Carino, Jezelle M.

3/BSCS/A

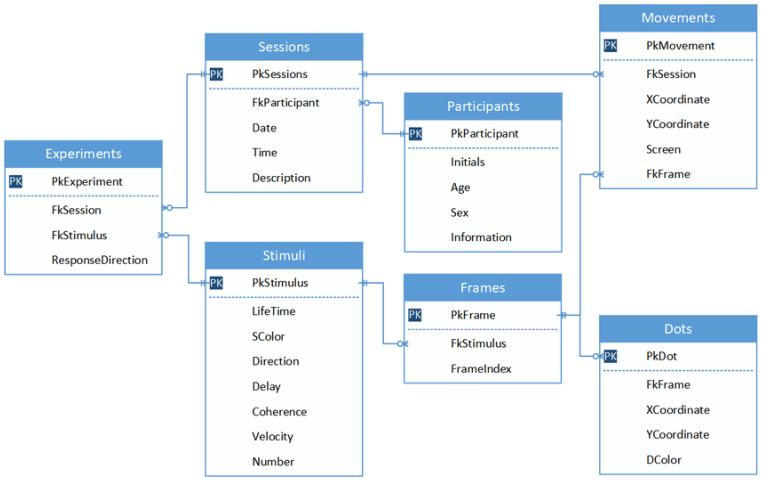
TYPES OF DATA BASE

1. Relational database

   A relational database is a form of database that stores and allows access to data elements that are linked. The relational model, a simple and obvious means of expressing data in tables, is the foundation of relational databases. Each row in a table in a relational database is a record with a unique ID called the key. The characteristics of the data are stored in the table's columns, and each record generally contains a value for each attribute, making it simple to construct links between data points.

   Here's an example of two tables that a small firm may use to process product orders. Each entry in the first table contains the customer's name, address, shipping and payment information, phone number, and other contact information. Each piece of data (attribute) has its own column, and each row has its own unique ID (key) assigned by the database. Each entry in the second table—a customer order table—includes the client's ID, the product ordered, the quantity, the size and color selected, and so on—but not the customer's name or contact information.

   The ID column is the only thing these two tables have in common (the key). However, the relational database may build a relationship between the two tables because of that shared column. The database can then go to the customer order table, pull the correct information about the product order, and use the customer ID from that table to look up the customer's billing and shipping information in the customer info table when the company's order processing application submits an order to the database. The warehouse can then fetch the proper goods, the consumer may receive their purchase on schedule, and the corporation can collect payment.



The Relational database Structure

The logical data structures—data tables, views, and indexes—are separated from the physical storage structures in the relational paradigm. As a result of this separation, database managers may adjust physical data storage without compromising logical data access. Renaming a database file, for example, does not rename the tables contained within it.

Database operations, which are clearly defined activities that enable programs to alter the data and structures of the database, are further divided into logical and physical categories. Physical operations indicate how that data should be accessible and then carry out the task, whereas logical operations allow an application to describe the material it requires.

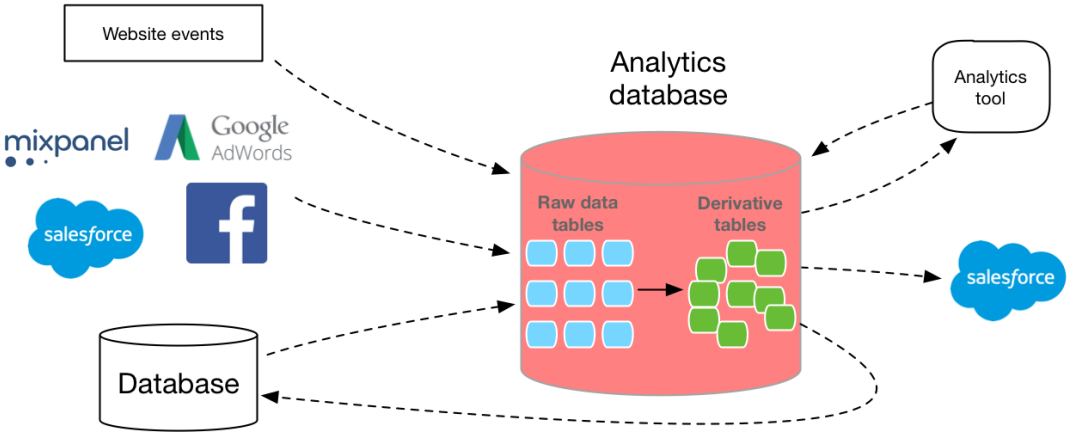
Relational databases follow particular integrity criteria to ensure that data is always valid and accessible. For example, an integrity rule can state that duplicate rows in a table are not permitted in order to prevent erroneous data from entering the database.

