SQL standard and its components

SQL (originally: Structured Query Language) is an internationally-recognized programming language for defining and maintaining relational databases.

The sql standard includes the following parts:

- SQL/Framework, SQL/Foundation, SQL/Bindings, SQL/Object
- New parts addressing temporal, transaction, and other aspects
- SQL/CLI(Call Level Interface)
- SQL/PSM(Persistent Stored Modules)

Object Relational Support in SQL-99

The SQL/Object specification in SQL-99 (also known as SQL3) extends SQL-92 to include object-oriented capabilities. Some of the features are:

- Type constructors
    - Complex types
    - User defined types - UDTs
- Object Identifiers using references
- Encapsulation of Operations (related methods for UDTs)
- Inheritance and function overloading (create TYPE under another TYPE)
- Unstructured complex objects

Evolution and current trends of database technology

- Complex object database management
- High Volume Database Applications
- Multimedia Databases
- Mobile Databases
- Spatial Databases
- Online Analytical Processing Database

Informix Universal Server

The Informix Universal Server is an Object Relational DBMS that combines relational and object database technologies from two previously existing products: Informix and Illustra. Illustra originated from POSTGRES DBMS. The current product, after acquisition of Informix by IBM, is called IBM Informix Dynamic Server.

Consider two dimensions of DBMS applications: Complexity of data (x), Complexity of queries (y) and Observe the possible quadrants:

- Quadrant 1 (x=0, y=0): simple data, simple query
- Quadrant 2 (x=0, y=1): simple data, complex query
- Quadrant 3 (x=1, y=0): complex data, simple query
- Quadrant 4 (x=1, y=1): complex data, complex query

Traditional RDBMSs belong to Quadrant 2, many object DBMSs belong to Quadrant 3. Informix Universal belongs to Quadrant 4; it extends the basic relational model by incorporating a variety of features that make it object-relational.

How it extends the relational model:

- Support for extensible data types
    - A number of new data types are provided
        - Two-dimensional geometric objects
        - Images
        - Time series
        - Text
        - Web pages
    - Can Declare new types:

            - **Opaque type**:Encapsulates a type (hidden representation)
            - **Distinct type**:Extends an existing type thru inheritance
            - **Row type**:Represents a composite type (like C's struct)
            - **Collection type**:Lists, sets, multi-sets (bags)

- Support for user-defined routines
    - supports **user-defined functions** and routines to manipulate **user-defined types**
    - Functions are implementedEither in **Stored Procedure** (**SPL**)or in a high-level programming language
    - Functions can define operations like**plus, times, divide, sum, avg, negate**
- Implicit notion of inheritance
    - Data Inheritance(type)
- Support for indexing extensions
- Database Blade API
    - Provide new data type and functions for specific application type
    - Special data type, image data type, time series data type, text data type etc.

Oracle 8

Oracle 8.X was enhanced to incorporate object relational features. Some of the features are:

- **VARRAY** for representing multi-valued attributes
- store extremely large objects
    - **RBLOB** (binary large object)
    - **CLOB** (character large object)
    - **BFILE** (binary file stored outside the database)
    - **NCLOB** (fixed-width multibyte CLOB)
- Partitioned tables and indexes

Implementation and Related Issues for Extended Type Systems

- The **ORDBMS** must dynamically link a user-defined function in its address space
- **Client-server issues**:
    - if a server needs to perform a function, it is best to do so in the DBMS (server) address space
- Queries should be possible to run inside functions
- Efficient storage and access of data
    - Especially given new types, is very important
- Design of the database is more complex
- Query processing and optimization is also complex

Nested Relational Model

A drawback of relational model is that it can represent data only in the simple first normal form. One of the improvisations to the relational model is Nested Relational model which is based on a principal that decompositions of relations according to multi-valued dependencies are not always necessary in order to prevent errors in updating.

In this model, each value in the model can be a set or a hierarchical structure. There are two advantages, i.e. the expressions of data in the model are natural or efficient and less join operations are necessary, which means data processing becomes generally faster.

In the beginning the model was called the non-first-normal-form(NF). Unstructured databases like NoSql support this feature.