Basic SQL Programming

Page A-1: Intro

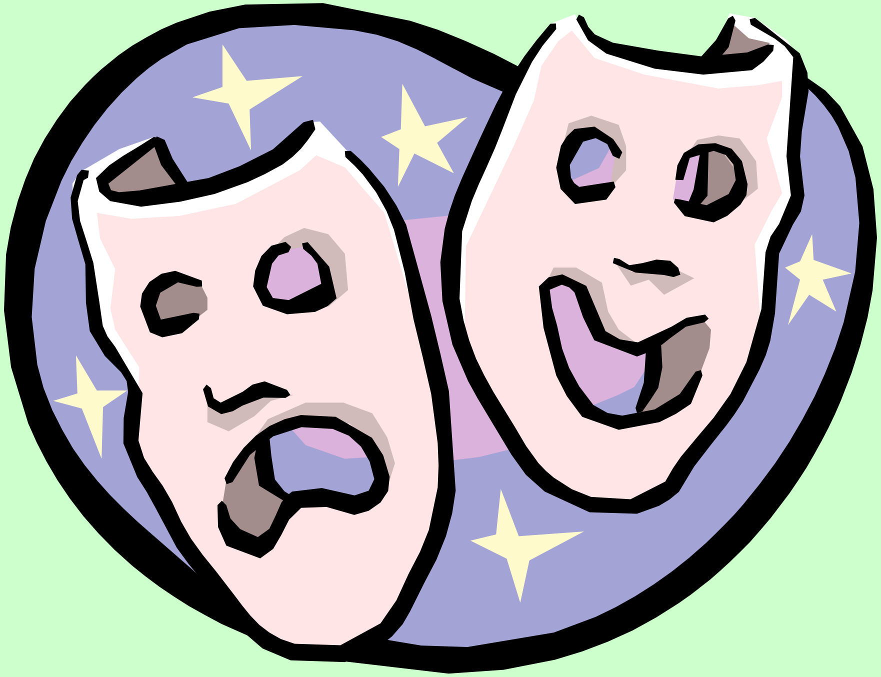
In the last module we learned that all computer programs are written in the context of an Information System application.

You also know that programming occurs during the later stages of the Systems Development Life Cycle (SDLC).

In this module we'll start writing SQL programs, and as the analyst responsible for these applications, I need to provide you, the programmer, with some documentation. Minimally you need:

1. A description of the application
2. A description of the system design
3. The program specifications

Page A-2: Talent Agency



Here is one of the scenarios that we'll be working on:

The Talons talent agency in Hollywood manages the careers of a number of actors and actresses.

They need a database to keep track of some of the personal and professional information about their clients.

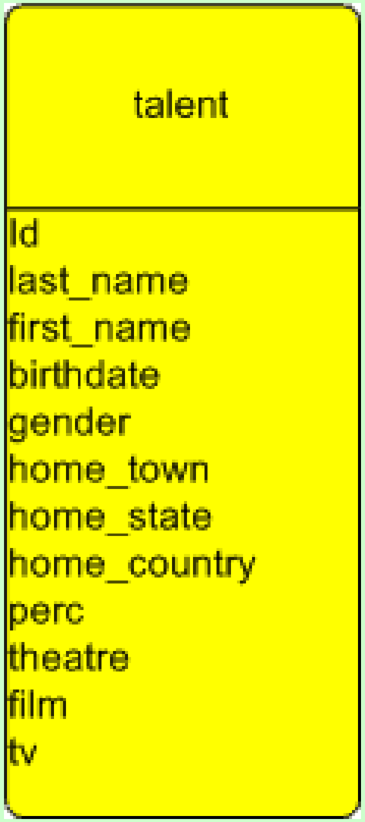
Among the personal bits of information that are important to the agency are:

birth_date,
home town,
home state,
home country.

Professional information includes:

stage name,
able/willing to work in theater,
able/willing to work in film,
able/willing to work in TV.

Page A-3: Talons System (cont)



A single-table database has been developed to track this information.

As you can see from the ER diagram, the table is named: talent, and it includes columns for each of the important fields of information that we had previously identified:

id
last_name
first_name
birthdate
gender
home_town
home_state
home_country
perc
theater
film
tv

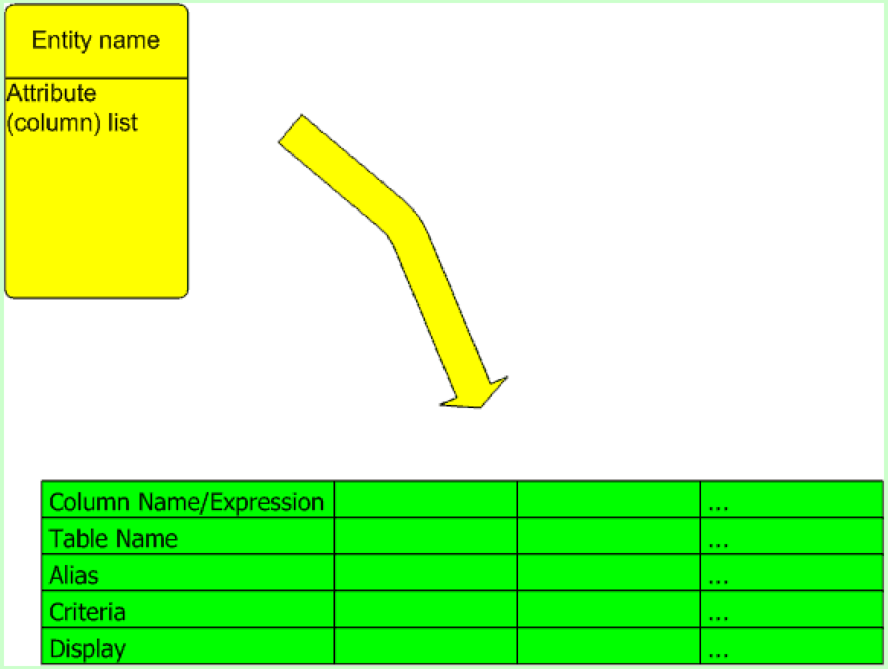
Page B-1: Problem 1

The user community has requested that the IT department prepare a listing showing the last names of all of the talent that our agency represents.

Your manager suggests that SQL might be the most appropriate tool to use for this request, and she turns the project over to you.

Page B-2: Problem Methodology

Since you don’t have much experience with SQL programming you decide to use a Table Build Chart (TBC) to help you model the requirements of the program.



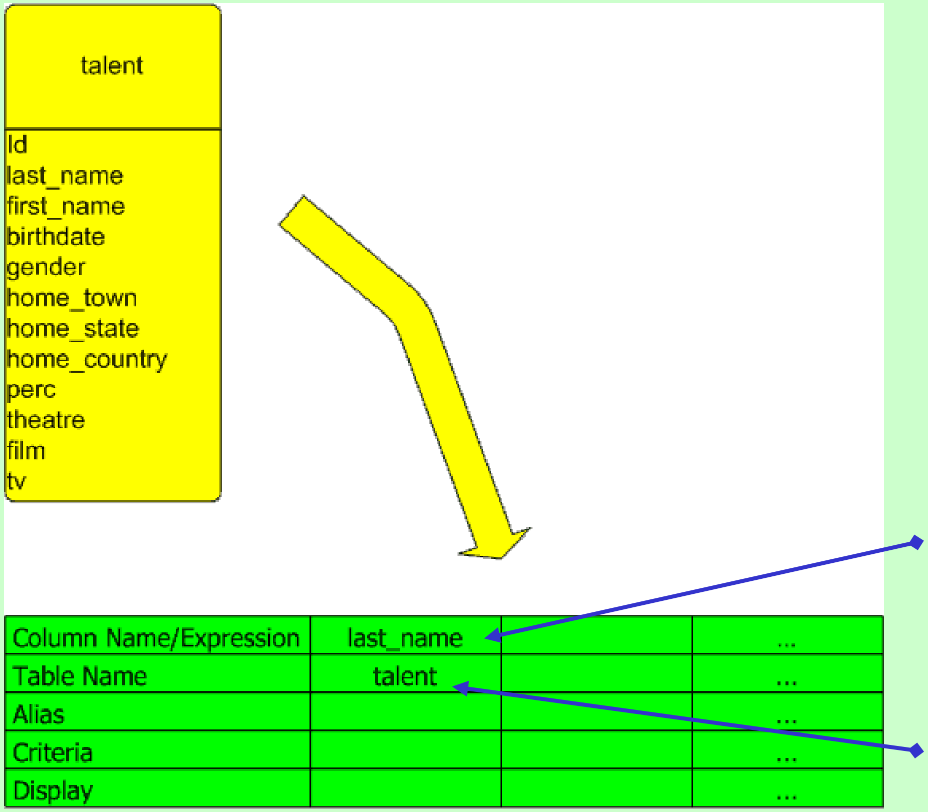
Column Name/Expression			…
Table Name			…
Alias			…
Criteria			…
Display			…

Here is one suggested design/development methodology:

1. Draw an ER diagram that includes all of the tables that will be used in the solution (yellow).
2. Draw a blank Table Build Chart (TBC) (green)
3. Fill in the TBC according to the program specifications
4. Code your solution

Page B-3 Problem 1- TBC Model

Let’s revisit the problem, and see how the Table Build Chart can help us map out our solution.



Column Name/Expression	last_name		...
Table Name	talent		...
Alias			...
Criteria			...
Display			...

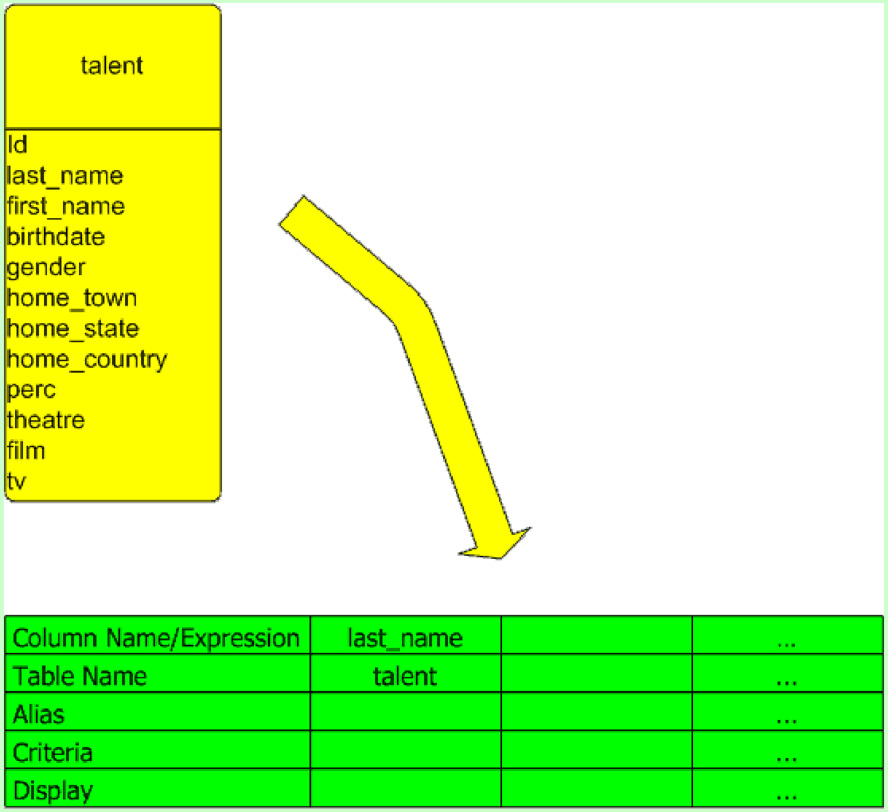
The user community has requested that the IT department prepare a listing showing the last names of all of the talent that our agency represents.

The Table Build Chart helps us organize our thoughts, better understand the problem, and get a ‘leg up’ on the programming by simply filling in some slots.

What columns do we need? The users only asked for one column’s worth of information -> the last name. So let’s add that entry to the TBC.

In our database application, which table is it that has this ‘last name’ information in it? Eh, that’d be the ‘talent’ table, so we can now record that tidbit of information too.

Page B-4: Problem 1 - Coding



talent

Id
last_name
first_name
birthdate
gender
home_town
home_state
home_country
perc
theatre
film
tv

Column Name/Expression	last_name		...
Table Name	talent		...
Alias			...
Criteria			...
Display			...

Now we transform the TBC into code.

