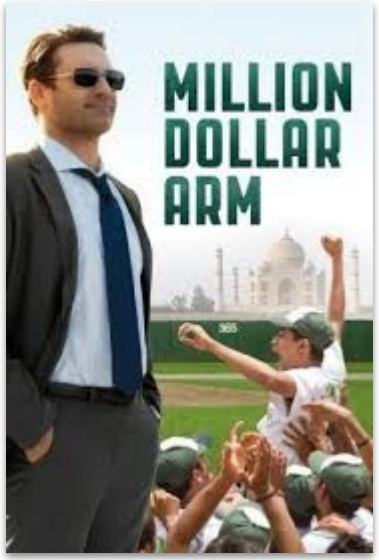
Graph your “Game” with *Go*

“**Strategic** *playbooks* unveiled”

Why the Drawing Board is Important!

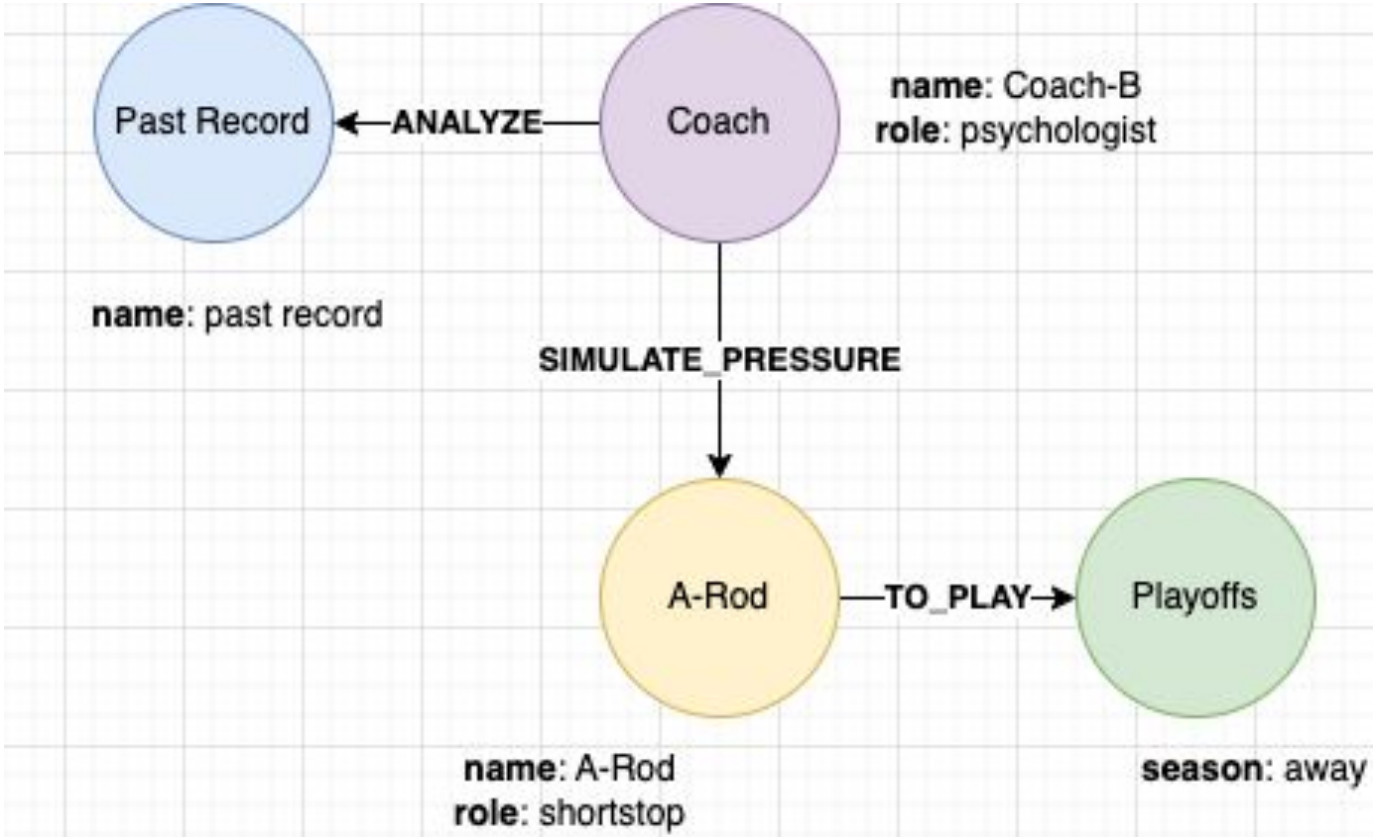


Case 1: Player Focus

Data Insight: Player X excels in games won by **large margins** but underperforms in close games.

Strategic Question: What coaching strategies can be implemented to enhance Player X's performance under **high-pressure** conditions?

