

Data Analyst Agent System Documentation

Overview

The Data Analyst Agent System is a comprehensive business intelligence solution built using Google's Agent Development Kit (ADK). The system provides automated data analytics, report generation, and data visualization capabilities for business data stored in databases, with all monetary values presented in Nigerian Naira (₦).

Technology Stack

- **Framework:** Google Agent Development Kit (ADK)
 - **Model:** Gemini 2.0 Flash
 - **Data Visualization:** Matplotlib
 - **Database:** SQL-compatible database systems
 - **Programming Language:** Python
 - **File Management:** Pathlib for cross-platform file handling
-

Agent Architecture

1. Main Agent: `analyst_agent`

Name: `analyst_agent`

Model: `gemini-2.0-flash`

Type: Primary orchestrator agent

Description

The main data analytics agent that serves as the central coordinator for business intelligence operations. It receives user requests, formulates analytical objectives, and delegates tasks to specialized sub-agents to provide comprehensive business reports and insights.

Capabilities

- **Report Generation:** Creates comprehensive business analytics reports
- **Data Analysis:** Performs complex business data analysis across multiple dimensions
- **Objective Planning:** Automatically breaks down user requests into actionable analytical objectives
- **Error Handling:** Robust error management with adaptive approach modification

- **Visual Integration:** Automatically incorporates relevant visualizations into reports
- **Currency Handling:** All financial data presented in Nigerian Naira (₦)

Sub-agents

1. **SQL Query Agent** (`sql_query_agent`)
2. **Data Visualization Agent** (`data_visualization_agent`)

Agent Tools

- `AgentTool(sql_query_agent)` - For database operations and data retrieval
- `AgentTool(data_visualization_agent)` - For creating charts and visual representations

Key Features

- **Automatic Business Context:** Retrieves business information from database without user input
 - **Smart Visual Selection:** Determines when visualizations enhance reports
 - **Comprehensive Reporting:** Provides analysis summaries alongside raw data
 - **Adaptive Processing:** Modifies approach when encountering errors
-

2. Sub-agent: `sql_query_agent`

Name: `sql_query_agent`

Model: `gemini-2.0-flash`

Type: Database operations specialist

Description

A specialized agent responsible for all database interactions, SQL query generation, and data retrieval operations. It serves as the primary data source for the analytics system.

Capabilities

- **Dynamic Query Generation:** Creates optimized SQL queries based on analytical requirements
- **Schema Analysis:** Automatically retrieves and analyzes database structure
- **Data Validation:** Ensures data accuracy and existence before query execution
- **Complex Query Support:** Handles joins, aggregations, and multi-table operations
- **Temporary Views:** Creates intermediate result sets for complex analyses
- **Active Data Filtering:** Automatically filters for active products (active=1)

Tools

- `draft_select_query` - Generates SELECT statements with full SQL feature support
- `draft_temp_view_query` - Creates temporary views for complex operations
- `list_tables` - Retrieves all available database tables
- `describe_table` - Gets detailed table schema information
- `execute_query` - Executes SQL queries against the database
- `get_schema_for_all_tables` - Comprehensive database schema retrieval

Workflow

1. Receive analytical requirements from parent agent
 2. Retrieve database schema and structure
 3. Generate and validate SQL queries
 4. Create temporary views if needed for complex operations
 5. Execute queries and return results
 6. Handle errors and retry with modified approaches
-

3. Sub-agent: `data_visualization_agent`

Name: `data_visualization_agent`

Model: `gemini-2.0-flash`

Type: Data visualization specialist

Description

A specialized agent focused on creating compelling and informative data visualizations. It automatically selects appropriate chart types based on data characteristics and analytical context.

