Nested SQL Queries, Joins and Views in PHP

Susanne Coates, Ph.D.

College of Arts and Humanities
Web and Application Services

Using Complex SQL Queries

Aggregate data from multiple tables

Display the Page using PHP

PHP vs. Complex Queries

Why?

PHP-SQL Review:

```
<?php
$servername = ”localhost”;
$username = ”root”;
$password = ”root”;
$dbname = ”example”;

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection
if (!$conn) {
    die(”Connection failed: ” . mysqli_connect_error());
}

$sql = ”SELECT * FROM page”;

$result = mysqli_query($conn, $sql);

while( $row = mysqli_fetch_assoc($result) ) {
      print_r($row);
    }

mysqli_close($conn);

?>
```

Page

Requirements
- Track revisions to body
- Multiple comments
- Track date and author

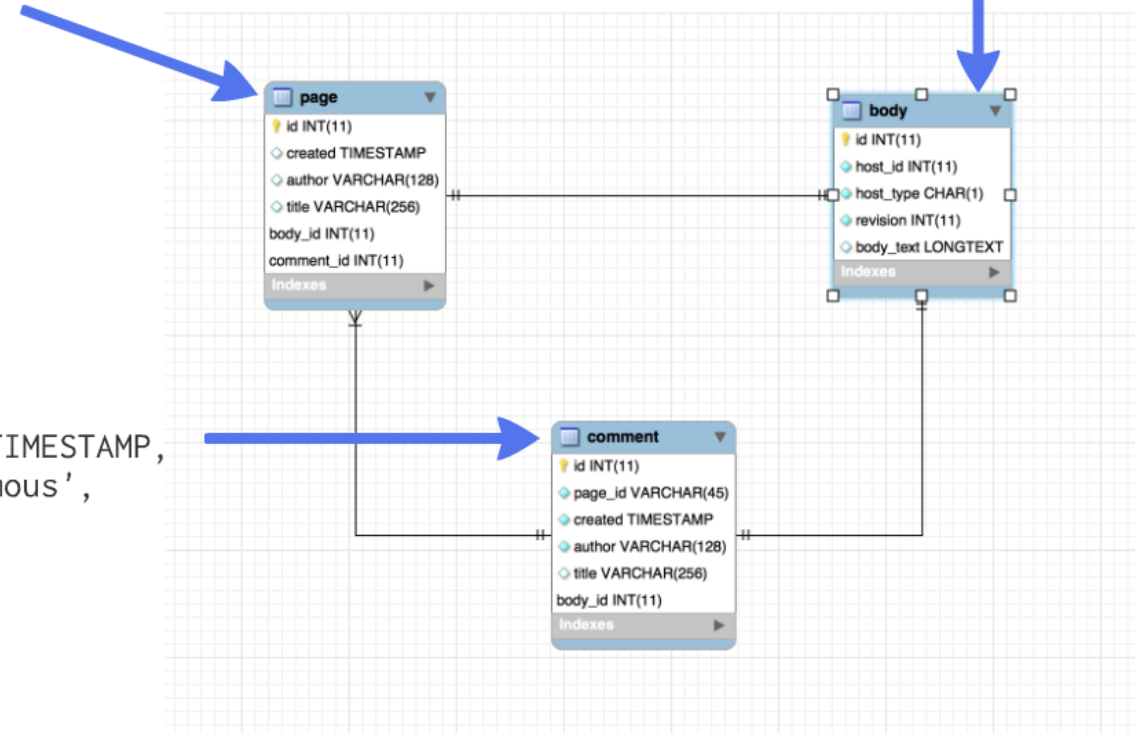
Data:
- Title
- Author
- Date
- Body Text
- Comments
  - 1
    - Title
    - Author
    - Date
    - Body Text
  - N ...

