**What is a relational database?**

A relational database is a database that stores data by relating tables made up of rows and columns. Relational databases allow data to be application independent and accessed by multiple applications. Rows in the table represent entries where there is a key that represents a unique identifier for that row. The columns in the row represent different attributes or fields of the row. Because the data is tightly structured the data can be queried from most databases using SQL, allowing for easy written queries (Allardice, 2015).

**What is the role of the relational database in an organization today?**

RDMSs allow storage and retrieval of traditional and numeric data, such as the amount of items in stock or the amount and date of delivery for a specific item. RDMSs allow storage and retrieval of multimedia data, including videos, audio, and images. For instance YouTube can be structured as a relational database with the different videos being entries.

**What are the main components of a relational database?**

The main components of a relational database are what make up a particular database, such as its: data model, schemas and instances, tables, relationships, keys, etc. The data model is describes the data concepts and its constraints. The data model is the description of the data concepts, the data relationships, and various data constraints that make up the content of the database. It is broken into three parts the conceptual data model, logical data model, and physical data model. The conceptual data model relays the high-level concepts, close to how the business users perceive the data. The logical data model displays concepts in businesslike language that abstract the physical data organization but are independent from the physical technology. (<https://en.wikipedia.org/wiki/Logical_data_model>). Examples of a logical data models are hierarchical, network, relational, and object oriented models. The physical model represents low-level concepts that describe the data’s physical storage details.

The schema and its state distinguish between the description of the data and the data itself. The database schema is a description of the database, clarifies the data and their definitions, which is not expected to change frequently. The schema indicates the different tables that make up the database by specifying what is a row and the columns it possesses. An example of a database schema is “STUDENT (number, name, address, and email)” (Chapter 1: Fundamental Concepts of Database Management). The state is the data at a particular moment and is constantly in flux.

**What are the main characteristics of a relational database?**

The characteristics of a relational database are redundancy, integrity, and concurrency (based on containing ACIO properties). Relational databases provide reduced redundancy from previous types of databases (such as a file based database), increases consistency of the data, and guarantees correctness. This is done by having applications access a single repository of data vice each application having an independently database. Integrity of data is enforced through syntactical and semantical rules. Syntactical rules clarify standardize formats for data and semantical rules are logical business rules. Concurrency is allowed through the use of transactions when the database implements the following characteristics: Atomicity (all or nothing), Consistency (invariant of always transactions from a consistent to another consistent state), Isolation (transactions run is if they were in isolation), and Durability (changes to the database can be made permanent).

**What are the main functions of the Relational Database Management System (RDBMS)?**

The main functions of a RDNMS allow a user to manipulate the data in a database. The primary functions of SQL are SELECT, FROM, and WHERE.

**Why the database design is important?**

A good database design allows the users to make a RDBMS that closely matches the real life system. It is important to make sure that the requirements are accurately defined and the schemas are correctly developed. Successful requirements development enables well founded logical and physical design choices. If the requirements are incorrectly identified, development and design rework may be necessary, as well as possibly increased hardware costs.

**What other types of databases exist today on the market?**

There are multiple types of databases categorized differently based their physical data level, conceptual data level, and external data level. The databases that we have covered so far are categorized by their data model. Types of databases covered by the material include file based databases, relational databases, and NoSQL databases. NoSQL databases can be further broken down into document databases, graph stores, Key-value stores, and wide-column stores. Additional database types include Entity-Relationship, Object-oriented, object-relational, and semi-structural (<http://jcsites.juniata.edu/faculty/rhodes/dbms/dbarch.htm> ).

**What is the fundamental difference between NoSQL and a relational database?**

NoSQL databases are semi-structured while relational databases only work with structured data. Because NoSQL databases store semi-structured data they are not as strictly defined and do not adhere to the relational data model, allowing their schemas and the data stored to be dynamic. NoSQL databases are distributed and open sourced, while relational databases are typically centralized and a mix of propriety or open source.

**What is the role of a relational database in the Big Data ecosystem?**

Relational databases can be used in conjunction with Big Data. It's common to use a big data query to create a result set that is then stored in a relational database for use in the generation of analyses and reports. An example of an environment that uses relational databases that uses a relational database to processed Big Data is the Microsoft Analytics Platform System.

**In your opinion, what is the future for relational databases? Explain your reasoning.**

I think that Relational databases are here to stay for the foreseeable future. Relational databases benefit from having a relatively easy learning curve based off the structured nature of the data model. Relational databases are also ubiquitous and they will stay that way until there is a very good reason to not use them. They are easy to use and they work very well for most applications.

Bibliography

SQL. (n.d.). Retrieved January 26, 2016, from https://en.wikipedia.org/wiki/SQL