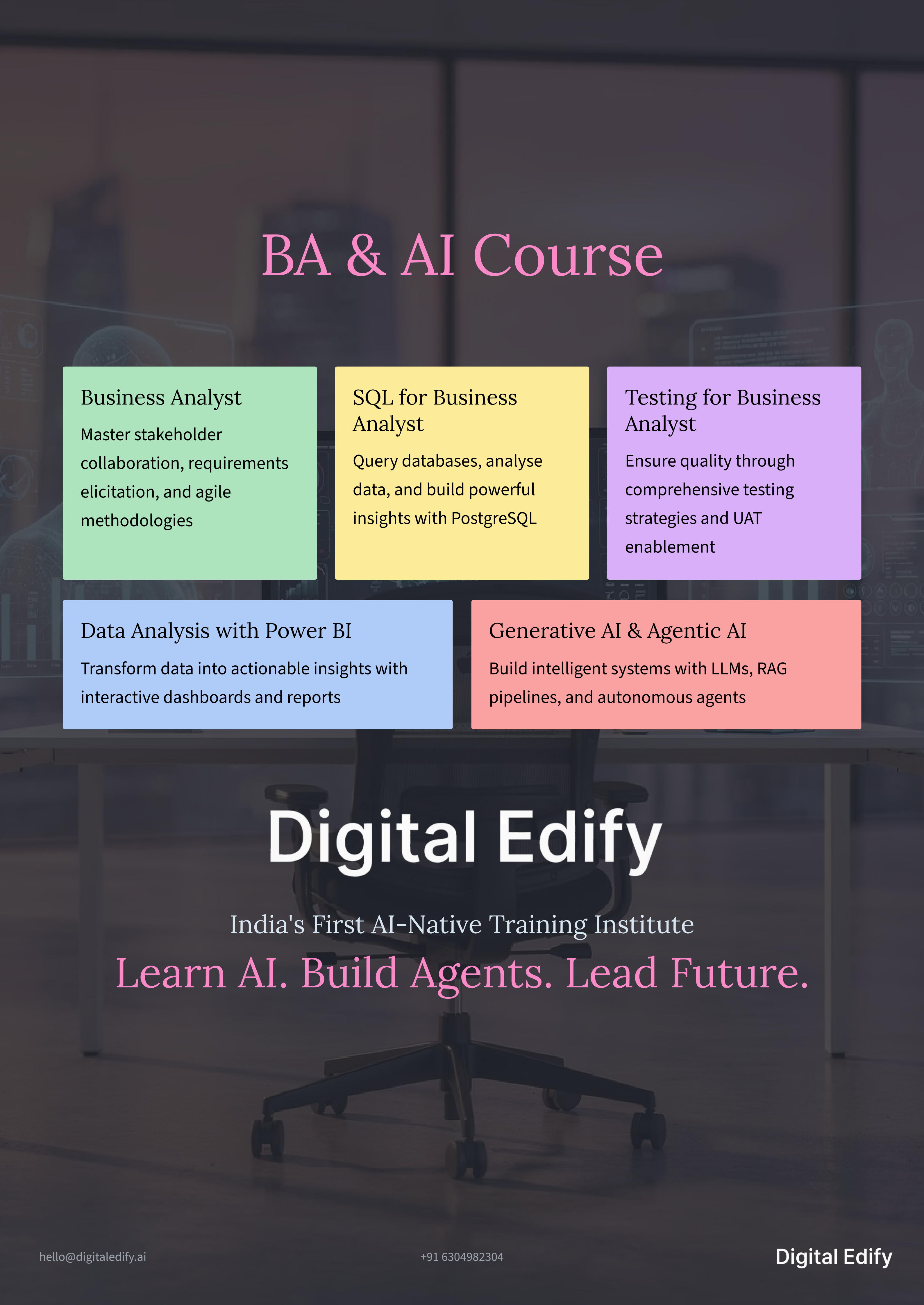


BA & AI Course

Business Analyst

Master stakeholder collaboration, requirements elicitation, and agile methodologies

SQL for Business Analyst

Query databases, analyse data, and build powerful insights with PostgreSQL

Testing for Business Analyst

Ensure quality through comprehensive testing strategies and UAT enablement

Data Analysis with Power BI

Transform data into actionable insights with interactive dashboards and reports

Generative AI & Agentic AI

Build intelligent systems with LLMs, RAG pipelines, and autonomous agents

Digital Edify

India's First AI-Native Training Institute

Learn AI. Build Agents. Lead Future.

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 how applications, Agile, cloud, and AI work together in modern digital systems.

Section 2: Foundations of Web (HTML, CSS, JS, TS)

Build responsive and interactive web interfaces using modern frontend technologies.

Section 3: Modern Frontend Framework – React JS

Create scalable, high-performance frontend apps with React and modern state management.

Section 4: Node.js & MongoDB for Backend Development

Develop secure, scalable backend services using Node.js, APIs, and MongoDB.

Section 5: Python for FullStack

Learn Python fundamentals for backend development, automation, and fullstack applications.

Section 6: SQL for AI & FullStack

Work with relational databases using SQL for data-driven and fullstack solutions.

Section 7: FullStack Python Framework – Django

Build end-to-end web applications using Django and REST APIs.

Section 8: Modern Python Framework – FastAPI

Develop fast and secure APIs using FastAPI and async programming.

Section 9: Generative AI & Agentic AI

Build intelligent AI solutions using LLMs, RAG, and agentic workflows.

SECTION 1

Application Life Cycle Management

Understanding Applications

Applications are software programmes designed to perform specific tasks. They range from simple desktop tools to complex web platforms serving millions of users globally.

Modern applications leverage diverse technologies across frontend, backend, and database layers to deliver seamless user experiences.

Web Technologies Stack

- **Frontend:** HTML, CSS, JavaScript, React for user interfaces
- **Backend:** Python, Java, Node.js for server-side logic
- **Databases:** SQL (MySQL, PostgreSQL) and NoSQL (MongoDB)

01

Planning

Define project scope, objectives, and feasibility

02

Analysis

Gather and document detailed requirements

03

Design

Create system architecture and technical specifications

04

Implementation

Develop and code the application

05

Testing

Validate functionality and quality

06

Deployment

Release to production environment

07

Maintenance

Ongoing support and enhancements

Agile & Scrum Framework

Waterfall vs. Agile

Traditional sequential approach versus iterative, flexible methodology

The Agile Mindset

Embrace change, collaboration, and continuous improvement

Popular Frameworks

Scrum, Kanban, XP, and hybrid 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

User Stories

Capture requirements from user perspective with acceptance criteria

Epics & Themes

Group related stories into larger initiatives

Estimation

Size stories using story points or t-shirt sizing

Backlog Management

Prioritise and refine using Google Sheets or Azure Boards

Computing & Data

Computing power forms the backbone of modern technology, enabling everything from simple calculations to complex artificial intelligence. Understanding the fundamental computing technologies and cloud service models is essential for navigating today's digital landscape.

