# SQL Programming

Distinct

## Page A-1: Intro

SQL provides a mechanism to insure that duplicate rows do not appear in the result table.

One of the critical features of a pure relational database design is that duplicate rows should not exist in any of the base tables.

The theory behind  relational db systems that mandates unique rows in base tables also suggests that rows in the result table be unique.

However SQL does NOT guarantee unique rows in the result table.  But, if uniqueness is critical to the particular application, SQL does have the necessary tools to support that need.

## Page B-1 DISTINCT

The DISTINCT phrase can be used within the SELECT clause to insure that duplicate rows are not created in the result table.

The additional syntax for this feature specifies that the DISTINCT keyword appear immediately after the keyword SELECT:

SELECT [ALL | DISTINCT] *column-list*

As you can see, there are two options here: ALL or DISTINCT. The use of brackets [] to enclose these keywords indicates that their use is optional. So in this description we see that the programmer may specify ALL or DISTINCT, or neither of those keywords.

The ALL option is the default, hence these two forms are equivalent:

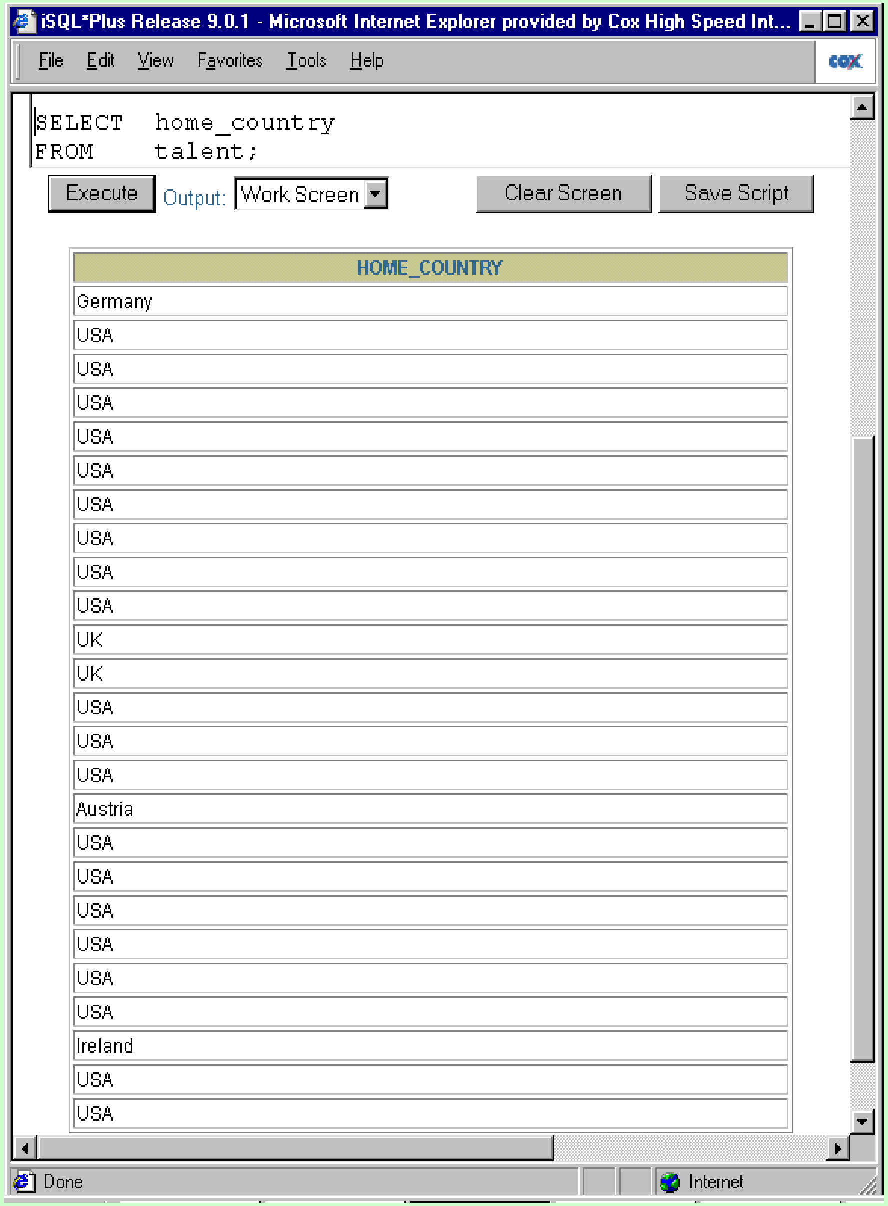
SELECT *column-list*

SELECT ALL *column-list*

As to stylistic conventions, I see little value in using the ALL keyword. It is commonly expected that all rows will be returned, hence the use of ALL does not improve readability nor understanding.

