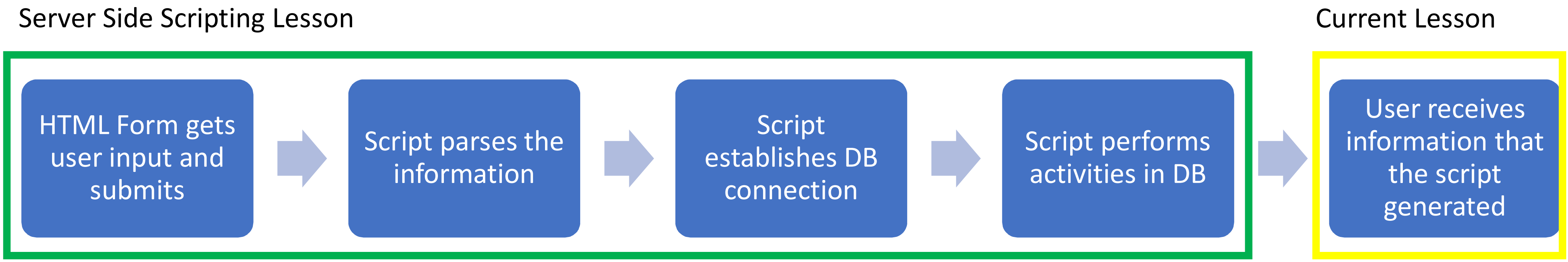
Connecting to a DB

Server Side Connection to a DB



Server Side Scripting Lesson

HTML Form gets user input and submits

Script parses the information

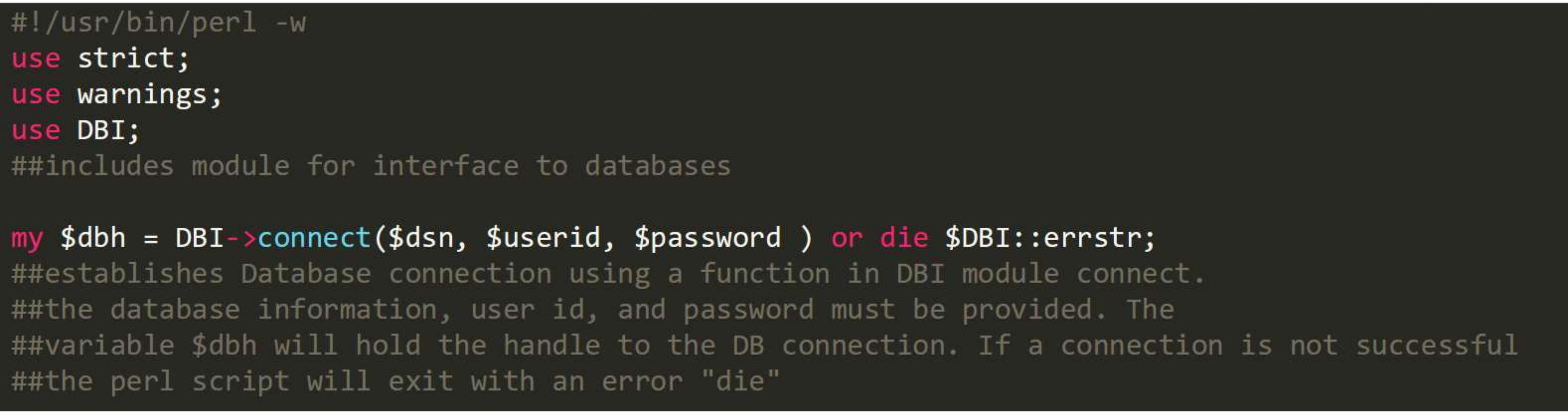
Script establishes DB connection

Script performs activities in DB

Current Lesson

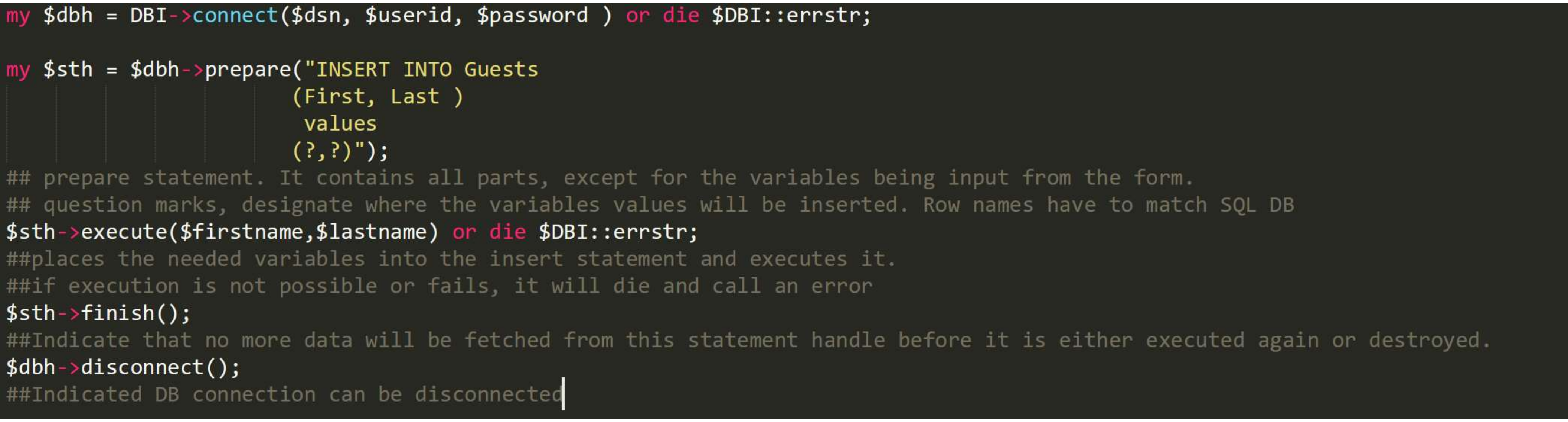
User receives information that the script generated

Reminder: How a DB Connection is Established



```perl
#!/usr/bin/perl -w
use strict;
use warnings;
use DBI;
##includes module for interface to databases

my $dbh = DBI->connect($dsn, $userid, $password ) or die $DBI::errstr;
##establishes Database connection using a function in DBI module connect.
##the database information, user id, and password must be provided. The
##variable $dbh will hold the handle to the DB connection. If a connection is not successful
##the perl script will exit with an error "die"
```

Reminder: Perl Statement Generation



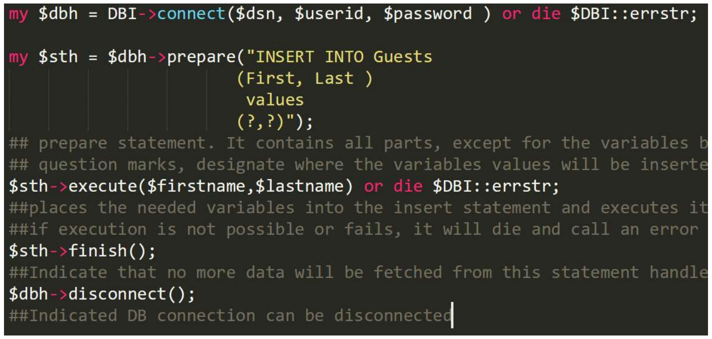
```
my $dbh = DBI->connect($dsn, $userid, $password ) or die $DBI::errstr;

my $sth = $dbh->prepare("INSERT INTO Guests
                       (First, Last )
                        values
                       (?,?)");
## prepare statement. It contains all parts, except for the variables being input from the form.
## question marks, designate where the variables values will be inserted. Row names have to match SQL DB
$sth->execute($firstname,$lastname) or die $DBI::errstr;
##places the needed variables into the insert statement and executes it.
##if execution is not possible or fails, it will die and call an error
$sth->finish();
##Indicate that no more data will be fetched from this statement handle before it is either executed again or destroyed.
$dbh->disconnect();
##Indicated DB connection can be disconnected
```

Workflow to Retrieve Data from DB

- Establish DB Connection (complete)
- Generate an SQL statement (partial)
- Process the response from the DB
- Post back appropriate results

