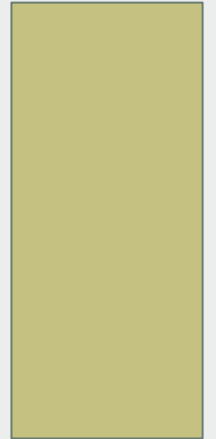


# DATABASES

TRICKS & THEORY

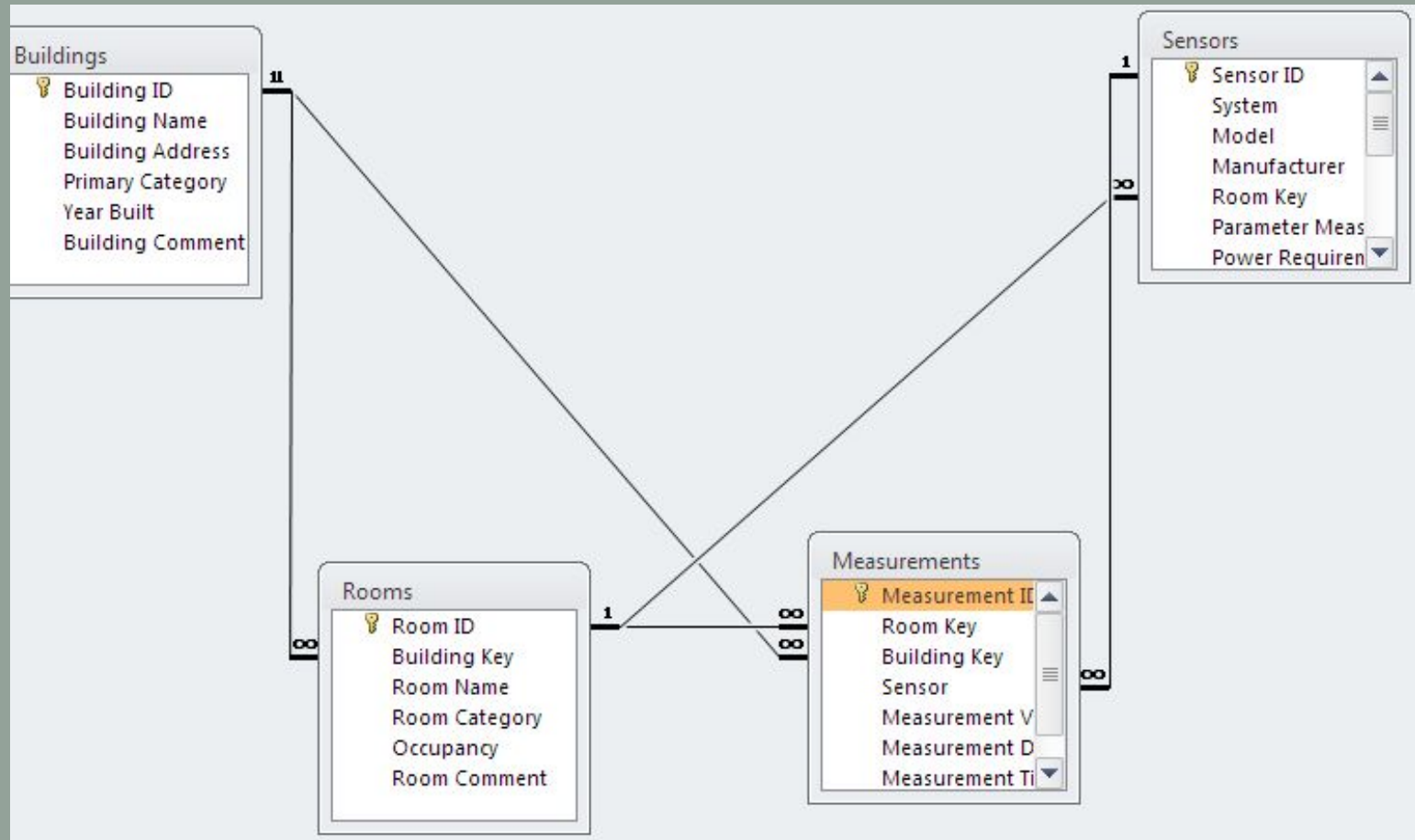


# LET'S LOOK AT THE RESOURCES DB

This is what I created from the info a previous year's students put on the web.