A Graph Says A Million Words

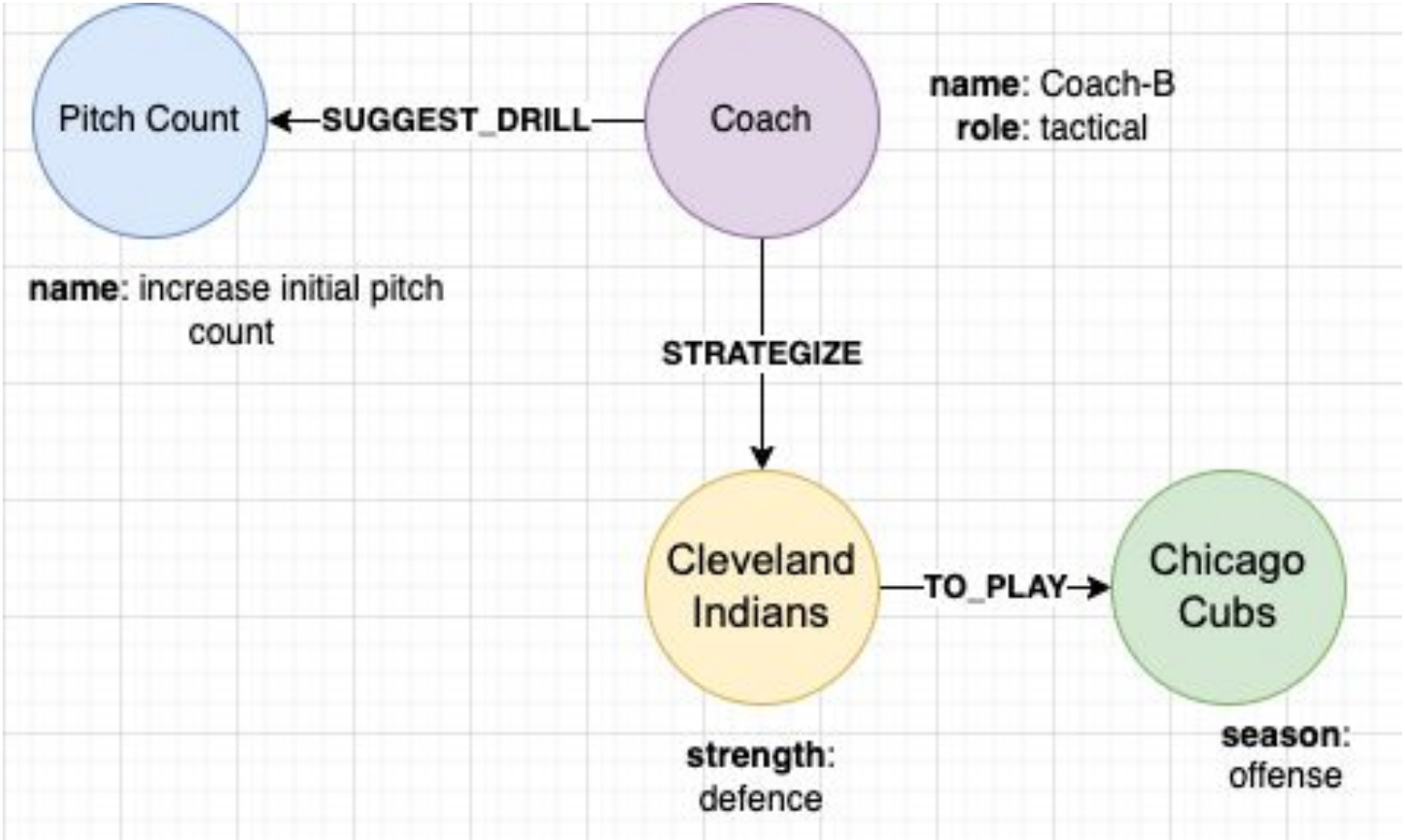


Case 2: Team Focus

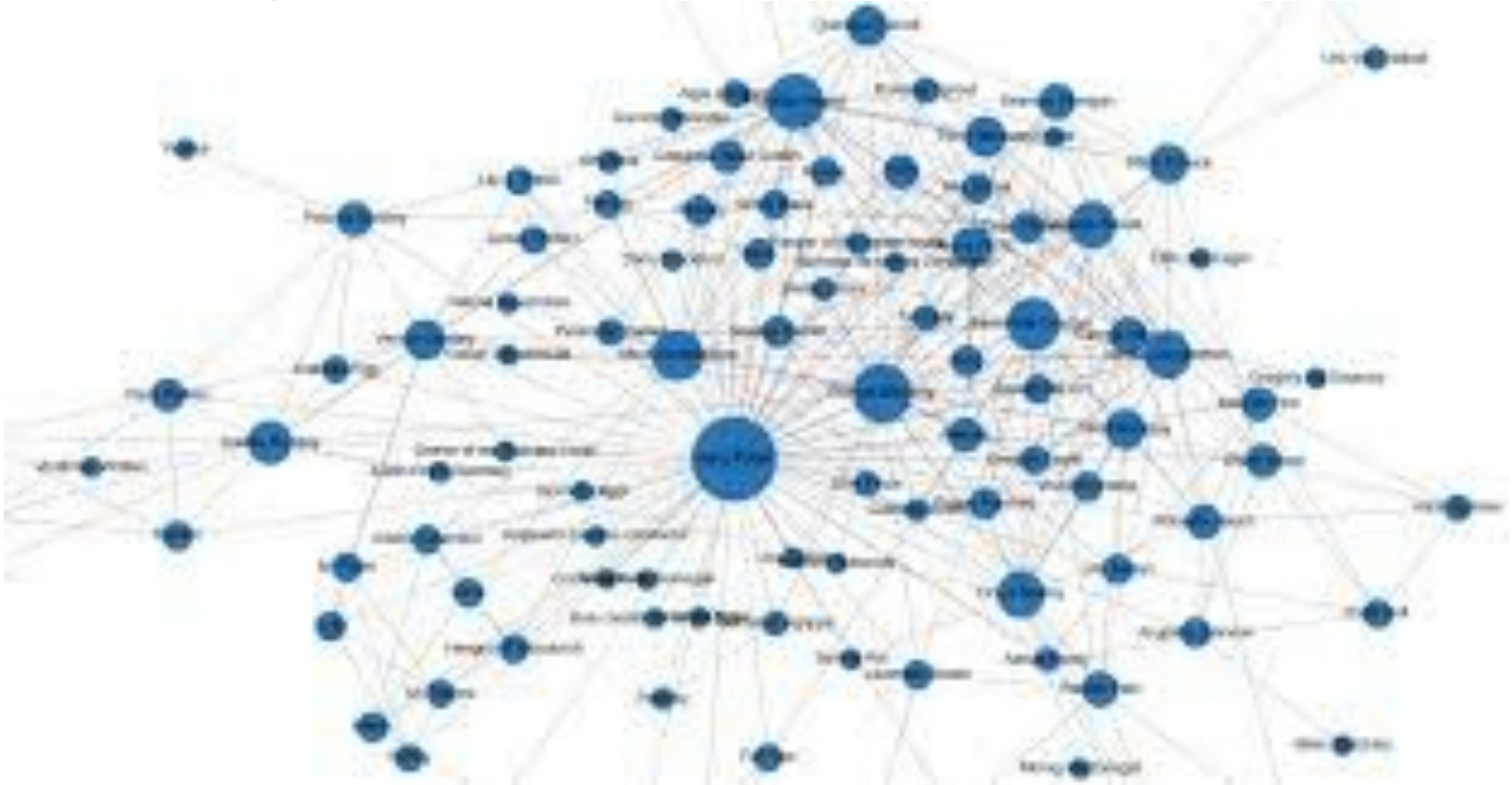
Data Insight: Our team consistently struggles against teams with robust **defensive** tactics.

