Relational Algebra

Outline

1. *Selection and Projection*
2. *Union, intersection, and difference*
3. *Renaming*
4. *Cross-Product, Join and Division*
5. *Expression Tree and Introduction of Query Optimization*

INTRO

ALGEBRA?

Mathematical system that consists of :

- ***Operands :*** *variables or values from which values can be constructed*
- ***Operators*** *: symbols denoting procedures that construct new values from given values*

Example:

$$ax^2 + bx - y = 0$$

INTRO

- ***Relational Algebra :*** a formal query language
- ***The relational algebra*** *is* ***important*** *because:*
  1. It provides a formal foundation for relational model operations;
  2. It is used as a basis for implementing and optimizing queries in the query processing and optimization modules that are integral parts of relational database management systems (RDBMS).
- ***Relational Algebra*** consists of :
  - **Basic Operator** (selection, projection, union, intersect, difference, cross-product)
  - **Additional Operator** (renaming, join, division)

1

SELECTION AND PROJECTION

SCHEMA

DataPasien		
IDPasien	**NoKTP**	**Nama Pasien**
PS00006	3507254111940001	Lidra Trifidya
PS00007	3606125204940003	Yutika Amelia Effendi

Dokter		
NID	**KodePoli**	**Nama**
D009	P01	Dr. Budikusnaedi
D0010	P02	Dr. Hariyanto Kusuma
D0011	P02	Dr. Sri Herianti
D0012	P03	Drg. Elvin Purwantari

Pemeriksaan				
IDPeriksa	**NID**	**IDPasien**	**TglPeriksa**	**Diagnosa**
C001	D009	PS00006	01-Jan-15	Radang Usus Buntu
C002	D010	PS00007	02-Jan-15	Gigi Berlubang
C003	D011	PS00007	10-Feb-15	Flu
C004	D019	PS00007	02-May-15	Radang Usus Buntu

SELECTION

Selection operators ***:*** *pick certain rows*

$\mathbf{T1} := \boldsymbol{\sigma}_{\mathbf{C}} (\mathbf{T2})$

- $\boldsymbol{C}$ is a condition (as in “if” statements) that refers to attributes of $\mathbf{T2}$
- $\mathbf{T1}$ is all those tuples of $\mathbf{T2}$ that satisfy $\boldsymbol{C}$

SELECTION

Example 1 : A patient who has IDPasien PS00006

$\sigma_{IDPasien=`PS00006`}(DataPasien)$

Result:

DataPasien		
IDPasien	**NoKTP**	**Nama Pasien**
PS00006	3507254111940001	Lidra Trifidya

SELECTION

Example 2 : Patients who are handled by a doctor with NID D009 on 01 January 2015

$$\sigma_{NID=`D009` \wedge TglPeriksa=`01-Jan-15`}(Pemeriksaan)$$

Result:

Pemeriksaan				
IDPeriksa	**NID**	**IDPasien**	**TglPeriksa**	**Diagnosa**
C001	D009	PS00006	01-Jan-15	Radang Usus Buntu

PROJECTION

Projection operators ***:*** *pick certain columns*

$\mathbf{T1} := \boldsymbol{\pi}_{\boldsymbol{C}} (\mathbf{T2})$

- $\boldsymbol{C}$ is a list of attributes from the schema of $\mathbf{T2}$
- $\mathbf{T1}$ is constructed by looking at each tuple of $\mathbf{T2}$, extracting the attributes on list $\boldsymbol{C}$, L , in the order specified, and creating from those components a tuple for $\mathbf{T1}$

PROJECTION

Example 1 : Show IDPeriksa and IDPasien on Table Pemeriksaan

$\pi_{IDPeriksa,\, IDPasien}(Pemeriksaan)$

Result:

Pemeriksaan	
IDPeriksa	**IDPasien**
C001	PS00006
C002	PS00007
C003	PS00007
C004	PS00007

SELECTION AND PROJECTION

Example 1 : Show IDPeriksa and IDPasien of Patients who are handled by a doctor with NID D009 on 01 January 2015

$$\pi_{IDPeriksa,\, IDPasien}(\sigma_{NID=`D009` \wedge TglPeriksa=`01-Jan-15`}(Pemeriksaan))$$

