Projection	π
Selection	σ
Join	$\bowtie$
Union	$\cup$
Difference	$-$
Attribute renaming	δ

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

r ∪ s

A	B
α	1
α	2
β	1
β	3

r - s

A	B
α	1
β	1

R

A	B
0	1
0	2
5	3

S

B	C
1	2
1	3
2	2

R ⋈S =

A	B	C
0	1	2
0	1	3
0	2	2

Note that <5, 3> is "lost".

If S is

C
2
3

then R ⋈S =

A	B	C
0	1	2
0	1	3
0	2	2
0	2	3
5	3	2
5	3	3

Let r and s be relations on schemas R and S respectively. Then, $r \bowtie s$ is a relation with schema $att(R) \cup att(S)$ obtained as follows:

- Consider each pair of tuples t_r from r and t_s from s.
- If t_r and t_s have the same value on each of the attributes in $att(R) \cap att(S)$, add a tuple t to the result, where
  - t has the same value as t_r on r
  - t has the same value as t_s on s

Note: *if $R \cap S$ is empty, the join consists of all combinations of tuples from R and S, i.e. their cross-product*

SQL
↓
Relational Algebra
↓
Query Rewriting
↓
Query Execution Plan
↓
Execution

Titles of currently playing movies:

$\pi_{TITLE}(schedule)$

Titles of movies by Berto:

$\pi_{TITLE}(\sigma_{DIR=BERTO}(movie))$

4. Find the pairs of actors acting together in some movie

Titles and directors of currently playing movies:

$\pi_{TITLE,\ DIR}(movie \bowtie schedule)$

$\pi_{actor1,\ actor2}(\delta_{actor \to actor1}(movie) \bowtie \delta_{actor \to actor2}(movie))$

5. Find the actors playing in every movie by Berto

$R \div S$: {a | <a, b> ∈R for every b∈S}

$\pi_{actor}(movie)$ –

$\pi_{actor}[(\pi_{actor}(movie) \bowtie \pi_{title}(\sigma_{dir=BERTO}(movie))) - \pi_{actor,title}(movie)]$

actor | title by Berto | actor acts in title

actors for which there is a movie by Berto in which they do not act

e.g.:

R

A	B
0	α
0	β
1	α
1	β
1	γ
2	α

S

B
α
β

R ÷ S

A
0
1

"Local" FDs

if F is a set of FD's over R and X⊆ attributes(R) then $\pi_X(F^+) = \{V \to W \in F^+ \mid V \subseteq X \text{ and } W \subseteq X\}$

Basic Correspondence

Algebra Operation		Calculus Operator
π	⟷	$\exists$
σ	⟷	t(A) comp c
$\cup$	⟷	$\vee$
$\bowtie$	⟷	$\wedge$
$-$	⟷	$\neg$
$\div$	⟷	$\forall$

$R \bowtie R$	⟶	R	
$R - R$	⟶	$\emptyset$	
$\emptyset \bowtie R$	⟶	$\emptyset$	
$\sigma_{A=a}(R \bowtie S)$	⟶	$\sigma_{A=a}(R) \bowtie S$	"pushing selection"
$\pi_{AB}(R \bowtie S)$	⟶	$R \bowtie \pi_B(S)$	"column elimination"
$\pi_A(\pi_{AB}(T))$	⟶	$\pi_A(T)$	"cascading projections"

$\pi_{drinker}(\sigma_{beer = Bass}(frequents \bowtie serves))$

↓

$\pi_{drinker}(frequents \bowtie \sigma_{beer = Bass}(serves))$

↓

$\pi_{drinker}(frequents \bowtie \pi_{bar}(\sigma_{beer = Bass}(serves)))$

R: ABC

SQL conjunctive query:

```
select t1.A, t2.B, t3.C
from R t1, R t2, R t3
where t2.A = t3.A and t1.B = 5 and
      t2.B = t3.B and t2.C = t1.B
```