Strategic Question: What offensive adjustments can we make to improve our competitive edge against defensively strong teams?

A Graph Speaks A Million Words



Knowledge Graphs

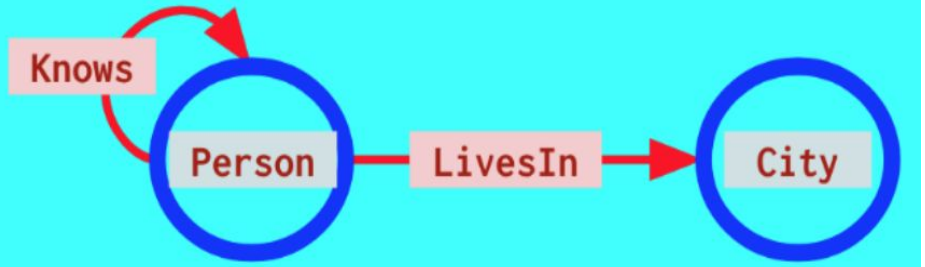


Why CypherQL?

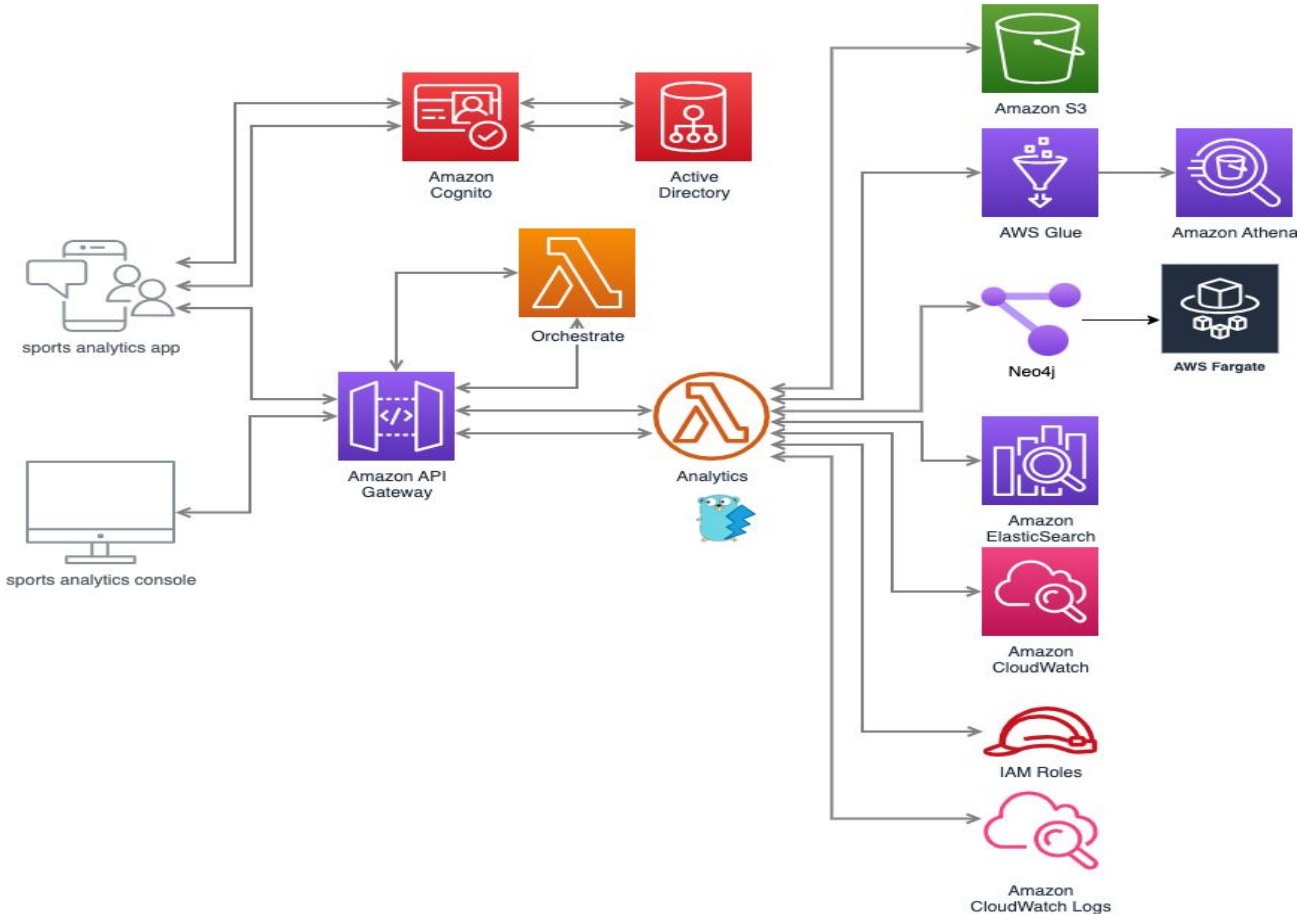
Cypher query language depicts patterns of nodes and relationships and filters those patterns based on labels and properties. Cypher's syntax is based on ASCII art, which is **text-based visual art** for computers. This makes the language **very visual and easy to read** because it both **visually and structurally represents the data** specified in the query.

```
CREATE GRAPH SocialNetwork {
    (Person {name STRING, dob DATE}),
    (City {name STRING}),

    (Person)-[LivesIn]->(City),
    (Person)-[Knows]->(Person)
}
```



