SQL Programming

Sorting Rows in the Result Table

Page A-1: Intro

One of the principles of a true relational database system is that there is no implicit ordering of either the columns or rows in the database tables.

The programmer should never expect to see any particular row as the first record, nor should she expect to see any particular row as the last record.

Rows of data in the base table just exist.

At least that's the theory. The behavior of many database systems is rather predictable, and although they shouldn't, many programmers still rely on their expectations.

Page B-1 ORDER BY

In a previous lesson we mentioned that SQL provided CRUD (create, retrieve, update, and delete) functionality.

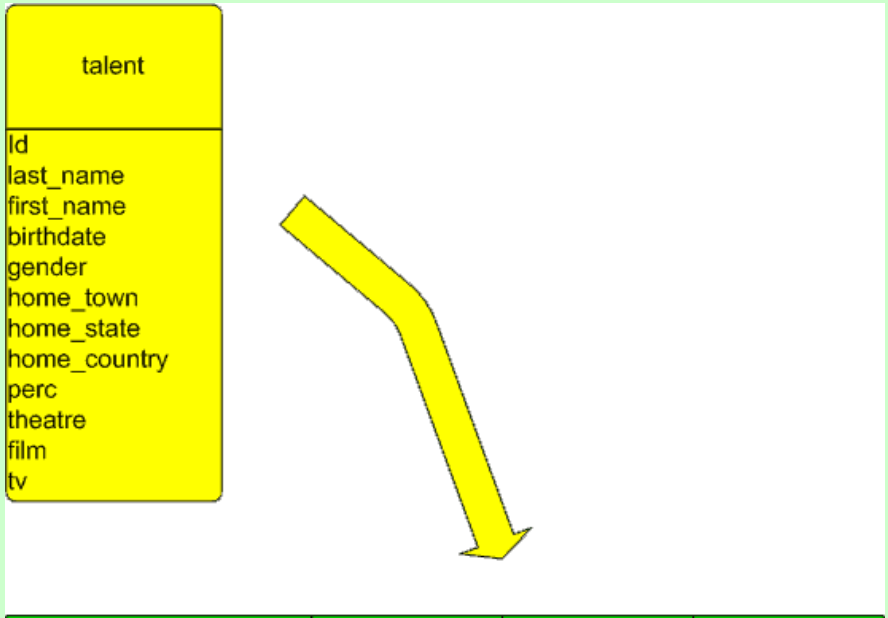
The practicality of using SQL to retrieve information from a database necessitates the ability to arrange the output and to be able to present the rows of the result table in some sorted order.

The ORDER BY clause is the mechanism in SQL that gives the programmer the ability to sort the rows in the result table.

Page B-2 Problem 8-1 Statement

The user community has requested an alphabetical listing of all of our clients.

Rephrase: Prepare a report showing the name information for each of our clients. Sort this report in ascending last_name order.



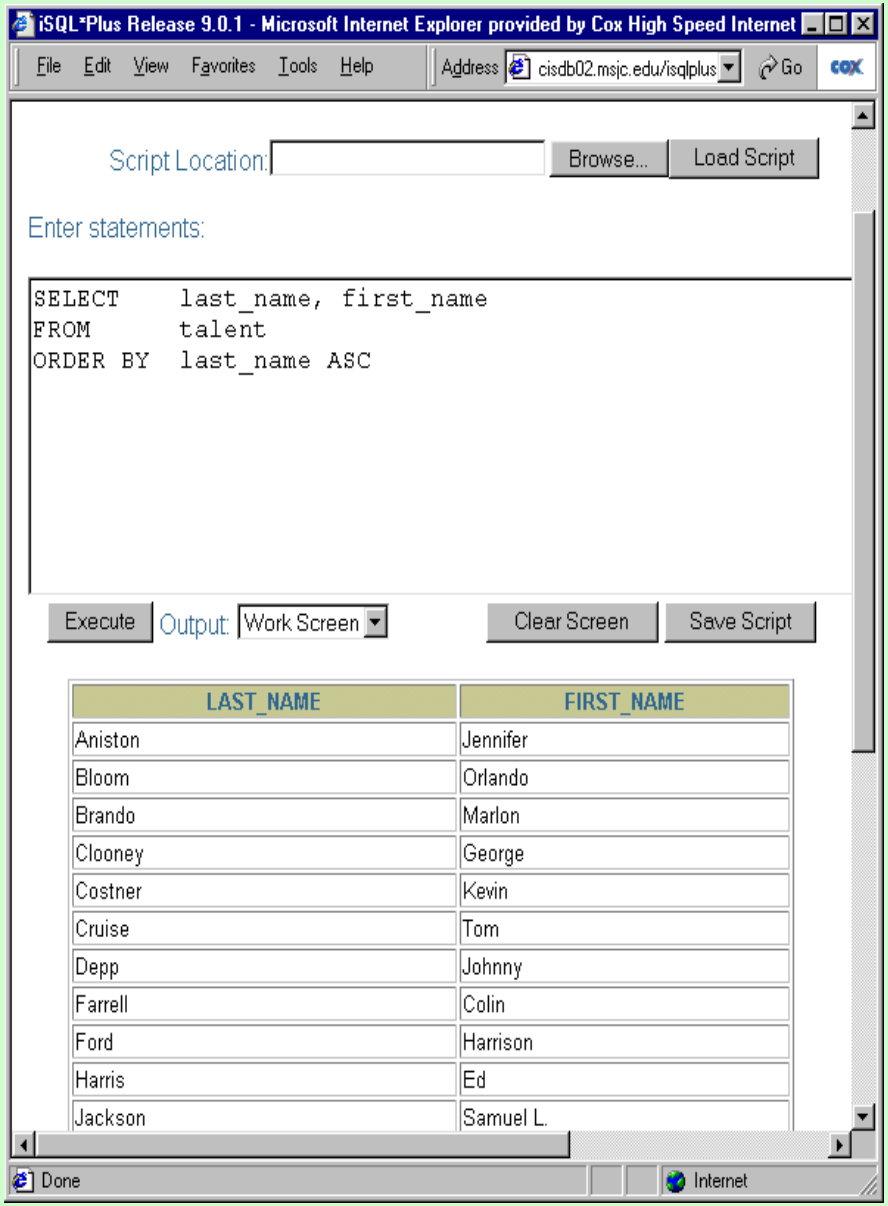
Column Name/Expression	last_name	first_name	...
Table Name	talent	talent	...
Alias	Last Name	First Name	...
Criteria	...		...
Display	Sort: asc		...

So let's put this together.

Step 1: Build the Table Build Chart (TBC)
Step 2: Double check your TBC solution
Step 3: Transform the TBC into code.

