Data Analyst with AI

A comprehensive professional development programme designed to transform aspiring analysts into AI-powered data professionals. This curriculum bridges traditional data analysis with cutting-edge artificial intelligence, covering everything from foundational IT concepts to advanced agentic AI systems.

Fundamentals of IT & AI

Build your foundation with application lifecycle management, Agile methodologies, and AI fundamentals

Python for AI & Data Analysis

Master Python programming from basics to advanced OOP concepts for data manipulation

SQL for AI & Data Analysis

Develop expertise in PostgreSQL, query optimization, and database design principles

Excel & Advanced Excel

Transform data with PivotTables, Power Query, and advanced analytical functions

PowerBI for Data Analysis

Create compelling visualisations and interactive dashboards with DAX and data modelling

Generative AI & Agentic AI

Harness LLMs, LangChain, and production AI systems for next-generation analytics

Digital Edify

India's First AI-Native Training Institute

Learn AI. Build Agents. Lead Future.

hello@digitaledify.ai

+91 6304982304

About Digital Edify

India's #1 Training Institute for the AI Era

Established: 2016

Headquarters: Hyderabad, Telangana

Reach: Global (Online + Offline)

The Transformation Narrative

Digital Edify has evolved from a premium training institute in the Automation Era to an AI-first organisation leading the Agentic AI revolution. Since 2016, we've transformed over 100,000 professionals and built partnerships with more than 1,000 industry leaders. Our journey reflects the technological evolution of our time—from traditional job placement to career transformation, and now to building AI-native professionals who will shape the future of work.

1

Automation Era (2016-2023)

Premium Training Institute focused on job placement with 100K+ students trained

2

AI Revolution (2024-2025)

AI-Powered Training with industry-AI integration and career transformation focus

3

Agentic AI Leadership (2026+)

AI First Institute building AI-Native Professionals with 1 Million AI-Native Vision

> "We started in the Automation Era. We evolved through the AI Revolution. Now, we're leading the Agentic AI Future—with 100,000+ professionals already transformed and 1,000+ industry partners trusting our graduates."

Vision & Mission

Vision

"To Create 1 Million AI-Native Professionals Who Will Build the Agentic Future of Work"

Mission

"We transform learners into AI-native professionals through industry-aligned programmes that integrate Agentic AI into every discipline—from development to data science to enterprise platforms."

Course Highlights

Section 1: Fundamentals of IT & AI

Learn core IT systems, agile workflows, computing basics, and AI foundations.

Section 2: Python for AI & Data

Develop strong Python skills for data handling, automation, and AI use cases.

Section 3: SQL for AI & Data

Query, design, and optimize databases for analytics and AI-driven applications.

Section 4: Excel & Advanced Excel for Data Analysis

Analyze, visualize, and automate data using advanced Excel tools and models.

Section 5: Power BI for Data Analysis

Build interactive dashboards and enterprise analytics with Power BI and DAX.

Section 6: Generative AI & Agentic AI

Build, deploy, and manage intelligent AI and agent-based systems at scale.

SECTION 1

Application Life Cycle Management

Application Fundamentals

- What defines an application
- Types of applications and their characteristics
- Web application architecture
- Client-server relationships

Web Technologies Stack

- Frontend: HTML, CSS, JavaScript, React
- Backend: Python, Java, Node.js
- SQL databases: MySQL, PostgreSQL
- NoSQL databases: MongoDB

1. **Planning**
   Define project scope and requirements
2. **Analysis**
   Examine feasibility and resources
3. **Design**
   Create system architecture
4. **Implementation**
   Build and code the solution
5. **Testing**
   Verify quality and functionality
6. **Deployment**
   Release to production
7. **Maintenance**
   Support and enhance

Agile & Scrum Framework

Waterfall vs Agile

Compare traditional sequential approaches with iterative Agile methodologies

Agile Mindset

Embrace flexibility, collaboration, and continuous improvement principles

Popular Frameworks

Explore Scrum, Kanban, and other Agile implementation approaches

Scrum Framework Components

Scrum Roles

- Product Owner
- Scrum Master
- Development Team

Scrum Events

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Scrum Artifacts

- Product Backlog
- Sprint Backlog
- Increment

01

Writing User Stories

Craft clear, actionable requirements from user perspectives

02

Defining Epics and Themes

Organise large bodies of work into manageable components

03

Setting Acceptance Criteria

Establish clear definitions of done for each story

04

Estimating Stories

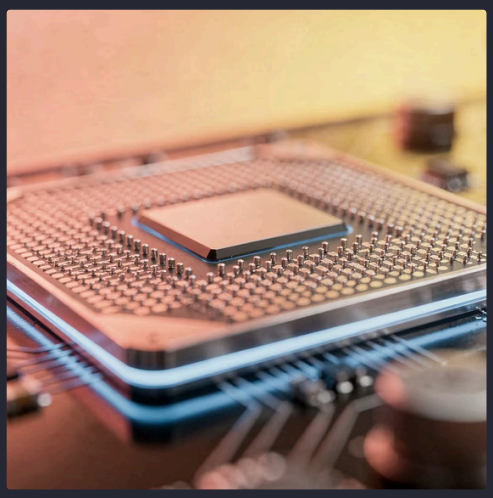
Apply story points and planning poker techniques

05

Managing Backlogs

Prioritise and refine work using tools like Azure Boards

Computing & Data Infrastructure



Central Processing Unit

General-purpose processors handling sequential operations and traditional computing tasks

Graphics Processing Unit

Parallel processing powerhouses accelerating AI training and complex data operations

Cloud Computing Revolution

Cloud platforms have democratised access to enterprise-grade computing resources, enabling data analysts to scale their work without massive infrastructure investments. Understanding cloud service models helps you choose the right level of abstraction for your projects.

