Tiles and Non-Relational Models

Shreeraksha Achutha
Taranpreet Kaur
Madhuri Nagaraj Hebbale
Sindhuja Madabushi

Supervisor: Gabriel Campero
May 2nd, 2017

Structure

1. Team Presentation

2. Topic Introduction

- Relational models
- Introduction to non relational models
- Types of non relational models
- Tile architecture
- Project goal

3. Project Schedule

Team Presentations – Roles (flexible roles)

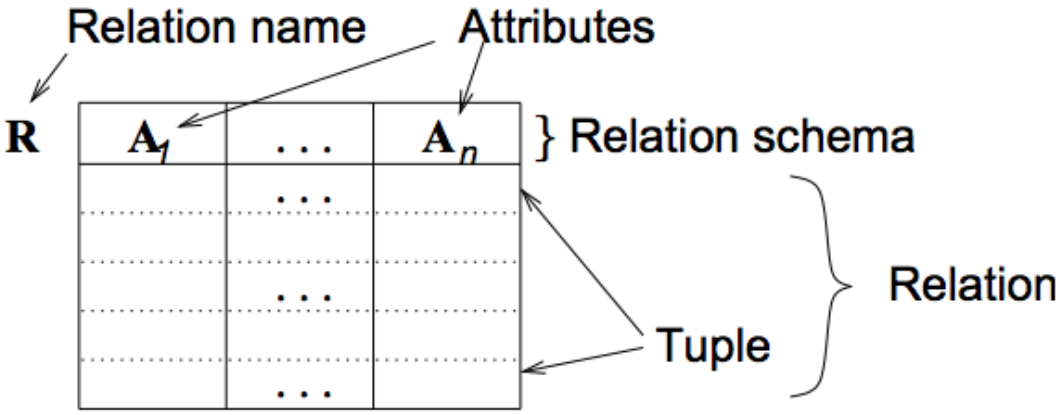
Shreeraksha Achutha
Program: Data and Knowledge Engineering
Main tasks: Literature research, paper presentation, prototypical implementation

Taranpreet Kaur
Program: Digital Engineering
Main tasks: Literature research, paper presentation, prototypical implementation

Madhuri Nagaraj Hebbale
Program: Data and Knowledge Engineering
Main tasks: Literature research, paper presentation, prototypical implementation

Sindhuja Madabushi
Program: Data and Knowledge Engineering
Main tasks: Literature research, paper presentation, prototypical implementation

Relational Models



Dr.-Ing. Eike Schallehn, Advanced Database Models lecture slides, OvG Universität Magdeburg, Fakultät für Informatik, Institut für Technische und Betriebliche Informationssysteme, 2017

- A database consists of set of tables.
- Each table has a predefined name and a set of predefined column names(attribute names).
- Rows are tuples of values that adhere to the predefined *schema*.
- Each row has one or more attributes known as relational key which can identify the row in the relation.
- Definition of attribute names for set of tables is called *relational schema.*
- Relational models come under *structured data models.*

Relational Models

Weaknesses of relational data model[WIE16]:

- **Inadequate representation of data for certain use cases:** Data is more complex in real world and and relational model is not best suited in some cases like key word search, etc.

- **Semantic Overloading:** Both entities and relationships are contained relation schemas. That is, there is no mechanism for differentiation between entities and relationships.

- **Weak support for recursion:** Recursive query is used to compute the transitive closure of some table attributes. Although relational algebra can be extended by unary transitive closure operation, expressing recursion in the query, it is still costly to compute in real RDBMS.

- **Homogeneity:** Fixed schema is a constraint, for relational model, we need schema before data.

Introduction to Non-Relational Data Models

- NoSQL databases are used for storage and retrieval of data that is modelled in means other than tabular relations.
- These are semi structured or unstructured data models.
- Data can be defined in the form of documents (XML, JSON), graphs, key-value pairs, etc.
- NoSQL databases have schema flexibility (generally).
- Used for managing persistent data from data sources like World Wide Web for which traditional RDBMS is not well suited.
- No Normalization is required for data.

