Development of a Data Type Upgraded BIGINT to "INTCAL" Data Type in a RDBMS

Pilli Sudheer, K.L.Narasimharao , Yannam Apparao

Abstract: **Traditional Database Management System which only provide for primitive data types such as INTEGER and CHARACTER. The data types offered in commercial database systems, CHAR INTEGER NUMBER, VARCHAR, VARCHAR2(),are sufficient for most relational database applications but range wise varing from one data type to on their data type now beginning to realize that the ability to create user-defined data types can greatly simplify their database design, in existing system the data type int range from -2,147,483,648 to 2,147,483,647. This paper proposed new data type in sql named " INTCAL " range between from -9,223,372,036,854,775,999 to 9,223,372,036,854,775,999.**

Index Terms: **RDBMS, sql, data bases, tables, columns, Data types and records**

I.INTRODUCTION

A data type is a set of representable values. Each value belongs to one data type. Items that can be referenced by a name, such as SQL parameters, columns, fields, attributes, and variables, also have declared types. MySQL supports three groups of data types:

- Numeric
- Date & time
- String

MySQL uses all the standard ANSI SQL numeric data types, so if you're coming to MySQL from a different database system, these definitions will look familiar. The following list shows the common numeric data types and their descriptions:

- INT - A normal-sized integer that can be signed or unsigned. If signed, the allowable range is from -2147483648 to 2147483647. If unsigned, the allowable range is from 0 to 4294967295. You can specify a width of up to 11 digits.
- TINYINT - A very small integer that can be signed or unsigned. If signed, the allowable range is from -128 to 127. If unsigned, the allowable range is from 0 to 255. You can specify a width of up to 4 digits.
- SMALLINT - A small integer that can be signed or unsigned. If signed, the allowable range is from -32768 to 32767. If unsigned, the allowable range is from 0 to 65535. You can specify a width of up to 5 digits.
- MEDIUMINT - A medium-sized integer that can be signed or unsigned. If signed, the allowable range is from -8388608 to 8388607. If unsigned, the allowable range is from 0 to 16777215. You can specify a width of up to 9 digits.
- BIGINT - A large integer that can be signed or unsigned. If signed, the allowable range is from -9223372036854775808 to 9223372036854775807. If unsigned, the allowable range is from 0 to 18446744073709551615. You can specify a width of up to 20 digits.
- FLOAT(M,D) - A floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 10,2, where 2 is the number of decimals and 10 is the total number of digits (including decimals). Decimal precision can go to 24 places for a FLOAT.
- DOUBLE (M,D) - A double precision floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 16,4, where 4 is the number of decimals. Decimal precision can go to 53 places for a DOUBLE. REAL is a synonym for DOUBLE.
- DECIMAL (M,D) - An unpacked floating-point number that cannot be unsigned. In unpacked decimals, each decimal corresponds to one byte. Defining the display length (M) and the number of decimals (D) is required. NUMERIC is a synonym for DECIMAL.

II. RELATEDWORK

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Manuscript received November 23, 2014

Mr. P.Sudheer, Assistant Professor, MLR Institute of Technology & Management, Dundigal, Quthbullar(M),R.R.Distic,Telangana-500043.
Mr. K.LNarasimharao, Assoc. Professor, MLR Institute of Technology & Management, Dundigal, Quthbullar(M),R.R.Distic,Telangana-500043.
Mr. Yannam Apparao, Associate Professor, MLR Institute of Technology & Management, Dundigal, Quthbullar(M), R.R.Distic, Telangana-500043.

Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.

So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

A Relational DataBase Management System (RDBMS) is a software that:

•Enables you to implement a database with tables, columns and indexes.

•Guarantees the Referential Integrity between rows of various tables.

•Updates the indexes automatically.

•Interprets an SQL query and combines information from various tables.

SQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

- SQL is released under an open-source license. So you have nothing to pay to use it.
- SQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- SQL uses a standard form of the well-known SQL data language.
- SQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- SQL works very quickly and works well even with large data sets.
- SQL is very friendly to PHP, the most appreciated language for web development.
- SQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- SQL is customizable. The open-source GPL license allows programmers to modify the SQL software to fit their own specific environments.

A. Intcal Data Type

```
create table account (
 account_number    char(10),
branch_name        char(10),
balance               integer,
address intcal );
```

B. Create A Database & SQL Data types

This paper gives the Creation of database while creating the database given a data type which is realed to mysql .

