

DATABASE DESIGN & MANAGEMENT

SI10317

PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS TARUMANAGARA

Course Schedule

- 1 Entity Relationship Modeling dan Alternative ER Notation – Appendix C
- 2 Exercises
3. Enhanced Entity–Relationship Modeling
4. Exercises
5. Normalization dan Exercises
6. Advanced Normalization dan Exercises
7. Review and the *DreamHome* Case Study
8. UTS - Presentasi Project



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Course Schedule

9. Methodology—Conceptual Database Design
10. Methodology—Logical Database Design
11. Exercises: Case Study Appendix A, B1, B2
12. Presentasi Project: Case Study
13. Distributed DBMSs—Concepts and Design
- 14. Query Processing**
15. Replication and Mobile Databases
16. Presentasi Project UAS



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Query Processing

Activities involved in retrieving data from the database.

- Aims of QP:
 - transform query written in high-level language (e.g. SQL), into correct and efficient execution strategy expressed in low-level language (implementing RA);
 - execute strategy to retrieve required data.



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Query Optimization

Example 21.1 Comparison of different processing strategies

Find all Managers who work at a London branch.

We can write this query in SQL as:

```
SELECT *  
FROM Staff s, Branch b  
WHERE s.branchNo = b.branchNo AND  
      (s.position = 'Manager' AND b.city = 'London');
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Three equivalent RA queries

(1) $\sigma_{(\text{position}=\text{'Manager'}) \wedge (\text{city}=\text{'London'}) \wedge (\text{Staff.branchNo}=\text{Branch.branchNo})} (\text{Staff} \times \text{Branch})$

(2) $\sigma_{(\text{position}=\text{'Manager'}) \wedge (\text{city}=\text{'London'})} (\text{Staff} \bowtie_{\text{Staff.branchNo}=\text{Branch.branchNo}} \text{Branch})$

(3) $(\sigma_{\text{position}=\text{'Manager'}}(\text{Staff})) \bowtie_{\text{Staff.branchNo}=\text{Branch.branchNo}} (\sigma_{\text{city}=\text{'London'}}(\text{Branch}))$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Example 20.1 - Different Strategies

- Kita asumsikan bahwa terdapat:
 - 1000 tuple di Staff, 50 tuple di Branch,
 - 50 Manager (satu manajer untuk setiap Branch), dan
 - 5 Branch di London.
- Untuk menyederhanakannya kita berasumsi bahwa:
 - tidak ada indeks atau sort key pada salah satu relasi,
 - hasil dari setiap operasi perantara apapun yang disimpan pada disk
 - biaya penulisan akhir diabaikan karena sama di setiap kasus
 - tuple diakses satu per satu
 - memori utama cukup besar untuk memproses operasi RA.



UNTAR
Universitas Tarumanagara

Terakreditasi
BAN-PT

A
Linggi

QS STARS
RATING SYSTEM
2019

ISIRI
KUALITAS

IABEE

CPA
AUSTRALIA

ICAEW
CHARTERED
ACCOUNTANTS

UNTAR untuk INDONESIA

Example 20.1 - Different Strategies

- Biaya (dalam akses disk) adalah:
 - Query pertama menghitung Cartesian Product dari Staff dan Branch, yang membutuhkan $(1000 + 50)$ akses disk untuk membaca relasi, dan membuat relasi dengan $(1000 * 50)$ tuple.
 - Kemudian membaca masing-masing tuple ini lagi untuk mengujinya terhadap seleksi predikat dengan biaya akses disk $(1000 * 50)$, memberikan total biaya:
- $(1000 + 50) + 2 * (1000 * 50) = 101\ 050$ akses disk



Example 20.1 - Different Strategies

- Biaya (dalam akses disk) adalah:
 - Query kedua “join” Staff dan Branch berdasarkan *branchNo*, yang membutuhkan (1000 + 50) akses disk untuk membaca setiap relasi.
 - Hasil join dua relasi memiliki 1000 tupel, satu untuk setiap staf (bahwa staf hanya dapat bekerja di satu cabang).
 - Sehingga operasi Selection memerlukan 1000 akses disk untuk membaca hasil dari proses join, sehingga total biaya:
- $2 * 1000 + (1000 + 50) = 3050$ akses disk.



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Example 20.1 - Different Strategies

- Biaya (dalam akses disk) adalah:
 - Query ketiga (terakhir), yang pertama membaca setiap tuple Staf untuk menentukan tuple Manajer, membutuhkan 1000 akses disk dan menghasilkan relasi dengan 50 tuple.
 - Operasi Selection kedua membaca setiap tuple Branch untuk menentukan branch nya London, yang membutuhkan 50 akses disk dan menghasilkan relasi dengan 5 tuple.
 - Operasi terakhir adalah 'join' dari relasi Staf dan Branch, yang membutuhkan $(50 + 5)$ akses disk, sehingga total biaya:
 - $1000 + 2 * 50 + 5 + (50 + 5) = 1160$ akses disk.



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Phases of Query Processing

- QP has four main phases:
 - decomposition (consisting of parsing and validation);
 - optimization;
 - code generation;
 - execution.

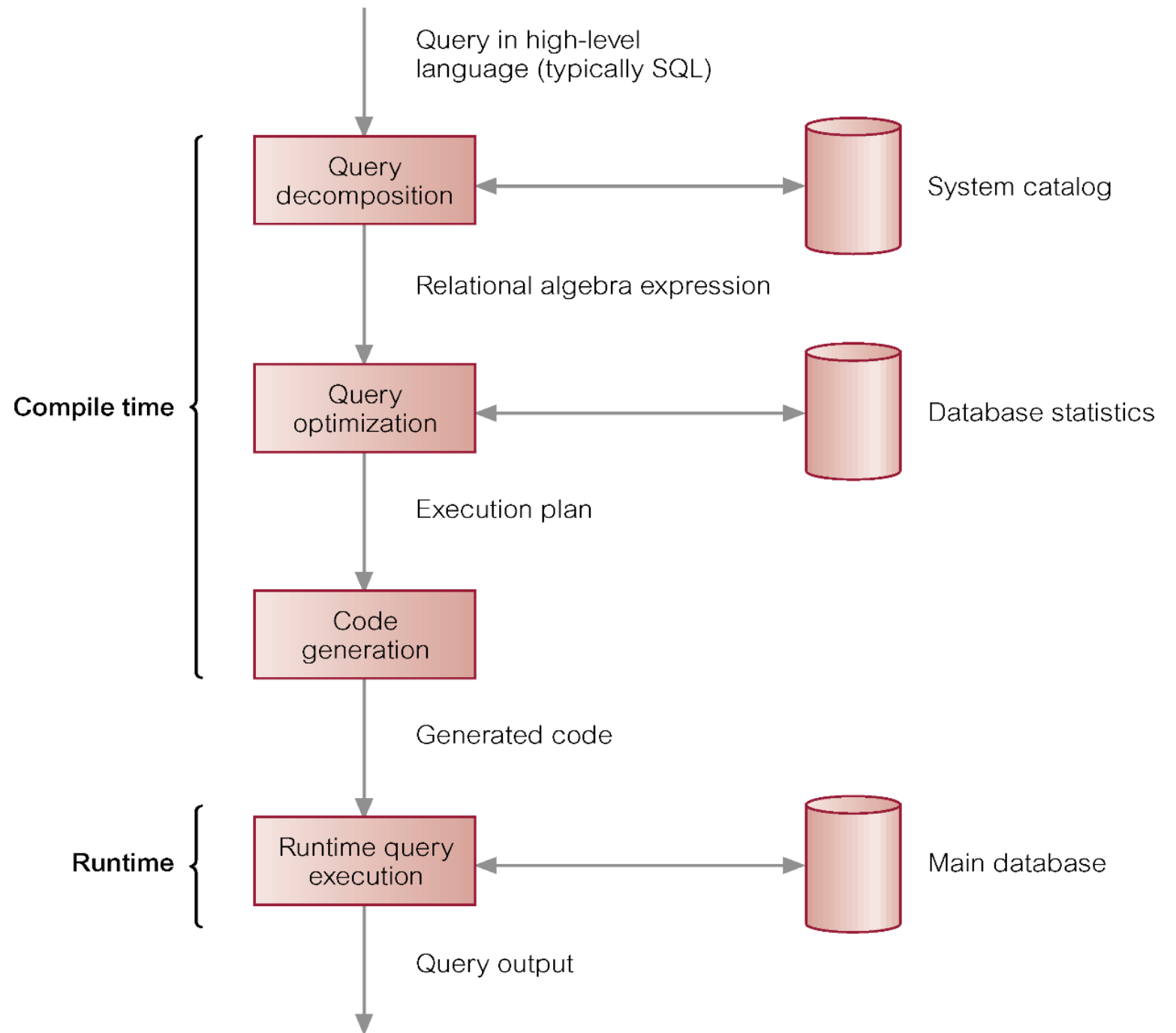


UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Phases of Query Processing



Query Decomposition

- Bertujuan mengubah high-level query menjadi relational algebra (RA) query.
- Memeriksa apakah query tersebut benar secara sintaksis (syntactically) dan semantik (semantically).
- Tahapan tipikal adalah
- Tahapan secara khusus (typical stages) seperti adalah:
 - analysis,
 - normalization,
 - semantic analysis,
 - simplification,
 - query restructuring.



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Analysis

- Menganalisis query secara leksikal dan sintaksis menggunakan teknik kompilator.
- Verifikasi relations dan atribut yang ada.
- Verifikasi operasi sesuai untuk tipe objek



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Analysis - Example