Types of Non-Relational Databases

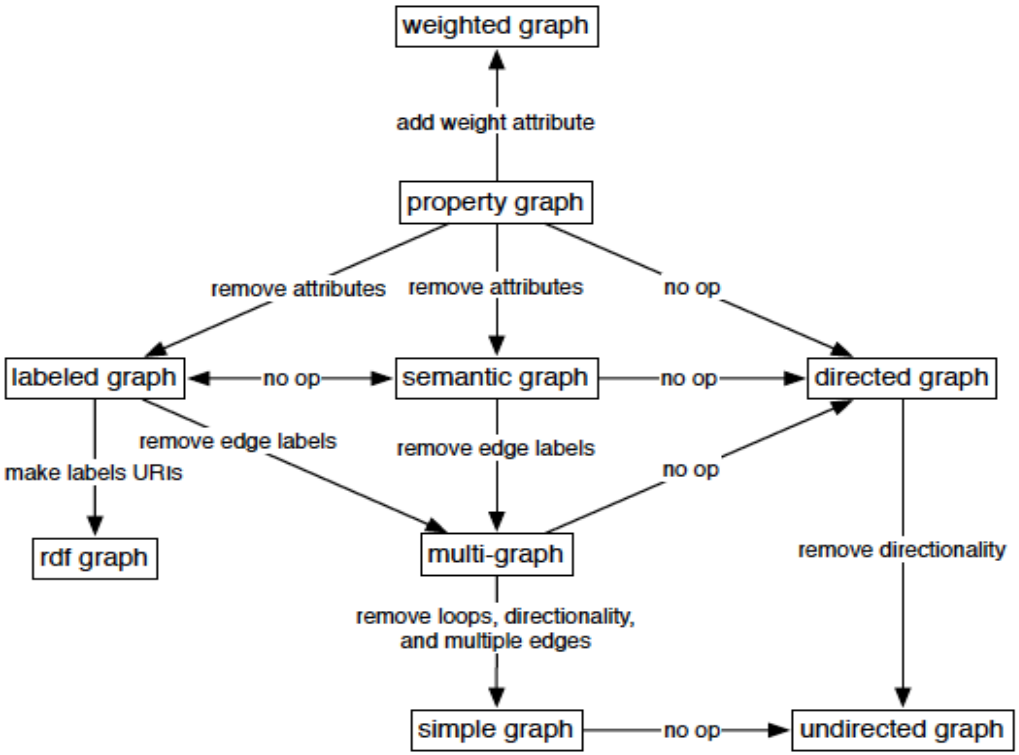
- Graph Databases
- Key Value Stores
- Document stores
- Extensible record stores

And many more…

Types of Non-Relational Databases

Graph Databases:

- Optimized for efficient processing of dense, interrelated datasets.
- Common types of graphs supported by the graph system are directed, labeled, attributed, multi-graphs known as property graphs.



Rodriguez, Marko A., and Peter Neubauer. "Constructions from dots and lines." *Bulletin of the American Society for Information Science and Technology* 36.6 (2010): 35-41.

Types of Non-Relational Databases

Key Value stores:

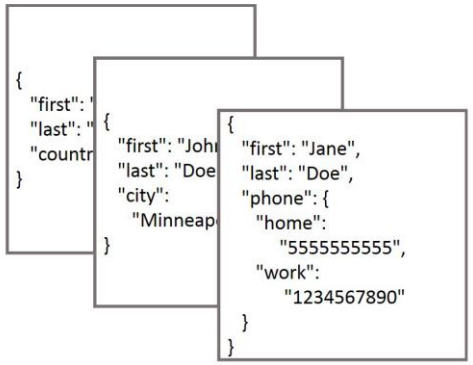
- Schema less collection of key value pairs.
- Eg. Redis, Oracle NoSQL.

