

CPET 499/ITC 250 Web Systems

Chapter 11 Working with Databases Part 1 of 3

Text Book:

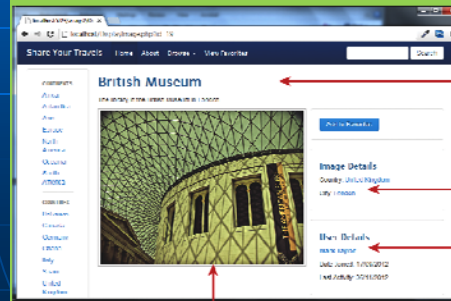
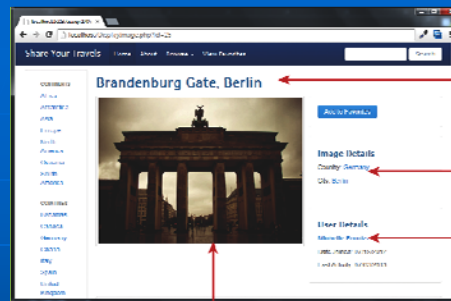
* Fundamentals of Web Development, 2015, by Randy Connolly and Ricardo Hoar, published by Pearson

Paul I-Hai Lin, Professor of ECET
<http://www.etcs.ipfw.edu/~lin>

Topics

- Database and Web Development
 - The Role of Database in Web Development
 - Database Design
 - Database Options
- SQL (Structure Query Language)
 - SELECT Statement
 - INSERT, UPDATE, and DELETE Statements
 - Transactions: Local Transactions, Distributed Transactions
 - Data Definition Statements
 - Database Indexes and Efficiency

Figure 11.1
Separating
content from
data



Content (data)
varies but the
markup (design)
stays the same.

Figure 11.2 How website use databases

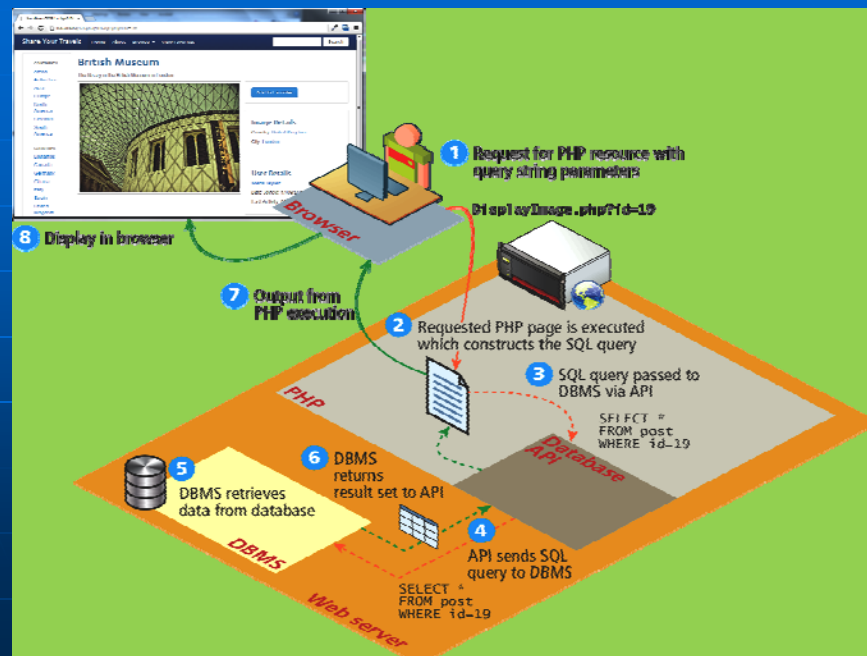


Figure 11.3 A database table

ArtWorkID	Title	Artist	YearOfWork
345	The Death of Marat	David	1793
400	The School of Athens	Raphael	1510
408	Bacchus and Ariadne	Titian	1520
425	Girl with a Pearl Earring	Vermeer	1665
438	Starry Night	Van Gogh	1889

CPET 499/ITC 250 Web Systems, Paul I. Lin

5

Figure 11.4 Diagramming a table

ArtWorks		ArtWorks		ArtWorks
ArtWorkID INT Title VARCHAR Artist VARCHAR YearOfWork INT		PK <u>ArtWorkID</u> Title Artist YearOfWork		<u>ArtWorkID</u> Title Artist YearOfWork