CPU

Central Processing Unit handles general-purpose computing tasks and sequential operations

GPU

Graphics Processing Unit excels at parallel processing for AI and machine learning workloads

Cloud Service Models

IaaS

Infrastructure as a Service provides virtualised computing resources including servers, storage, and networking

PaaS

Platform as a Service offers development and deployment environments without infrastructure management

SaaS

Software as a Service delivers applications over the internet on a subscription basis

Introduction to AI, Generative AI & Agentic AI

What is Artificial Intelligence?

Artificial Intelligence enables machines to simulate human intelligence, learning from data, recognising patterns, and making decisions. AI systems process vast amounts of information to solve complex problems and automate tasks.

How AI Works

AI systems learn through algorithms that identify patterns in data. They improve performance over time through training, adjusting their models based on feedback and new information.

Machine Learning

Algorithms that learn from data without explicit programming, enabling predictive analytics and pattern recognition

Deep Learning

Neural networks with multiple layers that process complex data like images, speech, and natural language

Generative AI

Creates new content including text, images, and code using large language models and diffusion models

Large Language Models

Transformer-based models like GPT and Claude that understand and generate human-like text across diverse tasks

Image Generation

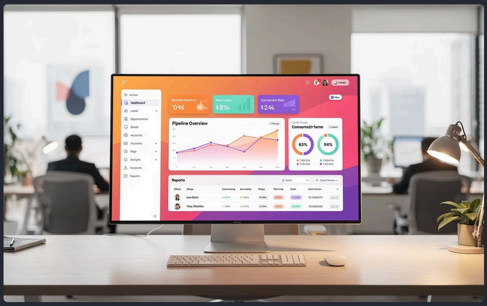
Models like DALL-E and Midjourney create stunning visuals from text descriptions

AI in Everyday Learning

From personalised education to intelligent tutoring systems transforming how we acquire knowledge

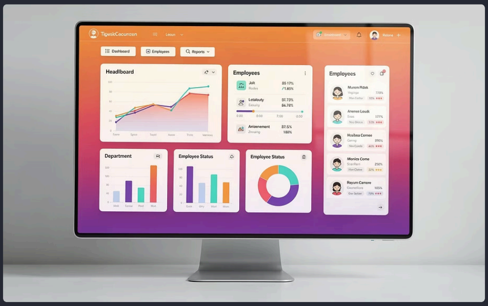
Real-World Applications

Modern applications transform how organisations operate across industries. From managing customer relationships to streamlining healthcare delivery, these systems drive efficiency, enhance decision-making, and improve user experiences at scale.



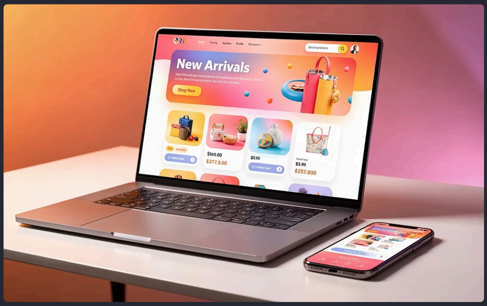
Customer Relationship Management

CRM systems centralise customer data, track interactions, and automate sales processes to build stronger relationships and drive revenue growth



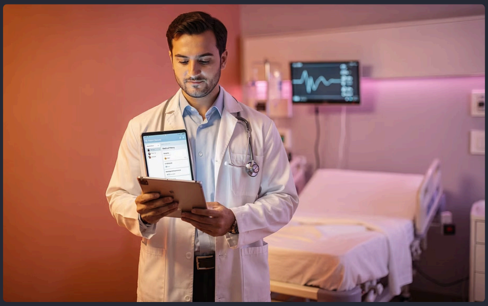
Human Resource Management Systems

HRMS platforms streamline recruitment, payroll, performance management, and employee engagement across the organisation



Retail & E-Commerce

Digital commerce platforms enable seamless shopping experiences with inventory management, payment processing, and personalised recommendations



Healthcare Applications

Electronic health records, telemedicine platforms, and diagnostic tools improve patient care and operational efficiency

SECTION 2

Foundations of Business Analysis in a Digital World

The modern Business Analyst operates at the intersection of technology, business strategy, and stakeholder collaboration. This role demands mastery of agile methodologies, requirement elicitation techniques, and emerging AI tools that enhance productivity and decision-making.

Modern BA Role

Bridge business needs with technical solutions through analysis and facilitation

Agile Mindset

Embrace iterative development and continuous stakeholder collaboration

ALM

Understand application lifecycle from conception to retirement

Scrum Framework

Master roles, events, and artifacts for effective sprint delivery

BA Tools

Leverage Sheets, Azure Boards, and GenAI for requirement management

User Stories

Capture requirements with epics, themes, and acceptance criteria

Stakeholder Collaboration & Agile Requirement Discovery

Effective stakeholder collaboration forms the foundation of successful projects. Business Analysts must identify key stakeholders, understand their needs, and facilitate productive conversations that uncover true requirements whilst managing diverse interests and expectations.

Discovery Phase

Explore problem space and identify opportunities through structured investigation

Stakeholder Analysis

Map stakeholders using Power/Interest Grid and Influence/Impact Matrix

User Personas

Create detailed profiles representing key user segments and their goals

Empathy Mapping

Understand stakeholder perspectives by capturing what they think, feel, say, and do. This technique reveals unspoken needs and pain points that drive better solutions.

RACI Matrix

Define clear accountability with Responsible, Accountable, Consulted, and Informed roles for each deliverable and decision point.

Agile Interviews

- Discovery interviews to understand context
- Contextual inquiry observing users in their environment
- Stand-up conversations for quick clarifications

Workshops & Facilitation

Lead collaborative sessions that generate consensus, prioritise features, and validate requirements with cross-functional teams.

GenAI for Stakeholder Management: Leverage AI tools to analyse stakeholder feedback, generate interview questions, and synthesise insights from multiple sources.

Eliciting & Prioritising Requirements

MoSCoW Method

Categorise requirements as Must have, Should have, Could have, or Won't have this time to establish clear priorities

RICE Scoring

Evaluate features using Reach, Impact, Confidence, and Effort to make data-driven prioritisation decisions

WSJF

Weighted Shortest Job First balances business value, time criticality, and risk reduction against job size

Requirements Management Excellence

Traceability

Link requirements to business objectives, test cases, and deliverables ensuring complete coverage and impact analysis.

Versioning

Track requirement changes over time with clear version control and change history documentation.

Validation

Confirm requirements are complete, consistent, testable, and aligned with stakeholder expectations.

