

Sure! I'll break down the three practicals for you step by step, and then I'll provide the theory for each one.

---

## 6. Import Data from Different Sources (Excel, SQL Server, Oracle)

### Step-by-Step Process:

1. **Import Data from Excel** using pandas.
  2. **Import Data from SQL Server** using pyodbc or SQLAlchemy.
  3. **Import Data from Oracle** using cx\_Oracle.
- 

### Code:

# 1. Import Data from Excel (using pandas)

```
import pandas as pd
```

```
# Load an Excel file into a DataFrame
```

```
excel_file_path = "C:/path_to_your_file.xlsx"
```

```
df_excel = pd.read_excel(excel_file_path)
```

```
# Display the first few rows of the dataset
```

```
print("Excel Data:")
```

```
print(df_excel.head())
```

# 2. Import Data from SQL Server (using pyodbc)

```
import pyodbc
```

```
# Define connection string for SQL Server
```

```
conn_str = 'DRIVER={ODBC Driver 17 for SQL  
Server};SERVER=your_server_name;DATABASE=your_database_name;UID=your_username;PWD=yo  
ur_password'
```

```
# Establish the connection
```

```
conn = pyodbc.connect(conn_str)
```

```
# Query SQL data
query = "SELECT * FROM your_table_name"
df_sql = pd.read_sql(query, conn)

# Display the first few rows of the dataset
print("SQL Server Data:")
print(df_sql.head())

# 3. Import Data from Oracle (using cx_Oracle)
import cx_Oracle

# Define connection string for Oracle
conn = cx_Oracle.connect('username/password@hostname:port/service_name')

# Query Oracle data
query = "SELECT * FROM your_table_name"
df_oracle = pd.read_sql(query, conn)

# Display the first few rows of the dataset
print("Oracle Data:")
print(df_oracle.head())
```

---

### Theory:

#### 1. Importing Data from Excel:

- `pandas.read_excel()` is used to read Excel files (.xls or .xlsx) into a DataFrame. It's ideal for importing data from spreadsheets into Python for analysis.

#### 2. Importing Data from SQL Server:

- `pyodbc` is a Python library that allows you to connect to SQL Server databases. The `pyodbc.connect()` function uses the connection string to connect to the database.
- Once connected, you can query the database using SQL commands and fetch the results into a DataFrame with `pandas.read_sql()`.

#### 3. Importing Data from Oracle:

- cx\_Oracle is used to connect Python with Oracle databases. You can run SQL queries on the Oracle database and import the results into a pandas DataFrame.
- 

## 7. Data Visualization from Extraction, Transformation, and Loading (ETL) Process

### Steps in ETL Process:

1. **Extraction:** Extract data from various sources.
2. **Transformation:** Clean and preprocess the data (e.g., handle missing values, normalize data).
3. **Loading:** Load the cleaned data into the target system (e.g., a DataFrame, SQL database).

### Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Example: Extraction of data
# Let's simulate extracting data from a CSV file
data = pd.read_csv('data.csv')

# Example: Transformation
# Drop rows with missing values
data_clean = data.dropna()

# Example: Loading the cleaned data into another system (here, we just print it)
print("Cleaned Data:")
print(data_clean.head())

# Data Visualization

# Plotting distribution of a column
plt.figure(figsize=(10,6))
sns.histplot(data_clean['column_name'], kde=True)
```

```
plt.title('Distribution of Column Name')
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.show()

# Correlation heatmap of the data
plt.figure(figsize=(10,6))
sns.heatmap(data_clean.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

---

### Theory:

#### 1. Extraction:

- Data is extracted from various sources, such as databases, APIs, or CSV/Excel files.
- You can use libraries like pandas to read data into DataFrames for easier manipulation.

#### 2. Transformation:

- After extracting the data, it is often necessary to transform it, including cleaning missing values, changing data types, and normalizing data.
- In this example, dropna() removes rows with missing values.

#### 3. Loading:

- The cleaned data is then loaded into a target system or simply visualized in Python.
- You can load data into SQL databases or use it for analysis directly in memory.

#### 4. Visualization:

- Data visualization is done using seaborn or matplotlib. A histogram shows the distribution of values in a column, and a correlation heatmap visualizes relationships between different variables.
- 

## 9. Data Analysis and Visualization Using Advanced Excel

### Steps in Excel:

1. **Import Data:** Use Data tab to import data from CSV or databases.
2. **Data Cleaning:** Use Excel's built-in functions like FILTER, REMOVE DUPLICATES, TEXT TO COLUMNS.

3. **Data Visualization:** Use Excel's charting features like PivotTables, Scatter Plots, Line Charts, etc.

#### Excel Example:

1. **Import Data:**
  - Go to Data → Get Data → Select the source (e.g., CSV, SQL Server).
2. **Data Cleaning:**
  - **Remove duplicates:** Data → Remove Duplicates.
  - **Handle missing values:** Use IFERROR, ISBLANK, or Go To Special to fill missing cells.
3. **Visualization:**
  - Use Pivot Tables: Go to Insert → PivotTable.
  - Create charts (e.g., bar charts, line charts) to visualize the data.

#### Steps for Creating a Chart:

1. Select the data range you want to visualize.
2. Go to Insert → Select the type of chart (e.g., Line, Bar, Pie).
3. Customize the chart by adding titles, labels, and adjusting colors.

---

#### Conclusion and Theory for All Practical Steps:

1. **Importing Data:** In modern data science, data often comes from multiple sources (Excel, SQL, Oracle). Python, through libraries like pandas, pyodbc, and cx\_Oracle, makes it simple to connect to and load data from different systems.
2. **ETL Process:** The process of extracting, transforming, and loading data is critical in ensuring that raw data is cleaned, prepared, and ready for analysis. Visualization in the ETL process helps you understand data distributions, relationships, and anomalies before deep analysis.
3. **Advanced Excel for Data Analysis:** Excel remains one of the most powerful and widely used tools for business intelligence. It allows easy import of data, cleaning using formulas, and advanced visualization techniques like charts and pivot tables. While Python offers more scalability, Excel is often preferred for business-driven, quick analysis.

---

Let me know if you need any more specific examples or clarifications for these tasks!