BCNF and 3NF

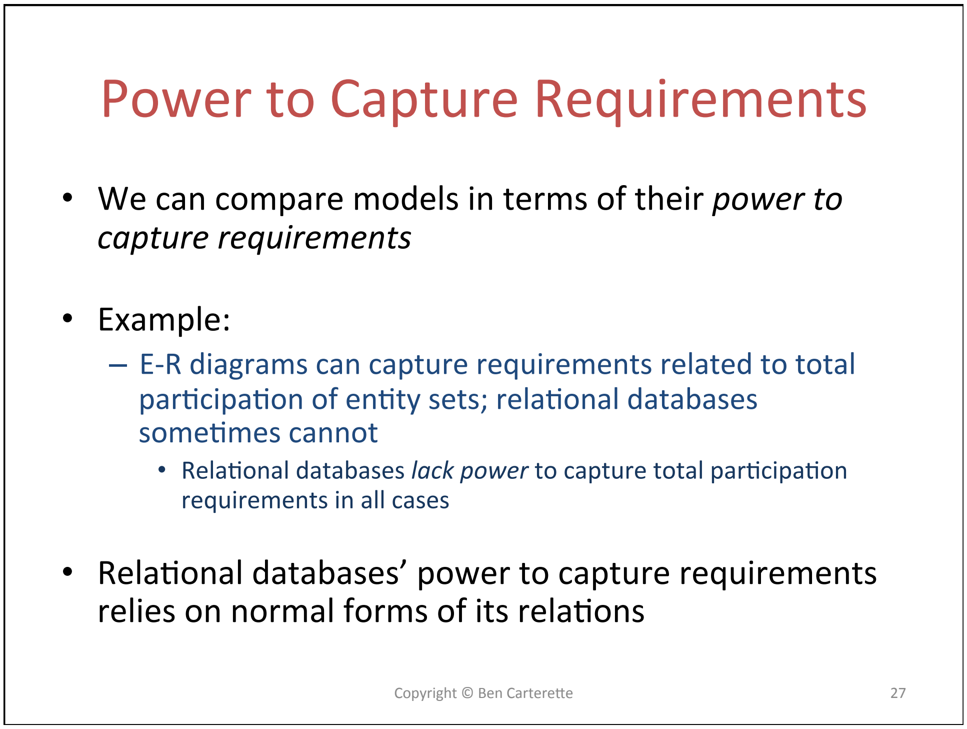
CISC637 Lecture #10

Ben Carterette

Determining BCNF

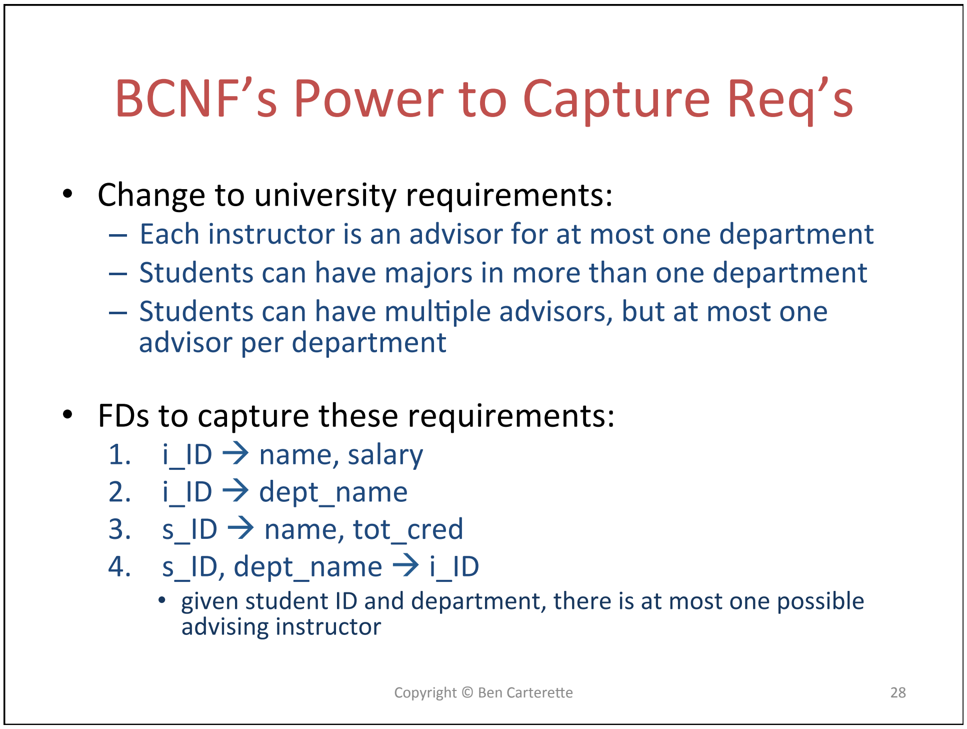
- You have a set of FDs and one or more relations
- You want to know if each relation is in BCNF