10,000 Feet


Amazon S3
Amazon Cognito
Active Directory
AWS Glue
Amazon Athena
Orchestrate
sports analytics app
Neo4j
AWS Fargate
Amazon API Gateway
Analytics
Amazon ElasticSearch
Amazon CloudWatch
sports analytics console
IAM Roles
Amazon CloudWatch Logs

Ground Level - Neo4j Model

```
// Create Teams
CREATE (cubs:Team {name: "Chicago Cubs"})
CREATE (indians:Team {name: "Cleveland Indians"})
CREATE (yankees:Team {name: "New York Yankees"})

// Create Players
CREATE (aRod:Player {name: "Alex Rodriguez", team: "Yankees"})
CREATE (kluber:Player {name: "Corey Kluber", team: "Indians"})
CREATE (griffey:Player {name: "Ken Griffey Jr", team: "Free Agent"})

// Create Games
CREATE (worldSeries2016:Game {name: "2016 World Series", type: "Playoff"})
CREATE (playoffGame:Game {name: "Playoff Game", type: "Playoff"})

// Create Relationships
CREATE (aRod)-[:PLAYED_IN {role: "batter", performanceRating: "Low"}]->(playoffGame)
CREATE (kluber)-[:PLAYED_IN {role: "pitcher", performanceRating: "High"}]->(worldSeries2016)
CREATE (griffey)-[:PLAYED_IN {role: "center_field", performanceRating: "High"}]->(playoffGame)
CREATE (cubs)-[:COMPETED_IN]->(worldSeries2016)
CREATE (indians)-[:COMPETED_IN]->(worldSeries2016)
```

Ground Level Golang Awesomeness - Connect

```go
func main() {
    dbUri := "neo4j://localhost" // scheme://host(:port) (default port is 7687)
    driver, err := neo4j.NewDriverWithContext(dbUri, neo4j.BasicAuth("neo4j", "letmein!", ""))
    if err != nil {
        panic(err)
    }
    // Starting with 5.0, you can control the execution of most driver APIs
    // To keep things simple, we create here a never-cancelling context
    // Read https://pkg.go.dev/context to learn more about contexts
    ctx := context.Background()
    // Handle driver lifetime based on your application lifetime requirements.
    // driver's lifetime is usually bound by the application lifetime, which usually implies o
    // application

    defer driver.Close(ctx) // Make sure to handle errors during deferred calls
    item, err := insertItem(ctx, driver)
    if err != nil {
        panic(err)
    }
    fmt.Printf("%v\n", item)
}
```