CPET 499/ITC 250 Web Systems, Paul I. Lin

6

Table 11.1 Common Database Table Data Types

- BIT – BOOLEAN or BOOL
- BLOG – Binary Large Object (images and other data objects)
- CHAR(n) – a fixed number of character (n = number of chars) that are padded with spaces to fill the field
- DATE – also TIME and DATETIME data types
- FLOAT - also DOUBLE, DECIMAL data types
- INT - also SMALLINT data type
- VARCHAR(n) – a variable number of characters (n = maximum number of chars with no space padding)

Figure 11.5 Foreign keys links tables

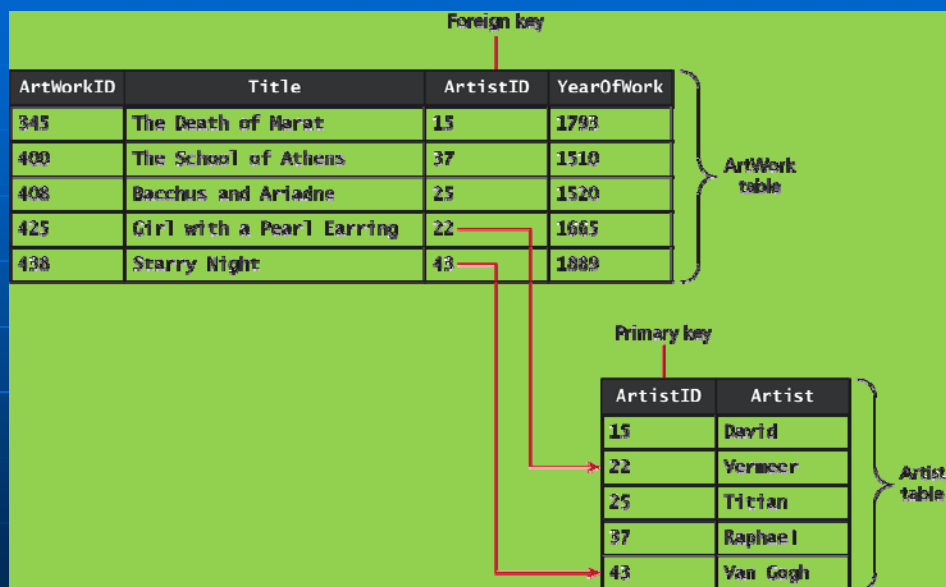


Figure 11.6 Diagramming a one-to-many relationship



Figure 11.7 Implementing a many-to-many relationship

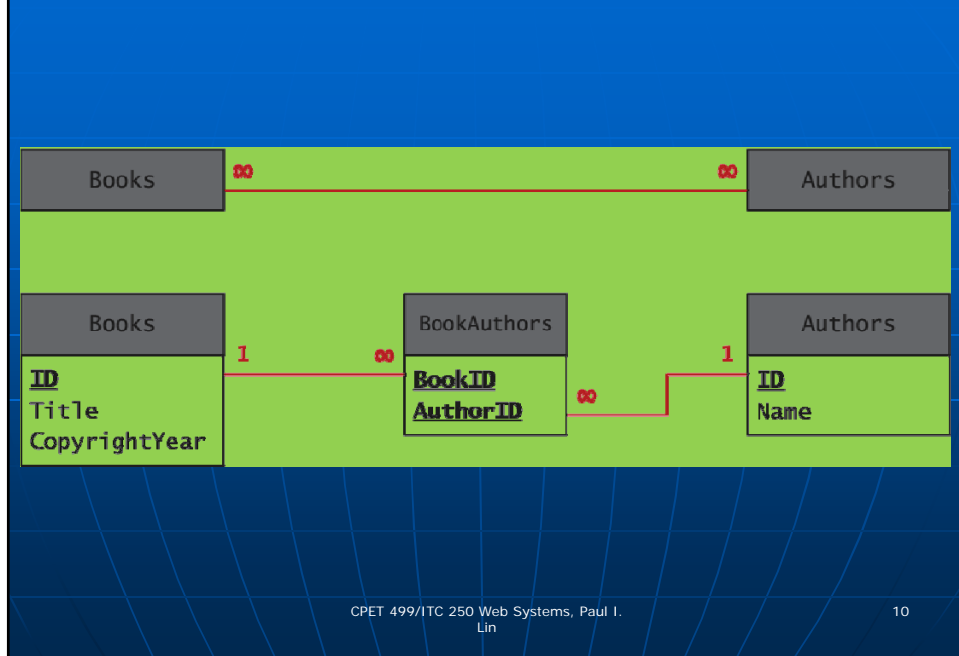


Figure 11.8 Database in the Enterprise

- SQL Databases
 - Financial and order management systems
 - Manufacturing system
 - Point of sales
 - Groupware and file servers
- Examples
 - DB2
 - Oracle DB
 - Microsoft SQL
 - MySQL
 - PostgreSQL
 - etc