- Procedure: For each relation:
  1. Determine its candidate key(s)
  2. Identify any relevant FDs from the set
     - The FDs where all fields are part of the relation, or such FDs that can be derived from others
  3. For each of the relevant FDs, ask:
     A. Is it trivial? If so, go to the next FD
     B. Is the left side one of the candidate keys from #1?
        – If so, go to the next FD
        – If not, the FD violates BCNF – the relation is NOT in BCNF



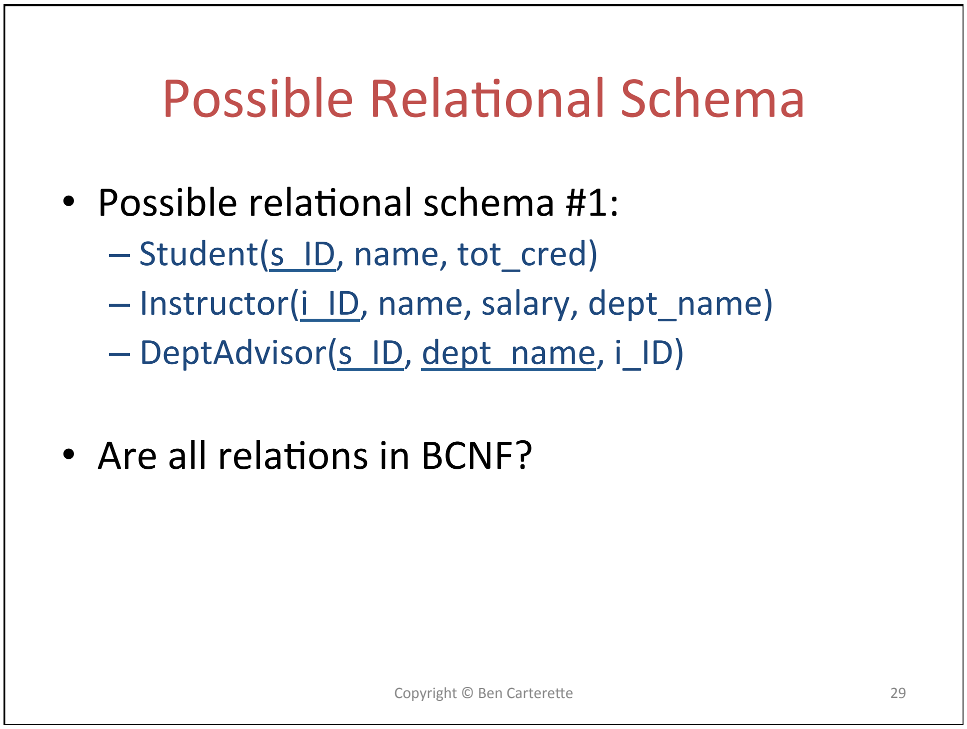
Power to Capture Requirements

- We can compare models in terms of their *power to capture requirements*
- Example:
  - E-R diagrams can capture requirements related to total participation of entity sets; relational databases sometimes cannot
    - Relational databases *lack power* to capture total participation requirements in all cases
- Relational databases' power to capture requirements relies on normal forms of its relations