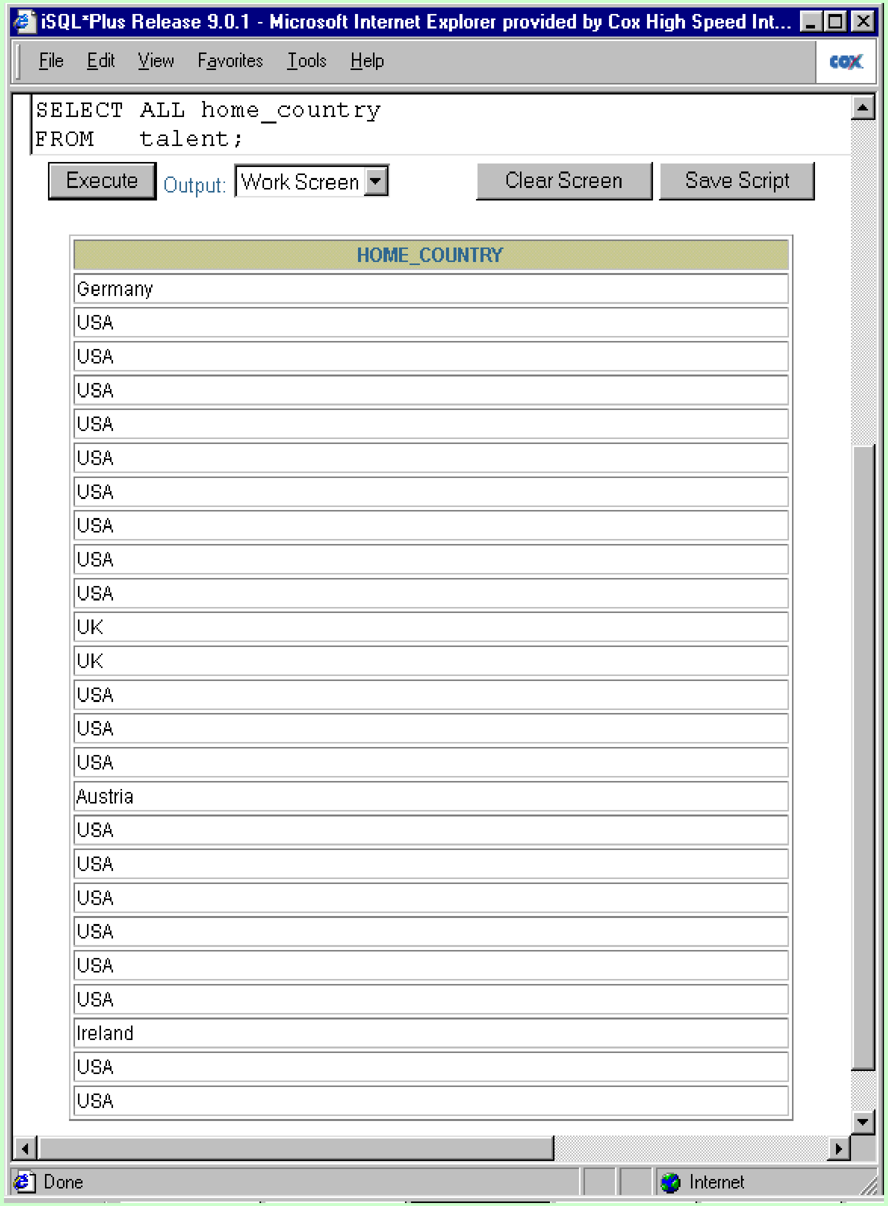
# Module 10: Distinct

# Page B-2 ALL



```
SELECT   home_country
FROM     talent;
```

| HOME_COUNTRY |
|---|
| Germany |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| UK |
| UK |
| USA |
| USA |
| USA |
| Austria |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| Ireland |
| USA |
| USA |