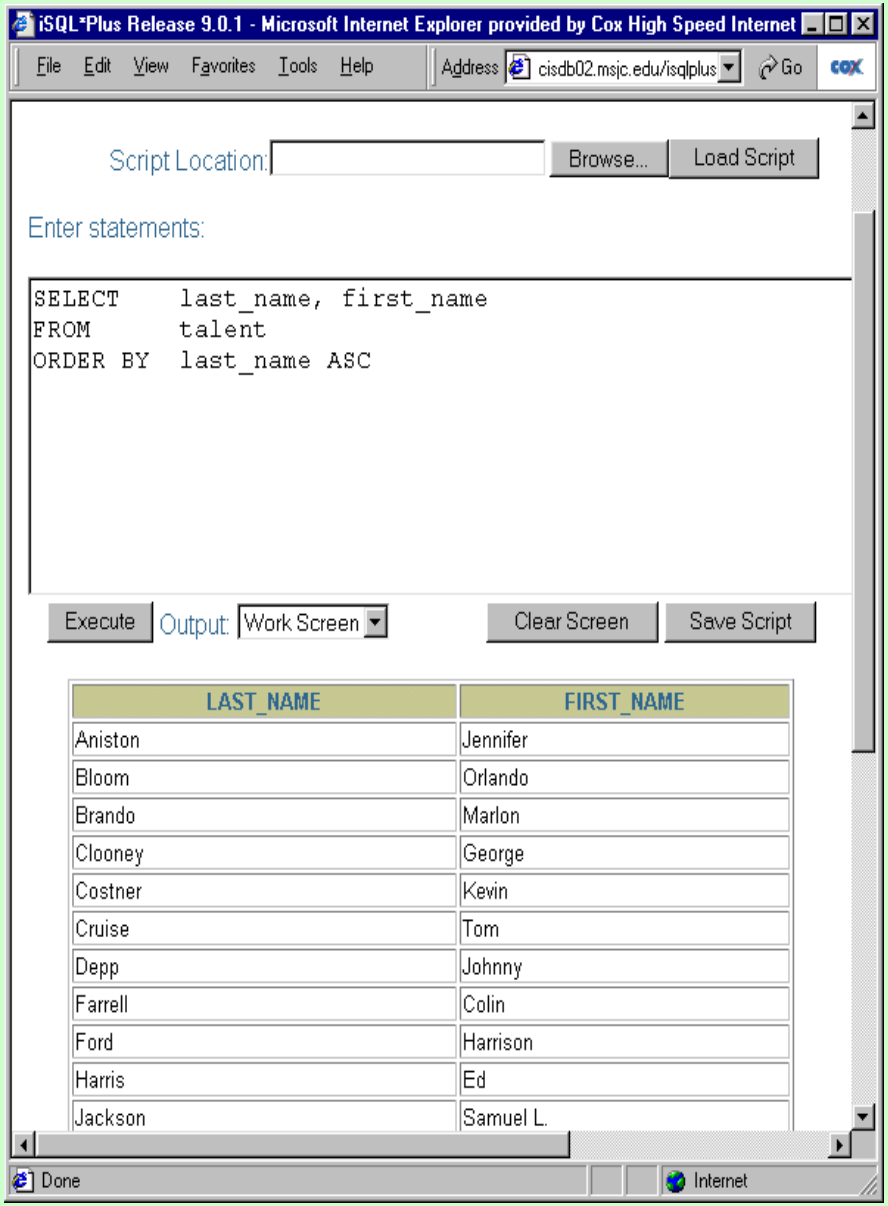
Notice how sort criteria are specified in the display row.

```
SELECT     last_name, first_name
FROM       talent
ORDER BY   last_name ASC
```



Here's the solution.

Page B-5: Problem 8-1 Analysis



The ORDER BY clause is the last clause in the SQL program.

If the information is to be sorted in ascending order, then the keyword ASC follows the name of the column. If the information is to be sorted in descending order, then the keyword DESC follows the column name.

Note also that there is no comma between the column name and the sorting criteria in the ORDER BY clause.

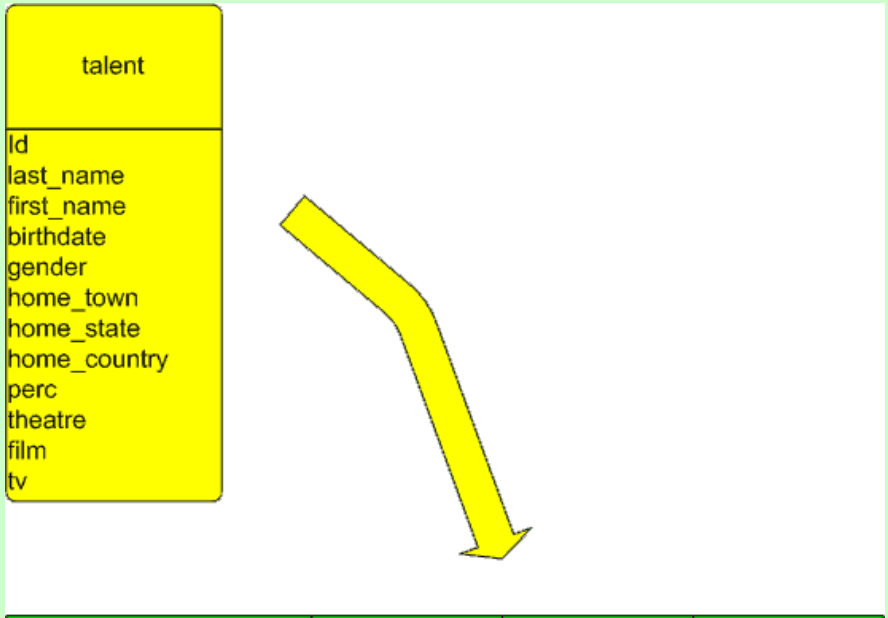
Page B-6 Problem 8-2 Statement

The user community has requested a listing of all of our clients arranged by age, with the youngest clients appearing at the top of the list.

Rephrase: Prepare a report showing the name and birth date information for each of our clients. Sort this report in descending birth date order.

 Problem 8-3 Code & Design

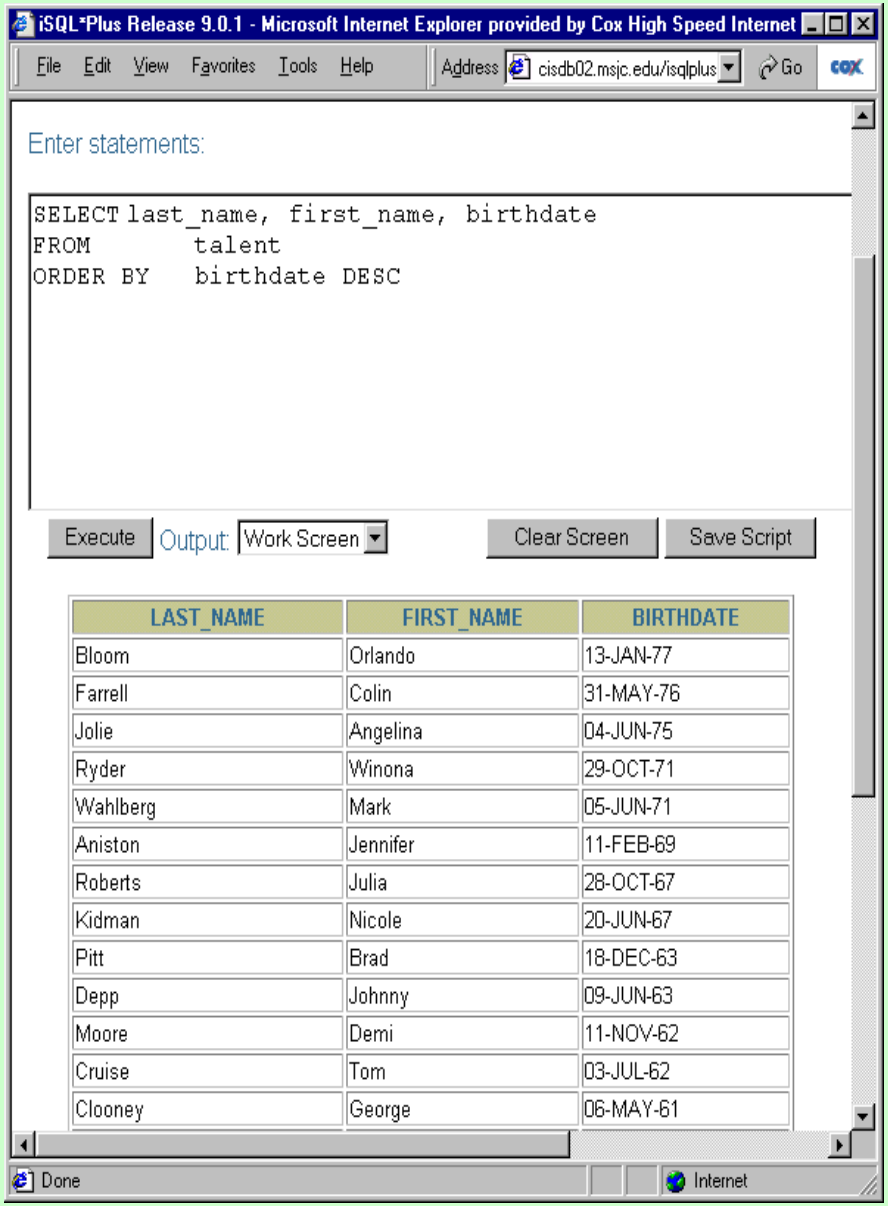
Step 1: Build the Table Build Chart (TBC)
Step 2: Double check your TBC solution
Step 3: Transform the TBC into code.



