International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359

Research Article | **December 2012**

Basic Component of Relational Database Management System used in Web Application

Deepak Singh[1], Vaibhav Srivastav[2]
[2]MCA Scholar
[1, 2] Department of Computer Application, Babu Banarsi Das National Institute of Technology and Management, Lucknow, Uttar Pradesh, India

Abstract:

We present an innovative method for implementing a relational database back-end and ad hoc query translation to web Application. [1]Our system guarantees support for purpose of Database System, view of data, database languages, database design, Object-based semi structured database, data storage and querying , transaction Management, Database Architecture, [5] Database Users and administration and schema or any combination of schemas. Furthermore, we ensure that DBMS contains information about a particular enterprise that inbuilt; Collection of interrelated data, Set of programs to access the data and an environment that is both convenient and efficient to use. Database Application contains details of; Banking: all transactions, Airlines: reservations, schedules, Universities: registration, grades, Sales: customers, products, purchases, Human resources: employee records, salaries, tax deductions, [16]Manufacturing: production, inventory, orders, supply chain, Online retailers: order tracking, customized recommendations etc. Databases touch all aspects of our lives underlying schema of the relational database. We have developed a prototype system which accepts relational queries and translates them to a set of equivalent SQL queries. These queries are then posed to the relational back-end, and results are obtained. We will discuss the architecture and implementation details of our relational prototype, and give a detailed description of the algorithms used in query translation.

Key-words: Architecture and design of relational Database, Relational Schema, Data Models, E-R Model, Object relational data model, XML query processing, transaction management.

I. Introduction

In the early days, database applications were built directly on top of file systems. [15]Drawbacks of using file systems to store data: Data redundancy and inconsistency (Multiple file formats, duplication of information in different files), Difficulty in accessing data (Need to write a new program to carry out each new task), Data isolation — multiple files and formats, Integrity problems (Hard to add new constraints or change existing ones). Atomicity of updates (Failures may leave database in an inconsistent state with partial updates carried out), Concurrent access by multiple users (Concurrent accessed needed for performance, Uncontrolled concurrent accesses can lead to inconsistencies), Security problems (Hard to provide user access to some, but not all, data). [16] Relation database systems offer solutions to all the above problems. Relational database management systems have evolved from hierarchical to network to relational models. Oracle provides the relational model to an object-relational model with the different level of architecture.

1. **Relational Database Management System**
   Different level of Abstraction
   a) **Physical level:** describes how a record (e.g., customer) is stored.
   b) **Logical level:** describes data stored in database, the relationships among the data, the entire database for a community of users. It hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints.

type *customer* = record

i. ***customer_id*:string;**
***customer_name*:string;**
***customer_street*:string;**
***customer_city* : integer;**

end;

   c) **View level:** application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes. As shown in Fig.1

