PuckStats: Analyzing NHL Game Data with SQL

SIDDHARTHA DHEER, University at Buffalo, USA
NOLAN PHILLIPS, University at Buffalo, USA

Ice hockey is a highly dynamic and data-intensive sport where player performance, team strategies, and match outcomes can be quantitatively analyzed through comprehensive statistical data. With the increasing reliance on analytics in professional sports, structured data management has become essential for deriving actionable insights. This paper presents *PuckStats*, a relational database system designed to store, manage, and query National Hockey League (NHL) game data from the 2024 season. By leveraging Structured Query Language (SQL) and relational database principles, this system enables efficient data retrieval and analysis, facilitating an in-depth exploration of player and team performance metrics.

ACM Reference Format:
Siddhartha Dheer and Nolan Phillips. 2025. PuckStats: Analyzing NHL Game Data with SQL. 1, 1 (April 2025), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Overview of the Project

Ice hockey is a sport that generates a vast amount of structured data, encompassing player statistics, game strategies, and match outcomes. With the growing importance of data analytics in professional sports, there is an increasing demand for well-structured and queryable datasets that allow analysts to derive meaningful insights. The *PuckStats* project aims to develop a robust relational database system that systematically organizes and stores National Hockey League (NHL) game data from the 2024 season.

By utilizing Structured Query Language (SQL) and relational database modeling techniques, this system ensures efficient data management and retrieval. The database schema is meticulously designed to capture essential hockey statistics, including *goals, faceoffs, hits, penalties, shifts, shots on goal, missed shots, and turnovers*. These attributes are structured within a well-normalized relational model to ensure data integrity, consistency, and optimized performance for querying player and team performance.

The primary objective of this project is to develop an analytical framework that enhances the accessibility and usability of NHL game data. By efficiently structuring match records, identifying statistical trends, and generating meaningful insights, *PuckStats* provides valuable benefits to various stakeholders, including *coaches, analysts, sports journalists, and data scientists*. Through advanced querying and data visualization capabilities, this system has the potential to significantly improve decision-making processes in hockey analytics.

1.2 Justification for a Database over Spreadsheet-Based Solutions

Traditional methods of sports data analysis, such as Excel spreadsheets, are fundamentally limited in handling large-scale and relational datasets. While spreadsheets may be suitable for small-scale tabular data, they lack the scalability, efficiency, and structural integrity required for complex multi-relational datasets. As the volume and complexity of hockey data grow, a database-driven approach becomes indispensable for managing structured data efficiently. The *PuckStats* database, built using SQL and relational database design principles, offers several advantages over spreadsheet-based solutions, ensuring robustness in data handling and query execution.

One of the most significant advantages of a relational database is scalability and efficiency. Unlike Excel, which experiences performance degradation when handling thousands of records, SQL databases effectively manage large datasets by organizing information into well-defined relations (tables) and employing indexing techniques to optimize query execution. This structured approach ensures fast and reliable data retrieval, making it ideal for large-scale data analysis.

Furthermore, SQL enables advanced querying capabilities that are not feasible within spreadsheet environments. Through operations such as JOINs, GROUP BY aggregations, subqueries, and filtering, complex multi-dimensional datasets can be analyzed with precision. For instance, the system can execute queries to assess player performance trends across multiple games, determine the most effective line combinations in a season, or evaluate goal-scoring efficiency under varying game conditions. These sophisticated queries provide data-driven insights that are difficult to achieve using manual spreadsheet functions.

Maintaining data integrity and normalization is another key advantage of a database-centric approach. The *PuckStats* database adheres to Boyce-Codd Normal Form (BCNF), ensuring the elimination of data redundancy and preserving referential integrity. By structuring data relationships through foreign keys and primary keys, the system prevents inconsistencies and anomalies caused by manual data entry errors. This design also facilitates real-time data updates, ensuring that records remain accurate, consistent, and logically structured.

Additionally, SQL databases provide performance optimization mechanisms that significantly enhance query execution. By leveraging indexing strategies, optimized queries, and referential constraints, the *PuckStats* database minimizes retrieval time and computational overhead, outperforming conventional spreadsheet-based approaches. These optimizations are particularly crucial in handling large and evolving datasets, where high-speed analytical processing is required.