Database

```
CREATE TABLE `example`.`body` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `host_id` INT NOT NULL,
  `host_type` CHAR(1) NOT NULL,
  `revision` INT NOT NULL,
  `text` LONGTEXT NULL,
  PRIMARY KEY (`id`));
```

```
CREATE TABLE `example`.`page` (
 `id` INT(11) NOT NULL AUTO_INCREMENT,
 `created` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
 `author` VARCHAR(128) NULL DEFAULT 'anonymous',
 `title` VARCHAR(256) NULL DEFAULT NULL,
 PRIMARY KEY (`id`));
```

```
CREATE TABLE `example`.`comment` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `page_id` VARCHAR(45) NOT NULL,
  `created` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `author` VARCHAR(128) NOT NULL DEFAULT 'anonymous',
  `title` VARCHAR(256) NULL,
  PRIMARY KEY (`id`));
```



Selecting Data

Page 1

```
Select * FROM Page WHERE id = 1;
```

```
Select created, author, title FROM page WHERE id = 1;
```

Body for Page 1

```
Select * FROM body WHERE host_id = 1;
```

```
Select body_text FROM body WHERE host_id = 1;
```

AS Assigns an alias

```
Select created, author, title FROM page WHERE id = 1;

With Aliases:

SELECT created AS date,
       author AS user,
       title
  FROM page AS pg
  WHERE id = 1;
```

Selecting Data

Page 1

```
SELECT created, author, title FROM page AS pg WHERE id = 1;
```

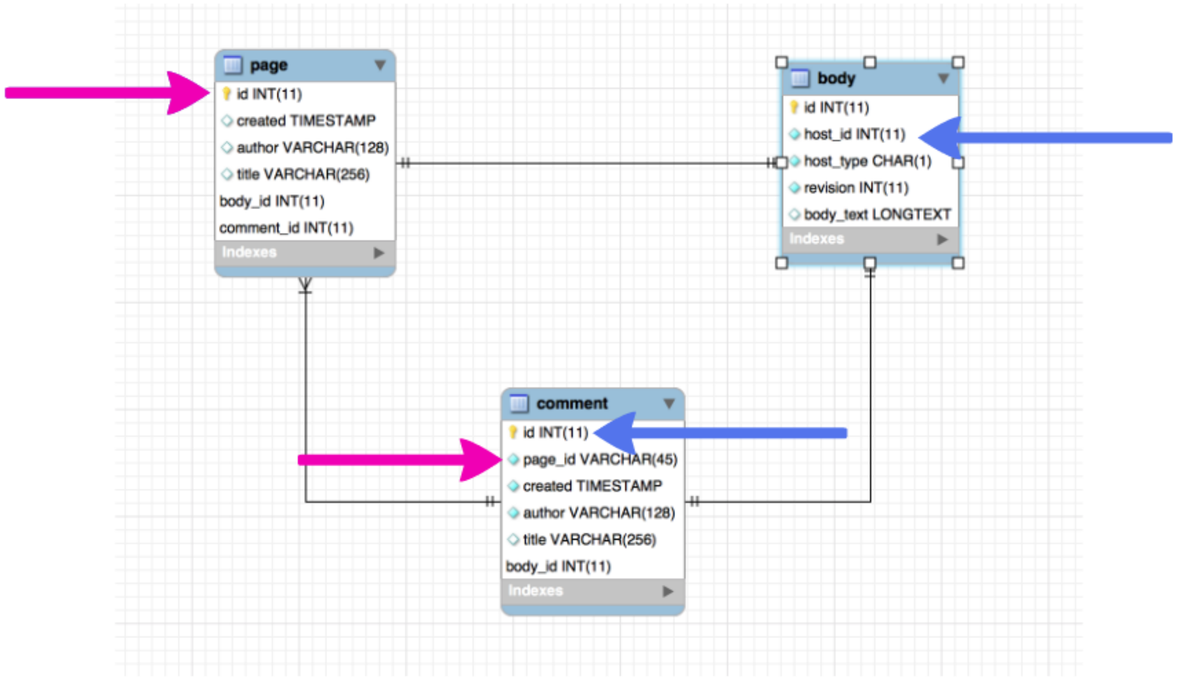
Body for Page 1

```
SELECT body_text FROM body AS bd WHERE host_id = 1;
```

