Filtering and Ordering Data

Introduction to SQL clauses

Clauses are portions of SQL that allow for further refinement of the query or additional work that must be accomplished by the SQL statement. The following clauses are covered in this section:

- The WHERE clause
- The ORDER BY clause

Filtering and Ordering Data

The WHERE clause

The most common SQL clause is the WHERE clause. The WHERE clause is used in conjunction with the SELECT statement to deliver a more refined search based on individual field criteria. This example could be used to extract a specific employee based on a first name:

```
SELECT * FROM employees WHERE name = 'Wally'
```

This statement uses the WHERE clause to limit the result returned to rows where the employee's name matches 'Wally'. The following would be shown (assuming only one employee had that name):

employeeid	name	username	password	email	roleid
1	Wally	wwebmaster	password	wally@vectacorp.com	1

It's important to note that the selection is made only when a certain criteria is true. If a record with the name of 'Wally' did not exist, it wouldn't return anything.

Filtering and Ordering Data

Looking for fields with null values

A field with a NULL value is a field with no value. If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field. When this is the case, the field will be saved with a NULL value. Consider the following data:

employeeid	name	username	password	email	roleid
1	Wally	wwebmaster	password	wally@vectacorp.com	1
2	Wilbur	wfounder	password	wilbur@vectacorp.com	2
3	Tina	ttechie		tina@vectacorp.com	1

We could query all fields with null values as follows:

```
SELECT name FROM employees WHERE password IS NULL
```

Filtering and Ordering Data

The ORDER BY clause

The ORDER BY clause provides you with a quick way of sorting the results of your query in either ascending or descending order. Consider the following table of information:

employeeid	name	username	password	email	roleid
1	Wally	wwebmaster	password	wally@vectacorp.com	1
2	Wilbur	wfounder	password	wilbur@vectacorp.com	2
3	Tina	ttechie	abc123	tina@vectacorp.com	1
4	Agnes	aaccountant	12345	agnes@vectacorp.com	2
5	Damon	ddeveloper	ispeakbinary	damon@vectacorp.com	1

If you selected all the records by using a SELECT all statement (SELECT *), it would return all the results, ordering them based on the value in the employeeid field: 1 through 5.

Filtering and Ordering Data

The ORDER BY clause

Using the SELECT command with an ORDER BY clause allows you to sort based on a different field name:

```
SELECT * FROM employees ORDER BY name
```

The preceding statement would return results, alphabetized in ascending order by name. You can also order by multiple columns by adding a comma after the field name and entering a second field name:

```
SELECT * FROM employees ORDER BY name, email
```

In this case, all records with identical name fields are sorted next by email. You can also adjust the ordering of the list. By default, results are returned in ascending (ASC) order. If you wanted the result in descending order, it would be written as follows:

```
SELECT * FROM employees ORDER BY name DESC
```