

GROUP BY functionality.

The GROUP BY functionality allows categories of groups to be created. This is very useful when used in conjunction with aggregate operators (On-line students should read the example on Aggregate operators first).

Let's look at an example of GROUP BY:

---

---

```
mysql> SELECT E.ClassID, COUNT(*) AS NumEnrolled
-> FROM Enrolled E
-> GROUP BY ClassID;
+-----+-----+
| ClassID | NumEnrolled |
+-----+-----+
| CS314   |           2 |
| CS430   |           1 |
| CS575   |           2 |
| CT320   |           1 |
+-----+-----+
4 rows in set (0.00 sec)
```

---

---

In this example, we have asked for a list of the classes and a count of the number of students enrolled in each. Every non-aggregate on the SELECT statement must be in the GROUP BY clause as shown above. The aggregate operation is then applied to the group. This keeps the problem described in the Aggregate function example from happening where you had an aggregate number trying to be combined with a row-by-row number.

The HAVING clause is a way of further qualifying the grouping done in the GROUP BY clause. Let's look at an example:

---

---

```
mysql> SELECT E.ClassID, COUNT(*) AS Num_Enrolled FROM Enrolled E
-> GROUP BY ClassID
-> HAVING COUNT(*) > 1;
+-----+-----+
| ClassID | Num_Enrolled |
+-----+-----+
| CS314   |           2 |
| CS575   |           2 |
+-----+-----+
2 rows in set (0.00 sec)
```

---

---

We used the HAVING clause to further eliminate classes whose COUNT was greater than 1.

This becomes a powerful feature, but has limitations.

Let's look at the makeup of a SELECT statement:

```
SELECT [DISTINCT] select-list  
FROM from-list  
WHERE row qualifications  
GROUP BY grouping-list  
HAVING group-qualification
```

The rules for using the GROUP BY and HAVING clauses are:

- The select-list is made up of aggregate operators and attributes (column names). To avoid the problem talked about in the aggregate operators example, every attribute in the select-list must be in the grouping-list.
- The expressions in the HAVING clause must have a single value per group. This determines whether an answer row is generated for a given group.
- If the group by is omitted, the entire table is a single group.

This is a good time to look at the order of operations in a SELECT statement. Let's look at the order:

1. FROM clause
2. WHERE clause
3. GROUP BY clause
4. HAVING clause
5. SELECT clause
6. ORDER BY clause

The FROM defines the relations, the WHERE defines the rows in the relation, GROUP BY build the grouping records, HAVING is a delimiter on those. At this point, the fields are selected and sorted.