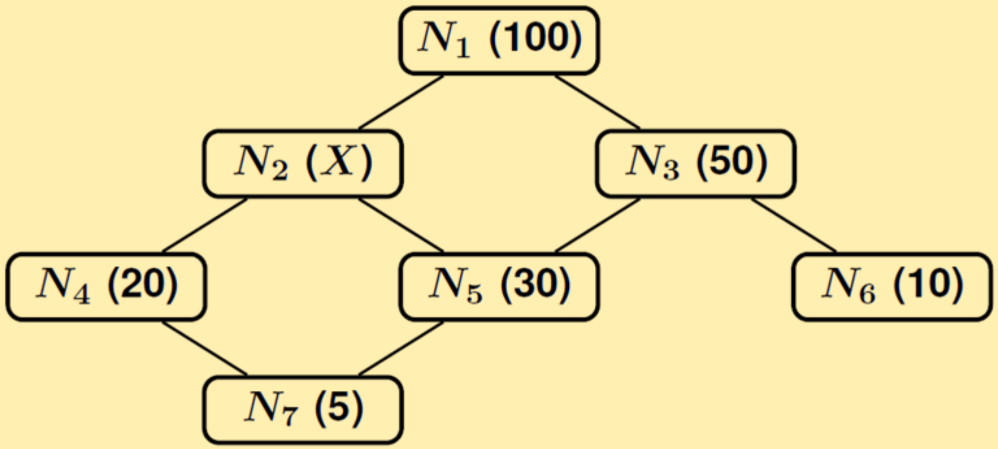
EXERCISE AT HOME

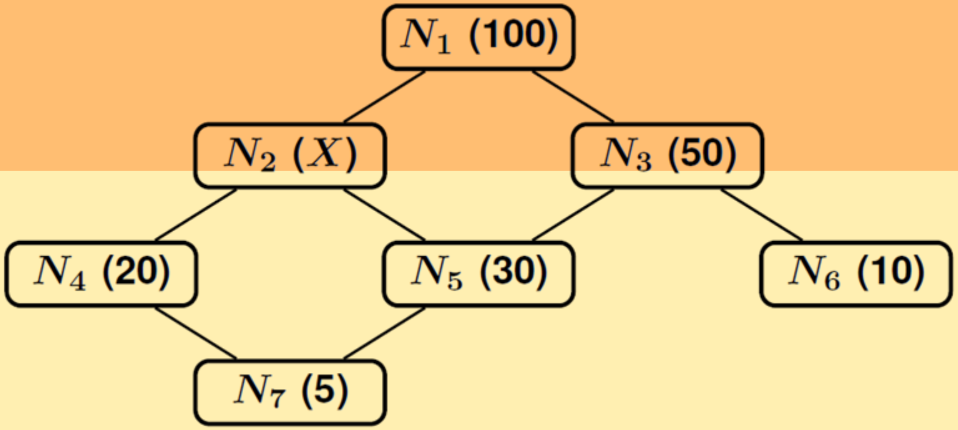
Let us consider the following lattice of possible candidate views to materialize. The numbers associated with the nodes represent the view size, measured in terms of the number of tuples in the view.



(a) **(1 point)** The value X for N_2 is unknown. What is the domain of admissible values for X?

(b) **(5 points)** Select 2 views to materialize, different from N_1, with the greedy algorithm HRU. Determine the various possible results of HRU on the basis of the unknown value X for N_2.