Ground Level - Golang Awesomeness - Insert

```go
func insertItem(ctx context.Context, driver neo4j.DriverWithContext) (*Item, error) {
    result, err := neo4j.ExecuteQuery(ctx, driver,
        "CREATE (n:Item { id: $id, name: $name }) RETURN n",
        map[string]any{
            "id":   1,
            "name": "Item 1",
        }, neo4j.EagerResultTransformer)
    if err != nil {
        return nil, err
    }
    itemNode, _, err := neo4j.GetRecordValue[neo4j.Node](result.Records[0], "n")
    if err != nil {
        return nil, fmt.Errorf("could not find node n")
    }
    id, err := neo4j.GetProperty[int64](itemNode, "id")
    if err != nil {
        return nil, err
    }
    name, err := neo4j.GetProperty[string](itemNode, "name")
    if err != nil {
        return nil, err
    }
    return &Item{Id: id, Name: name}, nil
}
```

Ground Level - Go In Lacrosse

```
CREATE (:Player {name: 'Player A', position: 'Attack'})
CREATE (:Player {name: 'Player B', position: 'Midfield'})
CREATE (:Player {name: 'Player C', position: 'Defense'})
CREATE (:Player {name: 'Player D', position: 'Goalkeeper'})
CREATE (:Game {date: '2023-04-01', opponent: 'Team X'})


### Low Submarine Shots and Rainbow Passes
MATCH (p1:Player {name: 'Player A'}), (g:Game {date: '2023-04-01'})
CREATE (p1)-[:LOW_SUBMARINE_SHOT {outcome: 'goal'}]->(g)

MATCH (p2:Player {name: 'Player B'}), (g:Game {date: '2023-04-01'})
CREATE (p2)-[:RAINBOW_PASS {outcome: 'complete'}]->(p1)

### Faceoffs, Turnovers, and Saves
MATCH (p3:Player {name: 'Player C'}), (g:Game {date: '2023-04-01'})
CREATE (p3)-[:FACEOFF_WON]->(g)

MATCH (p1:Player {name: 'Player A'}), (g:Game {date: '2023-04-01'})
CREATE (p1)-[:TURNOVER {cause: 'interception'}]->(g)

MATCH (p4:Player {name: 'Player D'}), (g:Game {date: '2023-04-01'})
CREATE (p4)-[:SAVE {shotBy: 'Opponent Player'}]->(g)
```

