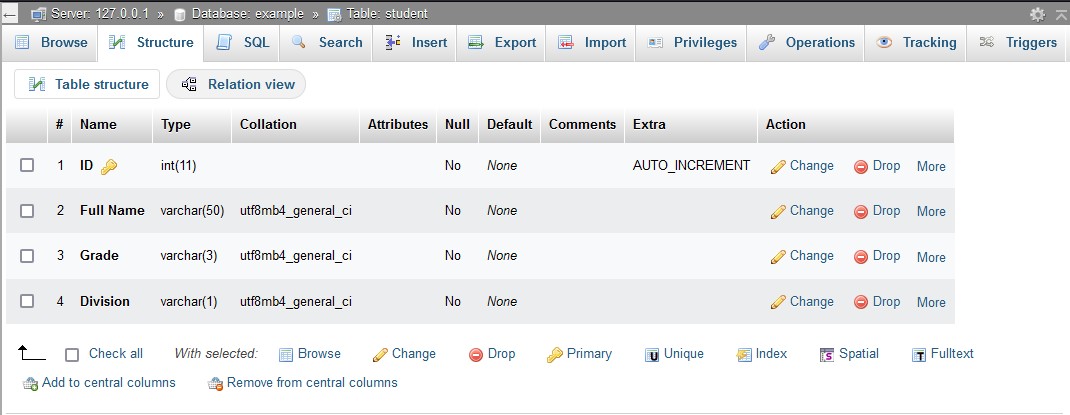
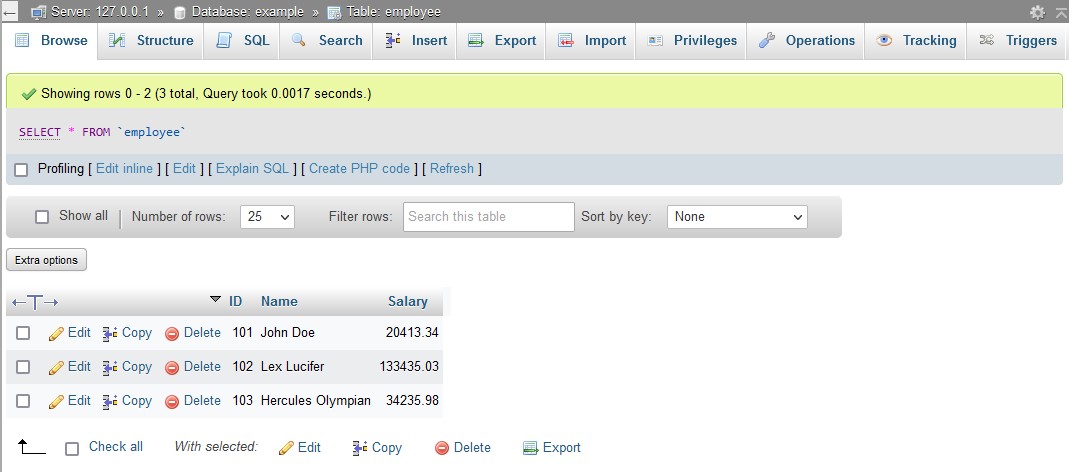
* **XAMPP**: **XAMPP** is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [web](https://en.wikipedia.org/wiki/Web_server) [server](https://en.wikipedia.org/wiki/Web_server) [solution stack](https://en.wikipedia.org/wiki/Solution_stack) package developed by Apache Friends,



consisting mainly of the [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server), [MariaDB](https://en.wikipedia.org/wiki/MariaDB) [database](https://en.wikipedia.org/wiki/Database), and [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) for scripts written in the [PHP](https://en.wikipedia.org/wiki/PHP) and [Perl](https://en.wikipedia.org/wiki/Perl) [programming languages](https://en.wikipedia.org/wiki/Programming_language). Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

# Database structure and contents:

* **Name**: *examp e*
* **Database contents:**
  + **Tables**:
    - *employ e*
      * **Structure**:
      * **Contents**:



* + - *stu ent*
      * **Structure**:
      * **Contents**:

# Program:

import mysql.connector as connector

from sys import exit

if \_\_name\_\_ == "\_\_main\_\_":

hostname = input("Enter host name [localhost]>")

hostname = hostname if hostname != "" else "localhost"

username = input("Enter your username > ")

if username == "":

exit("Please input the correct username")

print("If password is not set, just press [Enter] on the following prompt")

password = input("Enter your password > ")

database\_name = input("Enter the name of the database you want to backup > ")

if database\_name == "":

exit("Please input the correct database name")

print("Trying connection...")