Infrastructure as a Service (IaaS)

Virtualised computing resources including servers, storage, and networking. Maximum control and flexibility for custom configurations.

Platform as a Service (PaaS)

Development and deployment platforms with managed infrastructure. Focus on building applications without managing underlying systems.

Software as a Service (SaaS)

Ready-to-use applications delivered over the internet. Immediate access to tools without installation or maintenance overhead.

Introduction to AI, Generative AI & Agentic AI

Artificial intelligence has evolved from a theoretical concept to a practical tool that's reshaping data analysis. This module demystifies AI technologies, from foundational machine learning to cutting-edge generative models, providing the conceptual framework you need to leverage AI effectively in your analytical work.

Artificial Intelligence

Machines performing tasks requiring human intelligence

Generative AI

Models creating new content from learned patterns

Machine Learning

Systems learning from data without explicit programming

Deep Learning

Neural networks with multiple layers processing complex patterns

Generative AI Technologies

Large Language Models

Transformer-based models like GPT, Claude, and Gemini that understand and generate human-like text. These models power chatbots, content generation, code assistance, and sophisticated data analysis tasks.

Image Generation Models

Diffusion models and GANs creating photorealistic images from text descriptions. Applications span creative design, data visualisation enhancement, and synthetic data generation for training.

AI in Everyday Learning: From personalised recommendations to intelligent search, AI systems are already embedded in the tools you use daily. Understanding these technologies empowers you to leverage them strategically in your analytical workflows.

Real-World Applications

Customer Relationship Management

CRM systems centralise customer interactions, sales pipelines, and marketing campaigns. Data analysts extract insights on customer behaviour, sales performance, and campaign effectiveness to drive revenue growth and improve customer satisfaction.

Retail & E-Commerce Applications

E-commerce platforms generate vast datasets on customer behaviour, inventory, and transactions. Analysts optimise pricing strategies, forecast demand, personalise recommendations, and identify trends that inform merchandising and marketing decisions.

Human Resource Management Systems

HRMS platforms manage employee data, recruitment, performance, and payroll. Analysts leverage this data for workforce planning, retention analysis, compensation benchmarking, and identifying factors that drive employee engagement and productivity.

Healthcare Applications

Healthcare systems manage patient records, treatment outcomes, and operational metrics. Data analysts improve patient care through outcome analysis, resource optimisation, predictive modelling for disease prevention, and operational efficiency improvements.

SECTION 2

Python Fundamentals

Python has become the lingua franca of data analysis and AI, prized for its readability and extensive ecosystem. This module establishes your Python foundation, covering everything from basic syntax to control flow structures. You'll learn to think programmatically whilst writing clean, efficient code that forms the basis for more advanced data manipulation.

1

Environment Setup

Install Python interpreter and configure Visual Studio Code for optimal development experience

2

Core Syntax

Master Python's 35 keywords, naming conventions, and fundamental language structure

3

Data Types & Variables

Work with simple and complex types, understanding memory management and type conversion

4

Operators

Apply arithmetic, comparison, logical, and other operators for data manipulation

5

Control Flow

Implement conditional logic with if/elif/else and match-case statements

6

Loops

Iterate efficiently using while and for loops with range() function and control statements

> "Python's elegant syntax allows you to focus on solving problems rather than wrestling with language complexity. This accessibility makes it the perfect entry point for data analysis."

String Manipulation

Text data pervades modern datasets—from customer feedback to product descriptions. Mastering string manipulation is essential for cleaning, transforming, and extracting insights from textual information. This module covers Python's comprehensive string handling capabilities, from basic operations to sophisticated pattern matching and formatting techniques.

String Fundamentals

- String definition and syntax rules
- Positive and negative indexing
- Slicing with start:end:step notation
- Concatenation and repetition operations
- Understanding string immutability

String Formatting

- Modern f-strings for interpolation
- format() method for templates
- Alignment and padding techniques
- Number and date formatting

Essential String Methods

Case Conversion

upper(), lower(), title(), capitalize() for standardising text data

Search Operations

find(), index(), count() for locating substrings and patterns

Validation Checks

isalpha(), isdigit(), isalnum() for data quality verification

Trimming Methods

strip(), lstrip(), rstrip() for removing unwanted whitespace

Replacement

replace() for substituting text patterns throughout strings

Split & Join

split(), join() for parsing and reconstructing delimited data

Data Structures: Lists & Tuples

Lists and tuples are Python's fundamental sequential data structures, essential for storing and manipulating collections of data. Lists offer flexibility with their mutability, whilst tuples provide immutable, memory-efficient alternatives. Understanding when and how to use each structure is crucial for writing efficient data processing code.

Lists: Dynamic Collections

Mutable, ordered sequences supporting indexing, slicing, and comprehensive modification operations. Perfect for data that changes during processing.

Tuples: Immutable Sequences

Fixed, ordered collections that cannot be modified after creation. Ideal for data integrity, dictionary keys, and function returns.

