

Name: Kaysar G. Sarip

Yr/Crs/Sec: 3BSCS-A

TYPES OF DATABASES

- **RELATIONAL** - A relational database is a form of database that stores data and allows users to access it. Each row in a table represents a record with a unique identifier known as the key. The attributes of the data are stored in the table's columns, and each record typically contains a value for each attribute. A small business, for example, might employ two tables to process orders for its items. Each client has a unique ID (a key) in each row of the customer order table. The database can look up billing and shipping information in the customer details table using that ID. How Relational Databases are structured? The logical data structures are distinct from the physical storage structures in the relational paradigm. Physical operations govern how data should be accessed, while logical operations allow an application to describe what content it requires. Relational databases adhere to a set of integrity standards to ensure that data is always valid and accessible.
- **Analytical (OLAP)** - Online Analytical Processing (OLAP) is a type of software that allows users to simultaneously evaluate data from numerous database systems. It's a tool that lets analysts obtain and analyze company data from various perspectives.

Grouping, aggregating, and joining data is a common requirement for analysts. These data mining OLAP processes are time consuming. Data can be pre-calculated and pre-aggregated with OLAP, allowing for quicker analysis.

One or more cubes can be found in OLAP databases. The cubes have been intended to make creating and reading reports simple. Online Analytical Processing (OLAP) is a term that refers to the processing of data in real time.

- **Key-Value** - is a nonrelational database that stores data using a simple key-value mechanism. 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 simple to sophisticated compound objects. Key-value databases are extremely partitionable and can scale horizontally to scales that other databases cannot. If a current partition fills to capacity and extra storage space is necessary, Amazon DynamoDB assigns additional partitions to the database.
- **Column-Family**—a column family is a database object that holds connected data columns. It's a tuple (pair) made up of a key–value pair with the key mapped to a value that's a collection of columns.

Two types of column families exist:

- ✓ **Standard column family:** contains only columns
- ✓ **Super column family:** contains a map of super columns

- **Graph** - 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, and edges are used to store relationships between things in graph databases. An edge has a start node, an end node, a type, and a direction, and it can be used to define parent-child connections, actions, and ownership, among other things. The amount and types of relationships that a node can have are limitless.

In a graph database, you can traverse a graph by looking at specific edge types or the entire graph. Because the relationships between nodes are not calculated at query time but are persisted in the database, traversing the joins or relationships in graph databases is highly quick. When you need to construct linkages between data and query these associations fast, graph databases are useful for applications like social networking, recommendation engines, and fraud detection.

- **Document** - by employing the same document-model format as their application code, document databases make it easier for developers to store and query data in databases. Flexible indexing, strong ad hoc searches, and document analytics are all possible with document databases. Catalogs, user profiles, and content management systems are all good candidates for the document model.