DATABASE AUTOMATION

PROG8850

ASSIGNMENT 1

Database Automation and Scripting

SUBMITTED BY,

MARTIN JOHNY

8945124

Understanding Database Automation

Database Automation and Its Significance

Database automation involves scripts for deploying and tools to execute database management tasks which eventually minimize manual work while increasing efficiency. This automation system is important to modern data management because it securely and efficiently manages large amounts of data.

Key aspects of database automation include:

- **Backup and Recovery:** The backup and recovery process automatically create data backups on a regular schedule without requiring manual input from the user.
- **Schema Management:** Schema Management involves automated processes for modifying database structures through table additions or changes.
- **Monitoring and Alerts:** The monitoring and alerts system keeps track of database health and performance metrics which eventually helps to identify unusual activity.
- **Data Migration:** Automating data transfer between different database environments and instances.

Benefits of Automating Database Tasks

- **Reduced Errors:** Automation helps to avoid human mistakes that occur during manual database processes.
- **Increased Reliability:** Database automation improves reliability through consistent and accurate database operations.
- **Faster Deployments:** Faster deployment procedures minimize the time required to execute database changes.
- **Cost Efficiency:** Database automation helps to reduce operational expenses through minimized manual input requirements.

Real-World Example:

Database automation helps Netflix to manage its huge content library with greater efficiency. Automated processes for backup and replication maintain continuous service availability by minimizing downtime and helps to preventing data loss.

Python Scripting for Database Backup Automation

Script Overview:

The ***backup_script.py*** automates the backing up of a MySQL database. It ensures that backups are stored securely and has uniquely identified parameters to prevent overwriting.

Script Logic:

- **Database Connection Details:**

  The script defines connection strings of DB such as host, user, password, and database name.

- **Backup Directory Creation:**

  If the backup directory does not exist, the script creates one to store backups of the database.

- **Generating a Unique Filename:**

  A timestamp (YYYYMMDD-HHMMSS) is created to the backup filename to make each backup unique.

- **Executing the Backup Command:**

  The script runs the **mysqldump** command to generate a backup file.

- **Completion Message:**

  Once the backup is done the script gives a confirmation by printing the file path.

Python Scripting for Database Change Deployment

Script Overview:

The **deploy_changes_script.py** automates database schema modifications, such as adding new tables or columns. This helps to prevent manual errors and ensures consistent deployment.

Script Logic:

- **Establishing a Database Connection:**

  Use Python's **mysql.connector** to connect to the MySQL database with predefined connection strings

- **Executing SQL Commands:**

This script runs an SQL query to create a user table if it does not exist.

- **Changes committed:**

  After executing the SQL statement, the script commits the changes to apply and then it will be committed to the database.

- **Closing the Connection:**

  The script closes the database connection to release resources.