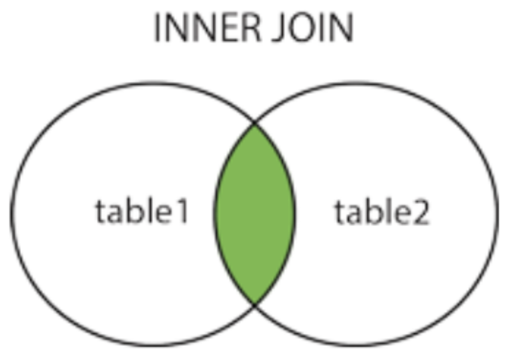
Can we combine the page and body queries into a single query?

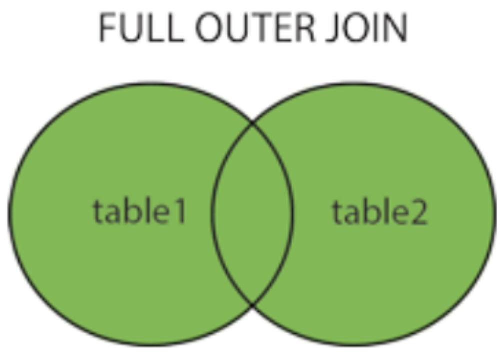
JOIN ... ON ...

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.



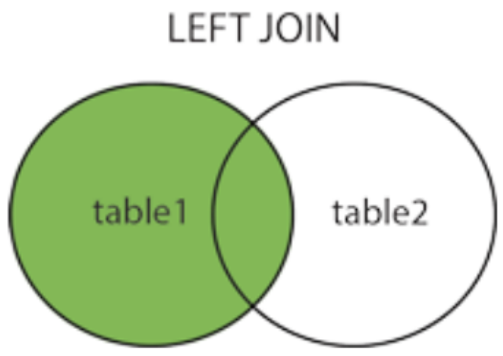
4 Basic Types of Joins

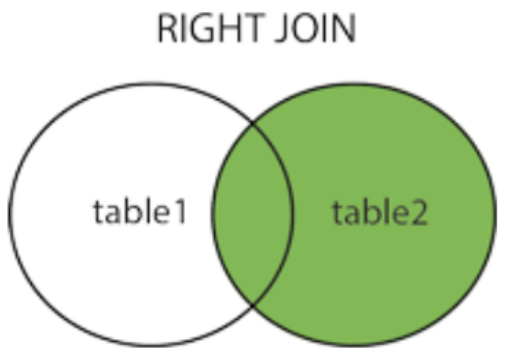




- INNER JOIN: Returns records that have matching values in both tables
- OUTER JOIN: Return all records when there is a match in either left or right table

- LEFT JOIN: Return all records from the left table, and the matched records from the right table
- RIGHT JOIN: Return all records from the right table, and the matched records from the left table





Source: https:// www.w3schools.com/sql/sql_join.asp

Display a Page with its Text

```
SELECT pg.created, pg.author, bd.host_type as type, pg.title, bd.body_text
FROM page AS pg
   JOIN body AS bd ON pg.id = bd.host_id
      WHERE pg.id = 1 AND bd.host_type = ”P” ;
```

Q: why does this display more than one page?

Find Latest Revision

```
SELECT MAX(revision)
FROM body
WHERE host_id = 1 AND host_type = ”P”
```

Q: Using this query, how would we find the next revision if we were adding a page or comment?

Q: What is a problem with using this approach?