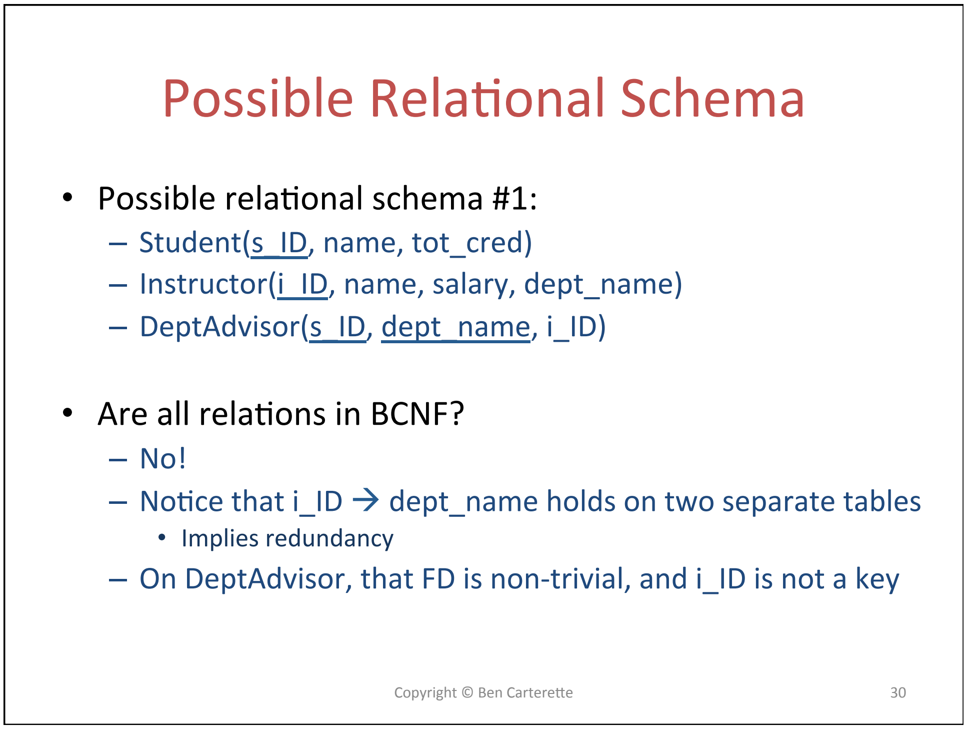
BCNF's Power to Capture Req's

- Change to university requirements:
  - Each instructor is an advisor for at most one department
  - Students can have majors in more than one department
  - Students can have multiple advisors, but at most one advisor per department
- FDs to capture these requirements:
  1. i_ID → name, salary
  2. i_ID → dept_name
  3. s_ID → name, tot_cred
  4. s_ID, dept_name → i_ID
     - given student ID and department, there is at most one possible advising instructor



Possible Relational Schema

- Possible relational schema #1:
  - Student(s_ID, name, tot_cred)
  - Instructor(i_ID, name, salary, dept_name)
  - DeptAdvisor(s_ID, dept_name, i_ID)
- Are all relations in BCNF?



Possible Relational Schema

- Possible relational schema #1:
  - Student(s_ID, name, tot_cred)
  - Instructor(i_ID, name, salary, dept_name)
  - DeptAdvisor(s_ID, dept_name, i_ID)
- Are all relations in BCNF?
  - No!
  - Notice that i_ID → dept_name holds on two separate tables
    - Implies redundancy
  - On DeptAdvisor, that FD is non-trivial, and i_ID is not a key

Possible Relational Schema

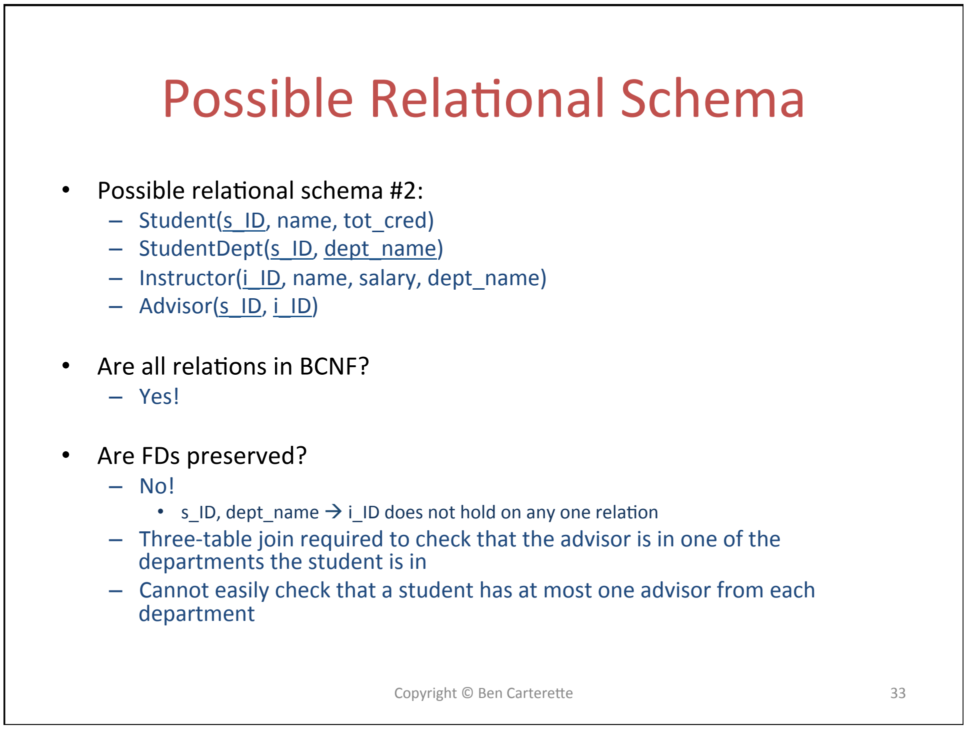
- Possible relational schema #2:
  - Student(s_ID, name, tot_cred)
  - StudentDept(s_ID, dept_name)
  - Instructor(i_ID, name, salary, dept_name)
  - Advisor(s_ID, i_ID)
- Are all relations in BCNF?