List Operations & Methods

Adding Elements

- append() - add single item
- insert() - add at position
- extend() - add multiple items

Removing Elements

- remove() - by value
- pop() - by index
- clear() - remove all

Searching & Sorting

- index() - find position
- count() - occurrences
- sort() - order elements
- reverse() - flip order

01

List Comprehensions

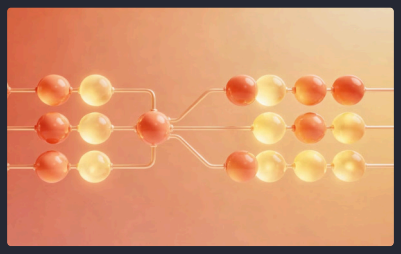
Concise syntax for creating lists from iterables with optional filtering and transformation

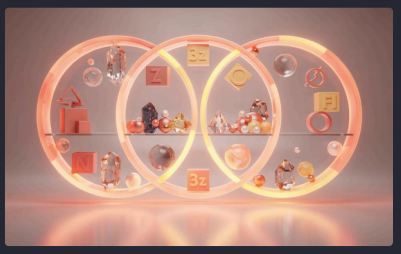
02

Tuple Packing & Unpacking

Elegant techniques for assigning multiple variables simultaneously and returning multiple values

Data Structures: Dictionaries & Sets





Dictionaries

Unordered key-value mappings providing O(1) lookup performance. Essential for data transformation, configuration management, and representing structured records.

Sets

Unordered collections of unique elements supporting mathematical operations. Perfect for deduplication, membership testing, and finding relationships between datasets.

Dictionary Operations

Core Methods

- keys() - retrieve all keys
- values() - retrieve all values
- items() - retrieve key-value pairs
- get() - safe key access
- update() - merge dictionaries

Advanced Techniques

- Dictionary comprehensions
- Nested dictionaries for hierarchical data
- Default values and error handling
- Practical data mapping applications

Set Operations

Union

Combine all unique elements from multiple sets

Intersection

Find common elements across sets

Subset & Superset

Test containment relationships between sets

Difference

Elements in one set but not another

Advanced Collections & Iterators

namedtuple

Lightweight objects with named fields, combining tuple efficiency with attribute access readability

Counter

Specialized dictionary for counting hashable objects, perfect for frequency analysis

defaultdict

Dictionary with default values for missing keys, eliminating KeyError exceptions

deque

Double-ended queue with O(1) append and pop operations from both ends

Iterators & Generators

Iterator Protocol

Objects implementing __iter__() and __next__() methods for sequential access. Custom iterators enable controlled iteration over complex data structures with memory efficiency.

Generator Functions

Functions using yield to produce values lazily, one at a time. Generators provide memory-efficient iteration over large datasets without loading everything into memory.

01

Lambda Functions

Anonymous single-expression functions for concise operations

02

map() Function

Apply function to every item in iterable

03

filter() Function

Select items matching condition

04

reduce() Function

Accumulate values into single result

Functions & Scope

Function Basics

Define functions with def keyword, call them with arguments, and return values. Functions encapsulate logic for reuse across your codebase.

Parameter Types

Positional arguments maintain order, keyword arguments provide clarity, default arguments offer flexibility, and *args/**kwargs handle variable inputs.

Scope Management

Local variables exist within functions, global variables span the module, and the global keyword enables modification of module-level variables.

Advanced Function Concepts

Multiple Returns

Return multiple values as tuples, enabling elegant unpacking and cleaner function interfaces.

Lambda Functions

Single-expression anonymous functions for inline operations and functional programming patterns.

Recursion

Functions calling themselves to solve problems through divide-and-conquer strategies.

Docstrings

Document functions with triple-quoted strings describing purpose, parameters, and return values for maintainable code

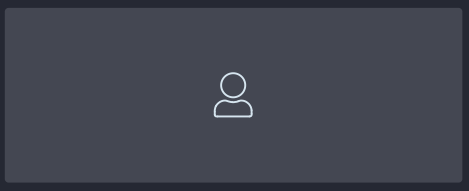
Built-in Functions

Leverage Python's extensive library of built-in functions like len(), sum(), max(), min() for common operations

Modules & Packages

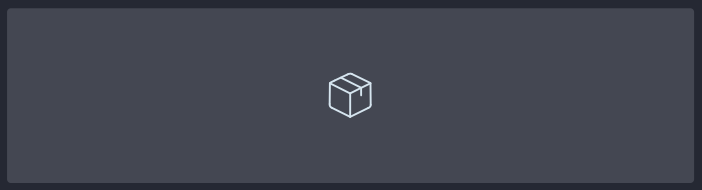
Built-in Modules

Standard library modules included with Python



User-Defined Modules

Custom modules you create for your projects



External Packages

Third-party libraries from PyPI ecosystem

Essential Built-in Modules

math

Mathematical functions and constants for numerical operations

random

Random number generation and sampling for simulations

datetime

Date and time manipulation for temporal data analysis

os

Operating system interfaces for file and directory operations

sys

System-specific parameters and functions for runtime control

Package Management

Creating Packages

Structure directories with __init__.py files to create importable packages. Organize related modules into logical hierarchies with nested packages for complex projects.

pip & Dependencies

Install external packages with pip, manage project dependencies using requirements.txt, and leverage popular packages like requests, pandas, and numpy for data work.

