Project Training

Session 2

Topics

- Presentation of SQL
  - History
  - Usage
- Presentation of Mysql, Postgres and SQLite
  - Core mechanism
  - Usage
- Database modelisation
  - How to modelize a database

What is SQL?

SQL is a language to operate databases; it includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an **ANSI** (American National Standards Institute) standard language, but there are many different versions of the SQL language depends on the SQL engine that you are using.

If you build a query that works on MySQL, it may not work on PostgreSQL or SQLite.

What is SQL?

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

What is SQL?

Also, they are using different dialects, such as –

- MS SQL Server using T-SQL,
- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL (native format) etc.

Why SQL?

SQL is widely popular because it offers the following advantages –

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

History

SQL was initially developed at IBM by Donald D. Chamberlin and Raymond F. Boyce in the early 1970s. This version, initially called SEQUEL (Structured English Query Language), was designed to manipulate and retrieve data stored in IBM's original quasi-relational database management system, System R, which a group at IBM San Jose Research Laboratory had developed during the 1970s. The acronym SEQUEL was later changed to SQL because "SEQUEL" was a trademark of the UK-based Hawker Siddeley aircraft company.

In 1986, IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software which later came to be known as Oracle.

SQL Process

When you are executing an SQL command for any RDBMS, the system determines the best way to carry out your request and SQL engine figures out how to interpret the task.

There are various components included in this process. These components are:

- Query Dispatcher
- Optimization Engines
- Classic Query Engine
- SQL Query Engine, etc.

A classic query engine handles all the non-SQL queries, but a SQL query engine won't handle logical files.

SQL Commands

The standard SQL commands to interact with relational databases are **CREATE**, **SELECT**, **INSERT**, **UPDATE**, **DELETE** and **DROP**. These commands can be classified into the following groups based on their nature

SQL Commands

DDL - Data Definition Language

- CREATE
  - Creates a new table, a view of a table, or other object in the database.
- ALTER
  - Modifies an existing database object, such as a table.
- DROP
  - Deletes an entire table, a view of a table or other objects in the database.

SQL Commands

DML - Data Manipulation Language

- SELECT
  - Retrieves certain records from one or more tables.
- INSERT
  - Creates a record.
- UPDATE
  - Modifies records.
- DELETE
  - Deletes records.

SQL Commands

DCL - Data Control Language

- GRANT
  - Gives a privilege to user.
- REVOKE
  - Takes back privileges granted from user.

What is RDBMS?

RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

What is a table?

The data in an RDBMS is stored in database objects which are called as tables.

This table is basically a collection of related data entries and it consists of numerous columns and rows.

Remember, a table is the most common and simplest form of data storage in a relational database.

What is a column?

In the context of a relational database, a column is a set of data values of a particular simple type, one for each row of the table. The columns provide the structure according to which the rows are composed. When a column allows data values of a single type, it does not essentially mean it only has simple text values. Other databases go beyond and let the data be stored as a file on Operating System whereas the column data only covers a pointer or a link to the actual file. Also, databases mostly let columns to have more complex data for example whole documents, images or even video clips.

What is a column?

Fixed length type:

Fields that contain a fixed number of bits are known as fixed length fields. A four byte field for example may contain a 31 bit binary integer plus a sign bit (32 bits in all). A 30 byte name field may contain a person's name typically padded with blanks at the end. The disadvantage of using fixed length fields is that some part of the field may be wasted but space is still required for the maximum length case. Also, where fields are omitted, padding for the missing fields is still required to maintain fixed start positions within a record for instance.

What is a column?

Variable length type:

A variable length field is not always the same physical size. Such fields are nearly always used for text fields that can be large, or fields that vary greatly in length. For example, a bibliographical database has many small fields such as publication date and author name, but also has abstracts, which vary greatly in length. Reserving a fixed-length field of some length would be inefficient because it would enforce a maximum length on abstracts, and because space would be wasted in most records (particularly if many articles lacked abstracts entirely).

