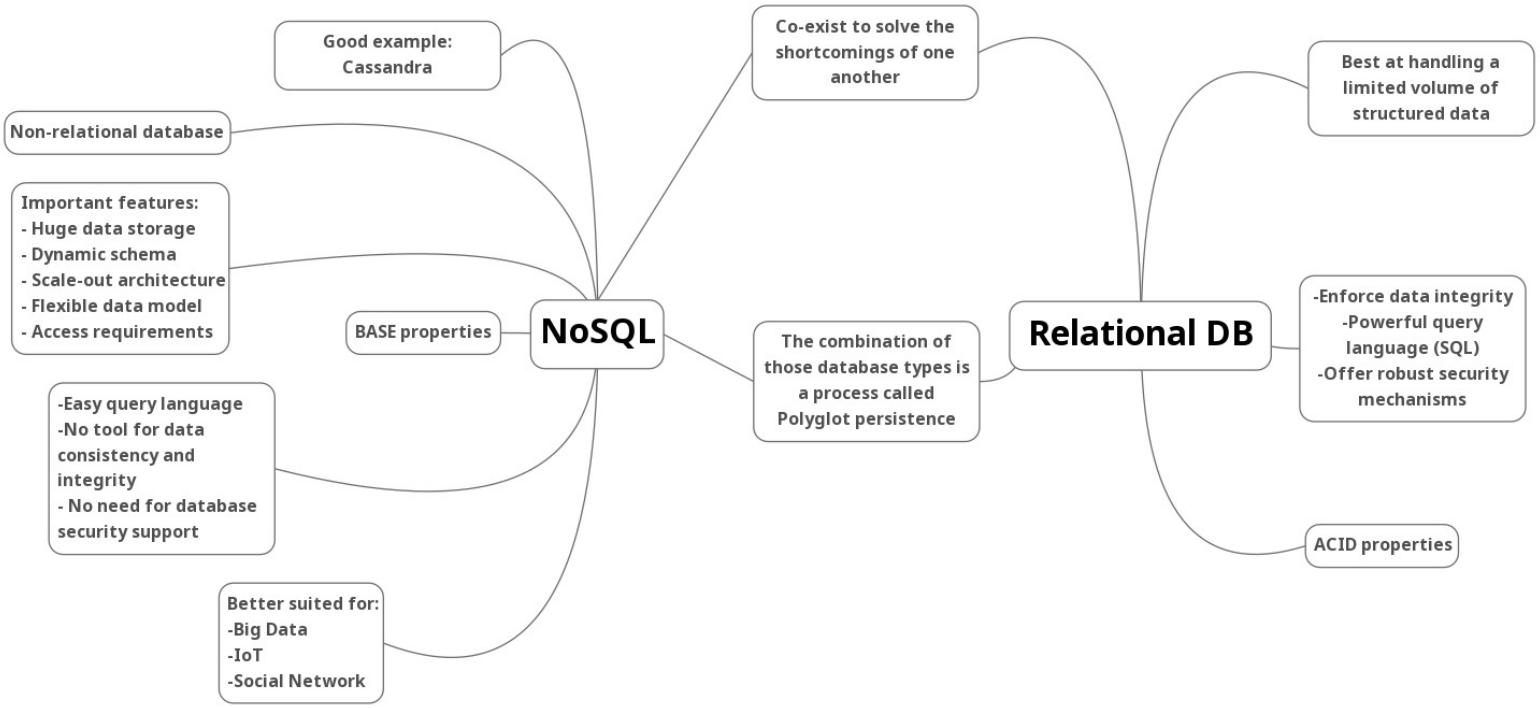
A Comparison of NoSQL and Relational Database Management Systems (RDBMS)

- Big enterprises like Facebook, Amazon, and Google have found that relational database technology has significant shortcomings when it comes to supporting massive quantities of data. NoSQL was created as a result of this phenomenon, which also led to its current rise in popularity.



i) **Cassandra:** Apache Cassandra is the only distributed NoSQL database that offers high availability, lightning-fast read-write speed, and limitless linear scalability required to support successful current applications. The capacity to manage organised, semi-structured, and unstructured data is a strength of Apache Cassandra. Initially created by Facebook, Apache Cassandra was made open-sourced in 2008 before becoming one of the top-level Apache projects in 2010.

ii) **ACID properties:**
Atomicity- Atomicity ensures that a transaction's commands are processed as a single entity and will either succeed or fail together.