Result:

Pemeriksaan	
IDPeriksa	**IDPasien**
C001	PS00006

2

UNION, INTERSECTION, AND DIFFERENTS

UNION, INTERSECTION, DIFFERENT

- **Union**: *R ∪ S returns a relation instance containing all tuples that occur in either* relation instance *R or relation instance S (or both)*
- **Intersection**: *R∩S returns a relation instance containing all tuples that occur in both R and S*
- **Set-difference**: *R–S returns a relation instance containing all tuples that occur* in *R but not in S.*

UNION, INTERSECTION, DIFFERENT

Union

Dokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0002	Bedah Saraf	Achmad Fahmi
D0003	Anak	Sulis
D0004	Anak	Prima

KepalaDokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0003	Anak	Sulis

$Dokter \cup KepalaDokter$		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0002	Bedah Saraf	Achmad Fahmi
D0003	Anak	Sulis
D0004	Anak	Prima

UNION, INTERSECTION, DIFFERENT

Intersection

Dokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0002	Bedah Saraf	Achmad Fahmi
D0003	Anak	Sulis
D0004	Anak	Prima

KepalaDokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0003	Anak	Sulis

$Dokter \cap KepalaDokter$		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0003	Anak	Sulis

UNION, INTERSECTION, DIFFERENT

Set-Difference

Dokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0002	Bedah Saraf	Achmad Fahmi
D0003	Anak	Sulis
D0004	Anak	Prima

KepalaDokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0003	Anak	Sulis

$Dokter - KepalaDokter$		
NID	**KodePoli**	**Nama**
D0002	Bedah Saraf	Achmad Fahmi
D0004	Anak	Prima

3

RENAMING

SCHEMA

DataPasien		
IDPasien	**NoKTP**	**Nama Pasien**
PS00006	3507254111940001	Lidra Trifidya
PS00007	3606125204940003	Yutika Amelia Effendi

Dokter		
NID	**KodePoli**	**Nama**
D009	P01	Dr. Budikusnaedi
D0010	P02	Dr. Hariyanto Kusuma
D0011	P02	Dr. Sri Herianti
D0012	P03	Drg. Elvin Purwantari

Pemeriksaan				
IDPeriksa	**NID**	**IDPasien**	**TglPeriksa**	**Diagnosa**
C001	D009	PS00006	01-Jan-15	Radang Usus Buntu
C002	D010	PS00007	02-Jan-15	Gigi Berlubang
C003	D011	PS00007	10-Feb-15	Flu
C004	D019	PS00007	02-May-15	Radang Usus Buntu

RENAMING

Renaming operators ***:*** changes attribute names for a relation without changing any values.

There are two ways to express *renaming:*

1. $\mathbf{T1} := \boldsymbol{\rho}_{A_1 \to Att_1, \ldots, A_n \to Att_n}(\mathbf{T2})$
2. $\mathbf{T1} := \boldsymbol{\rho}(A_1 \to Att_1, \ldots, A_n \to Att_n, \mathbf{T2})$

where $\mathbf{T1}$ has attributes $Att_1, \ldots, Att_n$ and the same tuples as $\mathbf{T2}$

RENAMING

Renaming tables : changes the name of table

There are two ways to express *renaming:*

1. $\mathbf{T1} := \boldsymbol{\rho}_{\boldsymbol{D}} (\mathbf{T2})$
2. $\mathbf{T1} := \boldsymbol{\rho} (\boldsymbol{D}, \mathbf{T2})$

where $\mathbf{T1}$ is a table $\mathbf{T2}$ whose name is changed to $\boldsymbol{D}$

RENAMING

Example 1 : Change IDPeriksa of Table Pemeriksaan to NomorUrut and IDPasien of Table Pemeriksaan to KodePasien

$\rho_{IDPeriksa \rightarrow NomorUrut, IDPasien \rightarrow KodePasien}\ (Pemeriksaan)$ **or**

$\rho(IDPeriksa \rightarrow NomorUrut, IDPasien \rightarrow KodePasien, Pemeriksaan)$

