LECTURE05: AN INTRODUCTION TO MYSQL

CS418/518: WEB PROGRAMMING

BY DR. JIAN WU

COURTESY: DR. JESSIE LI (PSU) & DR. LUKE ZHANG (PSU)

RELATIONAL DATABASES

- Data Organized in tables
  - Can be created, retrieved, deleted, and updated
  - Uses keys for manipulation
- Key: an identifier (generally not necessarily unique) of data entries
- Primary key – unique identifier for each instance (each row) in the table
- Foreign key – a key that points to the primary key of another table

STUDENT — foreign key table

StudentID	Name	DepartmentID
1	Jian Wu	12

Foreign key

DEPARTMENT — parent table

DepartmentID	DepartmentName	Head
12	Computer Science	Dr. Ravi Mukkamalla

DATABASE NORMALIZATION - FIRST NORM FORM

- All entries contain a single value
- Primary key to uniquely identify each row

GAMING

name	real_name	power 1	power 2	power 3	fair_address	city	state	zip
Clean Freak	John Smith	Strength	X-ray vision	Flight Invisibility	123 Poplar Ave	Townsburg	OH	45293
Soap Stud	Efram Jones	Speed			123 Poplar Ave	Townsburg	OH	45293
The Dustmite	Dustin Huff	Strength	Dirtiness	Laser Vision	452 Elm St. #3D	Burgtown	OH	45201

Missing values!

same data

No primary key

FIRST NORMAL FORM

- Add primary key to tables (could be a composite key but prefer a surrogate key)
- Each attribute is atomic (each value is treated as a single value, not separable)

GAMING

id	lair_id	name	real_name	power	lair_address	city	state	zip
1	1	Clean Freak	John Smith	Strength	123 Poplar Ave	Townsburg	OH	45293
2	1	Clean Freak	John Smith	X-ray vision	123 Poplar Ave	Townsburg	OH	45293
3	1	Clean Freak	John Smith	Flight	123 Poplar Ave	Townsburg	OH	45293
4	1	Clean Freak	John Smith	Invisibility	123 Poplar Ave	Townsburg	OH	45293
5	2	Soap Stud	Efram Jones	Speed	123 Poplar Ave	Townsburg	OH	45293
6	3	The Dustmite	Dustin Huff	Strength	452 Elm St. #3D	Burgtown	OH	45201
7	3	The Dustmite	Dustin Huff	Dirtiness	452 Elm St. #3D	Burgtown	OH	45201
8								
9								

primary key

Not a primary key

However, this table is not conformed to the second norm (see the next slide).

SECOND NORM FORM

- Separate tables for duplicate data

CHARACTER

Id	lair_id	Name	real_name
1	1	Clean Freak	John Smith
2	1	Soap Stud	Efram Jones
3	2	The Dustmite	Dustin Huff

POWER

Id	power
1	Strength
2	X-Ray vision
3	Flight
4	Invisiblity
5	Speed
6	Dirtiness
7	Laser Vision

ADDRESS

Id	lair_address	City	State	zip
1	123 Poplar Ave	Townburg	OH	45293
2	452 Elm St. #3D	Burgtown	OH	45201

CHARACTER_POWER

Char_Id	Power_id
1	1
1	2
1	3
1	4
2	5
3	1
3	6
3	7

THIRD NORM FORM

- Create separate tables for any transitive or partial dependencies

CHARACTER

Id	lair_id	Name	real_name
1	1	Clean Freak	John Smith
2	1	Soap Stud	Efram Jones
3	2	The Dustmite	Dustin Huff

ADDRESS

Id	lair_address	zip
1	123 Poplar Ave	45293
2	452 Elm St. #3D	45201

POWER

Id	power
1	Strength
2	X-Ray vision
3	Flight
4	Invisiblity
5	Speed
6	Dirtiness
7	Laser Vision