What is a column?

Variable length type:

Database implementations commonly store varying-length fields in special ways, in order to make all the records of a given type have a uniform small size. Doing so can help performance. On the other hand, data in serialized forms such as stored in typical file systems, transmitted across networks, and so on usually uses quite different performance strategies. The choice depends on factors such as the total size of records, performance characteristics of the storage medium, and the expected patterns of access.

What is a column?

FLV:

- Pros
  - Faster access
  - Usually small
  - Stored locally into the database files
- Cons
  - Less flexible
  - Wasted space

What is a column?

VLV:

- Pros
  - Flexibility
  - Store binary
- Cons
  - Slower since data are exported to other files.
  - Slower since data need to be parsed.

What is a row?

A row, also called a record or tuple—represents a single, implicitly structured data item in a table. In simple terms, a database table can be thought of as consisting of rows and columns or fields. Each row in a table represents a set of related data, and every row in the table has the same structure.

For example, in a table that represents companies, each row would represent a single company. Columns might represent things like company name, company street address, whether the company is publicly held, its VAT number, etc.. In a table that represents the association of employees with departments, each row would associate one employee with one department.

SQL Constraints

Constraints are the rules enforced on data columns on a table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database.

Constraints can either be column level or table level. Column level constraints are applied only to one column whereas, table level constraints are applied to the entire table.

SQL Constraints

- NOT NULL Constraint − Ensures that a column cannot have a NULL value.
- DEFAULT Constraint − Provides a default value for a column when none is specified.
- UNIQUE Constraint − Ensures that all the values in a column are different.
- PRIMARY Key − Uniquely identifies each row/record in a database table.
- FOREIGN Key − Uniquely identifies a row/record in any another database table.
- CHECK Constraint − The CHECK constraint ensures that all values in a column satisfy certain conditions.
- INDEX − Used to create and retrieve data from the database very quickly.

Data Integrity

The following categories of data integrity exist with each RDBMS –

- Entity Integrity – There are no duplicate rows in a table.
- Domain Integrity – Enforces valid entries for a given column by restricting the type, the format, or the range of values.
- Referential integrity – Rows cannot be deleted, which are used by other records.
- User-Defined Integrity – Enforces some specific business rules that do not fall into entity, domain or referential integrity.

Relational Databases

- SQLite:
  - A very powerful, embedded relational database management system.
- MySQL:
  - The most popular and commonly used RDBMS.
- PostgreSQL:
  - The most advanced, SQL-compliant and open-source objective-RDBMS

Relational Databases

- Object storage
- Implementing behavior within the database
- Concurrency control
- Transaction control
- Enforcing referential integrity

Relational Databases

- Object storage
    - To store an object in a relational database you need to flatten it – create a data representation of the object – because relational databases only store data. To retrieve the object you would read the data from the database and then create the object, often referred to as restoring the object, based on that data.

Relational Databases

- Implementing behavior within the database
    - Behavior is implemented in a relational database via stored procedures and/or stored functions that can be invoked internally within the database and often by external applications. Stored functions and procedures are operations that run within an RDBMS, the difference being what the operation can return and whether it can be invoked in a query. The differences aren't important for our purposes so the term "stored procedure" will be used to refer to both stored functions and stored procedures. In the past stored procedures were written in a proprietary language, such as Oracle's PL/SQL, although now Java is quickly becoming the language of choice for database programming. A stored procedure typically runs some SQL code, massages the data, and then hands back a response in the form of zero or more records, or a response code, or as a database error message.

Relational Databases

- Concurrency control
  - Consider an airline reservation system. There is a flight with one seat left on it, and two people are trying to reserve that seat at the same time. Both people check the flight status and are told that a seat is still available. Both enter their payment information and click the reservation button at the same time. What should happen? If the system is working properly only one person should be given a seat and the other should be told that there is no longer one available. Concurrency control is what makes this happen. Concurrency control must be implemented throughout your object source code and within your database.