Authors' addresses: Siddhartha Dheer, University at Buffalo, Buffalo, New York, USA; Nolan Phillips, University at Buffalo, Buffalo, New York, USA.

ACM XXXX-XXXX/2025/4-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1.3 Real-World Significance of the Problem

The increasing integration of data analytics in professional sports has transformed the way coaches, analysts, and enthusiasts interpret game dynamics and player performance. The ability to extract actionable insights from structured data has become a fundamental component of modern sports strategy, with applications spanning performance evaluation, game optimization, sports journalism, and even fantasy sports management. The *PuckStats* database aims to bridge the gap between raw NHL statistics and meaningful analytics, enabling various stakeholders to engage in data-driven decision-making.

For coaches and analysts, structured game data provides a foundation for evaluating player effectiveness, line performance, and in-game strategies. By analyzing historical records, coaching staff can identify high-performing players, assess faceoff success rates, monitor penalty trends, and optimize team formations based on statistical evidence. Advanced SQL queries allow for the identification of correlations between gameplay patterns and match outcomes, helping teams refine their strategies for future competitions.

Beyond the realm of team management, sports journalism and fan engagement have also been revolutionized by data-driven storytelling. For sports journalists and analysts, access to structured hockey data enhances in-depth game analysis, allowing for a comprehensive breakdown of player contributions, goal distributions, and seasonal trends. With the increasing popularity of interactive sports media, such insights enrich the viewing experience for audiences, providing them with statistically-backed narratives and comparisons across different games and seasons.

Furthermore, structured sports data is of great importance to sports betting and fantasy leagues, where accurate statistical modeling can provide users with quantitative insights for informed decision-making. By evaluating player consistency, team performance trends, and scoring probabilities, analysts can develop predictive models that support fantasy sports team selections and betting strategies. The ability to efficiently query and extract relevant statistics empowers users with real-time analytical tools, giving them a competitive edge in sports forecasting.

By developing a well-structured SQL database for NHL game analytics, the *PuckStats* project not only provides an optimized data management solution but also highlights the transformative potential of relational databases in sports analytics, strategic planning, and data-driven decision-making.

2 PROBLEM STATEMENT

The increasing role of data analytics in professional sports has revolutionized decision-making for teams, analysts, and fans alike. In the National Hockey League (NHL), performance metrics, player statistics, and game dynamics generate large volumes of data that require efficient organization and analysis. However, traditional methods of data storage, such as spreadsheets or unstructured formats, fail to adequately handle the complexity, volume, and relational nature of hockey statistics. The *PuckStats* project addresses this issue by developing a relational database system that enables structured, scalable, and efficient analysis of NHL game data.

One of the primary challenges in hockey analytics is the ability to derive meaningful insights from raw game data. NHL matches generate detailed records of player movements, scoring events, penalties, faceoffs, shots, and defensive plays, all of which are interdependent. Understanding the relationships between these variables is essential for evaluating team performance, player efficiency, and strategic patterns. Without a structured data model, querying and extracting insights from such a large dataset becomes inefficient and error-prone.

2.1 Why Does This Problem Require Structured Data Management?

A relational database model is essential for managing NHL data due to the interconnected nature of game events and statistics. Unlike simple tabular spreadsheets, which struggle with redundancy and inefficiencies, a relational database:

Eliminates Data Redundancy and Ensures Integrity
By normalizing the data into relations (tables), redundant information is minimized, and data consistency is maintained across multiple games and seasons.

Facilitates Complex Querying for Insights
SQL allows for multi-table joins, aggregations, and filtering, enabling advanced queries such as:

- Identifying top-performing players based on multiple performance indicators.
- Analyzing faceoff success rates in different game conditions.
- Evaluating the impact of penalties and power plays on match outcomes.

Improves Scalability and Performance Optimization
Large-scale NHL datasets contain millions of records, making unstructured storage inefficient. A well-indexed SQL database improves retrieval speed, ensuring real-time analysis for stakeholders.

Enhances Data Relationships for Predictive Analysis
Hockey data is not isolated—player performance is influenced by team formations, opposing strategies, and historical matchups. A relational model enables structured relationships, allowing for trend analysis and predictive modeling.

