Assignment 1: Database Automation and Scripting

Question 1: Understanding Database Automation

**1.1) What is Database Automation and Why It Matters**

* Automation involves using specialized software or scripts to automate the performance of regular and recurring tasks that manage databases such as backups, monitoring, updates and data integration.
* Nowadays, as a result of advanced digitalization, firms handle vast streams of information. Doing everything by hand with large amounts of data can lead to delays, increases the chance of mistakes and lowers operational efficiency. Automation automatizes these tasks in a fast and error-free manner.
* Just imagine manually backing up your database each night. One mistake or mishap could result in losing important information. Automation allows you to set up automatic backups so that you know your data is being safely and consistently protected.
* Bigger businesses tend to also handle bigger amounts of data. Automation allows you to increase your productivity without having to hire more staff. Security controls and updates are applied uniformly as systems are updated throughout the organization.
* Database automation is essential for companies to successfully handle their data needs.
  + Easily handle large amounts of information.
  + Enforce security consistently.
  + Respond faster to changing needs.
  + Reduce dependency on manual tasks.

**1.2) Benefits of Automating Database Tasks**

Automating database tasks has many practical advantages.

1. **Reduced Human Errors** 
   * Manual handling of tasks increases the risk of making errors such as missing or misinterpreting a step. Automation provides consistency and accuracy, so each task is carried out exactly the same way each time.
   * Example: The bank automates the process of installing software updates on its database servers. Without automation, the administrator could inadvertently skip a server or make an error when applying the patch.
2. **Increased Reliability and Uptime**
   * Automated systems execute tasks with consistency and predictability. As a result, these systems are more dependable than people for nights and weekends when only automated solutions are available.
   * Example: Amazon automates the management of thousands of databases within its platform. Automation handles backups, replication and failover to maintain continuous service.
3. **Faster Deployments and Rollouts**
   * Using automation, companies can roll out changes to their databases and ensure those changes are applied consistently in all environments.
   * Example: The company standardizes the schema for its application and handles database updates during deployment, updating all environments simultaneously. Automation helps ensure that different environments are always up to date with each other.
4. **Cost Savings**
   * Automation demands some initial investment but it greatly reduces the need for ongoing maintenance in the future. Reducing errors minimizes outages and cuts down on problem-solving efforts.
   * Example: Netflix automates data processing, data backups and health checks. It helps save money and enables systems to operate effectively with fewer database experts.

Question 2: Scripting for Database Automation

**Database Automation Scripts: Explanation**

I created two Python scripts to automate MySQL database tasks:

**2.1) Python Scripting for Database Backup Automation**

**Backup Script (backup\_script.py):**

* This script automatically backs up all my MySQL databases.
* It connects to MySQL using my credentials, lists all databases, and creates a backup file for each one.
* Each backup file is named with the database name and a timestamp (e.g., assignment1\_20250520\_223730.sql).
* I tested it and confirmed it works by checking the backup files in the mysql\_backups folder.

**2.2) Python Scripting for Database Change Deployment**

**Deployment Script (deploy\_changes\_script.py):**

* This script applies SQL changes (like creating new tables) to my database.
* I created a sample SQL file (add\_table.sql) with a command to create a test table.
* The script reads this file, executes the commands, and logs the results.
* I verified it worked by checking the MySQL Command Line Client and seeing the new table listed.

**How I Tested Everything**

* I set up MySQL, created a database (assignment1), and a user (student).
* I ran a test script (test\_connection.py) to confirm my Python scripts could connect to MySQL.
* I tested both the backup and deployment scripts and confirmed they worked as expected.