Improved Two-Week Premier League Database Project Plan

Overview & Scope

This plan outlines a collaborative approach for a team of four to develop a Premier League analytics database in two weeks, ensuring all members are involved across all aspects while maintaining clear accountability.

MVP (Must-Have) Components

- Operational database with complete match data (1993-2024)
- Core dimension tables (Teams, Seasons, Referees)
- Basic analytical star schema with TeamMatchStatsFact
- Fundamental queries for team performance, referee analysis
- Simple GUI for data visualization and querying

Nice-to-Have (Time Permitting)

- Advanced betting odds analysis
- Match prediction models
- Admin interface with data correction tools
- Additional fact tables beyond core TeamMatchStatsFact

Week 1: Database Design, ETL & Core Implementation

Days 1-2: Schema Design & Environment Setup (All Team Members)

Morning Sessions: Collaborative Schema Design (9:15 AM - 12:00 PM)

- All team members jointly review CSV files and agree on database structure
- Collectively finalize the operational and analytical schemas

Specific Task Ownership:

- **Person 1**: Implement and test Teams and Seasons tables (3 hours)
  - Define team standardization rules
  - Create reference table for team name variations

    - Review by Person 3
- **Person 2**: Implement and test Matches and MatchStatistics tables (4 hours)
    - Define field mappings from CSV to database
    - Create data type conversion rules
    - Review by Person 4
- **Person 3**: Implement and test Referees and BettingOdds tables (3 hours)
    - Create normalization structure for odds data
    - Define bookmaker reference data
    - Review by Person 1
- **Person 4**: Set up and document infrastructure (3 hours)
    - Configure shared repository
    - Create database connection utilities
    - Set up shared development environment
    - Review by Person 2

Assistance Examples:

- Person 1 helps Person 2 normalize team references in Matches table
- Person 3 assists Person 4 with database connection testing
- All members review each other's CREATE TABLE statements

Afternoon Sessions: Shared Utilities Development (1:00 PM - 4:30 PM)

- Develop and test common utilities needed for ETL process
- Establish code documentation standards

Day 2 Deliverables:

- Complete operational database schema SQL scripts
- Initial analytical database schema design
- Shared utilities for database operations
- Team agreement on data standardization approaches

Days 3-5: ETL Development & Data Loading (All Team Members)

Morning Sessions: ETL Framework Development (9:15 AM - 12:00 PM)

- Develop modular ETL components that all team members will use

ETL Component Ownership:

- **Person 1**: Team name standardization module (3 hours)
  - Create function to map variant team names to standard IDs
  - Build validation tests for team name mapping
  - Document standardization logic
  - Review by Person 2
- **Person 2**: Match data transformation module (4 hours)
  - Develop functions to parse and transform match data
  - Create data type validation tests
  - Handle special cases (postponed matches, etc.)
  - Review by Person 3
- **Person 3**: Betting odds normalization module (4 hours)
  - Create functions to extract and normalize odds data
  - Handle missing odds values
  - Calculate implied probabilities
  - Review by Person 4
- **Person 4**: Data validation and error handling framework (3 hours)
  - Develop logging system for ETL errors
  - Create data quality check functions
  - Implement error recovery mechanisms
  - Review by Person 1

Afternoon Sessions: Data Loading By Season Ranges (1:00 PM - 4:30 PM)

- Each person processes complete ETL for their assigned seasons
- Everyone applies all modules (standardization, transformation, validation)

Season Assignments:

- **Person 1**: Seasons 1993-1999 (7 seasons)
  - Load Teams and Seasons data first
  - Process match data with complete validation
  - Document any data anomalies from early seasons
- **Person 2**: Seasons 2000-2006 (7 seasons)
  - Focus on consistent match statistics fields
  - Track and document field additions over time
  - Verify referential integrity
- **Person 3**: Seasons 2007-2014 (8 seasons)
  - Handle increasingly complex betting data
  - Ensure bookmaker consistency across seasons

    - Verify data completeness
- **Person 4**: Seasons 2015-2024 (10 seasons)
    - Process most recent and detailed data
    - Document contemporary data structure
    - Perform cross-season validation

Data Validation Checkpoints:

- After each season load: Verify match counts, team appearances, goals totals
- After each day: Cross-check season boundary consistency
- Daily data quality reports with specific metrics

Day 5 Deliverable: Complete Operational Database

- Fully populated operational database with historical data
- Documentation of data anomalies and solutions
- Validation report showing data integrity metrics
- Initial indexes on critical fields

Week 2: Analytical Database, Queries & Application Development

Days 6-7: Analytical Database & Core Queries (All Team Members)

Morning Sessions: Analytical Schema Implementation (9:15 AM - 12:00 PM)

- Develop and execute ETL process from operational to analytical database

Analytical Component Ownership:

- **Person 1**: TeamMatchStatsFact implementation (4 hours)
    - Transform match data into team perspective (one record per team per match)
    - Calculate performance metrics (shot efficiency, etc.)
    - Create aggregation tables for team performance
    - Review by Person 3
- **Person 2**: BettingMarketFact implementation (4 hours)
    - Normalize and transform betting odds
    - Calculate derived betting metrics (implied probability, etc.)
    - Create bookmaker comparison structures

    - Review by Person 4
- **Person 3**: Dimension tables implementation (3 hours)
    - Create TimeDimension with proper hierarchies
    - Enhance TeamDimension with classification attributes
    - Build RefereeDimension with performance metrics
    - Review by Person 1
- **Person 4**: Incremental update process (3 hours)
    - Create procedures for adding new matches
    - Implement dimension table updates
    - Develop fact table refresh logic
    - Review by Person 2

OLTP to OLAP Transformation Examples:

- Converting normalized match records to denormalized team performance facts
- Translating team IDs to dimension references
- Calculating period-based aggregations (form over last N matches)
- Deriving home/away-specific metrics from base match data

Afternoon Sessions: Core Query Development (1:00 PM - 4:30 PM)

- Develop analytical queries that leverage the star schema

Query Development Ownership:

- **Person 1**: Team performance analysis queries (4 hours)
    - Home vs. away performance comparison
    - Form analysis with trending
    - Head-to-head history generation
    - Review by Person 2
- **Person 2**: Expected goals model and efficiency queries (4 hours)
    - Shot efficiency calculation
    - Expected goals model based on shot data
    - Performance vs. expectations analysis
    - Review by Person 3
- **Person 3**: Referee analysis queries (3 hours)
    - Referee bias detection
    - Discipline patterns analysis
    - Home advantage by referee

    - Review by Person 4
- **Person 4**: League table and standings queries (3 hours)
    - Dynamic league table generation
    - Historical position tracking
    - Form-based predictions
    - Review by Person 1

Testing Approach for Analytical Components:

- Unit tests for each calculated metric
- Validation of aggregation accuracy
- Performance testing of complex queries
- Cross-checking operational vs. analytical results

Day 7 Deliverable: Analytical Database and Queries

- Populated analytical star schema
- Suite of tested analytical queries
- Performance benchmarks for key queries
- Data validation confirming analytical accuracy

Days 8-10: GUI Development & Project Finalization (All Team Members)

Morning Sessions: GUI Component Development (9:15 AM - 12:00 PM)

- Develop Python-based GUI components for the application

GUI Component Ownership:

- **Person 1**: Team dashboard screens (4 hours)
    - Components:
        - Team performance comparison charts
        - Home vs. away performance tables
        - Form tracker with visual indicators
        - Head-to-head record display
    - Review by Person 4
- **Person 2**: Match analysis and prediction interface (4 hours)
    - Components:
        - Match detail view with statistics

        - Expected goals visualization
        - Form comparison between teams
        - Simple prediction indicators
    - Review by Person 1
- **Person 3**: League table and reports (4 hours)
    - Components:
        - Interactive league table with filters
        - Season selector and comparison
        - Performance metrics visualization
        - Export to PDF/CSV functionality
    - Review by Person 2
- **Person 4**: Data explorer and query interface (4 hours)
    - Components:
        - Parameter-based query builder
        - Results grid with sorting/filtering
        - Basic data visualization options
        - Query saving functionality
    - Review by Person 3

Afternoon Sessions: Integration & Testing (1:00 PM - 4:30 PM)

- Integrate GUI components with database
- Perform comprehensive system testing
- Create user documentation

Integration Tasks:

- Connect GUI components to database queries
- Implement navigation between screens
- Create consistent styling and user experience
- Test full application workflows

Testing Protocol:

- Functional testing of all GUI components
- Database query performance under load
- Error handling for edge cases
- Cross-platform compatibility testing

Final Deliverables (Day 10):

- Complete application with documentation
- User guide with examples
- Technical documentation of database and code
- Presentation materials for final demo

Cross-Cutting Responsibilities

Each team member maintains responsibility for a cross-cutting concern throughout the project:

- **Person 1**: Architecture consistency and performance optimization
  - Review database design decisions for consistency
  - Monitor and optimize query performance
  - Ensure efficient data access patterns
- **Person 2**: Data quality and transformation correctness
  - Verify data transformations preserve meaning
  - Develop and apply data quality metrics
  - Document data lineage and transformations
- **Person 3**: Documentation and standards compliance
  - Maintain comprehensive documentation
  - Ensure code adheres to agreed standards
  - Create user-facing documentation
- **Person 4**: Testing thoroughness and validation
  - Develop and execute test plans
  - Validate results against expected outcomes
  - Ensure error handling is comprehensive

Daily Schedule

9:00-9:15 AM: Daily stand-up

- Review yesterday's accomplishments
- Set goals for today
- Identify any blockers

9:15-12:00 PM: Morning session work

- Focus on primary development tasks
- Collaborative work as needed

12:00-1:00 PM: Lunch & informal check-in

1:00-4:30 PM: Afternoon session work

- Continue morning tasks or shift to integration
- Cross-functional collaboration

4:30-5:00 PM: End-of-day review

- Demo completed components
- Code review session
- Adjustment of next day's plan if needed

Risk Mitigation Strategies

Schedule Risks

- **Risk**: ETL process takes longer than expected
  - **Mitigation**: Prioritize fewer seasons initially, add more as time permits
  - **Fallback**: Focus on recent seasons (2010-2024) if time is limited
- **Risk**: GUI development is too ambitious
  - **Mitigation**: Start with essential screens, add features incrementally
  - **Fallback**: Simplify to basic visualization without advanced interactivity

Technical Risks

- **Risk**: Data inconsistencies between seasons
  - **Mitigation**: Create robust validation and transformation rules
  - **Fallback**: Document inconsistencies and handle special cases
- **Risk**: Performance issues with large dataset
  - **Mitigation**: Implement proper indexing from the start
  - **Fallback**: Optimize critical queries, defer less important ones

Team Risks

- **Risk**: Team member unavailability
  - **Mitigation**: Cross-train on all components
  - **Fallback**: Reassign tasks with clear priorities on essential components
- **Risk**: Integration challenges between components
  - **Mitigation**: Establish clear interfaces early
  - **Fallback**: Scheduled integration sessions with whole team

Project Tracking

- Daily updates to shared project board
- Version control with feature branches and pull requests
- Documentation updates with each significant component completion
- Daily data quality and progress metrics

This plan ensures all team members contribute across all aspects of the project while maintaining clear ownership and accountability for specific components. The structure promotes collaboration while delivering a complete solution within the two-week timeframe.