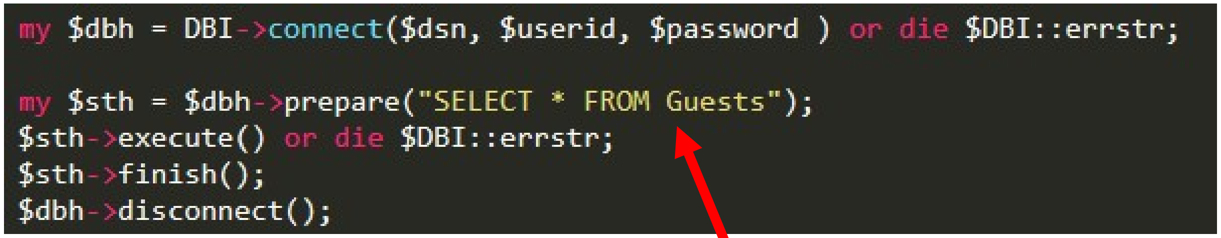
Perl Select Statement Generation

Previous code for insertion of records



```
my $dbh = DBI->connect($dsn, $userid, $password ) or die $DBI::errstr;

my $sth = $dbh->prepare("INSERT INTO Guests
                        (First, Last )
                         values
                        (?,?)");
## prepare statement. It contains all parts, except for the variables b
## question marks, designate where the variables values will be inserte
$sth->execute($firstname,$lastname) or die $DBI::errstr;
##places the needed variables into the insert statement and executes it
##if execution is not possible or fails, it will die and call an error
$sth->finish();
##Indicate that no more data will be fetched from this statement handle
$dbh->disconnect();
##Indicated DB connection can be disconnected
```

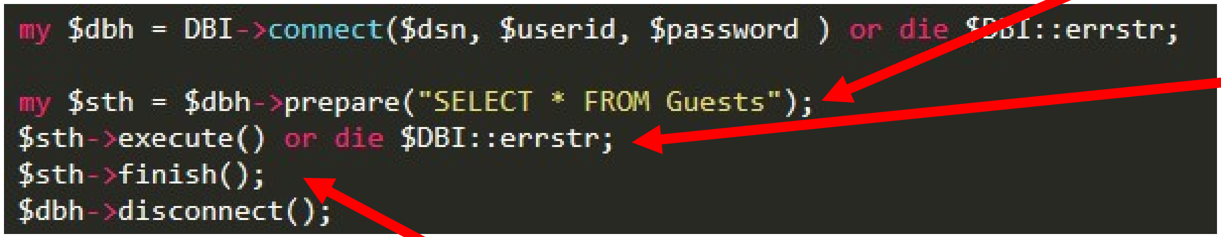
Code for retrieving records that have been inserted into table Guests



```
my $dbh = DBI->connect($dsn, $userid, $password ) or die $DBI::errstr;

my $sth = $dbh->prepare("SELECT * FROM Guests");
$sth->execute() or die $DBI::errstr;
$sth->finish();
$dbh->disconnect();
```

Should retrieve all records from the Table Guests

1 Prepare statement to retrieve all records from the Table Guests



```
my $dbh = DBI->connect($dsn, $userid, $password ) or die $DBI::errstr;

my $sth = $dbh->prepare("SELECT * FROM Guests");
$sth->execute() or die $DBI::errstr;
$sth->finish();
$dbh->disconnect();
```

