Csci 4131

Intro to Express (Framework for Developing Node.js applications)

Lecture 21, November 14th

Fall 2018

Dr. Dan Challou

Logistics

- Homework 6 (Login, Sessions, Database (SQL/MSQL) is out, and is due:

Tuesday November 27th at 2pm

- Grades for Homework 5 should be posted on Moodle. If you have an issue, go see a TA at any office hour, and if you can't make an hour this week, send an email to the help email, and then go see a TA at any office hour next week.
- Finally – the course schedule (posted on moodle) has been updated. Be sure to review it!!!!

Reading and Tutorials: Express, SQL, MySQL

- Express – framework for developing applications in Node.js
  - https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm
  - https://expressjs.com/
    - See the menu items – Getting Started, Guide, API Reference, Advanced Topics, Resources.
- SQL / MYSQL
  - Sebesta Chapter 13
  - https://www.w3schools.com/sql/
  - https://www.w3schools.com/sql/sql_ref_mysql.asp
  - https://www.w3schools.com/nodejs/nodejs_mysql.asp
  - PHP and MYSQL: SQL, MySQL Chapter 17 pp.542-565; Chapter 18

- Older:
  - Node.js
    - https://www.w3schools.com/nodejs/default.asp
    - https://www.tutorialspoint.com/nodejs/
    - https://nodejs.org/en/docs/guides
  - AJAX
    - Sebesta – Chapter 10
    - https://www.w3schools.com/xml/ajax_intro.asp

Express and Node.js training videos also available on Lynda

- You can access Lynda via the following link:
  http://lynda.umn.edu

Use your x.500 id and password to sign in.

The following videos are most helpful:

- 1. Node.js Essential Training (6h 22m - Detailed Node.js video)
- 2. Building a Website with Node.js and Express.js (3h 16m - Focuses on Express.js and Node.js)
- 3. Learning Node.js (1h 57m)

Last Lecture

- HW 6 Intro
- Intro to Express (Node.js development Framework)

Questions?

Today

- Relational Databases Revisited
  - SQL Overview / Refresher

Lectures follow my suggested HW 6 POA

- Translate old HW 5 to express version of HW5
- Re-aquaint myself with SQL / MySQL; research how to connect, query, and use SQL / MySQL; Understand how to interface Node.js / express with a MySQL Database
- Understand Express Sessions, update old HW 5 to use them (can implement a login page with JSON)
- Replace interface to JSON in HW 5 with interface to MySQL (Do SELECT and Insert Queries and use the results)

For HW 6

- Need to use the following technologies
  - Node.js / javascript
  - Express (Node.js framework)
  - SQL
  - MYSQL
  - Sessions (enabled by express)

Today

- SQL / MySQL
- Sessions

Review Exercise –

- Replace Node.js code to serve up the form to add a calendar entry (From HW5) with express code to do the same thing

```
const httpServer = http.createServer(function (req, res) {  switch(req.method) {
        // request for the addPlace.html page (form)
          if(req.url === '/addPlace.html') {
                getAddPlacePage(req, res);

function getAddPlacePage(req, res) {
  fs.readFile('client/addPlace.html', function(err, html) {
          if(err) {     throw err;    }
          res.statusCode = 200;
          res.setHeader('Content-type', 'text/html');
          res.write(html);    res.end();
          });
}
```

Relational Database Model

- Data is organized as a table--rows and columns.

- Each row in a table represents some related set of data items.

- Primary key: One of the column values is used for indexing in the table.

  This value is unique for each row.

- A database may contain multiple tables, with some values common in different tables.

  The values specify a relationship between different tables

- SQL (Structured Query Language) is used to query or update the tables.

Use a Relational Database When:

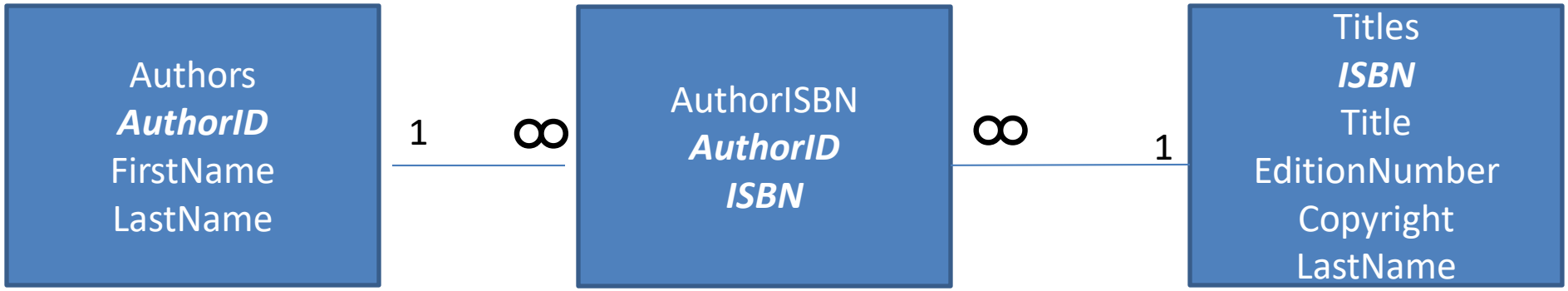
• You have relational data, e.g., you have a customer who purchases your products and those products have a supplier and manufacturer.

• You have large amounts of data and you need to be able to locate relevant information quickly.

• You need to start worrying about issues such as: scalability, reliability, ACID compliance. In computer science, ACID (Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantee that database **transactions** are processed reliably.