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

https://en.wikipedia.org/wiki/Key-value_database

Document Stores:

- Documents are the basic unit of data.
- Documents can comprise of scalar values, lists and nested documents.
- Eg. MongoDB



http://greachconf.com/speakers/jennifer-strater-no-nonsense-nosql/

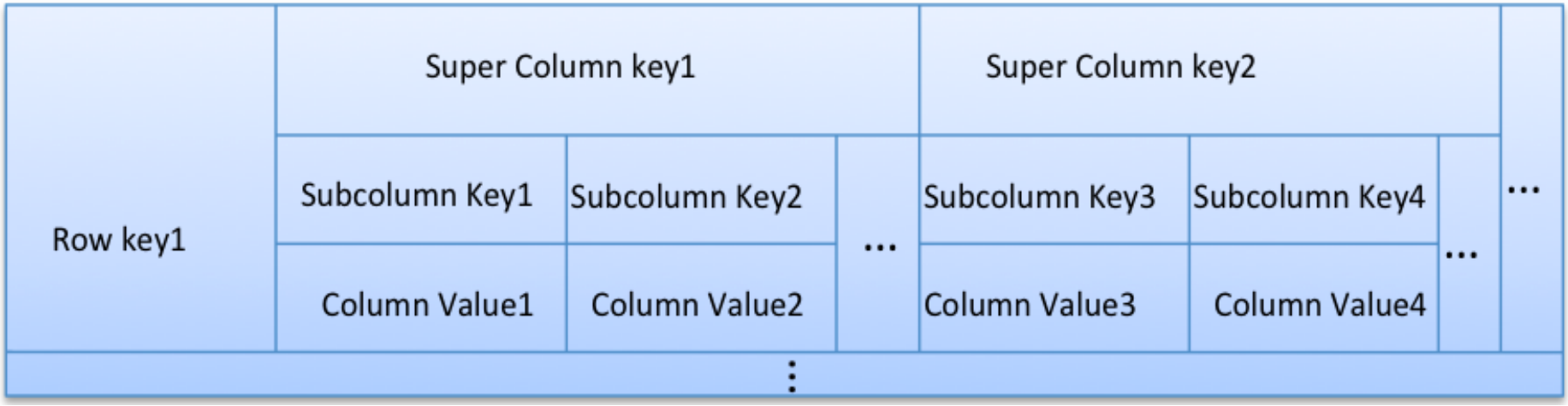
Types of Non-Relational Databases

Extensible Record Stores:

This system stores tables of extensible records that can be partitioned across multiple nodes.

Hybrid between tuple in relational DB and document in document store.

Eg. Amazon DynamoDB, Google BigTable, Apache Cassandra



http://www.ebaytechblog.com/2012/07/16/cassandra-data-modeling-best-practices-part-1/