Execute statement to retrieve all records from the Table Guests 2

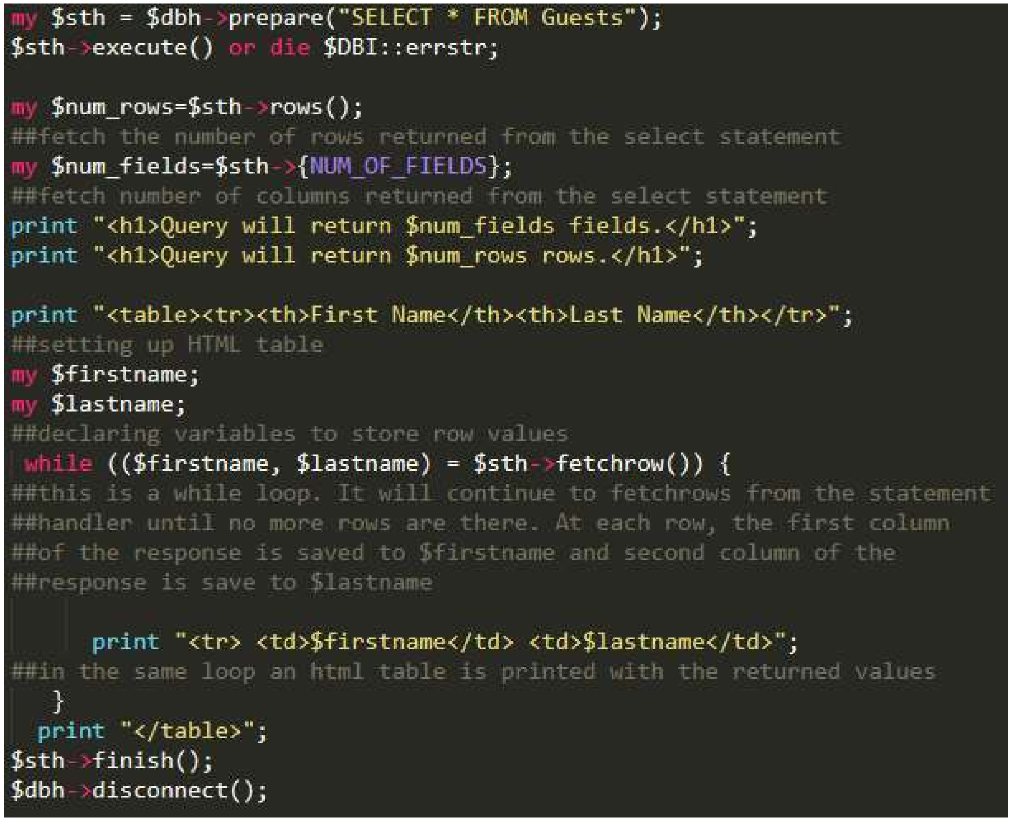
3 Indicate that no more data will be fetched from this statement handle before it is either executed again or destroyed.

2.5 **Data from statement handler needs to be processed in PERL!**

Perl Processing Database Response

- The statement handler has several attributes and methods
- Some important ones for receiving data are:
  - Fetchrow()
    - Fetches the next row of results for processing
  - Row()
    - Provides number of rows in the response
  - NUM_OF_FIELDS
    - Provides number of columns in the response

Perl Example

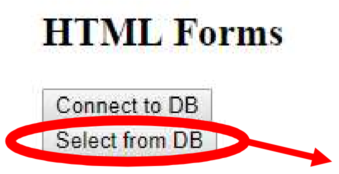


```perl
my $sth = $dbh->prepare("SELECT * FROM Guests");
$sth->execute() or die $DBI::errstr;

my $num_rows=$sth->rows();
##fetch the number of rows returned from the select statement
my $num_fields=$sth->{NUM_OF_FIELDS};
##fetch number of columns returned from the select statement
print "<h1>Query will return $num_fields fields.</h1>";
print "<h1>Query will return $num_rows rows.</h1>";

print "<table><tr><th>First Name</th><th>Last Name</th></tr>";
##setting up HTML table
my $firstname;
my $lastname;
##declaring variables to store row values
 while (($firstname, $lastname) = $sth->fetchrow()) {
##this is a while loop. It will continue to fetchrows from the statement
##handler until no more rows are there. At each row, the first column
##of the response is saved to $firstname and second column of the
##response is save to $lastname

   print "<tr> <td>$firstname</td> <td>$lastname</td>";
##in the same loop an html table is printed with the returned values
  }
 print "</table>";
$sth->finish();
$dbh->disconnect();
```

