SECTION 3

CSE 444

January 20th, 2011

Overview

- SQL un-nesting example
- E/R Diagrams
  - Translation from English or table schema to E/R
  - Translation E/R to table schema
- Functional Dependencies
- Boyce Codd Normal Form
  - Examples

Find persons that frequent some bar that serves some drink they like

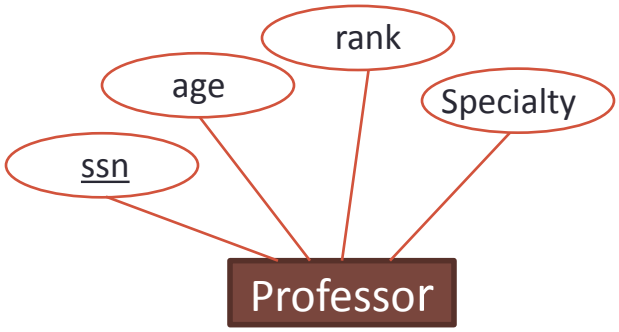
```
-- How can we unnest this?
select F.person
from Frequents F
where exists (select *
                  from Serves S
                  where S.bar = F.bar and
                        exists (select *
                                from Likes L
                                where L.person = F.person
                                      and S.drink = L.drink))
```

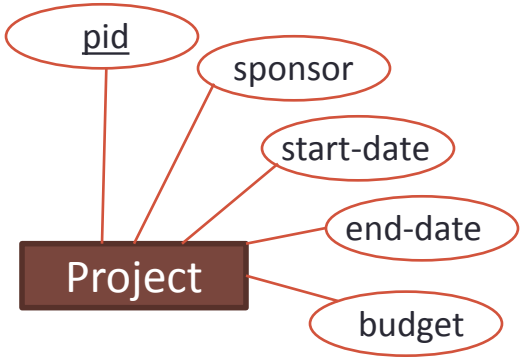
Find persons that frequent some bar that serves some drink they like

```
-- unnested version
select F.person
from Frequents F, Likes L, Serves S
where F.person = L.person and
        F.bar = S.bar and
        L.drink = S.drink
```

English to E/R Diagram

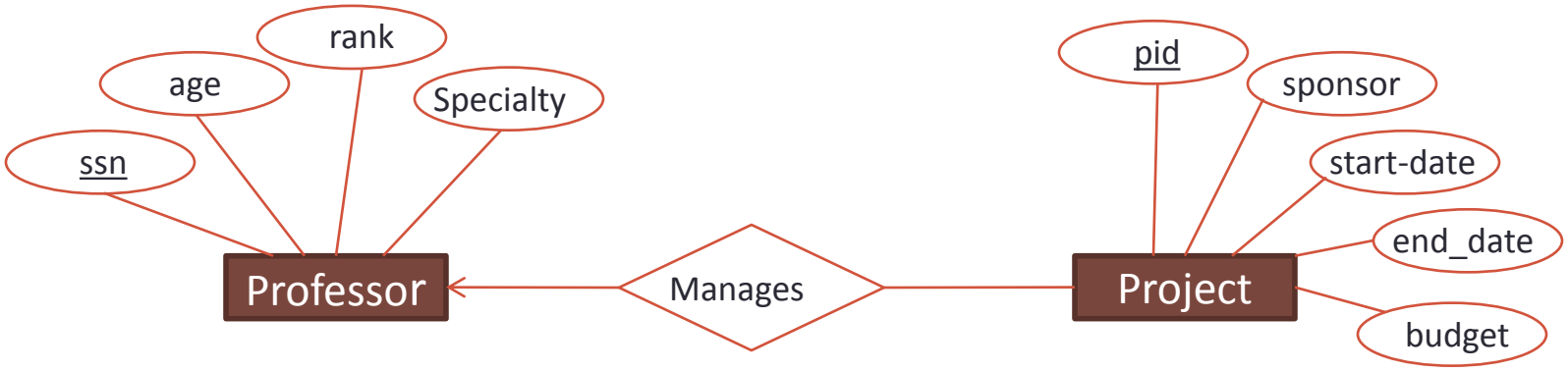
- Attributes
  - Professors have: ssn, age, rank, specialty
  - Projects have: pid, sponsor, start-date, end-date, budget





