

Interface Design

Our project uses a command-line interface (CLI) with an orange basketball-themed colour scheme, since we're working with NBA data. When the program is run, the user is greeted with a welcome message that briefly describes the database and what information is available to them. The welcome message instructs the user to type `help` to see a list of available commands. All query results are printed using dynamically sized columns with orange headers that are underlined for readability.

The help menu is divided into three sections. The first section lists the simple queries, which display entire tables. These simple queries are used as reference tools for getting the input needed for complex queries. For example, if a complex query requires a GameID, the user should run the game table query to find the GameID of interest, then use that value as input for the complex query. The second section lists the complex queries. Finally, the third section lists system-level commands like clearing or repopulating the database, clearing the screen, or exiting the program. Each command in the help menu includes the command the user must enter, along with any parameters, and a short description of what the command does.

The interface is implemented in Python. We created a `DatabaseManager` class, which serves as the central manager for all database operations. It manages the connection by reading the configuration file and runs SQL scripts to create tables and views, as well as clear the entire database. It also orchestrates data loading and querying through its `DataLoader` and `QueryManager` instance variables. The `DataLoader` loads and validates all CSV files into Python using `pandas`, then inserts the data into the SQL Server using prepared statements through `pymssql`. The `QueryManager` class handles all queries sent to SQL Server, again using prepared statements to prevent SQL injection. Finally, we created an `Interface` class that handles user interaction. It displays the welcome message, help menu, and neatly formatted query results. Additionally, it parses user input and sends commands to the `DatabaseManager`.

Diagrams

This section shows diagrams illustrating how the interface will look.

```
(base) johnnylee@Johnnys-MacBook-Air-3 Stage_6 % python main.py
=====
    🏀 NBA 2024-2025 SEASON DATABASE 🏀
=====

Welcome to the 2024-2025 NBA Season Database. This interface
allows you to explore teams, players, game, coaches, draft
stats, and more.

Type 'help' to see a list of available commands.

> █
```

Figure 1: Welcome Message Screenshot

```

> help
=====
HELP MENU - NBA DATABASE ⚽
=====

Simple Queries:
s1           List all player draft combine stats
s2           List all player drafts
s3           List all coaches along with their regular season stats
s4           List all coaches along with their playoff season stats
s5           List all arenas
s6           List all teams
s7           List all games in the 2024-2025 season
s8           List all players along with personal information

Complex Queries:
q1 <N>          Show the top N teams by wins along with their season averages
q2             Show coaches with 5+ seasons and a positive playoff win rate
q3 <team_name> [--avg]  Shows a team's average PPG (optionally include league average)
q4 <limit> <page>    List <limit> players from <page> by position/height with PPG/APG/SPG
q5             Show the team and coach that won the 2024-2025 championship
q6 <team_name>      Show a team's roster ordered by PPG, APG, and SPG
q7 <stat> <N>       Show the top N players ordered by a chosen stat (PPG, APG, SPG)
q8 <team_name>      Show the average age of players on a team
q9 <game_id>        Show both teams' stats for a specific game
q10 <min_attempts> <N>   Show the top N players ordered by 3P% with a minimum attempt requirement
q11 <game_id>        Show centers that are 6'5" and their points in a specific game
q12 <home_name> <away_name> Show a home team's win percentage against a specific visiting team
q13 <min_winrate>     Show coaches with a playoff winrate ≥ a decimal input (ex. 0.75)
q14             Show arenas where every team has won at least once

System:
clear-db        Clear the entire database (drop all tables)
load            Clear and populate the database
clear           Clear the terminal screen
exit            Quit the program
=====

> []

```

Figure 2: Help Menu Screenshot

Stage_6 — python main.py — 87x40

```
> s5
```

ArenaName	SeatingCapacity	OpeningYear	ArenaCityName
American Airlines Center	19200	2001	Dallas
Ball Arena	19520	1999	Denver
Barclays Center	17732	2012	Brooklyn
Capital One Arena	20356	1997	Washington
Chase Center	18064	2019	San Francisco
Crypto.com Arena	19079	1999	Los Angeles
Delta Center	18386	1991	Salt Lake City
FedExForum	17794	2004	Memphis
Fiserv Forum	17385	2018	Milwaukee
Footprint Center	17071	1992	Phoenix
Frost Bank Center	18418	2002	San Antonio
Gainbridge Fieldhouse	17274	1999	Indianapolis
Golden 1 Center	17688	2016	Sacramento
Kaseya Center	19600	1999	Miami
Kia Center	18846	2010	Orlando
Little Caesars Arena	20332	2017	Detroit
Madison Square Garden	19812	1968	New York City
Moda Center	19393	1995	Portland
Paycom Center	18203	2002	Oklahoma City
Rocket Mortgage Fieldhouse	19432	1994	Cleveland
Scotiabank Arena	19800	1999	Toronto
Smoothie King Center	17791	1999	New Orleans
Spectrum Center	19077	2005	Charlotte
State Farm Arena	16600	1999	Atlanta
Target Center	18798	1990	Minneapolis
TD Garden	19156	1995	Boston
Toyota Center	18104	2003	Houston
United Center	20917	1994	Chicago
Wells Fargo Center	21000	1996	Philadelphia

```
> █
```

Figure 3: Query Results Screenshot