

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from the bar, containing the date.

11/1/2017

SCS2109 Database 2

Assignment 3 - Polyglot Persistence

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Wasura Dananjith

15001415
2015/CS/141

Table of Contents

Introduction	2
Main Assumptions	2
Risks	2
Factors to be considered when designing the Database	2
Component Structure	3
Components.....	3
1. Manage User Sessions (Access User Profiles & Preferences)	3
Expected Features.....	3
Database Type.....	3
Physical Implementation.....	3
Data Model	4
2. Cricketer and Nation Profiles Management	4
Expected Features.....	4
Database Types	4
Physical Implementation.....	4
Data Model - RDBMS (Table Architecture)	5
3. Cricketer Records Management	6
Expected Features.....	6
Database Type.....	6
Physical Implementation.....	6
Data Model	6
4. Team and Match Records Management.....	7
Expected Features.....	7
Database Type.....	7
Physical Implementation.....	7
Data Model	7
Glossary.....	8
Acronyms	8
References.....	8

Introduction

This document will only describe the database design of “Cricket At Your Finger Tip” system.

Profile Information about Cricketers and Cricket Nations will be stored here, which includes text, image etc. (See Assumptions for more details)

Also, details about the performance of cricketers, data needed to do analytics to predict the scorers, information to do player selection will be stored too. All the records related to Cricket matches will be stored in the system throughout the years.

System will be depicted by 4 main components, and those will be discussed here. Polyglot Persistence plays a huge role here. Designing database is a risky task because of the large amount of data which system stores.

Main Assumptions

- Size of the database will be larger and larger by the time, but assume this is not a problem when considering about big data.
- Concurrent Transactions will take place, like update scores during matches etc. Assumed that it will not have a large impact on the system.
- Data validation will be mainly done by the front-end.
- Data can be only added by the organization which this database belongs.
- Confidential information about users and players will not be recorded.

Risks

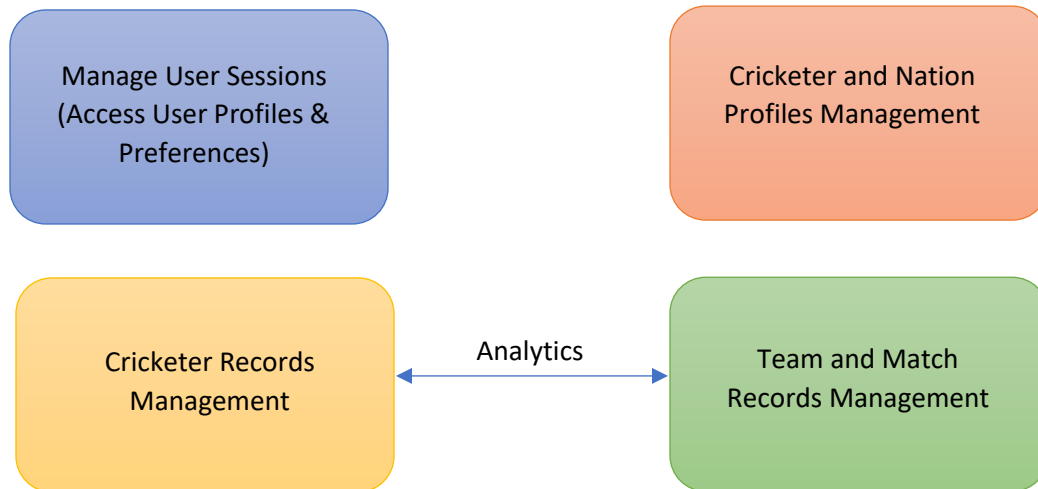
- Since this is a distributed system, Network infrastructure may lead to several problems such as loss of data in the transmission.
- Can be hard to troubleshoot and diagnose the system in case of a failure.

Factors to be considered when designing the Database

- Large amount of data may require more storage space, servers. (Sharding)
- Should be scalable.
- Concurrent access and transaction management.
- Efficient CRUD operations.
- Accuracy in analytics.

- Database should be reliable such that even if some components in the system fail, users must be able to use the rest of the components.