Column Name/Expression	last_name	first_name	birthdate
Table Name	talent	talent	...
Alias	Last Name	First Name	...
Criteria	...		...
Display			Sort: desc

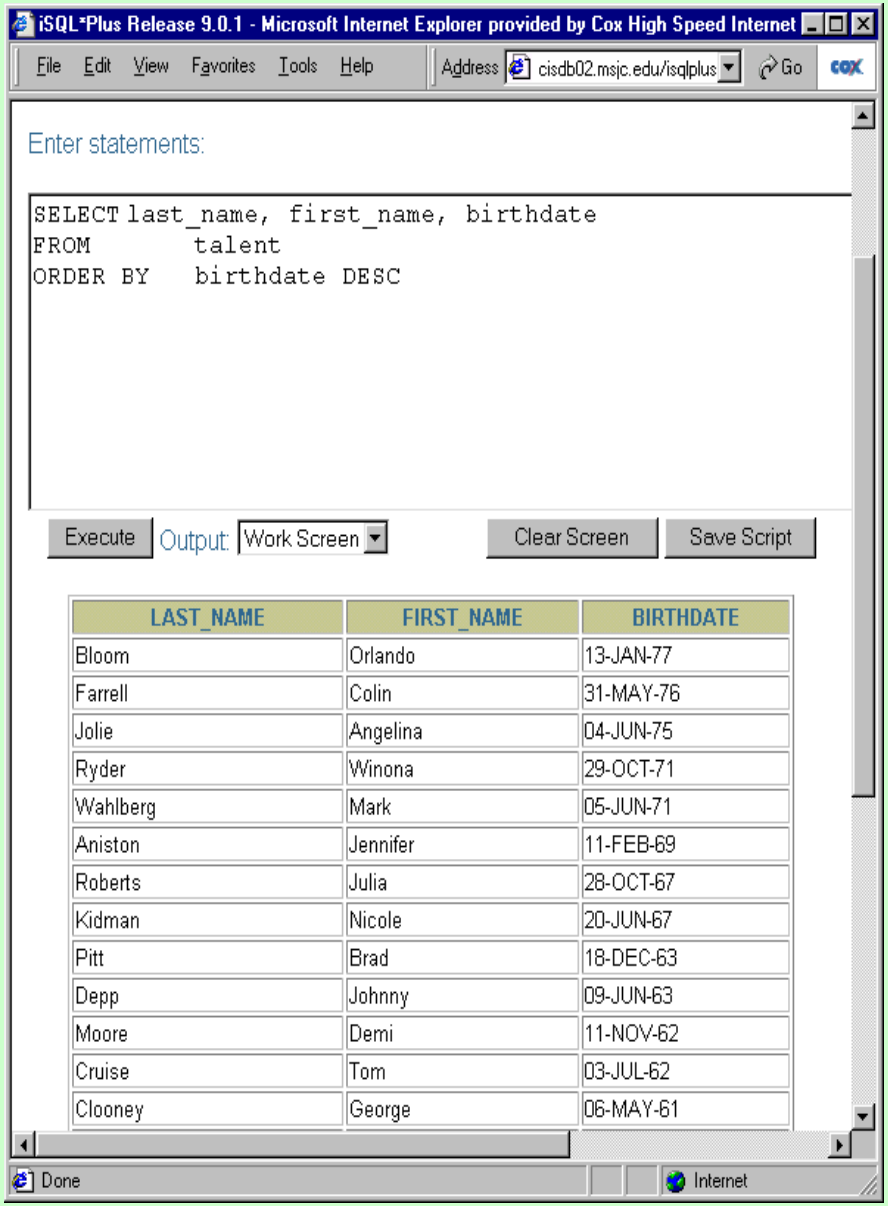
Notice how the sort criteria are specified in the display row.

```
SELECT     last_name, first_name, birthdate
FROM       talent
ORDER BY   birthdate DESC
```



Here's the solution.

Page B-9: Problem 8-2 Analysis



Notice that the sort on the birth date column is not an alphabetic sort. Rather the birth date column is sorted in the correct fashion, based on the date values.

This is just another example of the benefits of data types. Data types define the structure, format, and behavior of the objects in the data base.

One interesting note about the elements of the ORDER BY clause, is that it is not required that these elements also appear in the SELECT clause.

Consider the two different solutions that are presented on the following page.

LAST_NAME	FIRST_NAME	BIRTHDATE
Bloom	Orlando	13-JAN-77
Farrell	Colin	31-MAY-76
Jolie	Angelina	04-JUN-75
Ryder	Winona	29-OCT-71
Wahlberg	Mark	05-JUN-71
Aniston	Jennifer	11-FEB-69
Roberts	Julia	28-OCT-67
Kidman	Nicole	20-JUN-67
Pitt	Brad	18-DEC-63
Depp	Johnny	09-JUN-63
Moore	Demi	11-NOV-62
Cruise	Tom	03-JUL-62
Clooney	George	06-MAY-61
Pfeiffer	Michelle	29-APR-58

LAST_NAME	FIRST_NAME
Bloom	Orlando
Farrell	Colin
Jolie	Angelina
Ryder	Winona
Wahlberg	Mark
Aniston	Jennifer
Roberts	Julia
Kidman	Nicole
Pitt	Brad
Depp	Johnny
Moore	Demi
Cruise	Tom
Clooney	George
Pfeiffer	Michelle

Page B-12 Multicolumn Sorting

Occasionally the SQL programmer needs to sort on a secondary column, or series of columns.

The programmer can specify additional columns to sort on, and these secondary sort columns indicate how the output should be sorted, on those occasions when all of the values in the preceding columns are the same.

In other words, the secondary sort columns ‘kick in’ only when there are duplicate values in the primary sort columns.

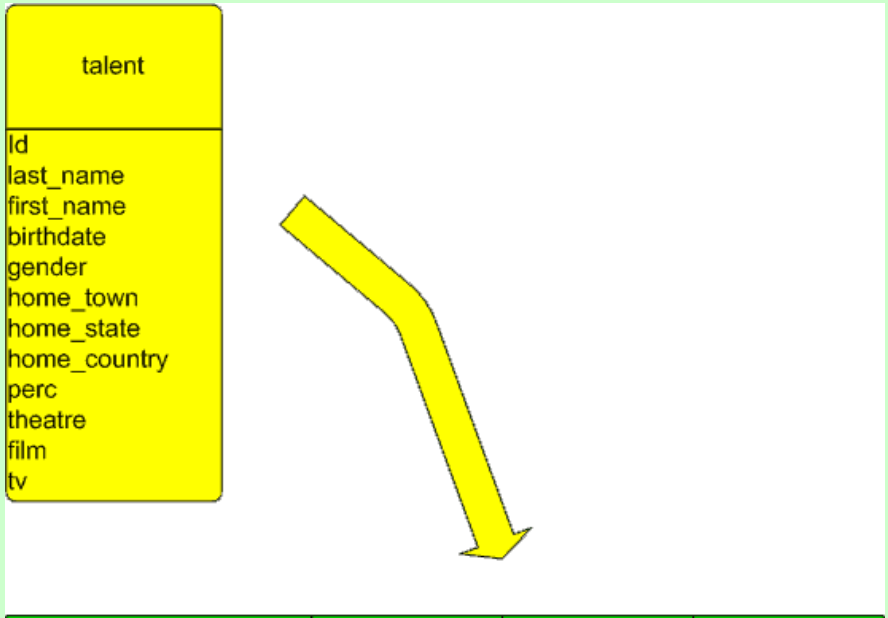
Page B-13 Problem 8-3 Statement

The user community has requested a listing of all of our clients alphabetized by last name, within country of origin.

Rephrase: Prepare a report showing the name and home country information for each of our clients. Sort this report using home country as the primary sort field, and last name as the secondary sort field.

Page B-14 Problem 8-3 Code & Design

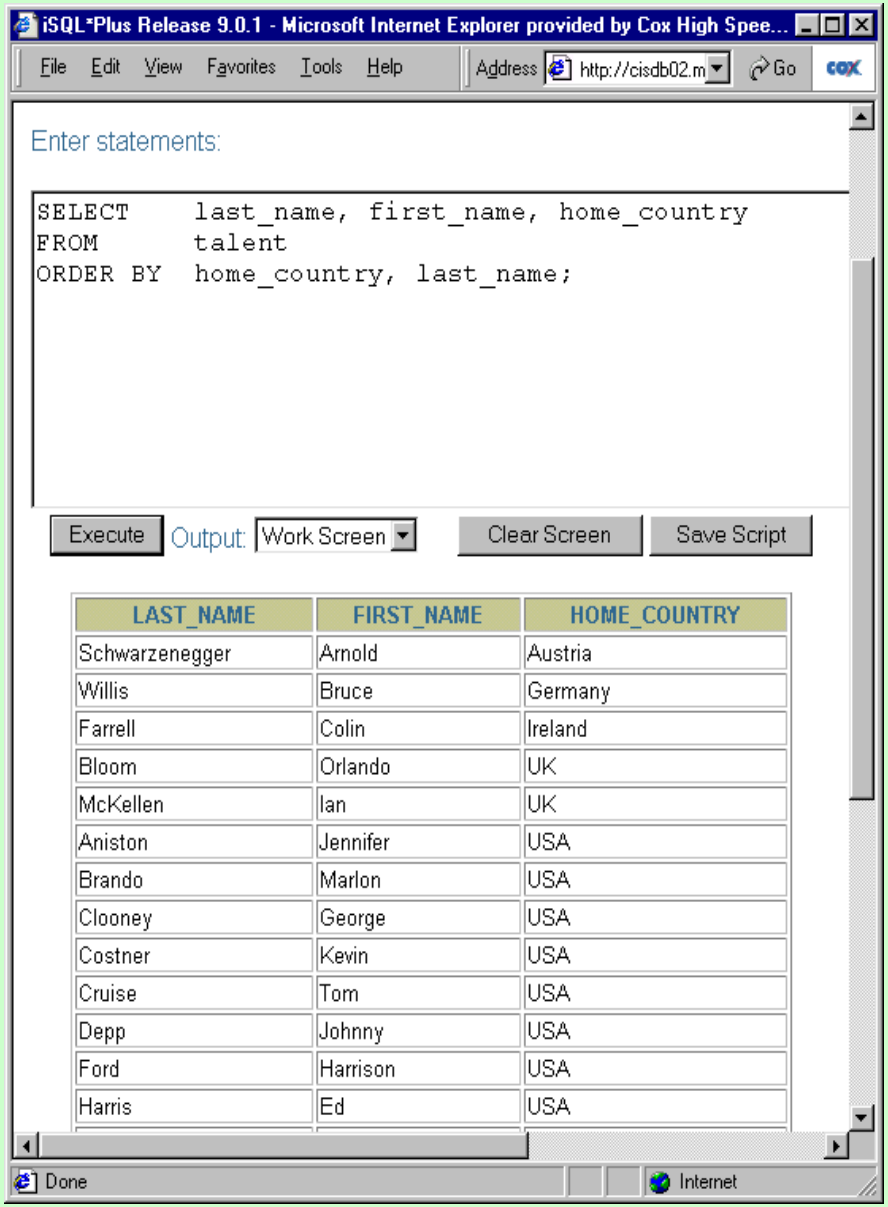
Step 1: Build the Table Build Chart (TBC)
Step 2: Double check your TBC solution
Step 3: Transform the TBC into code.



Column Name/Expression	last_name	first_name	Home country
Table Name	talent	talent	...
Alias	Last Name	First Name	...
Criteria	...		...
Display	Sort(2): asc		Sort(1): asc

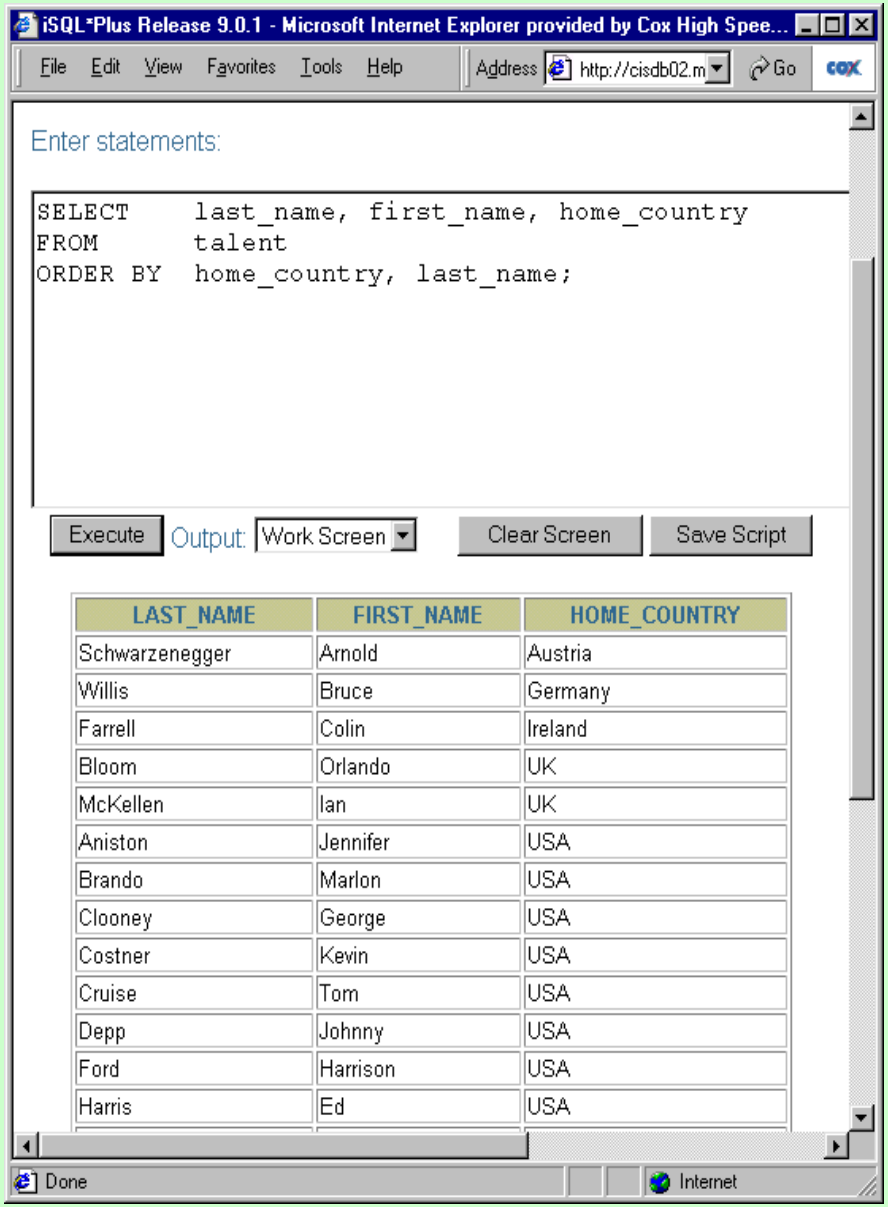
Notice how the sort criteria is specified in the display row.

```
SELECT    last_name, first_name, home_country
FROM      talent
ORDER BY  home_country, last_name;
```



Here's the solution.

According to this sort, all of the rows are arranged primarily in home_country order. And then, when there are duplicate values in the home_country column, those rows are sorted in last_name order.



Ascending is the default sort option, and since the primary and secondary sorts were both ascending sorts, the programmer opted to omit that keyword from the code.

