

Free essay on critique of relational database technology

[Business](#), [Management](#)



Introduction

Data is the building block of any system in the world today. A repository of such related data is called a database. A relational database organizes the data into tables that are related to each other. For example, in a relational database system, there exists one table that contains the data about cars manufactured by a company and the second table contains the information about car parts. These tables are related to each other, because each car is made up of parts, and each car part can be in one or more cars.

Haley (1991) defines a table as the principal building unit of a database. A table describes one logical entity and all the information about that entity. The data of a table is organized into rows, where each row represents a single record. Each column of the table contains one single unit of information within the table. For example, a table that contains information about cars may contain information about the model name, the year of manufacture, the color of the car, the name of the manufacturer, the city of manufacture, etc.

Example of the table:

In this table, each row represents information about one car. The data about these cars is organized into columns named Model, Manufacturer, Year of make, Color, and City of make. Each column therefore contains information about one aspect of the car. Each cell contains information about one aspect of one car. .

Relational Data Model

In a relational model, the data is organized into two-dimensional structures, called tables. These tables are called “relations”. A collection of such relations is called a database. This model views the real world in the form of entities and relationships and tries to model the databases around the same concepts. In the relational model, a relation represents an entity. An entity is a real world object about which the information is collected. The properties of the entities are called attributes. Entities are associated with one another through relationships (Hellerstein, 2007).

The primary goal of implementing the relational data model is to reduce duplication of data. This is done through the process of normalization. Data normalization is the process of removing the duplicity, or redundancy in the data. A database that is properly normalized works more efficiently. With little or no duplication, there are fewer chances of the data being corrupted. The person designing the database must be aware of the process of normalization in order to design a good relational model (Riordon, 2005).

Relational Database Management System (RDBMS) is a database system based on the relational model developed by E. F. Codd in the year 1970. This model allows the user to define the data structures, the storage and accessibility operations, and various integrity constraints (Codd, 1970). The properties of a relational table are:

- The values in each cell are atomic.
- Each row of the table is unique.
- All the values in a column are of the same type.
- The ordering and sequencing of the columns while storage is insignificant.

- The ordering and sequencing of the rows while storage is insignificant.
- Each column has a unique name.

Integrity Constraints

According to Healey (1991), an integrity constraint provides a mechanism for ensuring that the data conforms to the guidelines of the database system.

The two main purposes for which the integrity constraints are applied on the database system are:

- Data cleanliness: The integrity constraints, if applied correctly, verify that the data conforms to the basic levels of data correctness and consistency.

This prevents the introduction of dirty data.

- Query Optimization: The database systems utilize constraints at the time of optimizing the queries used to manipulate the data of the tables.

The common integrity constraints that a database administrator takes care of are:

- Unique Constraint: It is a column level constraint. It checks that the column on which the constraint is applied is unique. No values can be repeated through the column.
- NOT NULL Constraint: It is a column level constraint. It checks that the column on which this constraint is applied does not store any NULL values.
- FOREIGN KEY Constraint: This constraint is applied on columns belonging to different tables. It ensures that two keys in the related tables share a primary key to foreign key relationship.

Properties of Relational Database Model

Riordon (2005), states that any database in a relational database management system is expected to maintain four properties, known as ACID properties, to ensure that all the database transactions are processed reliably. ACID is an acronym that stands for Atomicity, Consistency, Isolation, and Durability. Cesarini & Soda (1985) define and describe these properties as below:

- Atomicity: is the ability of the databases that all the transactions are completed till the end. However, in case a part of the transaction fails in between, the whole transaction should rollback. It is also known as “ all-or-nothing” rule.
- Consistency: When the database exhibits this property, it ensures that after the completion of a transaction, the state of the database remains the same, as it was before the start of the transaction.
- Isolation: is the property of databases requiring that the outside operations are not able to see or access the data within an intermediate state during the transaction. It is important that a good database system design implements this property, for ensuring that one transaction is unaware of other transactions executing in parallel within the system (Sumathi, & Essakirajan, 2007).
- Durability: refers to the guarantee that once the user process has been notified of the transaction’s success, the transaction will not be undone. The transactions in the database system need to be able to survive the system failure, and the database systems need to verify the integrity constraints and won’t require the abortion of a transaction in case of a failure (Bagui, 2003).

Manipulating data through SQL

SQL, or Structured Query Language, is a language for querying and manipulating databases. A query is a statement that returns a set of records from the database. For example, you might want to see records of all the cars that are red in color (O'Reilly, 2004). To do so, below would be a sample SQL query:

It can be used not only for querying the database tables, but also for creating, updating and deleting tables and their data. It is capable of managing and manipulating both the content and the structure of the tables (Haley, 1991).

The language consists of different categories of statements. Each category has its own defined purpose:

- DDL Statements: DDL, or Data Definition Language statements allows a user to perform these types of tasks:
 - Create, update, and drop the database objects.
 - Granting and revoking roles and privileges.
 - Analyzing the data stored in a table, index, or a cluster.

In the case of DDL statements, the explicit commits are required to store the changes made to the table, if any. However, some database systems may implement auto-commit before and after every DDL transaction.