```
SELECT staff_no  
FROM Staff  
WHERE position > 10;
```

- Hasil analisisnya, bahwa query ini akan ditolak, dengan dua alasan:
 - Pada SELECT, staff_no tidak terdefinisi untuk relasi staff relation (harusnya staffNo).
 - Pada WHERE klausa, perbandingan '> 10' tidak sesuai/kompatibel dengan tipe position, yang variabelnya merupakan character string.



UNTAR
Universitas Tarumanagara

Terakreditasi
BAN PT

A
linggi

QS STARS
RATING SYSTEM
2019

AMBA
ACCREDITED

IAABE

CPA
AUSTRALIA

ICAEW
CHARTERED
ACCOUNTANTS

UNTAR untuk INDONESIA

Analysis

- Akhirnya, query diubah menjadi beberapa representasi internal yang lebih cocok untuk diproses.
- Beberapa jenis query tree dapat dipilih, disusun sebagai berikut:
 - Leaf node dibuat untuk setiap relasi dasar dalam query.
 - Non-leaf node dibuat untuk setiap relasi perantara yang dihasilkan oleh operasi RA
 - Root of tree merepresentasikan hasil query.
 - Urutan operasi diarahkan dari leaves to root (daun ke akar)

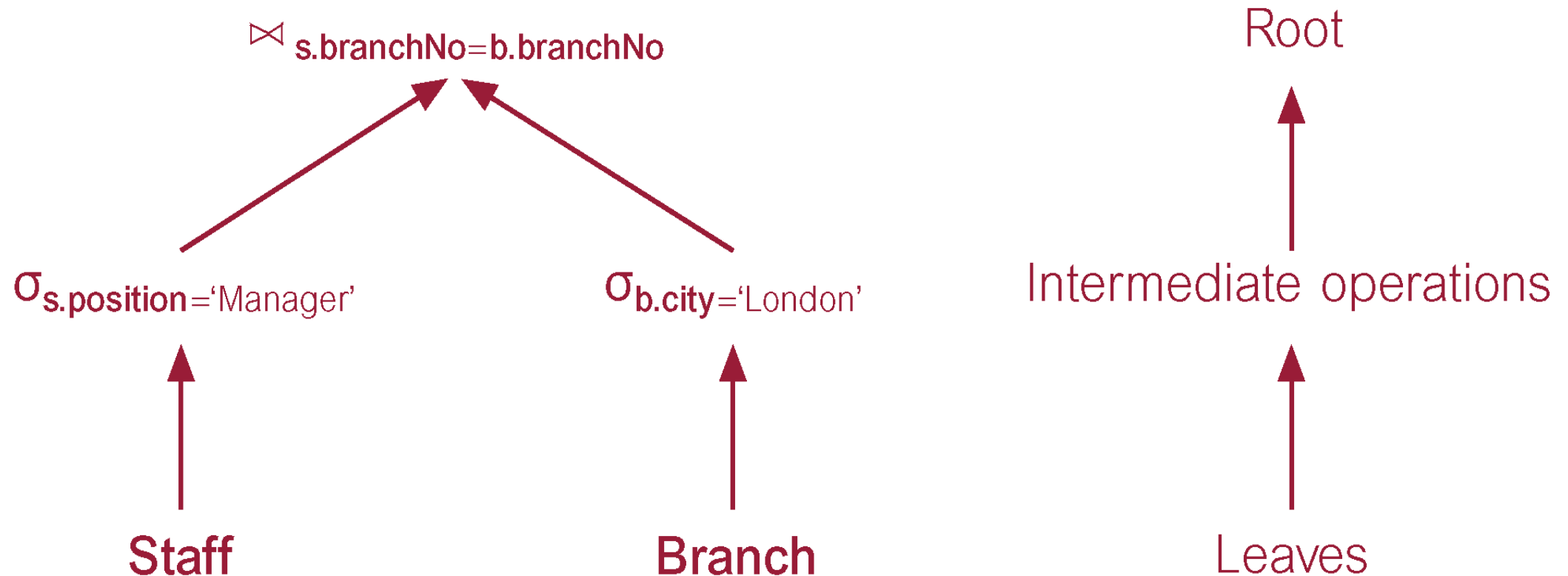


UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Figure 21.2. Example relational algebra tree.



Normalization

- Mengonversi query menjadi bentuk yang dinormalisasi untuk manipulasi yang lebih mudah.
- Predicate dapat diubah menjadi salah satu dari dua bentuk:

Conjunctive normal form:

$(\text{position} = \text{'Manager'} \vee \text{salary} > 20000) \wedge (\text{branchNo} = \text{'B003'})$

Disjunctive normal form:

$(\text{position} = \text{'Manager'} \wedge \text{branchNo} = \text{'B003'}) \vee$
 $(\text{salary} > 20000 \wedge \text{branchNo} = \text{'B003'})$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Semantic Analysis

- Tujuan, menolak query yang dinormalisasi dengan rumusan yang salah atau kontradiktif.
- Query salah diformulasikan jika komponen tidak berkontribusi pada pembuatan hasil.
- Query disebut kontradiktif jika predikatnya tidak dapat dipenuhi oleh tuple manapun.
 - Misalnya, predikat ($\text{position} = \text{'Manager'} \wedge \text{position} = \text{'Assistant'}$) pada relasi Staf adalah kontradiktif, karena staf tidak dapat menjadi Manager sekaligus Asisten.



UNTAR
Universitas Tarumanagara

Terakreditasi
BAN PT

A
Linggi

QS STARS
RATING SYSTEM
2019

GLAC
UNAR

IABEE

CPA
AUSTRALIA

ICAEW
CHARTERED
ACCOUNTANTS

UNTAR untuk INDONESIA

Example 21.2 - Checking Semantic Correctness

Consider the following SQL query:

```
SELECT p.propertyNo, p.street  
FROM Client c, Viewing v, PropertyForRent p  
WHERE c.clientNo = v.clientNo AND  
        c.maxRent >= 500 AND c.prefType = 'Flat' AND p.ownerNo = 'CO93';
```

- “Connection graph” relasi tidak sepenuhnya terhubung, sehingga query tidak dirumuskan dengan benar.
- Dalam hal ini kita hilangkan join condition ($v.propertyNo = p.propertyNo$) dari predikat.
- Perhatikan query berikut:



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Example 21.2 - Checking Semantic Correctness

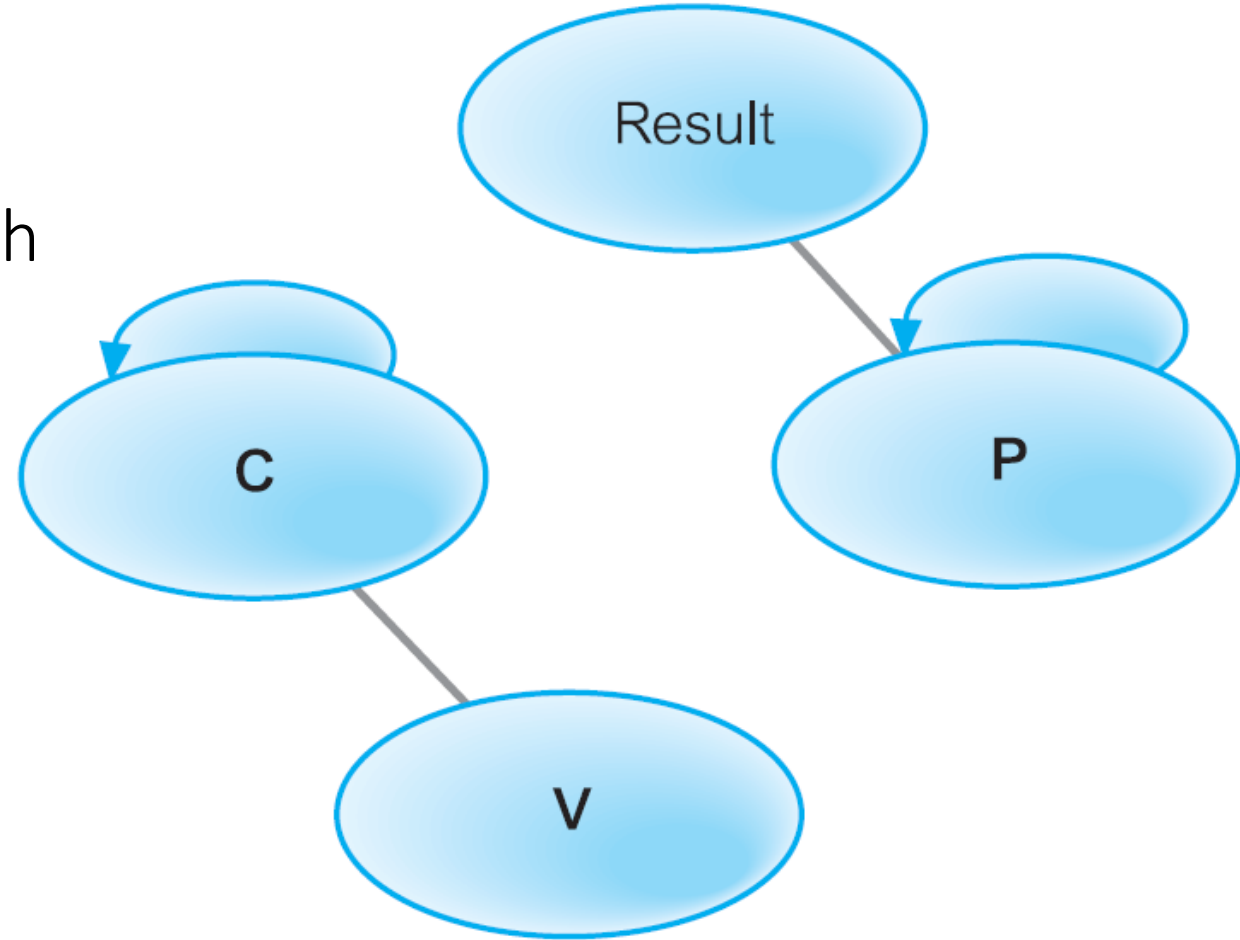
```
SELECT p.propertyNo, p.street  
FROM Client c, Viewing v, PropertyForRent p  
WHERE c.maxRent > 500 AND c.clientNo = v.clientNo AND  
v.propertyNo = p.propertyNo AND c.prefType = 'Flat' AND c.maxRent < 200;
```

- “Connection graph” atribut yang dinormalisasi untuk query ini lihat Gambar 21.3 (b),
- Ada siklus antara node c.maxRent dan 0 dengan nilai negative,
- Tidak ada client dapat memiliki sewa lebih besar dari £ 500 dan kurang dari £ 200,
- Maka query ini adalah kontradiktif.



Figure 21.3.

(a) Relation connection graph showing query is incorrectly formulated;



UNTAR
Universitas Tarumanagara

Terakreditasi
BAN PT

A
linggih