ZIPCODE

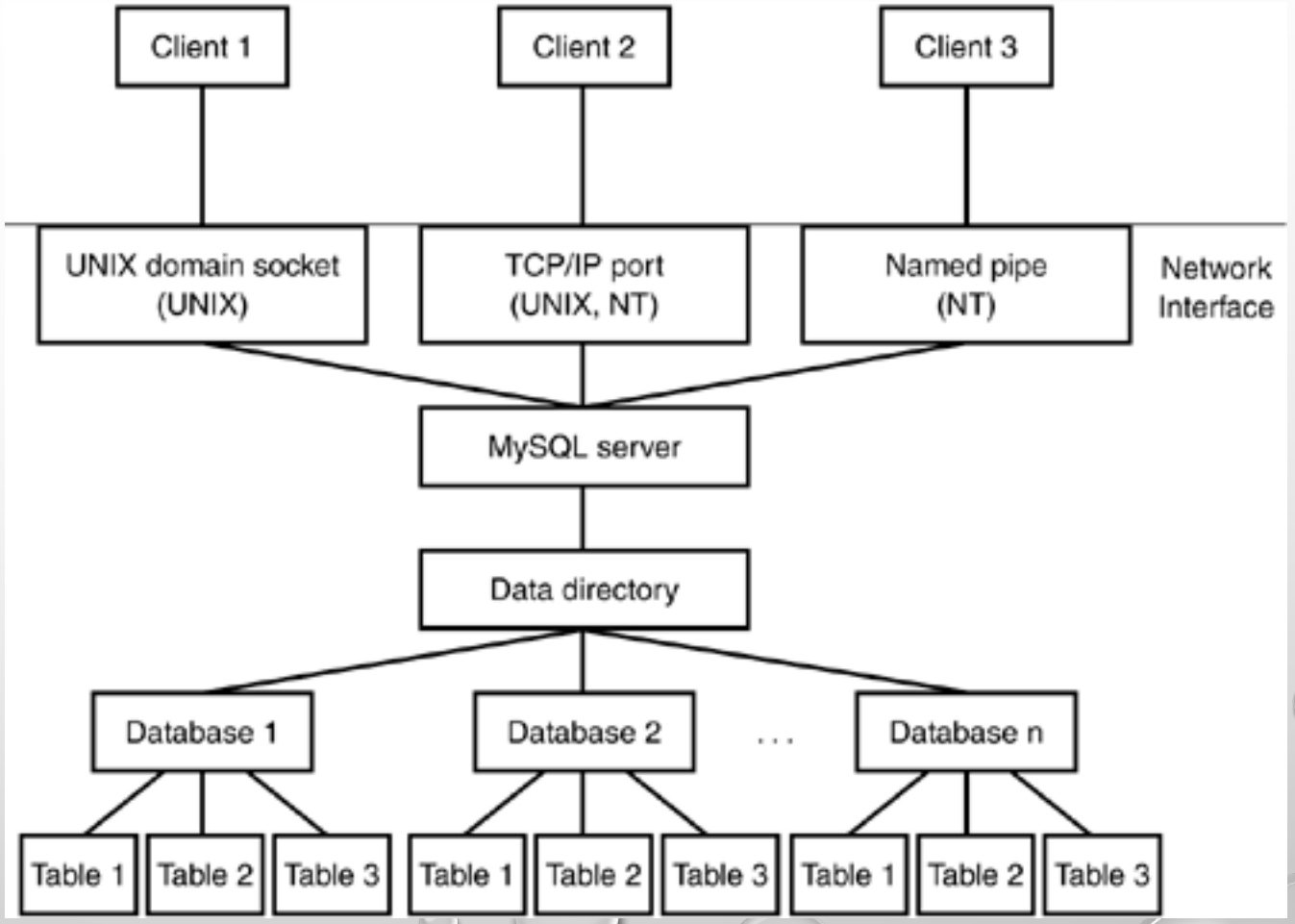
id	City	State
45293	Townburg	OH
45201	Burgtown	OH

CHARACTER_POWER

Char_Id	Power_id
1	1
1	2
1	3
1	4
2	5
3	1
3	6
3	7

MYSQL ARCHITECTURE

- A server contains databases
- A database contains tables
- A table contains rows
- A row contains values



USEFUL COMMANDS

- Login:
  - `$ mysql -h [server] -u [user] -p`
- Use a database:
  - `mysql> use [database];`
- Show tables inside a particular database:
  - `mysql> show tables;`
- Describe the schema of a table:
  - `mysql> desc [table];`
- Select values satisfying certain criteria from a table:
  - `mysql> SELECT * FROM [table] WHERE … ;`
- Display the total number of rows in a table:
  - `mysql> SELECT COUNT(*) FROM [table];`

TABLE COMMANDS (NOT CASE SENSITIVE)

- `CREATE DATABASE [database]` – create a new database
- `CREATE TABLE [table]` – create a new table
- `ALTER TABLE [table]` – modify existing tables such as adding a new column
- `DELETE FROM [table]` – erase data from tables
- `DESCRIBE` – show structure of tables
- `INSERT INTO [table] VALUES` – populate table
- `UPDATE [table]` – modify existing data in tables
- `DROP [table/database]` – destroy a table or database (value+structure) be very careful!