- CREATE TABLE IF NOT EXISTS `myflixdb`.`Members` (
- `membership_number` INT AUTOINCREMENT ,
- `full_names` VARCHAR(150) NOT NULL ,
- `gender` VARCHAR(6) ,
- `date_of_birth` DATE ,
- `physical_address` VARCHAR(255) ,
- `postal_address` VARCHAR(255) ,
- `contact_number` VARCHAR(75) ,
- `email` VARCHAR(255) ,
- PRIMARY KEY (`membership_number`))
- ENGINE = innodb;

C. Choosing the Right Type for a Column

For optimum storage, try to use the most precise type in all cases. For example, if an integer column is used for values in the range from 1 to 99999, MEDIUMINT UNSIGNED is the best type. Of the types that represent all the required values, this type uses the least amount of storage.

Tables created in MySQL use a new storage format for DECIMAL columns. All basic calculations (+, -, *, and /) with DECIMAL columns are done with precision of 65 decimal (base 10) digits.

MySQL calculations on DECIMAL values are performed using double-precision operations. If accuracy is not too important or if speed is the highest priority, the DOUBLE type may be good enough. For high precision, you can always convert to a fixed-point type stored in a BIGINT. This enables you to do all calculations with 64-bit integers and then convert results back to floating-point values as necessary.

RDBMSs use Structured Query Language (SQL) as the data definition language (DDL) and the data manipulation language (DML).

SQL includes statements for data definition, modification, querying and constraint

specification. The types of queries vary from simple single-table queries to complicated multi-table queries involving joins, nesting, set union/differences.

III. PURPOSE OF DATABASE SYSTEMS

- 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.

Example: Transfer of funds from one account to another should either complete or not happen at all

- Concurrent access anomalies: Example: Two people reading a balance and updating it at the same time
- Security problems: Database systems offer solutions to all the above problems

IV.VIEW OF DATA

A database is a collection of interrelated data and set of programs that allow users to accessand modify these data. A major purpose of a database system is to provide users with an abstract view of the data. That is, the system hides certain details of how the data stored and maintained.

A. Data Abstraction:

Major purpose of DBMS is to provide users with abstract view of data i.e. the system hides certain details of how the data are stored and maintained. Since database system users are not computer trained, developers hide the complexity from users through 3 levels of abstraction, to simplify user's interaction with the system.

i) Levels of Abstraction

- Physical level of data abstraction: How the data are actually stored. This s the lowest level of abstraction which describes how data are actually stored.
- Logical level of data abstraction: This level hides what data are actually stored in the database and what relations hip exists among them. Describes data stored in database, and the relationships among the data.
- View Level of data abstraction: View provides security mechanism to prevent user from accessing certain parts of database. application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

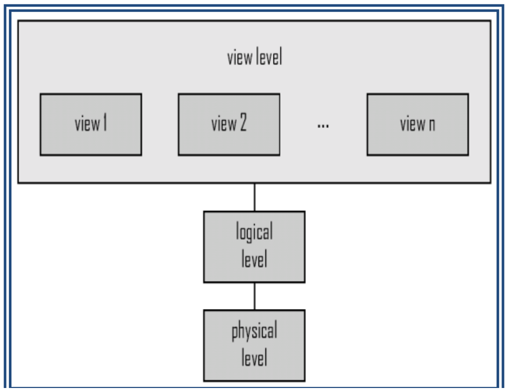

Figure 1.TheThree level of abstraction.

B. Instances and schemas:

Instance: The collection of information stored in the database at a particular movement is called an instance of the database.

Similar to types and variables in programming languages Schema: the overall design of the database is called the database schema.

Example: The database consists of information about a set of customers and accounts and the relationship between them .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.

C. Data models

Data Model: Underlying the structure of a database is the data model,A collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints.

- Relational model: The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name, Tables are also called known as relations.
- Entity-Relationship Model: The Entity –Relationship (E-R) data model uses a collection of basic objects, called entities, and relationships among these objects. An entity is a "thing" or "object" in the real world that is distinguishable from other object
- Object-Based Data Models: Object-oriented Programming (especially in Java, C++,or C#).
- Semi structured Data Model: The semi structured data model permits the specification of data where individual data items of the same type may have different sets of attributes.
- Other older models:
  - – Network Model
  - – Hierarchical Model

IV. DATA BASE LANGUAGES

A Database provides a DDL to specify the database schema and a DML to express database queries and updates.

A. Data-Manipulation Language

A data-manipulation language (DML) is a language that enables users to access or manipulate data as organized by the appropriate data model.