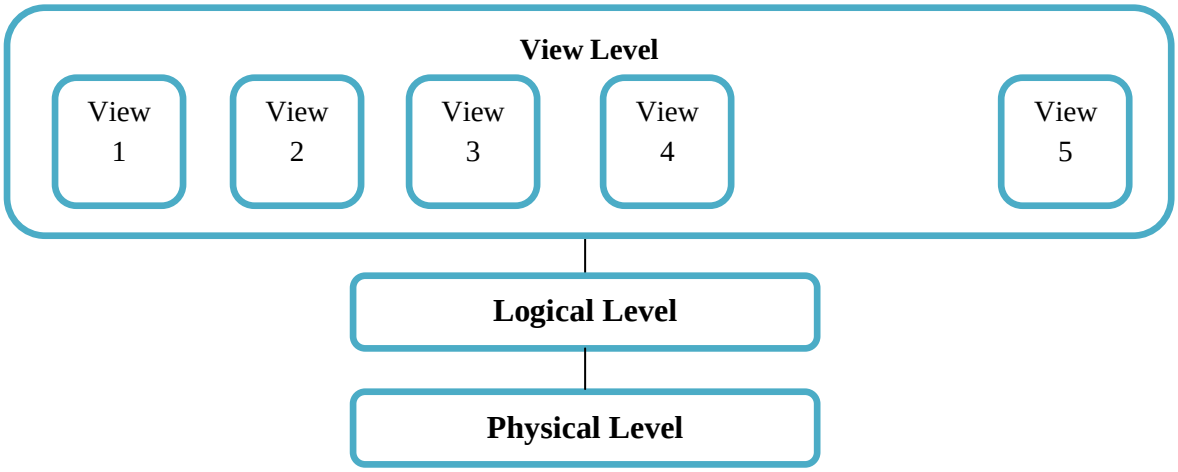

Fig.1

II. Instance and Schemas

[6]Database uses similar types and variables in programming languages. As, the schema and instance is being design at logical and physical level; **Schema** – the logical structure of the database e.g.: The database consists of information about a set of customers and accounts and the relationship between them). Schema is an analogous to type information of a variable in a program. **Physical schema**: database design at the physical level. **Logical schema**: database design at the logical level. **Instance** – the actual content of the database at a particular point in time (Analogous to the value of a variable).

Physical Data Independence – the ability to modify the physical schema without changing the logical schema (Applications depend on the logical schema). In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

Data Models

The tools for describing the data models are as follows: Data, Data relationships, Data Semantics, Data Constraints.

- Relational model
- Entity-Relationship data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semi structured data model (XML)
- Other older models:
  1. Network model
  2. Hierarchical model

The relational model has three major aspects:

a) **Structures** are well-defined objects (such as tables, views, indexes) that store or access the data of a database. Structures and the data contained within them can be manipulated by operations.

b) **Operations** are clearly defined actions that allow users to manipulate the data and structures of a database. The operations on a database must comply to a predefined set of integrity rules.

c) **Integrity rules** are the laws that govern which operations are allowed on the data and structures of a database. Integrity rules protect the data and the structures of a database.

Client/ Server Architecture and SQL

The DBMS is divided into two parts: a front-end (client) and a back-end (server) portion. The advantage of this division is the distribution of processes over multiple processors. This reduces the processing load and improves the system as a whole. [2] The client supports database applications; its focus is making requests and viewing data managed by the server. SQL (Structured Query Language) is the standard language for a DBMS. [3]The server portion receives and processes the SQL statements that originate from client applications. SQL statements execute all operations on the information in an Oracle database. SQL statements are divided into the following categories:

-Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  1. DML also known as query language.
- Two classes of languages:

1. **Procedural** – user specifies what data is required and how to get those data.
2. **Declarative (nonprocedural)** – user specifies what data is required without specifying how to get those data.

- SQL is the most widely used query language.

-Data Definition Language (DDL)

- Specification notation for defining the database schema
  Example:

```
create table account
(
        account-number char(10),
        balance  integer
)
```

- DDL compiler generates a set of tables stored in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)
  a. Database schema
  b. Data *storage and definition* language
     i. Specifies the storage structure and access methods used
  c. Integrity constraints
     i. Domain constraints
     ii. Referential integrity (**references** constraint in SQL)
     iii. Assertions
  d. Authorization

-Data Control Language (DCL)

Data control Language syntax similar to a computer programming language used to control access to data stored in a database.
Examples of DCL commands include:

- GRANT to allow specified users to perform specified tasks.
- REVOKE to cancel previously granted or denied permissions.

-Data Sub Language (DSL)

The end user uses a query language such as SQL or some application program. These languages include a data sub-language (DSL). A DSL is a subset of the total language that specifically deals with database objects and operations. The DSL is a combination of the DDL and the DML

-SQL (Structured Query Language)

SQL is widely used as a non-procedural language to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems.
Web application programs generally access databases through one of

- Language extensions to allow embedded SQL
- Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database.

