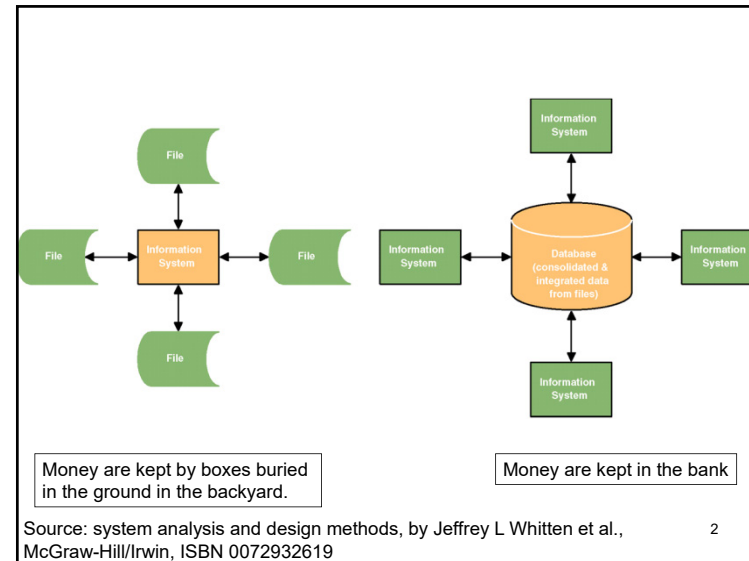


# Lecture 8

## SQL (I)



## Database vs. Files

- File – a collection of similar records.
  - Files are unrelated to each other except in the code of an application program.
  - Data storage is built around the applications that use the files.
  - Data are retrieved through file-reading of format only known to applications using the data file.
- Database – a collection of interrelated files
  - Records in one file (or table) are physically related to records in another file (or table).
  - Applications are built around the integrated database
  - Data are retrieved through a standardized interface or format

## Introduction to SQL database management systems (DBMS)

- Database is the bank for data.
- Use SQL language as the interface for storing & retrieving data.
- Many database products exist.
  - Commercial: Oracle Database 12c, IBM db2, Microsoft SQL server, ...
  - Free & Open source: Oracle MySQL, MariaDB, PostgreSQL, SQLite, ...
- One DBMS manages many databases, one database contains many tables, one table contains many rows & columns.

## Assignment #4

Due: 11/19/2015

- This assignment continues the development of your e-business software system.
  - We have done requirement specification and software estimation
- In this assignment, you are required to design and document your system using **UML diagrams** for software systems and **ER diagrams** for data modeling.
- The ER model is created for discovering business rules. The model then turns into your physical data model, and then becomes the blueprint for implementing the database (tables and fields).
- ER model is often further developed into class models.
- UML diagrams are used to design the system. They are then discussed to discover issues before actual implementation.

5

## Assignment #4

Due: 11/19/2015

- Please complete this assignment with software products such as Microsoft Visio (or others, no hand-drawings, pls.). The following diagrams are required for each of the software (2-tier and 3-tier) to be developed:
  - Use case diagrams (you have done it, hopefully)
  - Class diagrams
  - Deployment diagrams
  - A sequence diagram for online ordering used parts.
  - A sequence diagram for online posting of used parts to be sold.
  - A state diagram for shopping carts in the system.
- ER model should be the same for both 2-tier and 3-tier systems.
- Please do both UML diagrams for your core systems, plus two more functions in your high-priority function list.

6

## Assignment #4

Due: 11/19/2015

- Enumerate all business rules from your ER model, and make sure the generated business rules are reasonable, logical, and fulfills your business needs.
- The class diagram should focus on online ordering & posting, and it should be somehow extended from E-R model.

