1 Conceptual Design, Constraints, Views

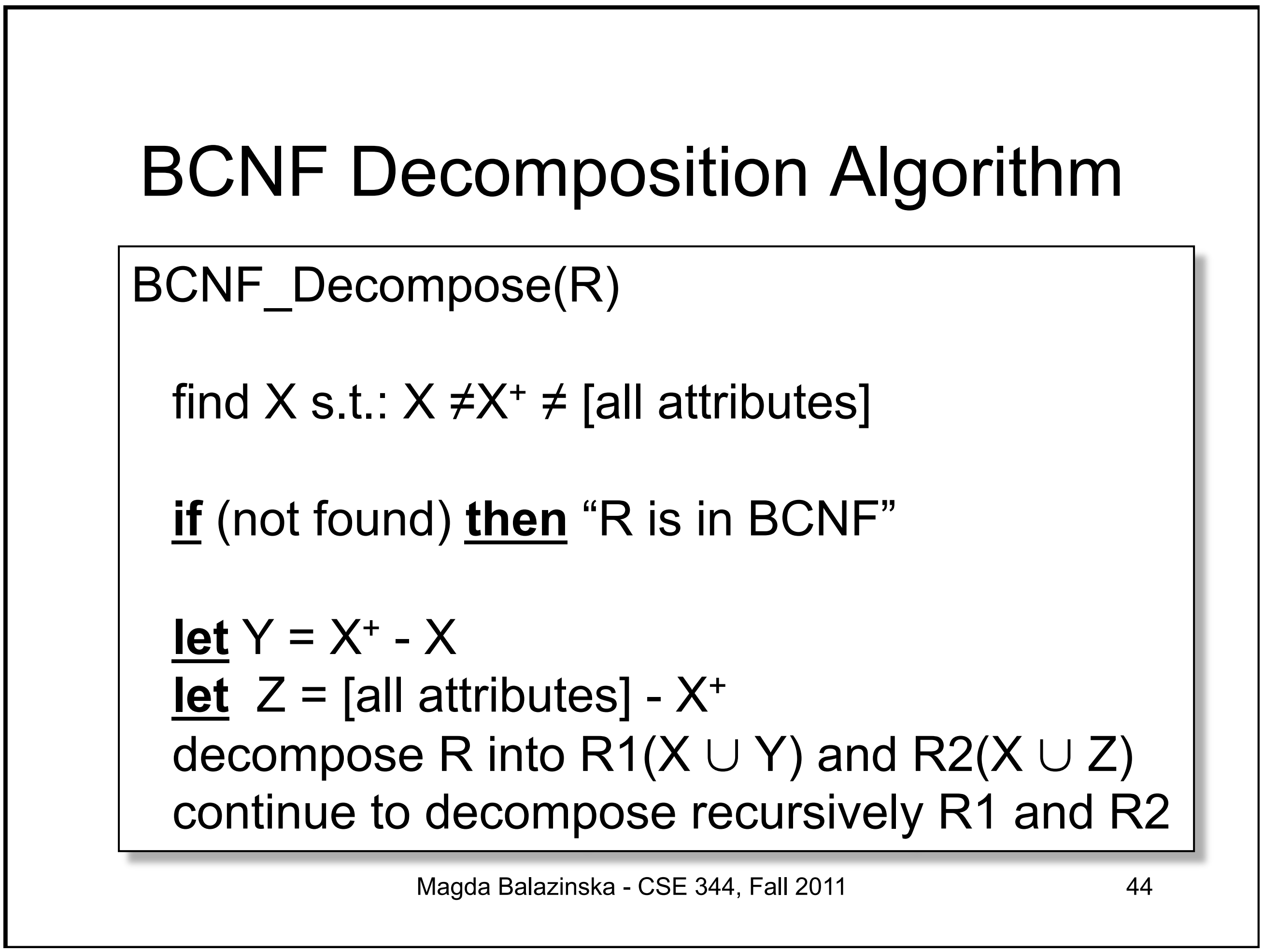
1. (25 points)

   (a) (10 points) Consider a relation $R(A, B, C, D, E)$ that satisfies the following functional dependencies:

$$A \rightarrow B$$
$$CD \rightarrow E$$

   Decompose the schema in BCNF. Show all your steps. A relation R is in BCNF if and only if: whenever there is a nontrivial functional dependency $A_1, A_2, ..., A_n \rightarrow B_1, B_2, ..., B_n$ for R, then $\{A_1, A_2, ..., A_n\}$ is a superkey for R.

   Answer (Show the steps leading to the BCNF decomposition and show the keys in the decomposed relations):



Solution: Straightforward application of BCNF Decomp Algo from lecture 16, slide 44.

Iteration 1: R
A+ = AB Decompose into R1 = AB, R2 = ACDE.
Continue to decompose R2 since R1 is in BCNF form already.

Iteration 2: R2
CD+ = CDE
Decompose R2 into R3 = CDE and R4 = CDA

2 Transactions

(b) (25 points)

Consider a database consisting of a single relation R:

R:

A	B
1	10
2	20

Two transactions run concurrently on this database, resulting in the following schedule:

Line	T1	T2
1	begin;	
2		begin;
3	update R set B = (select sum(B) from R) where A=1;	
4		update R set B = (select sum(B) from R) where A=2;
5	select * from R;	
6		select * from R;
7	insert into r values (3,300);	
8		insert into r values (4,400);
9	select * from R;	
10		select * from R;
11	update R set B = (select sum(B) from R) where A=1;	
12		update R set B = (select sum(B) from R) where A=2;
13	select * from R;	
14		select * from R;
15	commit;	
16		commit;

(a) (5 points) Is this schedule possible in SQL Lite ? If not, then indicate the first line where SQL Lite will change the schedule.

(a) **No: line 4**

Yes ? Or No (and indicate line number) ?

(b) (5 points) Is this schedule possible in SQL Server ? If not, then indicate the first line where SQL Server will change the schedule.

(b) **No: line 4**

Yes ? Or No (and indicate line number) ?

(c) (10 points) Consider running these two transactions in SQL Server, using isolation level SERIALIZABLE. Indicate the result of each of the six `select *` statements, as well as the content of the table after both transactions commit.

Line Number	Result of `select * from r;`
5	
9	
13	
after T1 commits	
6	
10	
14	
after T2 commits	

Solution:

Line Number	Result of `select * from r;`
5	(1,30), (2,20)
9	(1,30), (2,20), (3,300)
13	(1,350), (2,20), (3,300)
after T1 commits	(1,350), (2,20), (3,300)
6	(1,350), (2,670), (3,300)
10	(1,350), (2,670), (3,300), (4,400)
14	(1,350), (2,1720), (3,300), (4,400)
after T2 commits	(1,350), (2,1720), (3,300), (4,400)

(d) (5 points) Is the schedule for these transactions serializable ?

(d) **Yes. T1; T2**

Yes or No ?