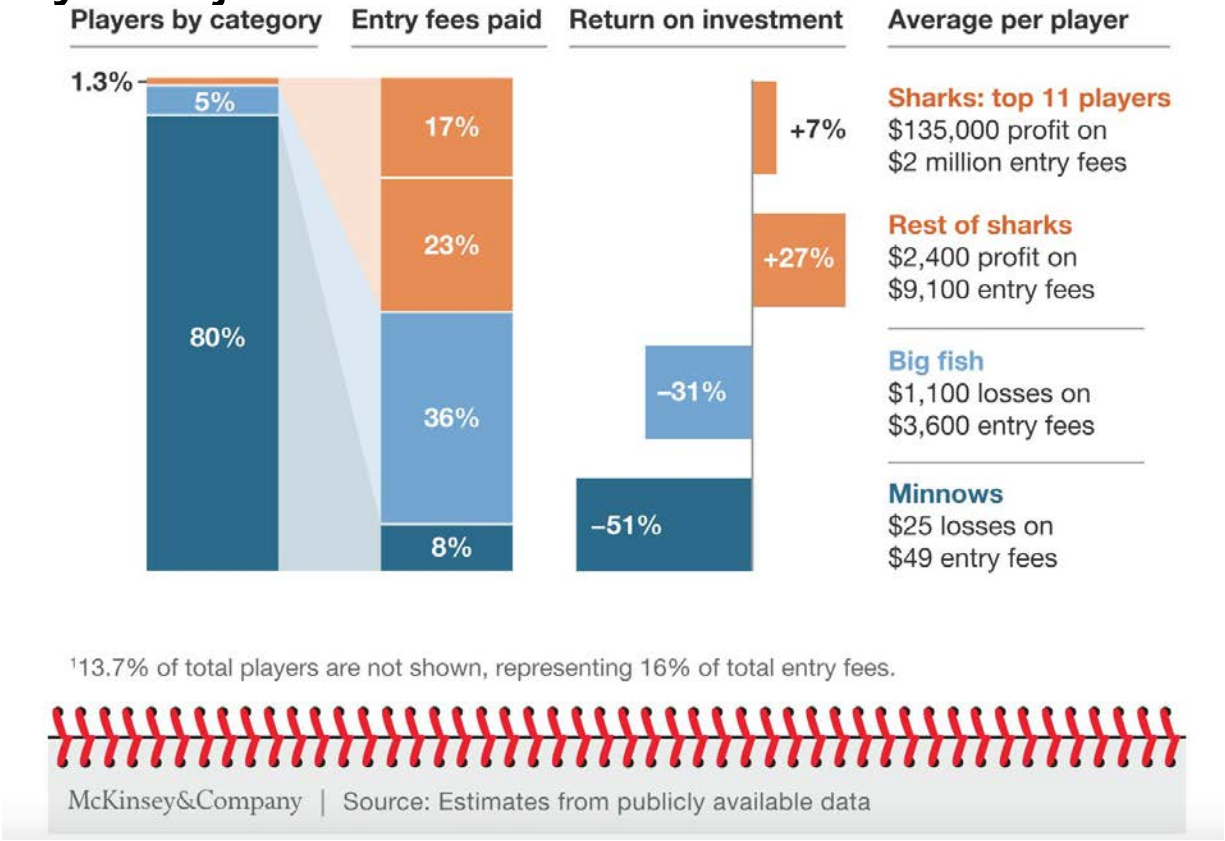
Teaser - Cosine Similarity

```
27        // Example: Calculate similarity between two players
28        result, err := session.Run("MATCH (p1:Player)-[:PLAYED_AGAINST]-(p2:Player) " +
29            "RETURN p1.name AS player1, p2.name AS player2, " +
30            "gds.alpha.similarity.pearson([p1.homeRuns, p1.battingAverage], [p2.homeRuns, p2.battingAverage]) AS similarity", nil)
31        if err != nil {
32            log.Fatal("Error querying similarity:", err)
33        }
34
35        for result.Next() {
36            fmt.Printf("Player 1: %s, Player 2: %s, Similarity: %f\n", result.Record().GetByIndex(0), result.Record().GetByIndex(1), result.Record().GetByIndex(2))
37        }
38
39        if err := result.Err(); err != nil {
40            log.Fatal(err)
41        }
42
43        // Example: Aggregate team success (simplified example)
44        // Assuming success is measured by the sum of home runs
45        aggResult, err := session.Run("MATCH (p:Player)-[:PLAYS_FOR]->(t:Team) " +
46            "RETURN t.name AS team, sum(p.homeRuns) AS totalHomeRuns", nil)
47        if err != nil {
48            log.Fatal("Error querying team success:", err)
49        }
```

Lot More


kendall_rank_correlation
spearman's_rank_correlation
hamming_distance
dot_product
jaccard_index
pearson_correlation
euclidean_distance
levenshtein_distance
mahalanobis_distance
manhattan_distance

Beyond just numbers



Links

https://github.com/neo4j/neo4j-go-drive {{boilerplate}}

https://github.com/rangarajl/graph-and-go/blob/main/model {{model}}

https://github.com/rangarajl/graph-and-go/blob/main/similarity.go {{query}}

Retrospective - Takeaways

- ❏ Blueprint - Thought provoker
- ❏ Indirect **Emphasis** on using a particular tech stack
- ❏ Hopefully inspire adoption to Data, AI
- ❏ Maybe Tiny Graphs - Who Knows?