The database server must reliably manage large amounts of data in a multi-user environment so that users can collectively access the same data. The server instance is a combination of background process and memory buffers. The Server consists of a database and server instance. The purpose of the database is to store and retrieve related information. [4]The System Global Area (SGA) is an area of memory used for database information shared by database users. When a database starts up, a SGA is allocated and background processes are started. Net8 is the mechanism the DBMS uses for interfacing with the communication protocols used by the networks that utilize distributed processing. [7]Communication protocols define the way that data is transmitted and received on a network.
For a properly engineered RDBMS-backed Web site, the RDBMS *client* is the Web server program, e.g., AOLserver. [9]The user types something into a form on a *Web client* (e.g., Netscape Navigator) and that gets transmitted to the *Web server* which has an already-established connection to an *RDBMS server* (e.g., Oracle). The data then go back from the RDBMS server to the RDBMS client, the Web server, which sends them back to the Web client: that an essential element here is database connection pooling. The Web server program must open connections to the database and keep them open. [10]When a server-side program needs to query the database, the Web server program finds a free database connection and hands it to the program Fig.2.

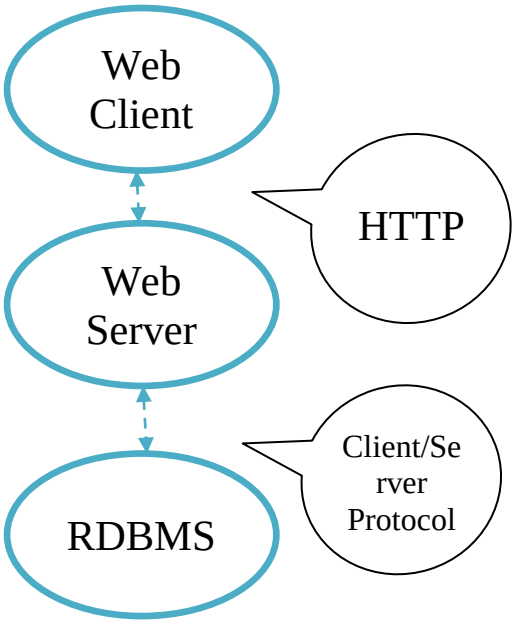

Fig.2

Database Design

The process of designing the general structure of the database, standardization makes your data model flexible and that makes working with your data much easier:

Logical Design – Deciding on the database schema. Database design requires that we find a "good" collection of relation schemas.

Business decision – What attributes should we record in the database?

Computer Science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?

Physical Design – Deciding on the physical layout of the database

The Entity-Relationship Model

E-R Model represents an enterprise as a collection of *entities* and *relationships*. In which **Entity**: a "thing" or "object" in the enterprise that is distinguishable from other objects; describe by the set of *attributes*. **Relationship**: an association among several entities. Fig.3

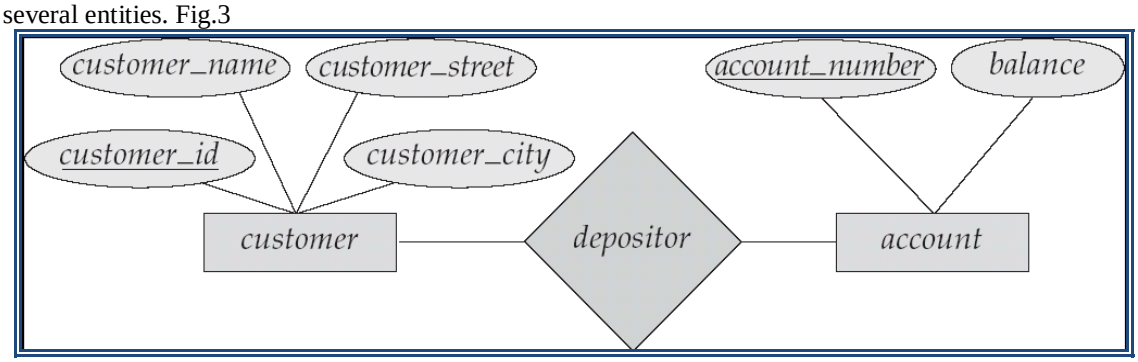

Fig.3

Object-Relational Data Models

Extend the relational data model by including object orientation and constructs to deal with added data types. Allow attributes of tuples to have complex types, including non-atomic values such as nested relations. Preserve relational foundations, in particular the declarative access to data, while extending modeling power. Provide upward compatibility with existing relational languages.