Tile Based Architectures for Data layout

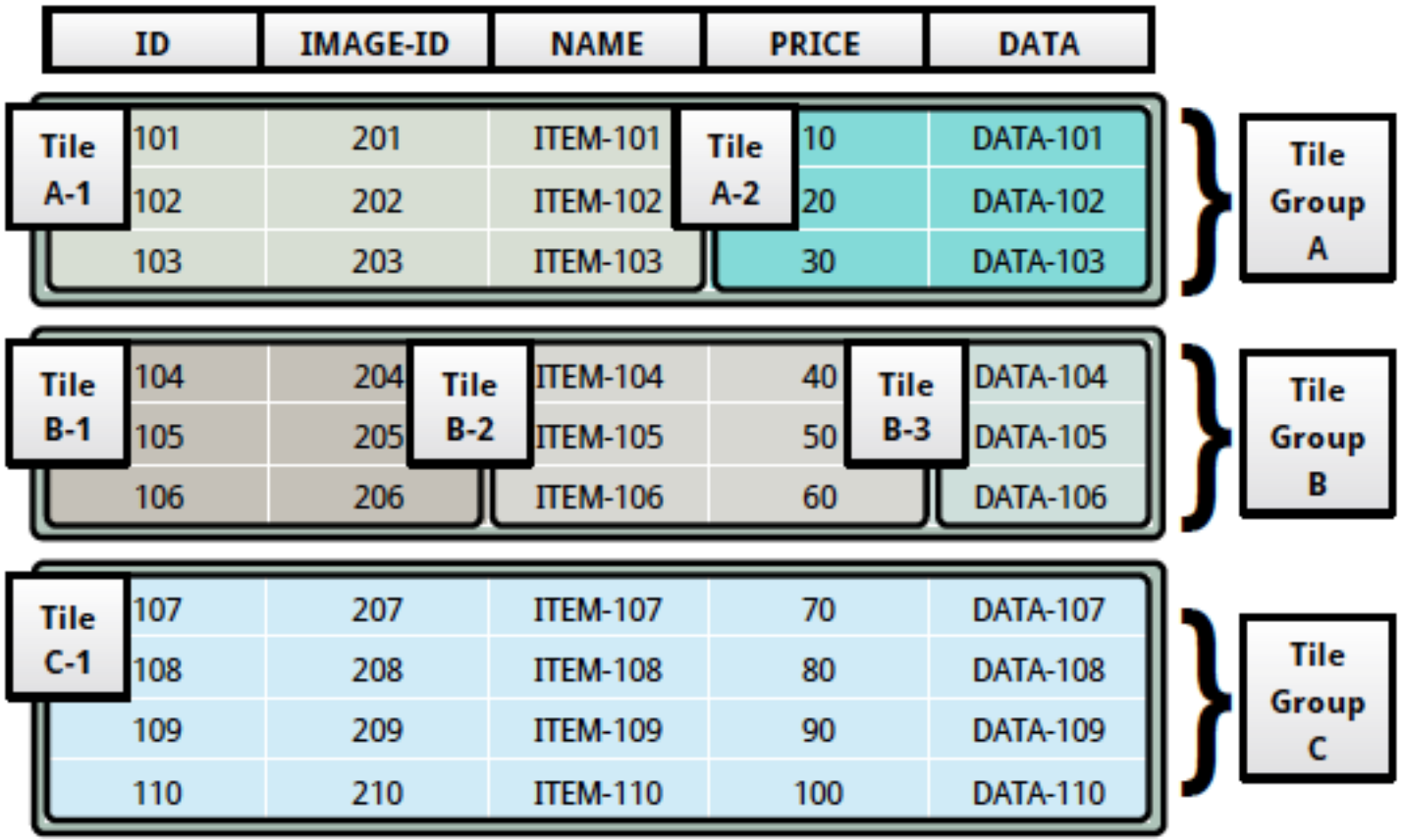
- Tile based architecture is FSM (flexible storage model) oriented DBMS architecture (expectations of OLTP + OLAP in same database)
- FSM is a storage model that uses the benefits of NSM(n-ary storage model) and DSM(Decomposition storage model)

Logical tile:

- Is an abstraction for the query engine (where all the primitive operations act over logical tiles).
- Hides the way that the data is stored (represents a physical tile, or a set of them).
- Performance benefits

Tile Based Architecture

Physical tile: The actual data is stored in physical tiles.



Arulraj, Joy, Andrew Pavlo, and Prashanth Menon. "Bridging the archipelago between row-stores and column-stores for hybrid workloads." *Proceedings of the 2016 International Conference on Management of Data*. ACM, 2016.

Project Goal

1. To evaluate experimentally (for the first time, ever) the potential of tile based architectures to support at least two non relational models.
2. What are the challenges, limitations and opportunities?

Schedule

Milestone	Date	Task Description
Milestone 1	**12.04 – 02.05**	DONE – Initial Research, Presentation and Team Organization
Milestone 2	**02.05 – 23.05**	Structure of the paper, introduction, Research on models and early implementation.
Milestone 3	**23.05 – 13.06**	First draft of the paper, early results of the implementation
Milestone 4	**13.06 – 04.07**	FINAL presentation with the results

Thank you!

Questions?

References

1. Rodriguez, Marko A., and Peter Neubauer. "Constructions from dots and lines."
2. Arulraj, Joy, Andrew Pavlo, and Prashanth Menon. "Bridging the archipelago between row-stores and column-stores for hybrid workloads."
3. Francesca Bugiotti and Luca Cabibbo Dipartimento di Ingegneria Universit`a Roma Tre. "A Comparison of Data Models and APIs of NoSQL Datastores."
4. Wiese, Lena. Advanced Data Management: For SQL, NoSQL, Cloud and Distributed Databases. Walter de Gruyter GmbH & Co KG, 2015.
5. Designing data intensive applications by Martin Kleppmann.