EXERCISE AT HOME

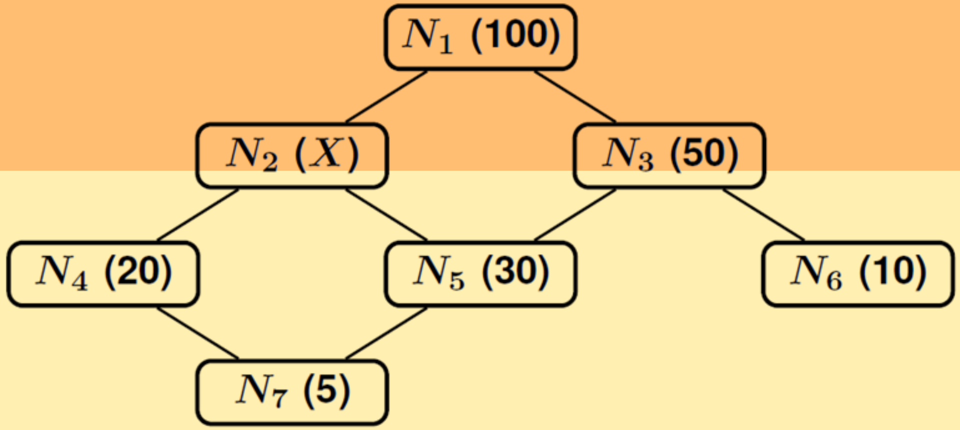


- max{ descendants of N_2 } = 30 ≤ X ≤ 100 = min{ ascendants of N_2 }

View	First Choice		
N_2	(100-X)•4		
N_3	(100-50)•4 = 200		
N_4	(100-20)•2 = 160		
N_5	(100-30)•2 = 140		
N_6	(100-10) = 90		
N_7	(100-5) = 95		

- (100-X)•4 > 200 iff X < 50

EXERCISE AT HOME



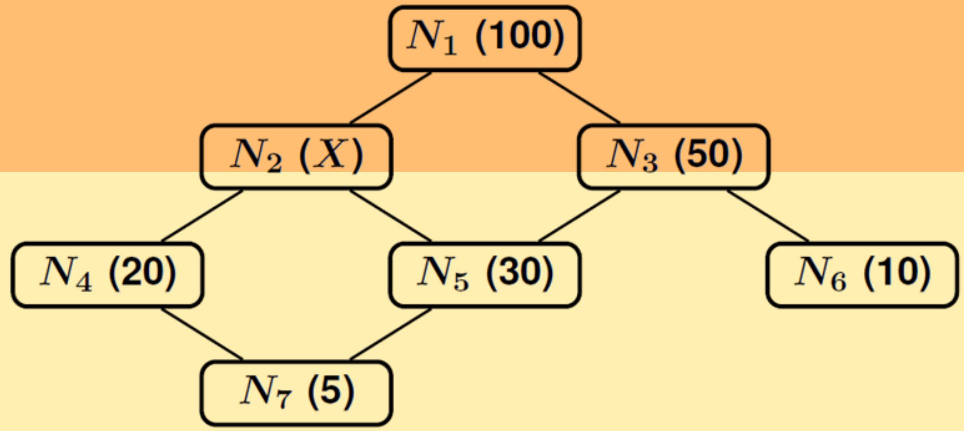
- max{ descendents of N_2 } = $30 \leq X \leq 100$ = min{ ascendents of N_2 }

View	First Choice	Second choice $30 \leq X < 50$	
N_2	$(100-X) \cdot 4$	-	
N_3	$(100-50) \cdot 4 = 200$	$(100-50) \cdot 2 + \cancel{(X-50) \cdot 2} = 100$	
N_4	$(100-20) \cdot 2 = 160$	$(X-20) \cdot 2 \leq 60$	
N_5	$(100-30) \cdot 2 = 140$	$(X-30) \cdot 2 \leq 40$	
N_6	$(100-10) = 90$	$(100-10) = 90$	
N_7	$(100-5) = 95$	$(X-5) \leq 45$	

- $(100-X) \cdot 4 > 200$ iff $X < 50$

$M = \{N_1, N_2, N_3\}$

EXERCISE AT HOME



- max{ descendents of N_2 } = $30 \leq X \leq 100$ = min{ ascendents of N_2 }

View	First Choice	Second choice $30 \leq X < 50$	Second choice $50 \leq X \leq 100$
N_2	$(100-X) \cdot 4$	-	$(100-X) \cdot 2 + \cancel{(50-X) \cdot 2 \leq}$ 100
N_3	$(100-50) \cdot 4 = 200$	$(100-50) \cdot 2 + \cancel{(X-50) \cdot 2}$ = 100	-
N_4	$(100-20) \cdot 2 = 160$	$(X-20) \cdot 2 \leq 60$	$(100-20)+(50-20) = 110$
N_5	$(100-30) \cdot 2 = 140$	$(X-30) \cdot 2 \leq 40$	$(50-30) \cdot 2 = 40$
N_6	$(100-10) = 90$	$(100-10) = 90$	$(50-10) = 40$
N_7	$(100-5) = 95$	$(X-5) \leq 45$	$(50-5) = 45$