Pattern:

R	A	B	C
t1	a	5	--
t2	a_1	5	--
t3	a_1	5	c

answer	A	B	C
	a	5	c

Minimized pattern:

R	A	B	C
t1	a	5	--
t3	a_1	5	c

answer	A	B	C
	a	5	c

Minimized SQL query:

```
select t1.A, 5 as B, t3.C
from R t1, R t3
where t1.B = 5 and t3.B = 5
```

Statements about valid data

- Keys
  "SSN uniquely determines all attributes of *employee*"
- Referential integrity
  "Every student is a person"
- Functional dependencies: extension of keys
  "Each employee works in no more than one department"
  ˀAME → DEPARTMENT

ˀ dependencies:

ck data integrity

ry optimization

ema design → "normal forms"

Precedence graph P(S) (S is schedule)

Nodes: transactions in S

Arcs: Ti → Tj whenever

- $p_i(A)$, $q_j(A)$ are actions in S
- $p_i(A) <_S q_j(A)$
- at least one of p_i, q_j is a write

More concise algorithm

Input: R, decomposition $\{R_1, \ldots, R_k\}$, set F of FDs

1. construct a pattern using variable *a* for each attribute A:

   for each R_i the pattern has one row with variable *a* for each attribute A of R_i and wildcards everywhere else

3. Chase the pattern with F
4. Output YES iff the resulting pattern has an entire row of variables

Repeat until no change

- For each $X \to A$ in F do
  - For all rows t_1, t_2 in P such that $t_1(X) = t_2(X)$, $t_1(A) \neq t_2(A)$ do
    - if $t_1(A)$, $t_2(A)$ are non-answer variables then replace one by the other everywhere in P
    - if $t_1(A)$ is a non-answer variable and $t_2(A)$ is a wildcard, then replace $t_2(A)$ by $t_1(A)$ everywhere in P
    - If $t_1(A)$, $t_2(A)$ are wildcards, replace both with a new variable
    - if $t_1(A)$ is an answer variable and $t_2(A)$ is a variable or wildcard, then replace $t_2(A)$ by $t_1(A)$ everywhere in P
    - if $t_1(A)$ is constant, $t_2(A)$ is variable or wildcard, then replace $t_2(A)$ by $t_1(A)$ everywhere in P
    - if $t_1(A)$ is constant, $t_2(A)$ is constant then STOP and output $\emptyset$

ˀD}

$F = \{A \to B, B \to C, C \to D, D \to A\}$

Does ρ preserve D → A? (Is it implied by the l

Decomposition: $\overset{*}{A}\overset{*}{B}$ $\overset{*}{B}\overset{*}{C}$ $\overset{*}{C}\overset{*}{D}$

Start with D

$D^+ = DABC$ so D → C is local to CD, add C

$C^+ = CDAB$ so C → B is local to BC, add B

$B^+ = BCDA$ so B → A is local to AB, add A

Boyce-Codd Normal Form (BCNF)

A relation scheme R is in BCNF wrt a set of FD's F over R, iff whenever $X \to A \in F^+$ and $A \notin X$, X is a superkey for R

A schedule is conflict serializable if it is conflict equivalent to some serial schedule.

1. An execution without any interleaving is OK
   serial: T1 ; T2 or T2 ; T1
2. If an execution has the same effect as a serial execution then it is also acceptable

CONCURRENCY CONTROL

ACID: Atomicity, Consistency, Isolation, Durability

HOMEWORK:

We first chase the pattern with the FDs. Applying $D \to C$ yields

R	A	B	C	D
	a_1	–	c	–
	–	b	8	–
	a_1	b	c_1	d
	–	5	c_1	d

answer	B	C	D
	b	c	d

Next, applying $C \to B$ yields

R	A	B	C	D
	a_1	–	c	–
	–	5	8	–
	a_1	5	c_1	d
	–	5	c_1	d

answer	B	C	D
	5	c	d

Using $B \to D$ produces

R	A	B	C	D
	a_1	–	c	–
	–	5	8	d
	a_1	5	c_1	d
	–	5	c_1	d

answer	B	C	D
	5	c	d

Applying once more $D \to C$ yields

R	A	B	C	D
	a_1	–	c	–
	–	5	8	d
	a_1	5	8	d
	–	5	8	d

answer	B	C	D
	5	c	d

This completes the chase. Eliminating duplicate rows produces

R	A	B	C	D
	a_1	–	c	–
	a_1	5	8	d
	–	5	8	d

answer	B	C	D
	5	c	d

This pattern can be minimized by mapping the third row to the second row (- to a_1). The minimized pattern is

R	A	B	C	D
	a_1	–	c	–
	a_1	5	8	d

answer	B	C	D
	5	c	d

(i) (1 *point*) Find all the keys of R.
Since A does not appear on the RHS of any FD in F, every key must contain A. The first two FDs show that AB and AC are superkeys. Since $A^+ = A$, $B^+ = BCDE$ and $C^+ = CD$, AC and AB are both keys. The only other possibility would be AD. However, $AD^+ = AD$. So the only keys are AB and AC.

(ii) (4*points*) Find a BCNF decomposition of R with lossless join with respect to F. (Show how the decomposition is obtained.)
The BCNF decomposition algorithm produces $\{BCE, BD, AB\}$ (see separate pdf file)

(iii) (2 *points*) Is the decomposition obtained in (i) dependency preserving with respect to F ?
The FD $B \to CE$ is local so is preserved. Consider $AB \to CDE$. This is not local so we apply the algorithm. We initially have AB, so B in BCE and BD. Now $B^+ = BCDE$, so we can add CE within BCE and D within BD. We now have $ABCDE$, so $AB \to CDE$ is preserved. Next, consider $AC \to BDE$. We start with AC, so we have C in BCE and A in AB. Since $C^+ = CD$ and $A^+ = A$, nothing can be added, so the FD is not preserved and the decomposition is not dependency preserving.

We first simplify the set of FDs:

(a) single RHS:

$$AB \to C, AB \to D, AB \to E, AC \to B$$
$$AC \to D, AC \to E, B \to C, B \to E, C \to D$$

(b) Eliminate redundant LHS attributes:
Consider $AC \to B$. Since $A^+ = A$ and $C^+ = CD$, neither A nor C are redundant.

The first-cut 3NF decomposition is ABC, BCE, CD. Since AB is a key included in ABC, no key needs to be added, so this is a 3NF, dependency preserving decomposition with lossless join. The decomposision is **not** in BCNF, because ABC violates BCNF. Indeed, $B^+ = BCDE$ so B determines C but not A within ABC.

$$F = \{C \to D, AC \to BDE, AB \to CDE, B \to CE\}$$

Let R be a relation with attributes $ABCDEG$ and

$$F = \{E \to D,\ C \to B,\ CBE \to AG,\ B \to A,\ G \to E\}.$$

(i) (1 *point*) Find all the keys of R.
Note that C does not occur on the right-hand side of any FD, so it must belong to every key. Since $C^+ = CBA$, C is not a key, so at least one attribute must be added (other than A, B which are already determined by C). Continuing the search, $CD^+ = CDBA$, $CE^+ = CEABDEG$, and $CG^+ = CGBABD$ so CE and CG are keys. Extending CD to a superkey requires adding at least one of E and G, so this would subsume the keys CE and CG so no other key can be obtained from CD. Thus, the only keys are CE and CG.

(ii) (4 *points*) Find a BCNF decomposition of R with lossless join with respect to F. (Show how the decomposition is obtained.)
The BCNF decomposition algorithm produces $\{ED, CB, GE, CA, CG\}$ (see separate pdf file for details).