2. **Analytical database**

Big data management is a specialty of analytical database software for corporate applications and services. Analytical databases are designed to respond quickly to queries and deliver sophisticated insights. They're

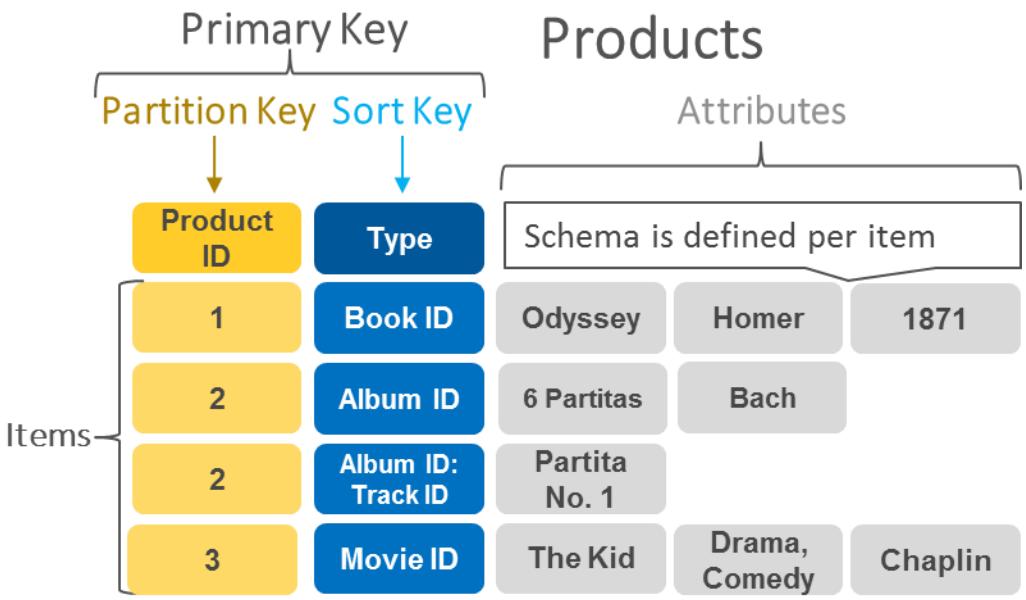
also more scalable than traditional databases, and they're generally columnar databases that can efficiently write and read data to and from hard disk storage to reduce query response times. Column-based storage, in-memory loading of compressed data, and the ability to search data by numerous characteristics are all aspects of analytical databases.

For intensive analytical workloads, analytical database software is intended to swiftly evaluate enormous volumes of data, performing up to 1,000 times quicker than an operational database. Business analysts, researchers, financial market analysts, big data analysts, geospatial analysts, and data scientists rely on analytical databases that can manage large amounts of data to be available.



3. **Key Value database**

A key-value database is a nonrelational database that stores data using a simple key-value manner. Data is stored in a key-value database as a collection of key-value pairs, with a key serving as a unique identifier. Both keys and values can be any type of object, from basic to sophisticated compound objects. Key-value databases are extremely partitionable and can scale horizontally to scales that other database cannot. If an existing partition fills to capacity and extra storage space is necessary, Amazon DynamoDB assigns new partitions to a table.