7

## SQL (I)

8

## Basic SQL Operations - Outline

- Database Creation & Connection
- Table & Index Creation
- Changing/Deleting Objects
- Basic Data Manipulation
  - Data insertion
  - Data retrieval
  - Data deletion and updating
- Advanced Select

9

## Introduction to SQL

- SQL
  - Structured Query Language
  - an ANSI standard computer language for accessing and manipulating databases.
  - Retrieve, insert, delete, update data against a database
  - CRUD – Creation, Retrieval, Updating, and Deletion.
- Not a complete language like Java, C++, ...
  - SQL is a sub-language of about 30 statements
  - Usually embedded in other languages or tool for database access
  - Portable across operating systems
  - Somewhat portable among DBMS vendors

Source: MIT 1.264 lecture notes

10

## Database Creation & Connection

- Create a new database
  - `CREATE DATABASE databaseName;`
- Use a database
  - `USE databaseName;`
- Show available databases; (MySQL)
  - `SHOW DATABASES;`

11

## Table & Index Creation

- Create a table
  - `CREATE TABLE tableName (  
    field1 datatype [(length)] [NULL, NOT NULL],  
    field2 datatype [(length)] [NULL, NOT NULL],  
    ...);`
  - `DESCRIBE tableName; (MySQL)`
- Create an index
  - `CREATE [UNIQUE] INDEX indexName ON tableName  
    (columnName)`

12

## Advanced Table Creation

```
CREATE TABLE tableName (  
  field1 datatype [(length)] [NULL, NOT NULL]  
  [PRIMARY KEY] [UNIQUE]  
  [DEFAULT ...], [CHECK ...], [FOREIGN KEY...]  
  ...);
```

- **PRIMARY KEY**: specify the column is used as a primary key
- **UNIQUE**: the values in the column should be unique
- **DEFAULT**: specify the default value if not defined
- **CHECK**: constrains the value of the column
  - CHECK (title\_id LIKE '[A-Z][A-Z][0-9][0-9][0-9][0-9]')
- **FOREIGN KEY**: Ensure the referential integrity of the data in one table to match values in another table

13

## Index

- Why use index?
  - Fast data retrieval and sorting
  - Unique index ensures the values of the indexed column are unique
- Why not use index?
  - Performance penalty on data insertion, deletion, update
- When?
  - On columns with frequent retrievals
  - On columns used to join other tables (to be covered)
  - On columns accessed in sorted orders

14

## Changing and Deleting Objects

- Change table definition
  - ALTER TABLE tableName [ADD fieldName datatype ...][DROP fieldName]
- Delete a table
  - DROP TABLE tableName
- Delete a database
  - DROP DATABASE databaseName
- Delete an index
  - DROP INDEX tableName.indexName
  - DROP INDEX indexName on tableName (MySQL)

15

## Data Insertion

- Inserting data
  - INSERT INTO tableName [(col1, col2, ...)] VALUES (val1 [, val2, ...]);
  - Not all data attributes (field values) need to present
  - DBMS will use default values for absent data fields
- Show data in a table
  - SELECT \* FROM tableName
- *Demonstration with MS-SQL through SQL Server Management Studio.*

16

## SQL syntax

- Free form, case insensitive

```
Create database dbname;
Create table newTable (
    f1 int primary key,
    f2 int
);
alter table newTable add f3 int;
alter table newTable drop column f2;
drop table newTable;
drop database dbname;
```

17

## Reviews

```
insert into newTable values (1, 3);
insert into newTable values (2, 4);
insert into newTable(f1) values (3);
insert into newTable (f1, f3) values (4, 3);
insert into newTable (f3, f1) values (3, 4);
insert into newTable (f3) values (5);
```

18

## Data Deletion and Updating

- Delete data

- `DELETE FROM` tableName `WHERE` [criteria]
- `DELETE FROM` newTable `WHERE` f1 < 3;

- Updating data

- `UPDATE` tableName `SET` fieldName=newValue [, fieldName1=newValue1, ...];  
(NOTE! This will update ALL records in the table, use `WHERE` to limit records to be updated.)
- `UPDATE` tableName `SET` [...] `WHERE` [criteria]
- `UPDATE` newTable `SET` f3 = 5;
- `UPDATE` newTable `SET` f3 = f3 \* 2 `WHERE` f1 < 3;

19

## Examples

- Find the average sale
  - `SELECT AVG(Amt) FROM` Orders;
- Find the average sale for a customer
  - `SELECT AVG(Amt) FROM` Orders `WHERE` Cust = 211;
- Add an office
  - `INSERT INTO` Offices (OfficeNbr, City, Region, Target, Sales) `VALUES` ('55', 'Dallas', 'West', 200000, 0);
- Delete a customer
  - `DELETE FROM` Customers `WHERE` Company = 'Connor Co';
- Raise a credit limit
  - `UPDATE` Customers `SET` CreditLimit = 75000 `WHERE` Company = 'AmaratungaEnterprise';

20