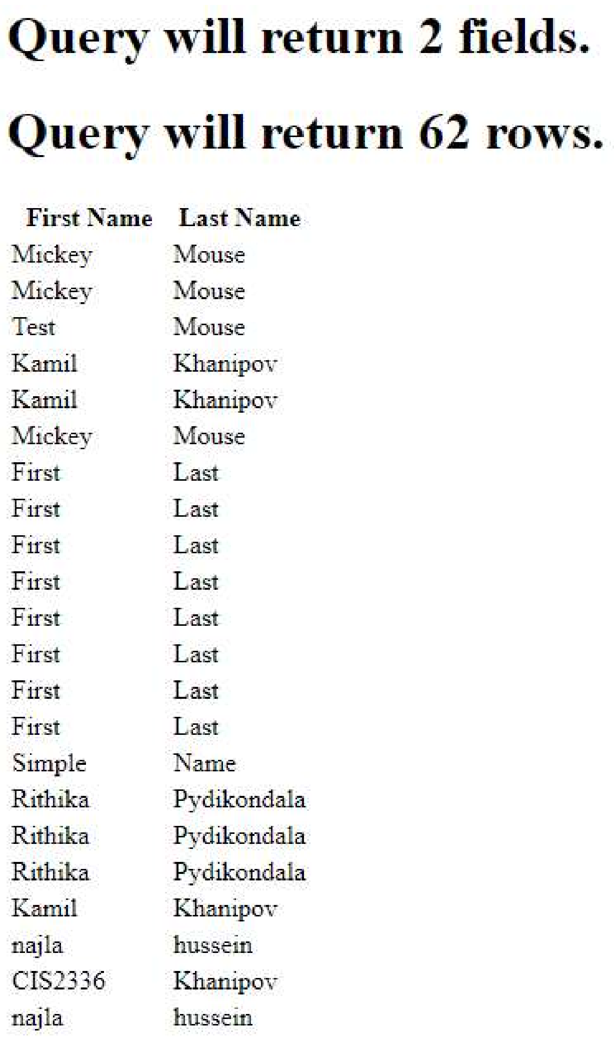
DIY

- Take the Perl file select_from.pl and adjust the database connection details to those from last class (if needed correct table name as well)
- Make an empty HTML page with one submit button with an “action” attribute set to select_from.pl
- Observe! You should see a list of all guests that have their names in the DB
- If you have “stubborn” web page, which refused to appear in your web page, then it “cashed”. To fix that in PC: Ctrl_F5; in MAC: Command_Shift_R.

Example: Demo



HTML Forms



Query will return 2 fields.

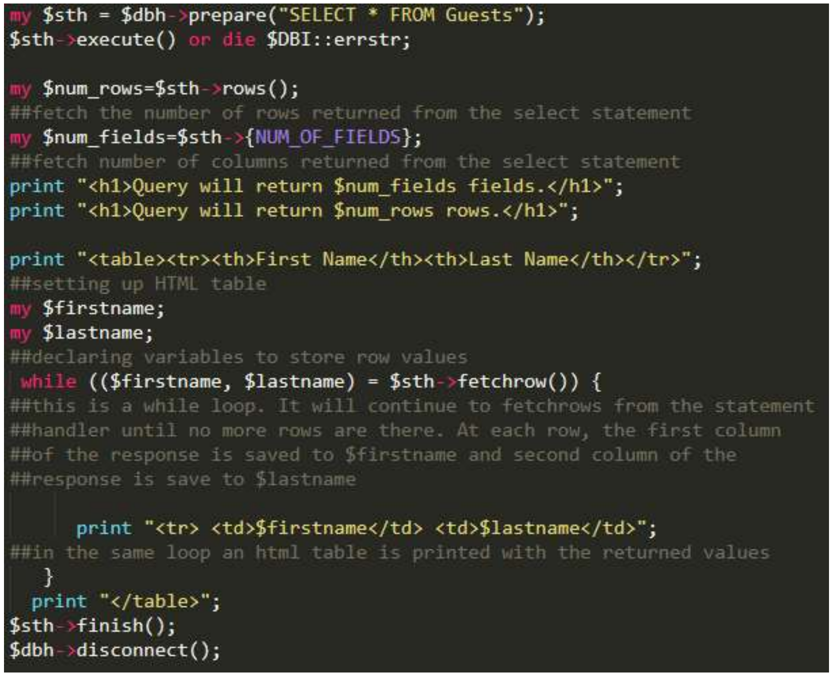
Query will return 62 rows.

First Name	Last Name
Mickey	Mouse
Mickey	Mouse
Test	Mouse
Kamil	Khanipov
Kamil	Khanipov
Mickey	Mouse
First	Last
First	Last
First	Last
First	Last
First	Last
First	Last
First	Last
First	Last
Simple	Name
Rithika	Pydikondala
Rithika	Pydikondala
Rithika	Pydikondala
Kamil	Khanipov
najla	hussein
CIS2336	Khanipov
najla	hussein