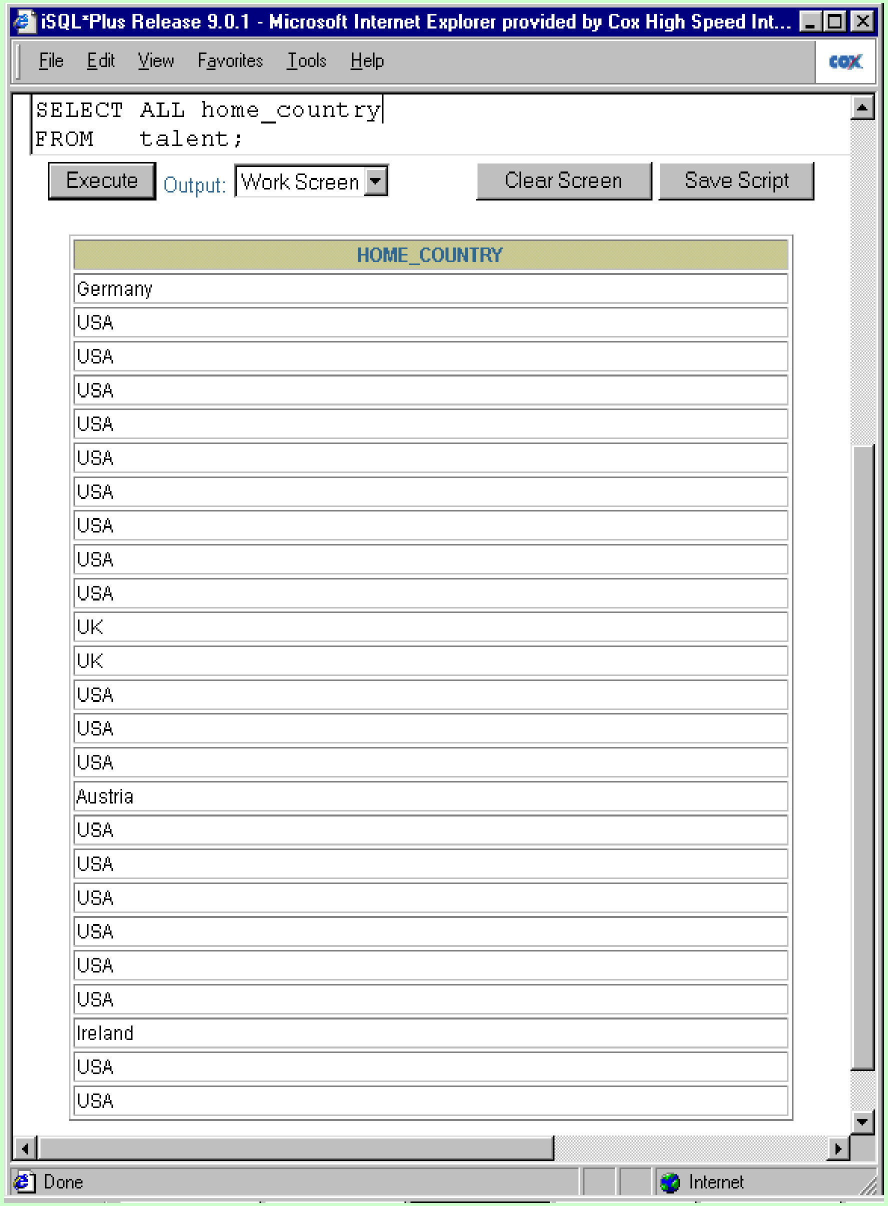


```
SELECT ALL home_country
FROM   talent;
```

| HOME_COUNTRY |
|---|
| Germany |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| UK |
| UK |
| USA |
| USA |
| USA |
| Austria |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| Ireland |
| USA |
| USA |