Alignment with OKRs

Ensure every requirement contributes to measurable objectives and key results that drive business outcomes.

GenAI for Requirements: Use AI to analyse requirement patterns, identify gaps, suggest acceptance criteria, and generate traceability matrices automatically.

Process Analysis & Value Stream Mapping

01

BPMN Modelling

Create standardised process diagrams using Business Process Modeling Notation

02

As-Is Mapping

Document current state processes to establish baseline understanding

03

SIPOC Analysis

Define Suppliers, Inputs, Process, Outputs, and Customers for clarity

04

Value Stream Mapping

Visualise material and information flow to identify waste and bottlenecks

05

To-Be Design

Create optimised future state with automation and efficiency improvements

Customer Journey Mapping

Visualise end-to-end customer experience across touchpoints, identifying pain points and opportunities for improvement at each stage of interaction.

Automation Opportunities

Identify repetitive, rule-based tasks suitable for automation using RPA, workflow engines, or AI-powered solutions.

Hands-On Practice: Create As-Is BPMN diagrams, design To-Be processes with automation, conduct Value Stream Mapping workshops, and identify five automation opportunities in real scenarios.

Crafting Agile Business & Functional Requirements

BRD

Business Requirements Document captures high-level business objectives and success criteria

FDD

Functional Design Document details system behaviour, features, and technical specifications

Integration

Define API requirements and system integration scenarios

NFRs

Specify non-functional requirements for performance, security, and scalability

Documentation Excellence

User Story Mapping

Transform user stories into detailed functional requirements with clear acceptance criteria and edge cases.

Traceability Matrix

Link requirements to objectives, test cases, and deliverables ensuring complete coverage and impact analysis.

Data Modelling

Design entity relationships, schema structures, and data flow diagrams that support functional needs.

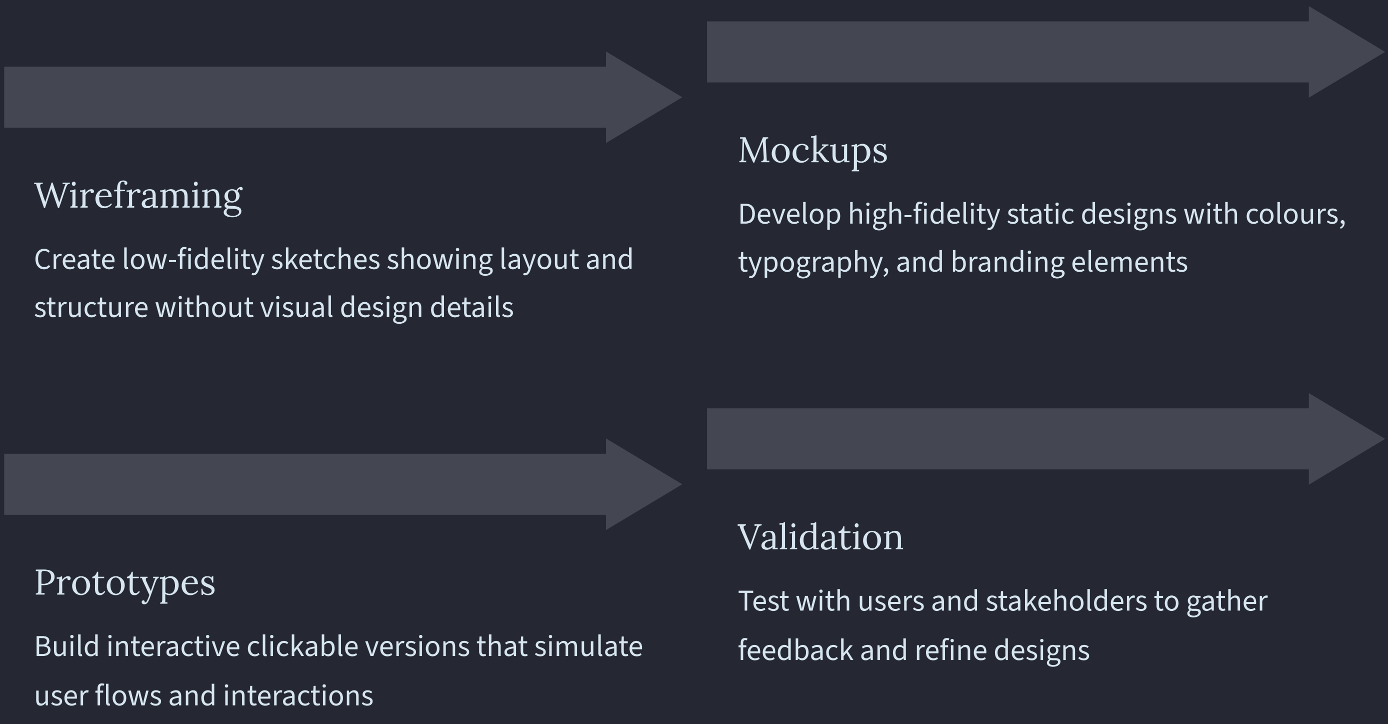
Review & Sign-Off

Facilitate stakeholder reviews and obtain formal approval before development begins.

GenAI for Documentation: Accelerate requirement writing with AI-generated templates, acceptance criteria suggestions, and automated traceability matrix creation.

Prototyping & Solution Design

Visual prototypes transform abstract requirements into tangible designs that stakeholders can evaluate and refine. Business Analysts leverage prototyping tools to validate concepts early, reduce development risk, and ensure solutions meet user needs before significant investment.



Wireframing

Create low-fidelity sketches showing layout and structure without visual design details

Mockups

Develop high-fidelity static designs with colours, typography, and branding elements

Prototypes

Build interactive clickable versions that simulate user flows and interactions

Validation

Test with users and stakeholders to gather feedback and refine designs

Design Thinking for BAs

Apply empathise, define, ideate, prototype, and test phases to solve problems creatively whilst maintaining user focus throughout the design process.

User-Centred Design

Prioritise user needs, behaviours, and preferences in every design decision to create intuitive, accessible solutions.

Prototyping Tools

- Balsamiq for rapid wireframing
- Figma for collaborative design
- Interactive prototypes with clickable hotspots

Usability Testing

Observe users interacting with prototypes to identify pain points and validate design decisions before development.

GenAI for UI/UX: Generate wireframe suggestions, create design variations, and produce user flow diagrams automatically based on requirements.

Agile Testing Strategy & UAT Enablement

Business Analysts play a crucial role in ensuring quality by bridging requirements and testing. They translate acceptance criteria into test scenarios, facilitate User Acceptance Testing, and validate that delivered solutions meet business expectations and user needs.

Test Scenarios

Transform requirements into comprehensive test scenarios covering happy paths and edge cases

UAT Planning

Coordinate user acceptance testing with stakeholders, environments, and test data

UAT Scripts

Create detailed step-by-step testing instructions with expected results

Quality Assurance Activities

Facilitating UAT Sessions

Guide business users through testing, answer questions, and document feedback in real-time during structured testing sessions.

