

The background of the slide features a faint, light-colored basketball court. Overlaid on the court are several network diagrams. These diagrams consist of nodes (represented by small circles in red, black, and grey) connected by thin lines, forming a complex web. The nodes are distributed across the slide, with some clusters appearing more dense than others.

NBA Fantasy Basketball Predictor

**A Data Management System
for Sports Analytics**

Harsh and Aryan

The Problem: Data Fragmentation

A background network diagram consisting of numerous nodes (represented by small circles in shades of red, black, and grey) connected by thin, light-colored lines. The nodes are distributed across the slide, with a higher density on the right side, creating a web-like structure that suggests interconnected data or a complex system.

CURRENT STATE:

Managers must manually toggle between 5+ tabs (Box Scores, Injury Reports, Betting Odds) to make decisions.

SOLUTION:

A Unified Data Management System that centralizes ingestion, storage, and analysis into a single "Source of Truth".

System Architecture: The ETL Pipeline

Extraction



nbastats.csv

- Ingesting raw nbastats.csv with 500+ player records



Transformation



- Cleaning null values
- handling UTF-8 encoding (e.g., 'Dončić')
- applying ESPN scoring weights



Loading



fantasy.db

- Storing structured data into persistent SQLite database named fantasy.db

How does it work?

The following screenshot shows how our project has two interfaces one where you can search any nba player and get their Overall rank , position rank and fantasy points average , the other tab helps in comparing players and tells you which player is better

The image displays two side-by-side browser windows showing the ESPN Fantasy Basketball Calculator application. Both windows are running on localhost.

Left Window (Search Player Tab):

- ESPN Fantasy Basketball Calculator**
- Search players, compute fantasy points, rankings, and compare players.
- Navigation: **Search Player** (active), Player Comparison
- Input: Search player by name: [Empty text box]

Right Window (Player Comparison Tab):

- ESPN Fantasy Basketball Calculator**
- Search players, compute fantasy points, rankings, and compare players.
- Navigation: Search Player, **Player Comparison** (active)
- Select Player 1: Nikola Jokić
- Select Player 2: Tyrese Maxey
- Player Comparison**
- Table:

	Metric	Player 1	Player 2
0	Player Name	Nikola Jokić	Tyrese Maxey
1	Position	C	PG
2	Fantasy Points	58.4	51.2
3	Overall Rank	1	3
4	Position Rank	1	2

Result: Nikola Jokić is the better fantasy option.

Database Design: SQL Implementation

The system transitions from flat files (like CSVs) to a robust Relational Database (2NF), improving data structure and manageability.

Players Table (Dimension)

Player_ID (PK)	Name	Team	Position
1	LeBron James	LAL	SF
2	Stephen Curry	GSW	PG

Player_Stats Table (Fact)

Stat_ID (PK)	Player_ID (FK)	Points	Rebounds	Assists
101	1	25.4	7.8	10.2
102	2	29.1	6.1	6.3

- Normalization ensures data integrity and reduces redundancy.

Methodology: SQL-Based Analysis



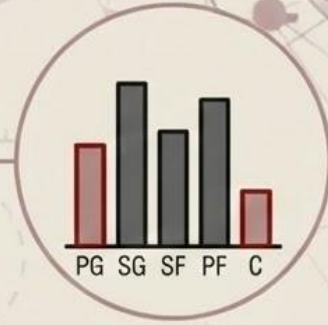
1. True Value Calculator

Ranking players by actual production, not reputation.



2. Efficiency Hunter

Calculating Fantasy Points Per Minute to find "Sleepers".



3. Positional Scarcity

Using GROUP BY to find which positions are hardest to fill.

Analytical Results



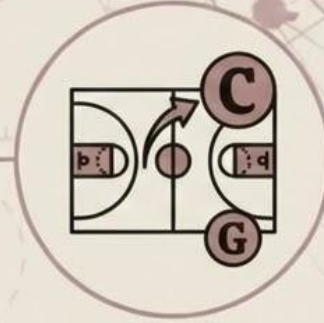
1. Undervalued Asset

Defensive stats (Stocks) are the most undervalued asset.



2. Top Waiver-Wire Targets

The 'Efficiency Per Minute' query identified backup Centers as top waiver-wire targets.



3. 'Big Man First' Strategy

Positional Analysis reveals a surplus of Guards but a scarcity of elite Centers, dictating a 'Big Man First' draft strategy.

Visualization: Streamlit Dashboard

The user-facing web application provides an interactive interface to the SQL-based analysis.



1. Dynamic Search

Real-time SQL querying of player ranks.



2. Comparator Engine

A Head-to-Head tool compares two players and algorithmically declares a 'Winner'.



This democratizes access to the data without requiring SQL knowledge.

Future Roadmap



Phase 1: Real-Time API

Replacing CSVs with
live nba_api data.

Phase 2: Predictive ML

Training a Random Forest
model to forecast future games.

Phase 3: Cloud Deployment

Hosting the database
on AWS.



This project successfully transformed raw, fragmented data into a competitive strategic advantage.



Thank you