**LAB #03**

**Data warehouse and Business Intelligence**

**Motivation:**To build understanding and practicing automation of table creation and data filing using python scripts

**Task 01: Demonstration** carefully observe the demonstration in the class and try to run the code on your local machine

*\*you received DW&BI\_lab\_03.zip it will be used for further tasks, if you are unable to find it , clone the following repository:*

[*https://github.com/shiza-asghar/Northwind\_datawarehouse*](https://github.com/shiza-asghar/Northwind_datawarehouse)

**Task 02:** create the database using UI (SSMS) and connect your database with your python file and create all the tables using python scripts *(you can change the data types but will have to handle them in later tasks)*

**Task 03a:** Fill the tables using running queries in batch for insert by reading the csv files available in the zip folder that are in format “<table\_name>.csv”

You can use the following code snippet

```python3:

Import pandas as pd

df.read\_csv(“path\_of\_file.csv ”,header = false)

```

**Task 03b:** Fill the tables whose csv files are not available by using the Faker module of python to generate synthetic data *(as it is used in demo)*

**Task 04:** Create the index on following keys:

**Primary Key (PK) Indexes:**

1. Orders.OrderID

2. Customers.CustomerID

3. Suppliers.SupplierID

4. Products.ProductID

5. Employees.EmployeeID

**Non-Primary Key (Non-PK) Indexes:**

1. Customers.CustomerName

2. Shippers.CompanyName

3. Orders.OrderDate

Using the following syntax :

```Sql:

CREATE INDEX <idx\_name>

ON <Table\_name>(<column\_name>);

```

**Task 05:** Run the following queries multiple times and observe the response time and store it , you can use python time module to observe the time difference

**Queries involving PK indexes:**

1. Retrieve all details of an order based on a specific `OrderID`.

2. Find all customer information using a specific `CustomerID`.

3. Retrieve supplier details for a given `SupplierID`.

4. Get product details using a specific `ProductID`.

5. Retrieve employee details by `EmployeeID`.

**Queries involving Non-PK indexes:**

1. Retrieve all customer details for a given `CustomerName`.

2. Find shipper details using a specific `CompanyName`.

3. Retrieve all orders placed on a particular `OrderDate`.

4. Find all orders placed by customers with a specific `CustomerName`.

5. Get the list of orders shipped by a specific shipper’s `CompanyName`.

**After observing the response time create a graph to demonstrate the difference between PK and non-PK query**

**The following code snippets will help you to reduce the working time :**

***Note:****you might have to multiply values with 100 or any power of them to magnify the minor differences*

**```python3:**

#Execute queries and measure execution time

execution\_times = {}

for query\_name, query in queries.items():

start\_time = time.time()

cursor.execute(query) cursor.fetchall()

# Fetch results to ensure query is fully executed

end\_time = time.time()

execution\_times[query\_name] = end\_time - start\_time

**````**

**```python3:**

plt.bar(execution\_times.keys(), execution\_times.values())

plt.xlabel('Queries')

plt.ylabel('Execution Time (seconds)')

plt.title('Query Execution Times')

plt.show()

**````**