1. Build the SELECT clause from the column name/expression entries:

```
SELECT  last_name
```

2. Build the FROM clause from the table name entries:

```
FROM     talent
```

3. Put it all together:

```
SELECT  last_name
FROM     talent
```

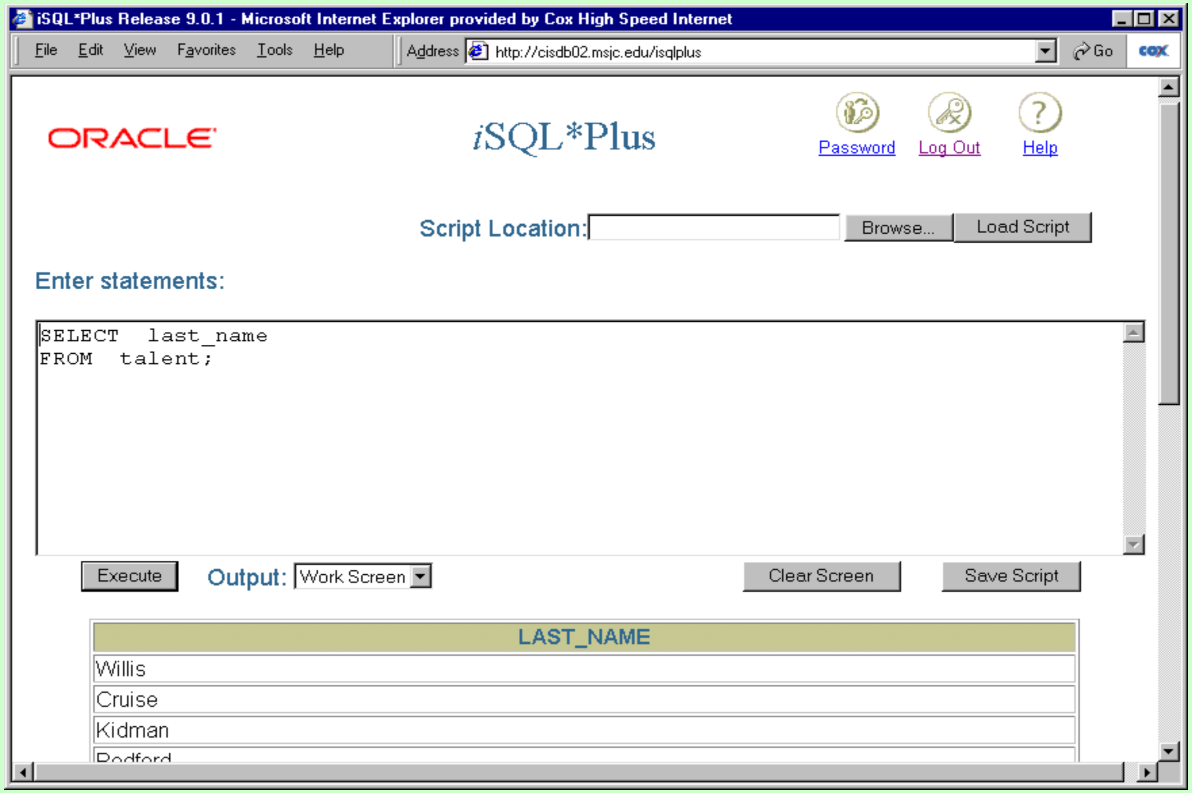
4. Enter the code in iSQL-Plus, and execute the program.

Page B-5: Problem 1 – (cont)

Type your SQL program in the box labeled: Enter Statements.

Click on the Execute button

Look for the output to be displayed below the input area.

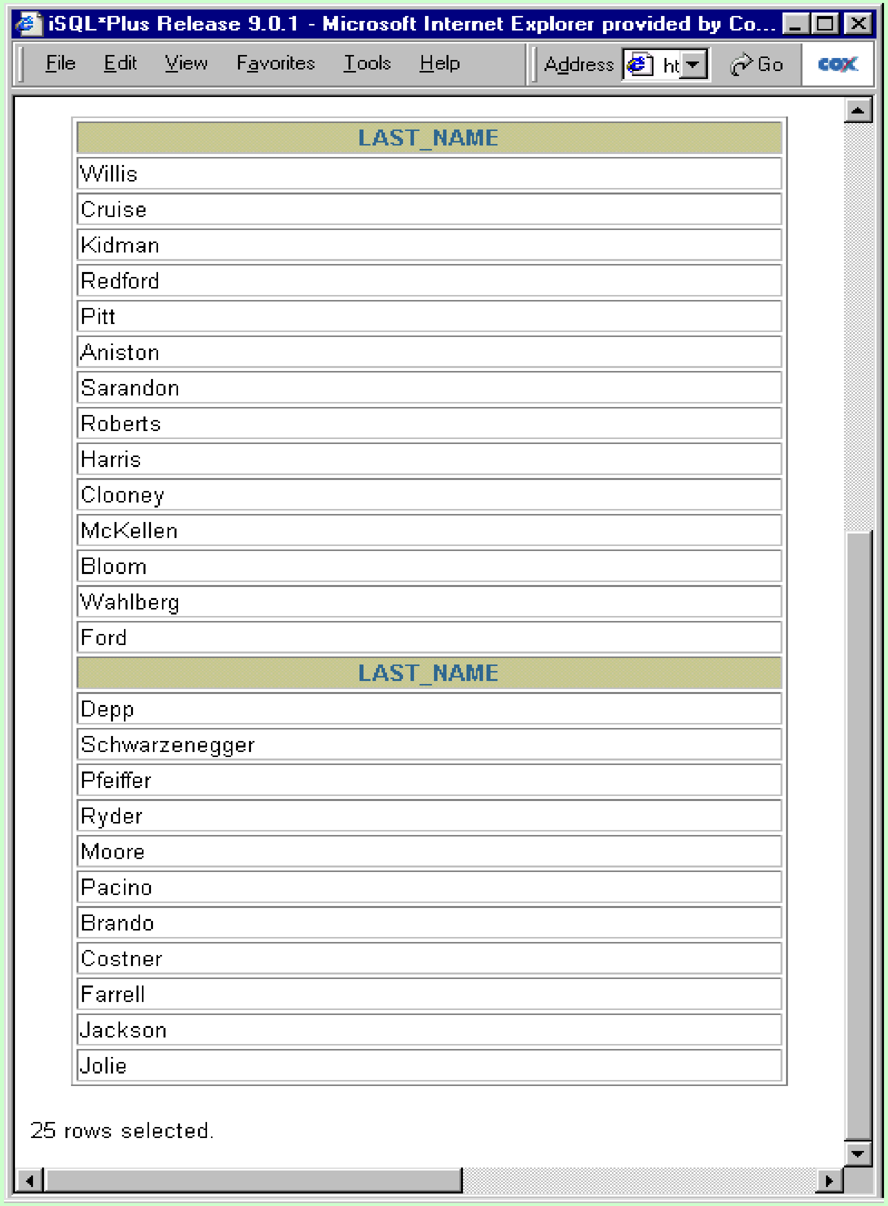


```
SELECT  last_name
FROM  talent;
```

```
LAST_NAME
Willis
Cruise
Kidman
Redford
```

Module 02: Basic SQL

Page B-6: Problem 1 (cont)


iSQL*Plus Release 9.0.1 - Microsoft Internet Explorer provided by Co...
File
Edit
View
Favorites
Tools
Help
Address
ht
Go
cox

LAST_NAME
Willis
Cruise
Kidman
Redford
Pitt
Aniston
Sarandon
Roberts
Harris
Clooney
McKellen
Bloom
Wahlberg
Ford

LAST_NAME
Depp
Schwarzenegger
Pfeiffer
Ryder
Moore
Pacino
Brando
Costner
Farrell
Jackson
Jolie

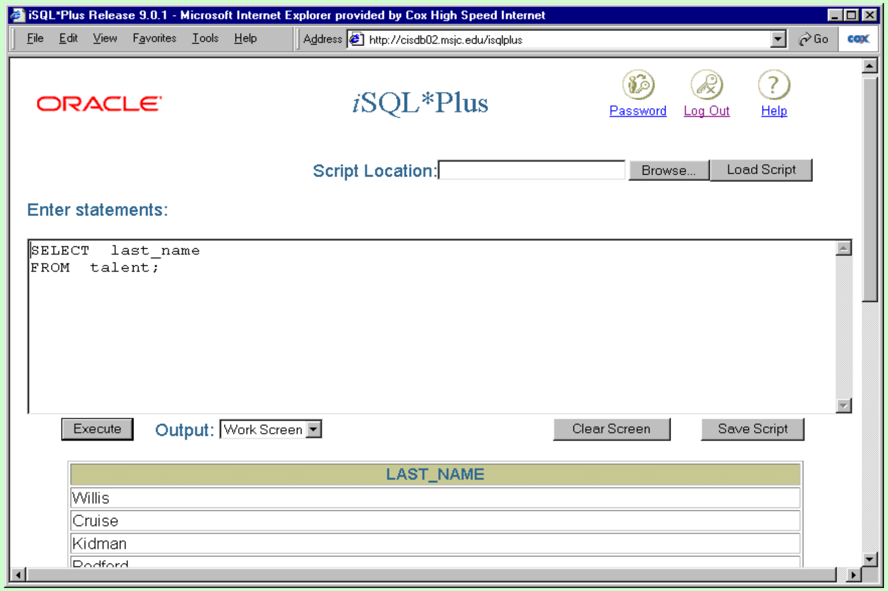
```
25 rows selected.
```

There are more lines displayed than fit on the screen, so you may need to use the scroll bar to see all of the output that was generated by this SQL program.

You should notice that for long outputs (more than a handful of lines), SQL will also indicate how many rows of data were included in the result table (in this example 25 rows were selected).

Module 02: Basic SQL

Page B-7: Problem 1 Analysis



```
SELECT  last_name
FROM  talent;
```

LAST_NAME
Willis
Cruise
Kidman
Redford

This SQL program is rather simple.

In the SELECT clause we list all of the columns that we want to have displayed.

The FROM clause tells SQL which table to use to find the information.

The SELECT clause identifies what columns

The FROM clause identifies which table(s)

Page B-8: Learning Strategy

Remember that it’s a good idea to type these programs in as you encounter them. The practice will help you learn the material more quickly.

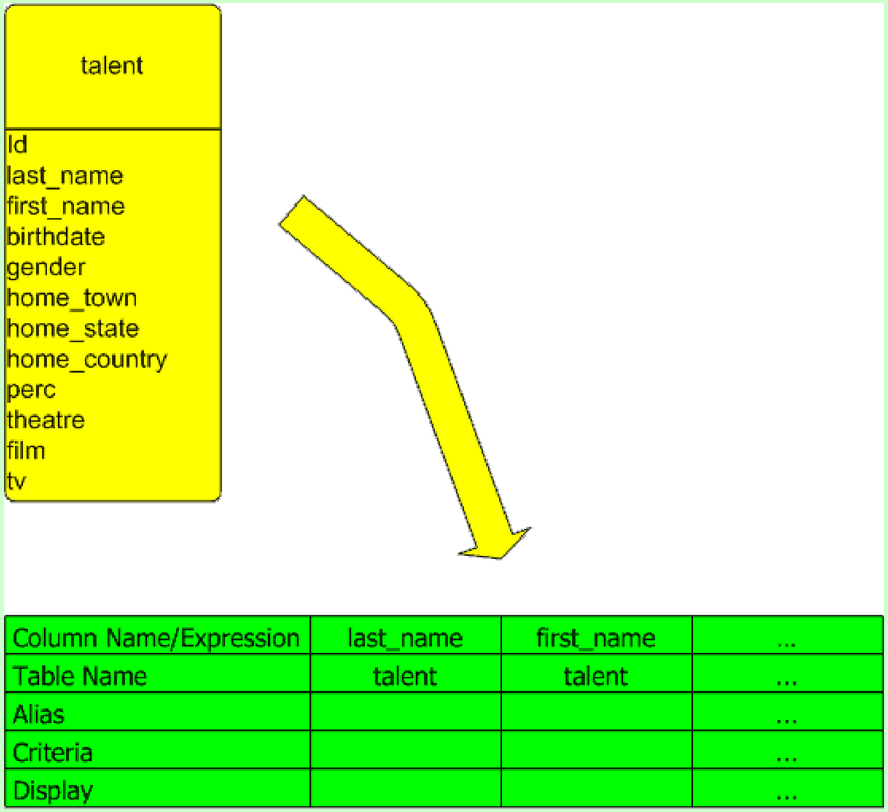
Page B-9: Problem 2

The user community has requested that the IT department prepare a listing showing both the last and first names of all of the talent that our agency represents.

Again, your manager suggests that SQL might be the most appropriate tool to use for this request, and she turns the project over to you.

Problem 2- TBC Model

Sketch out a quick solution for the problem using a TBC.