- $(100-X) \cdot 4 > 200$ iff $X < 50$

$M = \{N_1, N_2, N_3\}$

$M = \{N_1, N_3, N_4\}$

RELATIONAL DBMS EXTENSIONS FOR DW

- SQL extensions
- Index and storage structures
- Star query physical plans
- Materialized views

NEXT LESSON

- Optimization techniques for star queries with grouping and aggregations

TODAY: functional dependencies and their usage in query optimization

Functional dependencies

Given a relation schema R(T) and $X, Y \subseteq T$, a **functional dependency** (FD) is a constraint on R of the form $\mathbf{X} \rightarrow \mathbf{Y}$, i.e. **X functionally determines** Y or Y **is determined by X**, if

$$\forall\, r \text{ valid instance of } R.$$
$$\forall\, t1, t2 \in r.\ \text{if } t1[X] = t2[X] \text{ then } t1[Y] = t2[Y]$$

Convention: ..., X, Y, Z represent sets of attributes; A, B, C, ... represent single attributes; a set of attributes {A, B, C} is represented just as ABC.

FUNCTIONAL DEPENDENCIES: EXAMPLE

StudentsExams(<u>StudCode</u>, Name, City, Region, BirthYear, <u>Subject</u>, Grade)

StudCode	Name	City	Region	BirthYear	Subject	Grade
1234567	N1	C1	R1	1995	DB	30
1234567	N1	C1	R1	1995	SE	28
1234568	N2	C2	R2	1994	DB	30
1234568	N2	C2	R2	1994	SE	26

StudCode → Name City Region BirthYear ?

City → Region ?

Subject → Grade ? **NO**

StudCode Subject → Grade ?

Subject → Subject ? **trivial**

X → { } ? **trivial**

{ } → University ? **YES, if University is constant**

REASONING ABOUT FDs: LOGICAL IMPLICATION

Notation:

- R <T, F> is a relational schema with **attributes** T and a **set** of functional dependencies F. Example: F = { X->Y, Y->Z }
- A FD $\in$ F is a constraint on relational instances r of R <T, F>
  - r is a valid instance if $\forall$ t1, t2 $\in$ r. if t1[X] = t2[X] then t1[Y] = t2[Y]

Usage: defined by the designer, enforced by the DBMS. In practice, only the functional dependency **K** $\rightarrow$ **T** are enforced, when **K** is a key.

REASONING ABOUT FDs: LOGICAL IMPLICATION

Notation:

- R <T, F> is a relational schema with **attributes** T and a **set** of functional dependencies F. Example: F = { X->Y, Y->Z }
- A FD $\in$ F is a constraint on relational instances

Given a set F of FDs, other FDs will generally be 'implied' by this set in the following sense:

Definition Given a schema R <T, F>, we say that F *implies* $X \rightarrow Y$, if every instance r of R that satisfies F also satisfies $X \rightarrow Y$.

Example: { X->Y, Y->Z } implies X –> Z ?

INFERENCE RULES

To test if a FD implied by a set F, a set of inference rules can be used with the property of being **sound** and **complete** (F implies FD **iff** F |- FD)

Armstrong axioms:

- **If $Y \subseteq X$, then F |- $X \rightarrow Y$** (reflexivity **R**)
- **If F |- $X \rightarrow Y$ and $Z \subseteq T$, then F |- $XZ \rightarrow YZ$** (augmentation **A**)
- **If F |- $X \rightarrow Y$ and F |- $Y \rightarrow Z$, then F |- $X \rightarrow Z$** (transitivity **T**)

Exercises:

1. { X->Y, X->Z } |- X -> YZ (union **U**)
2. if $Z \subseteq Y$ then: X->Y |- X -> Z (decomposition **D**)
3. F |- X -> $A_1, \ldots, A_n$ iff F |- X -> A_1 and ... F |- X -> A_n