Result:

Pemeriksaan				
NomorUrut	**NID**	**KodePasien**	**TglPeriksa**	**Diagnosa**
C001	D009	PS00006	01-Jan-15	Radang Usus Buntu
C002	D010	PS00007	02-Jan-15	Gigi Berlubang
C003	D011	PS00007	10-Feb-15	Flu
C004	D019	PS00007	02-May-15	Radang Usus Buntu

RENAMING

Example 1 : Change the name of Table Pemeriksaan to Table DaftarPemeriksaan

$\rho_{DaftarPemeriksaan}(Pemeriksaan)$ **or**

$\rho(DaftarPemeriksaan, Pemeriksaan)$

Result:

DaftarPemeriksaan				
IDPeriksa	**NID**	**IDPasien**	**TglPeriksa**	**Diagnosa**
C001	D009	PS00006	01-Jan-15	Radang Usus Buntu
C002	D010	PS00007	02-Jan-15	Gigi Berlubang
C003	D011	PS00007	10-Feb-15	Flu
C004	D019	PS00007	02-May-15	Radang Usus Buntu

RENAMING

Example 1 : Change the name of Table Pemeriksaan to Table DaftarPemeriksaan, change IDPeriksa to NomorUrut and change IDPasien to KodePasien

$\rho_{DaftarPemeriksaan(IDPeriksa \rightarrow NomorUrut, IDPasien \rightarrow KodePasien)}\ (Pemeriksaan)$ **or**

$\rho(DaftarPemeriksaan(IDPeriksa \rightarrow NomorUrut, IDPasien \rightarrow KodePasien), Pemeriksaan)$

Result:

DaftarPemeriksaan				
NomorUrut	**NID**	**KodePasien**	**TglPeriksa**	**Diagnosa**
C001	D009	PS00006	01-Jan-15	Radang Usus Buntu
C002	D010	PS00007	02-Jan-15	Gigi Berlubang
C003	D011	PS00007	10-Feb-15	Flu
C004	D019	PS00007	02-May-15	Radang Usus Buntu

4

CROSS-PRODUCT, JOIN, AND DIVISIONS

CROSS-PRODUCT

Cross-Product operators **:** *a relation instance whose schema contains all the fi*elds of *R (in the same order as they appear in R) followed by all the fields of S* (in the same order as they appear in *S)*

$\mathbf{R1} := \mathbf{R2} \times \mathbf{R3}$

- Pair each tuple of **R3** with each tuple of **R2**.
- Schema of **R3**is the attributes of **R1**and then **R2**, in order

CROSS-PRODUCT

Dokter		
NID	**KodePoli**	**Nama**
D0002	Bedah Saraf	Achmad Fahmi
D0004	Anak	Prima

KepalaDokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0003	Anak	Sulis

$\mathbf{Dokter} \times \mathbf{KepalaDokter}$					
NID	**KodePoli**	**Nama**	**NID**	**KodePoli**	**Nama**
D0002	Bedah Saraf	Achmad Fahmi	D0001	Bedah Saraf	Agus Turchan
D0002	Bedah Saraf	Achmad Fahmi	D0003	Anak	Sulis
D0004	Anak	Prima	D0001	Bedah Saraf	Agus Turchan
D0004	Anak	Prima	D0003	Anak	Sulis

JOIN

Join *combines information from two or more relations*

- ***Join*** *is cross product operation, followed by selection and projection*

Type of Join :

- *Theta-join (conditional join)*
- *Natural join*
- *Left Outer Join*
- *Right Outer Join*
- *Full Outer Join*

JOIN

Theta Join (Conditional Join)

$\mathbf{R1 := R2 \bowtie_C R3}$

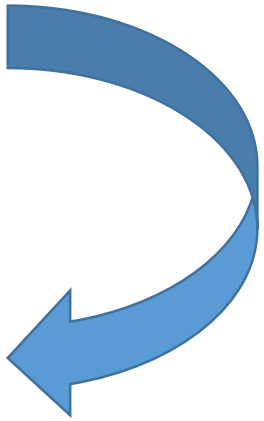
- Take Product **R3** and **R2**
- Apply $\boldsymbol{\sigma_C}$ to the result
- The special case of *theta join* is **Equijoin** (having equality operator as the condition **C**)