:

Possible Relational Schema

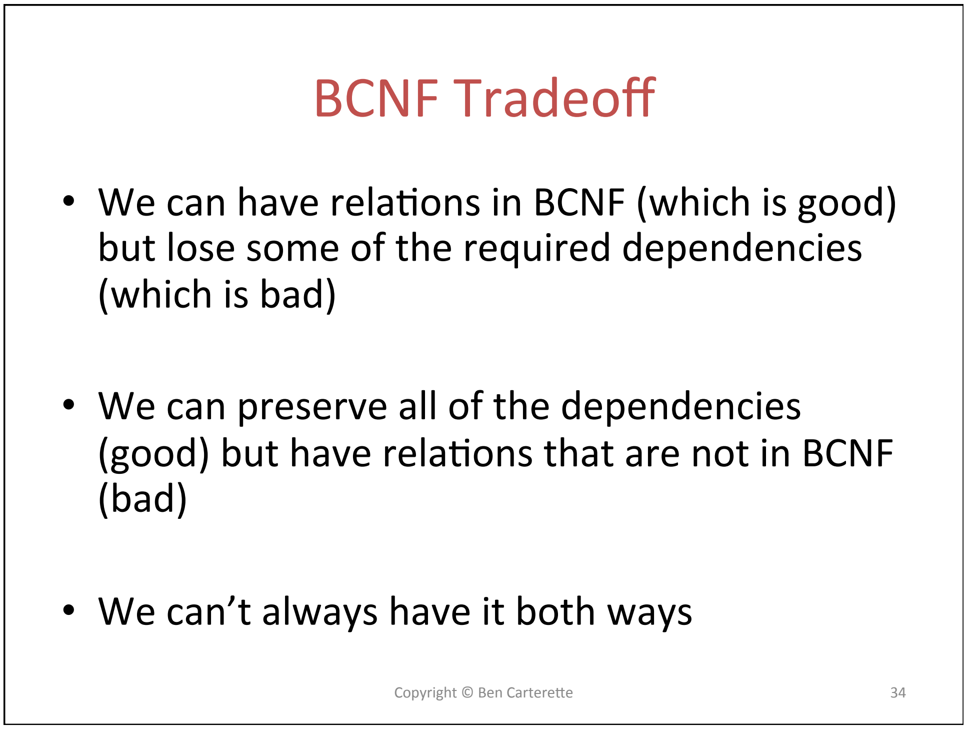
- Possible relational schema #2:
  - Student(s_ID, name, tot_cred)
  - StudentDept(s_ID, dept_name)
  - Instructor(i_ID, name, salary, dept_name)
  - Advisor(s_ID, i_ID)
- Are all relations in BCNF?
  - Yes!
- Are FDs preserved?

:



Possible Relational Schema

- Possible relational schema #2:
  - Student(s_ID, name, tot_cred)
  - StudentDept(s_ID, dept_name)
  - Instructor(i_ID, name, salary, dept_name)
  - Advisor(s_ID, i_ID)
- Are all relations in BCNF?
  - Yes!
- Are FDs preserved?
  - No!
    - s_ID, dept_name → i_ID does not hold on any one relation
  - Three-table join required to check that the advisor is in one of the departments the student is in
  - Cannot easily check that a student has at most one advisor from each department