By leveraging a well-designed SQL-based relational database, the *PuckStats* project provides an efficient and scalable approach to hockey analytics, empowering coaches, analysts, and fans with actionable insights. This structured approach transforms raw game data into a meaningful analytical framework, making it easier to evaluate player performance, optimize team strategies, and enhance fan engagement through data-driven insights.

3 TARGET USERS

The *PuckStats* database is designed to serve multiple stakeholders in the hockey analytics ecosystem, providing them with structured, accessible, and efficient means of querying NHL data.

Who Will Use the Database?
The primary users of the *PuckStats* database include:

- **Coaches and Team Analysts** – They will use the database to analyze player performance, assess team strategies, and optimize line formations based on historical trends.

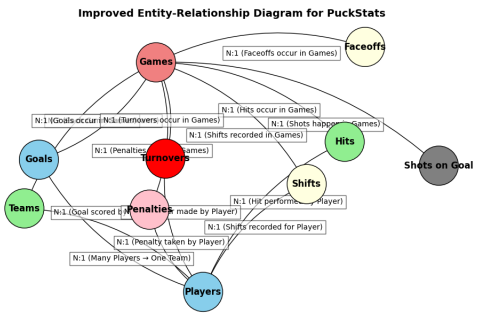

Fig. 1. Final E/R Diagram representing the relationships among Players, Teams, Matches, Events, and Statistics.

- **Sports Journalists and Broadcasters** – They can utilize the database to extract statistics, generate insights, and enhance sports commentary with in-depth data-driven narratives.
- **Fantasy Hockey Players and Sports Bettors** – Users engaged in fantasy sports and betting can leverage data analytics to predict outcomes, assess player consistency, and make informed decisions.
- **Academic Researchers and Data Scientists** – The database serves as a valuable resource for those conducting studies on sports performance metrics, data modeling, and predictive analytics.

Who Will Maintain/Administer the Database?
The administration and maintenance of the *PuckStats* database will be handled by:

- **Database Administrators** – Responsible for ensuring data integrity, optimizing query performance, and managing database security.
- **Developers and Data Engineers** – Tasked with implementing new features, managing ETL (Extract, Transform, Load) processes, and integrating real-time data sources.
- **Sports Organizations and Data Providers** – Entities such as NHL teams, analytics firms, or third-party data providers may manage and update the dataset with live game statistics.

By ensuring a structured maintenance framework, the *PuckStats* database will remain up-to-date, scalable, and optimized for real-time analytics, catering to diverse users in the hockey analytics domain.

4 E/R DIAGRAM

The following diagram represents the final entity-relationship model for the *PuckStats* database. It includes the primary entities and their relationships, ensuring structured representation of NHL game data.

Table	Primary Key (PK)	Foreign Keys (FK)	Relationship Type
Player	player_id	team_id	Many Players → One Team (N:1)
Team	team_id	None	Independent Entity
Match	match_id	home_team, away_team	Each Match has Two Teams (N:2)
Event	event_id	match_id, player_id	Events occur in Matches (N:1), Performed by Players (N:1)
Faceoff	None	game_id, win_player_id, losing_player_id	N:1 (Faceoffs occur in Games)
Shots on Goal	None	game_id, shooter, goalie	N:1 (Shots happen in Games)
Penalties	None	game_id, drawn_by, taken_by	N:1 (Penalties occur in Games)
Goals	None	game_id, scorer, goalie	N:1 (Goals occur in Games)
Hits	None	game_id, hitter, hittee	N:1 (Hits occur in Games)
Turnovers	None	game_id, player_id	N:1 (Turnovers occur in Games)
Shifts	None	game_id, player	N:1 (Shifts recorded in Games)

Table 1. Final Explanation of Table Connections and Cardinality

5 EXPLANATION OF TABLE CONNECTIONS & CARDINALITY

6 SCHEMA & DATA DESIGN

This section describes the schema and data design principles for the *PuckStats* database, including table attributes, primary and foreign keys, and normalization considerations.

6.1 List of Tables and Attributes

The database schema consists of the following tables:

6.2 Normalization Approach

To ensure data integrity and minimize redundancy, the database schema follows the principles of Boyce-Codd Normal Form (BCNF). The key considerations are:

- Each table has a well-defined primary key.
- Functional dependencies are preserved to eliminate redundancy.
- All non-trivial functional dependencies exist only on candidate keys, ensuring BCNF compliance.
- In cases where strict BCNF decomposition would lead to excessive joins, 3NF is selectively applied for efficiency.