JOIN

Theta Join

A	
NIDA	**NameA**
1	K
5	L

B	
NIDB	**NameB**
1	M
2	N



$\mathbf{A} \bowtie_{A.NIDA<B.NIDB} \mathbf{B}$			
NIDA	**NameA**	**NIDB**	**NameB**
1	K	2	N

JOIN

Equijoin

$$\mathbf{R1} := \mathbf{R2} \bowtie_{C} \mathbf{R3}$$

- The condition ($\mathbf{C}$) consists solely of equalities ($=$)
- Equijoin is a theta join using equality operator
- In Equijoin, the join operation do an additional projection in which one of the attribute in the condition is dropped

JOIN

Equijoin

Dokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0004	Anak	Prima

KepalaDokter		
NIDKepala	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0003	Anak	Sulis

Dokter $\bowtie_{Dokter.NID=KepalaDokter.NIDKepala}$ **KepalaDokter**				
NID	**KodePoli**	**Nama**	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan	Bedah Saraf	Agus Turchan

Because of equijoin, only one of the attribute in the condition, i.e. Dokter.NID=KepalaDokter.NIDKepala, is chosen, so ***KepalaDokter.NIDKepala*** *is projected out.*

JOIN

Natural Join

A useful join variant (natural join) connects two relations by:

- ❖ Equating attributes of the same name, and
- ❖ Projecting out one copy of each pair of equated attributes

$$\mathbf{R1} := \mathbf{R2} \bowtie \mathbf{R3}$$

JOIN

Natural Join

Dokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0004	Anak	Prima

KepalaDokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0003	Anak	Sulis

Dokter ⋈ KepalaDokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan

Because both of Table Dokter and Table KepalaDokter have same names of attributes, so only one NID, one KodePoli and one Nama are chosen.

JOIN

Natural Join

Dokter		
NID	**KodePoli**	**Nama**
D0001	Bedah Saraf	Agus Turchan
D0004	Anak	Prima

KepalaDokter		
NIDKepala	**KodePoliKepala**	**NamaKepala**
D0001	Bedah Saraf	Agus Turchan
D0003	Anak	Sulis

Dokter ⋈ KepalaDokter					
NID	**KodePoli**	**Nama**	**NIDKepala**	**KodePoliKepala**	**NamaKepala**
D0001	Bedah Saraf	Agus Turchan	D0001	Bedah Saraf	Agus Turchan
D0001	Bedah Saraf	Agus Turchan	D0003	Anak	Sulis
D0004	Anak	Prima	D0001	Bedah Saraf	Agus Turchan
D0004	Anak	Prima	D0003	Anak	Sulis

Because there is no equated attributes of Table Dokter and Table KepalaDokter, each tuple in Table Dokter is paired with each tuple in Table KepalaDokter (natural join becomes cross-product)

JOIN

Left Outer Join

- It is left join in SQL
- The relational algebra of left outer join is shown below.

$$\mathbf{R1} := \mathbf{R2} \ltimes_{C} \mathbf{R3}$$

JOIN

Left Outer Join

A	
NIDA	**NameA**
1	K
5	L

B	
NIDB	**NameB**
1	M
2	N

$A \ltimes_{A.NIDA>B.NIDB} B$			
NIDA	**NameA**	**NIDB**	**NameB**
1	K	NULL	NULL
5	L	1	M
5	L	2	N

JOIN

Right Outer Join

- It is right join in SQL
- The relational algebra of left outer join is shown below.

$$\mathbf{R1} := \mathbf{R2} \ltimes_{c} \mathbf{R3}$$

JOIN

Right Outer Join

A	
NIDA	**NameA**
1	K
5	L

B	
NIDB	**NameB**
1	M
2	N

$A \bowtie_{A.NIDA<B.NIDB} B$			
NIDA	**NameA**	**NIDB**	**NameB**
NULL	NULL	1	M
1	K	2	N