Consistency- Consistency ensures that modifications performed inside a transaction follow with database constraints.

Isolation- Isolation makes sure that every transaction takes place in an isolated environment.

Durability- Durability ensures that changes are persisted after the transaction has finished and the database has been updated.

iii) **BASE properties:**
Basically **A**vailable- The system will always be available in the case of a failure.

Soft State- Due to eventual consistency, the status of the data could change without any involvement from applications.

Eventually Consistent: Data won't be consistent right away, but eventually it will be.

iv) **Polyglot Persistence:** refers to handling different data storage requirements by using various data storage systems. It is a type of polyglot programming, which involves creating an application utilising several distinct programming languages. Simply described, it's a programme that makes use of many primary database technologies.

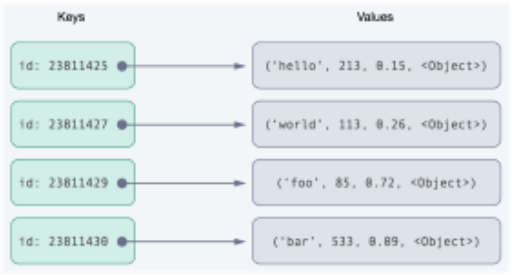
NoSQL Categories

Key value (KV) Store

-powerful approach in NoSQL database

-Data be kept as key-value pairs or key array pairs for the purpose of retrieval via keys.

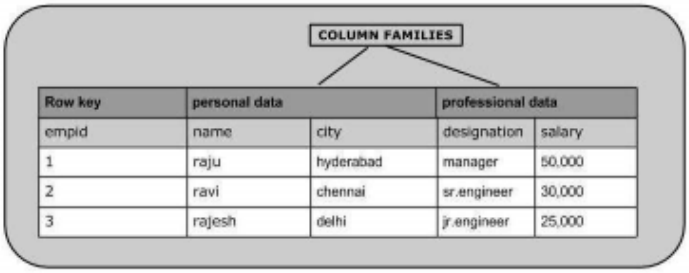
-Tables should have 2 columns. One column is for Primary Key, and another column is the holder of the logical values.



Column-Oriented (Column Family or Wide Column) Store

-The term "column family stores" refers to rows with several columns that store values closely and all columns arranged in accordance with the column family.

-New columns in rows may be added without inserting the values of already-existing rows.



Row key	personal data		professional data	
empid	name	city	designation	salary
1	raju	hyderabad	manager	50,000
2	ravi	chennai	sr.engineer	30,000
3	rajesh	delhi	jr.engineer	25,000