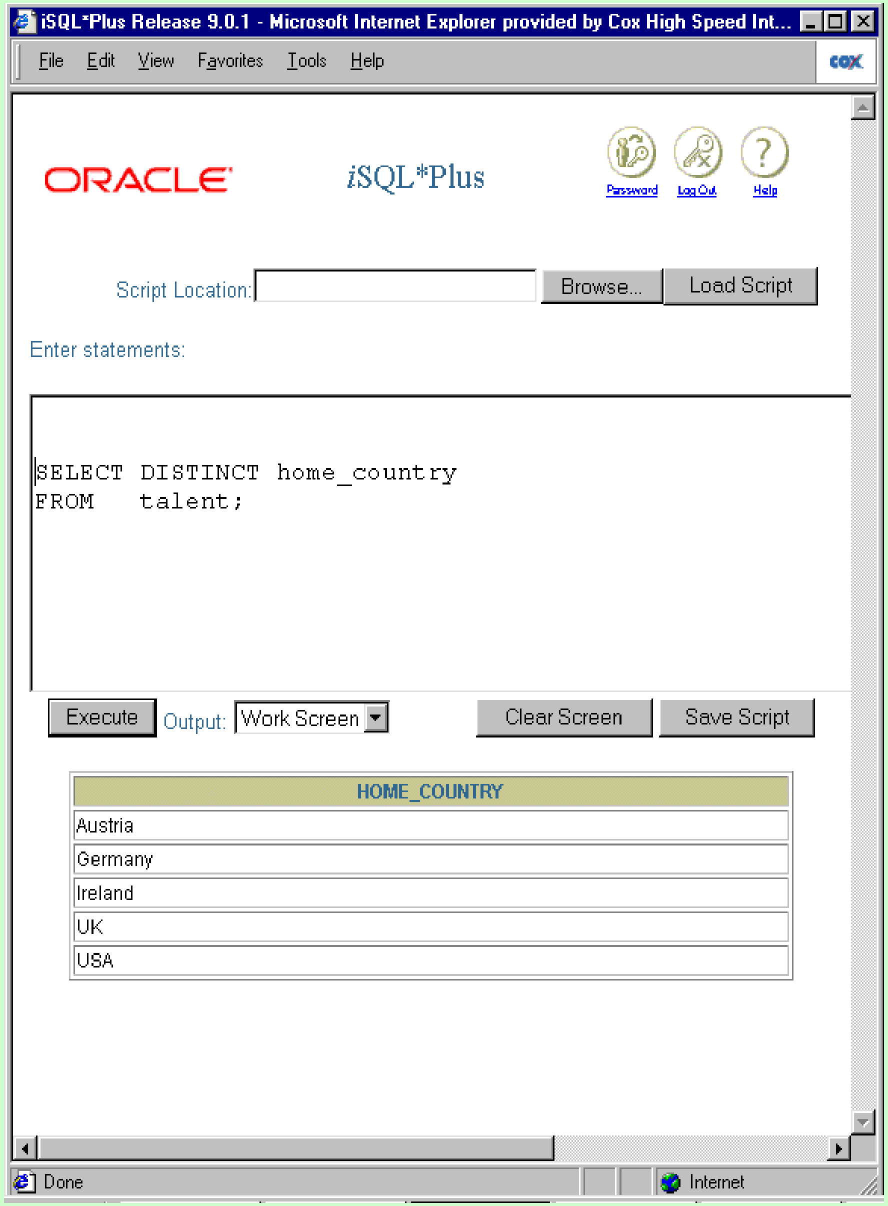
# Module 10: Distinct

# Page B-3 DISTINCT



```
SELECT ALL home_country
FROM   talent;
```

| HOME_COUNTRY |
|---|
| Germany |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| UK |
| UK |
| USA |
| USA |
| USA |
| Austria |
| USA |
| USA |
| USA |
| USA |
| USA |
| USA |
| Ireland |
| USA |
| USA |