CPET 499/ITC 250 Web Systems, Paul I.
Lin

11

Figure 11.8 Database in the Enterprise

- Non-SQL Databases
 - Documentation oriented, No relationship among stored data
 - Optimized to retrieve data using simple **Key-Value** syntax similar to PHP associative arrays
 - May be very large data sets
 - Examples:
 - Web server logs, geographical data
 - Twitter posted information
 - Non-SQL DB examples:
 - Couch DB, monoDB
 - Amazon SimpleDB , Google's Big Table

CPET 499/ITC 250 Web Systems, Paul I.
Lin

12

Figure 11.8 Database in the Enterprise

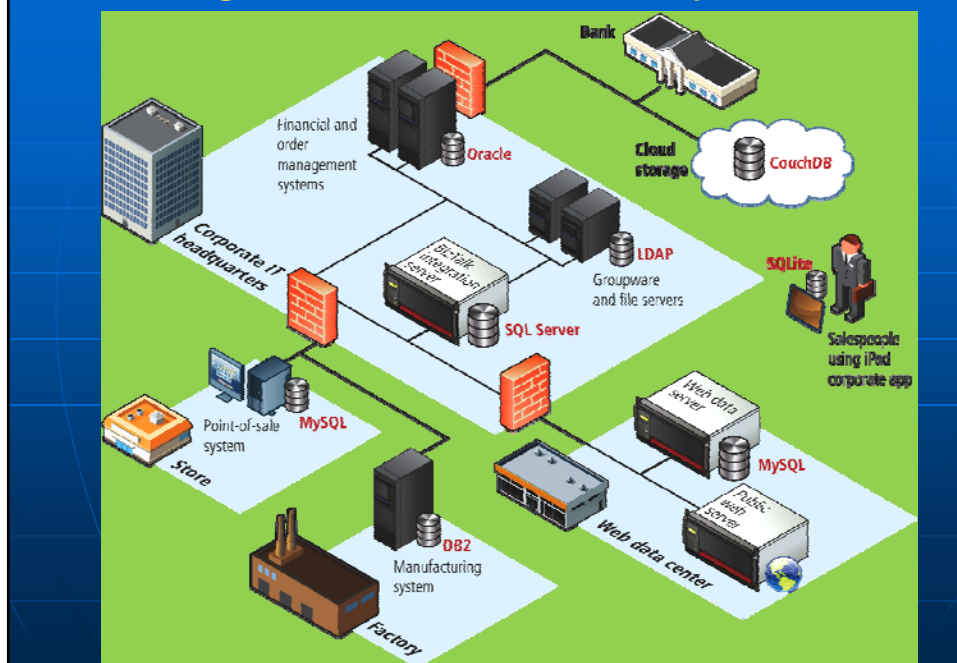
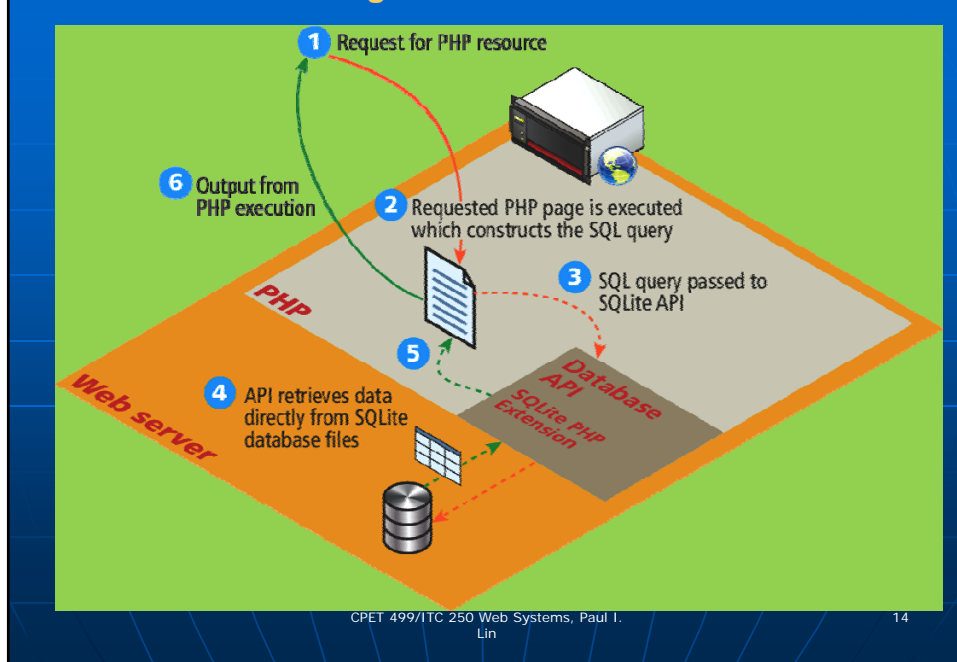


