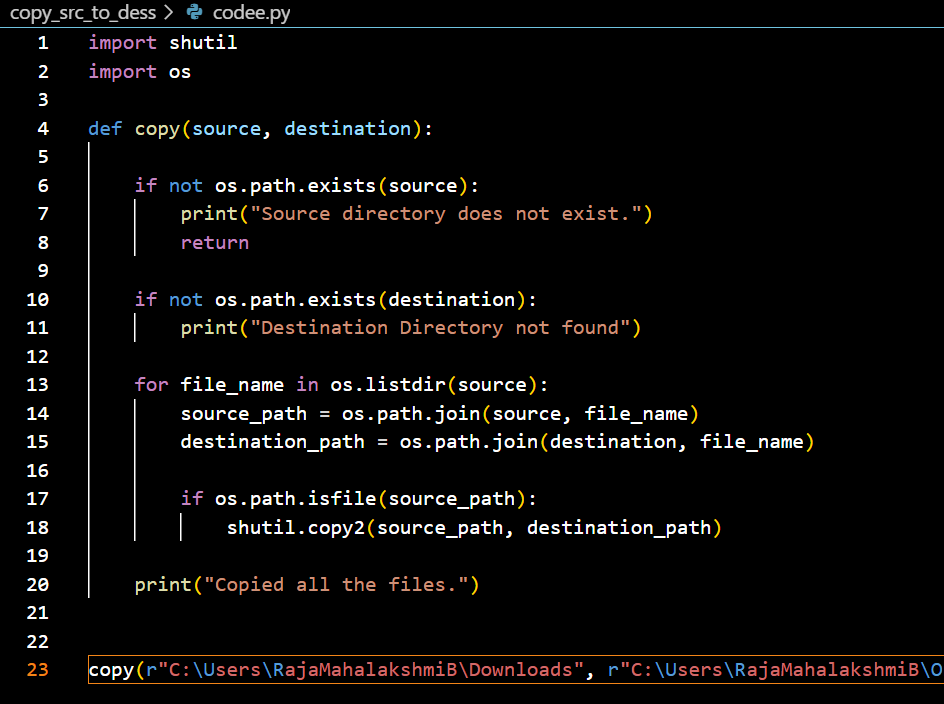
**Python Function to copy all files from Source to Destination**

****

**1.Import required modules**

import shutil

import os

shutil → Used for file operations like copying, moving, or deleting files.

os → Helps interact with the operating system, such as checking if a folder exists or listing files

**2.Define the function**

def copy(source,destination):

This function takes two inputs:

* source: The folder where files are currently stored.
* destination: The folder where files should be copied.

**3.Check if source folder exists**

if not os.path.exists(source):

print("Source directory does not exist.")

return

os.path.exists(source) → Checks if the source folder exists.

If the source does not exist, it prints an error message and stops execution (return).

**4. Checking If the Destination Folder Exists**

if not os.path.exists(destination):

print("Destination Directory not found")

* Similar to the source, this checks if the destination folder exists.
* But here, it only prints a warning and does not stop the program.

Issue: If the destination does not exist, the program should create it using os.makedirs(destination). Currently, it does not create the folder, which may cause an error if the folder is missing.

**5. Looping Through Files in the Source Folder**

for file\_name in os.listdir(source):

* os.listdir(source) → Gets a list of all files and folders inside the source directory.
* The for loop goes through each item one by one.

**6.Creating Full Paths for Source and Destination**

source\_path = os.path.join(source, file\_name)

destination\_path = os.path.join(destination, file\_name)

os.path.join(source, file\_name) → Combines the folder path and the file name to get the full path.

This is done for both source and destination files.

**7. Checking If the Item is a File**

if os.path.isfile(source\_path):

* os.path.isfile(source\_path) → Ensures that the item is a file and not a folder.
* This is necessary because os.listdir(source) lists both files and folders.

**8. Copying the File to the Destination**

shutil.copy2(source\_path, destination\_path)

shutil.copy2() → Copies the file along with metadata (like creation time and permissions).

The copied file will have the same name in the destination folder.

**9.Calling the function**

* The r before the string (r"path") makes it a raw string, which prevents issues with backslashes (\).

**Fact Table vs. Dimension Table**

**Fact Table**

A Fact Table contains numerical data that represents business events or transactions. It primarily consists of quantitative metrics that help in analyzing business performance. Each record in a fact table is linked to multiple Dimension Tables using foreign keys.

**Key Features of a Fact Table:**

* Stores numerical values like sales amount, order quantity, or revenue.
* Includes foreign keys that reference dimension tables.
* Usually the largest table in a data warehouse due to high data volume.
* Each row represents a unique event, such as a purchase or transaction.

**Examples of Fact Tables:**

* **Sales Data** → Contains transaction details like order amount, discount, and total revenue.
* **Inventory Data** → Holds stock levels, restock quantity, and reorder status.
* **Website Traffic Data** → Tracks page views, clicks, and time spent on a page.

**Dimension Table**

A Dimension Table contains descriptive (qualitative) data that provides context for the numbers in a fact table. These tables store details about people, places, products, and time.

**Key Features of a Dimension Table:**

* Stores textual or categorical information (e.g., customer name, product category).
* Has a primary key that uniquely identifies each record.
* Is smaller in size compared to the fact table.
* Provides context for the numbers in the fact table.

**Examples of Dimension Tables:**

* **Customer Table** → Stores customer details like name, email, location.
* **Product Table** → Holds product details like name, category, and brand.
* **Date Table** → Includes information like year, quarter, month, and day.

**Key Differences Between Fact and Dimension Tables**

| **Feature** | **Fact Table** | **Dimension Table** |
| --- | --- | --- |
| **Data Type** | Quantitative (numbers) | Qualitative (descriptive) |
| **Purpose** | Stores business events & metrics | Provides context to fact data |
| **Primary Key** | Usually a **surrogate key** | Has a **primary key** |
| **Foreign Keys** | Contains foreign keys referencing dimension tables | Not dependent on fact tables |
| **Size** | Large (stores massive transactional data) | Smaller in comparison |

**Why Are Fact and Dimension Tables Important?**

* **Fact Tables** help analyze business performance through measurable data.
* **Dimension Tables** provide context and meaning to the numbers in the fact table.
* Together, they enable efficient data retrieval, filtering, and reporting in data analytics.

This structured approach allows businesses to store and analyze data effectively, helping in making data-driven decisions.

**Surrogate key:**

A surrogate key is a unique identifier for a record in a table, usually a system-generated number. Unlike natural keys (like a customer ID or product code), a surrogate key has no real-world meaning—it is just used to ensure each record has a unique identity.

Example:

Imagine a Student Table where each student has a Student ID assigned by the system (like 101, 102, 103). Even if two students have the same name, their Student ID will be different. This Student ID acts as a surrogate key.

Why Use a Surrogate Key?

* Ensures uniqueness
* Doesn't change over time (unlike names or phone numbers)
* Simplifies database relationships

It’s commonly used in fact and dimension tables in data warehouses.

**Api ingestion**

Fetched data from an API URL using the requests module and written text cases using pytest framework

A computer screen shot of a program code

AI-generated content may be incorrect.