Table Name	Primary Key (PK)	Foreign Keys (FK)	Attributes
Players	player_id	team_id	player_id [PK], first_name, last_name, position, team_id [FK]
Teams	team_id	None	team_id [PK], team_abbrev, city, name, conference, division
Games	game_id	home_team, away_team	game_id [PK], away_id [FK], away_score, home_id [FK], home_score, winner [FK], loser [FK]
Faceoffs	None	game_id, win_player_id, losing_player_id	game_id [FK], win_player_id [FK], losing_player_id [FK], period, time_remaining, x_coord, y_coord, zone
Shots on Goal	None	game_id, shooter, goalie	game_id [FK], shooter [FK], goalie [FK], shot_type, x_coord, y_coord, zone
Penalties	None	game_id, drawn_by, taken_by	game_id [FK], penalty_type, drawn_by [FK], taken_by [FK], duration, x_coord, y_coord, zone
Goals	None	game_id, scorer, goalie	game_id [FK], shot_type, scorer [FK], goalie [FK], primary_assist [FK], secondary_assist [FK], x_coord, y_coord, zone
Hits	None	game_id, hitter, hittee	game_id [FK], hitter [FK], hittee [FK], x_coord, y_coord, zone
Turnovers	None	game_id, player_id	game_id [FK], player_id [FK], x_coord, y_coord, zone, play_type
Shifts	None	game_id, player	game_id [FK], start_time, end_time, duration, player [FK]

Table 2. Database Schema: Table Attributes, Primary and Foreign Keys

By adhering to BCNF and selectively using 3NF where necessary, the schema ensures efficient data retrieval while maintaining high performance for query execution.

7 UPDATED SAMPLE DATASETS & QUERIES

The following images present sample datasets and SQL queries demonstrating data retrieval operations in *PuckStats*:

Fig. 2. Query 1 – Team Wins Count.

This query ranks all NHL teams by their total number of wins by joining the `games` *and* `teams` *tables on team ID. It helps identify the top-performing teams across the season.*

SQL Code:

```
SELECT COUNT(*) AS wins, team_abbrev AS team
FROM games
INNER JOIN teams ON games.winner = teams.team_id
GROUP BY team_abbrev
ORDER BY wins DESC;
```

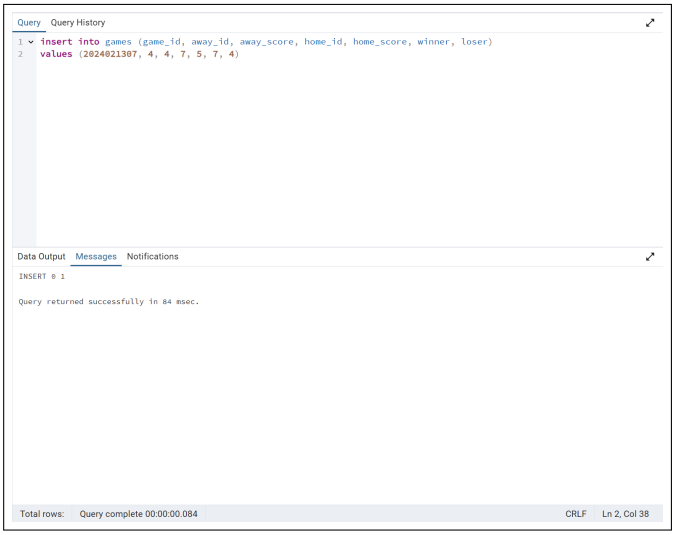

Fig. 3. Query 2 – Empty Net Goal Leaders

This query identifies players who scored the most goals when the opposing team had no goalie on the ice (empty net goals).