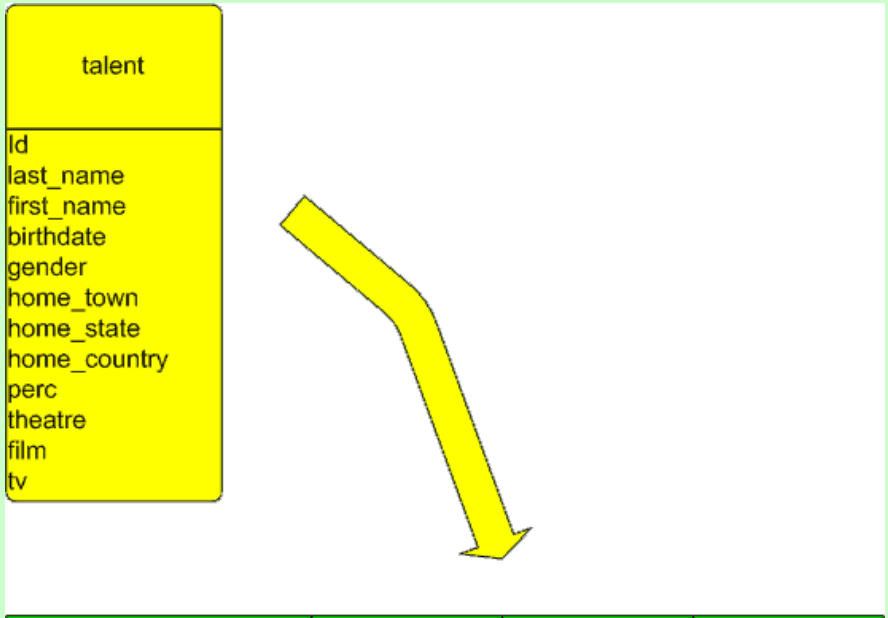
What do you think about that coding style?

Page B-13 Problem 8-4 Statement

The user community has requested a listing of all of our clients alphabetized by last name, within country of origin. But this time, put the Americans at the top of the list.

Rephrase: Prepare a report showing the name and home country information for each of our clients. Sort this report using home country as the primary sort field (in descending order), and last name as the secondary sort field (in ascending order).

Step 1: Build the Table Build Chart (TBC)
Step 2: Double check your TBC solution
Step 3: Transform the TBC into code.



Column Name/Expression	last_name	first_name	Home country
Table Name	talent	talent	...
Alias	Last Name	First Name	...
Criteria	...		...
Display	Sort(2): asc		Sort(1): desc

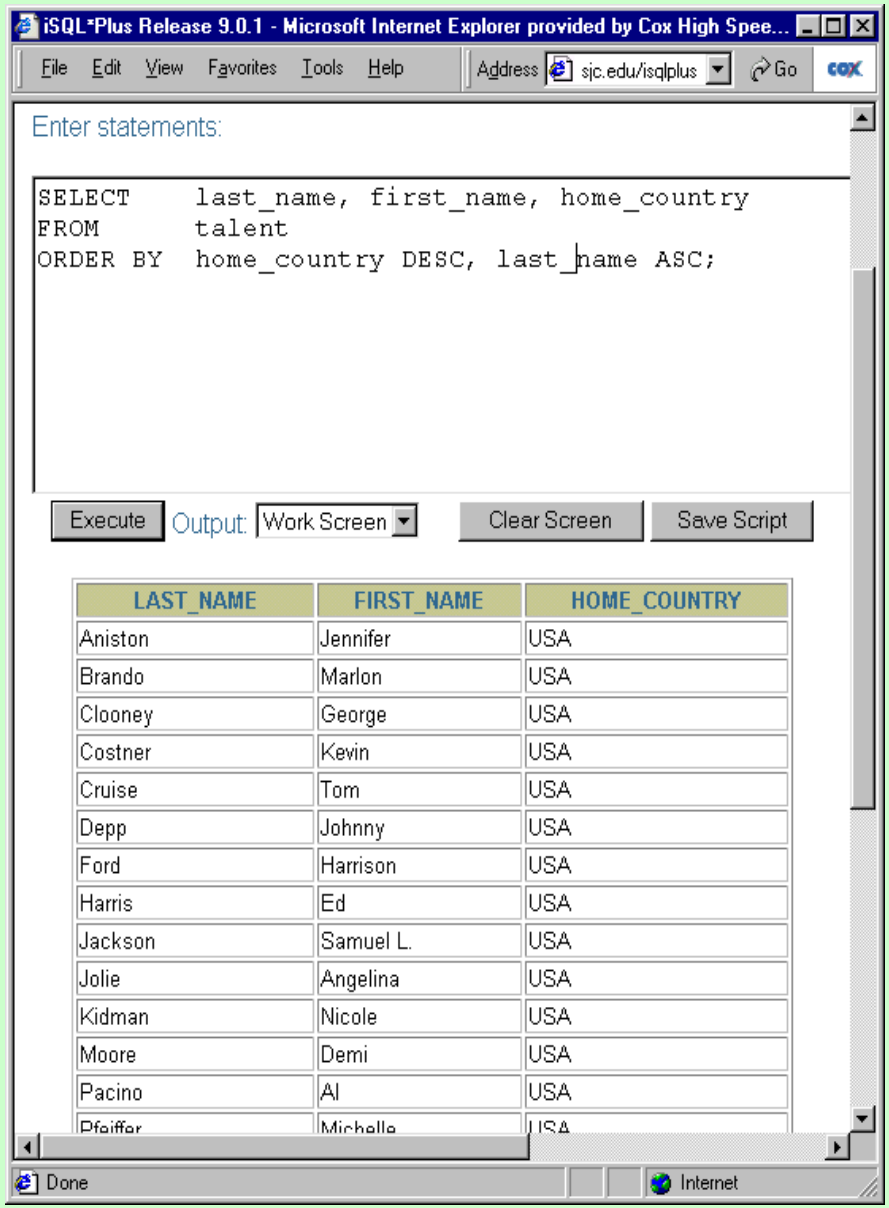
Notice how the sort criteria is specified in the display row.

```
SELECT    last_name, first_name, home_country
FROM      talent
ORDER BY  home_country DESC, last_name ASC;
```

Here's the solution.

Notice that the comma in the ORDER BY clause separates one *chunk* of sort information from the other *chunk*.

Page B-16: Problem 8-4 Analysis



The user community asked to have the Americans listed at the top of the report, and in this case the programmer used a trick, and just 'lucked out' with this program.

This program only works because USA, at present, is at the end of the list. But what happens when we add talent from West Germany, or Yugoslavia?

Given what we've discussed about SQL, this is the only solution available to you, but I hope you felt a little uneasy about this solution. I hope it didn't feel right to you.

This solution only works because the programmer knows about all of the data in the database. And while it's great to have those kinds of intuitions, it's impossible to know about all of the data, all of the time.

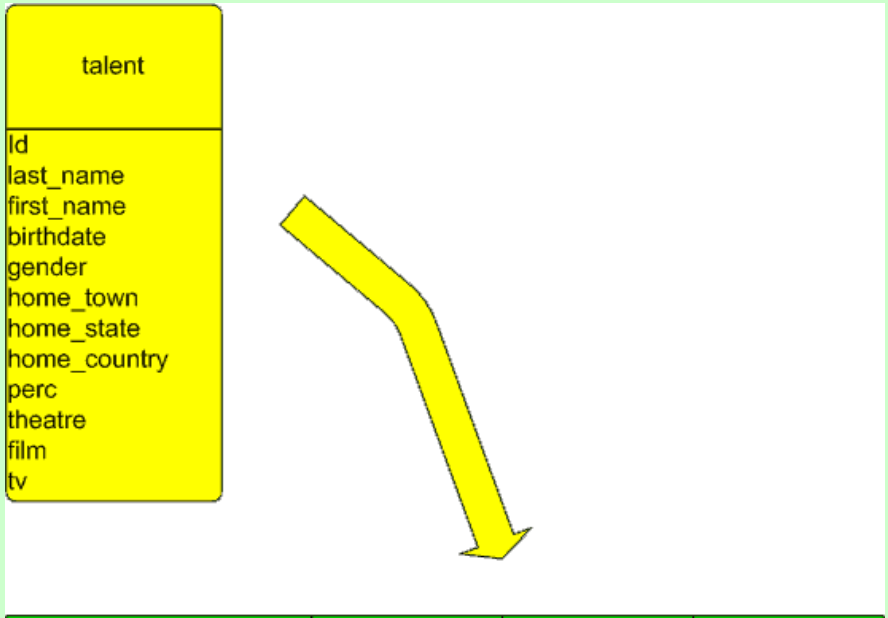
Page B-13 Problem 8-5 Statement

The user community has requested a listing of all of our clients alphabetized by last name, within country of origin. But this time, exclude the Americans from the report.

Rephrase: Prepare a report showing the name and home country information for each of our clients.