Examples: ALTER, CREATE, DROP, GRANT, PURGE, RENAME, REVOKE, TRUNCATE, UNDROP, etc.

- DML Statements: or Data Manipulation Language statements allow the access and manipulation of the data that is currently existing in the tables.

When any changes to the data are made, explicit commit is required to store the changes.

Examples: INSERT, DELETE, MERGE, SELECT, UPDATE

- TCL Statements: or Transaction Control Statements manage the changes made by DML statements.

Example: COMMIT, ROLLBACK, SAVEPOINT, and SET TRANSACTION

- Session Control Statements: allow the user to dynamically manage the properties of a database session. These statements do not implicitly commit the transaction.

Example: ALTER SESSION, and SET ROLE

- System Control Statement: Only one statement falls into this category: ALTER SYSTEM. It allows the user to dynamically manage the properties of a database system (Haley, 1991).

Strengths of Relational Database Model

Bagui (2003) describes the following advantages of storing the data into the collections of tabular forms:

- The design methodology is rigorous and based on sound theoretical concepts and foundations.
- All the other database structures can be worked out and represented in the form of relational tables. The tables are the most general form of the data representation.

- The relational database systems are comparatively easy to use and implement.
- New data can be added in the form of either new tables, or new records of data within the pre-existing tables, without any complexities.
- Retrieving ad hoc data by using the joins functionality and powerful SQL queries is extremely flexible.

Weaknesses of Relational Database Model

Despite the many advantages, the relational database systems have certain disadvantages as well, which need to be considered while working with RDBMS systems. Mansuri (n. d.), describes these weaknesses as follows:

- It requires more number of processing resources with the increasing number of users, when compared to other types of systems.
- Some complicated forms of queries, using multiple joins, can respond a little slower on the heavily loaded systems. However, this problem can be largely controlled through the apt use of indexes and various other optimization strategies (Bagui, 2003).

Moreover, relational databases that work with SQL have certain issues while working on the modern requirements, such as multimedia databases.

Multimedia data is large, and may swamp other data when stored in the relational databases. The data structures that are used to store multimedia are completely different from the traditional database systems. Moreover, these data structures do not easily comprehend to the content based searching. The standard SQL operations also do not apply to the multimedia form of data. Another limitation of the database model is that sequential

processing, though possible, is tedious. Since the relational model is built to provide dynamic access, sequential access requires some crafty manipulations.

These limitations have led developers to present new models of software development. The Object Oriented Model provides ways to store and access images, audios, and videos using compression techniques. NoSql is the solution suggested for providing sequential processing in applications such as pay roll systems (Mansuri, n. d.).

Many modern day applications use RDBMS technology and Object Oriented Planning as a base. Some examples of such applications are Enterprise Resource Planning (ERP) systems, SAP (System Applications and Products). ERP system is made up of many modules. Each module represents one functionality of a business enterprise. These modules talk to one another and make up a complete business IT solution. A common database built on the relational model underlies the ERP system. ERP systems can be built as per the requirements of the organization that uses it (Rashid et. al. 2002).

References

Edgar F. Codd from the ACM Portal

Mohammad A. Rashid (2002), Massey University-Albany, New Zealand, Liaquat Hossain Syracuse University, USA, & Jon David Patrick University of Sydney, Australia, The Evolution of ERP Systems: A Historical Perspective, Copyright © 2002, Idea Group Publishing

References

Bagui, S. 2003. ' Achievements and Weaknesses of Relational Databases'.

Journal of Object Technology, vol. 2 no. 4 pp. 30-37.

Cesarini, F. & Soda, G. 1985. ' Design and Implementation of a Relational DBMS'. The Computer Journal, vol. 28 no. 4 pp. 391-397.

Deperlioglu, O., Sarpkaya, Y., & Ergun, E. 2011. ' Development of a Relational Database For Learning Management Systems'. The Turkish Online Journal of Educational Technology, vol. 10 no. 4 pp. 108-111.

Healey, R G, 1991. Database Management Systems. 2nd ed. Colorado: Wiley & Sons.

Hellerstein, J. M., Stonebraker M., & Hamilton J. 2007. ' Architecture of a Database System'. Foundations and Trends in Databases, vol. 1 no. 2 pp. 143 - 156.

Mansuri, I, R., & Sarawgi, S. Integrating unstructured data into relational databases. IIT Bombay.

Moghaddam, A. I. 2009. ' Databases: From Paper-based to Web-based'. Library of Philosophy and Practice.

Relational Database Technology: A Crash Course. 2004. O'Reilly Media.

Riordon, R. M. 1999. Designing Relational Database Systems. Microsoft Press.

Riordon, R. M. 2005. Designing Effective Database Systems. Safari Publishers.

Simovici, D., & Tenney, R. 1995. Relational Database Systems. Massachusetts: Academic Press.

Sumathi, S., & Esakkirajan, S. 2007. Fundamentals of Relational Database Management Systems. Springer Science and Business Media.

Ullman, J. D., Molina, H. G., & Widom, J. 2008. Database Systems: The Complete Book. 2nd ed.