**Introduction (section)**

Recap

**Database type (section)**

Database type

How to compare databases?

Literature Survey

* Comparative study: Relational, Columnar, Graph

How we compare databases

Relational Graph databases

Graph databases

Distributed databases

Which is best and conclusion

**Schema (section)**

**Conclusion (section)**

Introduction  **(section)**

**Slide 1: Recap**

* "Sanskrit as a Language is Complex”
* Existing Sanskrit wordnet has limited number of words and very few verbs
* Revolutionizing Sanskrit Language Understanding by making an online version of the Verb thesaurus written by Manoj VR
* Build a schema for the data
* Compare different kinds of database for our task

**Database type (section)**

**Slide 2: Database type**

* Comparing databases
  + What kind of databases are worth exploring?
  + What metrics to track from these databases?

**Slide 3: A comparison between several NoSQL databases**

* Compares several NoSQL databases, but it does not explicitly list the specific databases being compared
* Mainly focuses on the comparison process and the features of the databases
* Qualitative and quantitative evaluations of the NoSQL databases
* Qualitative evaluations
  + persistence, replication, high availability, transactions, rack-locality awareness, implementation language, influences/sponsors, and license type
* Quantitative evaluations
  + Size of data
  + Results are not mentioned in the paper

**Slide 4: A performance comparison of SQL and NoSQL databases**

* investigate the performance of key-value stores implemented by both SQL and NoSQL databases
* Metrics compared:
  + Time taken to instantiate a database bucket.
  + Time taken to read values corresponding to given keys from the bucket
  + Time taken to write key value pairs to the bucket
  + Time taken to fetch all the keys in the bucket
* No observable correlation between the data model and performance
* Read performance of SQL Express was found to be better than some, but not all, of the NoSQL databases.

**Slide 5: What type of database?**

* Compare different kinds of databases for our use case
* Compare between
  + Relational databases (eg. SQL)
  + Graph databases (eg. Neo4J)
  + Distributed Nested Relational databases (eg. Pig on hdfs)
* Use english wordnet data to compare as we don’t have Sanskrit data

**Slide 6: What type of database?**

* Could not find dataset or database info for Princeton English Wordnet (1995)
* Used text files as a data source
* Open English Wordnet (<https://en-word.net>) is an open-source alternative with more modern words and publicly available code + data
* Photo(website)

**Slide 7: Open English Wordnet**

* Fast querying (photo)
* Data is stored as in-memory struct in Rust programming language
* Choice of language is also important. Same approach in javascript or python would be slow

**Slides 8: Data format**

* Show json

**Slide 9: Metrics to track**

* Metrics to track
  + No. of nodes
  + No. of relationships between nodes
  + Time taken to add nodes
  + Time taken to add relationships
  + Time taken to query a single node
  + Time to query children nodes of a node for up to depth N

**Slide 10: Relational databases (eg. SQL)**

**Slide 11: Graph databases (eg. Neo4J)**

**Slide 12: Distributed Nested Relational databases (eg. Pig on hdfs)**

**Slide 13: Which is best?**

* SQL performs well with respect to loading data and relationships
* Time taken to query nodes and it's children to a depth of N is exponential on N and this makes it
* Graph databases are very slow at losing the relationships and making the graph
* Querying upto 2000 children of a node only takes 450ms on a graph database
* Distributed Nested Relational databases are faster than SQL at loading data but very slow in querying

**Schema (section)**

**Slide 14: Schema**

* Show schema

**Conclusion (section)**

**Slide 15: Conclusion and Future Work**

* FInd efficient ways to put data
* Check if some database does it better using parallelism
* Use distributed and graphs together?

## 