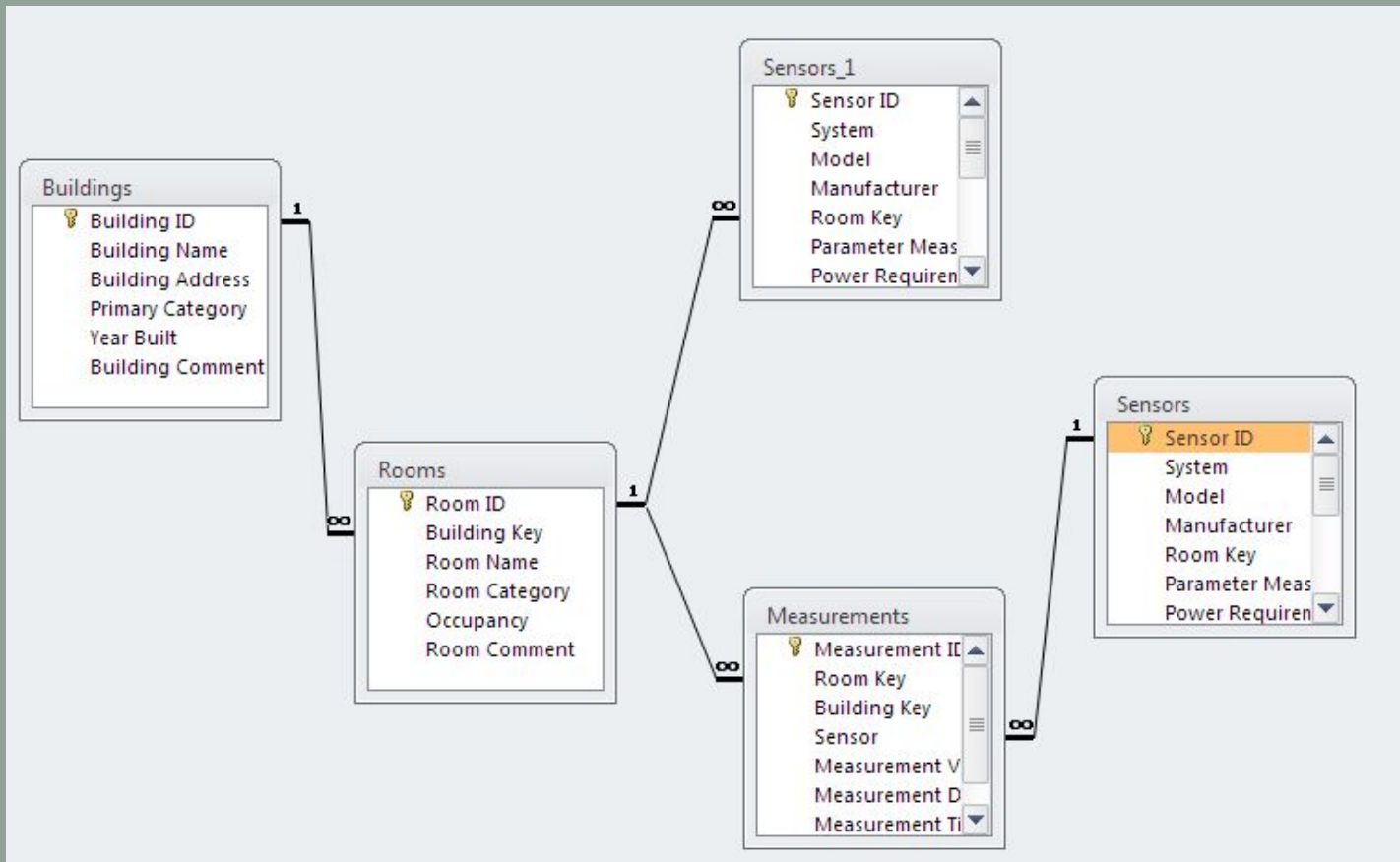
# HAZARDS OF RELATIONS

What's the problem with this Diagram?