CREATE A TABLE

- DEPARTMENT(DepartmentName, BudgetCode, OfficeNumber, Phone)

```
CREATE TABLE `department` (
  `DepartmentName` char(35) NOT NULL,
  `BudgetCode` char(30) NOT NULL,
  `OfficeNumber` char(15) NOT NULL,
  `Phone` char(12) NOT NULL,
  PRIMARY KEY (`DepartmentName`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Full command

```
CREATE TABLE department (
  DepartmentName varchar(35) NOT NULL PRIMARY KEY,
  BudgetCode varchar(30) NOT NULL,
  OfficeNumber varchar(15) NOT NULL,
  Phone varchar(12) NOT NULL
) DEFAULT CHARSET=utf8mb4;
```

Equivalent command

CREATE A TABLE WITH A FOREIGN KEY

DEPARTMENT(DepartmentName, BudgetCode, OfficeNumber, Phone)

EMPLOYEE(EmployeeNumber, FirstName, LastName, *Department*, Phone Email)

```
CREATE TABLE `EMPLOYEE` (
  `EmployeeNumber` int NOT NULL AUTO_INCREMENT,
  `FirstName` char(25) NOT NULL,
  `LastName` char(25) NOT NULL,
  `Department` char(35) NOT NULL DEFAULT 'Human Resources',
  `Phone` char(12) DEFAULT NULL,
  `Email` varchar(100) NOT NULL,
  PRIMARY KEY (`EmployeeNumber`),
  UNIQUE KEY `Email` (`Email`),
  KEY `EMP_DEPART_FK` (`Department`),
  CONSTRAINT `EMP_DEPART_FK` FOREIGN KEY (`Department`)
REFERENCES `DEPARTMENT` (`DepartmentName`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
```

Full command

```
CREATE TABLE EMPLOYEE (
  EmployeeNumber int NOT NULL PRIMARY KEY AUTO_INCREMENT,
  FirstName varchar(25) NOT NULL,
  LastName varchar(25) NOT NULL,
  Department varchar(35) NOT NULL DEFAULT 'Human Resources',
  Phone varchar(12) DEFAULT NULL,
  Email varchar(100) NOT NULL UNIQUE,
  CONSTRAINT `EMP_DEPART_FK` FOREIGN KEY (`Department`)
REFERENCES DEPARTMENT (DepartmentName) ON UPDATE CASCADE
) DEFAULT CHARSET=utf8mb4;
```

Equivalent command

ALTER THE TABLE SCHEMA

- Drop a column

```
ALTER TABLE EMPLOYEE DROP COLUMN Phone;
```

- Add a column

```
ALTER TABLE EMPLOYEE ADD COLUMN Address varchar (128);
```

- Change the data type of a column

```
alter table employee modify address varchar(256);
```

COUNT, SORT, AND GROUPING

- Count the number of records in a table satisfying a set of conditions

```
SELECT COUNT(*) FROM EMPLOYEE WHERE Department = 'Computer Science' AND
FirstName = 'Steve';
```

- Sort the employee names by their last names in alphabetical order

```
SELECT FirstName, LastName FROM EMPLOYEE ORDER BY LastName;
```

- Show the number of employees in each department and sort them in descendent order

```
SELECT COUNT(*) AS NUMBER, Department FROM EMPLOYEE GROUP BY Department
ORDER BY NUMBER DESC;
```

QUERY SYNTAX

- `SELECT [fieldnames] FROM [table] WHERE [criteria] ORDER BY [fieldname] LIMIT [offset, maxrows];`

QUERY EXAMPLE

CHARACTER

Table

Id	lair_id	Name	real_name
1	1	Clean Freak	John Smith
2	1	Soap Stud	Efram Jones
3	2	The Dustmite	Dustin Huff