SQL Code:

```
SELECT COUNT(*) AS eng,
       CONCAT(first_name, ' ', last_name) AS name
FROM goals
INNER JOIN players ON goals.scorer = players.player_id
WHERE goalie IS NULL
GROUP BY name
ORDER BY eng DESC;
```

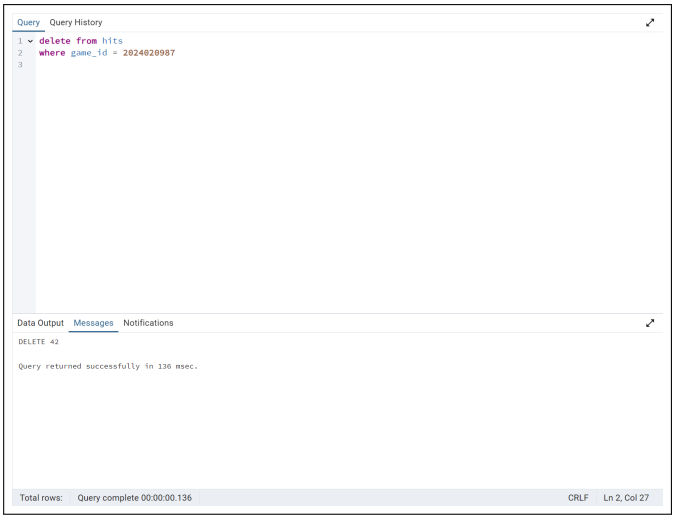

Fig. 4. Query 3 – Most Hit Players in Defensive Zone

This query lists players who were hit the most in the defensive zone, helping identify those frequently targeted under pressure.

SQL Code:

```
SELECT COUNT(*) AS times_hit,
       CONCAT(first_name, ' ', last_name) AS name
FROM hits
INNER JOIN players ON hits.hittee = players.player_id
WHERE zone LIKE 'D'
GROUP BY name
ORDER BY times_hit DESC;
```

This query analyzes which teams had the most missed shots during the final 2 minutes of the third period when their goalie was pulled.

SQL Code:

```
SELECT COUNT(*) AS missed_shots, team_abbrev
FROM missed_shots
INNER JOIN teams ON missed_shots.event_owner = teams.team_id
WHERE period = 3
  AND time_remaining < '02:00:00'
  AND goalie IS NULL
GROUP BY team_abbrev
ORDER BY missed_shots DESC;
```

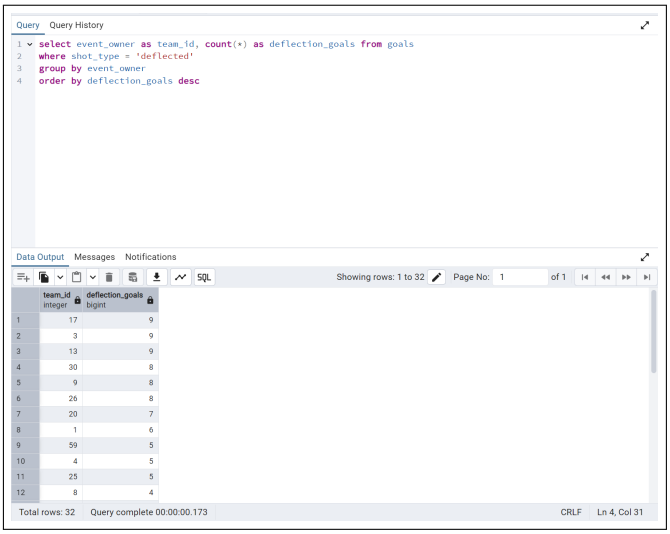

Fig. 5. Query 5 – Deflected Goals by Team.

This query counts how many deflected goals each team has scored, giving insights into redirection-based scoring tactics.

SQL Code:

```
SELECT event_owner AS team_id, COUNT(*) AS deflection_goals
FROM goals WHERE shot_type = 'deflected' GROUP BY event_owner
ORDER BY deflection_goals DESC;
```

ACM Reference Format:
. 2025. . 1, 1 (April 2025), 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1

1.1 Functional Dependencies

To further ensure clarity and logical consistency within the schema, the following functional dependencies were observed:

- **Teams:** `team_id` → `team_abbrev, city, name, conference, division`
- **Games:** `game_id` → `away_id, away_score, home_id, home_score, winner, loser`
- **Players:** `player_id` → `position, first_name, last_name`
- **Shots_on_goal:** `game_id, period, time_remaining` → all remaining attributes
- **Blocks, Missed_shots, Goals, Hits, Turnovers, Faceoffs, Penalties:** `game_id, period, time_remaining` → all respective event-specific attributes
- **Shifts:** `game_id, player, period, start_time` → `end_time, duration, event_owner`