BCNF Tradeoff

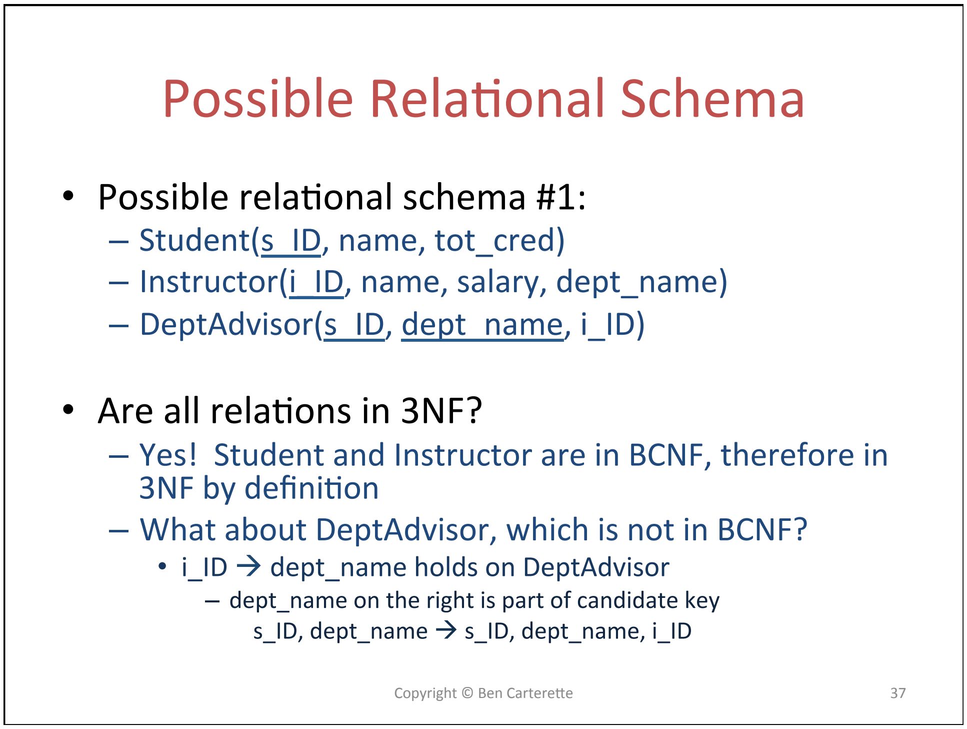
- We can have relations in BCNF (which is good) but lose some of the required dependencies (which is bad)
- We can preserve all of the dependencies (good) but have relations that are not in BCNF (bad)
- We can't always have it both ways

Third Normal Form

- Third Normal Form (3NF) provides a way to retain FDs that cannot be captured in BCNF
  - With minimal additional redundancy
- Checking for 3NF:
  - List all functional dependencies that hold on R
  - Each FD should meet one of the following criteria:
    - FD is trivial, or
    - FD is a superkey/candidate key, or
    - FD has at least one field on the right side that is *part of* some candidate key for R (and that is not on the left side of the same FD)
  - If any FD violates all three criteria, relation is not in 3NF

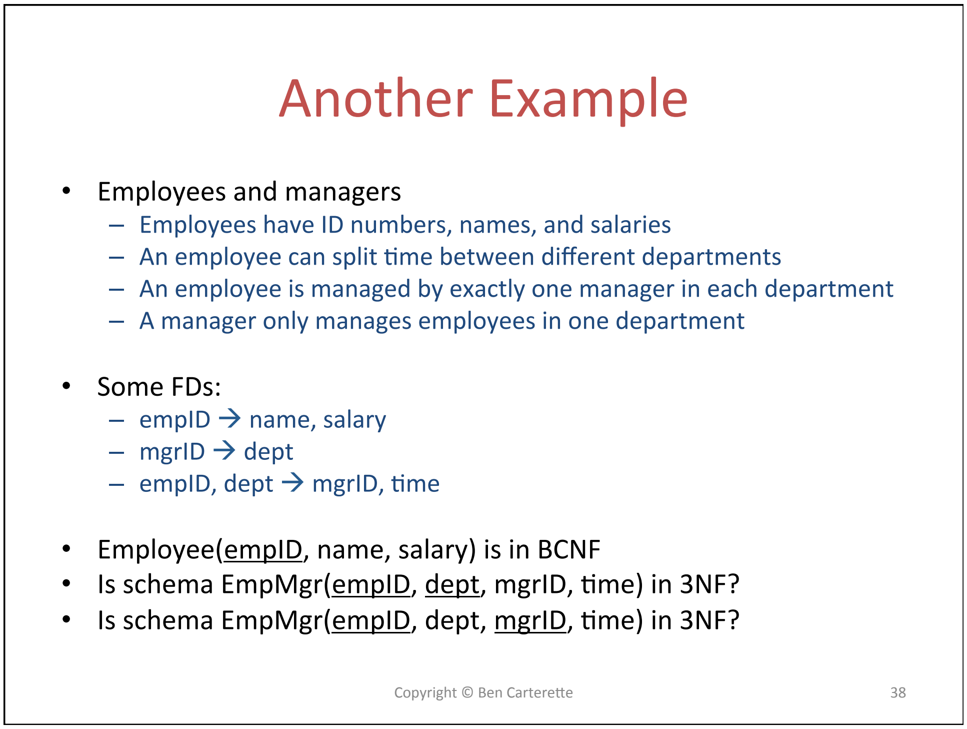
Possible Relational Schema

- Possible relational schema #1:
  - Student(s_ID, name, tot_cred)
  - Instructor(i_ID, name, salary, dept_name)
  - DeptAdvisor(s_ID, dept_name, i_ID)
- Are all relations in 3NF?