English to E/R Diagram

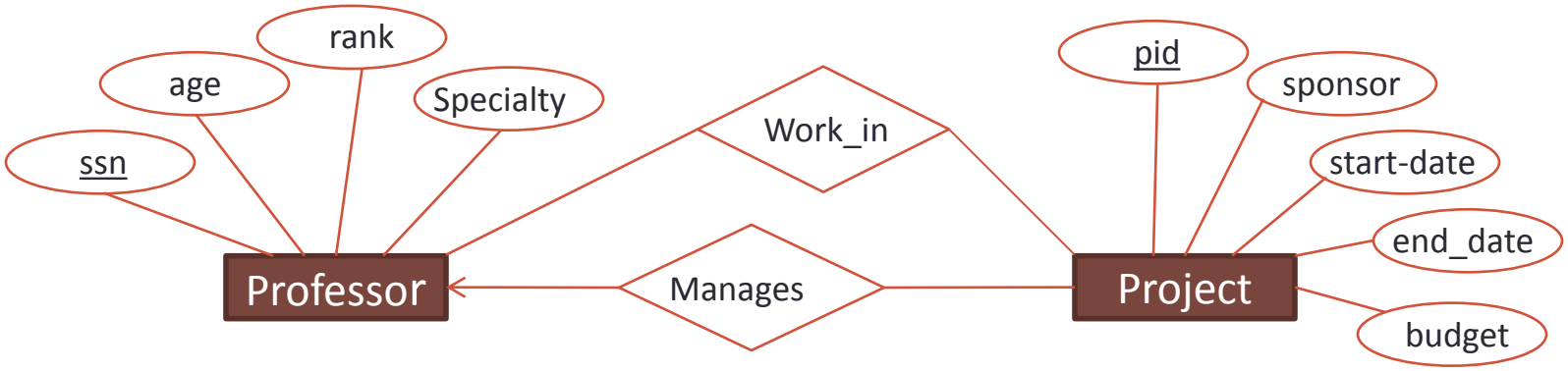
- Each project is managed by one professor (principal investigator)
- Professor can manage multiple projects



Example courtesy: Database Management Systems, 3rd E, R. Ramakrishnan and J. Gehrke

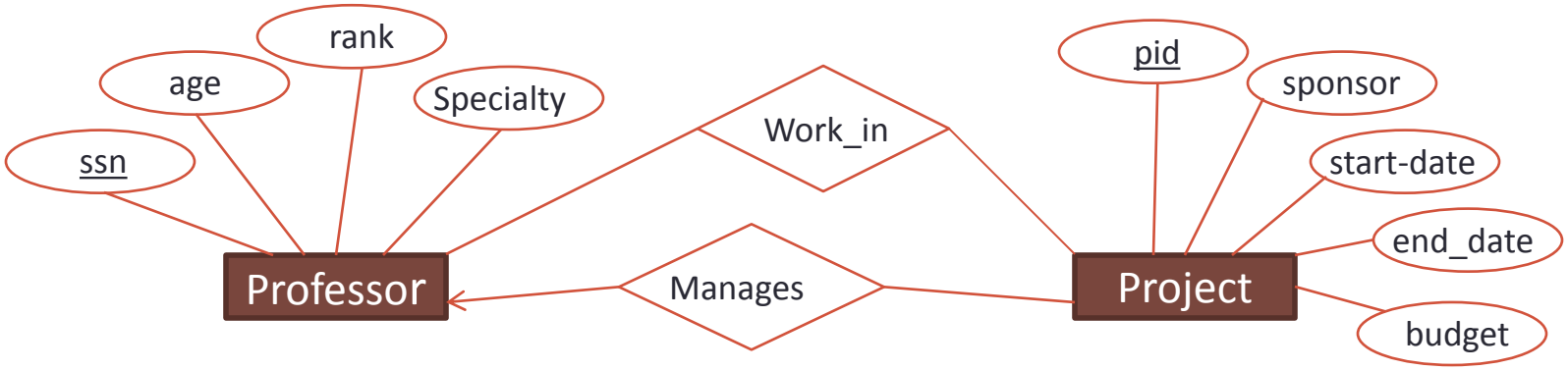
English to E/R Diagram

- Each project is worked on by one or more professors
- Professors can work on multiple projects