# FIXING IT

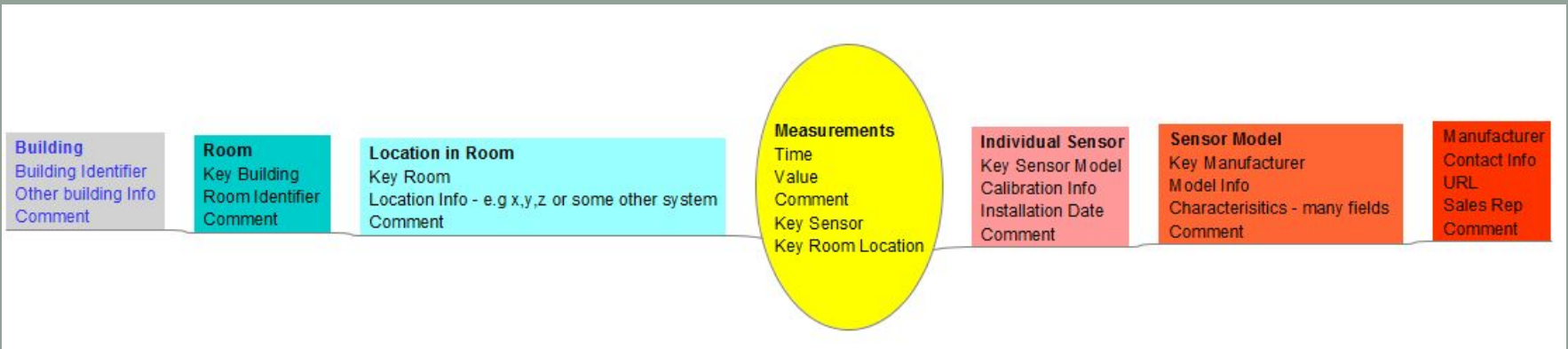
Add a second instance of a single table



# A PRETTY GOOD SET OF TABLES

Here's a view (using Freemind) of a set of tables that might do a good job for real building measurements

- It's easy to see that we could go on adding tables



# RELATIONAL DB LIMITATIONS

It's difficult to make changes

- Changing table relationships can be a nightmare

This means that to do a good job

- **Defining the goals is critical**
  - What you'll store
  - How you want to get it out
    - All the different ways
- Planning the entire DB follows from that need

This leads to other kinds of databases

- More later

# SQL

## What is SQL?

- Structured Query Language

## Why do We Need It

- Same standards issues as for BIM – now mostly solved

## Client Access to hosted DB

- It's the standard way to address DBs

It hides in Access as we'll see

# WHAT IT DOES FOR A DB

## Create

- Tables
- Queries

## Add to

- Forms

## Update

- Special Queries

## Extract from

- Queries
- Reports



# THE HEART OF RDB - SQL JOINS

## Inner Join

- The most common
- What you're creating in your databases

## An Example – From Wikipedia

The Tables

Employee table		Department table	
LastName	DepartmentID	DepartmentID	DepartmentName
Rafferty	31	31	Sales
Jones	33	33	Engineering
Steinberg	33	34	Clerical
Robinson	34	35	Marketing
Smith	34		
John	NULL		

## Inner Join - Leaves out John

IDEmployee	LastName	employee.D	department	Department
1	Rafferty	31	31	Sales
2	Jones	33	33	Engineering
3	Steinberg	33	33	Engineering
4	Robinson	34	34	Clerical
5	Smith	34	34	Clerical
(New)				

# THE HEART OF RDB - SQL JOINS

## Inner Join

- The most common
- What you're creating in your databases

## An Example – From Wikipedia

### The Tables

Employee table		Department table	
LastName	DepartmentID	DepartmentID	DepartmentName
Rafferty	31	31	Sales
Jones	33	33	Engineering
Steinberg	33	34	Clerical
Robinson	34	35	Marketing
Smith	34		
John	NULL		

```
SELECT *  
FROM employee INNER JOIN department ON employee.DepartmentID = department.DepartmentID;
```

### Inner Join - Leaves out John

IDEmployee	LastName	employee.D	department	Department
1	Rafferty	31	31	Sales
2	Jones	33	33	Engineering
3	Steinberg	33	33	Engineering
4	Robinson	34	34	Clerical
5	Smith	34	34	Clerical
(New)				

# OUTER JOIN

## Seldom used

- May create it by mistake
- All rows from a single table with info from other table if it exists
  - Will show a row even if not linked to other table
    - e.g In this same example John is included w/o dept.

```
SELECT *  
FROM employee LEFT JOIN department ON employee.DepartmentID = department.DepartmentID;
```

IDEmployee	LastName	employee.D	department	Department
6	John			
1	Rafferty	31	31	Sales
2	Jones	33	33	Engineering
3	Steinberg	33	33	Engineering
4	Robinson	34	34	Clerical
5	Smith	34	34	Clerical
(New)				

# CARTESIAN JOIN

Almost certainly a big mistake

All permutations of all rows of both table

Here's what we get with these same two tables

```
SELECT *  
FROM employee, department;
```

IDEmployee	LastName	employee.D	department	Department
1	Rafferty	31	31	Sales
1	Rafferty	31	33	Engineering
1	Rafferty	31	34	Clerical
1	Rafferty	31	35	Marketing
2	Jones	33	31	Sales
2	Jones	33	33	Engineering
2	Jones	33	34	Clerical
2	Jones	33	35	Marketing
3	Steinberg	33	31	Sales
3	Steinberg	33	33	Engineering
3	Steinberg	33	34	Clerical
3	Steinberg	33	35	Marketing
4	Robinson	34	31	Sales
4	Robinson	34	33	Engineering
4	Robinson	34	34	Clerical
4	Robinson	34	35	Marketing
5	Smith	34	31	Sales
5	Smith	34	33	Engineering
5	Smith	34	34	Clerical
5	Smith	34	35	Marketing
6	John		31	Sales
6	John		33	Engineering
6	John		34	Clerical
6	John		35	Marketing