Working with Data Formats

Data rarely arrives in perfect Python objects—it comes in files. This module teaches you to read, write, and manipulate data in various formats including text files, CSV, and JSON. These skills are fundamental for data ingestion, transformation, and export workflows that connect Python to the broader data ecosystem.

File Operations

Master CRUD operations with open() function, understanding file modes (r, w, a, r+). Learn reading methods (read, readline, readlines) and writing techniques (write, writelines) for text file manipulation.

CSV Files

Work with comma-separated values using csv.reader and csv.writer for basic operations. Leverage csv.DictReader and csv.DictWriter for dictionary-based access, enabling cleaner, more maintainable data processing code.

JSON Files

Handle JavaScript Object Notation with json.dump/dumps for serialization and json.load/loads for deserialization. JSON's hierarchical structure makes it ideal for configuration files and API data exchange.

File Handling Best Practices

- ## Context Managers

  Use 'with' statements to ensure files are properly closed, even when exceptions occur

- ## Path Operations

  Leverage os.path and pathlib for cross-platform file path manipulation and directory management

- ## Error Handling

  Implement try-except blocks to gracefully handle missing files and permission errors

- ## Data Validation

  Verify data integrity during file operations to prevent downstream processing errors

Advanced Python Concepts

Exception Handling

1

try Block

Code that might raise exceptions

2

except Block

Handle specific exception types

3

else Block

Execute if no exceptions occur

4

finally Block

Always execute cleanup code

Exception Techniques

- Catching specific exception types
- Raising exceptions with raise
- Re-raising exceptions for propagation
- Creating custom exception classes
- Understanding built-in exception hierarchy

Decorators

- Function decorators for behaviour modification
- Decorators with arguments
- Stacking multiple decorators
- Class decorators for class enhancement
- Practical applications: logging, timing, caching

Generators & Context Managers

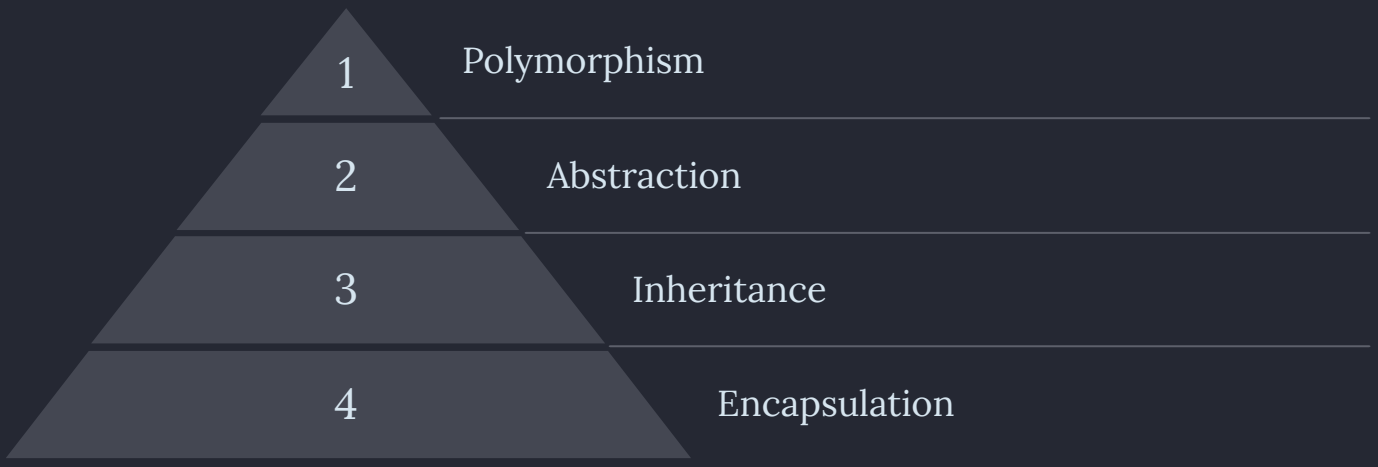
Generator Deep Dive

Explore generator expressions, infinite generators, and memory-efficient iteration patterns for processing large datasets without loading everything into memory.

Context Managers

Implement custom context managers using __enter__ and __exit__ methods or @contextmanager decorator for resource management, ensuring proper setup and cleanup.

Object-Oriented Programming



Classes & Objects Fundamentals

Core Concepts

- Classes as blueprints for objects
- Instance and class variables
- __init__ constructor method
- Understanding self parameter
- Instance methods for behaviour

Method Types

- Instance methods (default)
- Class methods with @classmethod
- Static methods with @staticmethod
- Special/magic methods (__str__, __repr__, __len__)

The Four Pillars

Encapsulation

Bundle data and methods, control access with public, protected (_), and private (__) modifiers

Inheritance

Single, multi-level, and multiple inheritance for code reuse. Use super() to access parent class methods

Abstraction

Abstract base classes (ABC module) define interfaces without implementation, enforcing contracts

Polymorphism

Method overriding and duck typing enable objects of different classes to be used interchangeably

SECTION 3

Foundations of Databases & PostgreSQL

RDBMS Fundamentals

Relational Database Management Systems organize data into tables with relationships, enabling structured storage and powerful querying capabilities

ACID Properties

Atomicity, Consistency, Isolation, Durability guarantee reliable transaction processing and data integrity

PostgreSQL

Enterprise-grade open-source database with advanced features, extensibility, and standards compliance

