INFO 5100
Application Engineering and Development

Week 9

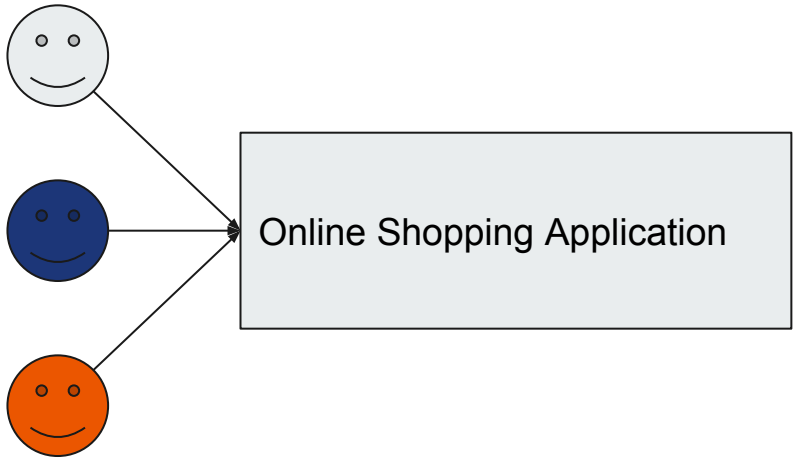
Agenda

- **Database Use Cases**
- Intro to Relational Database
- Work with Relational Database
- Advanced Concepts

Sample Application

- An online shopping application
- User can login and make purchases
- Product manager can login and add new products
- Accountant can login and aggregate data
- Business Analytics can login and perform analysis

Where to Store Data?



Where to Store Data?

- Client Side vs Server Side
  - Cannot store the data on client side, because different user will have different view of the data
  - Store on server side to keep a consistent view of data
- Memory vs Hard Disk
  - Cannot store the data in memory for long time, because if server goes down, data is gone
  - Store in hard disk for long term storage

- For reliable data storage, store in hard disk for persistence (持久化)

How to Store Data?

- Data can be stored in hard drive in many formats.
  - Store everything in one file
  - Store in different files
  - Store in JSON? XML? CSV?

- Data should be stored in a way that is easy to access
  - Retrieval
  - Aggregate
  - Filter

Database System

- Database System is a type of software that stores data for users in a centralized location, with support for
  - Create
  - Read
  - Update
  - Delete
- Might also support advanced functionalities such as
  - Aggregation 数据聚合
  - Searching

Types of Database System

- There are many ways to classify Database Systems, one way to classify Database Systems are RDBMS vs NoSQL
- Relational Database
  - MySQL
  - SQL Server
- NoSQL Database
  - Apache Cassandra
  - Amazon DynamoDB
  - LevelDB
  - Redis
  - Druid

Agenda

- Database Use Cases
- **Intro to Relational Database**
- Work with Relational Database
- Advanced Concepts

Relational Database (RDBMS)

- Relational Database is a type of Database Systems where data are modelled as related entities
- Theoretically based on relational algebra
- Besides CRUD operations, RDBMS normally also has nice support on
  - Transaction
  - ACID 文本
  - Concurrency Control
  - Referral Integrity
- Popular among small/medium sized application, can support high concurrency access
  - Wikipedia is powered by MySQL
  - Facebook newsfeed is built on top of MySQL

Table

- In online shopping application, in order to function, there will be multiple entities
  - Customers
  - Products
- In RDBMS, each entity are stored in a table

- Accountants has been doing this for long time, for example, Excel Sheets

Table

- For example, we can describe all the Products logically in a table

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19

Header

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19

Row

- Row represent a complete record for one entity

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19

Column

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19

- Column repr es for a column

Data Types

- The previous table is a logic view, RDBMS actually also stores the type of data
- Data in the table can have different types
  - ProductID is a number
  - ProductName is a string
  - But data in Column have to be in same data type
- Different RDBMS might support different data types, but generally they all support
  - Numeric
  - Date and Time
  - String

Data Types

- MySQL for example, supports the following data types
  - Numeric
  - Date and Time
  - String
- SQL Server, besides support the listed types, also support types such as
  - Image
  - XML

- RDBMS systems are implemented differently, especially how data are eventually stored in hard disk

Key

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19

Key

- In RDBMS there are some important types of Key
  - Primary Key
  - Foreign Key

Primary Key

- Primary Key is type of Key, that must has following attribute
  - Unique
  - Not empty
  - Table must has a Primary Key