Select Latest Revision of Page

```
SELECT pg.created, pg.author, bd.host_type as type, pg.title, bd.body_text

FROM page AS pg

   JOIN body AS bd ON pg.id = bd.host_id

      WHERE pg.id = 1 AND bd.host_type = "P" AND bd.revision = (

         SELECT max(revision)
         FROM body
         WHERE host_id = 1 AND host_type = "P"

      );
```

Select Comments for a Page

```
SELECT co.created, co.author, bd.host_type AS type, co.title, bd.body_text

FROM comment AS co

   JOIN body AS bd ON co.id = bd.host_id

      WHERE co.page_id = ”1” AND bd.host_type = ”C” AND bd.revision = (

        SELECT max(revision)
        FROM body
        WHERE host_id = co.id AND host_type=”C”

      );
```

Combine two Result Sets

```
SELECT pg.created, pg.author, bd.host_type as type, pg.title, bd.body_text

FROM page AS pg

   JOIN body AS bd ON pg.id = bd.host_id

      WHERE pg.id = 1 AND bd.host_type = "P" AND bd.revision = (

         SELECT max(revision)
         FROM body
         WHERE host_id = 1 AND host_type = "P"

      );
```

+

```
SELECT co.created, co.author, bd.host_type AS type, co.title, bd.body_text

FROM comment AS co

   JOIN body AS bd ON co.id = bd.host_id

      WHERE co.page_id = "1" AND bd.host_type = "C" AND bd.revision = (

        SELECT max(revision)
        FROM body
        WHERE host_id = co.id AND host_type="C"

      );
```

?

UNION

Combines the result-set of two or more SELECT statements on their columns.

Criteria:

- Select Statements must have the same number of columns
- Columns must also have similar data types
- Columns in each SELECT statement must also be in the same order

UNION selects distinct values, to allow duplicates use UNION ALL

UNION

```
SELECT column1, column2, column3 FROM table1

UNION

SELECT column1, column2, column3 FROM table2

UNION

SELECT column1, column2, column3 FROM table3

...
```

```
SELECT pg.created, pg.author, bd.host_type as type, pg.title, bd.body_text
    FROM page AS pg
    JOIN body AS bd ON pg.id = bd.host_id
    WHERE pg.id = 1 AND bd.host_type = "P" AND bd.revision = (
      SELECT max(revision)
      FROM body
      WHERE host_id = 1 AND host_type = "P"
    )

UNION

SELECT co.created, co.author, bd.host_type AS type, co.title, bd.body_text
    FROM comment AS co
    JOIN body AS bd ON co.id = bd.host_id
    WHERE co.page_id = 1 AND bd.host_type = "C" AND bd.revision = (
      select max(revision)
      FROM body
      WHERE host_id = co.id AND host_type="C"
    );
```

Large Result Set

Memory Limits

Inefficiency of Duplication

Select Within Results

VIEW

- Virtual Table
- Dynamic
- Combine SQL query results and present them as a single table

Syntax:

```
CREATE VIEW <View Name> AS SELECT * FROM (

  <YOUR COMPLEX QUERY GOES HERE>

) AS <Derived Table Alias>;
```

Create a view containing all of the revisions to the page and it's comments

```
CREATE VIEW page1_view AS SELECT * FROM (

  SELECT pg.created, pg.author, bd.revision, bd.host_type as type, pg.title, bd.body_text
      FROM page AS pg
      JOIN body AS bd ON pg.id = bd.host_id
      WHERE pg.id = 1 AND bd.host_type = "P"

  UNION

  SELECT co.created, co.author, bd.revision, bd.host_type AS type, co.title, bd.body_text
      FROM comment AS co
      JOIN body AS bd ON co.id = bd.host_id
      WHERE co.page_id = 1 AND bd.host_type = "C"

) AS my_derived_table;
```

Select data from the view

```
SELECT * FROM page1_view WHERE revision = 1 AND type = "P";
```

And, when you're done:

```
DROP VIEW page1_view;
```

Câu hỏi ? Mga tanong?

Imibuzo?

Questions?

Pytania? ¿Preguntas?

Vrae?