```
SELECT name, real_name FROM CHARACTER;
```

result

Name	real_name
Clean Freak	John Smith
Soap Stud	Efram Jones
The Dustmite	Dustin Huff

INNER JOIN

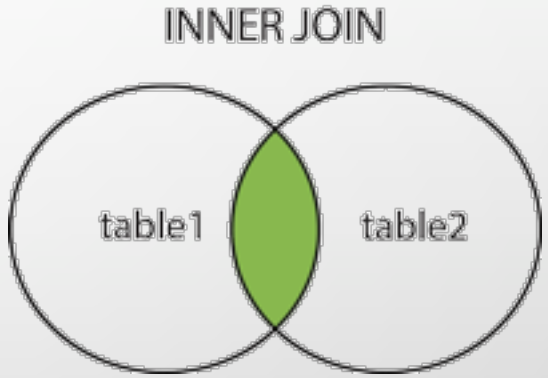
- `SELECT * FROM Table1 INNER JOIN Table2 ON Table1.name = Table2.name;`

Id	Name
1	Pirate
2	Monkey
3	Ninja
4	Spaghetti

Table1

Id	Name
1	Rutabaga
2	Pirate
3	Darth Vader
4	Ninja

Table2



OUTER JOIN

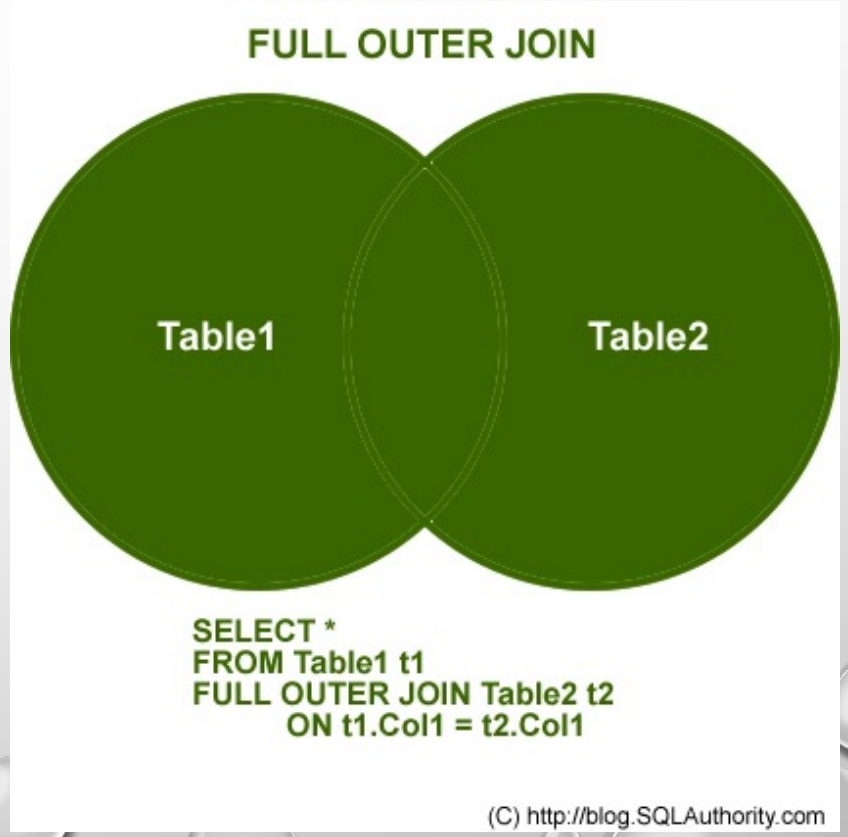
```
SELECT * FROM Table1 OUTER JOIN Table2 ON Table1.name =
Table2.name;
```

Id	Name
1	Pirate
2	Monkey
3	Ninja
4	Spaghetti

Table1

Id	Name
1	Rutabaga
2	Pirate
3	Darth Vader
4	Ninja

Table2



INSERT

- `INSERT INTO lecture4.CHARACTER (id,lair_id,name,real_name) VALUES (NULL, 4, "General Grime", 'Philip Grimaldi');`

CHARACTER

Id	lair_id	Name	real_name
1	1	Clean Freak	John Smith
2	1	Soap Stud	Efram Jones
3	2	The Dustmite	Dustin Huff
4	4	General Grime	Philip Grimaldi

The id is automatically added by the system

DELETES

- Delete selected rows:
  - `DELETE FROM lecture4.CHARACTER WHERE id=1;`
- Delete everything (be very cautious to do this!)
  - `DELETE FROM lecture4.CHARACTER;`

UPDATE DATA

- `UPDATE lecture4.CHARACTER SET name='Admiral Grime', real_name='Phillip J. Grimaldi' WHERE id=4;`

Be very careful about this condition!
Make sure it is set before you hit "Enter".

PHP AND MYSQL

- Connect to a MySQL Server
  - `$mysqli = new mysqli("hostname", "user", "password", "database")`
- Send MySQL commands/query to server
  - `$results = $mysqli->query("query")`
- Show error message generated by the MySQL server
  - `$mysqli->error`
- Release results array
  - `$results->free()`
  - Note: result object is only created for SELECT, SHOW, DESCRIBE or EXPLAIN queries.
  - For CREATE, INSERT, UPDATE there is nothing to "free".
- Close connection
  - `$mysqli->close()`

NOTES

- General notes
  - Do not use the same name for different columns.
    - If two or more columns of the result have the same column names, the last column will take precedence and overwrite the earlier data.
  - Can use alias in SELECT statement (AS keyword)
    - `SELECT Email AS EmployeeEmail FROM Employee;`
- `$results->fetch_array()`
  - Store data in both the numeric indices of the result array AND associate indices using the column name as the key (see the example on the next page)

FETCH_ARRAY()