Database Objects & Data Types

Core Objects

- Databases - top-level containers
- Schemas - logical groupings
- Tables - structured data storage
- Columns - individual data fields

Data Types

- Numeric: INTEGER, DECIMAL, FLOAT
- Character: VARCHAR, TEXT, CHAR
- Date/Time: DATE, TIMESTAMP, INTERVAL
- Boolean, JSON, Arrays, and more

Constraints & Referential Integrity

PRIMARY KEY

Uniquely identifies each row

FOREIGN KEY

Enforces relationships between tables

UNIQUE

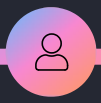
Ensures column values are distinct

NOT NULL

Requires value in column

CHECK

Validates data against condition



DEFAULT

Provides automatic value

Querying and Analyzing Data

SQL's true power emerges when querying and analyzing data. This comprehensive module covers everything from basic SELECT statements to sophisticated window functions and multi-table joins. You'll master filtering, aggregation, and analytical techniques that transform raw data into actionable insights.

01

SELECT Basics

Retrieve columns with aliases and expressions

02

WHERE Filtering

Apply conditions with comparison and logical operators

03

ORDER BY

Sort results ascending or descending

04

DISTINCT

Remove duplicate rows from results

05

LIMIT & OFFSET

Implement pagination for large result sets

Functions & Aggregations

String Functions

- UPPER, LOWER, CONCAT
- SUBSTRING, TRIM, LENGTH
- Pattern matching with LIKE

Numeric Functions

- ROUND, CEIL, FLOOR
- ABS, MOD, POWER
- Mathematical operations

Date/Time Functions

- CURRENT_DATE, NOW()
- EXTRACT, DATE_TRUNC
- Interval arithmetic

Advanced Analytics

Aggregate Functions

COUNT, SUM, AVG, MIN, MAX for summarizing data across rows

GROUP BY & HAVING

Group rows by columns and filter aggregated results

Window Functions

ROW_NUMBER, RANK, LAG, LEAD for advanced analytical queries

JOIN Operations

INNER, LEFT, RIGHT, FULL OUTER, CROSS, and SELF joins for combining tables

Advanced Queries & Data Manipulation

Subqueries & CTEs

Subqueries

Nested queries in WHERE, SELECT, and FROM clauses. Correlated subqueries reference outer query values.

Common Table Expressions

WITH clauses create named temporary result sets. Recursive CTEs handle hierarchical data like organizational charts.

Multiple CTEs

Chain multiple CTEs together for complex, readable query logic that's easier to debug and maintain.

Set Operators

UNION

Combine results from multiple queries, removing duplicates. UNION ALL includes duplicates for better performance.

INTERSECT

Return only rows that appear in both query results, useful for finding common elements.

EXCEPT

Return rows from first query that don't appear in second, perfect for difference analysis.

Data Manipulation & Transactions

UPDATE Statements

Modify existing data with expressions and JOIN operations for complex updates

DELETE Operations

Remove rows with subqueries. TRUNCATE vs DELETE for different use cases

Transaction Management

BEGIN, COMMIT, ROLLBACK with savepoints and isolation levels for data integrity

Database Programming & Automation

Professional database work requires automation through stored functions, procedures, and triggers. This module teaches you to write PL/pgSQL code that executes within the database, enabling complex business logic, automated data validation, and audit logging whilst maintaining data integrity and performance.

Schema Modifications

ALTER TABLE operations for adding, modifying, and dropping columns. Manage constraints dynamically as requirements evolve.

Indexes & Performance

Create B-tree, Hash, GIN, and GiST indexes. Understand when indexes improve performance and when they add overhead.

Views & Abstraction

Create regular and materialized views for data abstraction. Updatable views and refresh strategies for cached results.

PL/pgSQL Programming

Stored Functions

- PL/pgSQL language fundamentals
- Function parameters and return types
- Control structures: IF, CASE, LOOP
- Functions returning tables
- Exception handling in functions

Stored Procedures

- Procedure vs function differences
- Transaction control in procedures
- Parameter modes: IN, OUT, INOUT
- Calling procedures from applications

Triggers & Automation

1

BEFORE Triggers

Execute before data modification for validation

2

AFTER Triggers

Execute after modification for audit logging

3

INSTEAD OF Triggers

Replace default action on views

4

Trigger Functions

PL/pgSQL functions called by triggers

Database Design & Optimization

Entity-Relationship Modelling

Entities

Objects or concepts with independent existence

Attributes

Properties describing entities

M:N Relationships

Many-to-many associations

Relationships

Associations between entities

1:M Relationships

One-to-many associations

1:1 Relationships

One-to-one associations

Normalization

First Normal Form (1NF)

Eliminate repeating groups, ensure atomic values in each column

Second Normal Form (2NF)

Remove partial dependencies, ensure non-key attributes depend on entire primary key

Third Normal Form (3NF)

Eliminate transitive dependencies, ensure non-key attributes depend only on primary key

Query Optimization

EXPLAIN Analysis

Read execution plans to understand query performance and identify bottlenecks

Index Strategies

Choose appropriate index types and columns for optimal query performance

Query Rewriting

Restructure queries for better execution plans and reduced resource consumption

Maintenance

VACUUM, ANALYZE, and connection pooling for sustained performance

SECTION 4

Excel Fundamentals & Interface Mastery

Microsoft Excel remains the world's most ubiquitous data analysis tool. This module establishes your Excel foundation, covering interface navigation, cell references, basic formulas, and formatting techniques. You'll develop the muscle memory and conceptual understanding necessary for efficient spreadsheet work.