Acceptance Criteria Validation

Verify that delivered functionality meets documented acceptance criteria and business expectations before sign-off.

Defect Management

Log, prioritise, and track defects through resolution. Participate in triage meetings to assess severity and impact.

Testing Documentation

Maintain test plans, scripts, results, and traceability matrices linking tests to requirements.

GenAI for Testing: Automatically generate test cases from requirements, create test data sets, and produce comprehensive testing documentation.

Training, Communication & Change Management

Successful implementations require more than technical delivery. Business Analysts must prepare organisations for change through strategic communication, comprehensive training, and proactive resistance management that ensures smooth adoption and sustained value realisation.

Change Management

Apply structured approaches to transition individuals and teams to new ways of working

Communication Planning

Define stakeholder communication cadence, channels, and messaging strategies

Resistance Management

Identify concerns, address objections, and build champions

Training Materials

Create user guides, quick reference cards, and video tutorials

User Adoption

Monitor usage, gather feedback, and provide ongoing support

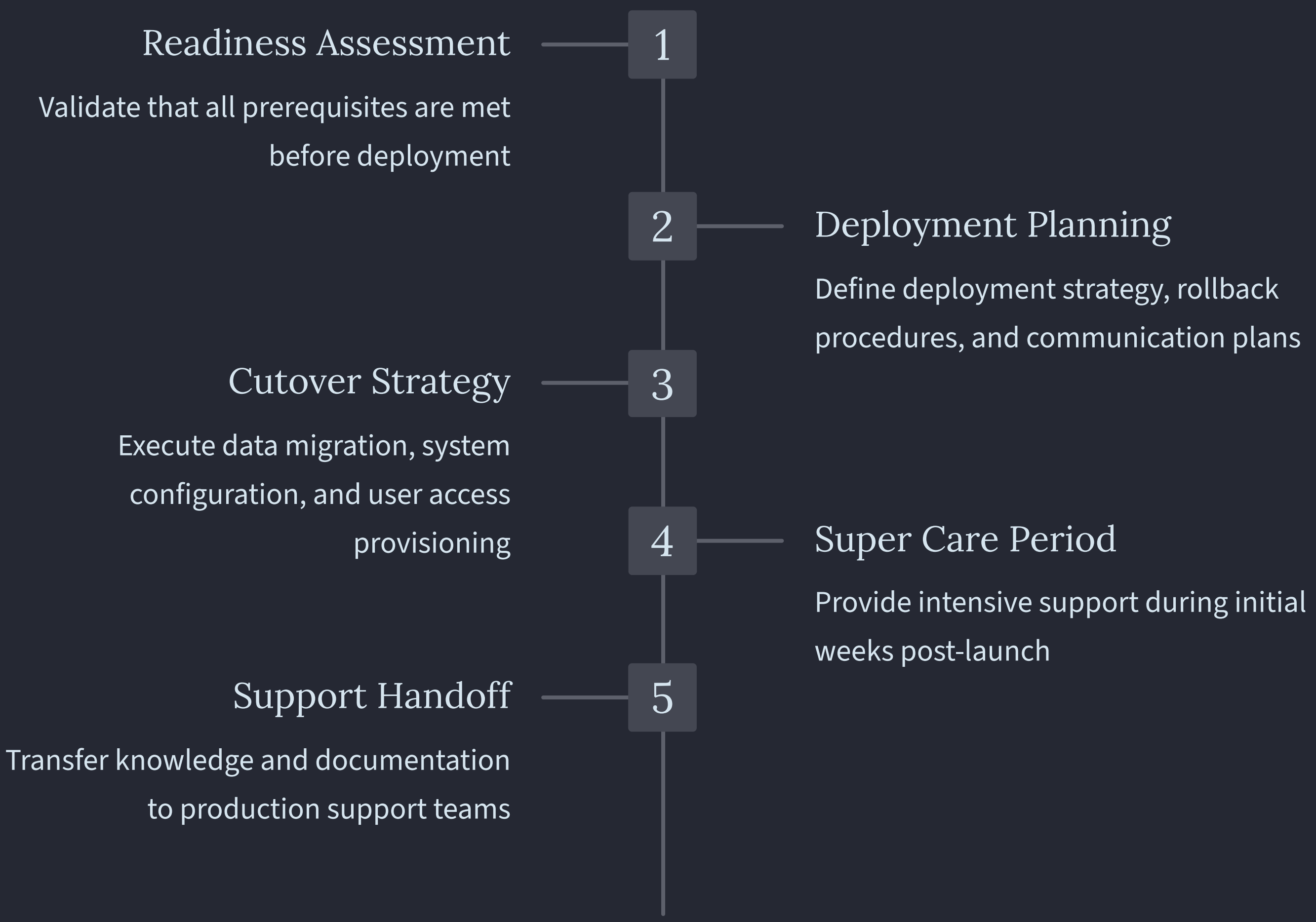
Training Sessions

Deliver engaging workshops and hands-on training for different user groups

GenAI for Training: Generate training content, create interactive tutorials, produce FAQs, and develop role-specific learning paths automatically.

Go-Live Planning, Super Care & Production Support

The transition to production requires meticulous planning and intensive support. Business Analysts coordinate go-live activities, provide super care during the critical stabilisation period, and ensure smooth handoff to ongoing support teams.



1 Readiness Assessment

Validate that all prerequisites are met before deployment

2 Deployment Planning

Define deployment strategy, rollback procedures, and communication plans

3 Cutover Strategy

Execute data migration, system configuration, and user access provisioning

4 Super Care Period

Provide intensive support during initial weeks post-launch

5 Support Handoff

Transfer knowledge and documentation to production support teams

Monitoring & Issue Resolution

Track system performance, user feedback, and incident reports. Triage issues, coordinate fixes, and communicate status to stakeholders throughout stabilisation.

Post-Implementation Review

Conduct retrospectives to capture lessons learned, document successes and challenges, and identify opportunities for process improvement.

Continuous Improvement & Data-Driven Insights

Business Analysts drive ongoing value by measuring outcomes, analysing data, and identifying optimisation opportunities. They define KPIs, create dashboards, and leverage analytics tools to provide insights that inform strategic decisions and continuous improvement initiatives.

85%

User Adoption

Track active users and feature utilisation rates

40%

Efficiency Gain

Measure process time reduction and productivity improvements

£2.5M

Business Value

Quantify cost savings and revenue impact

Data Analysis Capabilities

KPI Definition & Tracking

Establish measurable indicators aligned with business objectives. Monitor trends and identify anomalies requiring attention.

SQL for Requirements

Query databases to validate data availability, understand data structures, and inform requirement decisions.