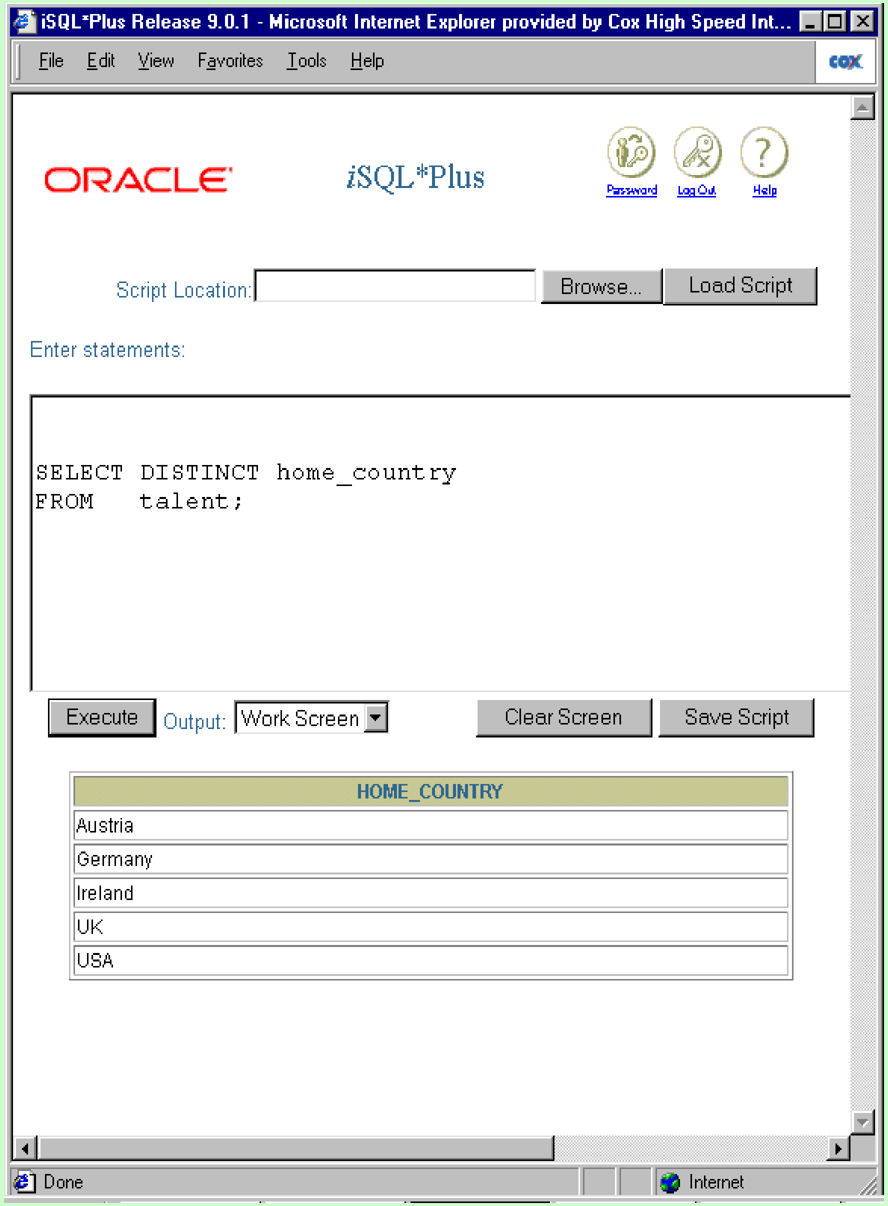


```
SELECT DISTINCT home_country
FROM   talent;
```

| HOME_COUNTRY |
|---|
| Austria |
| Germany |
| Ireland |
| UK |
| USA |

# Module 10: Distinct

# Page B-4: Sorted Output



```
SELECT DISTINCT home_country
FROM   talent;
```

| HOME_COUNTRY |
|---|
| Austria |
| Germany |
| Ireland |
| UK |
| USA |

On the previous slides you had the opportunity to view SQL statements that used each of these keywords.

As an aside, did you notice that the rows in the output table were sorted when the SELECT statement specified DISTINCT?

Although this is a common behavior in many vendor implementations of SQL, it is not part of the SQL standard. So, if sorted output is important in your application, explicitly use the ORDER BY clause in conjunction with the DISTINCT phrase.