Example courtesy: Database Management Systems, 3rd E, R. Ramakrishnan and J. Gehrke

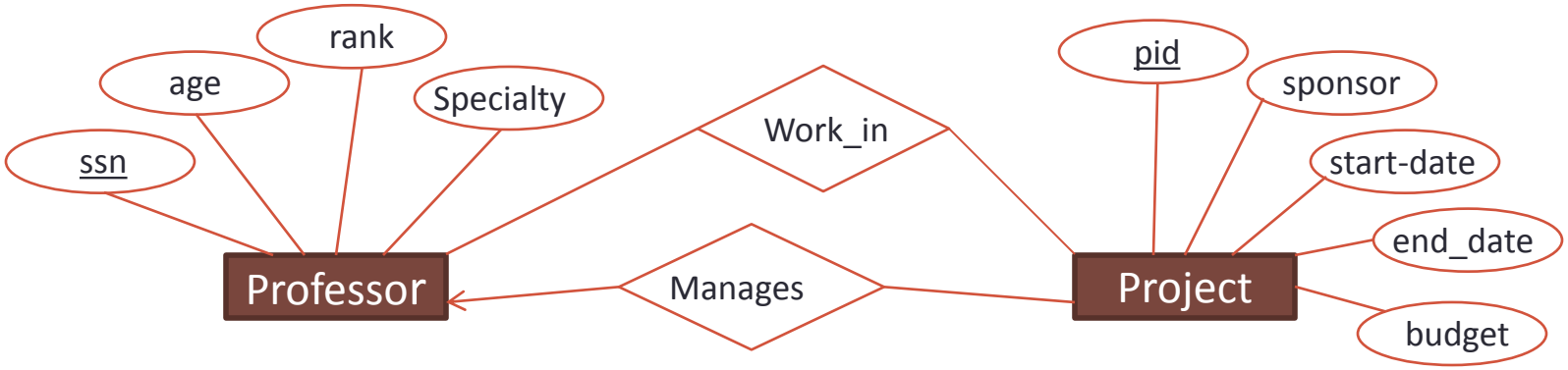
Convert to tables



- Professor(ssn, age, rank, specialty)
- Project(pid, sponsor, start_date, end_date, budget)
- Work_in(ssn, pid)
- Manages(ssn, pid)

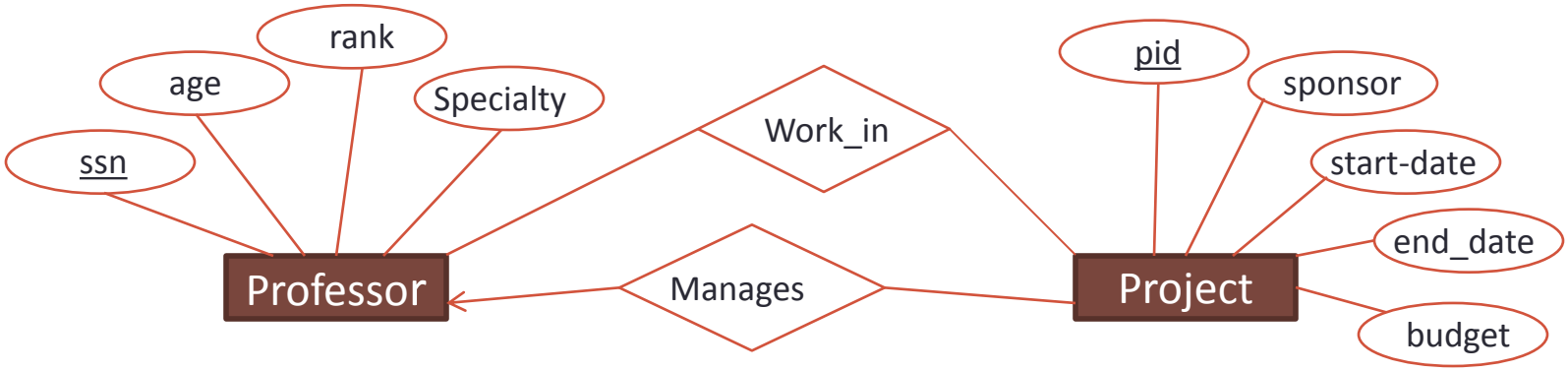
Example courtesy: Database Management Systems, 3rd E, R. Ramakrishnan and J. Gehrke