```
$query = "SELECT * FROM CHARACTER";
$results = $mysqli->query($query)
      or die($mysql->error.__LINE__);
$row = $results->fetch_array();
print_r($row);
```

Id	Lair_id	Name	Real_name
1	1	Clean Freak	John Smith
2	1	Soap Stud	Efram Jones
3	2	The dustmite	Dustin Huff

```
Array([0]=>Clean Freak[name]=>Clean Freak[1]=>John Smith[real_name]=>John Smith)
Array([0]=>Soap Stud[name]=>Soap Stud[1]=>Efram Jones[real_name]=>Efram Jones)
Array([0]=>The dustmite[name]=>the dustmine[1]=>Dustin Huff[real_name]=>Dustin Huff)
```

see dbdataretrieve.php

IMPORTANT: ALWAYS BACKUP YOUR DATA!

- Create a dump of all tables in a selective database (mydb)
  - https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html
  - `mysqldump --databases mydb > mydb.sql`
- Import your database
  - https://www.cyberciti.biz/faq/import-mysql-dumpfile-sql-datafile-into-my-database/
  - `mysql -u root --force --verbose < mydb.sql`

SQL EXERCISES

- Show the firstname and lastname of all employees.
  - `SELECT firstname, lastname FROM EMPLOYEE;`
- Show all distinct firstname of employees.
  - `SELECT DISTINCT(firstname) FROM EMPLOYEE;`
- Show all the employees with firstname is ‘Mary’ or the department is accounting;
  - `SELECT * FROM EMPLOYEE WHERE Firstname = ‘Mary’ OR Department = ‘Accounting’;`
- Show all the employee with the department is Finance or Accounting or Marketing;