Document-Based Stores: Document-oriented Database

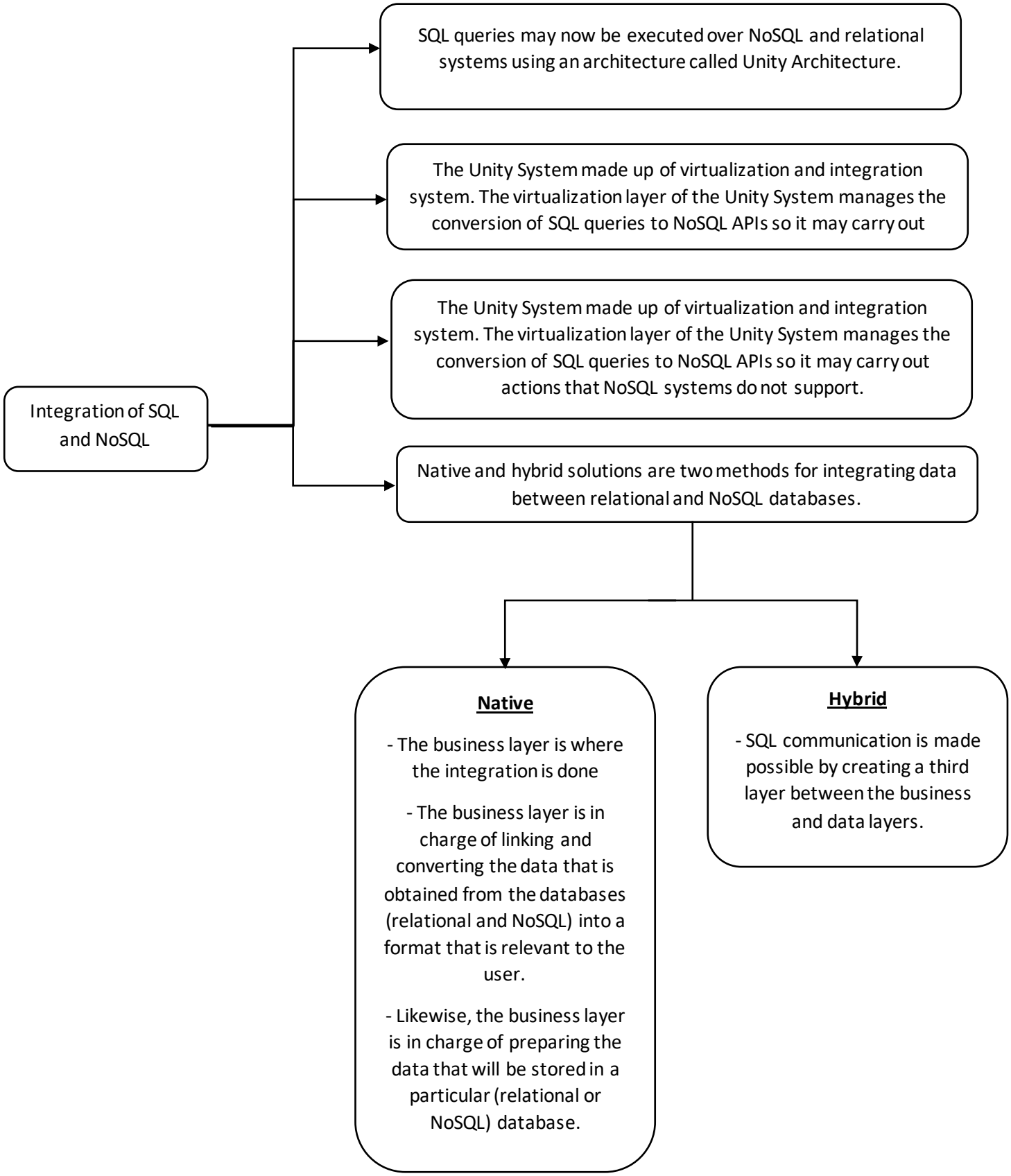
- Store collection of documents e.g. XML, PDF, JSON etc. and used for of data retrieval.
- Both keys and values may be searched for in the document.
- The key characteristics of document databases are fault tolerance and scalability.

Key	Document
1001	{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }
1002	{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }

Graph Database

-3 concepts: 1) Vertexes or Nodes are very similar to relational database tables.
2) Relationships between the Nodes called Edges.
3) Properties which are key-value pairs known as columns.

- This approach creates a link with data, allowing millions of records to be traversed.


Integration of SQL and NoSQL
SQL queries may now be executed over NoSQL and relational systems using an architecture called Unity Architecture.
The Unity System made up of virtualization and integration system. The virtualization layer of the Unity System manages the conversion of SQL queries to NoSQL APIs so it may carry out
The Unity System made up of virtualization and integration system. The virtualization layer of the Unity System manages the conversion of SQL queries to NoSQL APIs so it may carry out actions that NoSQL systems do not support.
Native and hybrid solutions are two methods for integrating data between relational and NoSQL databases.
Native
- The business layer is where the integration is done
- The business layer is in charge of linking and converting the data that is obtained from the databases (relational and NoSQL) into a format that is relevant to the user.
- Likewise, the business layer is in charge of preparing the data that will be stored in a particular (relational or NoSQL) database.
Hybrid
- SQL communication is made possible by creating a third layer between the business and data layers.

Conclusion

- Relational databases play an important role for managing ACID issues when the support of dynamic queries depends on the validity of the data. NoSQL is valuable for fraud detection over relational databases, particularly when faster access to the data is required. It is also necessary for scaling depending on need changes.

- The demand for relational databases among businesses will not go away any time soon, and neither will the distributed, product-based IT structure that exists now. NoSQL databases will continue to support major applications that the general public finds satisfying in the meantime.

- A major milestone is the integration of relational and NoSQL databases, which will support all user types and allow users to execute data analysis on data from either relational or NoSQL databases.