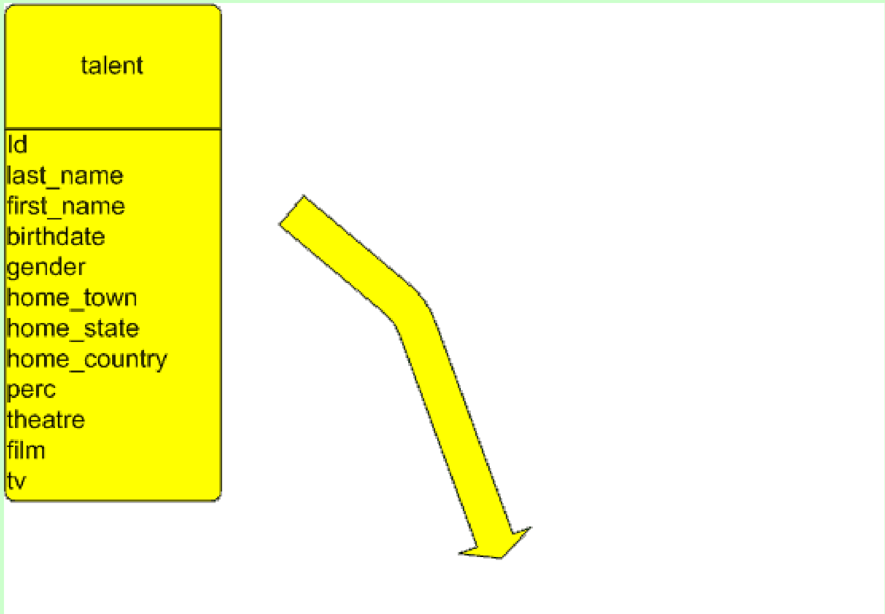
talent

Id
last_name
first_name
birthdate
gender
home_town
home_state
home_country
perc
theatre
film
tv

Column Name/Expression	last_name	first_name	...
Table Name	talent	talent	...
Alias			...
Criteria			...
Display			...

Module 02: Basic SQL

Page B-11: Problem 2 - Coding



talent

Id
last_name
first_name
birthdate
gender
home_town
home_state
home_country
perc
theatre
film
tv

Column Name/Expression	last_name	first_name	...
Table Name	talent	talent	...
Alias			...
Criteria			...
Display			...

Transform the TBC into code.

```
SELECT  last_name, first_name
FROM     talent;
```

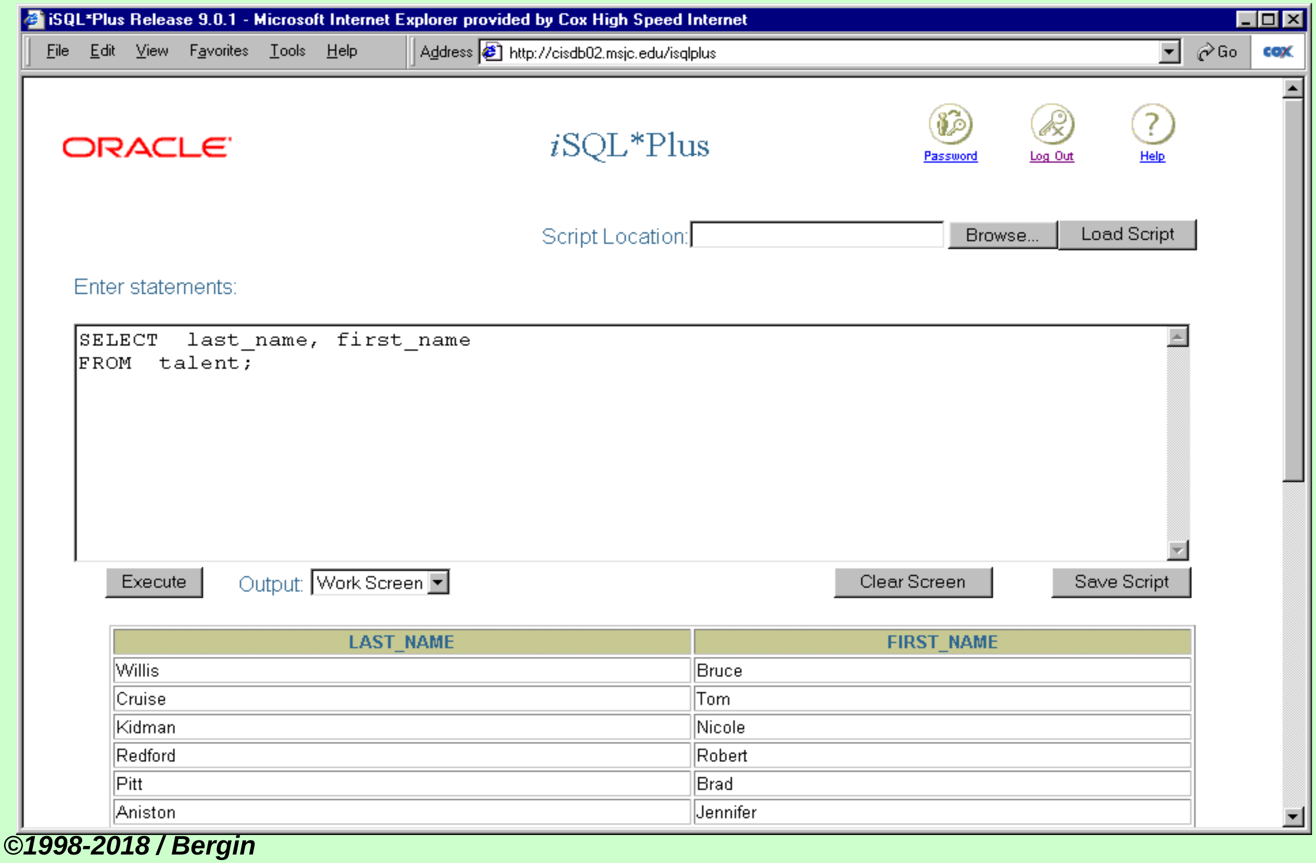
Enter the code in iSQL-Plus, and execute the program.

Page B-12: Problem 2 – (cont)

Type the SQL program in the box labeled: Enter Statements.

Click on the Execute button.

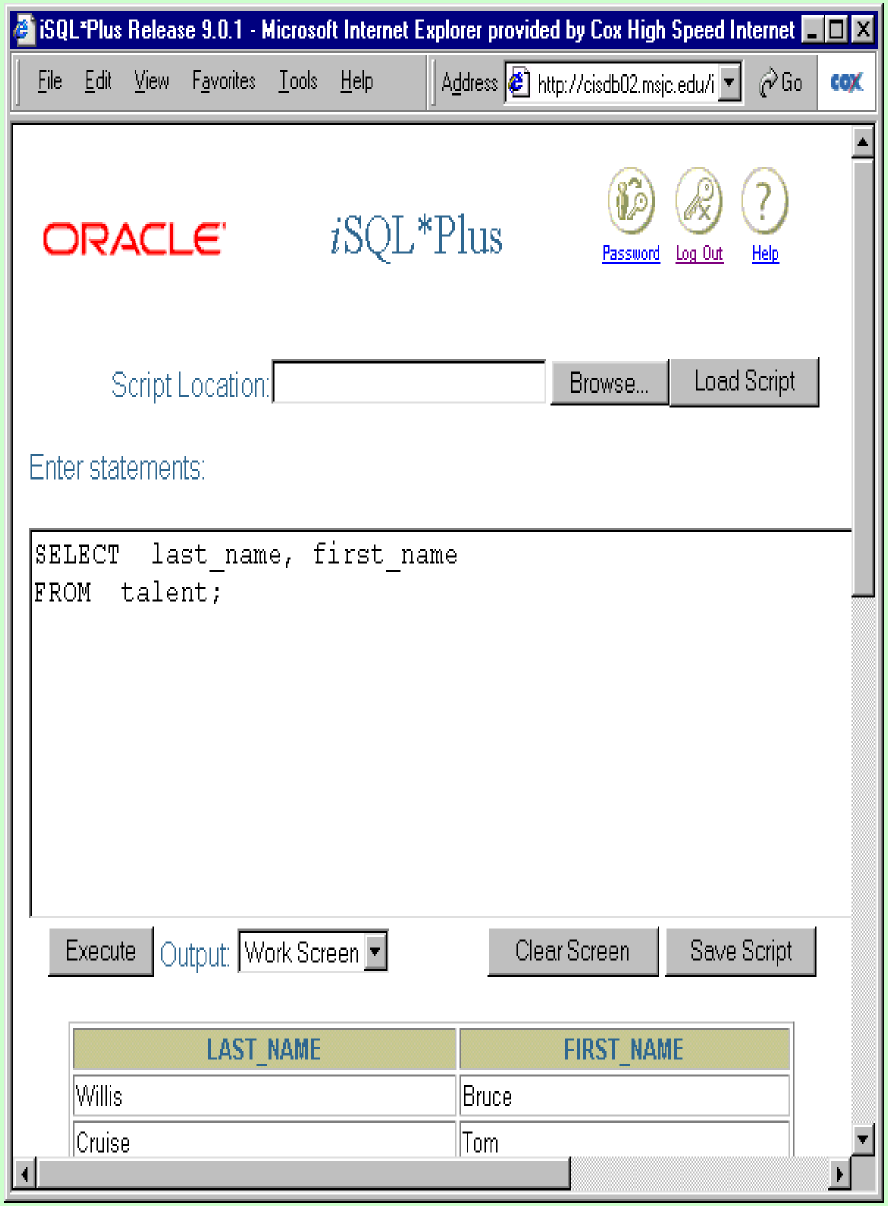
Look for the output to be displayed below the input area.


iSQL*Plus Release 9.0.1 - Microsoft Internet Explorer provided by Cox High Speed Internet
File Edit View Favorites Tools Help
Address http://cisdb02.msjc.edu/isqlplus
Go
ORACLE
iSQL*Plus
Password
Log Out
Help
Script Location:
Browse...
Load Script
Enter statements:
Execute
Output: Work Screen
Clear Screen
Save Script

```
SELECT  last_name, first_name
FROM  talent;
```

LAST_NAME	FIRST_NAME
Willis	Bruce
Cruise	Tom
Kidman	Nicole
Redford	Robert
Pitt	Brad
Aniston	Jennifer

Page B-13: Problem 2 Analysis


iSQL*Plus Release 9.0.1 - Microsoft Internet Explorer provided by Cox High Speed Internet
File Edit View Favorites Tools Help
Address http://cisdb02.msjc.edu/i
Go
ORACLE
iSQL*Plus
Password
Log Out
Help
Script Location:
Browse...
Load Script
Enter statements:
Execute
Output: Work Screen
Clear Screen
Save Script

```
SELECT  last_name, first_name
FROM  talent;
```

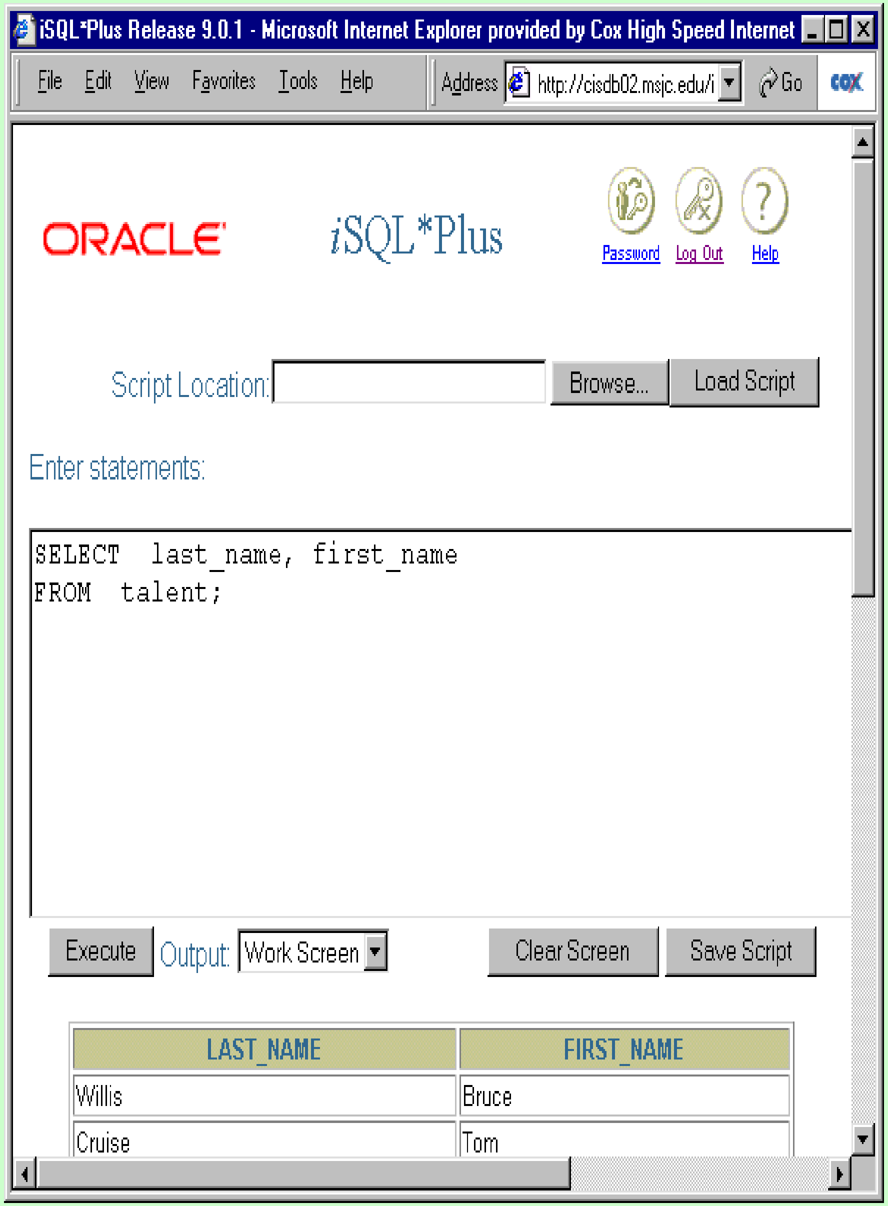
LAST_NAME	FIRST_NAME
Willis	Bruce
Cruise	Tom

Another column's worth of data requires another column specification in the SQL clause.