QS STARS
RATING SYSTEM
2019

AMBA
AMBA
AMBA

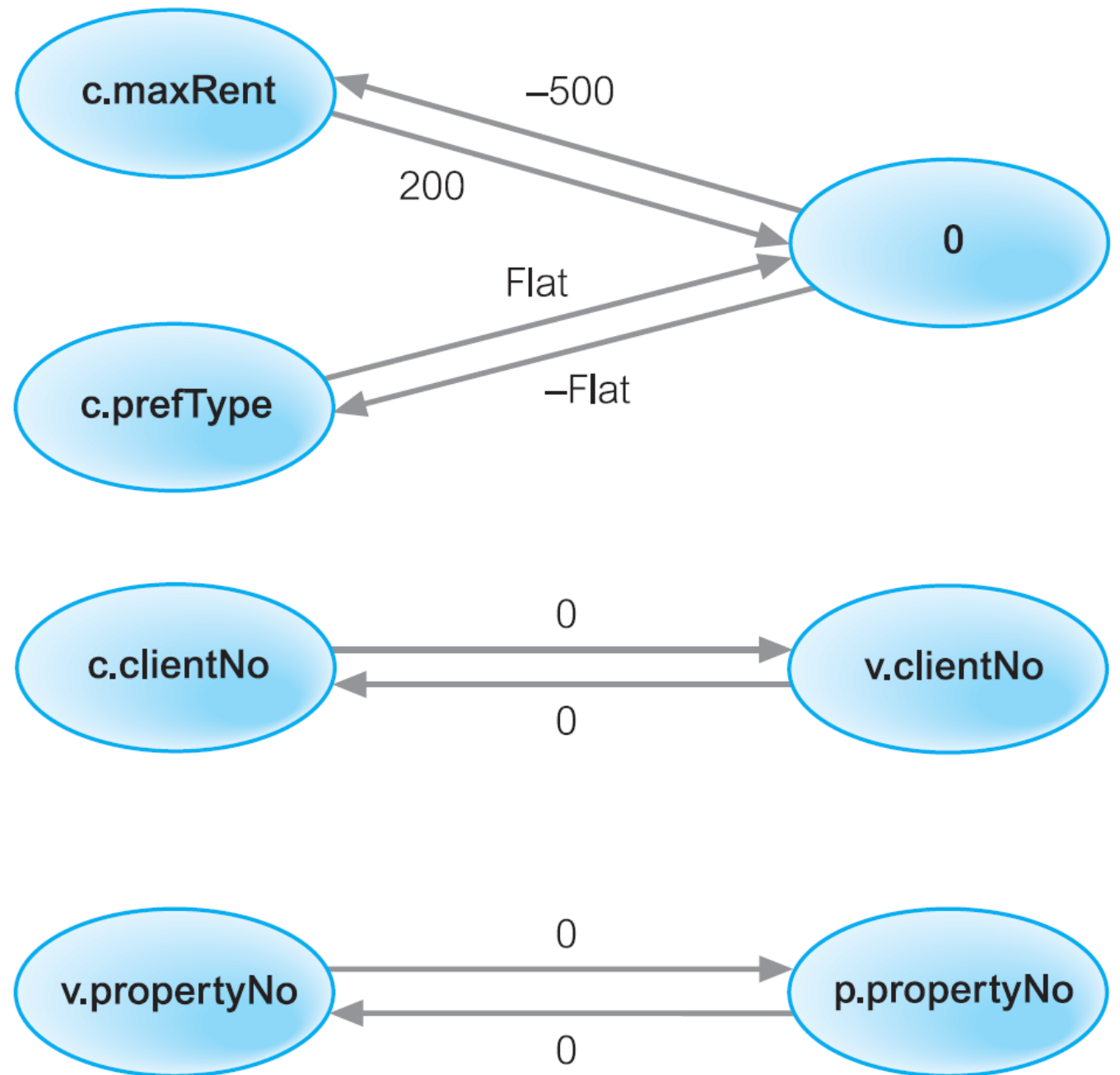
CPA
AUSTRALIA

ICAEW
CHARTERED
ACCOUNTANTS

UNTAR untuk INDONESIA

Figure 21.3.

(b) Normalized attribute connection graph showing query is contradictory.



(b)

Simplification

- Tujuan dari penyederhanaan (simplification):
 - detects redundant qualifications,
 - eliminates common sub-expressions,
 - transforms query to semantically equivalent but more easily and efficiently computed form.
- Biasanya, pembatasan akses, definisi tampilan, dan batasan integritas yang dapat menyebabkan redundansi
- Misalnya, pengguna memiliki hak akses yang sesuai, pengoptimalan awal adalah menerapkan aturan idempotency aljabar Boolean:



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Simplification

$$p \wedge (p) \equiv p$$

$$p \wedge \text{false} \equiv \text{false}$$

$$p \wedge \text{true} \equiv p$$

$$p \wedge (\sim p) \equiv \text{false}$$

$$p \wedge (p \vee q) \equiv p$$

$$p \vee (p) \equiv p$$

$$p \vee \text{false} \equiv p$$

$$p \vee \text{true} \equiv \text{true}$$

$$p \vee (\sim p) \equiv \text{true}$$

$$p \vee (p \wedge q) \equiv p$$



Simplification

Berikut ini definisi view dan query pada view:

```
CREATE VIEW Staff3 AS
```

```
SELECT staffNo, fName, lName, salary, branchNo
```

```
FROM Staff
```

```
WHERE branchNo = 'B003';
```

```
SELECT *
```

```
FROM Staff3
```

```
WHERE (branchNo = 'B003' AND  
salary > 20000);
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Simplification

Query ini bisa disederhanakan menjadi:

```
SELECT staffNo, fName, lName, salary, branchNo
```

```
FROM Staff
```

```
WHERE (branchNo = 'B003' AND salary > 20000) AND branchNo = 'B003';
```

and the **WHERE** condition reduces to (branchNo = 'B003' **AND** salary > 20000).



UNTAR
Universitas Tarumanagara

Terakreditasi
BAN-PT

A
unggul

QS STARS
RATING SYSTEM
2019

AMBA
ACCREDITED

EFMD
EQUIS

CPA
AUSTRALIA

ICAEW
CHARTERED
ACCOUNTANTS

UNTAR untuk INDONESIA

Simplification

```
CREATE ASSERTION OnlyManagerSalaryHigh  
CHECK ((position <> 'Manager' AND salary < 20000)  
OR (position = 'Manager' AND salary > 20000));
```

and consider the effect on the query:

```
SELECT *  
FROM Staff  
WHERE (position = 'Manager' AND salary < 15000);
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Query restructuring

Merupakan tahap akhir dekomposisi query dengan menyusun ulang query yang lebih efisien dalam implementasinya.



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

- Transformasi, mengubah satu ekspresi RA menjadi ekspresi setara secara optimal yang efisien.

Conjunctive Selection operations can cascade into individual Selection operations (and vice versa).

$$\sigma_{p \wedge q \wedge r}(R) = \sigma_p(\sigma_q(\sigma_r(R)))$$

This transformation is sometimes referred to as *cascade of selection*. For example:

$$\sigma_{\text{branchNo}='B003' \wedge \text{salary} > 15000}(\text{Staff}) = \sigma_{\text{branchNo}='B003'}(\sigma_{\text{salary} > 15000}(\text{Staff}))$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Commutativity of Selection operations.

$$\sigma_p(\sigma_q(R)) = \sigma_q(\sigma_p(R))$$

For example:

$$\sigma_{\text{branchNo}='B003'}(\sigma_{\text{salary}>15000}(\text{Staff})) = \sigma_{\text{salary}>15000}(\sigma_{\text{branchNo}='B003'}(\text{Staff}))$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

In a sequence of Projection operations, only the last in the sequence is required.

$$\Pi_L \Pi_M \dots \Pi_N(R) = \Pi_L(R)$$

For example:

$$\Pi_{IName} \Pi_{branchNo, IName}(Staff) = \Pi_{IName}(Staff)$$



Transformation Rules for RA Operations

Commutativity of Selection and Projection.

- Jika predikat p hanya melibatkan atribut dalam projection, maka operasi selection dan projection bersifat komutatif.

$$\Pi_{A_1, \dots, A_m}(\sigma_p(R)) = \sigma_p(\Pi_{A_1, \dots, A_m}(R)) \text{ where } p \in \{A_1, A_2, \dots, A_m\}$$

- Contoh:

$$\Pi_{fName, lName}(\sigma_{lName='Beech'}(Staff)) = \sigma_{lName='Beech'}(\Pi_{fName, lName}(Staff))$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Commutativity of Theta join (and Cartesian product).

$$R \bowtie_p S = S \bowtie_p R$$

$$R \times S = S \times R$$

Equijoin dan Natural join adalah kasus khusus Theta join, maka aturan ini juga berlaku untuk operasi Join ini. Misalnya, menggunakan Equijoin of Staff and Branch:

$$\text{Staff} \bowtie_{\text{Staff.branchNo}=\text{Branch.branchNo}} \text{Branch} = \text{Branch} \bowtie_{\text{Staff.branchNo}=\text{Branch.branchNo}} \text{Staff}$$



Transformation Rules for RA Operations

Commutativity of Selection and Theta join (or Cartesian product).

- Jika predikat selection hanya melibatkan atribut dari salah satu relasi yang join kan, maka operasi Selection and Join (or Cartesian product) bersifat komutatif.

$$\sigma_p(R \bowtie_r S) = (\sigma_p(R)) \bowtie_r S$$

$$\sigma_p(R \times S) = (\sigma_p(R)) \times S \quad \text{where } p \in \{A_1, A_2, \dots, A_n\}$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Commutativity of Selection and Theta join (or Cartesian product).

- Kemungkinan lain, Jika predikat selection adalah predikat conjunctive dari bentuk $(p \wedge q)$, di mana p hanya melibatkan atribut R , dan q hanya melibatkan atribut S , maka operasi Selection dan Theta join bersifat komutatif.

$$\sigma_{p \wedge q}(R \bowtie_r S) = (\sigma_p(R)) \bowtie_r (\sigma_q(S))$$

$$\sigma_{p \wedge q}(R \times S) = (\sigma_p(R)) \times (\sigma_q(S))$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Commutativity of Selection and Theta join (or Cartesian product).

For example:

$$\sigma_{\text{position}='Manager' \wedge \text{city}='London'}(\text{Staff} \bowtie_{\text{Staff.branchNo}=\text{Branch.branchNo}} \text{Branch}) = \\ (\sigma_{\text{position}='Manager'}(\text{Staff})) \bowtie_{\text{Staff.branchNo}=\text{Branch.branchNo}} (\sigma_{\text{city}='London'}(\text{Branch}))$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Commutativity of Projection and Theta join (or Cartesian product).

- Diketahui projection seperti berikut: $L = L_1 \cup L_2$, di mana L_1 hanya memiliki atribut dari R , dan L_2 hanya memiliki atribut dari S , asalkan join condition hanya berisi atribut dari L , Projection dan Theta join bersifat komutatif:

$$\Pi_{L_1 \cup L_2}(R \bowtie_r S) = (\Pi_{L_1}(R)) \bowtie_r (\Pi_{L_2}(S))$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Commutativity of Projection and Theta join (or Cartesian product).

For example:

$$\begin{aligned} &\Pi_{\text{position, city, branchNo}}(\text{Staff} \bowtie_{\text{Staff.branchNo}=\text{Branch.branchNo}} \text{Branch}) = \\ &(\Pi_{\text{position, branchNo}}(\text{Staff})) \bowtie_{\text{Staff.branchNo}=\text{Branch.branchNo}} (\Pi_{\text{city, branchNo}}(\text{Branch})) \end{aligned}$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Commutativity of Projection and Theta join (or Cartesian product).

- Jika kondisi join berisi atribut tambahan bukan di L, misal atribut $M = M_1 \cup M_2$ dimana M_1 hanya berasal dari R, dan M_2 berasal dari S, maka operasi Projection yang diperlukan adalah:

$$\Pi_{L_1 \cup L_2}(R \bowtie_R S) = \Pi_{L_1 \cup L_2}(\Pi_{L_1 \cup M_1}(R)) \bowtie_R (\Pi_{L_2 \cup M_2}(S))$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Commutativity of Projection and Theta join (or Cartesian product).

For example:

$$\Pi_{\text{position, city}}(\text{Staff} \bowtie_{\text{Staff.branchNo}=\text{Branch.branchNo}} \text{Branch}) =$$

$$\Pi_{\text{position, city}}((\Pi_{\text{position, branchNo}}(\text{Staff})) \bowtie_{\text{Staff.branchNo}=\text{Branch.branchNo}} (\Pi_{\text{city, branchNo}}(\text{Branch})))$$



UNTAR
Universitas Tarumanagara

Terakreditasi
BAN-PT

A
Linggi

QS
STARS
RATING SYSTEM
2019

GLAS
UKAS

IABEE

CPA
AUSTRALIA

ICAEW
CHARTERED
ACCOUNTANTS

UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Commutativity of Union and Intersection (but not set difference).

$$R \cup S = S \cup R$$

$$R \cap S = S \cap R$$



Transformation Rules for RA Operations

Commutativity of Selection and set operations (Union, Intersection, and Set difference).

$$\sigma_p(R \cup S) = \sigma_p(S) \cup \sigma_p(R)$$

$$\sigma_p(R \cap S) = \sigma_p(S) \cap \sigma_p(R)$$

$$\sigma_p(R - S) = \sigma_p(S) - \sigma_p(R)$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Commutativity of Projection and Union.

$$\Pi_L(R \cup S) = \Pi_L(S) \cup \Pi_L(R)$$

Associativity of Union and Intersection (but not Set difference).

$$(R \cup S) \cup T = S \cup (R \cup T)$$

$$(R \cap S) \cap T = S \cap (R \cap T)$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Associativity of Theta join (and Cartesian product).

- Cartesian product and Natural join are always associative:

$$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$$

$$(R \times S) \times T = R \times (S \times T)$$

- Jika kondisi join q hanya melibatkan atribut dari relasi S dan T , maka Theta join bersifat associative.



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Associativity of Theta join (and Cartesian product).

- Cartesian product and Natural join are always associative:

$$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$$

$$(R \times S) \times T = R \times (S \times T)$$

- Jika kondisi join q hanya melibatkan atribut dari relasi S dan T , maka Theta join bersifat associative seperti berikut.

$$(R \bowtie_p S) \bowtie_{q \wedge r} T = R \bowtie_{p \wedge r} (S \bowtie_q T)$$



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Transformation Rules for RA Operations

Associativity of Theta join (and Cartesian product).

For example:

$$\begin{aligned} & (\text{Staff} \bowtie_{\text{Staff.staffNo}=\text{PropertyForRent.staffNo}} \text{PropertyForRent}) \\ & \quad \bowtie_{\text{ownerNo}=\text{Owner.ownerNo} \wedge \text{Staff.IName}=\text{Owner.IName}} \text{Owner} \\ &= \text{Staff} \bowtie_{\text{Staff.staffNo}=\text{PropertyForRent.staffNo} \wedge \text{Staff.IName}=\text{IName}} \\ & \quad (\text{PropertyForRent} \bowtie_{\text{ownerNo}} \text{Owner}) \end{aligned}$$



Example 21.3 Use of Transformation Rules

For prospective renters of flats, find properties that match requirements and owned by CO93.

We can write this query in SQL as:

```
SELECT p.propertyNo, p.street
FROM Client c, Viewing v, PropertyForRent p
WHERE c.prefType = 'Flat' AND c.clientNo = v.clientNo AND
      v.propertyNo = p.propertyNo AND c.maxRent >= p.rent AND
      c.prefType = p.type AND p.ownerNo = 'CO93';
```



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Example 21.3 Use of Transformation Rules

Berdasarkan contoh (SQL) ini, kita asumsikan bahwa ada lebih sedikit properti yang dimiliki oleh pemilik CO93 daripada calon penyewa yang telah menentukan jenis propertinya Flat. Kita ubah SQL nya menjadi RA seperti berikut:

$$\begin{aligned} &\Pi_{p.\text{propertyNo}, p.\text{street}} \left(\sigma_{c.\text{prefType}='Flat' \wedge c.\text{clientNo}=v.\text{clientNo}} \right. \\ &\quad \wedge v.\text{propertyNo}=p.\text{propertyNo} \wedge c.\text{maxRent} \geq p.\text{rent} \wedge c.\text{prefType}=p.\text{type} \wedge \\ &\quad \left. p.\text{ownerNo}='CO93', ((c \times v) \times p) \right) \end{aligned}$$



Example 21.3 Use of Transformation Rules

- Representasi querynya seperti pada Gambar 21.4(a), sebagai pohon aljabar relasional kanonik (canonical relational algebra tree).
