1-1 AND RECURSIVE RELATIONSHIPS

Reading: Chapter 6

Organization Chart

- Org charts implicitly show a relationship between employees (e.g., supervisor), but how do we show this in a DB

Managing Director
Alice

Marketing
Ned
Andrew
Clare

Accounting
Todd
Nancy

Purchasing
Brier
Sarah

Personnel & PR
Sophie
1-1 Relationships

- Org chart has 2 relationships
  - A department has one or more employees, but
  - Each department has only one department head

The department/boss relationship is 1:1

Mapping a 1:1 Relationship

- Usual rules apply (foreign key / primary key)
- Where do you put the foreign key?
  - Departments
  - Employees
  - Both tables
Recursive Relationships

- Example
  - Each employee has a boss and
  - Boss has many employees
- Both are in the same table (Employees)
- Define a recursive relationship

Modeling a Recursive Relationship

- A recursive relationship relates an entity to itself
- Label recursive relationships
Mapping a Recursive Relationship

A recursive relationship relates an entity to itself

- Usual rules
- 1:m
  - The entity gets an additional column for the foreign key
  - Need a name different from the primary key
Example

- Two relationships

<table>
<thead>
<tr>
<th>ID</th>
<th>deptName</th>
<th>floor</th>
<th>phone</th>
<th>deptHeadID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Management</td>
<td>5 2001</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Marketing</td>
<td>1 2002</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Accounting</td>
<td>4 2003</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Purchasing</td>
<td>4 2004</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>Personnel &amp; PR</td>
<td>1 2005</td>
<td></td>
<td>9</td>
</tr>
</tbody>
</table>

SELECT firstName, Employees.salary
FROM Employees
WHERE (Employees.ID IN
(SELECT deptHeadID FROM Departments));
Let’s Look at a Sub-Query

• SELECT deptHeadID FROM Departments;

<table>
<thead>
<tr>
<th>deptHeadID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>9</td>
</tr>
</tbody>
</table>

Alternate SQL

• List the salary of each department’s boss.

```sql
SELECT Employees.firstName, Departments.deptName, Employees.salary
FROM Employees INNER JOIN Departments
ON Employees.ID = Departments.deptHeadID;
```

<table>
<thead>
<tr>
<th>firstName</th>
<th>salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>$75,000.00</td>
</tr>
<tr>
<td>Ned</td>
<td>$45,000.00</td>
</tr>
<tr>
<td>Todd</td>
<td>$38,000.00</td>
</tr>
<tr>
<td>Brier</td>
<td>$43,000.00</td>
</tr>
<tr>
<td>Sophie</td>
<td>$35,000.00</td>
</tr>
</tbody>
</table>
Querying a Recursive Relationship

- Find the salary of Nancy's boss.

```
SELECT Employees.firstName, Employees.salary, 
    Bosses.firstName, Bosses.salary, Employees.bossID 
FROM Employees, Employees AS Bosses 
WHERE (((Employees.firstName)='Nancy') AND 
    ((Employees.bossID)=Bosses.ID));
```

MS Access Design View

- Access will allow you to add multiple copies of a table to the design view
- Both tables will be available in the Field drop-down
- Will create a join of the copy with the original