Column specifications are separated from one another by commas.

Columns are displayed, left to right in the result table, in the order in which they are listed in the SELECT clause.

Page B-14: Design vs Analysis


iSQL*Plus Release 9.0.1 - Microsoft Internet Explorer provided by Cox High Speed Internet
File Edit View Favorites Tools Help
Address http://cisdb02.msjc.edu/i
Go
ORACLE
iSQL*Plus
Password
Log Out
Help
Script Location:
Browse...
Load Script
Enter statements:
Execute
Output: Work Screen
Clear Screen
Save Script

```
SELECT  last_name, first_name
FROM  talent;
```

LAST_NAME	FIRST_NAME
Willis	Bruce
Cruise	Tom

In how many different ways might we have satisfied this user request?

One alternative might be to list first names first. And after we learn about character functions we'll have a number of tools and techniques that we can use to 'dress up' our results. So, we can add another dozen or so possibly correct solutions if we want to consider those alternatives. *BUT* each of these alternatives would have been derived from their own unique TBC.

When we design a computer program, we're not interested in designing all of the possible solutions. We just need one solution (that works), and as part of our design effort we choose just that one solution and focus our energies on getting that one design to work.

Generally speaking, for each TBC, there is only one solution.

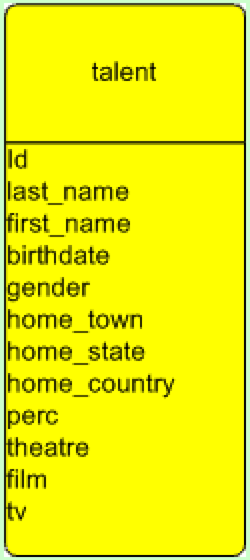
Page B-15: Definitions

A base table is a table that exists in the database. It is designed to store data.

A result table is a table that exists on screen, usually only for the duration of your SQL program.

SELECT statements only retrieve (R) data, they do not change (CrUD) any of the information in a base table.

Page B-16: Problem 3



talent

Id
last_name
first_name
birthdate
gender
home_town
home_state
home_country
perc
theatre
film
tv

They’re back! We’ve got another request from our users.

Now they need a listing showing all of the information that’s on file for the talent that our agency represents.

This is going to be one ‘hairy’ SELECT clause – each column will have to be listed:

```
SELECT  id, last_name, first_name,
birthdate, gender, home_town, home_state,
home_country, perc, theatre, film, tv
```

There ought to be a shortcut!

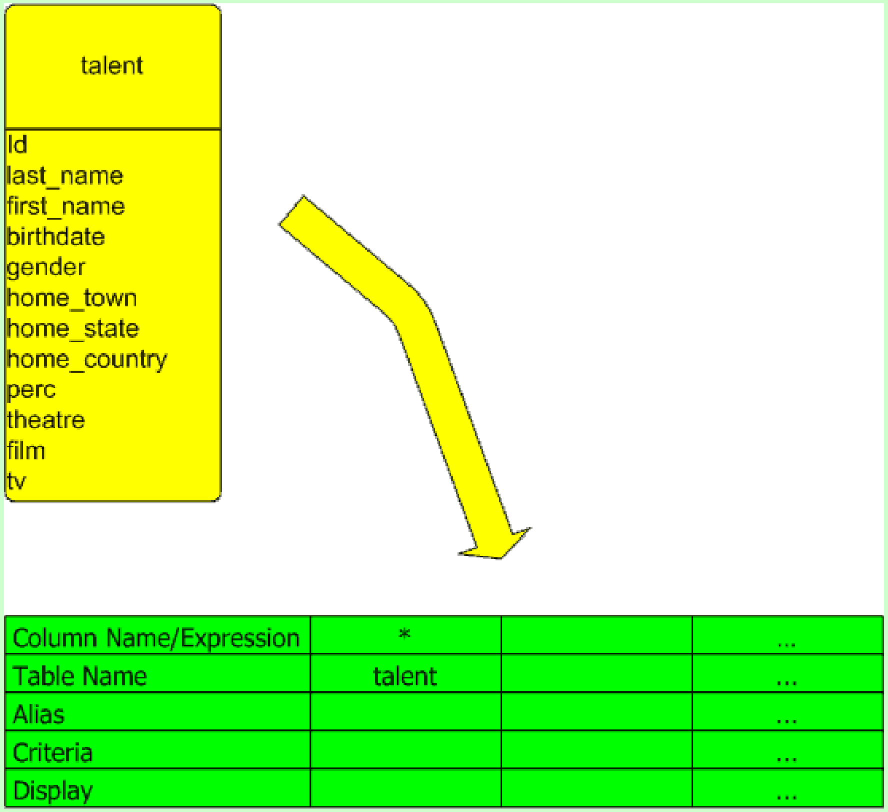
And there is. ☺

We can use the wildcard character “*” to indicate ‘*every column*’.

Page B-17 Problem 3- TBC Model

Sketch out a quick solution for the problem using a TBC.

Notice how we used the asterisk in the TBC.

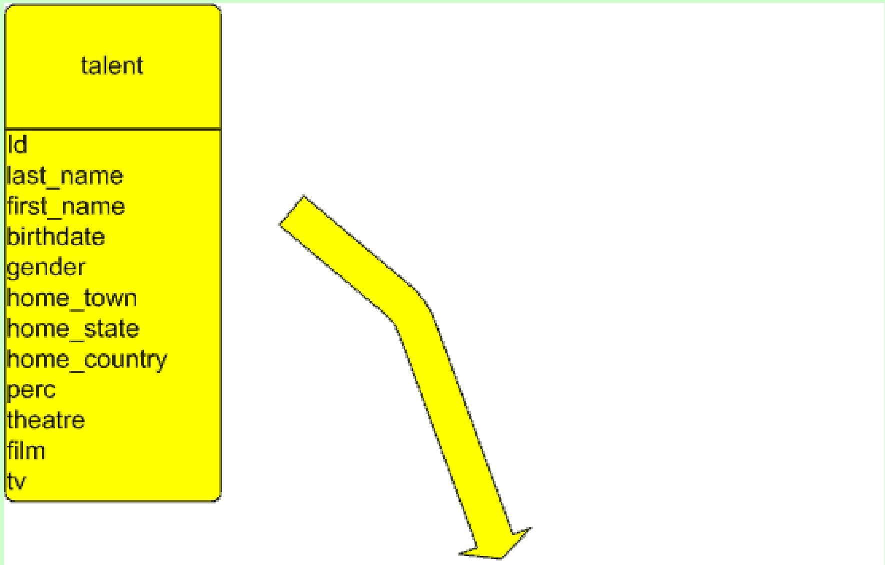


talent

Id
last_name
first_name
birthdate
gender
home_town
home_state
home_country
perc
theatre
film
tv

Column Name/Expression	*		...
Table Name	talent		...
Alias			...
Criteria			...
Display			...

Page B-18: Problem 3 - Coding



Column Name/Expression	*		...
Table Name	talent		...
Alias			...
Criteria			...
Display			...

Now, transform the TBC into code.

```
SELECT   *
FROM     talent;
```

Enter the code in iSQL-Plus, and execute your program.

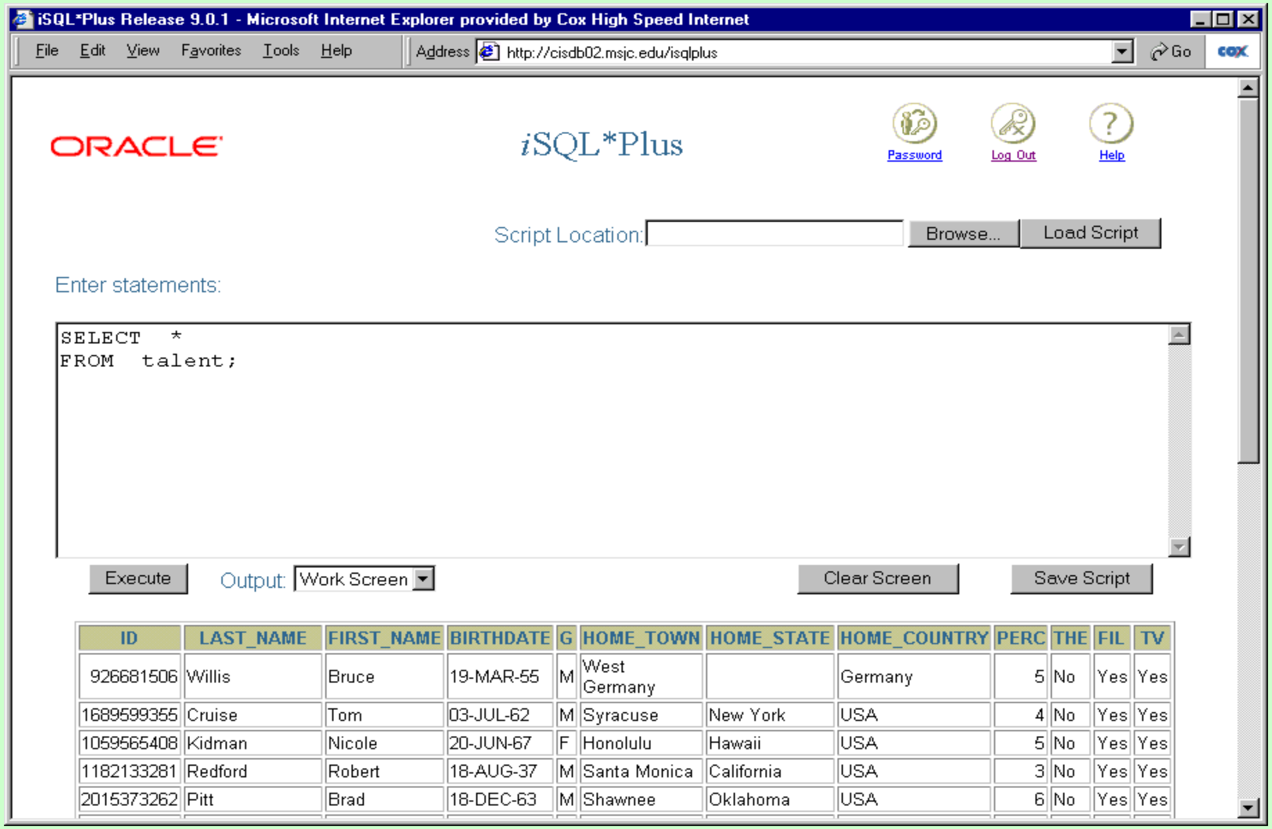
Module 02: Basic SQL

Page B-19: Problem 3 – (cont)

Type your SQL program in the box labeled: Enter Statements.

Click on the Execute button.

Look for your output to be displayed below the input area.



