Today …

- group by, having

Homework

- HW 7 due Thurs
Consider the following schema

\begin{verbatim}
customer(cust_num, name, address, c_rating, c_amount, c_balance, sp_num)
salesperson(sp_num, name, address, office)
\end{verbatim}

*Foreign Key*: customer.sp_num \rightarrow salesperson.sp_num

And the instance

\begin{center}
\begin{tabular}{cccccccc}
\hline
cust_num & name & address & c_rating & c_amount & c_balance & sp_num \\
1 & Alice & xxx & 5 & 1000 & 1000 & 101 \\
2 & Bob & yyy & 7 & 5000 & 4000 & 101 \\
3 & Chuck & zzz & 10 & 10000 & 1000 & 102 \\
\hline
\end{tabular}
\end{center}

Q: What does this query return?

\begin{verbatim}
SELECT sp_num, COUNT(*)
FROM customer
WHERE c_rating >= 5
GROUP BY sp_num;
\end{verbatim}

\begin{center}
\begin{tabular}{cc}
\hline
sp_num & COUNT(*) \\
101 & 2 \\
102 & 1 \\
\hline
\end{tabular}
\end{center}
Exercises:

Q: Write an SQL query that returns for each sales office the corresponding average credit balance of the customers whose salesperson is located in the office.

```sql
SELECT s.office, AVG(c.c_balance)
FROM salesperson s JOIN customer c USING (sp_num)
GROUP BY s.office
```

Q: Write an SQL query that returns the minimum credit rating of each salesperson's customers who have a credit amount over $1,000 and who work in the “Spokane” office.

```sql
SELECT s.sp_num, MIN(c.c_rating)
FROM salesperson s JOIN customer c USING (sp_num)
WHERE c.c_amount > 1000 AND s.office = 'Spokane'
GROUP BY s.sp_num
```

Q: Write an SQL query that returns the average customer credit amount and credit balance by credit rating.

```sql
SELECT AVG(c_amount), AVG(c_balance)
FROM customer
GROUP BY c_rating
```
A **HAVING** clause filters groups

Q: Can you guess what this query returns?

```
SELECT sp_num, COUNT(*)
FROM customer
GROUP BY sp_num
HAVING COUNT(*) > 1;
```

<table>
<thead>
<tr>
<th>salesperson</th>
<th>COUNT(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>2</td>
</tr>
</tbody>
</table>

- The **HAVING** clause is evaluated **against each group**
- A group is in the answer if it satisfies the **HAVING** conditions
- **HAVING** should (generally) be accompanied by a **GROUP BY**
**A more involved Group By example**

<table>
<thead>
<tr>
<th>cust_num</th>
<th>name</th>
<th>sp_num</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>alice</td>
<td>5</td>
</tr>
<tr>
<td>102</td>
<td>chuck</td>
<td>8</td>
</tr>
<tr>
<td>103</td>
<td>bob</td>
<td>NULL</td>
</tr>
<tr>
<td>106</td>
<td>george</td>
<td>5</td>
</tr>
<tr>
<td>107</td>
<td>debbie</td>
<td>5</td>
</tr>
<tr>
<td>109</td>
<td>eddy</td>
<td>2</td>
</tr>
<tr>
<td>110</td>
<td>fionna</td>
<td>8</td>
</tr>
</tbody>
</table>

`SELECT sp_num, COUNT(*)`  
`FROM customer`  
`GROUP BY sp_num;`

The intermediate result includes 4 groups

| 103  | bob   | NULL |
| 109  | eddy  | 2    |
| 101  | alice | 5    |
| 106  | george| 5    |
| 107  | debbie| 5    |
| 102  | chuck | 8    |
| 110  | fionna| 8    |

With “SELECT sp_num, COUNT(*)” evaluated for each group
<table>
<thead>
<tr>
<th>salesperson</th>
<th>COUNT(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>
Adding the `HAVING` clause ...

```sql
SELECT sp_num, COUNT(*)
FROM customer
GROUP BY sp_num
HAVING COUNT(*) > 1;
```

- Eliminates two of the groups

<table>
<thead>
<tr>
<th>salesperson</th>
<th>COUNT(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>
Group By attributes

What can appear in a SELECT clause under GROUP BY?

• the grouping attributes (attributes in GROUP BY clause) ... (*) see below
• or aggregate operators (which are applied to the group)

This query is not legal in SQL ... but is legal in MySQL

```
SELECT name
FROM customer
GROUP BY sp_num;
```

Q: Why not?

– name is not a grouping attribute, and is not an aggregate
– there could be many names for each group!

Q: What if we group by a primary key?

– then, there could only be one value for other attributes
– e.g.: SELECT name FROM customer GROUP BY cust_num

(*) Actually, can safely use any right-hand-side of a functional dependency

– if the left-hand-side is being grouped

Multiple attributes can appear in a GROUP BY

• Same rule applies ...
• No 2 rows in group have different values for attributes in the GROUP BY
Q: What can appear in a HAVING clause?

- Comparisons to only those attributes in the GROUP BY clause
  - same rules apply w.r.t. keys / FDs as well
- Comparisons to expressions resulting in one value for the group
  - like a comparison to an aggregate operator

**HAVING clause without a GROUP BY**

- Treats result of FROM and WHERE as a single group

Q: How does this differ to a WHERE clause?

- WHERE applies to each row
- HAVING applies to the whole group