XML: Extensible Markup Language

It is defined by the WWW Consortium (W3C). An originally intended as a document markup language not a database language. The ability to specify new tags, and to create nested tag structures made XML a great way to exchange **data**,

not just documents. XML has become the basis for all new generation data interchange formats. A wide variety of tools is available for parsing, browsing and querying XML documents/data.

Storage Management

Storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
The storage manager is responsible to the following tasks:

- Interaction with the file manager
- Efficient storing, retrieving and updating of data

There are various issues like Storage access, File organization, Indexing and hashing.

Query Processing

The retrieval of information from database according to retrieval criteria, the database itself remaining unchanged. [12]In the context of a specific query language, the technique of translating the retrieval criteria specified using the language into more primitive database-access software, including a selection among different methods to choose the most efficient in the particular circumstances and technique like Parsing and translation, Optimization and Evaluation Fig.4.

- Alternative ways of evaluating a given query
  1. Equivalent expressions.
  2. Different algorithms for each operation.
- Cost difference between a good and a bad way of evaluating a query can be enormous
- Need to estimate the cost of operations
  1. Depends critically on statistical information about relations which the database must maintain.
  2. Need to estimate statistics for intermediate results to compute cost of complex expressions.

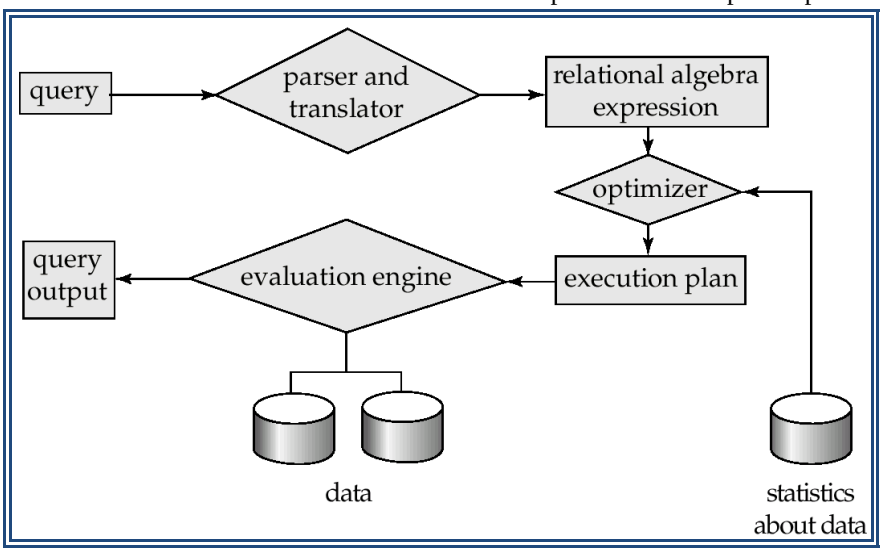

Fig.4

Transaction Management

A **transaction** is a collection of operations that performs a single logical function in a database application **Transaction-management component** ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures. **Concurrency-control manager** controls the interaction among the concurrent transactions, to ensure the consistency of the database [14].

Database Users

Users are differentiated by the way they expect to interact with the system or database server

- **Application programmers** – interact with system through DML calls
- **Sophisticated users** – form requests in a database query language
- **Specialized users** – write specialized database applications that do not fit into the traditional data processing framework
- **Naïve users** – invoke one of the permanent application programs that have been written previously

Database Administrator

A **database administrator** (**DBA**) is responsible for the flexibility, integrity, performance and security of a **database**. They will also be involved in the structuring, planning and development of the **database**, as well as troubleshooting any issues on behalf of the users.

- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise's information resources and needs.
- Database administrator's duties include:
  1. Schema definition
  2. Storage structure and access method definition
  3. Schema and physical organization modification
  4. Granting user authority to access the database
  5. Specifying integrity constraints
  6. Acting as liaison with users
  7. Monitoring performance and responding to changes in requirements

III. Results

The RDBMS server relies on research relational prototypes evolve into commercial systems and parallel and distributed database systems. Network, hierarchical data models in widespread use, high transaction processing and Object-oriented database systems lien into the performance. Large decision support, data-mining applications and large multi-terabyte data warehouses are to be act as a effective relational database system. The successful network configuration of the DBMS has allowed the Introduction to Database Systems course to take place in a professional environment. Creation of user accounts for the students and implemented a backup and recovery system for the database is fruitful for current users.

