

# Using Python for Complex Interconnected Calculations

---



**Janani Ravi**

CO-FOUNDER, LOONYCORN

[www.loonycorn.com](http://www.loonycorn.com)

# Overview

**Transaction and analytical processing**

**Control flow and control structures**

**Comparisons and logical operators**

**Branching based on conditions**

**Iterating based on conditions**

**Iterating over lists, tuples and dictionaries**

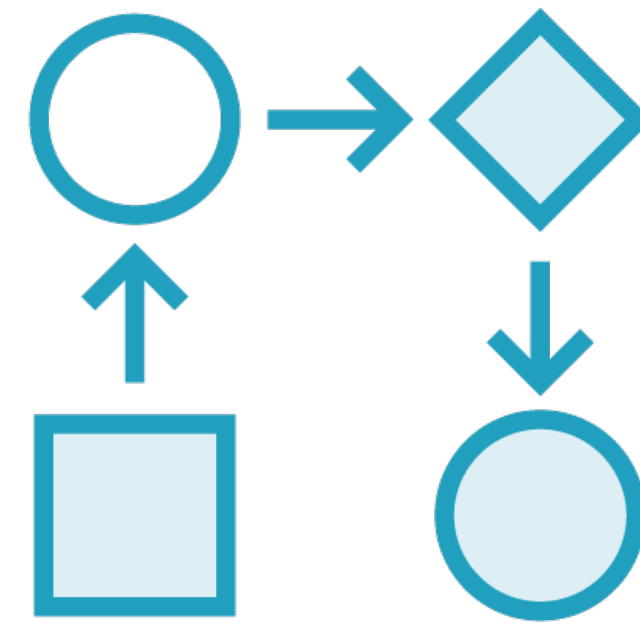
**Break, continue and pass**

# Two Hats of a Data Professional



## Find the Dots

Identify important elements in a dataset



## Connect the Dots

Explain those elements via relationships with other elements

# Essential Steps in Connecting the Dots

**Processing Data for  
Use in Models**

**Building and Refining  
Models**

**Incorporating Real-  
world Data into  
Models**

# Essential Steps in Connecting the Dots

Processing Data for  
Use in Models

Building and Refining  
Models

**Incorporating Real-  
world Data into  
Models**

# Transactional and Analytical Processing



## **Order Management Support**

**John is responsible for tracking and delivering orders on time**



## **Revenue Analyst**

**Anna is responsible for tracking and monitoring revenues**

# Order Management Support



**20 deliveries in Kent, WA are delayed**

**The courier company has had a computer outage**

**John assigns the orders to another courier company in the region**

# Revenue Analyst



**Her manager wants an update on last month's revenues**

**Last month was an unusually slow one**

**Anna pulls up data for the last 5 years to check for seasonal effects**



# Transactional and Analytical Processing



**Transactional Processing**



**Analytical Processing**

# Transactional and Analytical Processing

## Transactional Processing

Ensure correctness of **individual entries**

Access to **recent** data, from the last few hours or days

**Updates** data

Fast **real-time** access

Usually a **single** data source

## Analytical Processing

Analyzes **large batches** of data

Access to **older** data going back months, or even years

Mostly **reads** data

**Long** running jobs

**Multiple** data sources

For Analytical Processing use-cases,  
Python is a serious alternative to  
running SQL on data warehouses