Address: http://cisdb02.msjc.edu/isqlplus

Script Location:

Enter statements:

```
SELECT  *
FROM  talent;
```

Output: Work Screen

ID	LAST_NAME	FIRST_NAME	BIRTHDATE	G	HOME_TOWN	HOME_STATE	HOME_COUNTRY	PERC	THE	FIL	TV
926681506	Willis	Bruce	19-MAR-55	M	West Germany		Germany	5	No	Yes	Yes
1689599355	Cruise	Tom	03-JUL-62	M	Syracuse	New York	USA	4	No	Yes	Yes
1059565408	Kidman	Nicole	20-JUN-67	F	Honolulu	Hawaii	USA	5	No	Yes	Yes
1182133281	Redford	Robert	18-AUG-37	M	Santa Monica	California	USA	3	No	Yes	Yes
2015373262	Pitt	Brad	18-DEC-63	M	Shawnee	Oklahoma	USA	6	No	Yes	Yes

Page B-20: Problem 3 Analysis

When you include the * option in the SELECT clause, you may not specify any other columns.

In this regard, the * stands alone.

Page C-1: Metadata

The asterisk is a pretty handy SQL feature, but how does it work?

You can think of it this way:

First the database references its metadata in order to determine what columns belong in the table. Then it replaces the * with the names of all of these columns prior to executing your program.

This is an excellent example of the self-describing nature of the database and how it contains not only the data that is pertinent to the business application, but *metadata* as well, so that it can manage its own information structures.

Page C-2: Ordering

Now here's a question for you…

What determines how the columns are ordered in the result table when you use the * wildcard?

If you ran that SQL program 100 times, the result table would always be the same, ie. the columns would always be listed in the same order.

This is because most relational database management systems present the columns in the same order as they were listed when the table was originally created. This consistent ordering of columns is not a requirement of RDBMS, in fact, one of the underlying principles of RDBMS is that the programmer can never rely on columns being stored in any particular order, nor rows being stored in any particular order.

This consistent ordering is a feature ☺

Page C-3: Metadata Commands

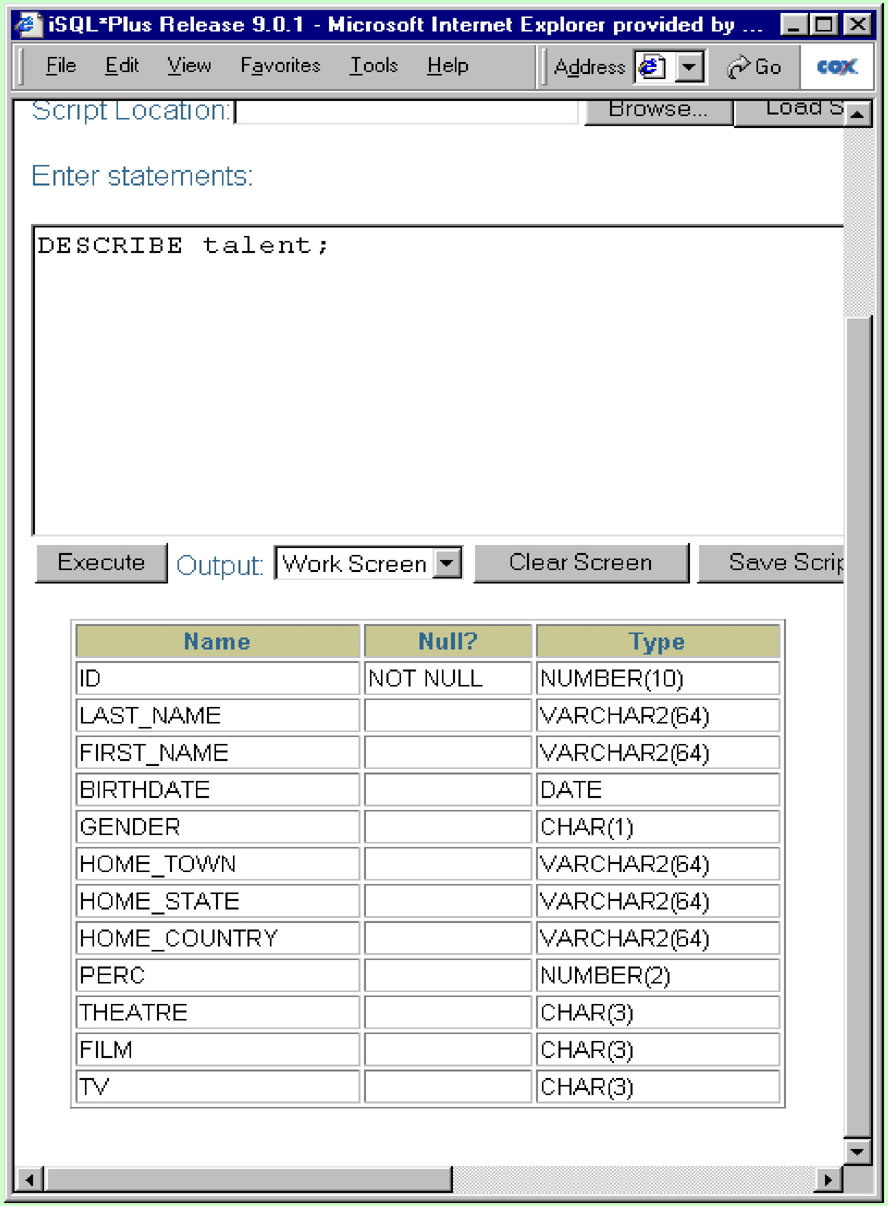
The Oracle database includes a handy metadata command: DESCRIBE.

This command is not part of SQL, rather it is an extension to SQL provided by Oracle. If you use another database system (SQL Server, SYBASE, MySQL…), it is possible that DESCRIBE will not work.

One format of the DECRIBE command is to specify a table name immediately after the DESCRIBE keyword. Eg.

```
DESCRIBE  talent
```

Page C-4: Example



```
DESCRIBE talent;
```

Name	Null?	Type
ID	NOT NULL	NUMBER(10)
LAST_NAME		VARCHAR2(64)
FIRST_NAME		VARCHAR2(64)
BIRTHDATE		DATE
GENDER		CHAR(1)
HOME_TOWN		VARCHAR2(64)
HOME_STATE		VARCHAR2(64)
HOME_COUNTRY		VARCHAR2(64)
PERC		NUMBER(2)
THEATRE		CHAR(3)
FILM		CHAR(3)
TV		CHAR(3)

The Oracle database includes a handy metadata command: DESCRIBE.

This command is not part of SQL, rather it is an extension to SQL provided by Oracle. If you use another database system (SQL Server, SYBASE, …), it is unlikely that DESCRIBE will work.

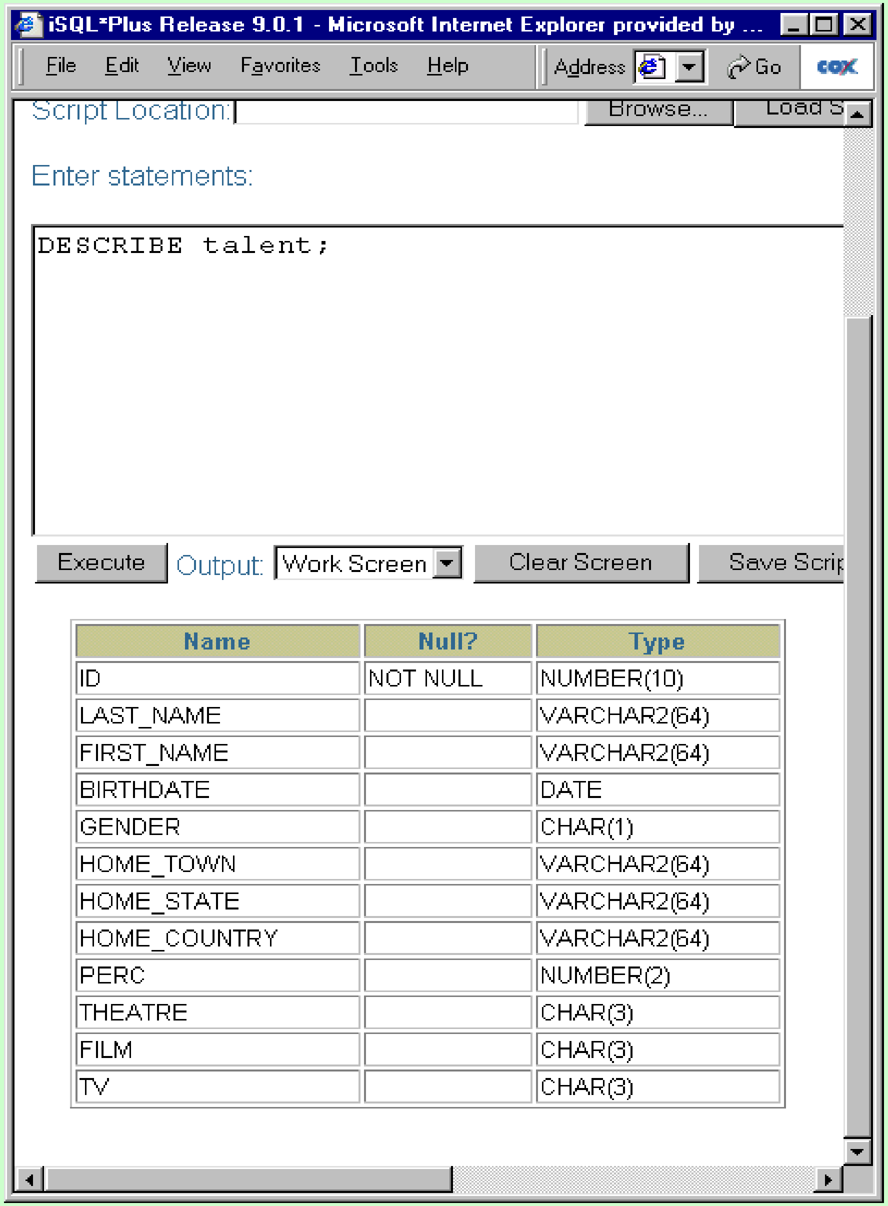
One format of the DECRIBE command is to specify a table name immediately after the DESCRIBE keyword. Eg.

DESCRIBE talent

The result table will include three columns:

1. Column name,
2. Whether that column may or may not accept null values
3. The datatype (and size) of the column

Page D-1: Stylistic Conventions



```
DESCRIBE talent;
```

Output: Work Screen

Name	Null?	Type
ID	NOT NULL	NUMBER(10)
LAST_NAME		VARCHAR2(64)
FIRST_NAME		VARCHAR2(64)
BIRTHDATE		DATE
GENDER		CHAR(1)
HOME_TOWN		VARCHAR2(64)
HOME_STATE		VARCHAR2(64)
HOME_COUNTRY		VARCHAR2(64)
PERC		NUMBER(2)
THEATRE		CHAR(3)
FILM		CHAR(3)
TV		CHAR(3)

Programmers generally follow stylistic conventions when they code their programs.

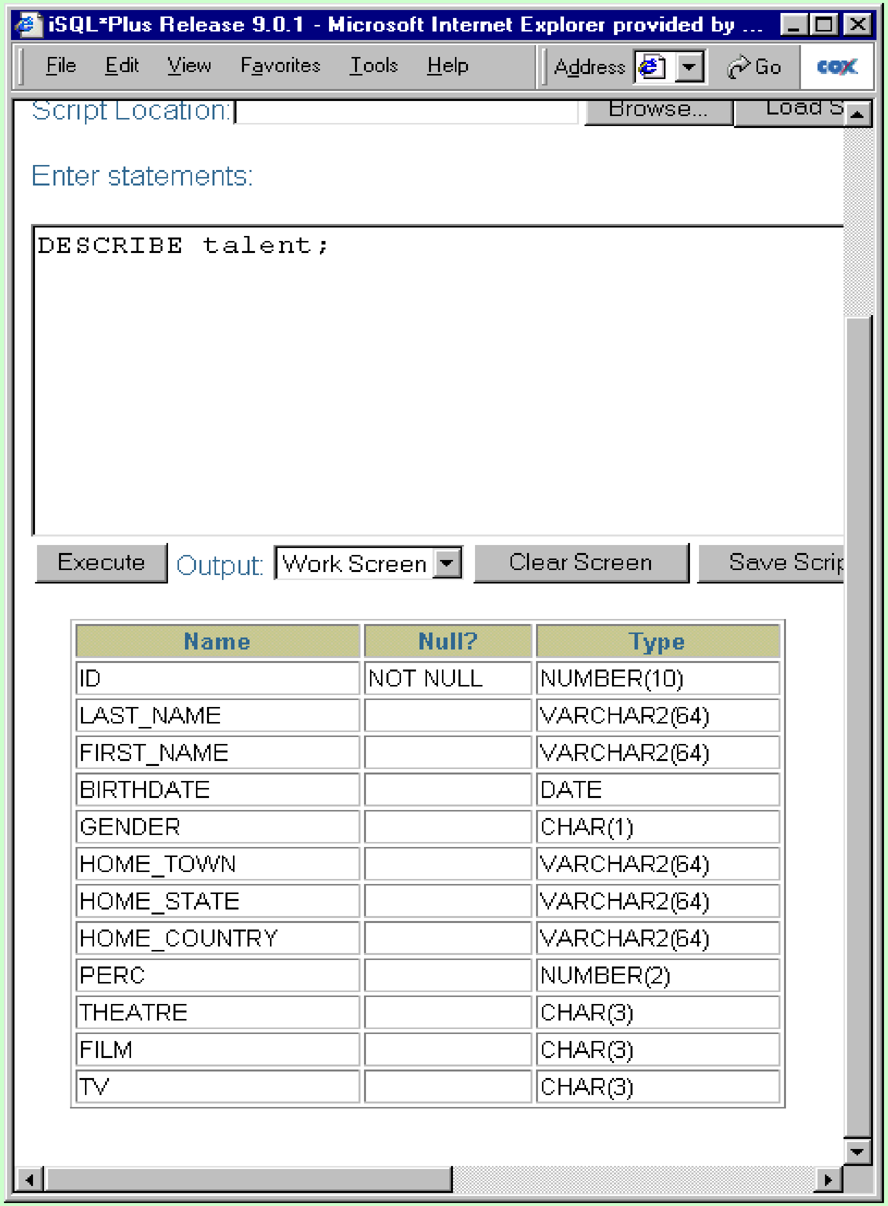
Stylistic conventions dictate how the program will be coded for the purpose of making it easier for humans to understand the program.

Stylistic conventions do not influence the computer in how it interprets the program. Stylistic conventions ***cannot*** change the behavior of a program.

One convention that I follow suggests that every clause of a SQL program be written on a separate line.

If you go back and review the sample SQL programs that I've shared with you, you'll notice that the SELECT clause is typed on its own line, and so is the FROM clause.

Page D-2: SC cont



```
DESCRIBE talent;
```

Name	Null?	Type
ID	NOT NULL	NUMBER(10)
LAST_NAME		VARCHAR2(64)
FIRST_NAME		VARCHAR2(64)
BIRTHDATE		DATE
GENDER		CHAR(1)
HOME_TOWN		VARCHAR2(64)
HOME_STATE		VARCHAR2(64)
HOME_COUNTRY		VARCHAR2(64)
PERC		NUMBER(2)
THEATRE		CHAR(3)
FILM		CHAR(3)
TV		CHAR(3)

I also use uppercase characters for the reserved words in SQL, and I use lowercase characters for the identifiers.

Identifiers are items such as:

Table names

Column names

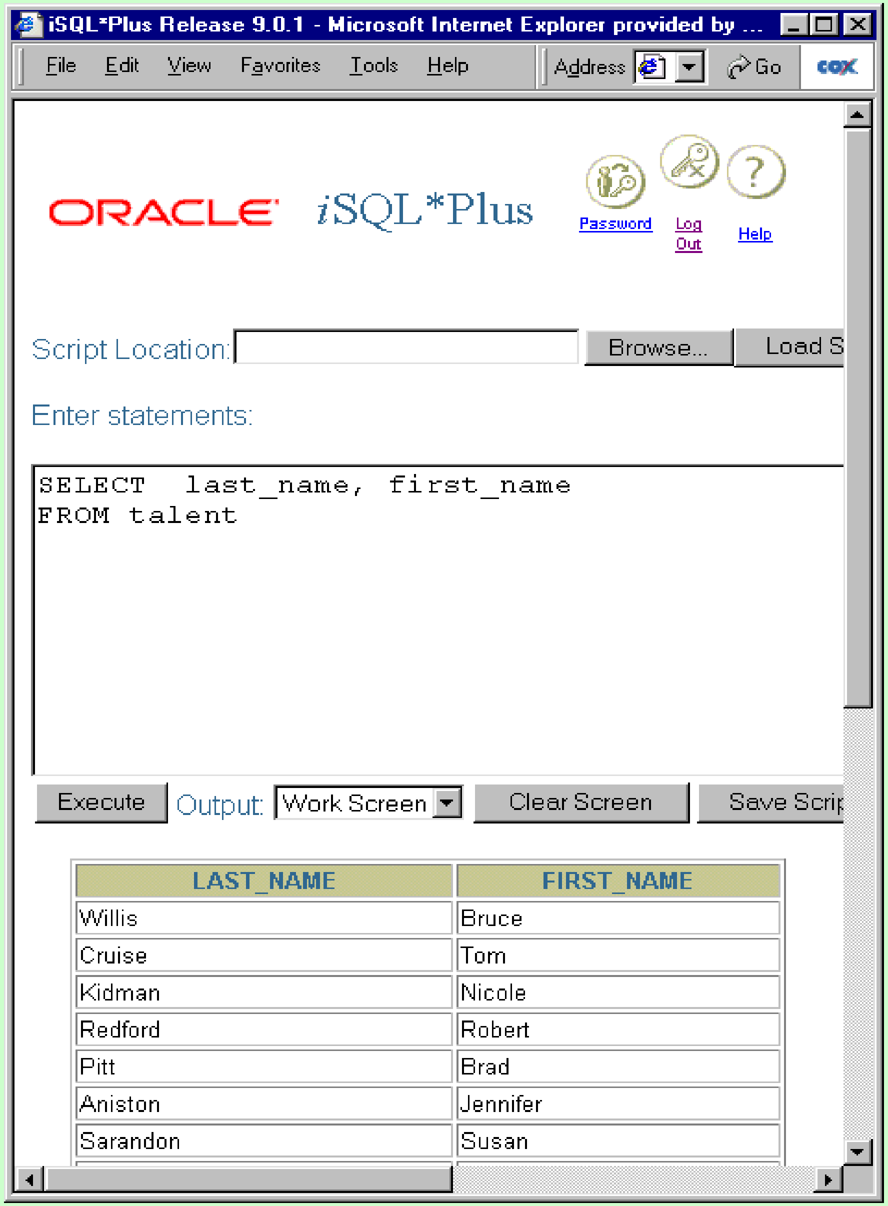
Remember that English trick I mentioned regarding new jargon? You can try it here.

Identifiers identify things ☺

Identifiers are the names (or labels) that you assign to the tables and columns that you use in SQL.

Module 02: Basic SQL

Page D-3: SC cont

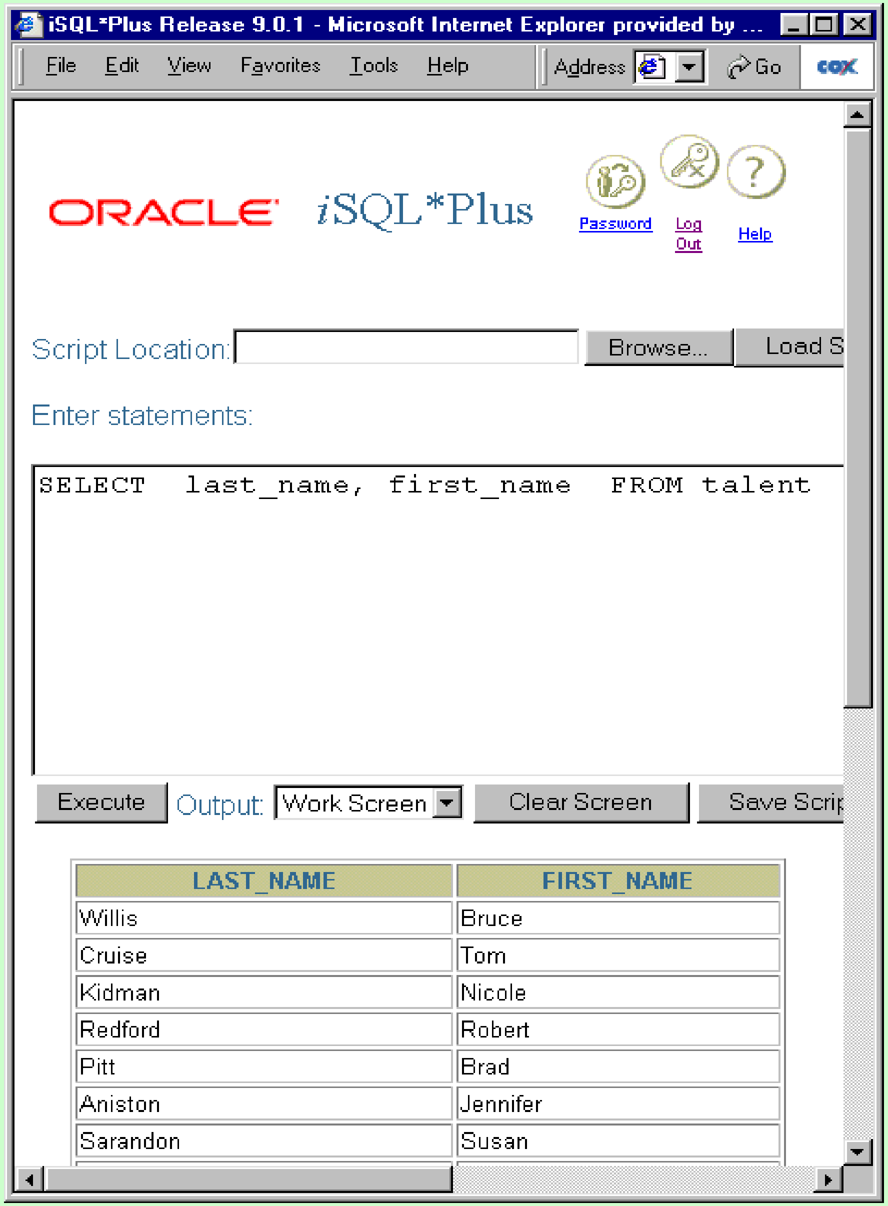


```
SELECT  last_name, first_name
FROM talent
```

LAST_NAME	FIRST_NAME
Willis	Bruce
Cruise	Tom
Kidman	Nicole
Redford	Robert
Pitt	Brad
Aniston	Jennifer
Sarandon	Susan

Each of the following slides demonstrates a BAD variation on our coding convention.

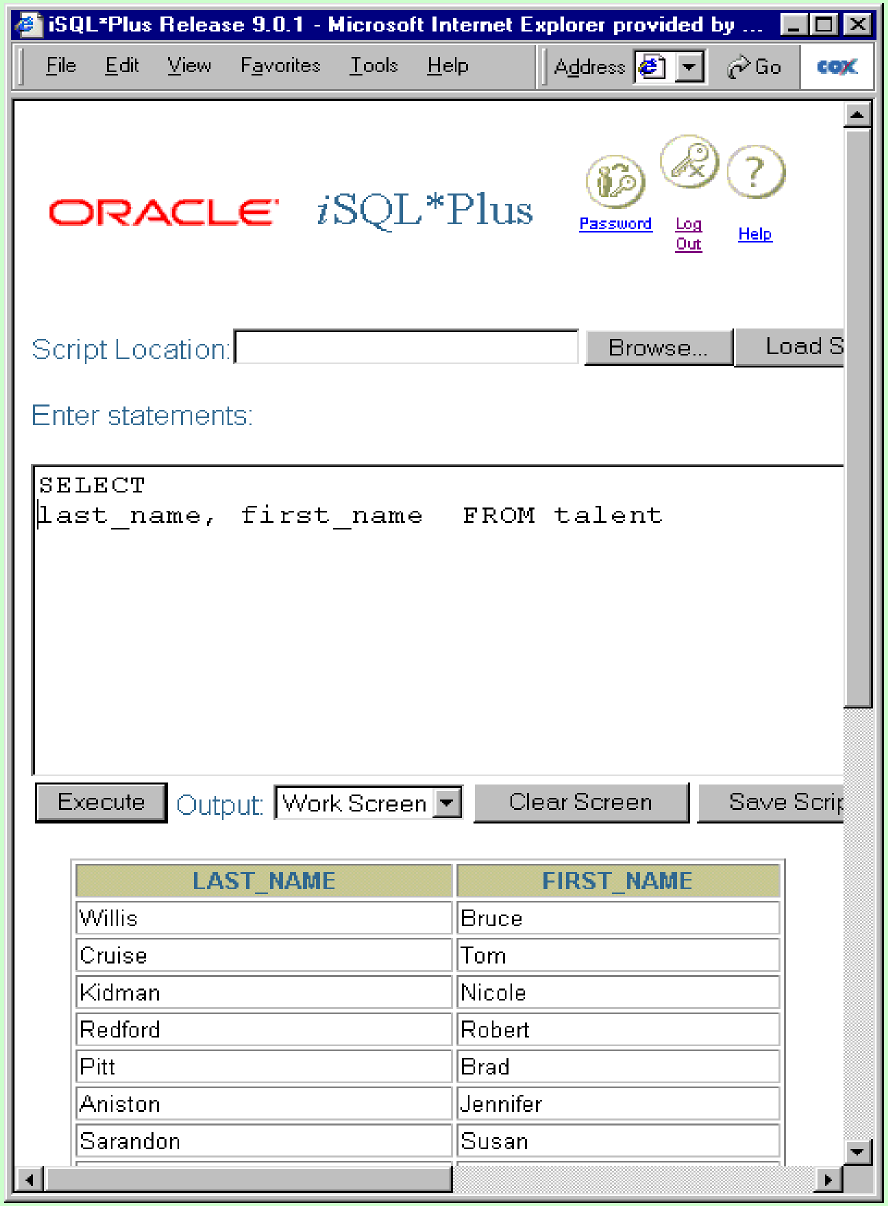
As you step thru these examples, you might notice that the meaning of the program can be obscured by its form. Hence, the goal of all stylistic conventions is to come up with a guideline that allows the meaning of the program to stand out, and be readily apparent to the reader.



```
SELECT   last_name,  first_name   FROM talent
```