Capabilities

- **Smart Chart Selection:** Automatically chooses optimal visualization types
- **Multi-format Support:** Creates bar charts, line charts, scatter plots, and pie charts
- **Visual Enhancement:** Applies colors, styling, and visual flair to charts
- **Context-aware Labeling:** Generates informative labels and titles
- **File Management:** Saves visualizations with organized naming conventions
- **Responsive Design:** Adapts chart orientation and sizing for optimal presentation

Tools

- `create_and_save_bar_chart` - For categorical data comparison and distribution
- `create_and_save_line_chart` - For trend analysis and time-series data
- `create_and_save_scatter_chart` - For correlation and relationship analysis
- `create_and_save_pie_chart` - For proportion and percentage breakdowns

Chart Types & Use Cases

Bar Charts

- **Purpose:** Distribution comparison, magnitude analysis
- **Features:** Horizontal/vertical orientation, customizable colors, adjustable bar width
- **Best for:** Sales by product, revenue by region, category comparisons

Line Charts

- **Purpose:** Trend analysis, time-series data
- **Features:** Multiple line styles, markers, custom axis limits
- **Best for:** Sales over time, growth trends, performance tracking

Scatter Charts

- **Purpose:** Relationship analysis, correlation studies
- **Features:** Variable point sizes, color mapping, transparency control
- **Best for:** Price vs. sales correlation, customer behavior analysis

Pie Charts

- **Purpose:** Part-to-whole relationships, percentage breakdowns
- **Features:** Slice explosion, custom colors, comprehensive labeling
- **Best for:** Market share, category distribution, proportion analysis

Common Reports and Analytics

Sales Reports

- **Total Sales Over Time:** Trend analysis with customizable time periods (daily, weekly, monthly, yearly)
- **Top-Selling Products:** Volume and revenue-based product rankings
- **Revenue Per Customer:** Customer value analysis and segmentation

Inventory Analytics

- **Low Stock Alerts:** Products approaching or below minimum thresholds
- **Expiration Management:** Soon-to-expire and expired product tracking
- **Stock Level Overview:** Comprehensive inventory status reporting

Customer & Behavior Analytics

- **Customer Retention Rate:** Loyalty and engagement metrics over time
- **Visit Frequency Analysis:** Customer visit patterns and timing
- **Demographics Breakdown:** Customer base analysis for targeted marketing

Order Performance

- **Order Status Tracking:** Pending vs. fulfilled order analysis
- **Discount Impact Analysis:** Promotion effectiveness and margin impact

Supplier Performance

- **Supply Fulfillment Rates:** Supplier reliability assessment
 - **Cost Analysis:** Supplier cost comparison and negotiation insights
-

Technical Implementation Details

Database Integration

- Automatic schema discovery and validation
- Active record filtering (active=1 for products)
- Support for complex joins and aggregations
- Temporary view creation for multi-step analyses

File Management

- Organized visual storage by business ID
- PNG format for all generated charts
- Cross-platform path handling using Pathlib
- Automatic file naming and organization

Error Handling

- Comprehensive try-catch blocks in visualization tools
- Adaptive query modification on database errors
- Graceful degradation with informative error messages
- Retry mechanisms with alternative approaches

Performance Optimization

- Efficient query generation with appropriate limits
 - Selective data retrieval based on analytical needs
 - Optimized visualization rendering
 - Memory-conscious temporary view management
-

Usage Guidelines

For Business Users

1. Request reports using natural language
2. Specify time periods or focus areas when relevant
3. System automatically handles business identification and data retrieval
4. Visualizations are automatically generated when beneficial

For Developers

1. Extend capabilities by adding new tools to sub-agents
2. Modify chart types by updating visualization agent tools
3. Add new report types through analyst agent instruction updates
4. Ensure database compatibility through proper schema design

Best Practices

- Allow agents to retrieve business context automatically
 - Specify analytical objectives clearly
 - Trust agent judgment on visualization selection
 - Review generated SQL queries for optimization opportunities
 - Organize visual outputs by business for easy access
-

System Requirements

- Python environment with matplotlib
- Google ADK framework
- SQL-compatible database with proper schema
- File system access for visual storage
- Sufficient memory for data processing and visualization generation