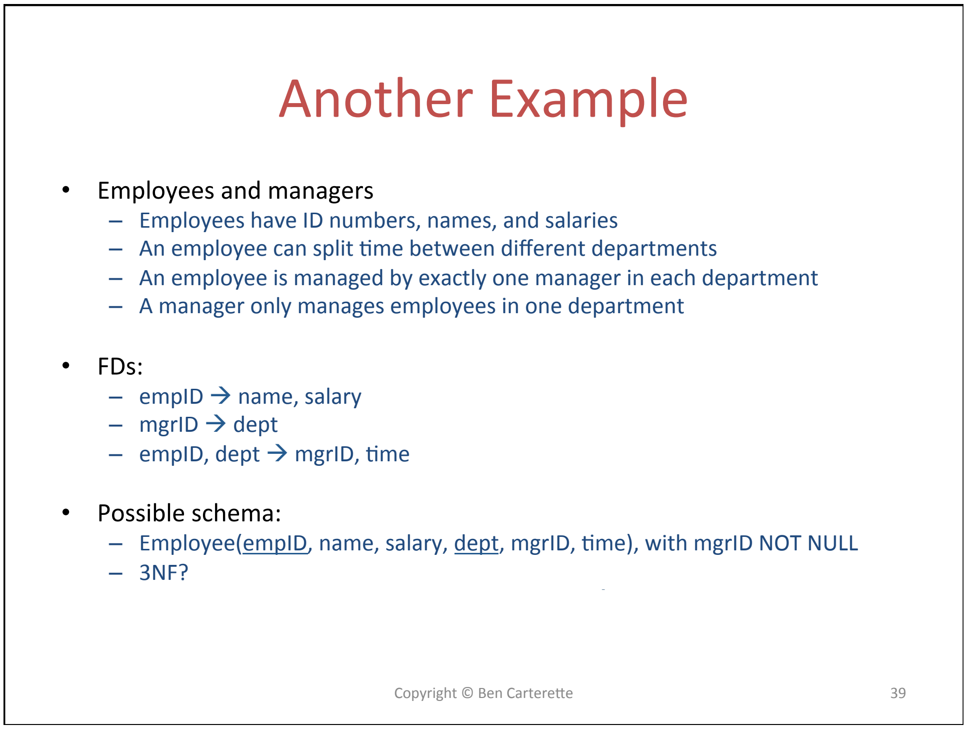
Possible Relational Schema

- Possible relational schema #1:
  - Student(s_ID, name, tot_cred)
  - Instructor(i_ID, name, salary, dept_name)
  - DeptAdvisor(s_ID, dept_name, i_ID)
- Are all relations in 3NF?
  - Yes! Student and Instructor are in BCNF, therefore in 3NF by definition
  - What about DeptAdvisor, which is not in BCNF?
    - i_ID → dept_name holds on DeptAdvisor
      - dept_name on the right is part of candidate key
        s_ID, dept_name → s_ID, dept_name, i_ID