2 QUERY ANALYSIS AND EXECUTION PLANS

2.1 Query 6: Power Play Point Leaders

Objective: Identify players with the most power play points, including goals and assists during man-advantage scenarios.

```
SELECT p.first_name, p.last_name,
  COALESCE(SUM(CASE WHEN g.scorer = p.player_id THEN 1 ELSE 0 END), 0) AS pp_goals,
   COALESCE(SUM(CASE WHEN g.primary_assist = p.player_id
   OR g.secondary_assist = p.player_id THEN 1 ELSE 0 END), 0) AS pp_assists,
  COALESCE(SUM(CASE WHEN g.scorer = p.player_id OR g.primary_assist = p.player_id
   OR g.secondary_assist = p.player_id THEN 1 ELSE 0 END), 0) AS pp_points
FROM goals g
JOIN games gm ON g.game_id = gm.game_id
JOIN players p ON p.player_id IN (g.scorer, g.primary_assist, g.secondary_assist)
WHERE (
  (g.event_owner = gm.home_id AND g.home_skaters > g.away_skaters
  AND NOT (g.home_skaters = 6 AND g.away_skaters = 5)) OR
  (g.event_owner = gm.away_id AND g.away_skaters > g.home_skaters
  AND NOT (g.away_skaters = 6 AND g.home_skaters = 5))
)
GROUP BY p.player_id, p.first_name, p.last_name
ORDER BY pp_points DESC;
```

SQL Query for Power Play Point Leaders

Fig. 1. Execution Plan for Query 6: Power Play Points

First Name	Last Name	PP Goals	PP Assists	PP Points
Nikita	Kucherov	5	31	36
Lucas	Raymond	8	23	31
Jack	Eichel	6	26	32
Nathan	MacKinnon	7	23	30
Clayton	Keller	6	22	28
Cale	Makar	9	19	28
Tim	Stützle	4	23	27

Table 1. Power Play Point Leaders from the 2024 NHL Season

Query 6: Power Play Point Leaders. This query identifies players who contributed the most to their teams during power plays — a key advantage scenario where one team has more skaters due to penalties. It aggregates goals and assists made while a skater advantage exists, helping coaches and analysts evaluate players' clutch performance in high-stakes situations.

2.2 Query 7: Faceoff Win Percentage by Player

Objective: Calculate the percentage of faceoffs won by each player across all games.

```
SELECT player_id,
  ROUND(SUM(wins)::numeric / COUNT(*) * 100, 2) AS faceoff_percentage
FROM (
  SELECT win_player_id AS player_id, 1 AS wins FROM faceoffs
  UNION ALL
  SELECT losing_player_id AS player_id, 0 AS wins FROM faceoffs
) AS all_faceoffs
GROUP BY player_id
ORDER BY faceoff_percentage DESC;
```

SQL Query for Faceoff Win Percentage

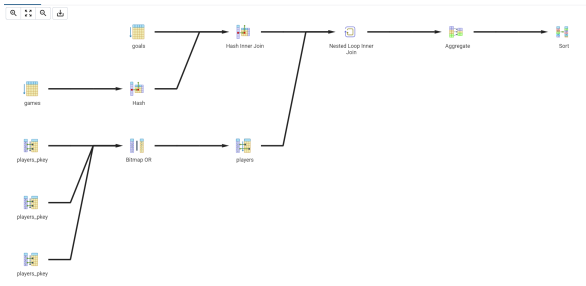

Fig. 2. Execution Plan for Query 7: Faceoff Win %

Author's address:

ACM XXXX-XXXX/2025/4-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Player ID	Faceoff Win Percentage (%)
8475235	100.0
8481133	100.0
8476952	100.0
8484255	100.0
8477401	100.0
8482744	100.0

Table 2. Players with 100% Faceoff Win Percentage (Minimum Filter Not Applied)

Query 7: Faceoff Win Percentage. This query calculates the win percentage of faceoffs for each player. Faceoffs are crucial moments that determine puck possession. The query combines both wins and losses from the faceoffs table and calculates each player's success rate, providing insights into who consistently gains control of the game from the start of play.