## Page B-5: Cross-Platform Development

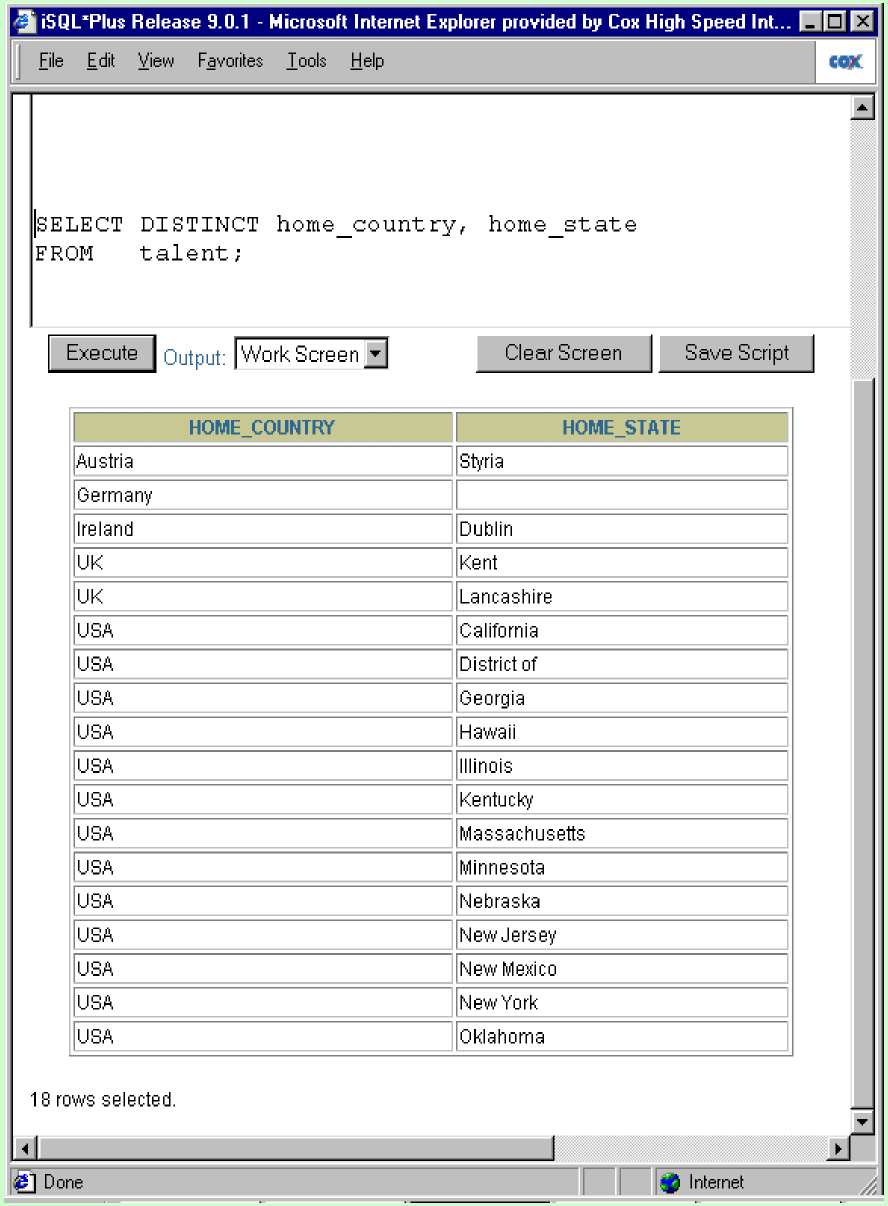
Let me clarify this point.

If you are writing a SQL application that is expected to run on many platforms (Oracle, MySQL, SQL Server, …) then you'll want to write your code so that it conforms to the SQL standard.

Similarly, if you think that your company might change database vendors someday, then you'll also want to write 'to the standard'.

In my opinion, the only time you'll ever want to deviate from the standard is when you're trying to optimize your code (application) for performance reasons. But be aware of the costs you incur when you do so. Every program that deviates from the standard will have to be re-written, tested, and debugged, if and when you decide to port it (convert it) to another platform.

# Page B-6 Scope



```
SELECT DISTINCT home_country, home_state
FROM   talent;
```

| HOME_COUNTRY | HOME_STATE |
|---|---|
| Austria | Styria |
| Germany | |
| Ireland | Dublin |
| UK | Kent |
| UK | Lancashire |
| USA | California |
| USA | District of |
| USA | Georgia |
| USA | Hawaii |
| USA | Illinois |
| USA | Kentucky |
| USA | Massachusetts |
| USA | Minnesota |
| USA | Nebraska |
| USA | New Jersey |
| USA | New Mexico |
| USA | New York |
| USA | Oklahoma |

18 rows selected.