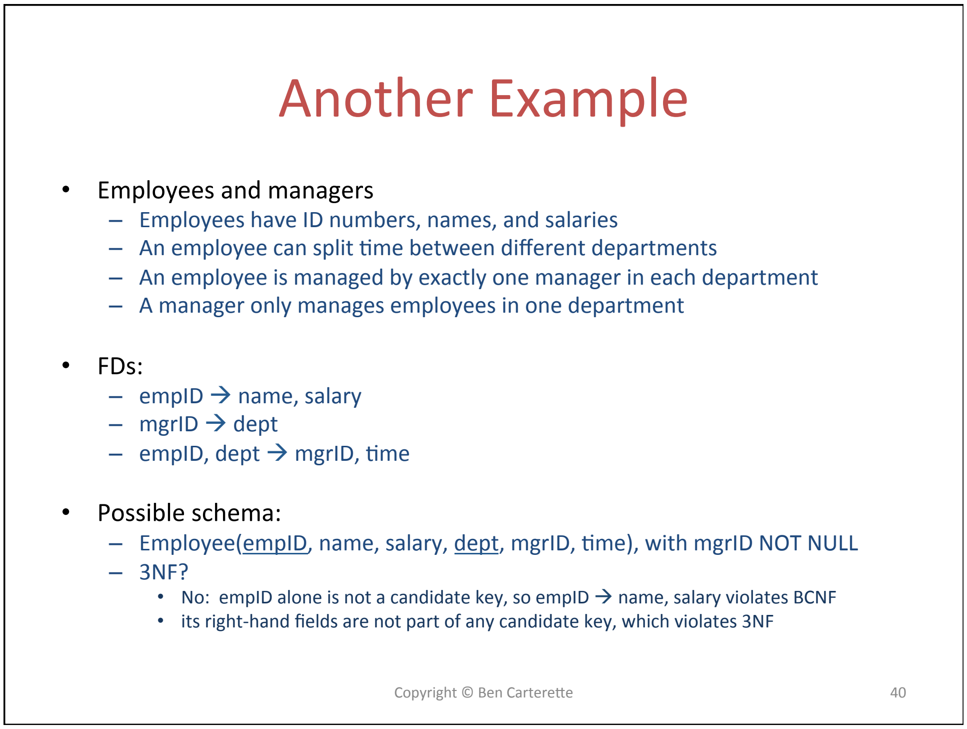
Another Example

- Employees and managers
  - Employees have ID numbers, names, and salaries
  - An employee can split time between different departments
  - An employee is managed by exactly one manager in each department
  - A manager only manages employees in one department
- Some FDs:
  - empID → name, salary
  - mgrID → dept
  - empID, dept → mgrID, time
- Employee(empID, name, salary) is in BCNF
- Is schema EmpMgr(empID, dept, mgrID, time) in 3NF?
- Is schema EmpMgr(empID, dept, mgrID, time) in 3NF?



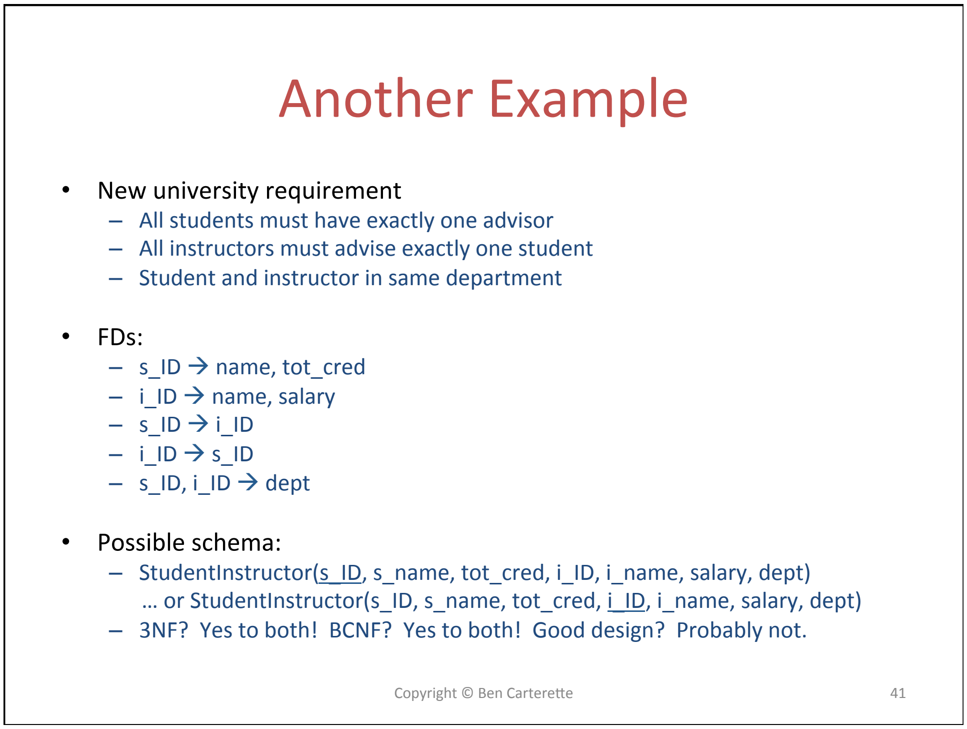
Another Example

- Employees and managers
  - Employees have ID numbers, names, and salaries
  - An employee can split time between different departments
  - An employee is managed by exactly one manager in each department
  - A manager only manages employees in one department
- FDs:
  - empID → name, salary
  - mgrID → dept
  - empID, dept → mgrID, time
- Possible schema:
  - Employee(empID, name, salary, dept, mgrID, time), with mgrID NOT NULL
  - 3NF?