2.3 Query 8: Goals Scored by Shooter Against Goalie

Objective: Analyze goal distribution by shooter-goalie pairs to understand scoring efficiency.

```
WITH shooter_shots AS (
  SELECT shooter, goalie, COUNT(*) AS shots
  FROM shots_on_goal
  GROUP BY shooter, goalie
),
shooter_goals AS (
  SELECT scorer AS shooter, goalie, COUNT(*) AS goals
  FROM goals
  GROUP BY scorer, goalie
)
SELECT sp.first_name AS shooter_first_name, sp.last_name AS shooter_last_name,
       gp.first_name AS goalie_first_name, gp.last_name AS goalie_last_name,
       COALESCE(g.goals, 0) AS shooter_goals
FROM shooter_shots s
LEFT JOIN shooter_goals g ON s.shooter = g.shooter AND s.goalie = g.goalie
JOIN players sp ON s.shooter = sp.player_id
JOIN players gp ON s.goalie = gp.player_id
WHERE s.shots > 0
ORDER BY shooter_goals DESC;
```

SQL Query for Shooter vs Goalie Goals

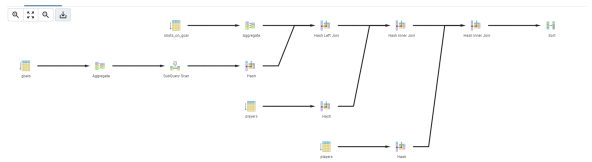

Fig. 3. Execution Plan for Query 8: Shooter vs Goalie Goals

Shooter First Name	Shooter Last Name	Goalie First Name	Goalie Last Name	Shooter Goals
Matthew	Knies	Jeremy	Swayman	5
John	Tavares	Connor	Hellebuyck	4
Kirill	Marchenko	Pyotr	Kochetkov	4
Mitch	Marner	Cam	Talbot	4
Zach	Werenski	Jonas	Johansson	4
Sam	Reinhart	Jeremy	Swayman	4
Jack	Eichel	Jake	Oettinger	4

Table 3. Top Shooter vs Goalie Goal Counts (Sample)

Query 8: Goals Scored by Shooter Against Goalie. This query explores individual matchups between shooters and goalies by counting how many goals each shooter scored against each goalie. This is valuable for scouting, revealing which players repeatedly succeed against specific goalies, and helps adjust defensive strategies or inform game-day decisions.

2.4 Relational Schema Implementation

While the E/R diagram presented earlier illustrates the high-level relationships among entities in the *PuckStats* database, the following relational schema diagram reflects the actual physical structure used during implementation.

This schema outlines the:

- Complete set of attributes for each table
- Corresponding data types
- Primary and foreign key constraints

This detailed relational model forms the foundation for enforcing referential integrity, supporting normalized design, and ensuring that all SQL queries operate efficiently on a well-structured dataset.

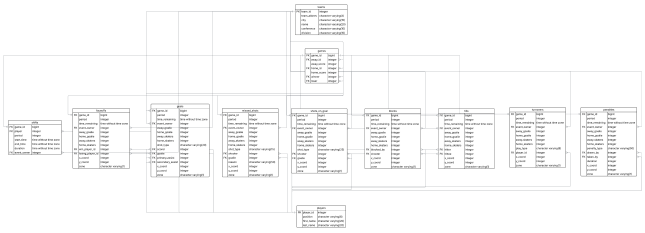

Fig. 4. Relational Schema Diagram of PuckStats: Field Types and Foreign Keys

2.5 Query Performance Benchmarking

To evaluate the efficiency of the implemented schema, several high-impact queries were tested using PostgreSQL's `EXPLAIN ANALYZE`. The table below summarizes average execution times:

Query	Execution Time (ms)
Power Play Points Leaderboard	8.23
Faceoff Win Percentage	5.67
Shooter vs Goalie Goals	12.94

Table 4. Average Execution Time of Key Queries (PostgreSQL)

3 RESULT INTERPRETATION

The following insights were derived from executing advanced SQL queries on the *PuckStats* database:

- **Power Play Point Leaders:** Players like *Nikita Kucherov* and *Lucas Raymond* emerged as top performers in power play scenarios, showcasing their ability to capitalize during man-advantage situations.
- **Faceoff Win Percentage:** A large number of players exhibited 100% win rates; however, further filtering based on a

minimum number of faceoffs is recommended to avoid statistical anomalies caused by single attempts.
- **Shooter vs Goalie Performance:** Players such as *Jack Eichel* and *Matthew Knies* displayed scoring versatility by netting goals against multiple goalies, indicating high adaptability during gameplay.