try:

connection = connector.connect(host = hostname, user = username, password = password, database = database\_name)

cursor = connection.cursor()

print("Connection successful")

cursor.execute("show tables;")

table\_names : list[str] = []

for record in cursor.fetchall():

table\_names.append(record[0])

backup\_database\_name = database\_name + "\_backup"

cursor.execute(f"create database {backup\_database\_name};")

cursor.execute(f"use {backup\_database\_name};")

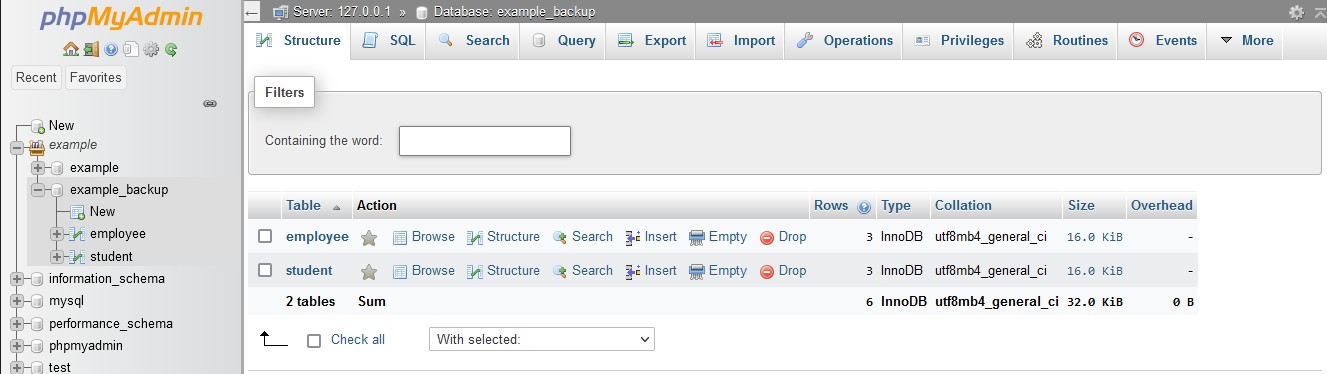
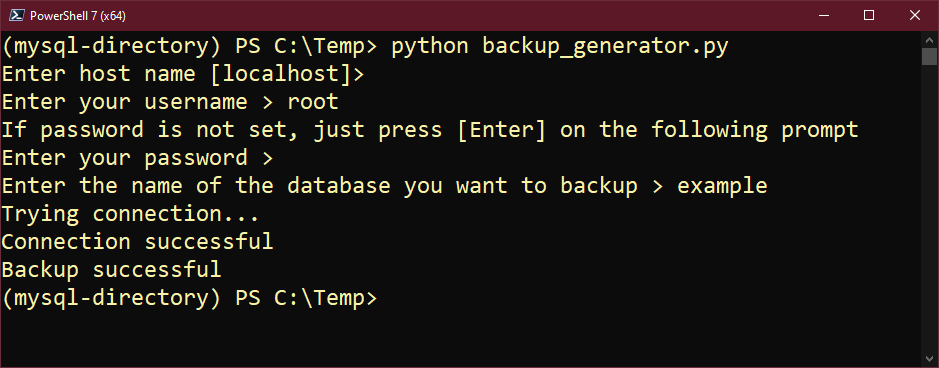
for table\_name in table\_names:

cursor.execute(f"create table {table\_name} select \* from {database\_name}.{table\_name}")

print("Backup successful")

except:

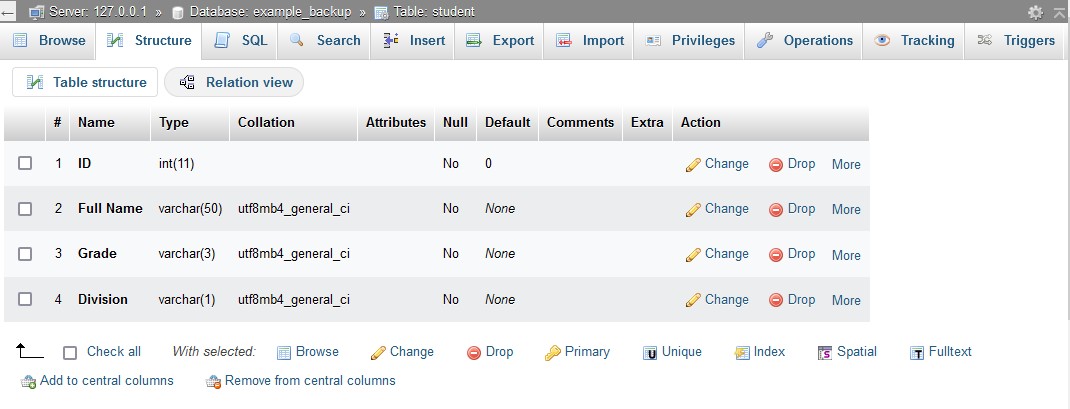
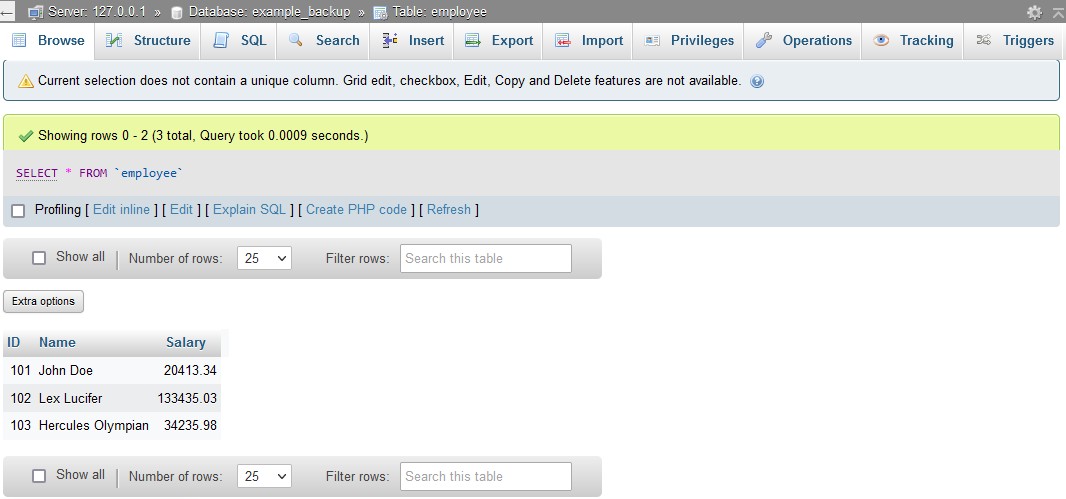
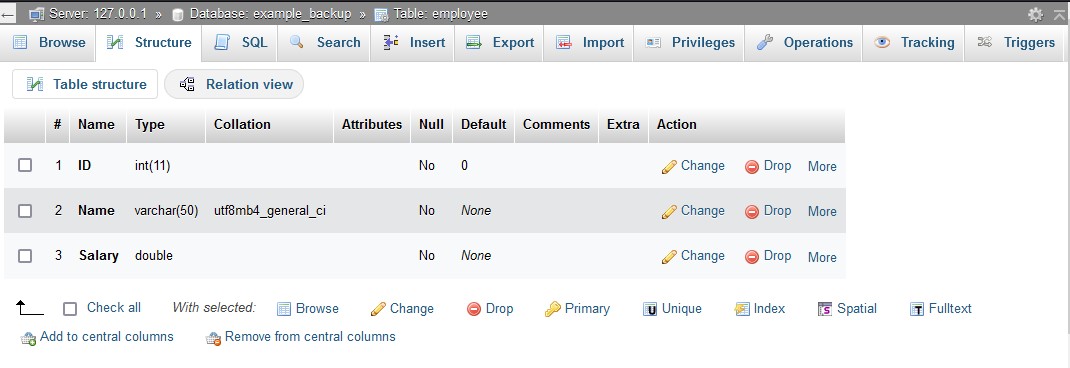
exit("Connection unsuccessful")



# Output:

**Database structure and contents:**

* **Name**: *examp e\_backup*
* **Database contents:**
  + **Tables:**
    - *employ e*
      * **Structure:**



* + - * **Contents**:
    - *stu ent*
      * **Structure**:
      * **Contents**:

