Chapter 8: ORDER BY

Section 8.1: Sorting by column number (instead of name)

You can use a column's number (where the leftmost column is '1') to indicate which column to base the sort on, instead of describing the column by its name.

Pro: If you think it's likely you might change column names later, doing so won't break this code.

Con: This will generally reduce readability of the query (It's instantly clear what 'ORDER BY Reputation' means, while 'ORDER BY 14' requires some counting, probably with a finger on the screen.)

This query sorts result by the info in relative column position `3` from select statement instead of column name `Reputation`.

SELECT DisplayName, JoinDate, Reputation FROM Users ORDER BY 3

DisplayName	JoinDate	Reputation
Community	2008-09-15	**1**
Jarrod Dixon	2008-10-03	**11739**
Geoff Dalgas	2008-10-03	**12567**
Joel Spolsky	2008-09-16	**25784**
Jeff Atwood	2008-09-16	**37628**

Section 8.2: Use ORDER BY with TOP to return the top x rows based on a column's value

In this example, we can use GROUP BY not only determined the *sort* of the rows returned, but also what rows *are* returned, since we're using TOP to limit the result set.

Let's say we want to return the top 5 highest reputation users from an unnamed popular Q&A site.

Without ORDER BY

This query returns the Top 5 rows ordered by the default, which in this case is "Id", the first column in the table (even though it's not a column shown in the results).

```
SELECT TOP 5 DisplayName, Reputation
FROM Users
```

returns...

DisplayName	Reputation
Community	1
Geoff Dalgas	12567
Jarrod Dixon	11739
Jeff Atwood	37628
Joel Spolsky	25784

With ORDER BY

```
SELECT TOP 5 DisplayName, Reputation
FROM Users
```

```
ORDER BY Reputation desc
```

returns...

DisplayName	Reputation
JonSkeet	**865023**
Darin Dimitrov	**661741**
BalusC	**650237**
Hans Passant	**625870**
Marc Gravell	**601636**

Remarks

Some versions of SQL (such as MySQL) use a `LIMIT` clause at the end of a `SELECT`, instead of `TOP` at the beginning, for example:

```
SELECT DisplayName, Reputation
FROM Users
ORDER BY Reputation DESC
LIMIT 5
```

Section 8.3: Customizeed sorting order

To sort this table `Employee` by department, you would use `ORDER BY Department`. However, if you want a different sort order that is not alphabetical, you have to map the `Department` values into different values that sort correctly; this can be done with a CASE expression:

Name	Department
Hasan	IT
Yusuf	HR
Hillary	HR
Joe	IT
Merry	HR
Ken	Accountant

```
SELECT *
FROM Employee
ORDER BY CASE Department
         WHEN 'HR'         THEN 1
         WHEN 'Accountant' THEN 2
         ELSE                   3
         END;
```

Name	Department
Yusuf	**HR**
Hillary	**HR**
Merry	**HR**
Ken	**Accountant**
Hasan	**IT**
Joe	**IT**

Section 8.4: Order by Alias

Due to logical query processing order, alias can be used in order by.

```
SELECT DisplayName, JoinDate as jd, Reputation as rep
FROM Users
ORDER BY jd, rep
```

And can use relative order of the columns in the select statement .Consider the same example as above and instead of using alias use the relative order like for display name it is 1 , for Jd it is 2 and so on

```
SELECT DisplayName, JoinDate as jd, Reputation as rep
FROM Users
ORDER BY 2, 3
```

Section 8.5: Sorting by multiple columns

SELECT DisplayName, JoinDate, Reputation FROM Users ORDER BY JoinDate, Reputation

DisplayName	JoinDate	Reputation
Community	**2008-09-15**	**1**
Jeff Atwood	**2008-09-16**	**25784**
Joel Spolsky	2008-09-16	**37628**
Jarrod Dixon	**2008-10-03**	**11739**
Geoff Dalgas	2008-10-03	**12567**