**Task 5:**

**Write a script to automate the backup of a database and its periodic restoration for testing purposes.**

**Solution**:

|  |
| --- |
| # import necessary packages for python  def backup\_database():  # Generate backup file name with timestamp  timestamp = time.strftime('%Y%m%d%H%M%S')  backup\_file = f'{database}\_{timestamp}.bak'  backup\_path = os.path.join(backup\_dir, backup\_file)  # Backup database  conn\_str = f'DRIVER={driver};SERVER={server};DATABASE={database};UID={username};PWD={password}'  conn = pyodbc.connect(conn\_str)  cursor = conn.cursor()  cursor.execute(f'BACKUP DATABASE {database} TO DISK = \'{backup\_path}\'')  conn.close()  print(f'Database backup created: {backup\_file}')  def restore\_database(backup\_file):  # Restore database from backup file  backup\_path = os.path.join(backup\_dir, backup\_file)  conn\_str = f'DRIVER={driver};SERVER={server};DATABASE=master;UID={username};PWD={password}'  conn = pyodbc.connect(conn\_str)  cursor = conn.cursor()  cursor.execute(f'RESTORE DATABASE {database} FROM DISK = \'{backup\_path}\'')  conn.close()  print(f'Database restored from: {backup\_file}')  if \_\_name\_\_ == "\_\_main\_\_":  # Backup database periodically (e.g., daily)  backup\_database()  # Restore database for testing purposes (sample restoration)  # Replace 'backup\_file\_name.bak' with the actual backup file name  restore\_database('backup\_file\_name.bak') |

**Backup Database Function**: Connects to the SQL Server database and executes a backup command to create a backup file with a timestamp in the specified backup directory.

**Restore Database Function**: Connects to the SQL Server database and executes a restore command to restore the database from a specified backup file