  - `SELECT * FROM EMPLOYEE WHERE Department IN ('Finance’, 'Accounting’, 'Marketing');`

MYSQL EXERCISES

- Find the employees that have firstname ending with “y” and has 4 letters in total.
  - `SELECT * FROM EMPLOYEE WHERE Firstname LIKE '___y';`
- Find employees that have first name starts with letter “R” and ends with letter “d”.
  - `SELECT * FROM EMPLOYEE WHERE Firstname LIKE 'R%d';`
- Find employees that have first name with the 3rd letter is “a” and ends with letter “r”.
  - `SELECT * FROM EMPLOYEE WHERE Firstname LIKE '__a%r';`

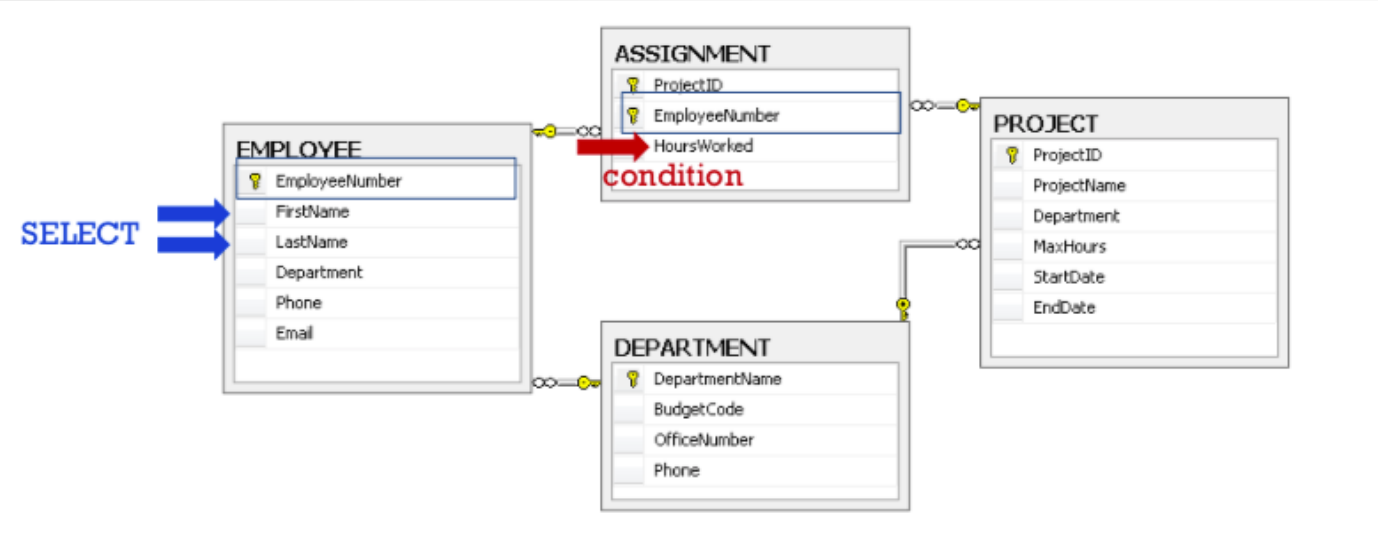
MYSQL EXERCISES

- Sort employees by LastName in descending order, and then by FirstName in ascending order
  - `SELECT * FROM employee ORDER BY lastname DESC, firstname ASC;`
- Count how many projects with MaxHours less than 130;
  - `SELECT COUNT(*) FROM PROJECT WHERE MaxHours < 130;`
- Count how many employees in each department?
  - `SELECT Department,COUNT(*) AS NumOfEmployees FROM EMPLOYEE GROUP BY Department;`

MYSQL EXERCISES IN MULTIPLE TABLES

- Show the names of employees who worked less than 20 hours

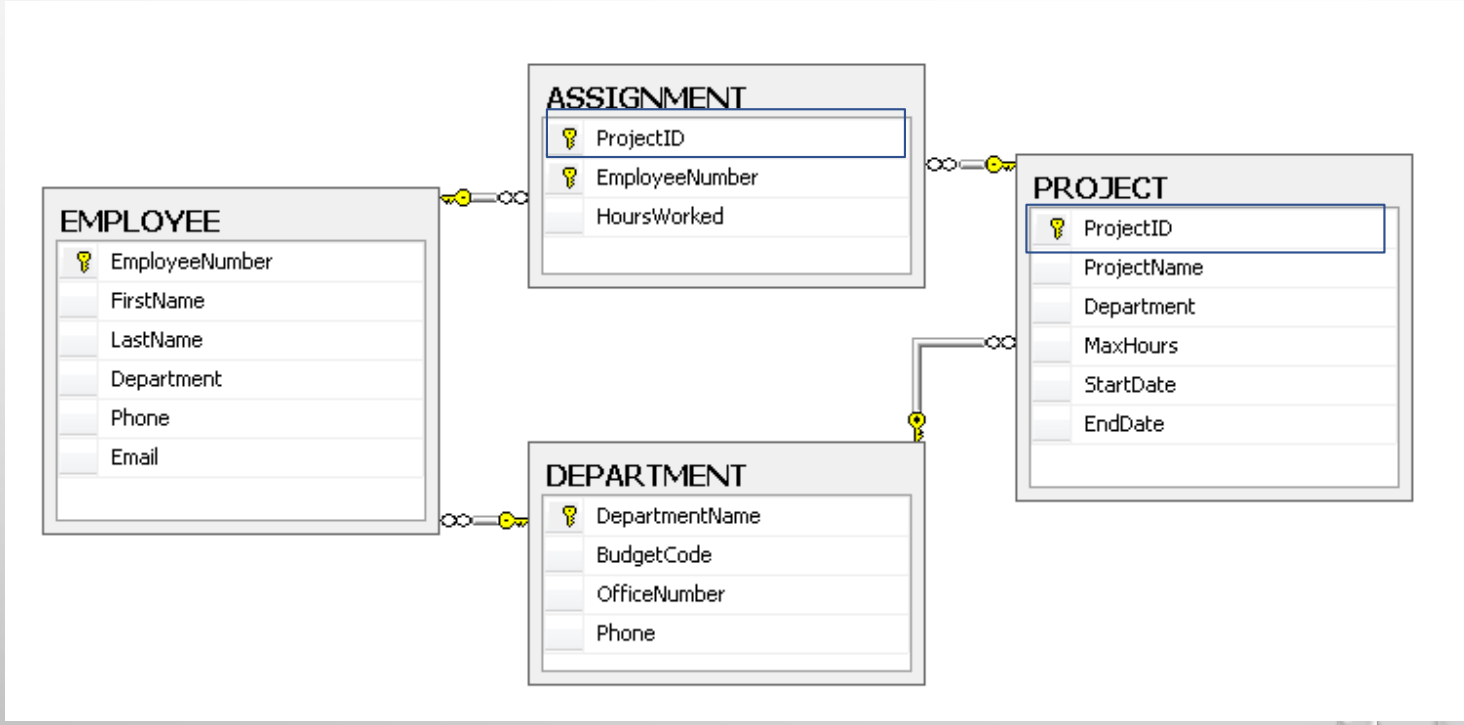
```
SELECT FirstName, LastName,HoursWorked FROM EMPLOYEE AS E, ASSIGNMENT AS
A WHERE E.EmployeeNumber = A.EmployeeNumber AND HoursWorked<20;
```