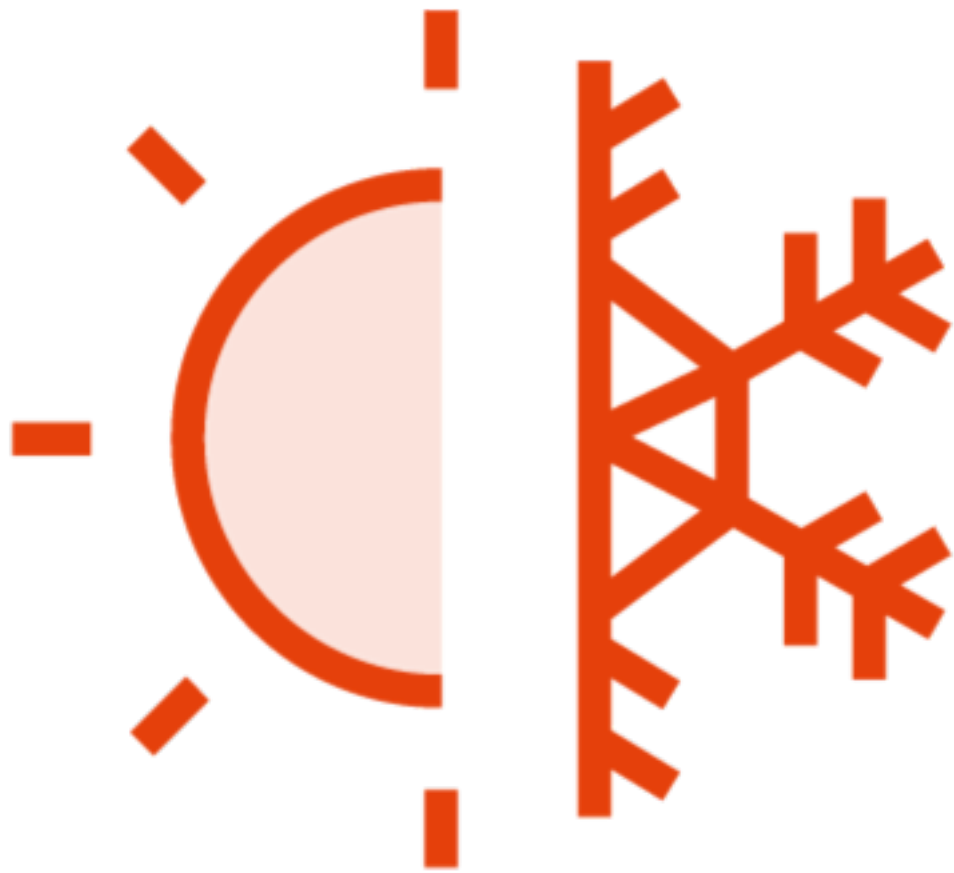
# Essential Analytical Building Blocks

**Conditional  
Execution**

**Repeated  
Execution**

**Re-use of Logic**

# Conditional Execution



**Spreadsheets make it very hard to change logic based on conditions**

- Only locally possible (within a cell)

**Same holds for SQL-based technologies**

- Sub-queries

# Transactional and Analytical Processing



## Small Data

Both these objectives could be achieved  
using the **same** database system

# Transactional and Analytical Processing



## Big Data

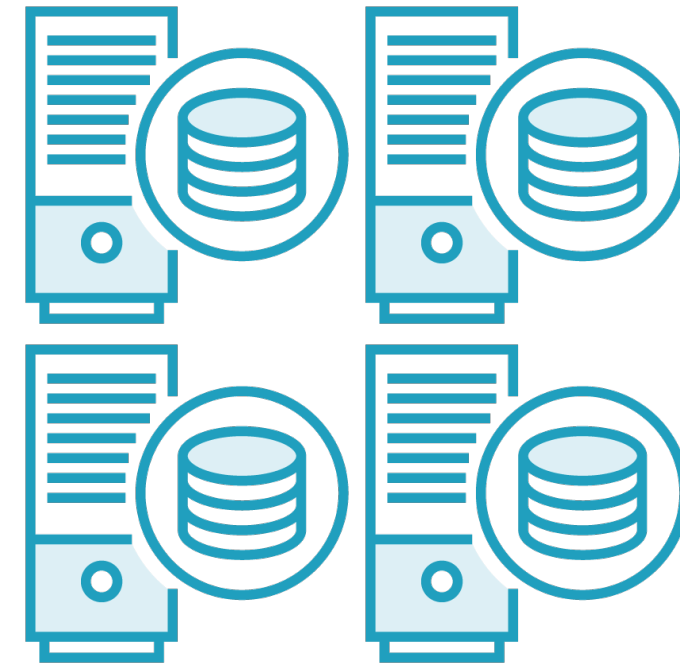
**Very hard** to meet all requirements  
with the **same** database system

# Transactional and Analytical Processing



**Transactional Processing**

**Traditional  
RDBMS**

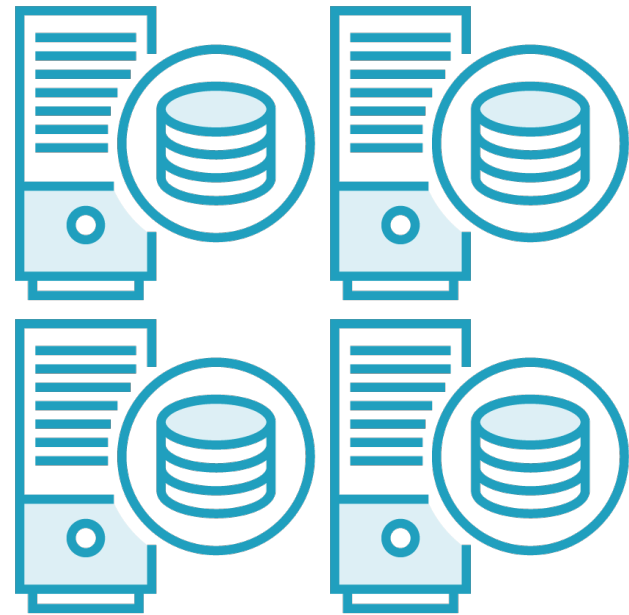


**Analytical Processing**

**Data Warehouse**



# 3 Vs of Big Data



**Volume: Amount of data**

**Variety: Number and type of sources**

**Velocity: Batch and streaming**

Python works seamlessly with several  
technologies data warehousing  
technologies for building ML models  
with big data

# Demo

**Conditional execution using if, if-else,  
and if-elif statements**

# Demo

**Working with for-loops to perform a sequence of operations a predefined number of times**

# Demo

**Using while loops to perform a sequence of operations so long as a condition evaluates to true**

# Demo

**The break, continue, and pass keywords to control execution**

# Summary

**Transaction and analytical processing**

**Control flow and control structures**

**Comparisons and logical operators**

**Branching based on conditions**

**Iterating based on conditions**

**Iterating over lists, tuples and dictionaries**

**Break, continue and pass**