Component Structure



Components

1. Manage User Sessions (Access User Profiles & Preferences)

Expected Features

- Rapid access for reads and writes.
- No need to be durable.
 - Session length is not persistent data
- Frequent Reads and Writes.
 - Update Last Login Time when login
 - Flush data when logout (if needed)
- Needs fast access.
 - Retrieve password for the particular username when login

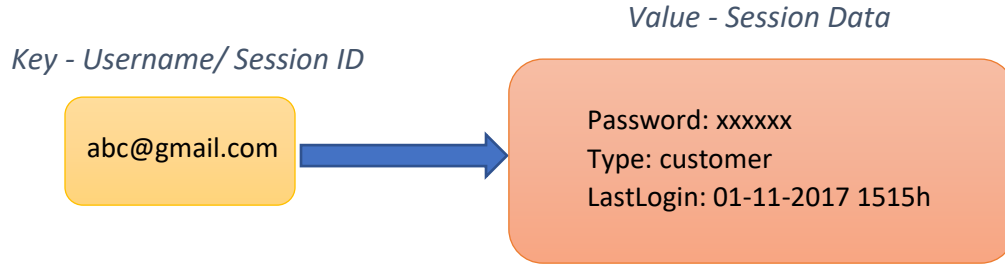
Database Type

- Key-Value Database

Physical Implementation

- Redis
- Riak

Data Model



2. Cricketer and Nation Profiles Management

Expected Features

- Stores long term data about cricketers and nations.
- Data will be changed infrequently.
 - No need to change personal details frequently (Name, DOB, Batting Style etc.)
 - Details about nations will not be changed frequently.
- Keep the profiles as data can be kept consistent at all.

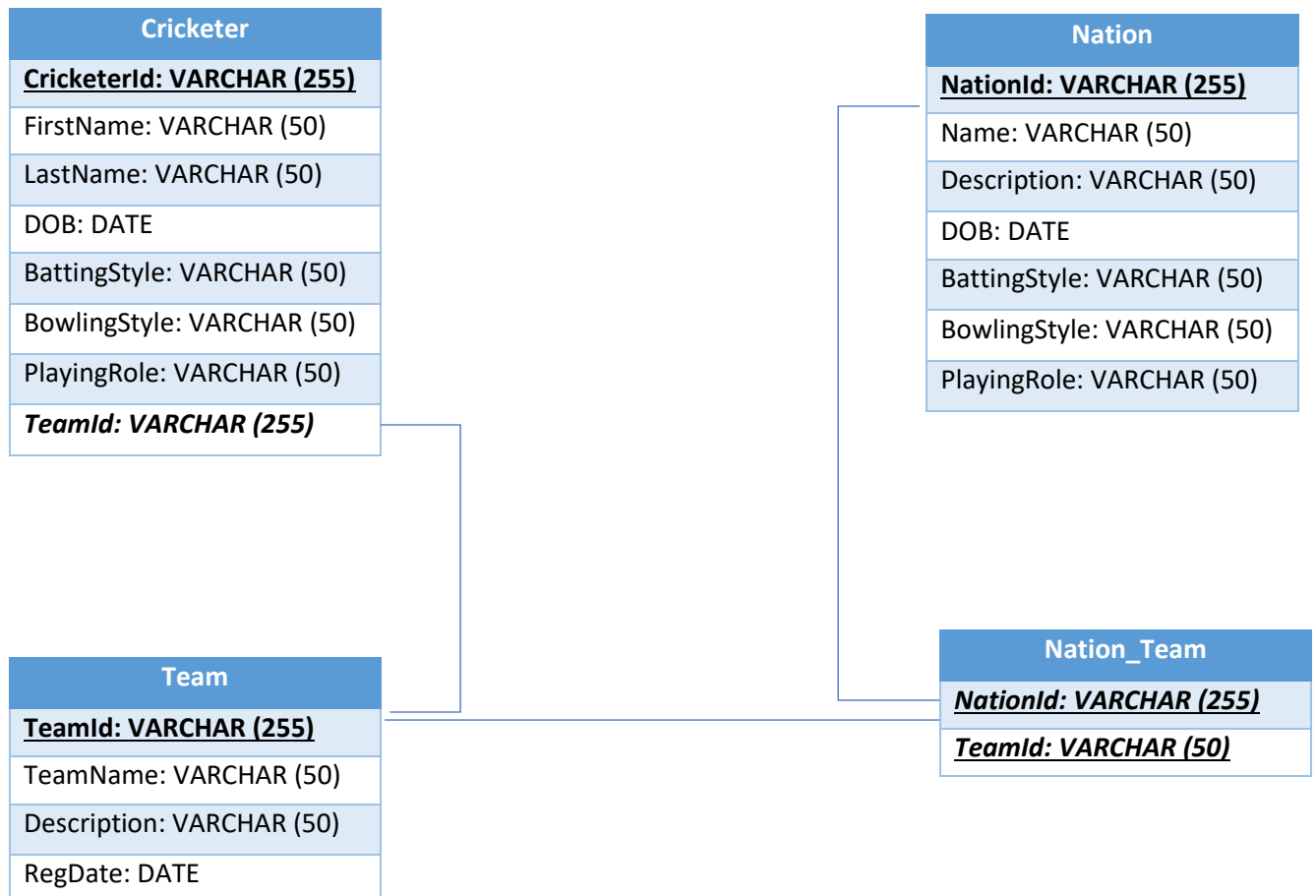
Database Types

- RDBMS
- or
- Document Database

Physical Implementation

- MySQL (RDBMS)
- or
- MongoDB (Document Database)