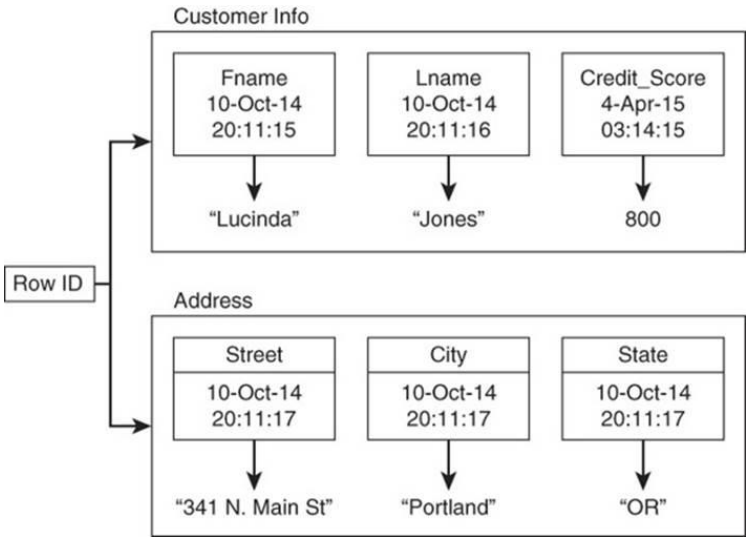
4. **Column-family database**

It's difficult to define what constitutes Big Data or a vast database. Is a MySQL table with a million entries considered a huge database? To some, it is, while to others, it is just an ordinary, though not little, table. When dealing with tables with billions of rows and tens of thousands of columns, however, there is little space for dispute. By any standard, that is a very large database (VLDB).

With a limited number of huge servers, relational databases could expand to VLDBs, but the cost would be prohibitive for most. For this kind of database, key-value databases are handy, but they lack assistance for organizing multiple columns and storing frequently used data together. Although document databases may scale to this level, they may lack some of the capabilities you'd anticipate at this level, such as a SQL-like query language.

Demands for very big database management systems are putting pressure on companies like Google, Facebook, Amazon, and Yahoo! The paper "BigTable: A Distributed Storage System for Structured Data"2 released by Google in 2006 presented a new form of database called a column family database. This database was created by Google for a number of its major services, including web indexing, Google Earth, and

Google Finance. BigTable became the standard for building massive NoSQL databases.



A collection of column families organizes a row in a column family database. A data value is indexed by a row, a column name, and a time stamp; column families are made up of related columns.

Some of the most scalable databases are column family databases. They provide developers the freedom to alter the columns within a column family. They also provide high availability, including cross-data center availability in some circumstances.

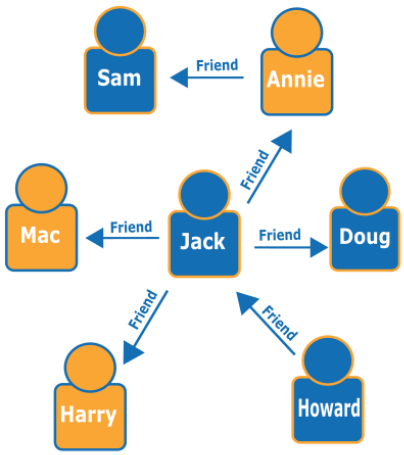
5. **Graph database**

Graph databases are designed specifically for storing and navigating relationships. Relationships are first-class citizens in graph databases, and they account for the majority of the database's value. Nodes are used to store data entities, while edges are used to store relationships between things in graph databases. An edge contains a start node, an end node, a type, and a direction, and it may be used to define parent-child connections, actions, and ownership, among other things. The amount and types of relationships that a node can have is limitless.

A graph can be navigated along specified edge types or over the whole graph in a graph database. Because the associations between nodes are not computed at query time but are stored in the database,

traversing the joins or relationships in graph databases is highly fast. When you need to construct linkages between data and query these associations fast, graph databases are useful for use cases like social networking, recommendation engines, and fraud detection.

A social network graph is depicted in the graph below. You may figure out who a person's "friends of friends" are by looking at the people (nodes) and their relationships (edges)—for example, Howard's pals.



For recommendation applications, graph databases are an excellent solution. You may record associations between information categories such as client interests, friends, and purchase history in graph databases. You may provide product suggestions to a user based on which items are purchased by others who follow the same sport and have comparable purchase histories using a highly accessible graph database. Alternatively, you may find folks who share a buddy but haven't met yet and offer a friendship referral(just like Facebook and other social media sites).

6. **Document database**

A document database is a nonrelational database that stores and queries data as JSON-like documents. By utilizing the same document-model format as their application code, document databases make it easier for developers to store and query data in a database. Documents and document databases can adapt to the demands of applications due to their flexible, semi-structured, and hierarchical nature. The

document model is particularly suited to use cases where each document is unique and changes over time, such as catalogs, user profiles, and content management systems. Flexible indexing, strong ad hoc searches, and analytics over collections of documents are all possible with document databases.

Document databases provide a number of benefits, including:

- A straightforward data model that developers can deal with quickly and easily.
- A flexible schema that permits the data model to adapt as the demands of the application change.
- The capacity to scale out horizontally.

Document databases are general-purpose databases that may be employed in a number of use cases and sectors due to their benefits.

```
{
  "_id": "5cf0029caff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  }
  "hobbies": ["surfing", "coding"]
}
```