IV. Conclusions

The object approach in relation to a DBMS is a way to frame solutions to problems. It is an approach of abstraction that allows developers to represent entities with particular states and behaviors. Objects are manipulated according to specific protocol: this process is known as the object's interface. The result of this allows developers to reuse code in different applications or databases. It models the application or database as a set of objects collaborating with one another to fulfill their responsibilities.

The DBMS supports multi-users executing a variety of database applications operating on the same data and do not suffer from slow processing performance. The DBA can selectively control the availability of data. The DBA has the ability to provide fail-safe security features to limit and monitor data access. These features enforce rules of data integrity. The result of this eliminates the need of coding and managing checks in developing database applications. Oracle combines the data physically located on different computers into one logical database that can be accessed by all network users. The software is developed to work under different operating systems. The applications developed for Oracle can be transferred to any operating system with little or no modification. Discussions will focus on the power of the DBMS its robust, scalable, efficient, etc.

References

[1] Barry & Associates. OODBMS Facts. April 2001. http://www.odbmsfacts.com.

[2] Chaterjee, Jagadish. Introduction to RDBMS, OODBMS and ORDBMS. January 3, 2005. http://www.aspfree.com/c/a/Database/Introduction-to-RDBMS-OODBMS-and-ORDBMS/

[3] Cigler, James B. Orooji, Ali. ORR: Object-Relational Rapprochement. *COMPSAC '99. Proceedings. The Twenty-Third Annual International.* October *27*-29, 1999. Page(s):42 - 48

[4] Devarakonda, Ramakanth S., Object-relational Database Systems — The Road Ahead, *Crossroads*, March 2001. Volume 7 Issue 3.

[5] Fong, Joseph. Converting Relational to Object-Oriented Database. *d*. March 1997. Volume 26 Issue 1.

[6] Kim, Won. Object-Oriented Database Systems: Strengths And Weaknesses. *Journal of Object-Oriented Programming Focus On ODBMS.* 1992.

[7] Kisworo, M.W.; Rajagopalan, P. Implementation of an Object-Oriented Front-End to a Relational Database System. *IEEE Region 10 Conference on Computer and Communication Systems.* 24-27 Sept. 1990 Page(s):811 – 815.

[8] McClure, Steve. Object Database vs. Object-Relational Databases. *IDC Bulletin* #14821E - August 1997.

[9] McFarland, Gregory, Rudmik, Andres, and Lange, David - Modus Operandi, Inc. Jan 31, 1999. Object-Oriented Database Management Systems Revisited: An Updated DACS State-of-the-Art Report. https://www.dacs.dtic.mil/techs/oodbms2/oodbms-toc.shtml.

[10] Obasanjo, Dare 2001. An Exploration of Object-Oriented Database Management Systems. http://www.25hoursaday.com/WhyArentYouUsingAnOODBMS.html.
[11] Rahayu, W.; Chang, E.; Dillon, T.S. Implementation of Object-Oriented Association Relationships in Relational Databases. *Database Engineering and Applications Symposium*, July 1998. Page(s):254 – 263.
[12] Smith, Karen E., Zdonik, Stanley B., INtermedia: A Case Study of the Differences Between Relational and Object-Oriented Database Systems. *OOPSLA '87 Proceedings.* October 4-8, 1987.
[13] Sujithan, K. R. An Object Model of Data, Based on the ODMG Industry Standard for Database Applications. The Institution of Electrical Engineers. 1995.
[14] Zand, Mansour, Collins, Va, Caviness, Dale. A Survey of Current Object-Oriented Databases. *Data Base Advances*, February 1995. Volume 26, No. 1.
[15] Philip Diazand Prof. W.C. Harris, Department of physical, Environment and Computer Sciences Medgar Evers College of the City University of New York
[16] Silberschatz, Korth and Sudarshan Database System Concept 5th Edition