(iii) (2 *points*) Is the decomposition obtained in (ii) dependency preserving with respect to F ?
The FDs $E \to D$, $C \to B$, $G \to E$ are local so are preserved. The FD $CBE \to AG$ is *not* preserved. To see this, start with CBE. We now have the following starred attributes in the decomposition:

$$E^*D, C^*B^*, GE^*, C^*A, C^*G$$

Since $E^+ = ED$ and D is in ED we can add D. Since $C^+ = CBA$ and A is in CA, we can add A. We now have $CBEDA$ and the starred attributes

$$E^*D^*, C^*B^*, GE^*, C^*A^*, C^*G$$

We are now done, because $E^+ = ED$ and $C^+ = CBA$ so nothing can be added. Since $CBEDA$ does not contain G, $CBE \to AG$ is not preserved. So the decomposition is not dependency preserving.

We first eliminate redundancies in F.

- single attributes on right-hand sides of FDs:

$$E \to D,\ C \to B,\ CBE \to A,\ CBE \to G,\ B \to A,\ G \to E$$

- eliminate redundant FDs. The FDs $E \to D$, $C \to B$, $CBE \to G$, $G \to E$ are not redundant (because each is the only FD producing the attribute on the right-hand side). The only redundant FD is $CBE \to A$ (it is implied by $B \to A$). We are left with

$$E \to D,\ C \to B,\ CBE \to G,\ B \to A,\ G \to E$$

- eliminate redundant attributes on left-hand sides of FDs. The only FD with multiple attributes on its left-hand side is $CBE \to G$. Attribute B is redundant because $CE^+ = CEDBGA$ which includes G. C is not redundant because $E^+ = E$ and E is not redundant because $C^+ = CBA$ (so none determines G).

The result of eliminating redundancies is

$$E \to D,\ C \to B,\ CE \to G,\ B \to A,\ G \to E$$

Problem with BCNF:

- Not every relation schema can be decomposed into BCNF relation schemas which have lossless join *and* preserve the dependencies.

⇓

Third Normal Form (3NF)

- A relation scheme R is in Third Normal Form wrt a set F of fd's over R, if whenever X→A holds in R and $A \notin X$ then either X is a superkey or A is prime

$A \in att(R)$ is prime: $A \in K$ for some key K

3NF is weaker than BCNF

-Every relation in BCNF is also in 3NF
- Every relation in 3NF is not necessarily also in BCNF
-Not every relation in BCNF has lossless join wrt FDs
- 2P ensures serializability but not prevent deadlock.

S_1, S_2 are conflict equivalent schedules if S_1 can be transformed into S_2 by a series of swaps of adjacent non-conflicting actions.

Non-conflicting actions:

- actions on different data
- read/read on the same data

2. Give an example of two transactions T1 and T2 that lock the sam data entities, such that T1 satisfies two-phase locking, T2 violates two se locking, and there is a schedule for T1 and T2 that is not confli alizable.

ution An example is the following:

$$T1 = l(A); w(A); l(B); w(B); u(A); u(B)$$
$$T2 : l(A); w(A); u(A); l(B); w(B); u(B)$$

=$r_1(A)w_1(A)r_2(A)w_2(A)r_1(B)w_1(B)r_2(B)w_2(B)$

SC′=$r_1(A)w_1(A)\ r_1(B)w_1(B)r_2(A)w_2(A)r_2(B)w_2(B)$

Theorem 2PL ⇒ conflict serializable schedule

List the directors such that every actor is cast in one of his/her movies.

$$\pi_{director}(movie) - \pi_{director}[\pi_{director}(movie) \bowtie \pi_{actor}(movie) - \pi_{director,actor}(movie)]$$

List all pairs of actors who act together in at least one movie.

$$\pi_{actor_1,actor_2}[\delta_{actor \to actor_1}(\pi_{title,actor}(movie)) \bowtie \delta_{actor \to actor_2}(\pi_{title,actor}(movie))]$$