**10 Realistic Projects for Data Engineer, Database Administrator, and IT Infrastructure/System Administrator Roles**

**1. Centralized Logging Infrastructure (IT Infrastructure + Data Engineer)**

**Goal:** Implement a centralized logging solution to collect, store, and analyze logs from various servers and applications.

**Tools:** ELK Stack (Elasticsearch, Logstash, Kibana), Filebeat

**Steps:**

1. Install Filebeat on servers to forward logs.
2. Set up Logstash to process and parse logs.
3. Configure Elasticsearch to index and store log data.
4. Visualize logs in Kibana (dashboards, error trends).

**Functionality:** Identify errors, slow system behavior, or potential security breaches in real-time.

**2. Active Directory and Group Policy Automation (IT Infrastructure/System Admin)**

**Goal:** Automate user provisioning and policy deployment using Active Directory and GPO.

**Tools:** Windows Server, PowerShell

**Steps:**

1. Design OU structure (Users, Computers, Departments).
2. Create user provisioning scripts with PowerShell.
3. Deploy security policies via GPO (password policy, firewall rules).
4. Test new users and validate policy application.

**Functionality:** Ensure security compliance and reduce manual IT workload.

**3. Enterprise Backup & Disaster Recovery Plan (IT Infrastructure + DBA)**

**Goal:** Implement backup strategies for both infrastructure and databases.

**Tools:** Veeam, Windows Server Backup, pgBackRest (PostgreSQL), RMAN (Oracle)

**Steps:**

1. Identify critical systems and DBs for backup.
2. Schedule full/incremental backups.
3. Test recovery procedures.
4. Document RTO and RPO objectives.

**Functionality:** Business continuity in case of hardware failures or data loss.

**4. Automated ETL Pipeline for Data Ingestion (Data Engineer)**

**Goal:** Create a robust ETL pipeline that extracts data from APIs/files, transforms it, and loads into a data warehouse.

**Tools:** Python, Apache Airflow, PostgreSQL, Pandas

**Steps:**

1. Create Airflow DAG for daily ingestion.
2. Extract from API/CSV.
3. Transform data (rename columns, fix types).
4. Load into PostgreSQL table.

**Functionality:** Automates data preparation for reporting.

**5. Database Health Monitoring and Performance Tuning (DBA)**

**Goal:** Monitor database usage and optimize queries for performance.

**Tools:** pgAdmin, Oracle AWR Reports, SQL Profiler, Nagios

**Steps:**

1. Set up monitoring for CPU, I/O, memory usage.
2. Identify slow queries.
3. Tune indexes and optimize queries.
4. Implement alerts for anomalies.

**Functionality:** Ensures database performance and reduces downtime.

**6. Data Warehouse for Business Intelligence (DBA + Data Engineer)**

**Goal:** Design and build a scalable data warehouse using a star schema for business analytics.

**Tools:** PostgreSQL, dbt, Power BI, Snowflake

**Steps:**

1. Design dimension/fact tables.
2. Implement transformation models in dbt.
3. Load data into data warehouse.
4. Connect BI tool and build dashboards.

**Functionality:** Provides clean, structured data for business decision-making.

**7. Cloud Infrastructure Setup with Monitoring (IT Infrastructure)**

**Goal:** Deploy a hybrid cloud environment with monitoring tools.

**Tools:** AWS/Azure, Zabbix/Prometheus, Terraform

**Steps:**

1. Set up VMs, security groups, and networking.
2. Deploy monitoring agents.
3. Configure alerts and dashboards.
4. Use Terraform to automate future deployments.

**Functionality:** Scalable infrastructure with visibility and control.

**8. Role-Based Access Control System (IT Infrastructure + DBA)**

**Goal:** Enforce data and system security using RBAC policies.

**Tools:** Active Directory, PostgreSQL/MySQL, LDAP

**Steps:**

1. Create roles and assign user privileges.
2. Set permissions on shared folders and databases.
3. Audit access and monitor policy violations.
4. Implement MFA/SSO where applicable.

**Functionality:** Restricts access based on job roles to improve security.

**9. Data Quality Validation Framework (Data Engineer)**

**Goal:** Ensure incoming data meets defined quality standards.

**Tools:** Great Expectations, Airflow, Pandas

**Steps:**

1. Define data expectations (e.g., no nulls in ID column).
2. Integrate checks in ETL workflows.
3. Generate validation reports.
4. Send alerts on failure.

**Functionality:** Prevents dirty data from affecting reporting and analytics.

**10. SQL Database Migration and Version Control (DBA + Infra)**

**Goal:** Migrate SQL databases between environments with version control.

**Tools:** Liquibase/Flyway, Git, PostgreSQL/MySQL

**Steps:**

1. Write migration scripts.
2. Store and version them in Git.
3. Use Liquibase to deploy changes.
4. Validate and rollback if needed.

**Functionality:** Ensures controlled and trackable changes to database schema.