Data Model - RDBMS (Table Architecture)



- **Primary Keys are Bold Underlined**
- **Foreign Keys are in Bold Italics Underlined**

Assumptions

- One Cricketer belongs to one Team only.
- A Nation can have one or more Teams.
- Above diagram depicts just a sketch with some basic attributes. There can be more attributes to be considered.

3. Cricketer Records Management

Expected Features

- Need to maintain player records regarding different cricket matches - Scores, Inning Level etc.
- Need to facilitate large scale analytics.
- Column Database is preferred
 - Column structure will allow access to the same types of data efficiently
 - Ideal to be distributed across multiple servers to increase performance and assure the integrity in case of a failure
- **Score prediction and Player Selection is mainly done by this component.**

Database Type

- Column Family Database

Physical Implementation

- Cassandra

Data Model

Super Column Family

Column Name	Key	Value
Match	InningCount	InningDetails
Cricketer	CricketerId	Scores

Scores Family (inside the Cricketer Scores)

Column Name	Key	Value
Batting	MatchId	Description
Bowling	MatchId	Description

4. Team and Match Records Management

Expected Features

- Many complex relationships between the players who took part in a Match (Game), Venue, Time etc.
- Need efficient access through shortest paths to the required information. (Fast Traversal)
- **Player selection can be done here too along with the previous component.**

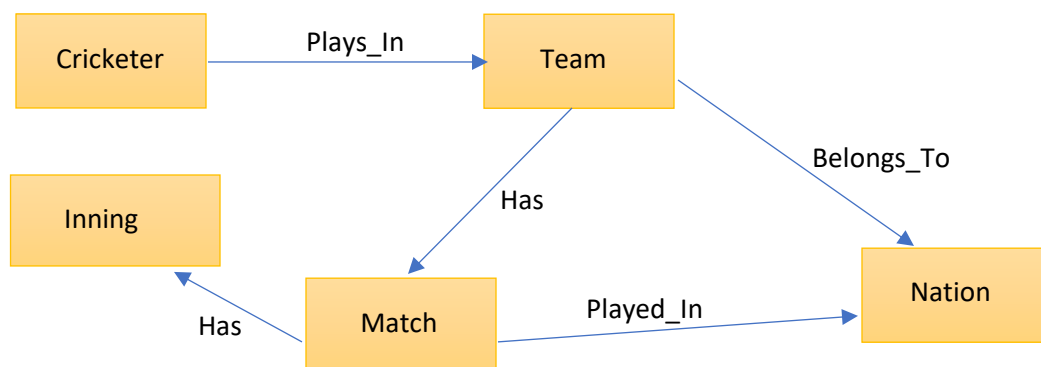
Database Type

- Graph Database

Physical Implementation

- Neo4j

Data Model



Glossary

- **Concurrent** - Existing, happening, or done at the same time.
- **Polyglot Persistence** - Polyglot Persistence is a fancy term to mean that when storing data, it is best to use multiple data storage technologies, chosen based upon the way data is being used by individual applications or components of a single application.
- **Sharding** - Sharding is a type of database partitioning that separates very large databases into smaller, faster, more easily managed parts called data shards. The word shard means a small part of a whole.
- **Transaction** - A transaction is a very small unit of a program and it may contain several lowlevel tasks. A transaction in a database system must maintain Atomicity, Consistency, Isolation, and Durability – commonly known as ACID properties – in order to ensure accuracy, completeness, and data integrity.

Acronyms

- **NoSQL** - Not only SQL
- **RDBMS** - Relational Database Management System
- **SQL** - Structured Query language

References

- *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence [2012]*, by Martin Fowler and Pramod Sadalage
- <http://www.espnrcricinfo.com>
- <https://martinfowler.com/articles/nosql-intro-original.pdf>
- <http://www.jamesserra.com/archive/2015/07/what-is-polyglot-persistence>
- <http://www.jamesserra.com/archive/2015/04/types-of-nosql-databases>