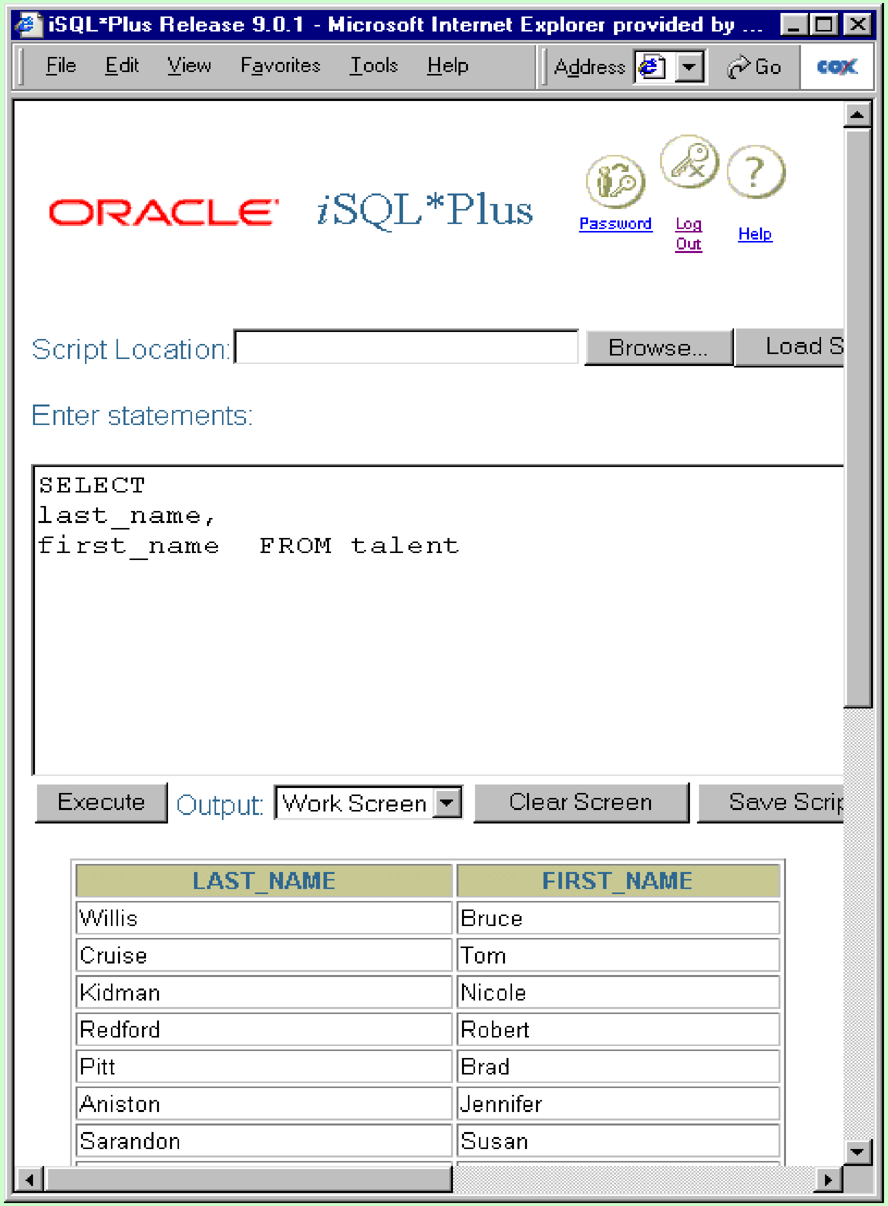
LAST_NAME	FIRST_NAME
Willis	Bruce
Cruise	Tom
Kidman	Nicole
Redford	Robert
Pitt	Brad
Aniston	Jennifer
Sarandon	Susan

Page D-5: SC cont



```
SELECT
last_name,  first_name   FROM talent
```

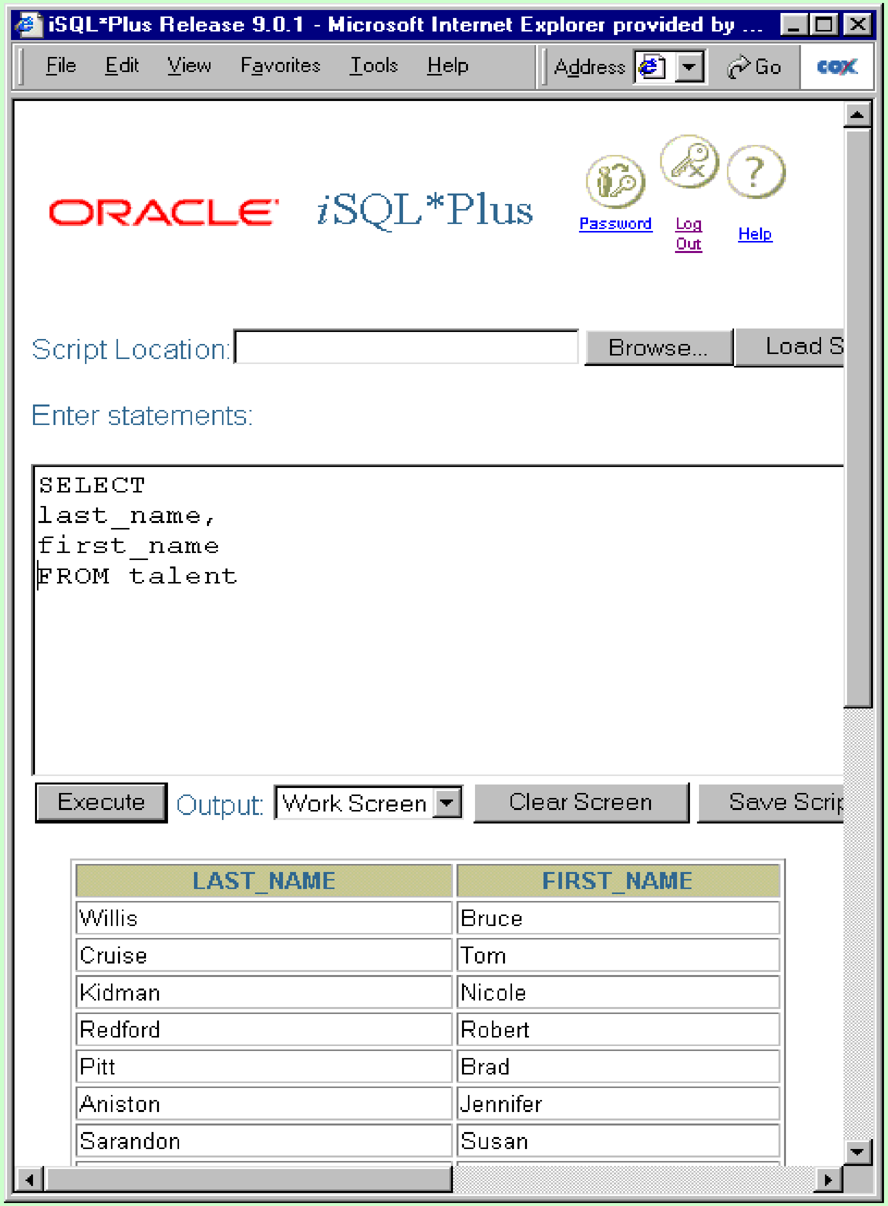
LAST_NAME	FIRST_NAME
Willis	Bruce
Cruise	Tom
Kidman	Nicole
Redford	Robert
Pitt	Brad
Aniston	Jennifer
Sarandon	Susan



```
SELECT
last_name,
first_name   FROM talent
```

LAST_NAME	FIRST_NAME
Willis	Bruce
Cruise	Tom
Kidman	Nicole
Redford	Robert
Pitt	Brad
Aniston	Jennifer
Sarandon	Susan

Page D-7: SC cont

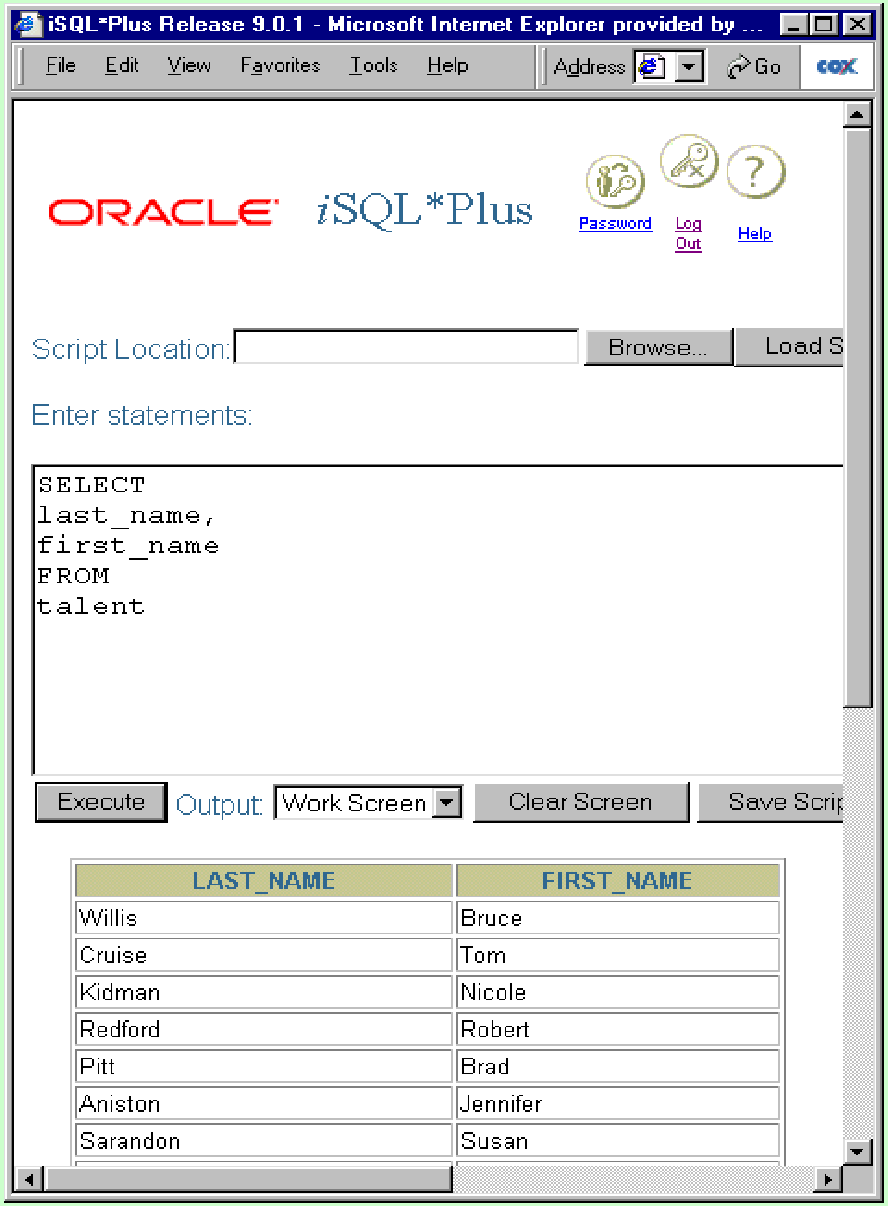


```
SELECT
last_name,
first_name
FROM talent
```

LAST_NAME	FIRST_NAME
Willis	Bruce
Cruise	Tom
Kidman	Nicole
Redford	Robert
Pitt	Brad
Aniston	Jennifer
Sarandon	Susan

Module 02: Basic SQL

Page D-8: SC cont



```
SELECT
last_name,
first_name
FROM
talent
```

LAST_NAME	FIRST_NAME
Willis	Bruce
Cruise	Tom
Kidman	Nicole
Redford	Robert
Pitt	Brad
Aniston	Jennifer
Sarandon	Susan

Module 02: Basic SQL

Page E-1: Problem 4

The user community has issued yet another request for IT services.