Sort this report using home country as the primary sort field, and last name as the secondary sort field. Include only those records for talent who were born outside of the USA.

Step 1: Build the Table Build Chart (TBC)
Step 2: Double check your TBC solution
Step 3: Transform the TBC into code.

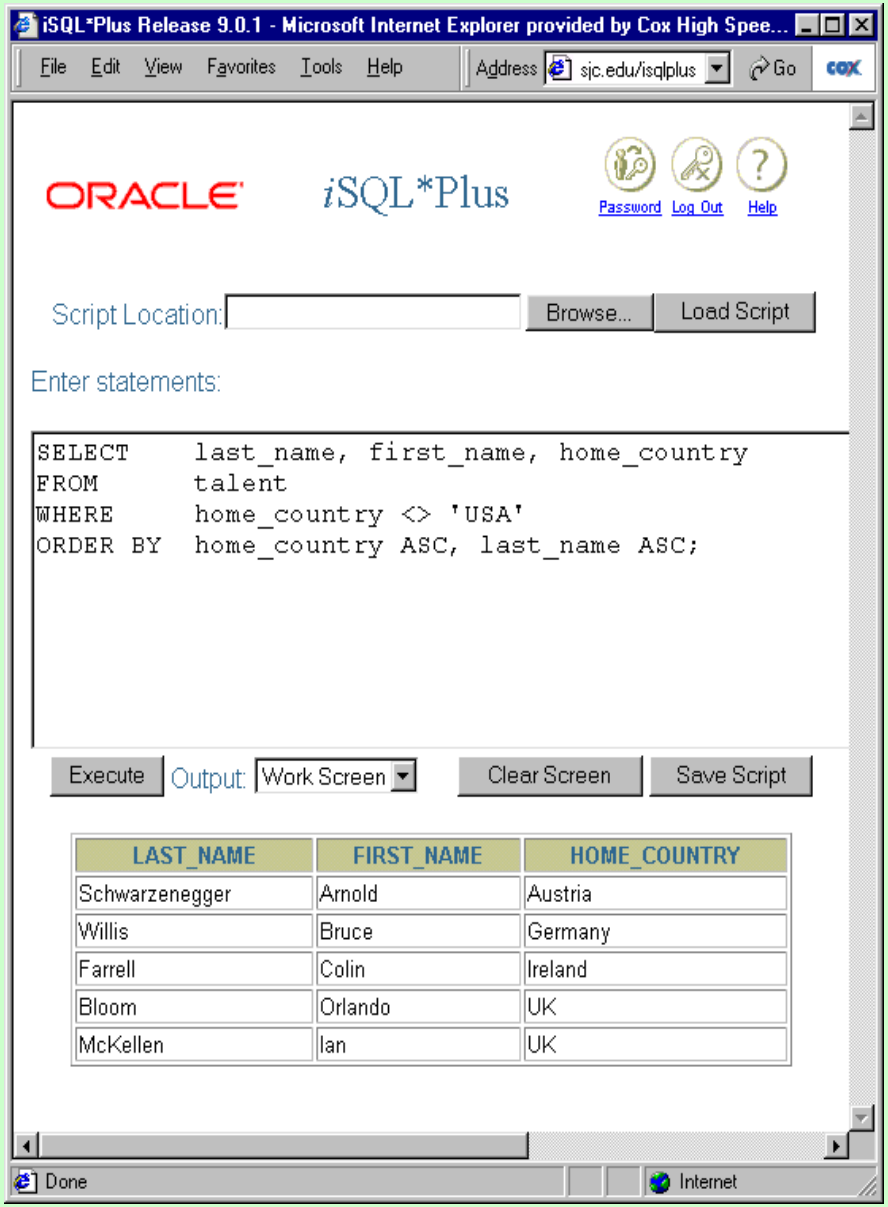


Column Name/Expression	last_name	first_name	Home country
Table Name	talent	talent	...
Alias	Last Name	First Name	...
Criteria	...		<> USA
Display	Sort(2): asc		Sort(1): desc

The TBC is getting pretty busy.

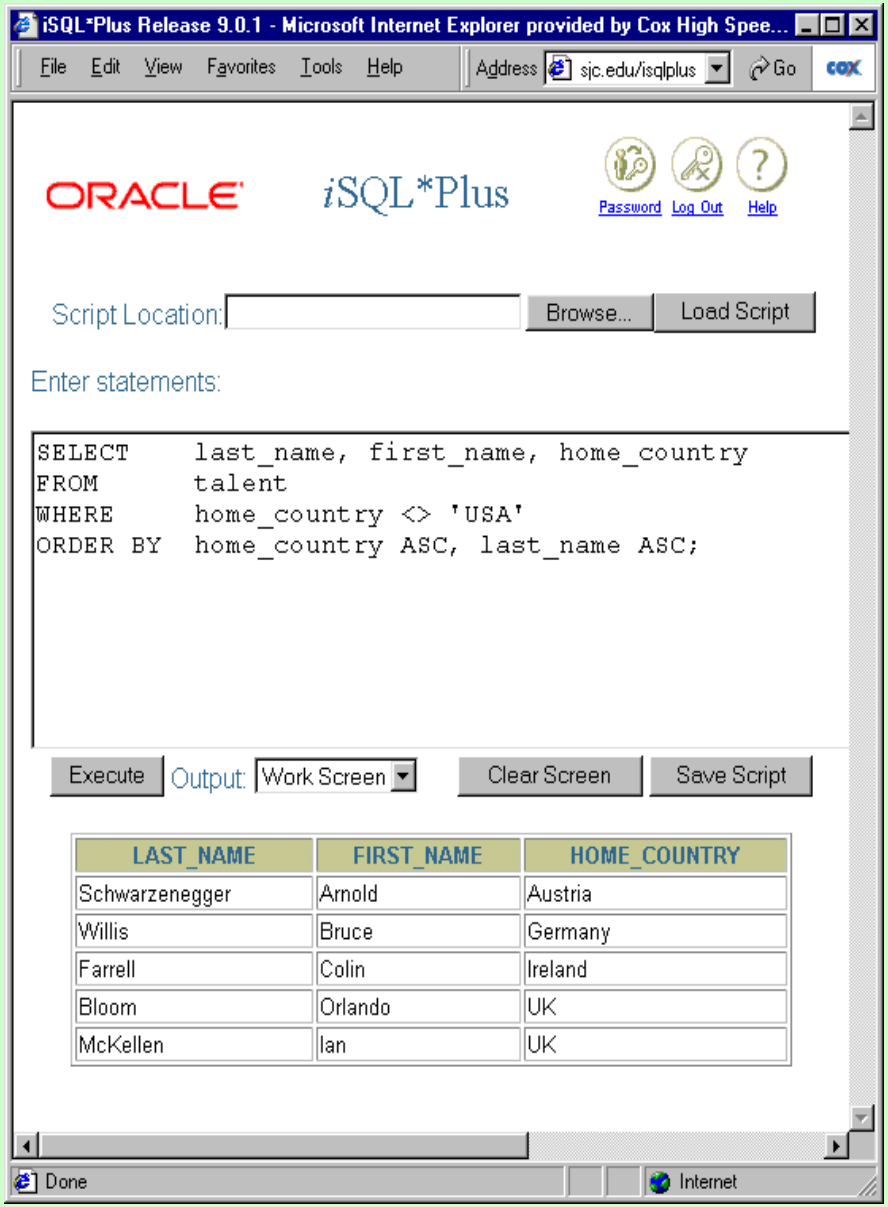
Notice how the home_country column specifies both a sort field, as well as criterion for inclusion

```
SELECT    last_name, first_name, home_country
FROM      talent
WHERE     home_country <> 'USA'
ORDER BY  home_country ASC, last_name ASC;
```



Here's the solution.