Interface Mastery

Navigate the Ribbon, Quick Access Toolbar, and workbook structure efficiently. Master keyboard shortcuts that dramatically accelerate your workflow and productivity.

Cell References

Understand relative, absolute ($), and mixed references for flexible formula design. Proper reference usage is fundamental to creating maintainable spreadsheets.

Essential Functions

Master SUM, AVERAGE, COUNT, MIN, MAX and arithmetic operators. These building blocks form the foundation for more complex calculations.

Formatting

Apply number formats, currency, dates, and custom formats. Conditional formatting highlights patterns and exceptions automatically for visual analysis.

Workbook Management

Organization

- Working with multiple sheets
- Named ranges for clarity
- Protecting sheets and workbooks
- File formats: .xlsx, .xlsm, .csv

Output

- Page setup and print areas
- Headers and footers
- PDF export for sharing
- Data validation for input control

Data Organization, Analysis & Visualization

Raw data requires organization before analysis. This module teaches you to structure, clean, and visualize data effectively. You'll master sorting, filtering, Excel Tables, and chart creation—transforming messy datasets into clear, actionable insights through both tabular analysis and visual storytelling.

01

Data Organization

Structure data following best practices for analysis

02

Sorting & Filtering

Single and multi-level sorts, AutoFilter, Advanced Filter

03

Data Cleaning

Remove duplicates, Flash Fill, text functions

04

Excel Tables

Format as Table with structured references

05

Visualization

Create compelling charts and dashboards

Essential Functions

Text Functions

- LEFT, RIGHT, MID
- CONCATENATE, TEXTJOIN
- TRIM, UPPER, LOWER

Date/Time Functions

- TODAY, NOW, DATE
- DATEDIF, WORKDAY
- NETWORKDAYS

Logical Functions

- IF, AND, OR, NOT
- IFS, SWITCH
- Nested logic

Chart Types & Visualization

Column & Bar Charts

Compare values across categories

Line Charts

Show trends over time

Pie Charts

Display proportions of a whole

Scatter Plots

Reveal relationships between variables

Advanced Formulas & Lookup Functions

Lookup Functions

VLOOKUP & HLOOKUP

Classic lookup functions searching vertically or horizontally through tables

XLOOKUP & XMATCH

Modern, flexible lookup functions replacing VLOOKUP with enhanced capabilities

INDEX & MATCH

Powerful combination for two-dimensional lookups with maximum flexibility

Array Formulas & Dynamic Arrays

Traditional Arrays

CSE formulas (Ctrl+Shift+Enter) perform calculations across ranges, enabling complex multi-cell operations in a single formula.

Dynamic Arrays (Excel 365)

FILTER, SORT, SORTBY, UNIQUE functions automatically spill results across multiple cells, revolutionizing data manipulation.

Statistical & Financial Functions

Conditional Aggregation

SUMIF, SUMIFS, COUNTIF, COUNTIFS, AVERAGEIF, AVERAGEIFS for criteria-based calculations

Mathematical Functions

ROUND, ROUNDUP, ROUNDDOWN, MOD, ABS for precise numerical operations

Financial Functions

PMT, FV, PV, RATE, NPV, IRR for investment and loan calculations

Error Handling

IFERROR, IFNA, ISERROR for graceful error management in formulas

PivotTables & PivotCharts

Create PivotTable

Select data source and insert PivotTable

Configure Fields

Drag fields to Rows, Columns, Values, Filters

Customize Values

Set aggregation: Sum, Count, Average, % of Total

Group & Format

Group dates/numbers, apply styles

Add Interactivity

Insert Slicers and Timelines for filtering

Advanced PivotTable Features

Calculated Fields & Items

- Create custom calculations within PivotTables
- Calculated fields for new metrics
- Calculated items for custom groupings
- Combining multiple data sources

PivotCharts

- Visual representation of PivotTable data
- Interactive charts linked to PivotTables
- Chart types optimized for pivot data
- Refreshing data and maintaining formatting

Introduction to Power Query

Get & Transform

Connect to Excel, CSV, folders, databases, and web sources

Data Transformation

Clean, reshape, and combine data before analysis

Repeatable Workflows

Save transformation steps for automatic refresh

Advanced Analytics, Automation & Data Modeling

Excel Data Model & Power Pivot

1

Import Tables

Load multiple tables into Data Model

2

Create Relationships

Define primary and foreign key connections

3

DAX Calculations

Build calculated columns and measures

4

Analyze

Create PivotTables from Data Model

DAX Fundamentals

Data Analysis Expressions (DAX) enable sophisticated calculations within Power Pivot. Learn to create calculated columns that extend your data model and measures that aggregate data dynamically based on PivotTable context.

Relationship Types

One-to-many relationships connect dimension tables to fact tables. Understanding cardinality and filter direction ensures accurate calculations across related tables in your data model.