CLOSURE OF A SET OF FDs

The FDs **implied** by F (the **closure** of **F**) are defined as:

$$\mathbf{F^+ = \{ X \rightarrow Y \mid F \mid\!- X \rightarrow Y \}}$$

Implication problem: to test whether a FD $X \rightarrow Y \in F^+$ (without computing the whole closure of F)

Exercises:

3. $F \mid\!- X \to A_1, \ldots, A_n$ iff $F \mid\!- X \to A_1$ and ... $F \mid\!- X \to A_n$

CLOSURE OF A SET OF ATTRIBUTES

Definition Given a scheme R <T, F>, and $X \subseteq T$, the closure of X is

$$X^+ = \{ A \in T \mid F \vdash X \rightarrow A \}$$

A procedure to solve the implication problem without computing the whole closure of F follows from the following result.

Theorem $F \vdash X \rightarrow Y$ iff $Y \subseteq X^+$

Proof. *We proved it as exercise (3)*

SLOW CLOSURE

A simple algorithm to compute X^+ is the following

- faster algorithm exist

Algorithm SLOW CLOSURE

input $R\langle T, F\rangle$, $X \subseteq T$

output X^+

begin

$X^+ = X$

while (changes to X^+) **do**

for each $W \rightarrow V$ **in** F **with** $W \subseteq X^+$ **and** $V \not\subseteq X^+$

do $X^+ = X^+ \cup V$

end

EXAMPLE

$F = \{DB \rightarrow E, B \rightarrow C, A \rightarrow B\}$. Is $AD \rightarrow E$ in F^+ ?

$X^+ = AD$

$X^+ = ADB$

$X^+ = ADBE$ ⟵ We can stop here

$X^+ = ADBEC$

FDs AND (SUPER)KEYS

Recall our **assumption**: tables are **sets** of tuples

Definition Given a scheme R <T, F>, we say that $W \subseteq T$ is a **key** of R if

$W \rightarrow T \in F^+$ (W is a **superkey**) **and**

$\forall\, V \subset W.\ V \rightarrow T \notin F^+$ (if $V \subset W$, V **is not a superkey**)

EXERCISE: FD's lift over cartesian product

If $X \rightarrow Y$ holds in R, is this still the case in R x S ?

Hypothesis. If $\forall$ t1, t2 $\in$ R. if t1[X] = t2[X] then t1[Y] = t2[Y]

Conclusion. $\forall$ w1, w2 $\in$ R x S. if w1[X] = w2[X] then w1[Y] = w2[Y]

$$w1 = t1 \circ s1 \Rightarrow w1[X] = t1[X]$$

If X is a key for R and Y a key for S, then is XY a key for R x S ?

- XY is a superkey:

  X -> R1, …, Rn, Y -> S1, …, Sm |- XY -> R1, …, Rn, S1, …, Sm

  iff {R1, …, Rn, S1, …, Sm} $\subseteq$ $\{X,Y\}^+$

  $\{X,Y\}^+$ = {X,Y,R1,…,Rn,S1,…,Sm}

- no subset of XY is a superkey: exercise at home

EXERCISE: FD's lift over selections (and then over joins)

If $X \rightarrow Y$ holds in R, is this still the case in $\sigma_C(R)$?

Hypothesis. If $\forall$ t1, t2 $\in$ R. if t1[X] = t2[X] then t1[Y] = t2[Y]

Conclusion. $\forall$ w1, w2 $\in \sigma_C(R)$. if w1[X] = w2[X] then w1[Y] = w2[Y]

w1 = t1 for some t1 $\in$ R $\Rightarrow$ w1[X] = t1[X]

If X is a key for R, then is X a (super)key for $\sigma_C(R)$?