# TRICKS OF USING ACCESS

Plan, Plan, Plan - It's hard to change

Test with simple version to be sure basics work - then add

Check the relationships carefully

Use Autonumber

- Don't worry about the value of the number - the user should never see them
- If you don't, then be sure you have a UNIQUE primary key

Establish a consistent naming System

Document what you do

- It's easy to build a documentation table

Learn about Dropdown Boxes

Use separate table for each set of lists

Use Queries rather than lists when working

# OTHER TYPES OF DB

## Flat File

- Excel – yes it works as a database, particularly with Vlookup

## Object Oriented

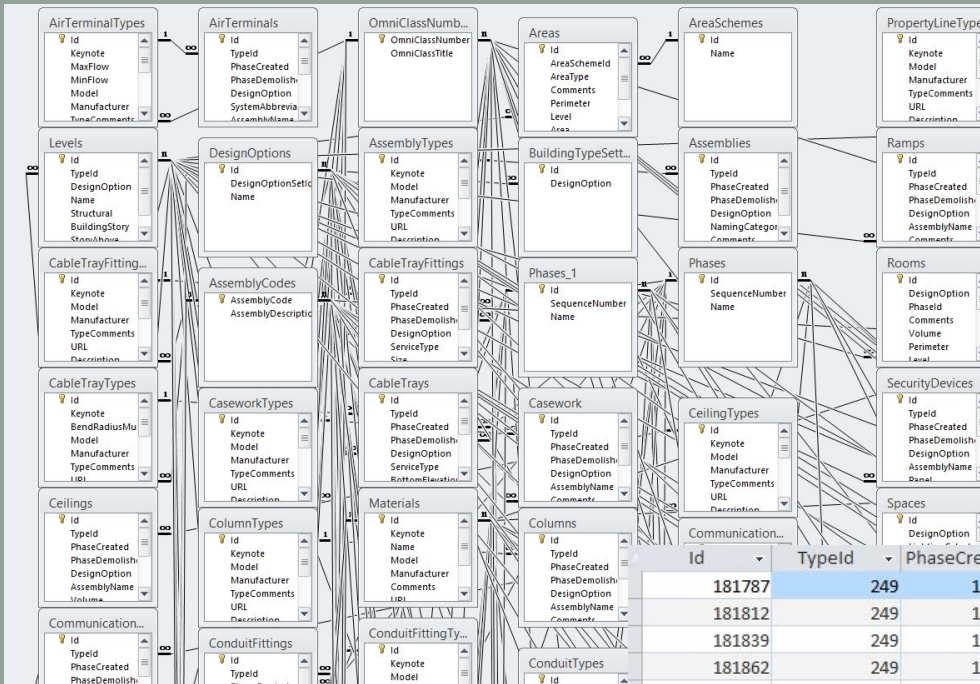
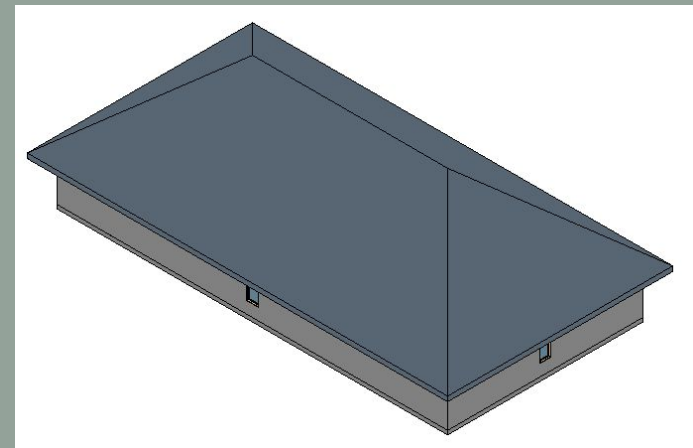
- Much more easily modified
- It's aimed at situations such as we run into with BIM

## Big data - NoSQL

- Yes, it means that it doesn't fully obey SQL rules
- Amazon, Google, Facebook use this approach

# LET'S LOOK AT WHAT REVIT PRODUCES

In Revit a very simple building  
 Can Export as Database  
 This is what get as a piece of the  
 relationship Diagram



*Note: in Revit-2016 this export process has become more powerful AND more complicated.*

Walls Table

Id	TypeId	PhaseCreate	PhaseDemo	DesignOptic	AssemblyNa	EstimatedRe	Volume	Area
181787	249	118390					18.4814618757	90.95207616
181812	249	118390					9.08656143974	44.71732992
181839	249	118390					18.5255103037	91.16884992
181862	249	118390					8.96070878822	44.09797632