What-If Analysis & Optimization

Goal Seek

Find input value needed to achieve desired output result

Scenario Manager

Compare multiple sets of input values and their outcomes

Data Tables

One and two-variable tables showing result variations

Solver

Optimize objectives subject to constraints for complex problems

SECTION 5

Introduction to Business Intelligence & Power BI

Business Intelligence

Transform raw data into actionable insights through visualization, analysis, and reporting

Power BI Architecture

Desktop for development, Service for sharing, Mobile for consumption

Interface Navigation

Report, Data, and Model views for comprehensive development workflow

Connecting to Data Sources

File Sources

Excel, CSV, JSON, XML files

Databases

SQL Server, PostgreSQL, MySQL, Oracle

Cloud Services

Azure, AWS, Google Cloud, Salesforce

Web Sources

APIs, web pages, online services

Connection Modes

Import

Data copied into Power BI for fast performance. Best for most scenarios.

DirectQuery

Query source directly for real-time data. Slower but always current.

Live Connection

Connect to Analysis Services or Power BI datasets.

Data Preparation & Modeling

01

Data Profiling

Assess quality, identify issues, understand distributions

02

Transform Data

Filter, split, merge, pivot, unpivot operations

03

Combine Queries

Append rows or merge columns from multiple sources

04

Create Relationships

Define connections between tables in data model

05

Optimize Model

Design efficient schemas for performance

Data Modeling Principles

Star Schema

Central fact table surrounded by dimension tables. Optimized for analytical queries with clear relationships and fast aggregations. Preferred design for most Power BI models.

Snowflake Schema

Normalized dimension tables with multiple levels. More complex but reduces data redundancy. Use when dimension tables are very large or shared across multiple fact tables.

Relationships & Hierarchies

Primary & Foreign Keys

Unique identifiers in dimension tables connect to fact table foreign keys

Cardinality

One-to-many relationships are most common in star schemas

Filter Direction

Single or bi-directional filtering affects how slicers propagate

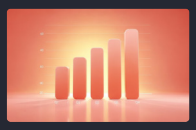
Hierarchies

Year > Quarter > Month > Day for drill-down analysis

Visual Reports & DAX Fundamentals

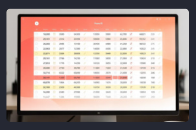
Visualization transforms data into insights. This module covers data visualization principles, Power BI's extensive visual library, and interactive elements that engage users. You'll also begin learning DAX (Data Analysis Expressions), the formula language that powers sophisticated calculations and business metrics in Power BI.

Core Visualizations



Charts

Bar, column, line, area, pie charts for standard comparisons and trends

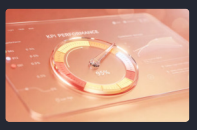


Tables & Matrices

Detailed data display with conditional formatting and drill-down

Maps

Geographic visualizations showing spatial patterns and distributions



KPIs & Cards

Single-value displays highlighting key metrics and performance indicators

Interactive Elements

Slicers

Interactive filters for user-driven data exploration

Bookmarks

Save report states for guided storytelling

Drill-through

Navigate to detailed pages from summary views

Introduction to DAX

DAX Fundamentals

- Calculated columns vs measures
- Aggregation functions: SUM, AVERAGE, COUNT
- Logical functions: IF, AND, OR, SWITCH
- Text and date/time functions

Essential DAX Functions

- CALCULATE - modify filter context
- FILTER - create filtered tables
- RELATED - access related table values
- Creating KPIs and business metrics

Advanced DAX & Enterprise Deployment

This module advances your DAX skills with time intelligence, context manipulation, and performance optimization. You'll also learn enterprise deployment strategies including publishing, sharing, data refresh configuration, and integration with Microsoft 365 tools—skills essential for delivering Power BI solutions in organizational settings.

Time Intelligence

Year-to-date, month-to-date, quarter-to-date calculations. Prior period comparisons and growth rates for temporal analysis.

Context Understanding

Filter context from slicers and visuals vs row context in calculated columns. Mastering context is key to correct DAX calculations.

Advanced Techniques

Variables for readability and performance. Iterator functions (SUMX, AVERAGEX) for row-by-row calculations. Optimization strategies.

Publishing & Sharing

Power BI Service

- Publishing reports from Desktop
- Workspace organization and permissions
- Dashboards vs reports distinction
- Creating Power BI apps for distribution

Data Refresh

- Scheduled refresh configuration
- On-premises data gateway setup
- Refresh failure troubleshooting
- Incremental refresh for large datasets

Integration & Governance

Microsoft 365 Integration

Embed in Teams, SharePoint, Excel, PowerPoint for seamless collaboration

Row-Level Security

Restrict data access based on user identity for data governance

Admin Portal

Tenant settings, capacity management, usage metrics monitoring

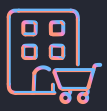
Deployment Pipelines

Development, test, production workflows for enterprise change management

Advanced Analytics & Enterprise Features

Power BI's advanced features unlock sophisticated analytical capabilities. This module covers custom visuals, AI-powered insights, R and Python integration, and enterprise deployment strategies. You'll learn to leverage cutting-edge analytics whilst understanding governance, security, and performance optimization for production environments.

Advanced Visualizations



Custom Visuals

Extend Power BI with community-created visuals from AppSource. Access specialized charts, advanced analytics, and industry-specific visualizations beyond built-in options.

Advanced Chart Types

Waterfall charts show cumulative effects, funnel charts display process stages, decomposition trees enable hierarchical exploration of drivers behind metrics.

R & Python Integration

Execute R and Python scripts within Power BI for statistical analysis, machine learning models, and custom visualizations beyond DAX capabilities.