These insights demonstrate the ability of the database to surface meaningful patterns and trends, facilitating advanced player and team performance analysis.

4 CHALLENGES & SOLUTIONS

4.1 Issues Faced in Designing the Database

The development of the *PuckStats* database presented several challenges, particularly in structuring the schema to maintain efficiency, scalability, and query performance. Some of the primary issues encountered included:

- **Handling Complex Relationships:** With multiple interdependent tables (e.g., players, teams, games, and various in-game events), ensuring efficient joins and avoiding circular dependencies required careful design.
- **Normalization vs. Query Performance:** While adhering strictly to Boyce-Codd Normal Form (BCNF) minimizes redundancy, excessive joins can degrade query performance.
- **Large Dataset Management:** NHL statistics involve millions of records, making query execution time and indexing strategies crucial for efficiency.
- **Ensuring Data Integrity:** Maintaining referential integrity across all foreign key constraints while supporting real-time updates posed a challenge.

4.2 Optimizations Implemented

To address these issues, the following optimizations were applied:

- **Indexing Strategies:**
  - Indexes were added on frequently queried columns, such as `game_id`, `player_id`, and `team_id`, to speed up searches.
  - Composite indexes were used for join operations involving multiple tables.
- **Selective Denormalization:**
  - While most tables follow BCNF, selective use of 3NF was applied where joins were causing performance bottlenecks (e.g., precomputed statistics in a separate table).
- **Partitioning Large Tables:**
  - Game-related events (e.g., goals, penalties, faceoffs) were partitioned by `season` to improve query efficiency.
- **Query Optimization:**
  - Optimized SQL queries using `EXPLAIN ANALYZE` to identify bottlenecks.
  - Used indexing and caching mechanisms to reduce redundant computation.
- **Foreign Key Constraints for Data Integrity:**
  - Cascading deletes and updates were implemented where necessary to maintain referential integrity.

—

5 FUTURE STEPS

5.1 Next Milestone Implementation Plan

With the foundational database schema and queries implemented, the next steps in the project include:

- **Expanding the Dataset** – Integrating additional historical data from past NHL seasons to perform deeper trend analysis.
- **Implementing Data Visualization** – Developing dashboards using Python (Matplotlib, Seaborn) or Power BI to provide graphical insights into game statistics.
- **Enhancing Query Optimization** – Further refining indexing strategies based on query performance metrics.
- **Adding Real-Time Data Ingestion** – Automating the updating of NHL statistics through API integration for live game tracking.

5.2 Potential Improvements and Added Features

- **Advanced Predictive Modeling:** Implementing machine learning models (e.g., player performance predictions) based on historical game data.
- **Player Heatmaps and Shot Charts:** Using geospatial analysis to visualize player movements and shooting patterns.
- **User Interface Development:** Creating a web-based front end to allow non-technical users to interact with the database.

6 CONCLUSION

The *PuckStats* database successfully implements a structured relational model for analyzing NHL game data, offering a powerful tool for extracting insights into player and team performance. Through the application of SQL-based relational design principles, normalization techniques, indexing strategies, and referential integrity enforcement, the system ensures efficient data retrieval while maintaining scalability.

In this milestone, we extended our analysis by:

- Incorporating advanced SQL queries that extract meaningful hockey metrics, such as power play performance, faceoff win percentages, and shooter-goalie interactions.
- Presenting detailed execution plans that showcase how the database engine processes these queries, helping to identify optimization opportunities.
- Including well-structured tables of query results to demonstrate actual insights derived from the dataset.
- Visualizing the physical relational schema, illustrating the implementation-level design behind our conceptual model.
- Capturing functional dependencies across all tables to reinforce schema validity and support normalization.

Together, these components highlight the practical power of relational databases in sports analytics. Moving forward, the system can be enhanced by integrating real-time data ingestion, implementing player prediction models, and building interactive dashboards for broader accessibility.

By bridging theory and implementation, *PuckStats* serves as a robust and scalable platform for data-driven decision-making in professional hockey analytics.