Relational Databases

- Transaction control
  - A transaction is a collection of actions on your database – such as the saving of, retrieval of, or deletion of data – which form a work unit. A flat transactions is an “all-or-nothing” approach where all the actions must either succeed or be rolled back (canceled). A nested transaction is an approach where some of the actions are transactions in their own right. These sub-transactions are committed once successful and are not rolled back if the larger transaction fails. Transactions may also be short-lived, running in thousandths of a second, or long-lived, taking hours, days, weeks, or even months to complete. Transaction control is a critical concept for all developers to understand.

Relational Databases

- Enforcing referential integrity
    - Referential integrity (RI) is the assurance that a reference from one entity to another entity is valid. For example, if a customer references an address, then that address must exist. If the address is deleted then all references to it must also be removed, otherwise your system must not allow the deletion. Contrary to popular belief, RI isn't just a database issue, it's an issue for your entire system.

      All developers should understand basic strategies for implementing referential integrity!

SQLite

SQLite is an amazing library that gets embedded inside the application that makes use of. As a self-contained, file-based database, SQLite offers an amazing set of tools to handle all sorts of data with much less constrained and ease compared to hosted, process based (server) relational databases.

When an application uses SQLite, the integration works with functional and direct calls made to a file holding the data (i.e. SQLite database) instead of communicating through an interface of sorts (i.e. ports, sockets). This makes SQLite extremely fast and efficient, and also powerful thanks to the library's underlying technology.

SQLite

- Advantages of SQLite
  - File based
  - Standards-aware
  - Great for developing and even testing
- Disadvantages of SQLite
  - No user management
  - Lack of possibility to tinker with for additional performance

MySQL

MySQL is the most popular one of all the large-scale database servers. It is a feature rich, open-source product that powers a lot of web-sites and applications online. Getting started with MySQL is relatively easy and developers have access to a massive array of information regarding the database on the internet.

Despite not trying to implement the full SQL standard, MySQL offers a lot of functionality to the users. As a standalone database server, applications talk to MySQL daemon process to access the database itself -- unlike SQLite.

MySQL

- Advantages of MySQL
  - Easy to work with
  - Scalable and powerful
  - Feature rich
  - Secure
  - Speedy
- Disadvantages of MySQL
  - Known limitations
  - Reliability issues
  - Stagnated development (since owned by Oracle)

PostgreSQL

PostgreSQL is the advanced, open-source [object]-relational database management system which has the main goal of being standards-compliant and extensible. PostgreSQL, or Postgres, tries to adopt the ANSI/ISO SQL standards together with the revisions.

Compared to other RDBMSs, PostgreSQL differs itself with its support for highly required and integral object-oriented and/or relational database functionality, such as the complete support for reliable transactions, i.e. Atomicity, Consistency, Isolation, Durability (ACID).

PostgreSQL

Due to the powerful underlying technology, Postgres is extremely capable of handling many tasks very efficiently. Support for concurrency is achieved without read locks thanks to the implementation of Multiversion Concurrency Control (MVCC), which also ensures the ACID compliance.

PostgreSQL is highly programmable, and therefore extendible, with custom procedures that are called "stored procedures". These functions can be created to simplify the execution of repeated, complex and often required database operations.

PostgreSQL

Although this DBMS does not have the popularity of MySQL, there are many amazing third-party tools and libraries that are designed to make working with PostgreSQL simple, despite this database's powerful nature. Nowadays it is possible to get PostgreSQL as an application package through many operating-system's default package manager with ease.

PostgreSQL

- Advantages of PostgreSQL
  - An open-source SQL standard compliant RDBMS
  - Strong community
  - Strong third-party support
  - Extensible
- Disadvantages of PostgreSQL
  - Popularity
  - Hosting
  - Complex configuration
  - Performance (on read intensive tasks)

Entity relationship diagrams

There are several notations for data modeling. The actual model is frequently called "Entity relationship model", because it depicts data in terms of the entities and relationships described in the data. An entity-relationship model (ERM) is an abstract conceptual representation of structured data. Entity-relationship modeling is a relational schema database modeling method, used in software engineering to produce a type of conceptual data model (or semantic data model) of a system, often a relational database, and its requirements in a top-down fashion.