Again, you should notice that the DISTINCT operator removes duplicate ***rows*** from the result table.

In this example, only unique rows, that is combinations of home_country and home_state values, are carried forward to the result table.

In this regard then, we can say that the *scope* of the DISTINCT operator is the entire column list

## Page B-7 Intuitions

Now as we've been working though these modules one of the things that I've tried to emphasize is that you develop your intuitions about how SQL operates. This conceptual understanding will help you when you come to learn the more complex features of the language.

In this regard I have emphasized that the WHERE clause identifies rows in the base table that should be *carried forward* to the result table. The WHERE clause tells SQL which rows to keep.

I have specifically avoided using phrases such as 'the WHERE clause is used to *eliminate* rows', because the notion of 'identifying the keepers' is critical to my understanding of how SQL works.

## Page B-8: Intuitions - 2

But here we are discussing the nature of the DISTINCT phrase and I'm talking about eliminating duplicate rows, removing duplicate entries.

What's up with that? Why the change in focus?

'Cause that's the way I think it works.

While the WHERE clause is all about keeping rows and carrying them forward, I see the DISTINCT phrase as simply eliminating, and removing duplicates

But if it helps you, we can phrase the workings of the DISTINCT phrase in a more positive light.

*The scope of the DISTINCT phrase is the entire column-list, in this regard, the DISTINCT option ensures that only single instances of unique column patterns are carried over to the result table.*

## Page C-1: Conceptual Inquiry

Should DISTINCT be a phrase, or a clause?

DISTINCT and ALL are phrases within the SELECT clause, and quite frankly, this placement just doesn't jibe with my intuitions.

Does it make sense to you?

When I think about the clauses in the Select statement, my reasoning runs along these lines:

FROM: makes a copy of the base table(s) for this statements processing.

WHERE: identifies which base table rows are carried forward to the result table

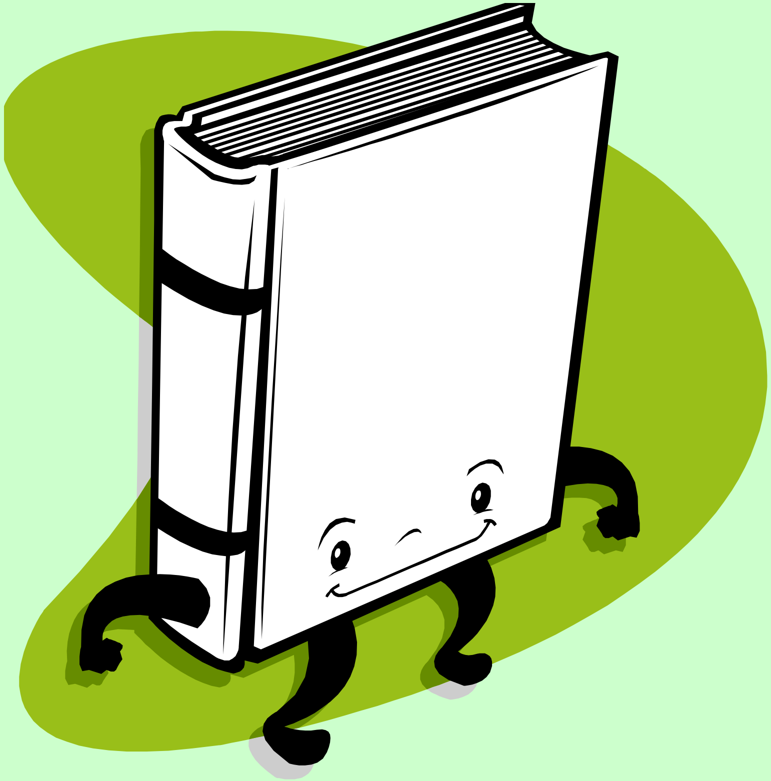
SELECT: identifies which columns are ultimately kept in the result table

## Page C-2: Conceptual Inquiry

With these kinds of intuitions the SELECT clause is the last place I’d expect any row-level activity to be taking place.

Post your comments to this inquiry on the Bb site.

# Page T-1: Terminology



DISTINCT, unique

ALL

Scope

Port, convert

## 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