JOIN

Full Outer Join

- It is full join in SQL
- The relational algebra of left outer join is shown below.

$$\mathbf{R1} := \mathbf{R2} \ \overset{\leftarrow}{\bowtie}_{C} \mathbf{R3}$$

JOIN

Full Outer Join

A	
NIDA	**NameA**
1	K
5	L

B	
NIDB	**NameB**
1	M
2	N

$A \overset{\mathcal{I}}{\bowtie}_{A.NIDA<B.NIDB} B$			
NIDA	**NameA**	**NIDB**	**NameB**
1	K	2	N
5	L	NULL	NULL
NULL	NULL	1	M

DIVISION

Division

R1 := R2/R3

- All attributes in **R3** (denominator) must als**o** be presented in **R2** (numerator)
- **R2** has some extra attributes
- The schema of the result relation **R1** contains the attributes defined in **R2** and not defined in **R3**

DIVISION

Division

A

sno	*pno*
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

B1

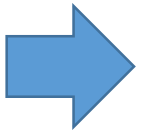
pno
p2

B2

pno
p2
p4

B3

pno
p1
p2
p4



A/B1

sno
s1
s2
s3
s4

A/B2

sno
s1
s4

A/B3

sno
s1

5

EXPRESSION TREE AND INTRODUCTION OF QUERY OPTIMIZATION

EXPRESSION TREE

- **Leaves :** can be called operands -- either variables standing for relations or particular, constant relations
- **Interior nodes** : can be called operators -- applied to their child or children

EXPRESSION TREE

Example 1 : Show IDPeriksa and IDPasien of Patients who are handled by a doctor with NID D009 on 01 January 2015

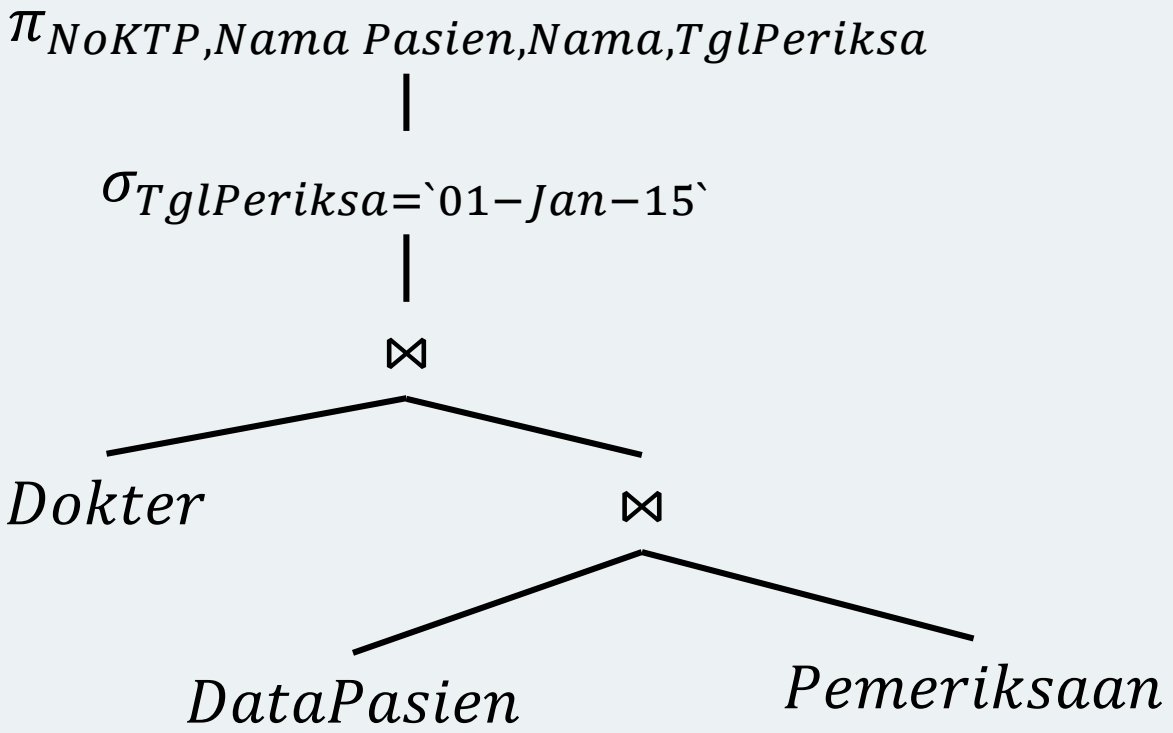
$\pi_{IDPeriksa,\, IDPasien}(\sigma_{NID=`D009` \wedge TglPeriksa=`01-Jan-15`}(Pemeriksaan))$