Convert to tables



- Professor(ssn, age, rank, specialty)
- Project(pid, sponsor, start_date, end_date, budget, ssn)
- Work_in(ssn, pid)

Convert to tables




```
CREATE TABLE Professor (
        ssn INT PRIMARY KEY,
        age INT,
        urank VARCHAR(30),
        specialty VARCHAR(30)
);
```

```
CREATE TABLE Work_In (
        ssn INT REFERENCES Professor(ssn),
        pid INT REFERENCES Project(pid),
        PRIMARY KEY (ssn, pid)
);
```

```
CREATE TABLE Project (
        pid INT PRIMARY KEY,
        sponser INT,
        start_date DATE,
        end_date DATE,
        budget FLOAT,
        ssn INT REFERENCES Professor(ssn)
);
```

- Professor(ssn, age, rank, specialty)
- Project(pid, sponsor, start_date, end_date, budget, ssn)
- Work_in(ssn, pid)

Data Anomalies

- Redundancy is Bad, why?

- Redundancy
- Update
- Delete

Functional Dependencies

R	A	B	C	D	E	F
	a1	b1	c1	d1	e1	f1
	a1	b1	c2	d1	e2	f3
	a2	b1	c2	d3	e2	f3
	a3	b2	c3	d4	e3	f2
	a2	b1	c3	d3	e4	f4
	a4	b1	c1	d5	e1	f1

- Dependencies for this relation:
  - $A \rightarrow B$
  - $A \rightarrow D$
  - $B,C \rightarrow E,F$
- Do they all hold in this instance of the relation R?

- How would you go by finding these in an unknown table?
- Functional dependencies are specified by the database programmer based on the intended meaning of the attributes.

Keys

- Keys
  - Superkey
  - Key

Garcia-Molina, problem 3.3.2 (i)

Consider a relation S(A,B,C,D) with FDs
A → B, B → C, and B → D.

a. Give the nontrivial FDs that follow from the given FDs. Restrict to 1 attr on right side.
b. What are all the keys of S?
c. What are the superkeys that aren't keys?

a) A -> C, A -> D
b) Only {A}
c) Any set of two or more keys that includes A

Boyce-Codd Normal Form (BCNF)

- What is it?

- Why do it?

BCNF Decomposition Algorithm

BCNF_Decompose(R)

find X s.t.: X ≠X^+ ≠ [all attributes]

if (not found) **then** "R is in BCNF"

let Y = X^+ - X
let Z = [all attributes] - X^+
decompose R into R1(X ∪ Y) and R2(X ∪ Z)
continue to decompose recursively R1 and R2

A table R(A,B,C,D,E) : Example 1

Consider the following FDs:

- CD → E BAD
- D → B BAD
- A → CD

Which one are the bad dependences ?