Entity relationship diagrams

- Identify the entities
  - The first step in making an ERD is to identify all of the entities you will use. An entity is nothing more than a rectangle with a description of something that your system stores information about. This could be a customer, a manager, an invoice, a schedule, etc. Draw a rectangle for each entity you can think of on your page. Keep them spaced out a bit.

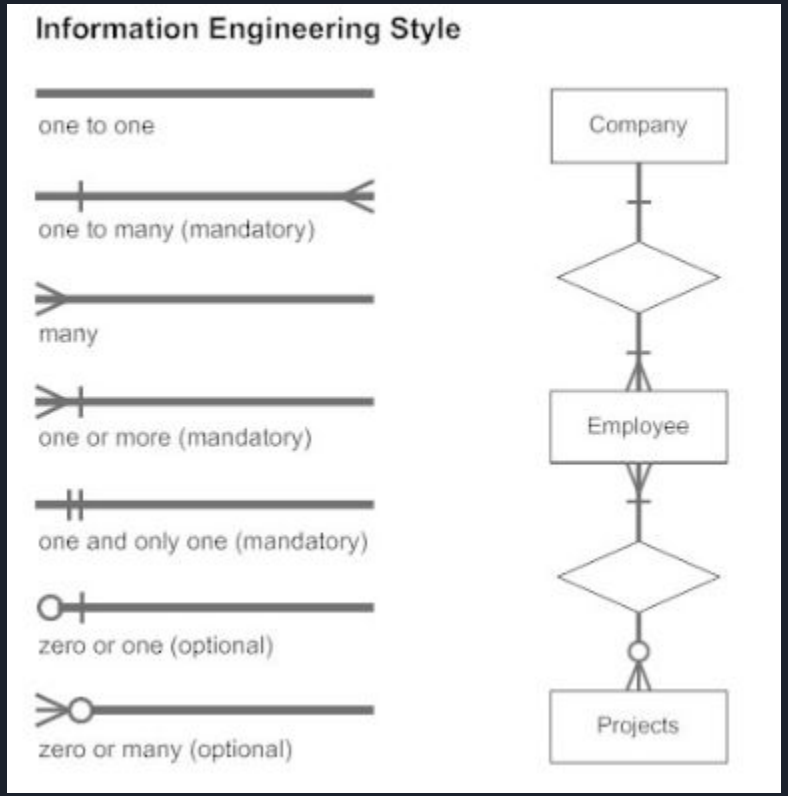
Entity relationship diagrams

- Identify the entities
    - The first step in making an ERD is to identify all of the entities you will use. An entity is nothing more than a rectangle with a description of something that your system stores information about. This could be a customer, a manager, an invoice, a schedule, etc. Draw a rectangle for each entity you can think of on your page. Keep them spaced out a bit.
- Identify relationships.
    - Look at two entities, are they related? If so draw a solid line connecting the two entities.