• You need to use reporting or intelligence tools to work out business problems.

Step 1 – Design your Database
Entity – Relationship Diagram
Tables for a Book Publisher Database



Primary Key:
AuthorID

Primary Key:
Composite:
AuthorID + ISBN

Primary Key
ISBN

- Three tables identified / defined in our example:
  - Authors table
  - AuthorISBN table
  - Titles table

Example: Authors Table (1)

Three fields are defined:

authorID:	**This is the primary key, defined as Integer; auto-increment assigns next integer value for this field whenever a new row is added. This field value has to be unique for each row.**
firstName:	**a String containing author's first name**
lastName:	**a String containing author's last name**

Example: Author's Table (2)

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Kate	Steinbuhler
5	Sean	Santry
6	Ted	Lin
7	Praveen	Sadhu
8	David	McPhie
9	Cheryl	Yaeger
10	Marina	Zlatkina
11	Ben	Wiedermann
12	Jonathan	Liperi

AuthorISBN table

- isbn String containing the ISBN of a book

- authorID ID number of one of the authors of a book whose ISBN is stored in the row

In this table, these two fields together form a unique value and therefore they are a composite primary key

AuthorISBN table

isbn	authorID
0130125075	1
0130125075	2
0130161438	1
0130161438	2
0130161438	3
0130284173	3
0130284173	6
0130284173	7
0130284181	8

Titles table

isbn	String
title	String
editionNumber	String
copyright	Year of copyright -Integer

One row of the Titles table

isbn	0132575663
title	Java How to Program
editionNumber	9
copyright	2012

Structured Query Language (SQL)

- Create a new table
- Query the database
  - Read a table's rows matching some criteria
  - Select a subset of rows
  - Make queries spanning multiple tables
    - Join Operation
- Update the database
  - Insert data into a table
  - Update the data in a table

SQL Keywords

- SELECT — Select (retrieve) columns form one or more tables
- FROM — Specifies tables
- WHERE — Selection criteria for rows
- INNER JOIN — Join rows from multiple tables
- ORDER BY — Ordering criteria for rows
- INSERT — Insert data in a specified table
- UPDATE — Updates data in a specified table
- DELETE — Deletes data from a specified table
- CREATE — Creates a new table
- DROP — Delete an existing table
- COUNT — Count the number of records that satisfy a given search criteria

Select Query

SELECT * FROM tablename;

- Select all columns from a given table

SELECT * FROM Authors;

SELECT authorID, lastName FROM Authors;

Where Clause

WHERE clause is used to specify selection criteria.

```
SELECT columnName1, columnName2,…
FROM tableName
WHERE criteria;
```

Example:

```
SELECT title, editionNumber, copyright
FROM Titles
WHERE copyright > 2006;
```

Operators in The WHERE Clause

The following operators can be used in the WHERE clause:

Operator	Description
=	Equal
<>	Not equal. **Note:** In some versions of SQL this operator may be written as !=
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

Exercise 1 (Do alone, Turn in at end of class with name and X.500 id)

- Write a query that returns the information on Titles which are the third edition of a book.

Pattern Matching with Where

Pattern matching can be specified in the WHERE clause using the LIKE operator.

Example:

```
SELECT authorID, firstName, lastName
FROM Authors
WHERE lastName LIKE 'D%';
```

ORDER By Clause

SELECT columnName1, columnName2,…
FROM tableName
ORDER BY column ASC;

Use ASC for ascending order, Use DESC for descending order.

Example:

```
SELECT authorID, firstName, lastName
FROM Authors
ORDER BY lastName ASC;
```

ORDER By Clause

Example:

```
SELECT isbn, title, editionNumber, copyright
FROM Titles
WHERE title LIKE '%How to Program%'
ORDER BY title ASC, copyright DESC;
```

Join Operation: Merging Data from Multiple Tables

```
SELECT columnName1, columnName2,…
FROM tableName
INNER JOIN table2
ON table1.columnName = table2.columnName;
```

Example:

```
SELECT firstName, lastName, isbn
FROM Authors
INNER JOIN AuthorISBN
ON Authors.authorID = AuthorISBN.authorID
ORDER BY lastname, firstName;
```

Join Operation using MySQL

Example:

```
Select firstName, lastName, isbn
From Authors, AuthorISBN
Where Authors.authorID = AuthorISBN.authorID;
```

Insert Operation

INSERT INTO tableName (columnName1, columnName2,…columnNameN)

VALUES (value1, value2, … valueN);

Example:

INSERT INTO Authors (firstName, lastName)

VALUES('Jack', ‘Kennedy’);

Update Operation

```
UPDATE tableName
SET columnName1 = value1,
    columnName2 = value2,
     …
    columnNameN = valueN
WHERE criteria;
```

Example:

```
UPDATE Authors
SET lastName = 'Jones',
WHERE lastName = ‘Kennedy' AND firstName = 'Jack’;
```

Exercise 2 (Do alone, turn in at end of class with name and x.500 id)

- Write a query to find all the Authors whose last names start with Sa

Example: Authors Table (1)

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Kate	Steinbuhler
5	Sean	Santry
6	Ted	Lin
7	Praveen	Sadhu
8	David	McPhie
9	Cheryl	Yaeger
10	Marina	Zlatkina
11	Ben	Wiedermann
12	Jonathan	Liperi

MySQL – demo

- Log into MySQL using your credentials
- Create and populate the books table using the SQL in the file

  products1.sql

Practice doing Queries – select, insert, delete, update

Next Time

- Sessions
- Using the Database