$$\pi_{IDPeriksa,\, IDPasien}$$
$$|$$
$$\sigma_{NID=`D009` \wedge TglPeriksa=`01-Jan-15`}$$
$$|$$
$$Pemeriksaan$$

EXPRESSION TREE

Example 2 : Show ID of patient, name of patient, name of doctors and the check date on 01 January 2015

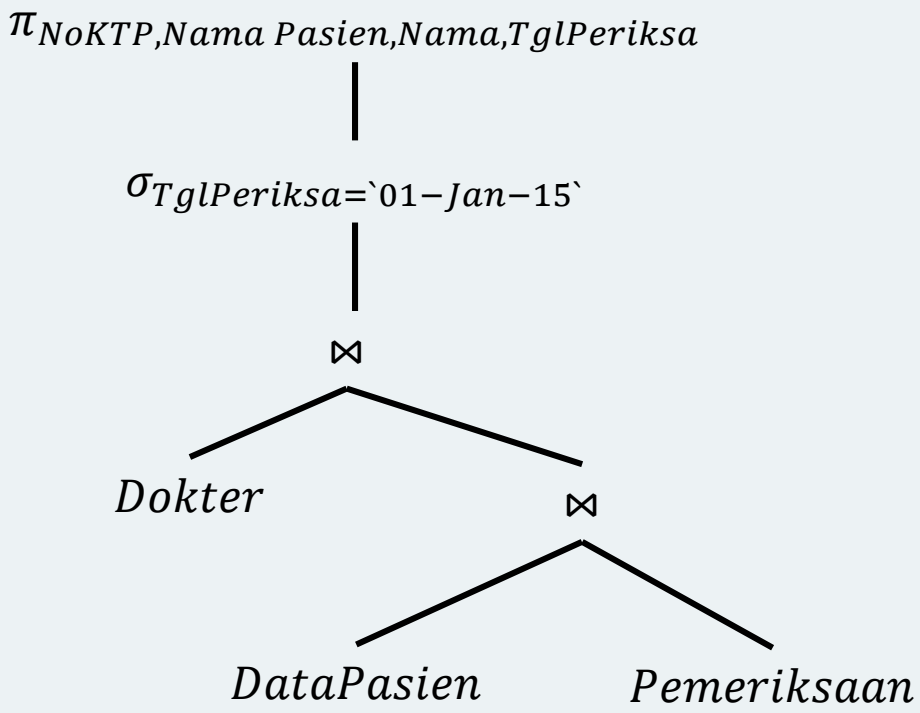
$$\pi_{NoKTP,Nama\ Pasien,Nama,TglPeriksa}(\sigma_{TglPeriksa=`01-Jan-15`}(Dokter \bowtie (DataPasien \bowtie Pemeriksaan)))$$