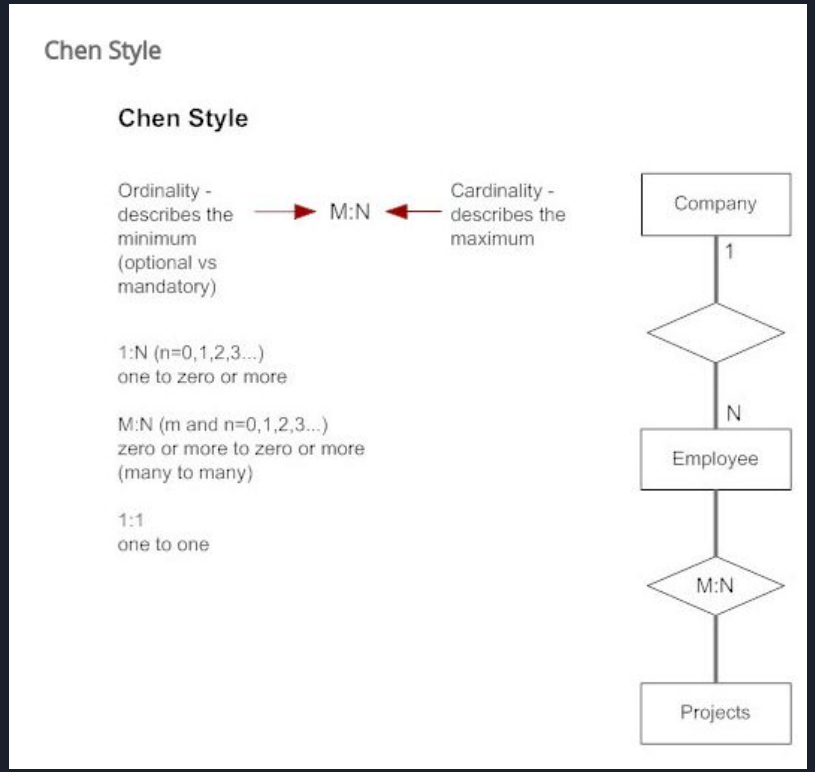
Entity relationship diagrams

- Describe the relationship.
  - How are the entities related? Draw an action diamond between the two entities on the line you just added. In the diamond write a brief description of how they are related.
- Add attributes.
  - Any key attributes of entities should be added using oval-shaped symbols.
- Complete the diagram.
  - Continue to connect the entities with lines, and adding diamonds to describe each relationship until all relationships have been described. Each of your entities may not have any relationships, some may have multiple relationships. That is okay.

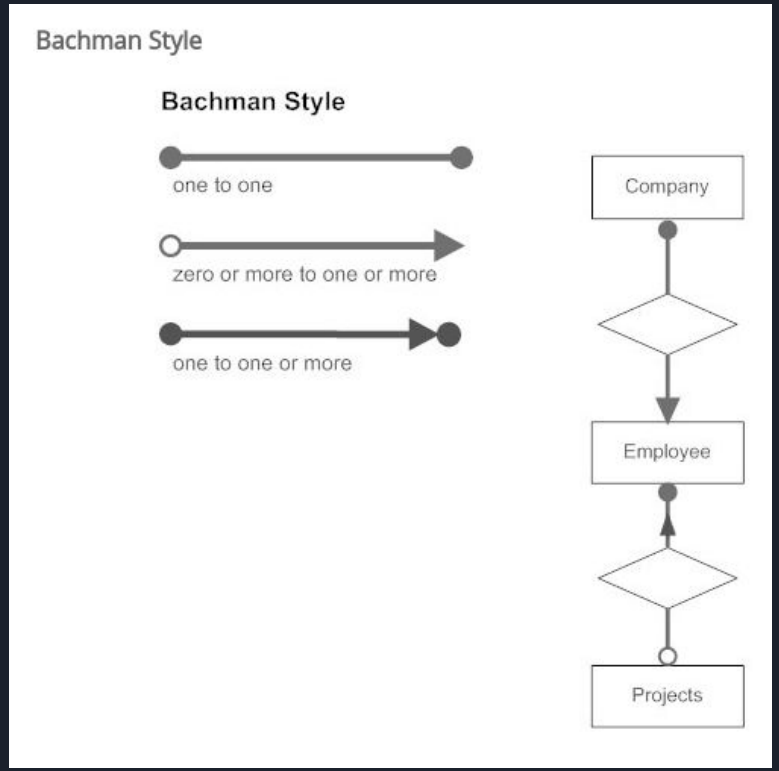
Entity relationship diagrams



Entity relationship diagrams



Entity relationship diagrams



Entity relationship diagrams