- X is a superkey:

  X -> R1, …, Rn |- X -> R1, …, Rn

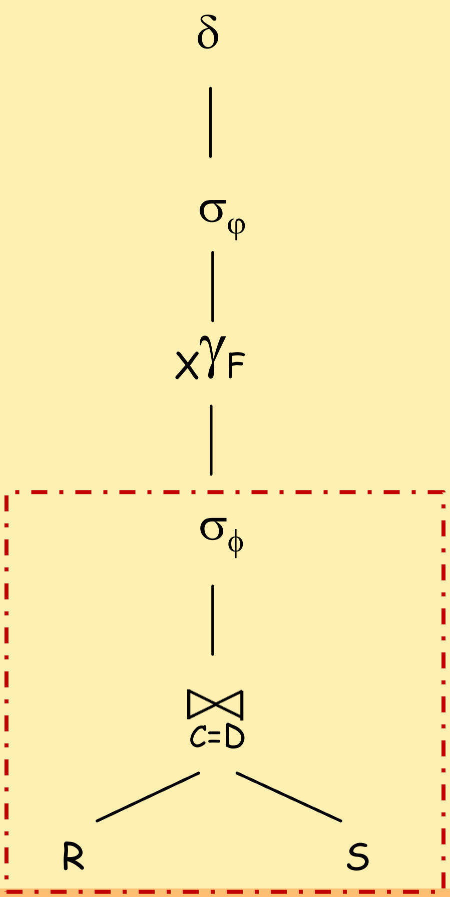
- no subset of X is a superkey: FALSE

  for R(<u>A</u>, <u>B</u>), AB is a key

  for $\sigma_{B=1}(R)$, A is a key

ASSUMPTIONS

- The tables do not have null values, and **have primary keys**:
  - A key constraint uniquely identifies each record in a table.
  - Tables are then **sets** of tuples
- Table in FROM clause have no attribute with the same name
- Queries are a single SELECT with possibly GROUP BY and HAVING but without subselect and ORDER BY clauses.
- Since superkeys are lifted after join and restriction, then $\sigma_C(R \bowtie S)$ is a **set** of tuples

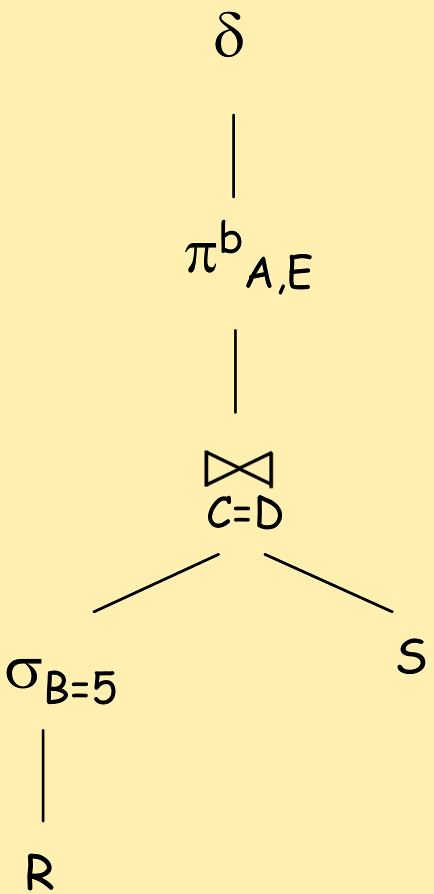


EXAMPLE

R(<u>A</u>,B,C)   S(<u>D</u>,E)