1. unique
2. not empty
3. a table must have a primary key

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19

Foreign Key

- Foreign Key is type of Key, that can be used to infer relationships between Tables
  - Foreign Key for one Table is the Primary Key of another Table

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19

Foreign Key

- In Products Table, there is a Foreign Key to Suppliers Table

SupplierID	SupplierName	ContactName	Address	City	PostalCode
1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	Londona	EC1 4SD
2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	New Orleans	70117

Agenda

- Database Use Cases
- Intro to Relational Database
- **Work with Relational Database**
- Advanced Concepts

Access Relational Database

- There are normally drivers to help developers to manipulate data in RDBMS programmatically
  - ODBC (Open Database Connectivity)
  - JDBC (Java Database Connectivity)
  - Etc

- Most of the drivers supports interact with RDBMS through SQL

SQL

- It is a standard, rather than a language designed for specific RDBMS
- Most common RDBMS supports SQL
- Easy to learn, only handful of keywords, but highly flexible and can be very complex
- Some RDBMS implemented extra query functions on top of SQL, so true to avoid those functions

- Playground: https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

SQL - DDL

- DDL: Data Definition Language
- Used to define, change, and delete Table, View, Index, etc

```
CREATE TABLE Suppliers (
      SupplierID INT NOT NULL AUTO_INCREMENT,
      SupplierName VARCHAR(32) NOT NULL DEFAULT "",
      ContactName VARCHAR(32) NOT NULL DEFAULT "",
      Address VARCHAR(32) NOT NULL DEFAULT "",
      City VARCHAR(32) NOT NULL DEFAULT "",
      PostalCode VARCHAR NOT NULL DEFAULT "",
      PRIMARY KEY ( SupplierID )
);
```

SQL - DML

- DML: Data Manipulation Language
- Used to insert, update, and delete data against RDBMS

```
INSERT INTO Products VALUES (78, "test", 1, 1, "some unit", 200);
```

- For example, to insert data into Products table

```
UPDATE Products SET ProductName="test2", Unit="other unit" WHERE ProductID=78;
```

- For example, to update certain fields

```
DELETE FROM Products WHERE ProductID=78;
```

SQL - DQL

- DQL: Data Query Language
- Used to query data from RDBMS

```
SELECT * FROM Products;
```

- For example, to find all the Products

```
SELECT ProductName, SupplierID from Products WHERE ProductID=3;
```

- For example, to find particular information

```
SELECT count(*) from Products WHERE Price > 2;
```

SQL - DQL

- The DQL statement is quite powerful and can do more things

```
SELECT CustomerID FROM Orders GROUP BY CustomerID HAVING count(*) > 5;
```

- For example, to find all the CustomerID for the Customers that have more than 5 purchases

- For example, to find all the CustomerNames for the Customers that have more than 5 purchases

```
SELECT CustomerName
FROM Customers
WHERE CustomerID IN (SELECT CustomerID FROM Orders GROUP BY CustomerID HAVING count(*) > 5);
```

Agenda

- Database Use Cases
- Intro to Relational Database
- Work with Relational Database
- **Advanced Concepts**

Transaction

- A unit of work performed within a Database System
- The goal is to provide better data integrity
  - Allow easier recovery from failures
  - Isolation between programs accessing database concurrently

- For example, a user is purchasing in our online shopping application and spent 100 USD.
  - 100 USD need to be credited from buyer's account
  - 100 USD need to be added to seller's account
  - If any of the operation fails, both operations need to reverted

ACID

- Atomicity
  - Requires each transaction be “all or nothing”. A transaction might have multiple parts, if one of them fail, then the entire transaction fails.
- Consistency (Correct) may less speed
  - Ensures after the transaction database is updated from one valid state to another.
- Isolation
  - Ensures concurrent execution of transactions are executed correctly and not interfere with each other
- Durability
  - Once the transaction has been committed, it will remain that way even in the event of power loss, etc

Concurrency Control

- Database systems are normally accessed concurrently, without concurrency control, programs might overwrite each other's data.
- There are many ways to implement concurrency control
  - Locking
  - Timestamp Ordering
  - Commitment Ordering
  - Multiversion Concurrency Control

- MVCC has gain a lot of attention because of good performance

Referral Integrity

- Ensures a reference from one entity to another is valid, this can be enforced through relational algebra

- For example: When a customer purchase something from online shopping application, the database record should never contain non-exist product information.

Database Engine

- Database Engi
- Interface betv