Power BI & Tableau

Create interactive dashboards and reports that visualise performance metrics and support data-driven decision-making.

Process Optimisation

Analyse performance data to identify bottlenecks, inefficiencies, and opportunities for automation or improvement.

Career Growth: Specialise in domains like data analytics, product management, enterprise architecture, or advance into leadership roles as Senior BA, Lead BA, or BA Manager.

SECTION 3

Software Testing Fundamentals

Software testing ensures applications function correctly, meet requirements, and deliver quality user experiences. Understanding testing fundamentals prevents costly failures and builds confidence in software reliability across desktop, web, and mobile platforms.

What is Software Testing?

Systematic evaluation of software to identify defects, verify functionality, and validate that requirements are met before release

Why Testing Matters

Prevents catastrophic failures, protects users, reduces costs, and maintains organisational reputation through quality assurance

Real-World Testing Failures

Therac-25

Radiation therapy machine software defects caused patient deaths due to inadequate testing and safety mechanisms.

Mars Climate Orbiter

Unit conversion error between metric and imperial systems destroyed £125 million spacecraft.

Boeing 737 MAX

Software issues in flight control system contributed to fatal crashes, grounding entire fleet.

Quality

Meeting requirements and user expectations

QA vs QC

Quality Assurance (process) vs Quality Control (product)

Errors & Defects

Mistakes, bugs, defects, and failures defined

STLC

Software Testing Life Cycle phases

SDLC Models & Testing Approaches

Different software development models require tailored testing strategies. Understanding Waterfall, V-Model, and testing approaches ensures quality is built into every phase of development through appropriate verification and validation activities.

Waterfall Model

Sequential phases with testing after development completes. Clear milestones but limited flexibility for changes.

V-Model

Verification and validation model where testing activities correspond to each development phase.

V-Model Components

Documentation

- **BRS:** Business Requirements Specification
- **SRS:** Software Requirements Specification
- **HLD:** High-Level Design
- **LLD:** Low-Level Design

Static Testing

- **Review:** Informal examination of documents
- **Walkthrough:** Author-led presentation
- **Inspection:** Formal defect detection process

Testing Approaches

White Box Testing

Internal structure and code testing with knowledge of implementation details

Black Box Testing

Functional testing without knowledge of internal code structure

Grey Box Testing

Combination approach with partial knowledge of internal workings

Testing Levels & Functional Testing

Unit Testing

Test individual components using frameworks like JUnit, NUnit, pytest

Integration Testing

Verify component interactions using incremental or big bang approaches

System Testing

Validate complete system functionality and non-functional requirements

UAT

Business users validate system meets requirements through alpha and beta testing

Functional Testing Deep Dive

GUI Testing

- Visual design verification
- Functional elements testing
- Content and interaction validation

Database Testing

- CRUD operations validation
- Data integrity testing
- SQL basics for testers

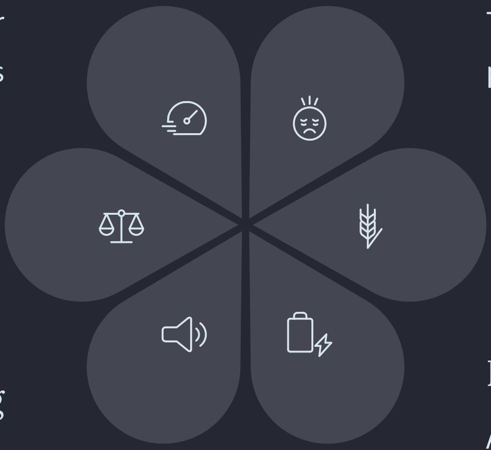
Additional Testing

- Object properties testing
- Error handling validation
- Calculations and computations
- Links testing (internal, external, anchor, email)
- Cookie and session testing

Usability Testing: Assess user-friendliness, measure satisfaction metrics, and ensure intuitive navigation and interaction patterns.

Non-Functional Testing & Testing Types

Non-functional testing validates system qualities beyond features, including performance, security, and compatibility. Understanding various testing types ensures comprehensive coverage and appropriate test selection for different scenarios and risk profiles.



Load Testing

Verify system behaviour under expected user loads

Scalability Testing

Verify system can scale to meet demand

Volume Testing

Test with large data volumes

Stress Testing

Test system limits and breaking points

Spike Testing

Validate response to sudden traffic increases

Endurance Testing

Assess stability over extended periods

Additional Testing Types

Security Testing

Validate authentication, authorisation, and encryption mechanisms protect against vulnerabilities and unauthorised access.

Compatibility Testing

Verify functionality across different hardware, operating systems, and browsers.

Recovery Testing

Assess system's ability to recover from crashes, hardware failures, and other disasters.

Installation Testing

Test fresh installation, upgrades, uninstall, and reinstall scenarios.

- **Regression Testing**

  Verify existing functionality after changes (unit, regional, full)

- **Smoke & Sanity Testing**

  Build verification and focused testing of specific functionality

- **Exploratory Testing**

  Unscripted testing to discover unexpected issues

Test Design Techniques & Documentation

Equivalence Class Partitioning

Divide inputs into valid and invalid classes, testing one value from each partition

Boundary Value Analysis

Test values at boundaries and just beyond to catch edge case defects

Decision Table Testing

Map conditions, actions, and rules for complex business logic scenarios

State Transition Testing

Model states, events, and transitions using diagrams and tables

Error Guessing

Leverage experience to predict likely defect areas

Test Documentation

Test Plan

Define scope, approach, resources, schedule, and deliverables for testing activities.

Use Cases & Scenarios

Document actor interactions, actions, and goals that inform test case creation.

Test Cases

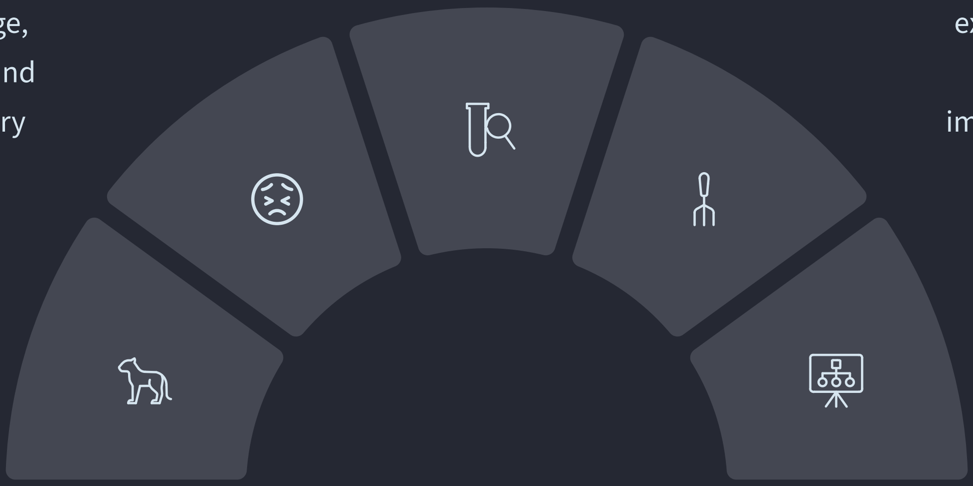
Detailed steps, test data, expected results, and traceability to requirements.