CD+ = BCDE

CD is not a superkey

D+ = BD

D is not a superkey

A+ = ABCDE

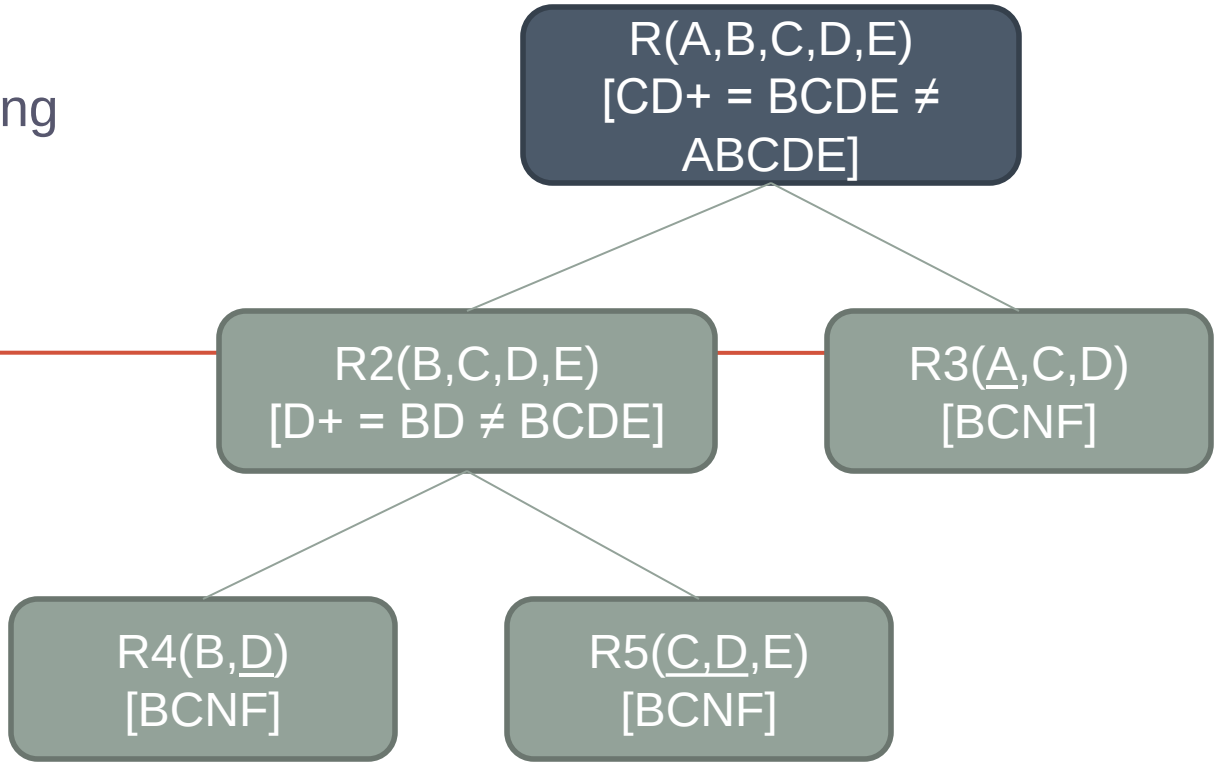
A is a superkey

NOTE: A SET OF ATTRIBUTES X IS A SUPERKEY IF X+ = ABCDE

A table R(A,B,C,D,E) : Example 1

Consider the following FDs:

- CD → E BAD
- D → B BAD
- A → CD



NOTE: A SET OF ATTRIBUTES X IS A SUPERKEY IF X+ = ABCDE

2 more BCNF decompositions

R(A, B, C, D)