Figure 11.9 SQLite



SQL

- **SELECT** statement – retrieve data from the database
- **INSERT** statement – add new records
- **UPDATE** statement – update existing records
- **DELETE** statement – delete existing records

**Figure
11.10 SQL
SELECT
from a
single table**

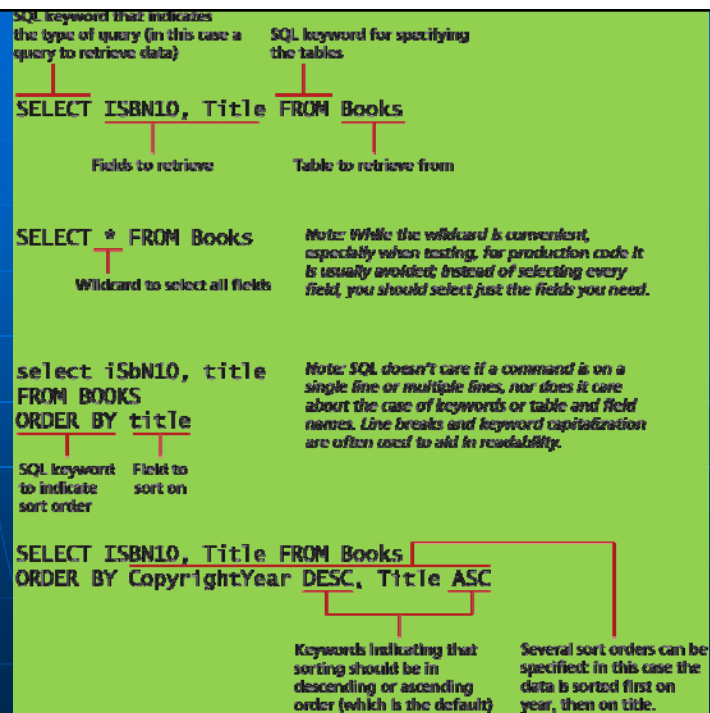


Figure 11.12 Using the WHERE clause

```
SELECT isbn10, title FROM books
WHERE copyrightYear > 2010
```

SQL keyword that indicates
to return only those records
whose data matches the
criteria expression

Expressions take form:
field operator value

```
SELECT isbn10, title FROM books
WHERE category = 'Math' AND copyrightYear = 2014
```

Comparisons with strings require string
literals (single or double quote)