RTM & Defect Tracking

Requirements Traceability Matrix and bug life cycle management.

Agile Testing & Tools

Agile methodologies transform testing from a phase to a continuous activity integrated throughout development. Testers collaborate closely with teams, automate where possible, and leverage tools like JIRA to manage test cases, track defects, and maintain quality in fast-paced sprints.



Agile Principles

Embrace change, collaboration, and iterative delivery

Scrum Framework

Roles, artifacts, and events structure testing activities

Testing in Agile

Continuous testing throughout sprint lifecycle

JIRA & GitHub

Test management and version control tools

Real Projects

Hands-on experience with actual implementations

Scrum Framework Deep Dive

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
- Burndown Charts

JIRA for Testing: Create test cases, track bugs, manage sprints, and generate reports. Integrate with version control systems like GitHub for complete traceability.

SECTION 4

Introduction to Business Intelligence & Power BI

Business Intelligence transforms raw data into actionable insights that drive strategic decisions. Power BI provides a comprehensive platform for connecting to data sources, creating interactive visualisations, and sharing insights across organisations through intuitive dashboards and reports.

BI Fundamentals

Modern analytics approaches that turn data into competitive advantage

Power BI Components

Desktop, Service, Mobile, and Gateway architecture

Interface Navigation

Master the workspace and create your first report

Connecting to Data Sources

Data Source Types

- File sources (Excel, CSV, JSON)
- Database connections (SQL Server, Oracle)
- Cloud services (Azure, AWS, Google)
- Web sources and APIs

Connection Modes

- **Import:** Data copied into Power BI
- **DirectQuery:** Live connection to source
- **Live Connection:** Analysis Services models

Performance Considerations: Import mode offers best performance for most scenarios. Use DirectQuery for real-time data requirements or when data volumes exceed Power BI limits.

Data Preparation & Modeling

01

Power Query

Transform and clean data with intuitive interface

02

Data Profiling

Assess quality and identify issues early

03

Transformations

Filter, split, merge, and reshape data

04

Combining Queries

Append and merge operations

05

Data Modeling

Create relationships and hierarchies

Data Modeling Best Practices

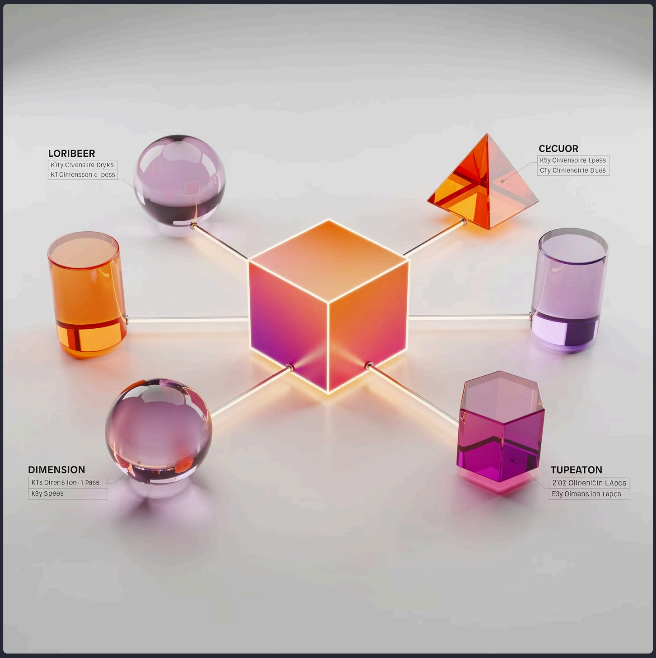
Schema Design

Star Schema: Fact tables surrounded by dimension tables for optimal query performance.

Snowflake Schema: Normalised dimensions for complex hierarchies.

Relationships

Define one-to-many relationships between fact and dimension tables using primary and foreign keys.



Hierarchies

Create drill-down paths like Year > Quarter > Month > Day for intuitive navigation

Date Dimensions

Build comprehensive date tables with fiscal periods, holidays, and time intelligence

Optimisation

Remove unnecessary columns, use appropriate data types, and minimise cardinality

Building Visual Reports & Dashboards

Effective visualisations communicate insights instantly and drive action. Power BI offers extensive chart types, interactive elements, and design capabilities that transform data into compelling stories. Mastering visualisation principles ensures reports engage audiences and support decision-making.

Core Visualisations

Bar charts, line graphs, pie charts, tables, matrices, maps, and KPI cards for diverse data representation

Interactive Elements

Slicers, filters, bookmarks, and drill-through enable users to explore data dynamically

Mobile Optimisation

Design responsive layouts that adapt to different screen sizes and devices

Data Visualisation Principles

Chart Selection

- Bar charts for comparisons
- Line charts for trends over time
- Pie charts for composition (use sparingly)
- Scatter plots for correlations
- Maps for geographical data

Dashboard Design

- Clear visual hierarchy
- Consistent colour schemes
- Appropriate white space
- Logical layout flow
- Actionable insights highlighted

Storytelling with Data: Guide viewers through insights with intentional design. Use colour to highlight key findings, arrange visuals to support narrative flow, and provide context through titles and annotations.

DAX & Advanced Analytics

DAX Fundamentals

Syntax, calculated columns vs measures, and basic functions

Essential Functions

Aggregation, logical, text, and date/time operations

CALCULATE & FILTER

Modify filter context for sophisticated calculations

Time Intelligence

YTD, MTD, QTD, and period comparisons

Advanced Patterns

Variables, iterators, and performance optimisation

Advanced Visualisations & Enterprise Deployment

Power BI extends beyond standard charts with custom visuals, AI capabilities, and enterprise features. Publishing to the cloud enables collaboration, whilst governance ensures security and compliance across the organisation.

Custom Visuals

AppSource marketplace offers specialised charts like waterfall, funnel, and decomposition trees

R & Python Integration

Leverage statistical computing and advanced analytics within Power BI

AI Visuals

Key Influencers, Q&A, and Smart Narratives provide automated insights

Enterprise Deployment

Publishing & Sharing

- Workspace management and collaboration
- Dashboards vs reports
- Power BI apps for distribution
- Integration with Teams, SharePoint, Excel

Data Refresh

- Gateway configuration
- Scheduled refresh
- Incremental refresh for large datasets

Governance & Security

- Row-Level Security (RLS)
- Object-Level Security (OLS)
- Admin portal and tenant settings
- Compliance and audit trails

Performance Optimisation

- Aggregations for large datasets
- Dataflows for reusable ETL
- Deployment pipelines
- Capacity management

Enterprise Features: APIs enable embedded analytics. Premium capacity supports paginated reports, AI features, and larger datasets. Deployment pipelines streamline development, test, and production workflows.