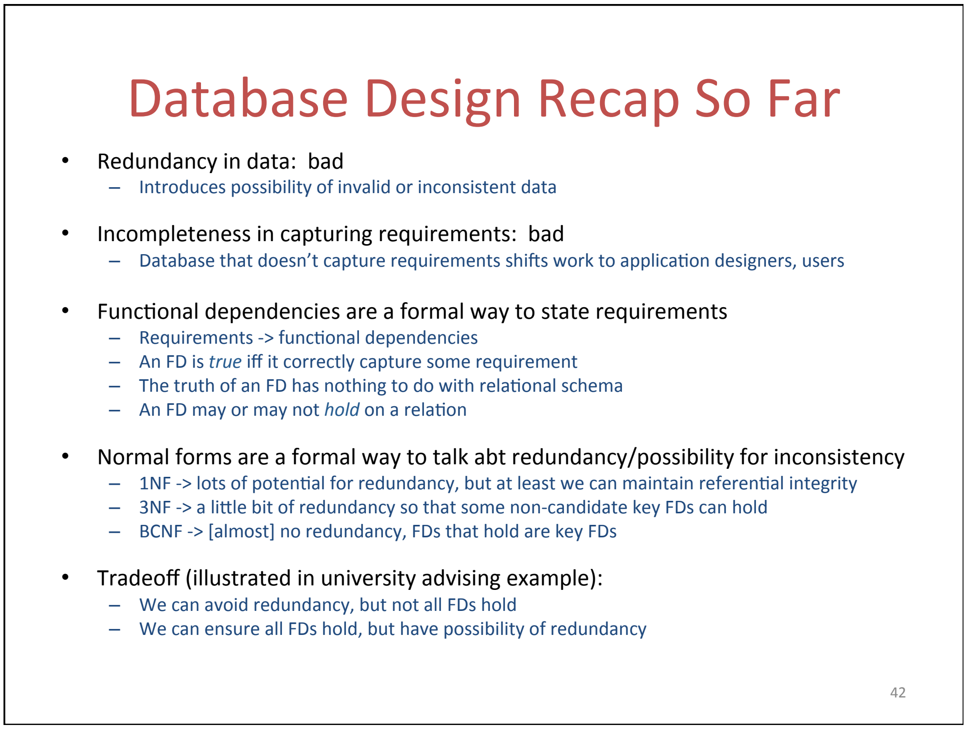
Another Example

- Employees and managers
  - Employees have ID numbers, names, and salaries
  - An employee can split time between different departments
  - An employee is managed by exactly one manager in each department
  - A manager only manages employees in one department
- FDs:
  - empID → name, salary
  - mgrID → dept
  - empID, dept → mgrID, time
- Possible schema:
  - Employee(empID, name, salary, dept, mgrID, time), with mgrID NOT NULL
  - 3NF?
    - No: empID alone is not a candidate key, so empID → name, salary violates BCNF
    - its right-hand fields are not part of any candidate key, which violates 3NF



Another Example

- New university requirement
  - All students must have exactly one advisor
  - All instructors must advise exactly one student
  - Student and instructor in same department
- FDs:
  - s_ID → name, tot_cred
  - i_ID → name, salary
  - s_ID → i_ID
  - i_ID → s_ID
  - s_ID, i_ID → dept
- Possible schema:
  - StudentInstructor(s_ID, s_name, tot_cred, i_ID, i_name, salary, dept)
    … or StudentInstructor(s_ID, s_name, tot_cred, i_ID, i_name, salary, dept)
  - 3NF? Yes to both! BCNF? Yes to both! Good design? Probably not.



Database Design Recap So Far

- Redundancy in data: bad
  - Introduces possibility of invalid or inconsistent data
- Incompleteness in capturing requirements: bad
  - Database that doesn’t capture requirements shifts work to application designers, users
- Functional dependencies are a formal way to state requirements
  - Requirements -> functional dependencies
  - An FD is *true* iff it correctly capture some requirement
  - The truth of an FD has nothing to do with relational schema
  - An FD may or may not *hold* on a relation
- Normal forms are a formal way to talk abt redundancy/possibility for inconsistency
  - 1NF -> lots of potential for redundancy, but at least we can maintain referential integrity
  - 3NF -> a little bit of redundancy so that some non-candidate key FDs can hold
  - BCNF -> [almost] no redundancy, FDs that hold are key FDs
- Tradeoff (illustrated in university advising example):
  - We can avoid redundancy, but not all FDs hold
  - We can ensure all FDs hold, but have possibility of redundancy