The types of access are:

• Retrieval of information stored in the database
• Insertion of new information into the database
• Deletion of information from the database
• Medication of information stored in the database

There are basically two types:
Procedural DMLs require a user to specify what data are needed and how to get those data.• Declarative DMLs (also referred to as non procedural DMLs) require user to specify what data are needed without specifying how to get those data.

B. Data- Definition Language (DDL):

We specify a database schema by a set of definitions expressed by a special language called a data-definition language(DDL).The DDL is also used to specify additional properties of the data.SQL provides a rich DDL that allows one to define tables, integrity constraints, assertions,

```
Example:    create table account (
                account_number    char(10),
                branch_name       char(10),
                balance           integer)
```

In addition, the DDL statement updates the data dictionary, which contains metadata; the schema of a table is an example of metadata.The PersonID column is of type int and will hold an integer.The LastName, FirstName, Address, and City columns are of type varchar and will hold characters, and the maximum length for these fields is 255 characters.

C. Intcal data type

```
create table account (
 account_number    char(10),
 branch_name       char(10),
 balance           integer,
address intcal );
```

V. DATA BASE ACCESS FROM APPLICATION PROGRAMS:

Application programs are programs that are used to interact with the database.To access the database, DML Statements need to be executed form the host language. ‘

There are two ways o do this.

- By Providing an Application Program interface (set of procedures) that can be used to send DML and DDL statement to the database and retrieve the results.(ODBC and JDBC).
- By extending the host language syntax to embed DML calls within the host language program. A special character prefaces DML calls and preprocessor called the DML pre complier ,converts the DML statements to normal procedure calls in the host language

SQL: widely used non-procedural language

Example 1:Find the name of the customer with customer-id 192-83-7465

```
        select    customer.customer_name
        from      customer
        where     customer.customer_id = ‘192-83-7465’
```

Example 2: Find the balances of all accounts held by the customer with customer-Id 192-83-7465.

```
        select    account.balance
        from       depositor, account
        where     depositor.customer_id = ‘192-83-7465’
and  depositor.account_number = account.account_number
```

Example 3:Find the name of the customer with customer-id 192-83-7465

```
        select    customer.customer_name
        from      customer
        where     customer.customer_id = ‘192-83-7465’
```

VI. CONCLUSION

“ Intcal “gives a one data type in sql here we can extend the range of the INT with meaningful size like data type in sql named “INTCAL“ range between from -9,223,372,036,854,775,999 to 9,223,372,036,854,775,999. By using this we can implements in SQL also in mobile application .

REFERENCES

[1]M. Abdallah, R. Guerraoui, and P. Pucheral. One-phase commit: Does it make sense? InICPADS ’98: Proceed-ings of the 1998 International Conference on Parallel and Distributed Systems , page 182, Washington, DC, USA, 1998. IEEE Computer Society.

[2]Y. J. Al-Houmaily and P. K. Chrysanthis. 1-2pc: the one-two phase atomic commit protocol. In Proceedings of the 2004 ACM Symposium on Applied Computing (SAC), Nicosia, Cyprus, March 14-17, pages 684–691, 2004.

[3]R. Bayer, H. Heller, and A. Reiser. Parallelism and recovery in database systems.ACM Trans. Database Syst.5(2):139–156, 1980.

[4]P. A. Bernstein and N. Goodman. Multiversion concurrency control - theory and algorithms.ACM Trans.Database Syst., 8(4):465–483, 1983.

[5]P. A. Bernstein, V. Hadzilacos, and N. Goodman.Concurrency Control and Recovery in Database Systems.Addison-Wesley, 1987.

[6]W. Cellary and G. Jomier. Consistency of versions in object-oriented databases. In D. McLeod, R. Sacks-Davis,and H.-J. Schek, editors, 16th International Conference on Very Large Data Bases, August 13-16, 1990, Brisbane,Queensland, Australia, Proceedings, pages 432–441. Morgan Kaufmann, 1990.

[7]I.-M. A. Chen, V. M. Markowitz, S. Letovsky, P. Li, and K. H. Fasman. Version management for scientificdatabases. In P. M. G. Apers, M. Bouzeghoub, and G. Gardarin, editors,Advances in Database Technology-EDBT’96, 5th International Conference on Extending Database Technology, Avignon, France, March 25-29,1996, Proceedings ,volume 1057 ofLecture Notes in Computer Science, pages 289–303. Springer, 1996.