$C \rightarrow D$, $C \rightarrow A$, $B \rightarrow C$

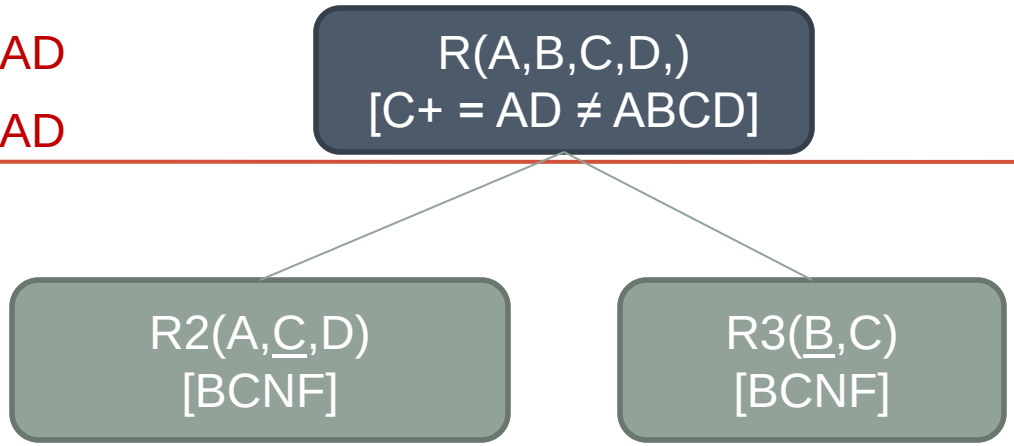
S(A, B, C, D, E)

$AB \rightarrow C$, $DE \rightarrow C$, $B \rightarrow D$

A table R(A,B,C,D) : Example 2

Consider the following FDs:

- C → D, C+ = AD BAD
- C → A, C+ = AD BAD
- B → C, B+ = ABCD



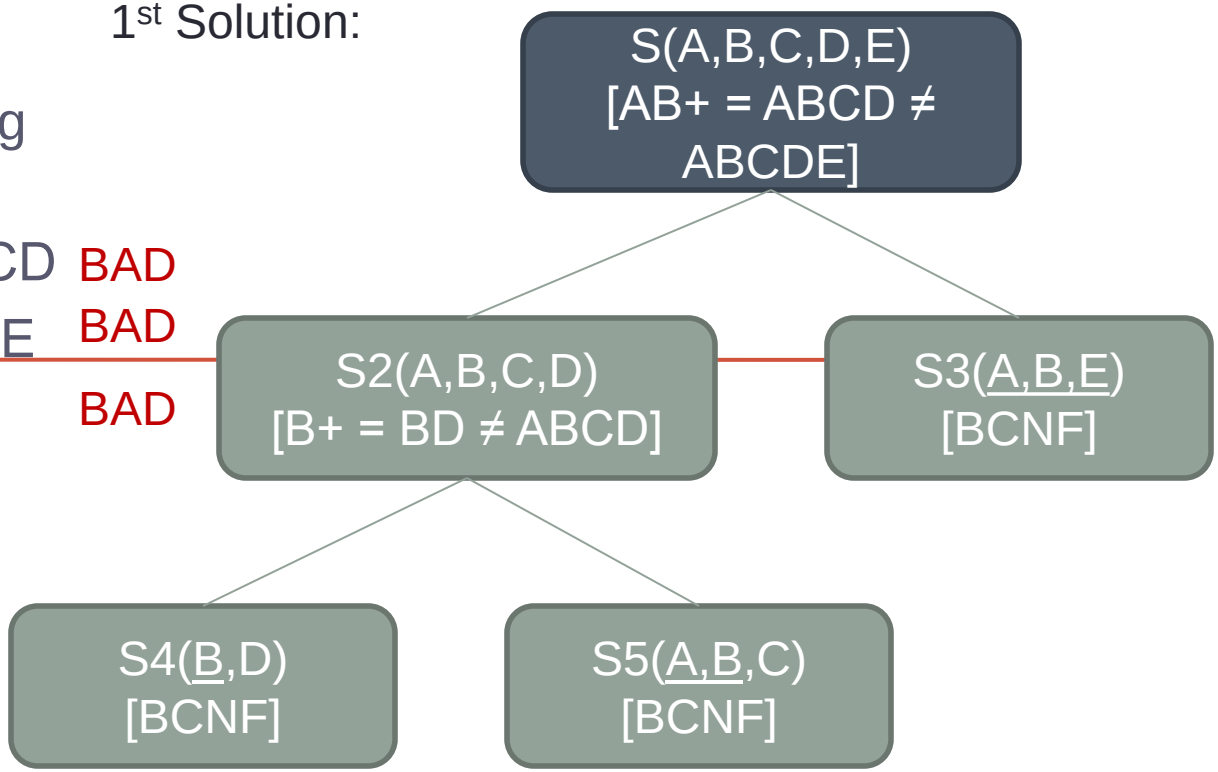
NOTE: A SET OF ATTRIBUTES X IS A SUPERKEY IF X+ = ABCD