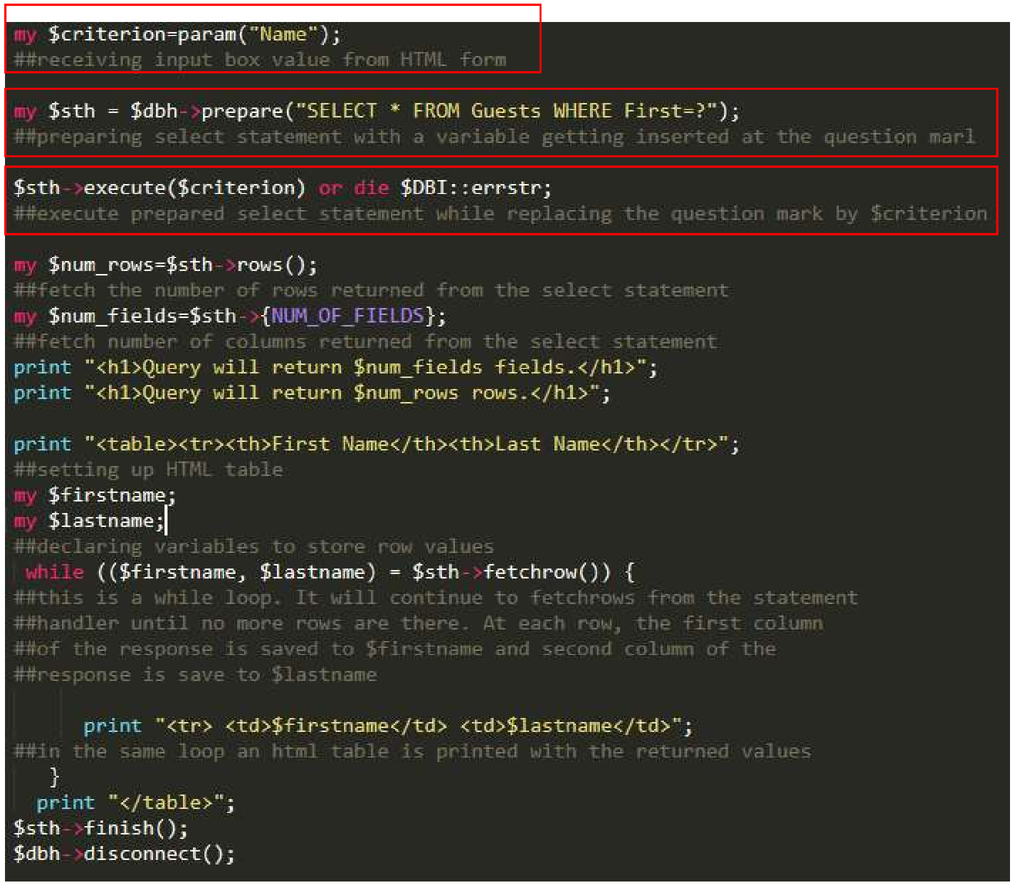
Customization of Select Statement

- Same as with insert statements we can customize Select statements based on input from an HTML form
- For example, expand previous example searching for guests by first name instead of listing all of them. Whats needed?
  - HTML Form with input field and submit button
  - Perl script
    - Get input field value from HTML form using param() function
    - Establish DB connection
    - Generate Select statement
    - Execute Select statement (inserting variables for first name)
    - Process and output results of select statement