```
+-----------+-----------+-------------+
| FirstName | LastName  | HoursWorked |
+-----------+-----------+-------------+
| Tom       | Caruthers |       15.00 |
| Heather   | Jones     |       10.00 |
+-----------+-----------+-------------+
2 rows in set (0.00 sec)
```

MYSQL EXERCISE

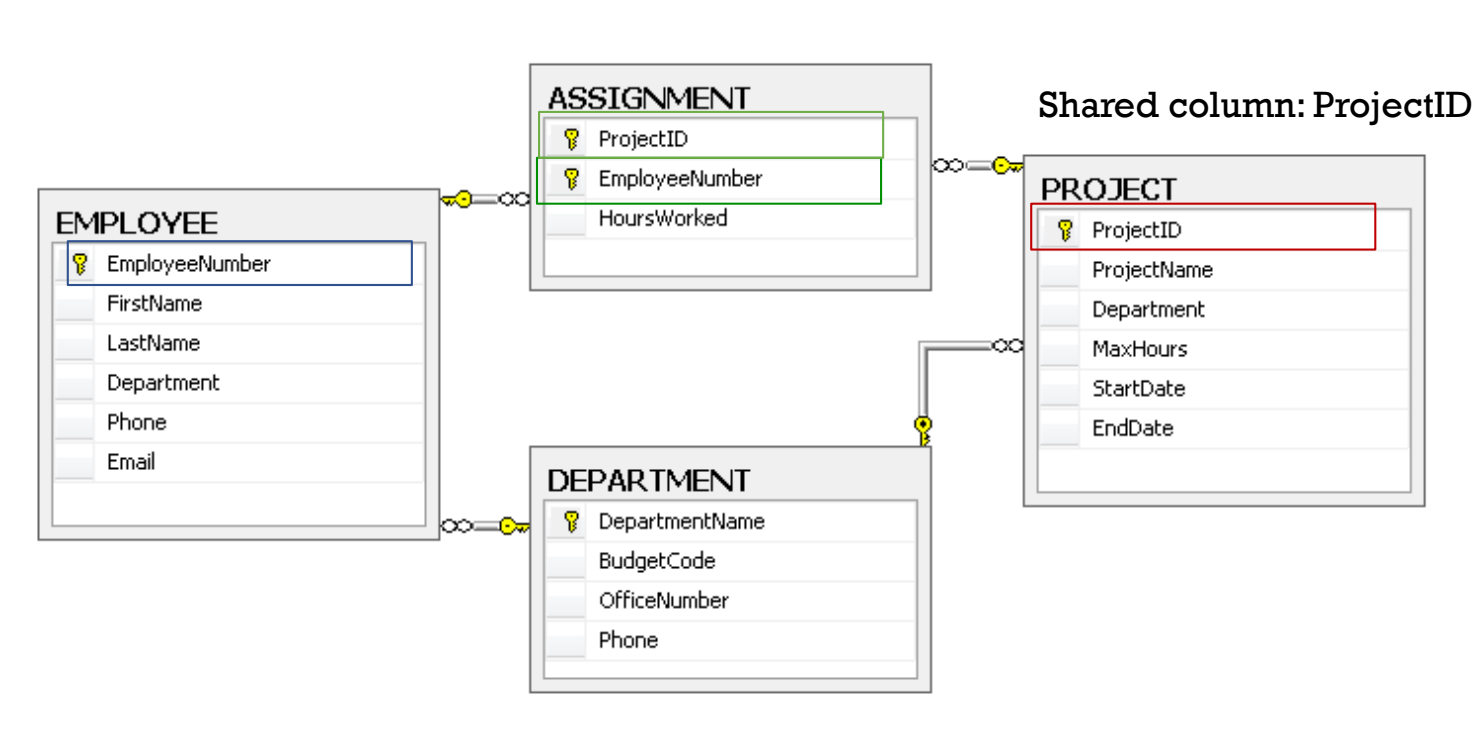
- Show the project names assigned to EmployeeNumber 15;
  - `SELECT ProjectName FROM PROJECT P, ASSIGNMENT A WHERE A.ProjectID=P.ProjectID AND A.EmployeeNumber=15;`