A table S(A,B,C,D,E) : Example 3

1st Solution:

Consider the following FDs:

- AB → C, AB+ = ABCD BAD
- DE → C, DE+ = CDE BAD
- B → D, B+ = BD BAD



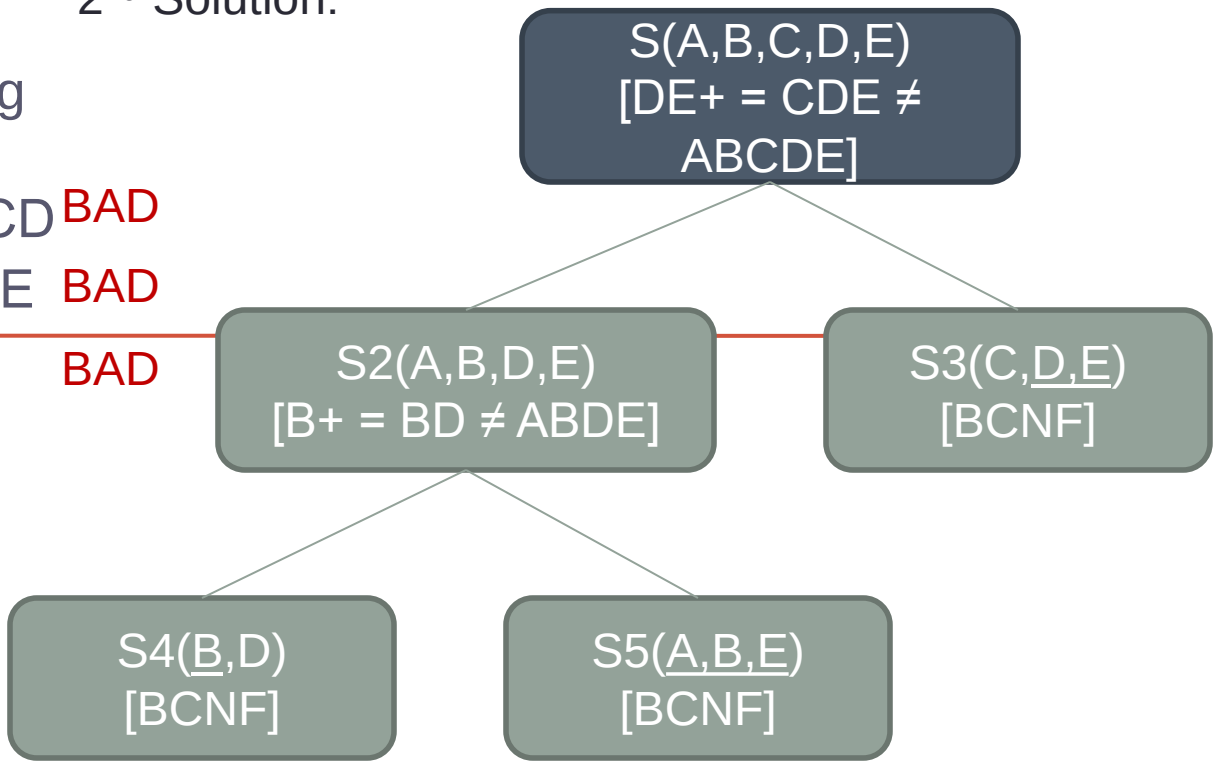
NOTE: A SET OF ATTRIBUTES X IS A SUPERKEY IF X+ = ABCDE