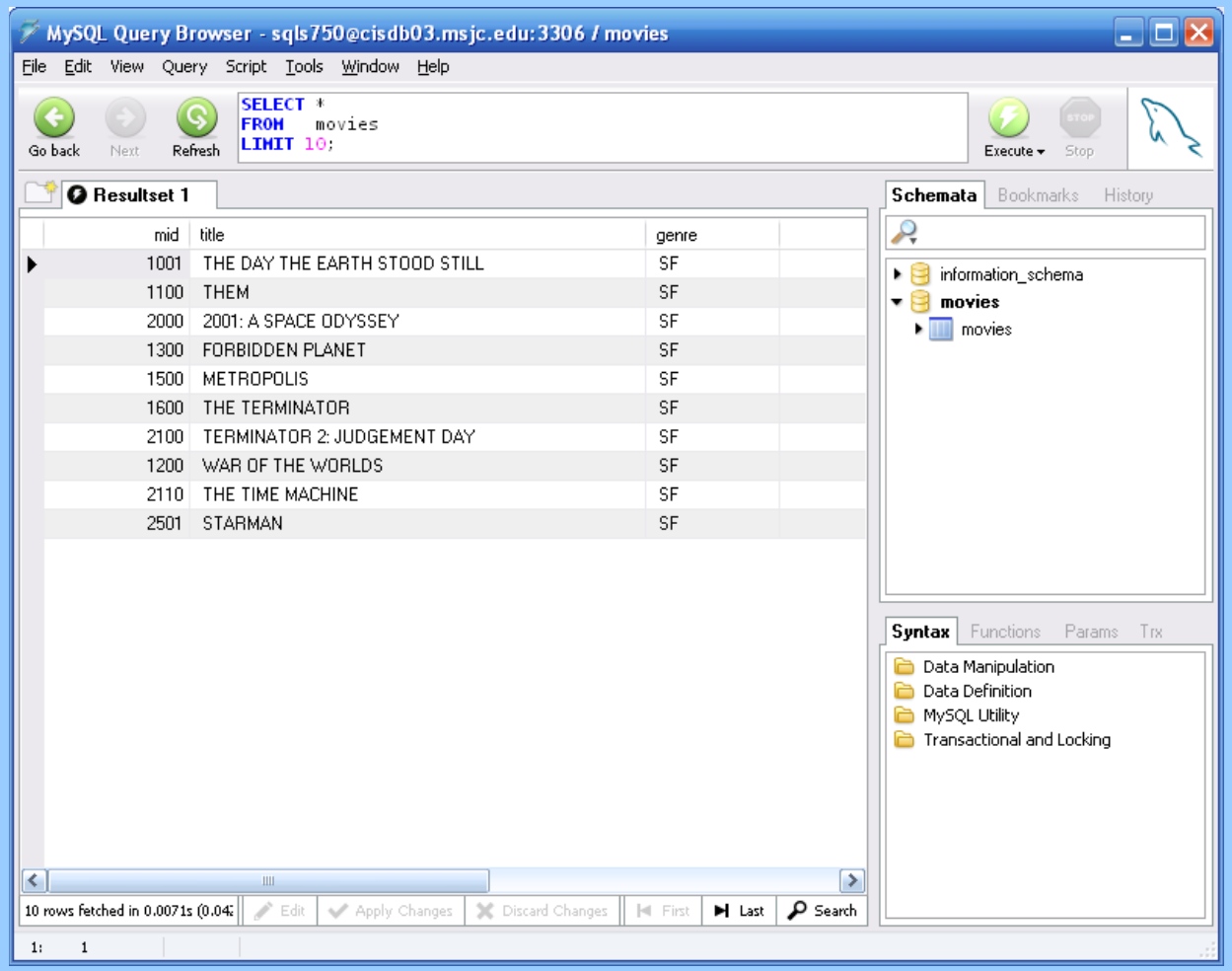
Page B-16: Problem 8-5 Analysis



This program demonstrates how the WHERE clause can be used in conjunction with the ORDER BY clause.

Page C-1: Limiting Output

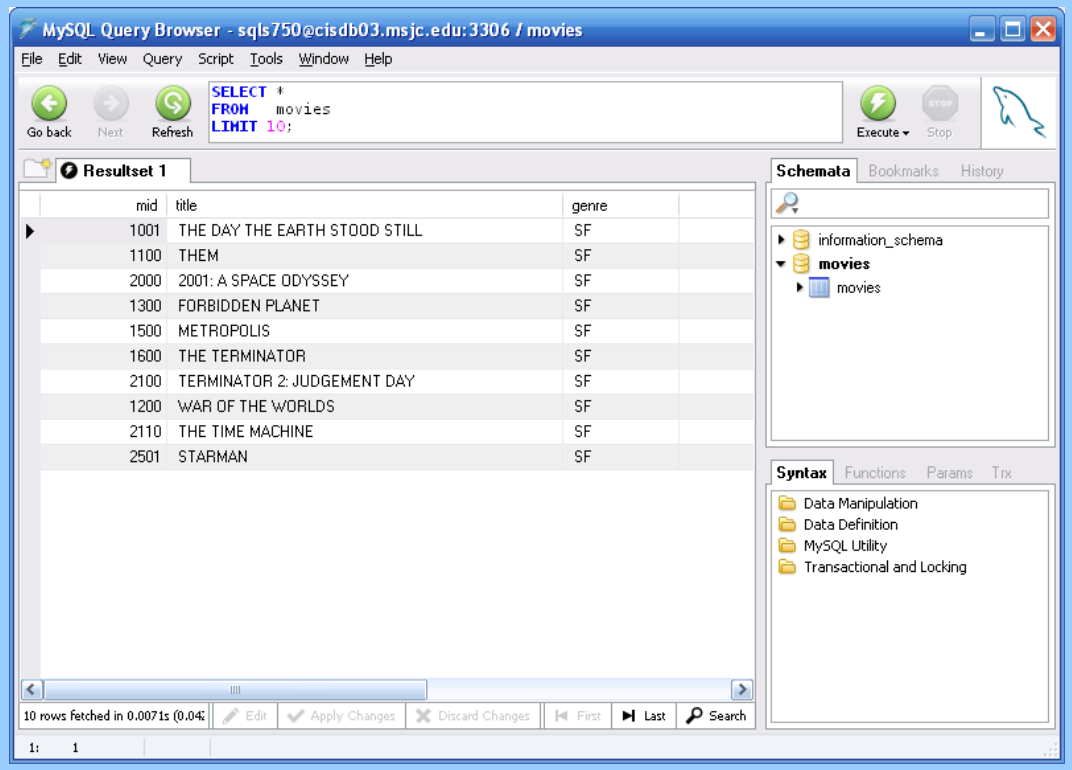
In MySQL we can use the LIMIT clause to limit the number of lines of output that will be displayed in the final result table.



The LIMIT clause has two forms:

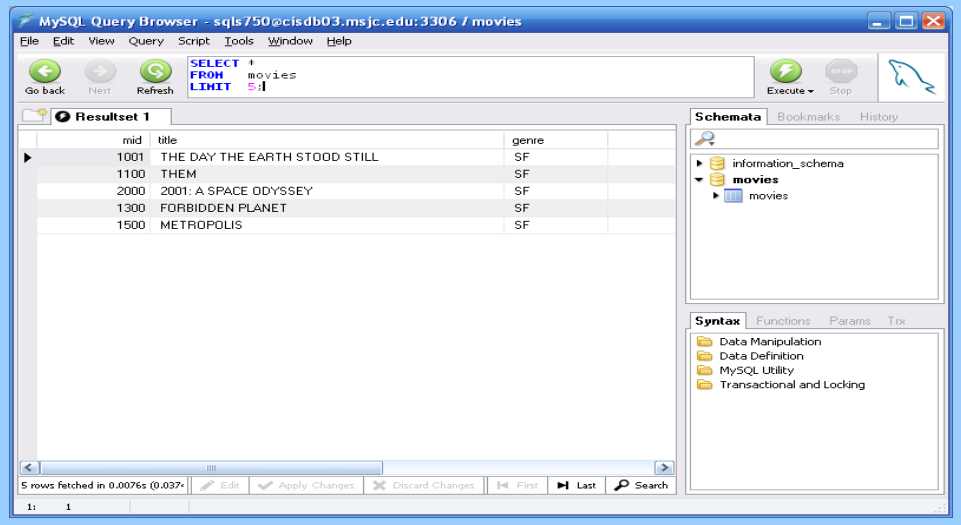
1. LIMIT number
2. LIMIT skip, number

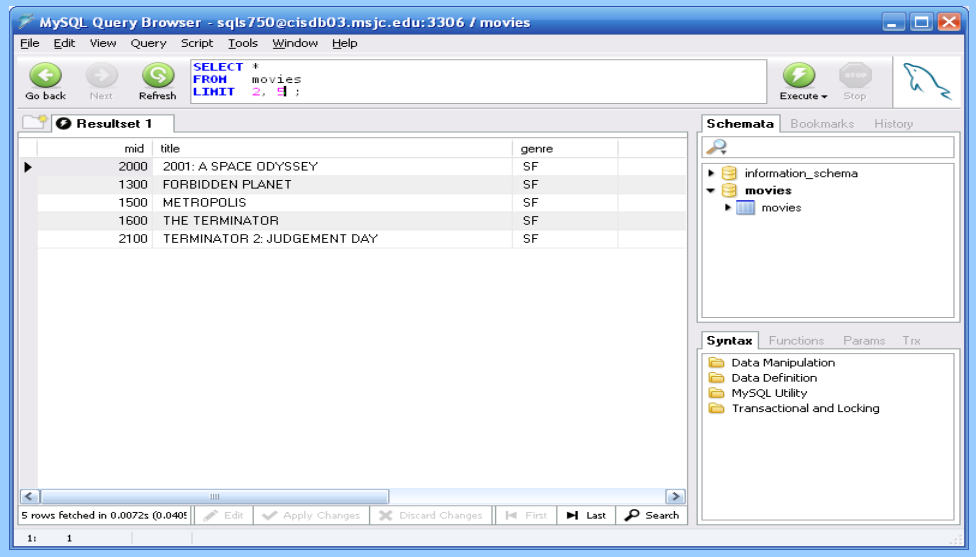
In this example, we show how the first form can be used to limit the total number of rows that will be displayed in the final result table.



Module 07: Sorting Rows

Page C-2: LIMIT skip, number

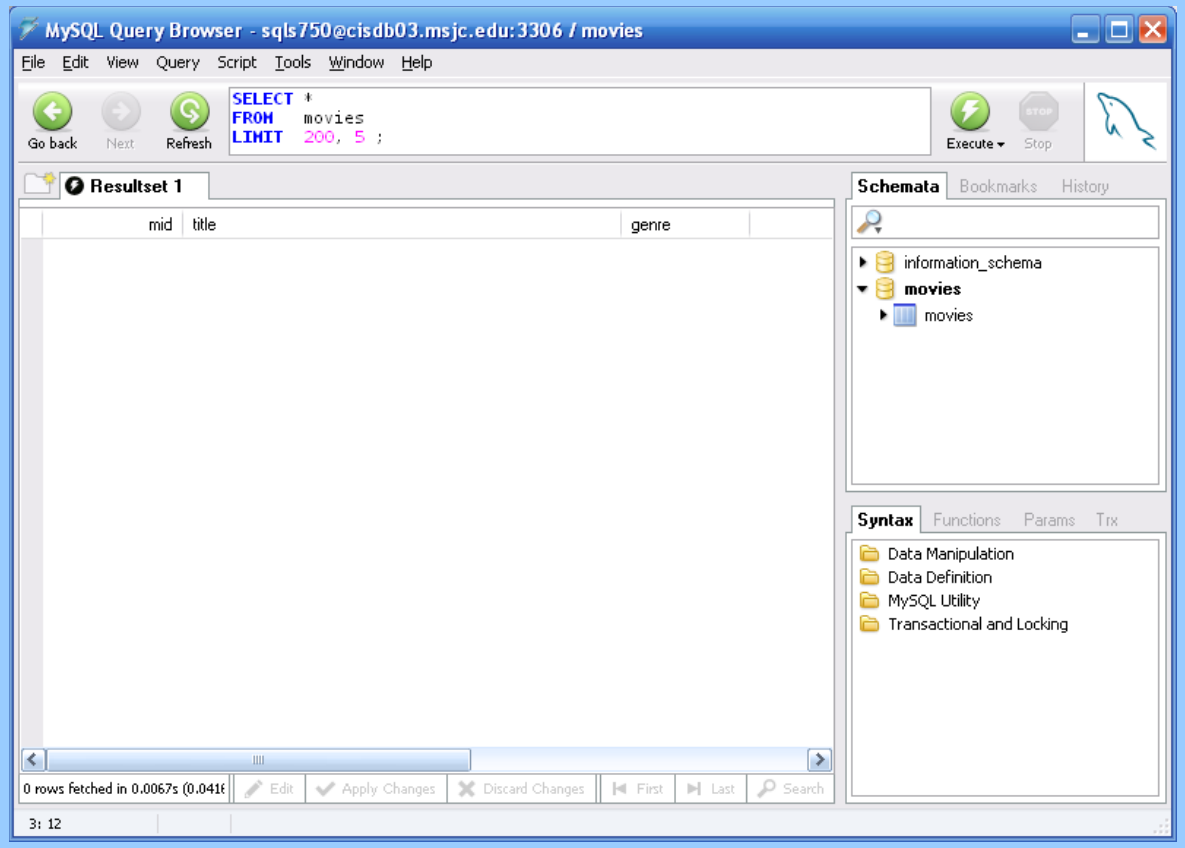




In this example, you can see how the second form can be used to display a number of lines from the result table, after having first skipped over some.

“After skipping 2 records, show the next 5”

If there are fewer lines in the result table than the number of lines you told it to skip over, the final result table will be empty. In essence, you skipped over all of the lines in the table.



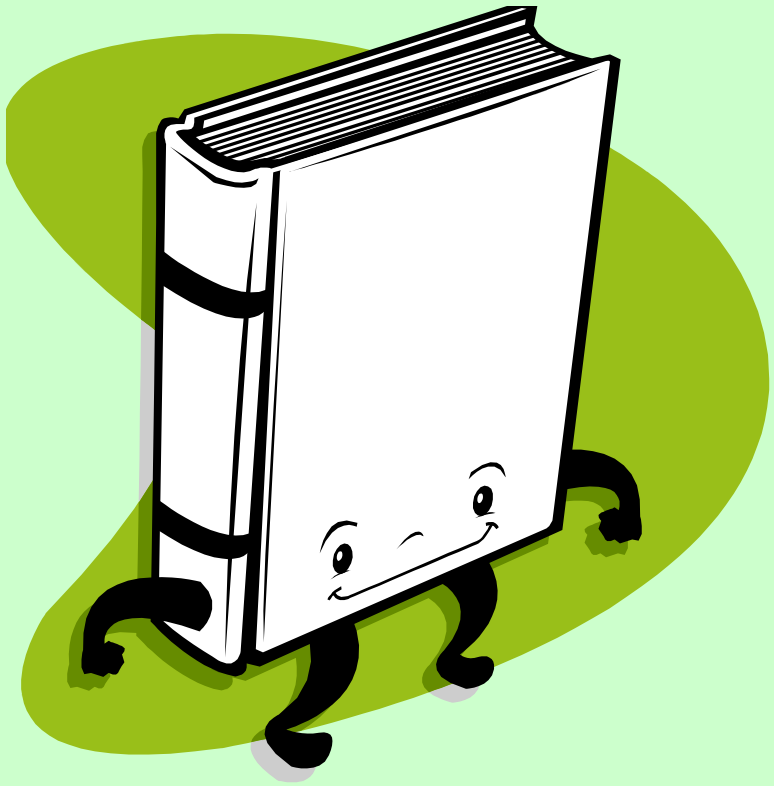
Page C-4: Limiting Output

The LIMIT clause is often used in conjunction with the ORDER BY clause, to find the highest value in a list, or the lowest value in a list.

And we should point out a performance issue here. The LIMIT clause does NOT lessen the workload on the database server in the amount of work it has to perform to satisfy your query.

The only workload that is lessened by the LIMIT clause is the amount of network traffic that would have been consumed had the entire result table been displayed.

Page T-1: Terminology



ORDER BY

Primary column, primary sort column
Secondary column, secondary sort column

ASC, DESC

LIMIT

Page Z-1: End Notes

Please drop me an email if you noticed any errors in this module. I'd also appreciate reading your comments, criticisms, and or suggestions as to how this module could be improved.

Thanks,

bil

That’s All