AI Visuals

Key Influencers identify factors driving metrics, Q&A enables natural language queries, Smart Narratives generate automated insights from data patterns.

Enterprise Deployment

Security & Governance

Row-Level Security (RLS), Object-Level Security (OLS), sensitivity labels, audit logs

Performance Optimization

Aggregations, incremental refresh, query folding, DirectQuery optimization

Dataflows

Centralized data preparation, reusable transformations, self-service ETL

APIs & Embedding

REST APIs, embedded analytics, custom applications, automated deployment

SECTION 6

Foundations of Generative AI

Generative AI represents a paradigm shift in how we interact with technology. This module explores Large Language Models, transformer architecture, and the landscape of frontier models. You'll understand how these systems work, compare major providers, and learn strategic considerations for model selection and cost optimization in production environments.

LLM Fundamentals

Neural networks trained on vast text corpora

Cost Optimization

Balancing performance and expense

Model Selection

Matching capabilities to use cases

Transformer Architecture

Attention mechanisms enabling context understanding

Tokenization

Breaking text into processable units

Model Evolution

GPT-1 through 2026 frontier models

Major LLM Providers

GPT (OpenAI)

Industry-leading models with broad capabilities, extensive API ecosystem, and strong reasoning performance

Claude (Anthropic)

Emphasis on safety and helpfulness, excellent for nuanced tasks, strong context windows

Gemini (Google)

Multimodal capabilities, integration with Google ecosystem, competitive performance

DeepSeek

Cost-effective alternative with strong performance, open-source options, growing ecosystem

Prompt Engineering & RAG Systems

Advanced Prompting

Zero-shot, few-shot, and chain-of-thought techniques for optimal model performance

Context Engineering

Design effective context windows with relevant information and clear instructions

Reasoning Optimization

Leverage reasoning modes for complex problem-solving and multi-step tasks

Hallucination Reduction

Techniques to minimize false information and improve factual accuracy

Multimodal Prompting

Combine text, images, and audio for rich, context-aware interactions

Advanced RAG Techniques

Agentic RAG

Self-improving retrieval systems that learn from interactions and refine search strategies

MCP-Enhanced RAG

Model Context Protocol integration for standardized tool access and enhanced capabilities

Hybrid Search

Combine semantic similarity with keyword matching for comprehensive retrieval

Document Processing

Handle PDFs, images, tables at scale with intelligent chunking and metadata extraction

hello@digitaledify.ai

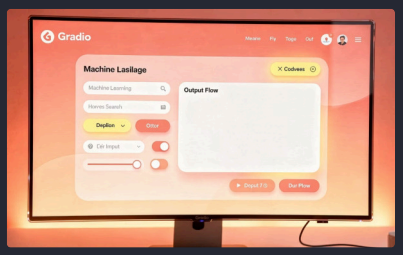
+91 6304982304

Digital Edify

Production Deployment & Agentic AI

Deployment Interfaces

There was an error generating this image



Streamlit

Rapid Python-based web apps for data science and ML models with minimal code

Gradio

Quick ML model interfaces with automatic API generation for demos and testing

Production Considerations

Cost Optimization

- Model selection based on task complexity
- Caching strategies for repeated queries
- Batch processing for efficiency
- Monitoring token usage and costs

Governance & Security

- EU AI Act compliance requirements
- API security and rate limiting
- Data privacy and encryption
- Audit trails and monitoring

Introduction to Agentic AI

Plan

Break down complex tasks into steps

Reason

Analyze information and make decisions

Observe

Monitor results and adjust strategy

Act

Execute actions using tools and APIs

LangGraph 1.0 & Advanced Workflows

LangGraph 1.0 Fundamentals

Graph Architecture

Nodes represent operations, edges define flow, enabling complex branching logic and parallel execution

State Management

Durable state persists across nodes, enabling long-running workflows and failure recovery

Node Caching

Development optimization caches node results, accelerating iteration during workflow design

Pre/Post Hooks

Inject validation, logging, and guardrails before and after node execution

Advanced Workflow Patterns

1

Parallel Execution

Deferred nodes run concurrently

2

Conditional Routing

Decision trees based on state

3

Iterative Refinement

Loops for quality improvement

4

Type-Safe Streaming

Real-time output delivery

Production Agentic Systems

1. Security & Compliance
2. Observability & Monitoring
3. Multi-Agent Coordination
4. LangGraph Platform Deployment

Persistence & Human-in-the-Loop

Durable State Management

PostgreSQL and Redis backends persist workflow state across sessions. Multi-day workflows maintain context, enabling complex processes that span extended timeframes with restart and failure recovery capabilities.

Human-in-the-Loop (HITL)

Critical decisions require human approval before proceeding. HITL implementations pause workflows, present context to reviewers, and resume based on human input—essential for compliance and high-stakes decisions.

Multi-Agent Systems

Agent Specialization

Design agents with specific capabilities and domains of expertise for complex tasks

A2A Protocol

Google's Agent-to-Agent Protocol enables standardized communication between agents

LangSmith Observability

Monitor agent behaviour, trace decisions, debug failures, and optimize performance

MCP Security

Model Context Protocol security model with prompt injection prevention and guardrails

> "The future of AI isn't just about smarter models—it's about systems that can autonomously plan, reason, and act whilst maintaining security, compliance, and human oversight. Production agentic systems represent the convergence of these capabilities."

hello@digitaledify.ai

+91 6304982304

Digital Edify