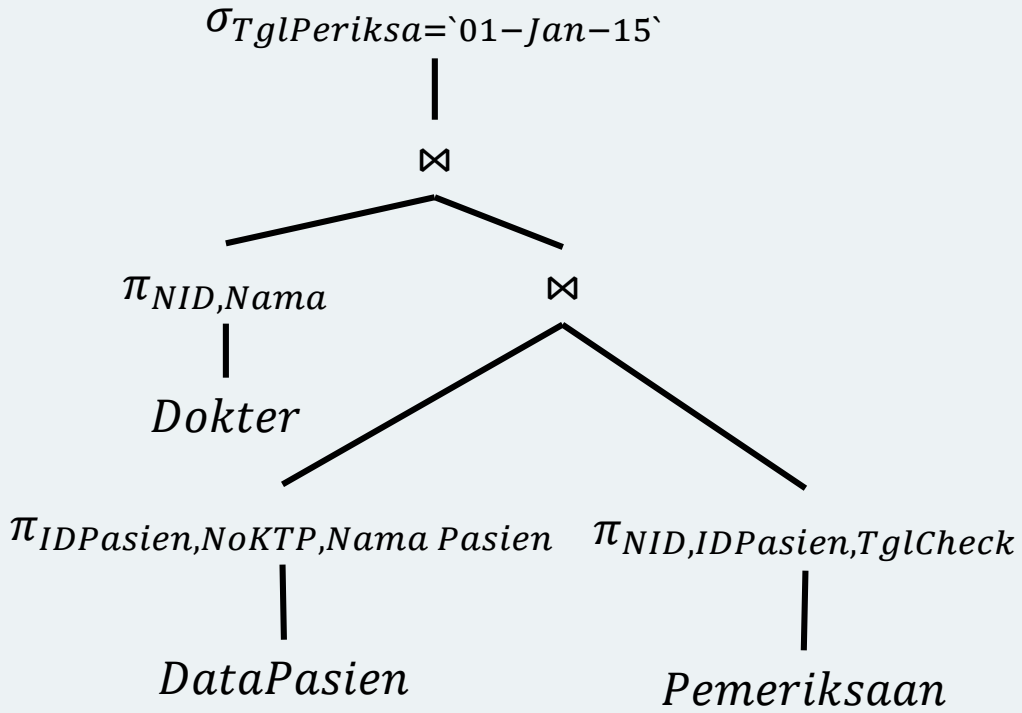
QUERY OPTIMIZATION

- *Relational Algebra can be used in query optimization by comparing the query with expression tree*
- *Example:* **Show ID of patient, name of patient, name of doctors and the check date on 01 January 2015**

1st Option



2nd Option



6

EXERCISES

SCHEMA

DataPasien(*IDPasien: string, NoKtp: integer, Nama Pasien: string)*

Dokter(*NID: string, KodePoli: string, Nama: string)*

Poli(*KodePoli: string, NamaPoli: string)*

Pemeriksaan(*IDPeriksa: string, NID: string, IDPasien: string, TglPeriksa: date, Diagnosa: string)*

IDPasien	NoKTP	Nama Pasien
PS00006	3507254111940001	Lidra Trifidya
PS00007	3606125204940003	Yutika Amelia Effendi

NID	KodePoli	Nama
D009	P01	Dr. Budikusnaedi
D0010	P02	Dr. Hariyanto Kusuma
D0011	P02	Dr. Sri Herianti
D0012	P03	Drg. Elvin Purwantari

KodePoli	NamaPoli
P01	Bedah Umum
P02	Mulut
P03	THT

IDPeriksa	NID	IDPasien	TglPeriksa	Diagnosa
C001	D009	PS00006	01-Jan-15	Radang Usus Buntu
C002	D010	PS00007	02-Jan-15	Gigi Berlubang
C003	D011	PS00007	10-Feb-15	Flu
C004	D019	PS00007	02-May-15	Radang Usus Buntu

Create the relational algebra, the expression tree and the table result

1. Select id of patient and check date of patient who got “Gigi Berlubang” or “Flu
2. Select KTP number and name of patients who checked on May
3. Select name of patients, name of doctors, the doctor’s poly, and check date of patients who handled by doctor Dr. Budikusnaedi
4. Select a pair of doctors who are in the same poly (a pair **cannot** contain of same doctor)
5. Select patients who had been checked in all poly

Fill the results of the questions based on the below tables

R2 = (B C)

B	C
5	6
7	8
9	10

R = (A B)

A	B
1	2
5	6
1	2

S = (B C)

B	C
3	4
7	8

6. $\sigma_{A*B>5}(R)$= …
7. $\pi_{C}(R2)$ = …
8. $R \times S =$ …
9. $R \bowtie_{R.B<R2.C} R2 =$ …
10. $\pi_{C}\ (S \bowtie (R \bowtie R2))$= …