F = { A->BC, D->E, … what else? … }

F = { A->BC, D->E, {}->B, C->D, D->C }



DERIVING FUNCTIONAL DEPENDENCIES IN SQL RESULTS

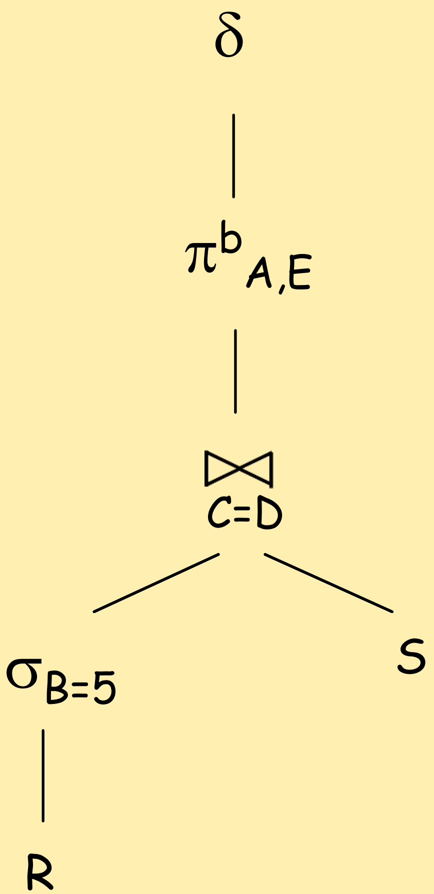
Which functional dependencies hold in the result of a query

SELECT * FROM-WHERE when all the tables in **FROM** have a key ?

1. Let F the initial set of FDs where their determinants are the keys of every table used in the query.
2. Let C the WHERE condition. If a conjunct of C is a predicate $Ai = \mathbf{c}$, then F is extended with the functional dependency $\{\} \rightarrow Ai$.
3. If a conjunct of C is a predicate $Aj = Ak$, e.g. a join condition, F is extended with the functional dependencies $Aj \rightarrow Ak$ and $Ak \rightarrow Aj$.

EXAMPLE

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



F = { A->BC, D->E, … what else? … }

F = { A->BC, D->E, {}->B, C->D, D->C }

F |- AE-> AD ?

$A \in \{A,E\}^+$ $D \in \{A,E\}^+$?

Apply SLOW CLOSURE

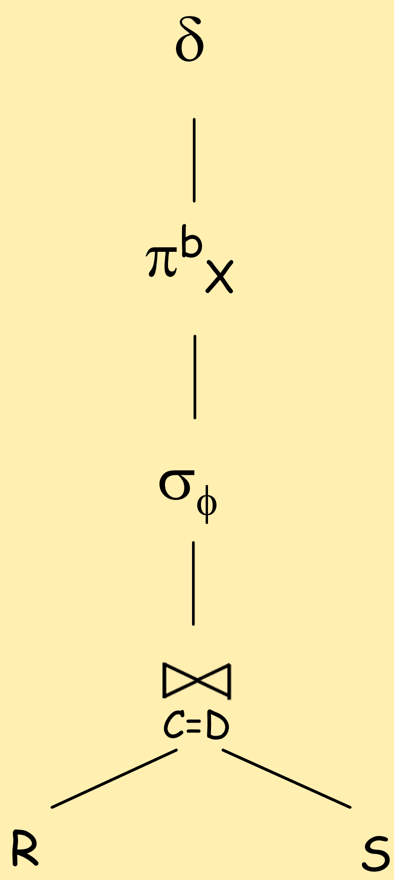
F |- AE -> AD   YES

In general: if X -> Y and Y is a super-key then X is a super-key

Since AD is a superkey after the join, AE is a superkey!

Is δ needed ???   δ is **NOT** needed

GENERALIZATION OF THE EXAMPLE




```
SELECT  DISTINCT X
FROM  R, S
WHERE C=D AND ϕ
```

When **DISTINCT** is useless ?

X is a (super)key in $\sigma_{\phi}(R \bowtie S)$

e.g., if X determines the keys of R and S

DERIVING FUNCTIONAL DEPENDENCIES IN SQL RESULTS

An algorithm to compute the closure of an attribute set X in $\sigma_{\phi}(R \bowtie S)$, which works **directly on SQL** without explicitly using functional dependencies.

1. Let $X^+ = X$
2. Add to X^+ all attributes *Ai* such that *Ai* = c is a conjunct of the selection.
3. Repeat until X^+ is changed
   a) Add to X^+ all attributes *Aj* such that predicate *Aj* = *Ak* is a conjunct of the selection, **and** *Ak* $\in X^+$.
   b) Add to X+ all attributes of a table if X+ contains a key for that table.

$R(\underline{A},B,C) \quad S(\underline{D},E)$

SELECT ~~DISTINCT~~ A, E
FROM R, S
WHERE C=D AND B=5

$\{A, E\}^+ = \{A, E, B, C, D\}$