A table S(A,B,C,D,E) : Example 3

2nd Solution:

Consider the following FDs:

- AB → C, AB+ = ABCD BAD
- DE → C, DE+ = CDE BAD
- B → D, B+ = BD BAD



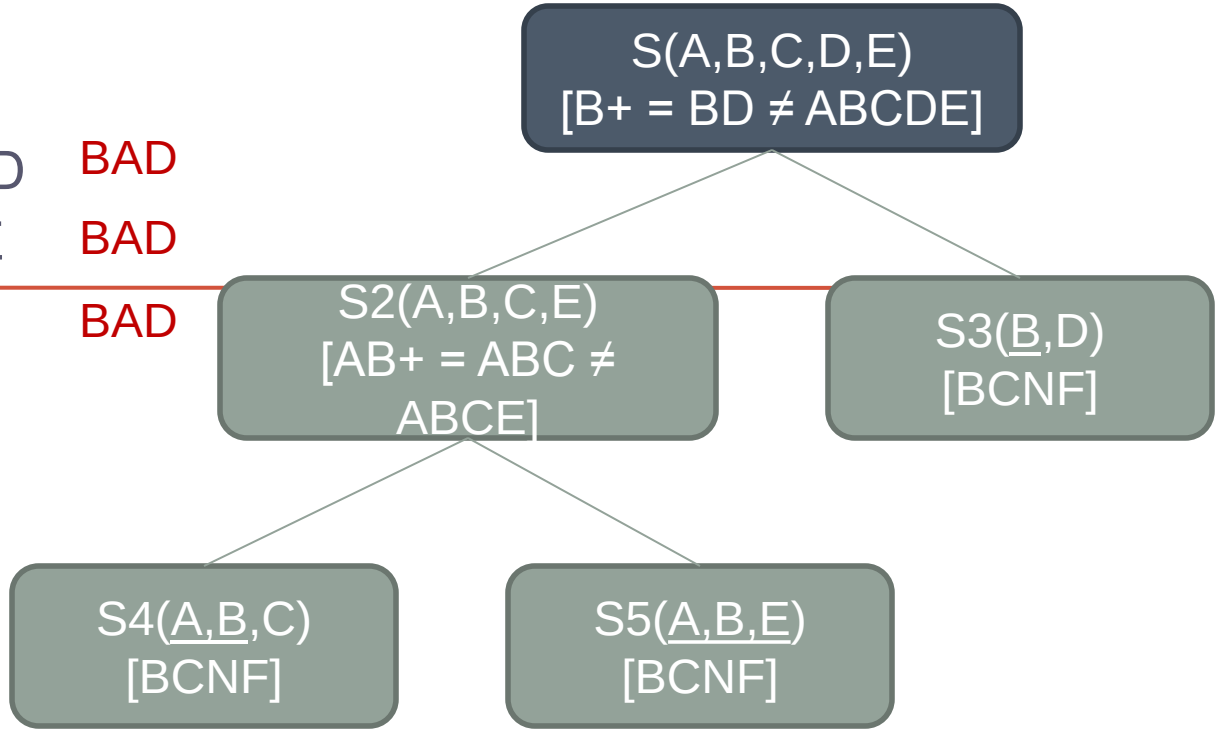
NOTE: A SET OF ATTRIBUTES X IS A SUPERKEY IF X+ = ABCDE

A table S(A,B,C,D,E) : Example 3

3rd Solution:

Consider the following FDs:

- AB → C, AB+ = ABCD BAD
- DE → C, DE+ = CDE BAD
- B → D, B+ = BD BAD



NOTE: A SET OF ATTRIBUTES X IS A SUPERKEY IF X+ = ABCDE

Notes

- Project 1 due tomorrow January 21st, 5pm
- Office hours 10:30-12 in 006

- Homework 1 is out, due January 28th

- Course wiki is up
  - https://cubist.cs.washington.edu/wiki/index.php/CSE444
  - Good source of practice problems and explanations