Figure 11.12 SQL SELECT from multiple table using an INNER JOIN

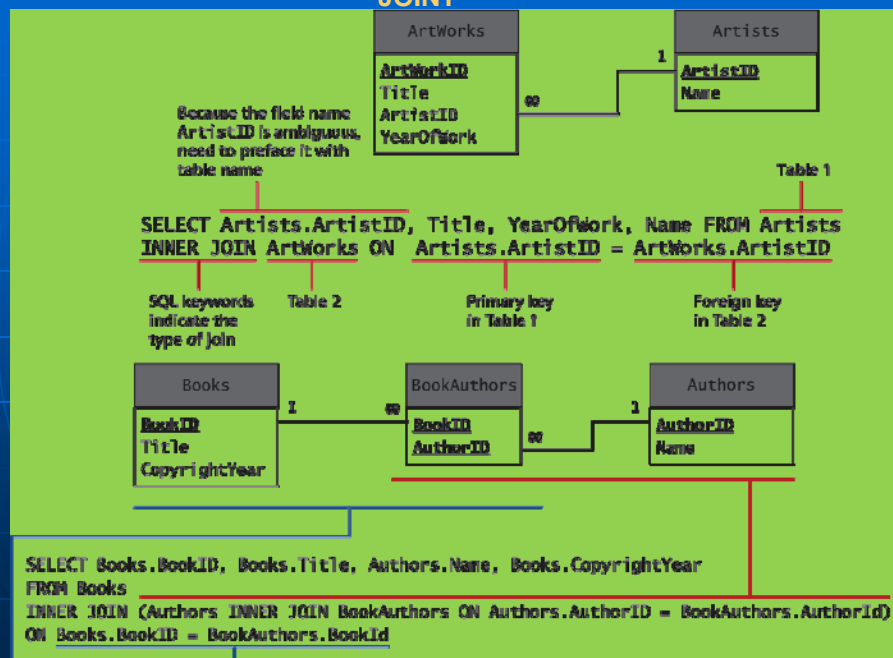


Figure 11.13 Using GROUP BY with aggregate functions

This aggregate function returns a count of the number of records. Defines an alias for the calculated value

```
SELECT Count(ArtWorkID) AS NumPaintings
FROM ArtWorks
WHERE YearOfWork > 1900
```

Note: This SQL statement returns a single record with a single value in it.

NumPaintings
745

Count number of paintings after year 1900

```
SELECT Nationality, Count(ArtistID) AS NumArtists
FROM Artists
GROUP BY Nationality
```

Note: This SQL statement returns as many records as there are unique values in the group-by field.

Nationality	NumArtists
Belgium	4
England	15
France	36
Germany	27
Italy	53

SQL keywords to group output by specified fields

Figure 11.14 SQL INSERT, UPDATE, and DELETE

SQL keywords for inserting (adding) a new record Table name Fields that will receive the data values

```
INSERT INTO ArtWorks (Title, YearOfWork, ArtistID)
VALUES ('Night Watch', 1642, 105)
```

Note: Primary key fields are often set to **AUTO_INCREMENT**, which means the DBMS will set it to a unique value when a new record is inserted.

Values to be inserted. Note that string values must be within quotes (single or double).

```
INSERT INTO ArtWorks
SET Title='Night Watch', YearOfWork=1642, ArtistID=105
```

Nonstandard alternate MySQL syntax, which is useful when inserting record with many fields (less likely to insert wrong data into a field)

```
UPDATE ArtWorks
SET Title='Night Watch', YearOfWork=1642, ArtistID=105
WHERE ArtWorkID=54
```

It is essential to specify which record to update, otherwise it will update all the records!

Specify the values for each updated field. Note: Primary key fields that are **AUTO_INCREMENT** cannot have their values updated.

```
DELETE FROM ArtWorks
WHERE ArtWorkID=54
```

It is essential to specify which record to delete, otherwise it will delete all the records!

Local and Distributed Transactions

■ Transactions

- A transaction refers to a sequence of steps that are treated as a single unit, and provide a way to gracefully handle errors and keep your data properly consistent when error occur.
- Local Transactions: transactions handled by the DBMS
- Distributed Transactions: transactions involve multiple hosts, several of which we may have no control over.

Local and Distributed Transactions

■ Web Storefront Example

- A customer: Checkout, Verified address, entered credit card, select shipping option
- Click Pay for Order Button, what happen?
 1. Write order records to the website database.
 2. Check credit card service to see if payment is accepted.
 3. If payment is accepted, send message to legacy ordering system.
 4. Remove purchased item from warehouse inventory table and added to the order shipped table.
 5. Send message to shipping provider

Local and Distributed Transactions

- Web Storefront Example
- Distributed Transactions
 - Local database writes
 - External credit card processor
 - External legacy ordering system
 - External shipping systems

Local Transactions

- Through SQL Statements or API calls
 - START TRANSACTION
 - COMMIT
 - ROLLBACK

- LISTING 11.1

START TRANSACTION

INSERT INTO orders ...

INSERT orderDetails ...

UPDATE inventory ...

COMMIT

Figure 11.15 Distributed transaction processing