SECTION 5

Foundations of Databases & PostgreSQL

Databases form the backbone of modern applications, storing and managing vast amounts of structured information. PostgreSQL, a powerful open-source relational database, provides robust features for data integrity, complex queries, and enterprise-scale operations.

RDBMS Fundamentals

Relational databases organise data in tables with defined relationships and enforce integrity constraints

ACID Properties

Atomicity, Consistency, Isolation, Durability guarantee reliable transactions

PostgreSQL

Advanced open-source database with extensive features and strong community support

Database Objects & Data Types

Core Objects

- Databases and schemas
- Tables and columns
- Indexes for performance
- Views for abstraction

Constraints

- PRIMARY KEY for unique identification
- FOREIGN KEY for referential integrity
- UNIQUE, NOT NULL, CHECK, DEFAULT

Data Types

- **Numeric:** INTEGER, DECIMAL, FLOAT
- **Character:** VARCHAR, TEXT
- **Date/Time:** DATE, TIMESTAMP
- **Boolean:** TRUE/FALSE
- **Special:** JSON, ARRAY, UUID

PostgreSQL Tools: Use psql command-line interface for quick operations or pgAdmin 4 for visual database management, query building, and administration tasks.

Querying and Analysing Data

SELECT Basics

Retrieve columns with aliases and expressions

Filtering

WHERE clause with comparison and logical operators

Sorting

ORDER BY for ascending or descending results

Aggregation

COUNT, SUM, AVG, MIN, MAX with GROUP BY

Joins

Combine tables using INNER, LEFT, RIGHT, FULL

Essential SQL Operations

Filtering Operators

- Comparison: =, !=, >, <, >=, <=
- Logical: AND, OR, NOT
- BETWEEN for ranges
- IN for multiple values
- LIKE for pattern matching
- IS NULL for missing data

Advanced Features

- DISTINCT removes duplicates
- LIMIT and OFFSET for pagination
- String functions (UPPER, LOWER, CONCAT)
- Date functions (EXTRACT, DATE_TRUNC)
- Window functions (ROW_NUMBER, RANK, LAG)

Advanced Queries & Data Manipulation

Advanced SQL techniques enable sophisticated data analysis and manipulation. Subqueries, CTEs, and set operators provide powerful tools for complex queries, whilst transaction management ensures data integrity during updates and deletions.

Subqueries

Nested queries in WHERE, SELECT, or FROM clauses for complex filtering and calculations

CTEs

Common Table Expressions create readable, reusable query components including recursive patterns

Set Operators

UNION, INTERSECT, EXCEPT combine results from multiple queries

Data Manipulation Operations

UPDATE Statements

Modify existing records with expressions and joins. Use WHERE clauses carefully to avoid unintended updates.

```
UPDATE products
SET price = price * 1.1
WHERE category = 'Electronics'
AND stock > 0;
```

DELETE Operations

Remove records with subqueries. TRUNCATE provides faster alternative for entire tables.

Transaction Management

BEGIN, COMMIT, and ROLLBACK ensure data consistency. Savepoints enable partial rollbacks within transactions.

```
BEGIN;
UPDATE accounts
SET balance = balance - 100
WHERE id = 1;
UPDATE accounts
SET balance = balance + 100
WHERE id = 2;
COMMIT;
```

Concurrency Control: Transaction isolation levels (READ COMMITTED, REPEATABLE READ, SERIALIZABLE) balance consistency with performance. PostgreSQL uses MVCC for efficient concurrent access.

Database Programming & Automation

PostgreSQL's programming capabilities enable automation, complex logic, and reusable code. Stored functions, procedures, and triggers transform databases from passive storage into active systems that enforce business rules and automate workflows.

Stored Functions

Reusable PL/pgSQL code with parameters, return types, and control structures

Stored Procedures

Transaction control and complex operations without return values

Triggers

Automatic execution on INSERT, UPDATE, DELETE for validation and audit logging

Views & Materialized Views

Standard Views

Virtual tables that simplify complex queries and provide abstraction layers. Updatable views allow modifications to underlying tables.

Materialized Views

Physically stored query results for performance. Refresh periodically to update data whilst maintaining fast query response.

Indexes

B-tree, Hash, GIN, and GiST indexes accelerate queries. Create indexes on frequently filtered or joined columns.

When to Index

- Foreign key columns
- WHERE clause columns
- JOIN conditions
- ORDER BY columns

PL/pgSQL Features: Control structures (IF, CASE, LOOP), exception handling, dynamic SQL, and functions returning tables enable sophisticated database logic.

Database Design & Optimisation

ER Modeling

Define entities, attributes, and relationships with cardinality

Normalisation

Apply 1NF, 2NF, 3NF to eliminate redundancy and anomalies

Design Best Practices

Naming conventions, data type selection, key strategies

Query Optimisation

EXPLAIN plans, index strategies, query rewriting

Performance Tuning

Query Analysis

Use EXPLAIN and EXPLAIN ANALYZE to understand execution plans. Identify sequential scans, nested loops, and missing indexes.

Index Strategies

Create appropriate indexes but avoid over-indexing. Composite indexes for multi-column queries. Partial indexes for filtered data.

Database Maintenance

- VACUUM removes dead tuples
- ANALYZE updates statistics
- Connection pooling reduces overhead
- Table partitioning for large datasets

Normalisation Trade-offs: Whilst normalisation reduces redundancy, denormalisation may improve read performance for specific use cases. Balance data integrity with query efficiency based on access patterns.

SECTION 6

Foundations of Generative AI

Large Language Models represent a breakthrough in artificial intelligence, enabling machines to understand and generate human-like text. Understanding LLM fundamentals, architecture, and model selection empowers Business Analysts to leverage AI effectively whilst managing costs and optimising performance.