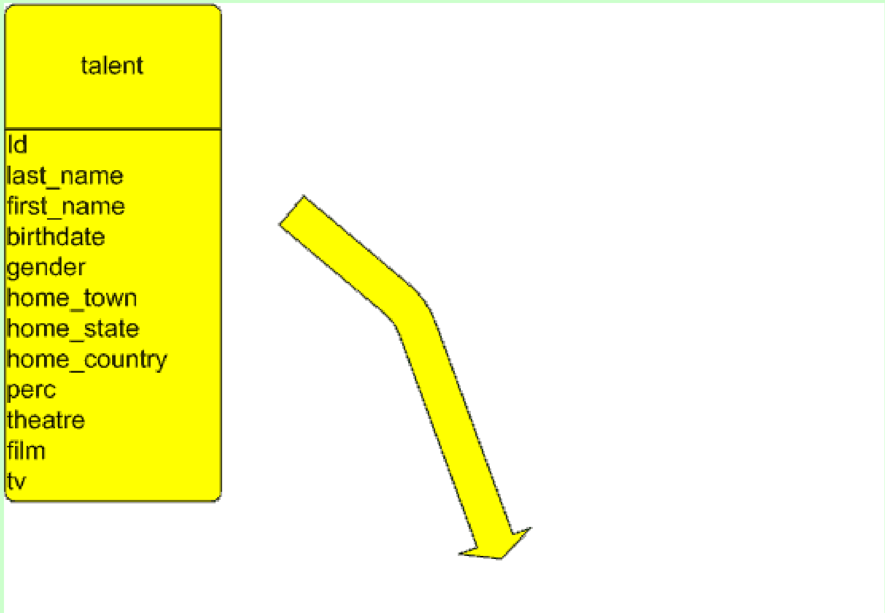
As it turned out, they weren't too happy with that name report we gave them. They didn't like the column heading over the name columns, and they'd like a new report that shows the same information, but without the underscore character (_) in the names of the columns.

Page E-2 Problem 4- TBC Model

Let’s sketch out a quick solution for the problem using a TBC.

You know how to build the column name and table name entries, but how do we go about renaming the column names?

Here’s where we can use another of the rows in the TBC. The ‘Alias’ row is used when you want to rename (or provide an alias for) a column in the result table.

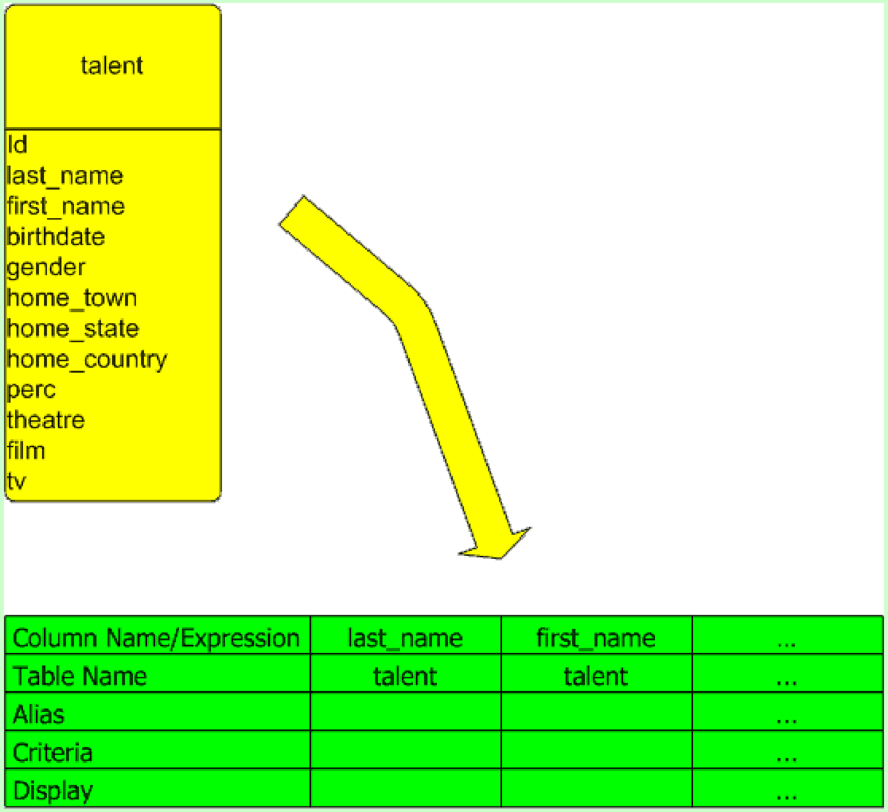


talent

Id
last_name
first_name
birthdate
gender
home_town
home_state
home_country
perc
theatre
film
tv

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

Page E-3: Problem 4 - Coding



Column Name/Expression	last_name	first_name	...
Table Name	talent	talent	...
Alias			...
Criteria			...
Display			...

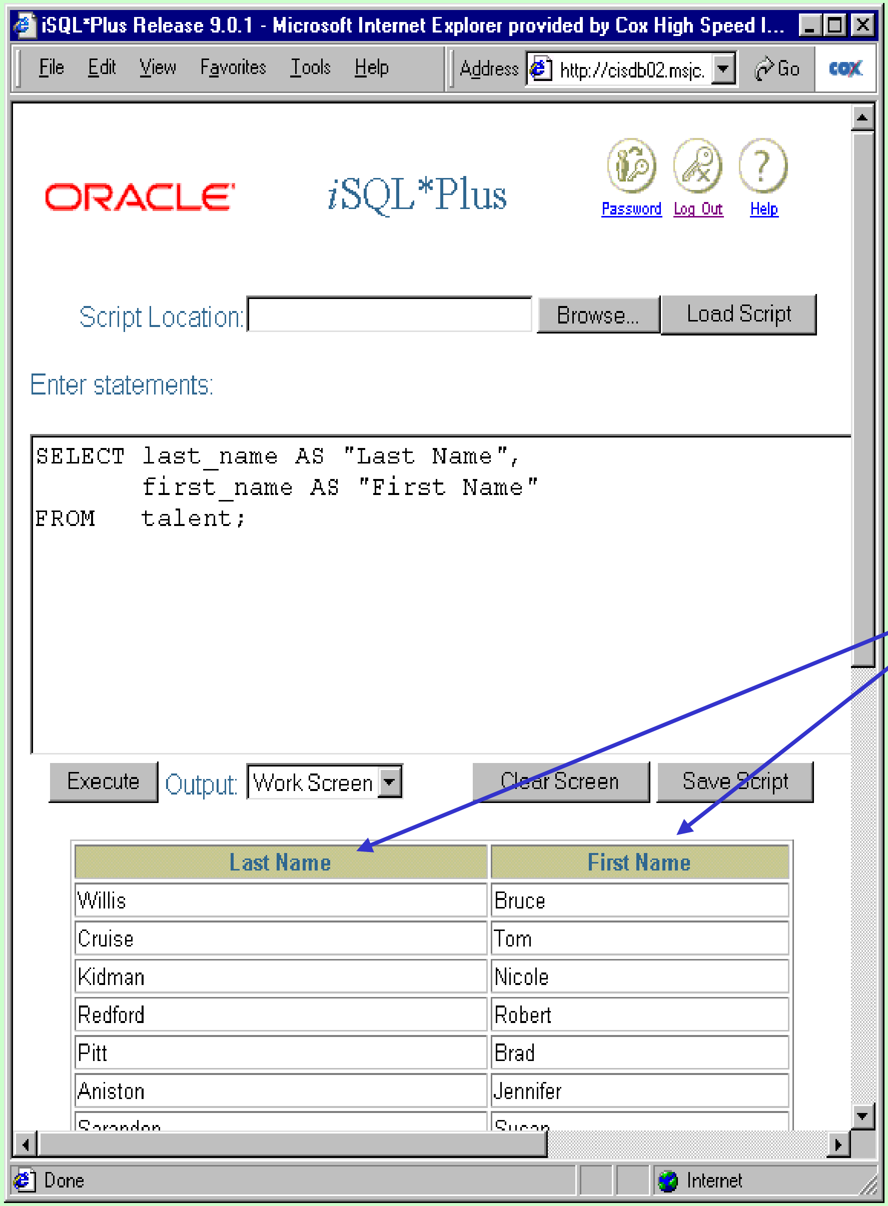
Transform the TBC into code.

Notice how the column aliasing is handled

```
SELECT    last_name AS “Last Name”,
          first_name AS “First Name”
FROM      talent;
```

Enter the code in iSQL-Plus, and execute your program.

Page E-4: Problem 4 – (cont)



```
SELECT last_name AS "Last Name",
       first_name AS "First Name"
FROM   talent;
```

Last Name	First Name
Willis	Bruce
Cruise	Tom
Kidman	Nicole
Redford	Robert
Pitt	Brad
Aniston	Jennifer
Sarandon	Susan

Type your SQL program in the box labeled: Enter Statements.

Click on the Execute button.

Look for your output to be displayed below the input area.

Notice how the column headings have been renamed.

Page E-5: Problem 4 Analysis

We use the keyword AS to specify an AliaS. I think it’s a pretty good mnemonic that the first and last letters in the word *alias* spell *AS*.

Note: The use of AS is optional in some dialects of SQL, and this code is permitted

```
SELECT    last_name “Last Name”,
          first_name “First Name”
FROM     talent;
```

But this is difficult to read, so one more stylistic convention:

> Always use the AS keyword when specifying a column alias.

Page E-6: Problem 4 Analysis

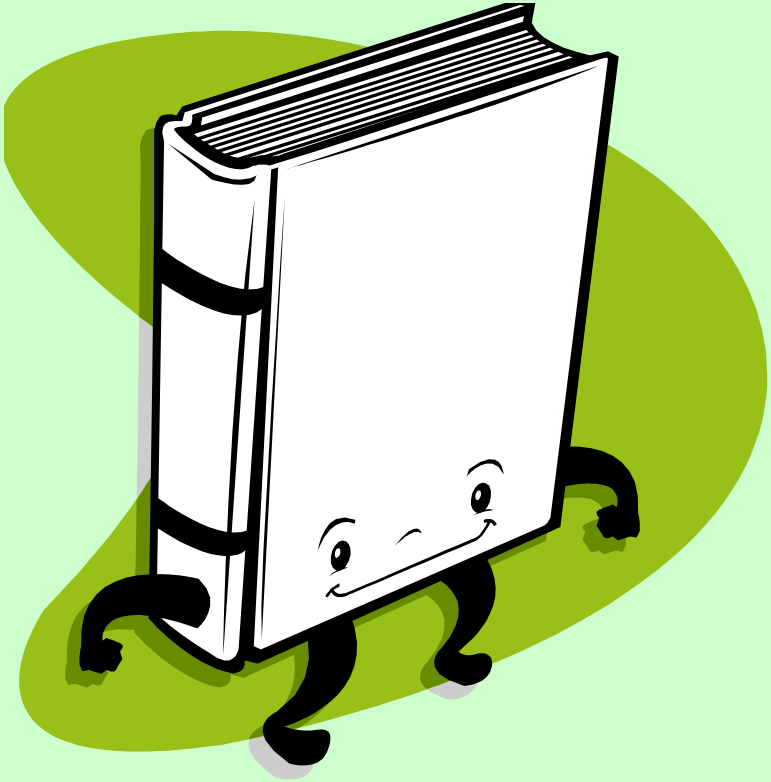
Commas are still used to identify column boundaries in the result table. Count the commas in the SELECT clause and add 1 to determine the number of columns that’ll appear in the result table.

What happens when you type in this code:

```
SELECT    last_name,  “Last Name”,
          first_name, “First Name”
FROM     talent;
```

And column aliases are the only place that double quotes are allowed in SQL (at least until I tell you otherwise).

Page T-1: Terminology



Base table
Result table

Table Build Chart

Asterisk wildcard

Metadata
Metadata command – DESCRIBE

Stylistic conventions

Use of single-quotes ‘
Use of double-quotes “

Alias

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