```
+----------------------------+
| ProjectName                |
+----------------------------+
| 2010 Q3 Product Plan       |
| 2010 Q3 Portfolio Analysis |
| 2010 Q4 Product Plan       |
+----------------------------+
```

MYSQL EXERCISE 3 TABLES

- The names of employees who are assigned with projects associated with Finance department
  - `SELECT FirstName, LastName FROM EMPLOYEE AS E, PROJECT AS P, ASSIGNMENT AS A WHERE E.EmployeeNumber = A.EmployeeNumber AND P.ProjectID = A.ProjectID AND P.Department = 'Finance';`



FirstName	LastName
Mary	Jacobs
Rosalie	Jackson
Tom	Caruthers
Heather	Jones

BACKUP SLIDES BEYOND THIS POINT

CS418/518

MYSQL EXAMPLES

- authentication.php

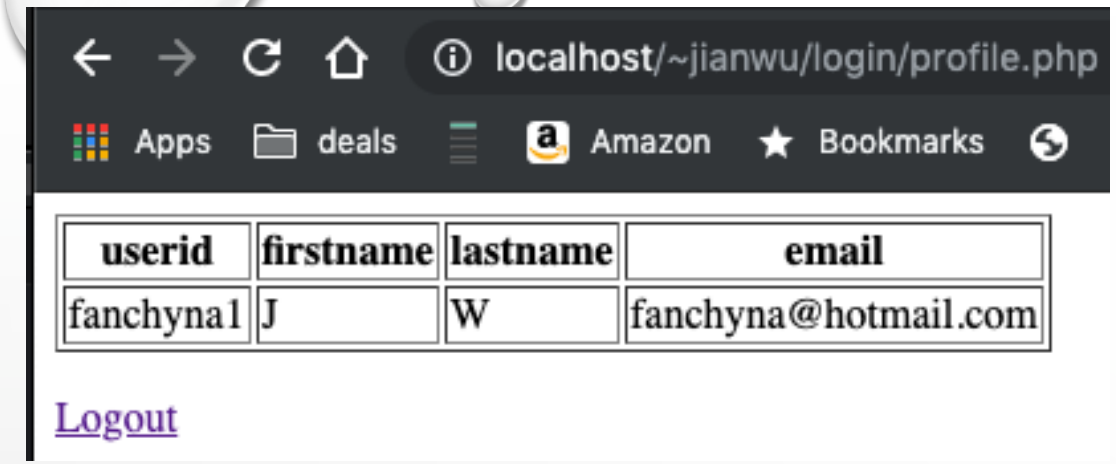
```php
$server = "localhost";
$sqlUsername = "root";
$sqlPassword = "september2019";
$databaseName = "php";

$conn = new mysqli($server, $sqlUsername, $sqlPassword,
    $databaseName);
```

authentication.php

```php
function authenticateUser($connection, $username, $password)
{
  // User table which stores userid and password
  // Change to your own table name
  $userTable = "userprofile";

  // Test the username and password parameters
  if (!isset($username) || !isset($password))
    return false;

  $pa = md5($password);
  // Formulate the SQL statment to find the user
  $sql = "SELECT *
     FROM $userTable
     WHERE userid = '$username' AND password = '$pa'";
  echo $query;

  // Execute the query
  $query_result = $connection->query($sql);
  if (!$query_result) {
          echo "Sorry, query is wrong";
          echo $query;
  }

  // exactly one row? then we have found the user
  $nrows = $query_result->num_rows;
  if ( $nrows != 1)
    return false;
  else
    return true;
}
```

login.php

```php
require 'authentication.php';

//never forget to start the session
session_start();
$errorMessage = '';

//are user ID and Password provided?
if (isset($_POST['txtUserId']) && isset($_POST['txtPassword'])) {

    //get userID and Password
    $loginUserId = $_POST['txtUserId'];
    $loginPassword = $_POST['txtPassword'];

    //connect to the database
    $connection = new mysqli($server, $sqlUsername, $sqlPassword,
        $databaseName);

    // Authenticate the user
    if (authenticateUser($connection, $loginUserId, $loginPassword))
    {
        //the user id and password match,
        // set the session
        $_SESSION['db_is_logged_in'] = true;
        $_SESSION['userID'] = $loginUserId;

        // after login we move to the main page
        header('Location: main.php');
        exit;
    } else {
        $errorMessage = 'Sorry, wrong username / password';
    }
```



profile.php

```php
$conn = new mysqli($server, $sqlUsername, $sqlPassword, $databaseName);

// Prepare query
$table = "userprofile";
$uid = $_SESSION['userID'];
$sql = "SELECT userid, firstname, lastname, email FROM $table where
    userid = '$uid'";

// Execute query
$query_result = $conn->query($sql);
if (!$query_result) {
    echo "Query is wrong: $sql";
        die;
}

// Output query results: HTML table
echo "<table border=1>";
echo "<tr>";

// fetch attribute names
while ($fieldMetadata = $query_result->fetch_field()) {
    echo "<th>".$fieldMetadata->name."</th>";
}
echo "</tr>";

// fetch table records
while ($line = $query_result->fetch_assoc()) {
    echo "<tr>\n";
        foreach ($line as $cell) {
            echo "<td> $cell </td>";
        }
    echo "</tr>\n";
}
echo "</table>";
```

BACKUP SLIDES BEYOND THIS POINT