LLM Fundamentals

Neural networks trained on vast text corpora to predict and generate language

Model Selection

Balance capability, cost, and latency for use cases

Tokenization

Text broken into tokens for processing and cost calculation

Transformer Architecture

Attention mechanisms enable understanding of context and relationships

Major LLMs

GPT, Claude, Gemini, DeepSeek each with unique strengths

Evolution

From GPT-1 to 2026 frontier models with expanding capabilities

Cost Optimisation Strategies

Token Management

- Minimise prompt length
- Cache common responses
- Use smaller models when appropriate
- Batch requests efficiently

Model Selection

- GPT-4 for complex reasoning
- GPT-3.5 for simpler tasks
- Claude for long context
- DeepSeek for cost efficiency

Prompt Engineering & RAG Systems

Advanced Prompting

Zero-shot, few-shot, and chain-of-thought techniques guide model behaviour and improve output quality

Context Engineering

Structure prompts with clear instructions, examples, and constraints to optimise reasoning

Hallucination Reduction

Ground responses in facts, request citations, and validate outputs against source material

Multimodal Prompting

Combine text, images, and audio inputs for richer AI interactions

RAG Architecture

Vector Databases

ChromaDB, Pinecone, and Qdrant store embeddings for semantic search. Retrieve relevant context before generation.

Embedding Strategies

Choose appropriate embedding models. Chunk documents effectively. Balance chunk size with retrieval accuracy.

01

Document Processing

Ingest and chunk documents at scale

02

Embedding

Convert text to vector representations

03

Retrieval

Semantic and keyword hybrid search

04

Generation

LLM creates response using retrieved context

Production Deployment & Agentic AI

Deploying AI systems to production requires robust infrastructure, security, and governance. Agentic AI extends capabilities beyond simple generation, enabling autonomous systems that plan, reason, and act to accomplish complex tasks.

User Interfaces

Streamlit and Gradio create interactive AI applications rapidly

LangGraph Platform

Deploy production-grade agentic systems with monitoring

Security & Governance

API security, rate limiting, and EU AI Act compliance

Introduction to Agentic AI

Agentic Fundamentals

Agents plan multi-step workflows, reason about problems, and take actions using tools. They operate autonomously to achieve goals.

LangChain 1.0 Agents

Create_agent abstraction with middleware enables customisation. Integrate multiple LLM providers seamlessly.

Model Context Protocol

MCP standardises tool integration, enabling agents to access databases, APIs, and external systems securely.

Enterprise Adoption

Agentic AI transforms customer service, data analysis, content creation, and workflow automation at scale.

Cost Optimisation: Monitor token usage, cache responses, batch requests, and select appropriate models for each task. Implement rate limiting and budget controls.

LangGraph 1.0 & Advanced Workflows

LangGraph 1.0 provides sophisticated orchestration for agentic AI systems. Its graph-based architecture enables complex workflows with parallel execution, conditional routing, and iterative refinement whilst maintaining state across multi-step processes.

Graph Architecture

Nodes represent actions, edges define flow, state persists across execution

Node Caching

Speed development by caching node outputs during iteration

Pre/Post Hooks

Implement guardrails, validation, and logging at execution boundaries

Advanced Workflow Patterns

1

Parallel Execution

Deferred nodes run concurrently for improved performance

2

Conditional Routing

Decision trees direct flow based on intermediate results

3

Iterative Refinement

Loops enable quality improvement through multiple passes

4

Type-Safe Streaming

Real-time output with guaranteed type safety

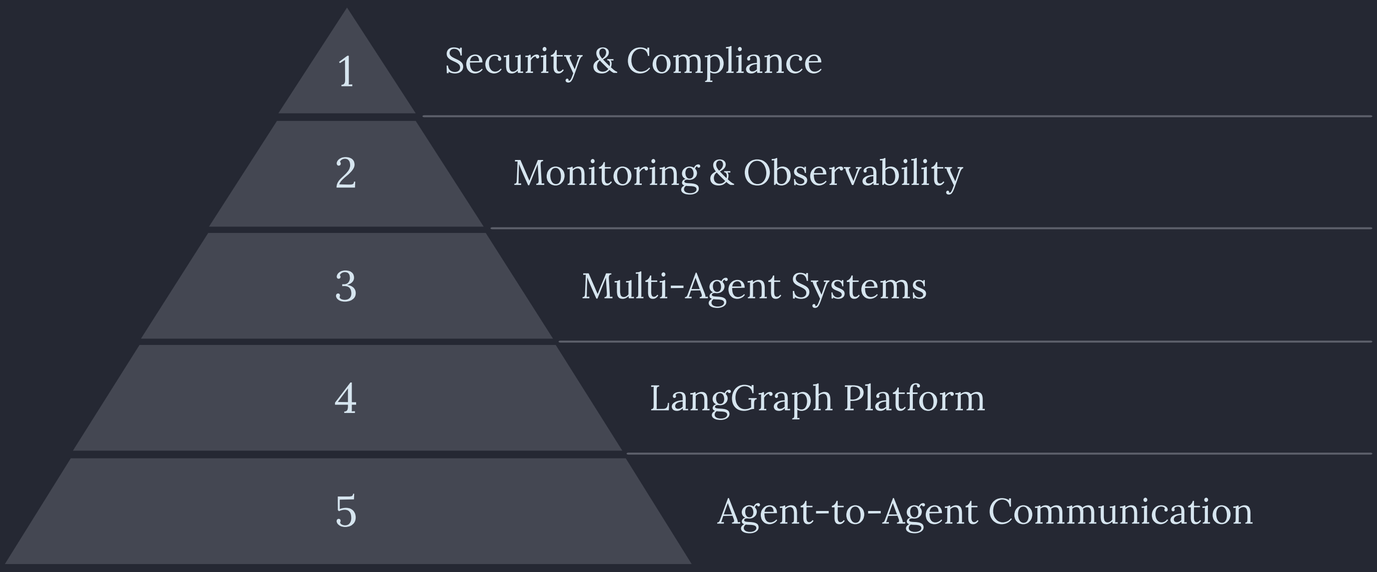
Use Case Examples

- Essay evaluation with rubric scoring
- Customer feedback routing by sentiment
- Multi-stage approval workflows
- Quality-gated content generation

Production Features

- Durable state management
- PostgreSQL and Redis persistence
- Human-in-the-loop integration
- Restart and failure recovery

Production Agentic Systems



1 Security & Compliance

2 Monitoring & Observability

3 Multi-Agent Systems

4 LangGraph Platform

5 Agent-to-Agent Communication

Enterprise Deployment

Multi-Agent Design

Specialised agents handle specific domains. Coordinator agents orchestrate workflows. Agents communicate through structured protocols.

LangSmith Observability

Monitor agent behaviour, trace execution paths, debug failures, and optimise performance with comprehensive telemetry.

Google A2A Protocol

Standardised agent-to-agent communication enables interoperability. Agents negotiate, delegate, and collaborate on complex tasks.

Security Model

MCP security controls tool access. Prompt injection prevention protects against attacks. Audit trails ensure compliance.

99.9%

Uptime

Production-grade reliability with failure recovery

10x

Efficiency

Automation reduces manual effort dramatically

100%

Audit Trail

Complete compliance and governance tracking

Agent Guardrails: Implement safety checks, validate outputs, enforce rate limits, and maintain human oversight for critical decisions. Balance autonomy with control.