Adding functionality to Perl Script

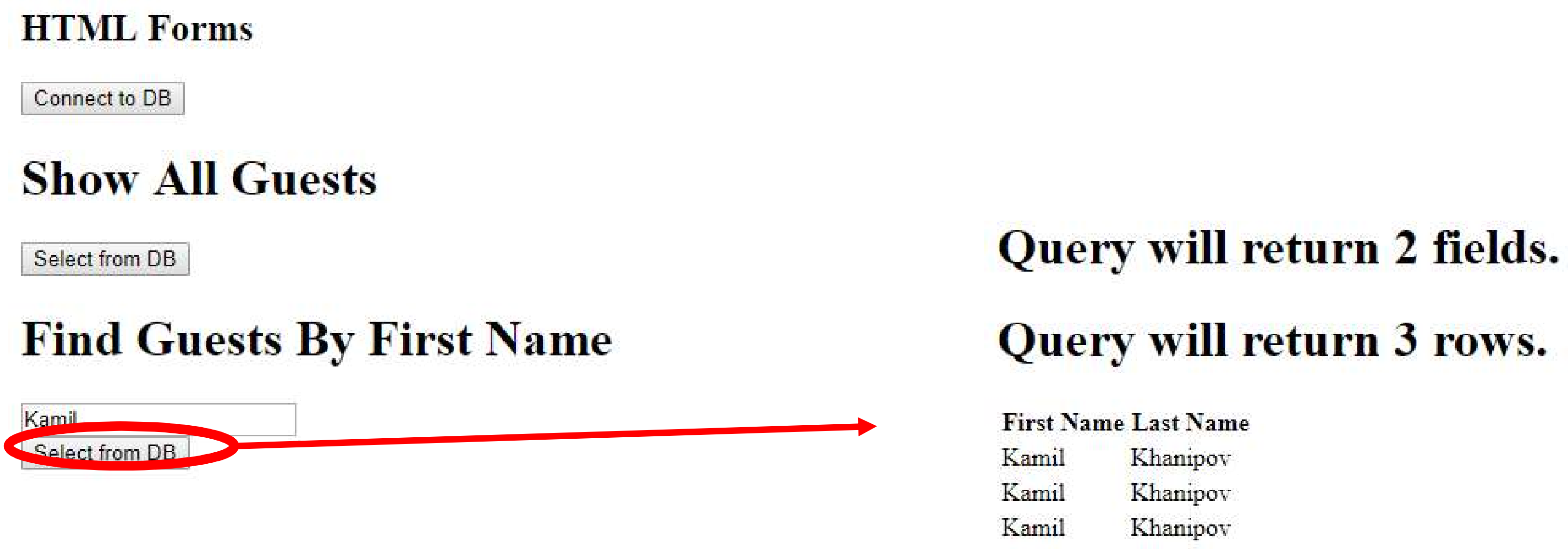


```perl
my $sth = $dbh->prepare("SELECT * FROM Guests");
$sth->execute() or die $DBI::errstr;

my $num_rows=$sth->rows();
##fetch the number of rows returned from the select statement
my $num_fields=$sth->{NUM_OF_FIELDS};
##fetch number of columns returned from the select statement
print "<h1>Query will return $num_fields fields.</h1>";
print "<h1>Query will return $num_rows rows.</h1>";

print "<table><tr><th>First Name</th><th>Last Name</th></tr>";
##setting up HTML table
my $firstname;
my $lastname;
##declaring variables to store row values
 while (($firstname, $lastname) = $sth->fetchrow()) {
##this is a while loop. It will continue to fetchrows from the statement
##handler until no more rows are there. At each row, the first column
##of the response is saved to $firstname and second column of the
##response is save to $lastname

      print "<tr> <td>$firstname</td> <td>$lastname</td>";
##in the same loop an html table is printed with the returned values
   }
  print "</table>";
$sth->finish();
$dbh->disconnect();
```



```perl
my $criterion=param("Name");
##receiving input box value from HTML form

my $sth = $dbh->prepare("SELECT * FROM Guests WHERE First=?");
##preparing select statement with a variable getting inserted at the question marl

$sth->execute($criterion) or die $DBI::errstr;
##execute prepared select statement while replacing the question mark by $criterion

my $num_rows=$sth->rows();
##fetch the number of rows returned from the select statement
my $num_fields=$sth->{NUM_OF_FIELDS};
##fetch number of columns returned from the select statement
print "<h1>Query will return $num_fields fields.</h1>";
print "<h1>Query will return $num_rows rows.</h1>";

print "<table><tr><th>First Name</th><th>Last Name</th></tr>";
##setting up HTML table
my $firstname;
my $lastname;
##declaring variables to store row values
 while (($firstname, $lastname) = $sth->fetchrow()) {
##this is a while loop. It will continue to fetchrows from the statement
##handler until no more rows are there. At each row, the first column
##of the response is saved to $firstname and second column of the
##response is save to $lastname

      print "<tr> <td>$firstname</td> <td>$lastname</td>";
##in the same loop an html table is printed with the returned values
   }
  print "</table>";
$sth->finish();
$dbh->disconnect();
```

Example: Demo



HTML Forms

Show All Guests

Find Guests By First Name

Kamil

Query will return 2 fields.

Query will return 3 rows.

First Name	Last Name
Kamil	Khanipov
Kamil	Khanipov
Kamil	Khanipov

DIY: Make the Jump

- Start with form_extended.html and the DB you created last week for it
- Populate the DB with 5-10 entries.
- Add an additional form in the html file with an input field that submits to a new Perl Script.
- Have the new Perl Script process the input field based on first name, and generate a "SELECT FROM WHERE" statement, and process it.
- Try various search queries.
- Submit your code, screenshot of DB, and HTML page output.
- Extra thought: Check number of rows returned from statement handler, if nothing is returned you could display an error message.