

HARVARD SUMMER SCHOOL

Building Dynamic Websites

Computer Science S-75

Section 5

PDO, JOIN, INDEX, COMMIT, Walkthrough 1

INDEX

- Improves performance
- Uses additional storage

```
CREATE [UNIQUE] INDEX index_name  
ON table_name (column_name, ...)
```

PDO

- PHP Data Objects
- Provides a data-access abstraction layer
 - Regardless of database, commands stay the same
- Does not provide a database abstraction layer
 - If you use features specific to your database, they are not automatically translated
- Some of the drivers available:
 - Informix, MS SQL, MySQL, Oracle, SQLite

PDO

```
$dsn = 'mysql:host=localhost;dbname=thename';
```

```
$dbh = new PDO($dsn, $user, $pwd);
```

PDO::query() – Execute statement and return result set

PDO::exec() – Execute statement and return number of affected rows

PDO::prepare() – Prepares a statement for execution

PDO::bindValue() – Binds a value to a parameter

PDO::execute() – Executes a prepared statement

Why use prepared queries?

- Performance
 - Preparation is performed once
 - Parameters can have new values bound before each execution
- Security
 - Parameters are properly escaped automatically

JOIN

- Creates a new temporary table based upon the combination of two or more tables

```
SELECT column_name, ...  
FROM table_name  
[{LEFT | RIGHT} OUTER | INNER] JOIN table_name  
[ON condition]
```

INNER JOIN

Students

| First | Last | ID |
|-------|---------|----|
| Chris | Gerber | 1 |
| David | Malan | 2 |
| Alain | Ibrahim | 3 |

Grades

| Student | Project | Grade |
|---------|---------|-------|
| 1 | 0 | B |
| 1 | 1 | B+ |
| 2 | 0 | A |
| 2 | 1 | A |
| 4 | 0 | C |

| First | Last | ID | Student | Project | Grade |
|-------|--------|----|---------|---------|-------|
| Chris | Gerber | 1 | 1 | 0 | B |
| Chris | Gerber | 1 | 1 | 1 | B+ |
| David | Malan | 2 | 2 | 0 | A |
| David | Malan | 2 | 2 | 1 | A |

INNER JOIN

```
SELECT *  
FROM students  
JOIN grades  
ON students.id = grades.student
```


LEFT JOIN

Students

| First | Last | ID |
|-------|---------|----|
| Chris | Gerber | 1 |
| David | Malan | 2 |
| Alain | Ibrahim | 3 |

Grades

| Student | Project | Grade |
|---------|---------|-------|
| 1 | 0 | B |
| 1 | 1 | B+ |
| 2 | 0 | A |
| 2 | 1 | A |
| 4 | 0 | C |

| First | Last | ID | Student | Project | Grade |
|-------|---------|----|---------|---------|-------|
| Chris | Gerber | 1 | 1 | 0 | B |
| Chris | Gerber | 1 | 1 | 1 | B+ |
| David | Malan | 2 | 2 | 0 | A |
| David | Malan | 2 | 2 | 1 | A |
| Alain | Ibrahim | 3 | | | |

LEFT JOIN

```
SELECT *  
FROM students  
LEFT JOIN grades  
ON students.id = grades.student
```

RIGHT JOIN

Students

| First | Last | ID |
|-------|---------|----|
| Chris | Gerber | 1 |
| David | Malan | 2 |
| Alain | Ibrahim | 3 |

Grades

| Student | Project | Grade |
|---------|---------|-------|
| 1 | 0 | B |
| 1 | 1 | B+ |
| 2 | 0 | A |
| 2 | 1 | A |
| 4 | 0 | C |

| First | Last | ID | Student | Project | Grade |
|-------|--------|----|---------|---------|-------|
| Chris | Gerber | 1 | 1 | 0 | B |
| Chris | Gerber | 1 | 1 | 1 | B+ |
| David | Malan | 2 | 2 | 0 | A |
| David | Malan | 2 | 2 | 1 | A |
| | | | 4 | 0 | C |

RIGHT JOIN

```
SELECT *  
FROM students  
RIGHT JOIN grades  
ON students.id = grades.student
```

FULL OUTER JOIN*

Students

| First | Last | ID |
|-------|---------|----|
| Chris | Gerber | 1 |
| David | Malan | 2 |
| Alain | Ibrahim | 3 |

Grades

| Student | Project | Grade |
|---------|---------|-------|
| 1 | 0 | B |
| 1 | 1 | B+ |
| 2 | 0 | A |
| 2 | 1 | A |
| 4 | 0 | C |

| First | Last | ID | Student | Project | Grade |
|-------|---------|----|---------|---------|-------|
| Chris | Gerber | 1 | 1 | 0 | B |
| Chris | Gerber | 1 | 1 | 1 | B+ |
| David | Malan | 2 | 2 | 0 | A |
| David | Malan | 2 | 2 | 1 | A |
| Alain | Ibrahim | 3 | | | |
| | | | 4 | 0 | C |

COMMIT

- Specifies that all commands complete, or none do
- Need to specify InnoDB to add support for transactions

START TRANSACTION;

...

[COMMIT | ROLLBACK];

COMMIT

```
$dbh->beginTransaction();  
foreach($dbh->query("SELECT balance FROM  
users WHERE id=1") as $row) {  
    $bal = $row['balance']+100;  
    $dbh->exec("UPDATE users SET balance=$bal  
WHERE id=1");  
}  
$dbh->commit();
```

Walkthrough 1: C\$75 Finance

- Application architecture
- Database design
- Yahoo! stock quotes
- Other considerations

Yahoo! Stock Quotes

- <http://finance.yahoo.com/d/quotes.csv?s=GOOG+YHOO&f=snl1&e=.csv>
- **s**: Symbol
- **n**: Name
- **l1**: Last Trade
- <http://www.gummy-stuff.org/Yahoo-data.htm>

Live Demo