- Untuk meningkatkan efisiensi strategi eksekusi, kita gunakan aturan transformasi berikut:



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Example 21.3 Use of Transformation Rules

- Aturan 1, memisahkan conjunction dari operasi Selection menjadi operasi Selection individual.
- Aturan 2 dan Aturan 6, susun kembali operasi Selection dan kemudian melakukan perubahan Selections and Cartesian products.
- Hasil dari dua langkah pertama ini ditunjukkan pada Gambar 21.4 (b).

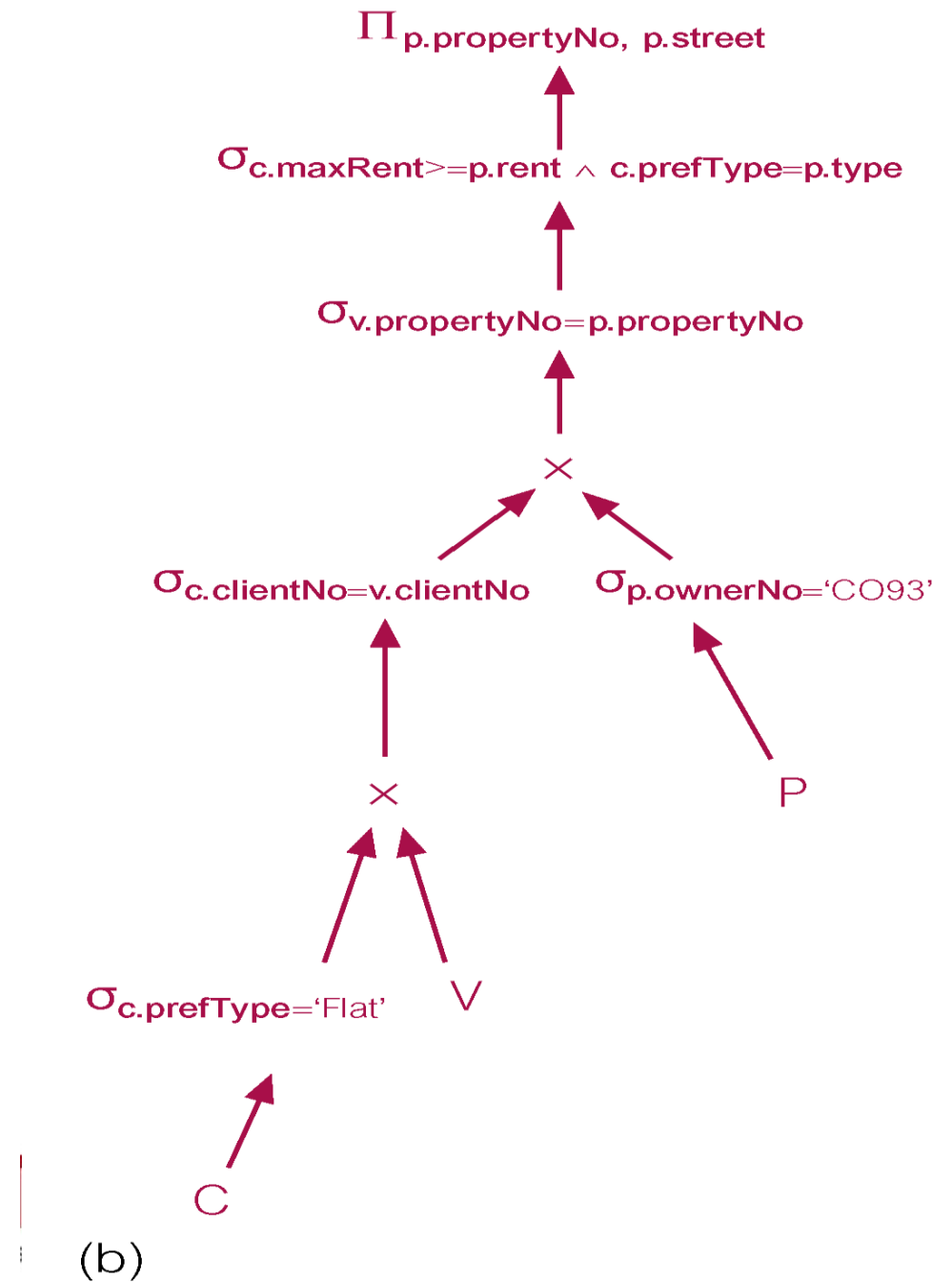
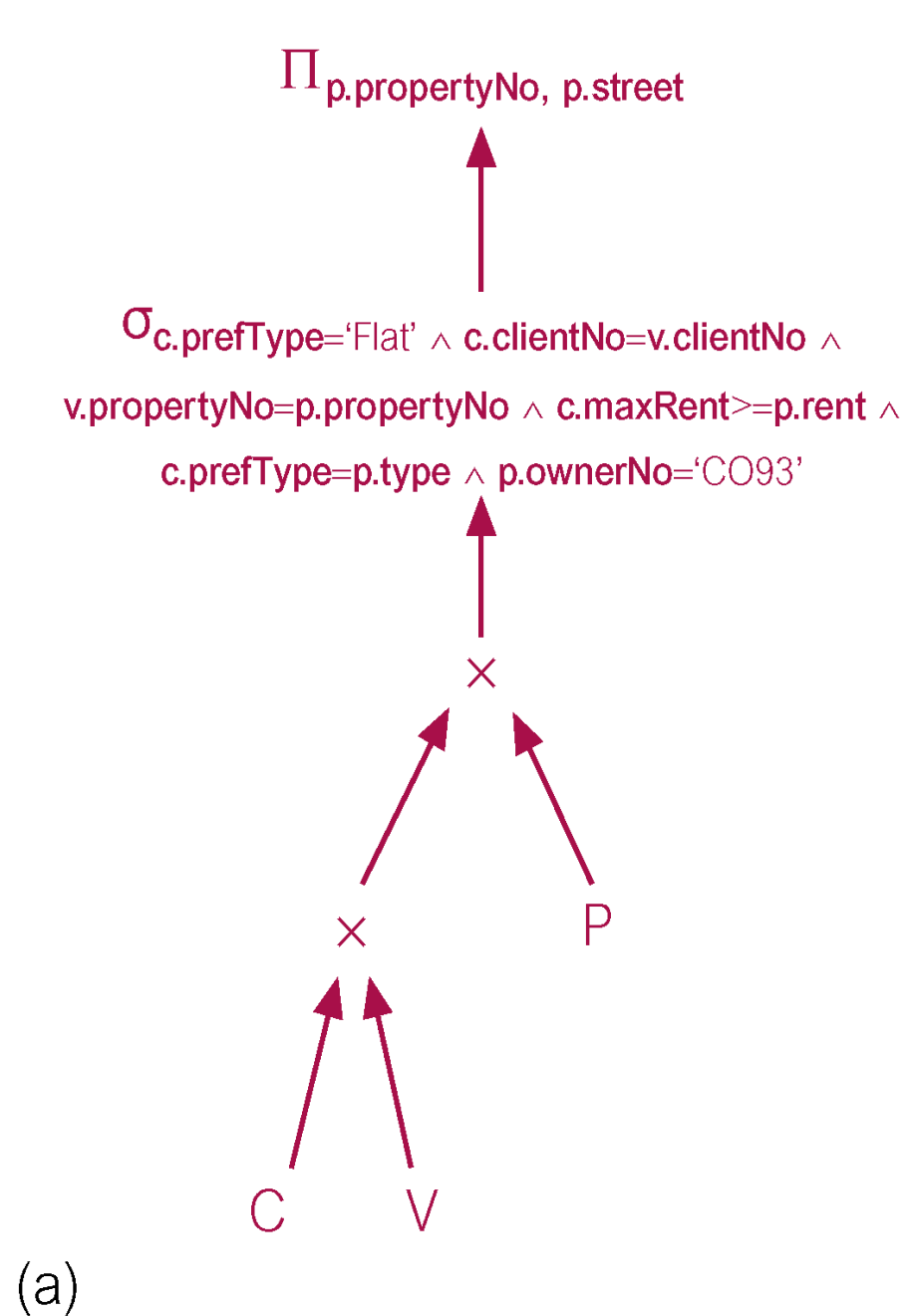


UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Figure 21.4.
Relational
algebra tree
for Example
21.3



Example 21.3 Use of Transformation Rules

- Operasi Selection dengan predikat Equijon dan Cartesian product dapat disederhanakan dengan operasi Equijoin:

$$\sigma_{R.a=S.b}(R \times S) = R \bowtie_{R.a=S.b} S$$

- Hasil dari langkah ini ditunjukkan di Gambar 21.4 (c).
- Aturan 11, untuk menyusun ulang Equijoins, sehingga adanya selection yang lebih ketat pada (p.ownerNo = 'CO93') dilakukan pertama kali, seperti yang ditunjukkan pada Gambar 21.4 (d)



Figure 21.4.
Relational
algebra tree
for Example
21.3

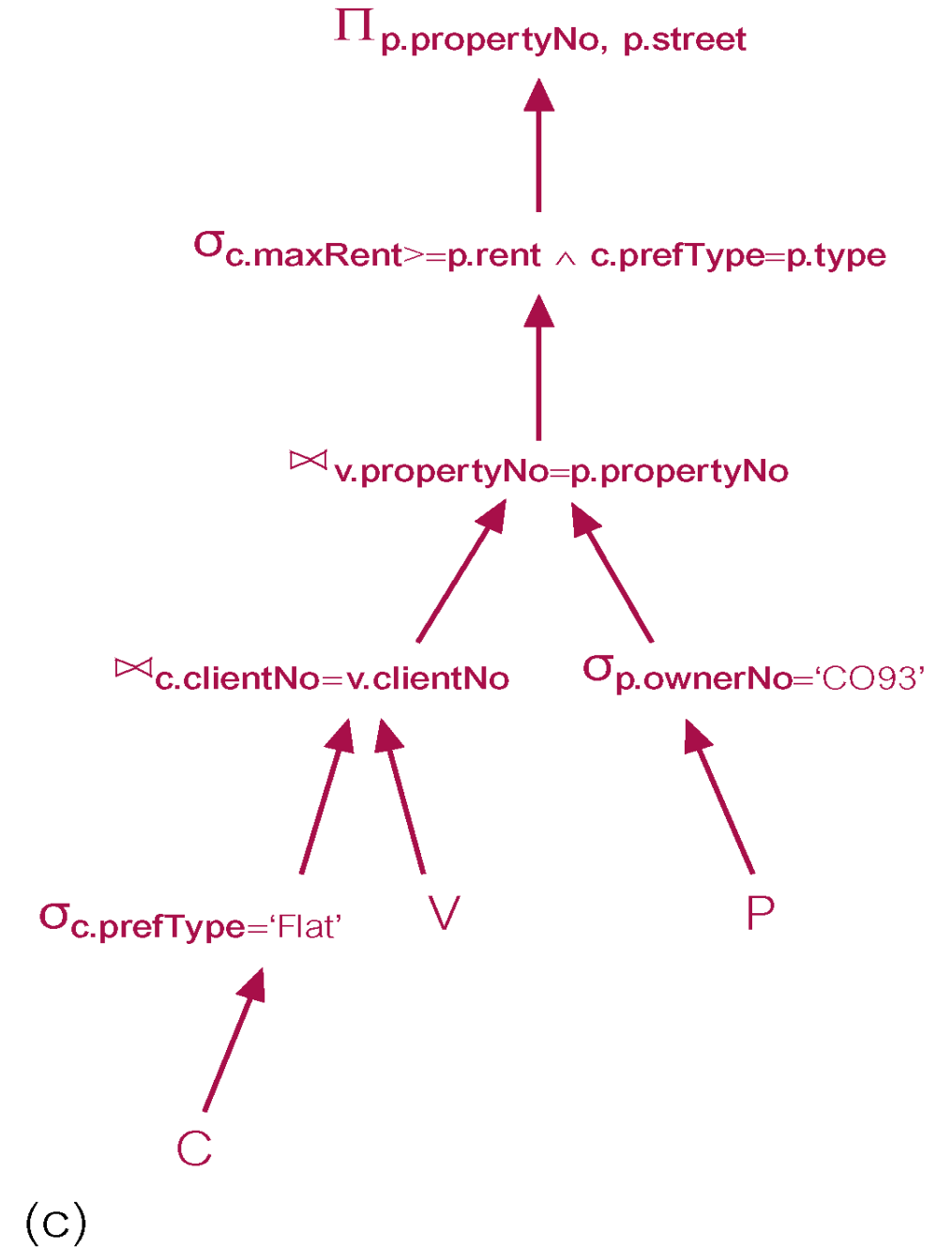
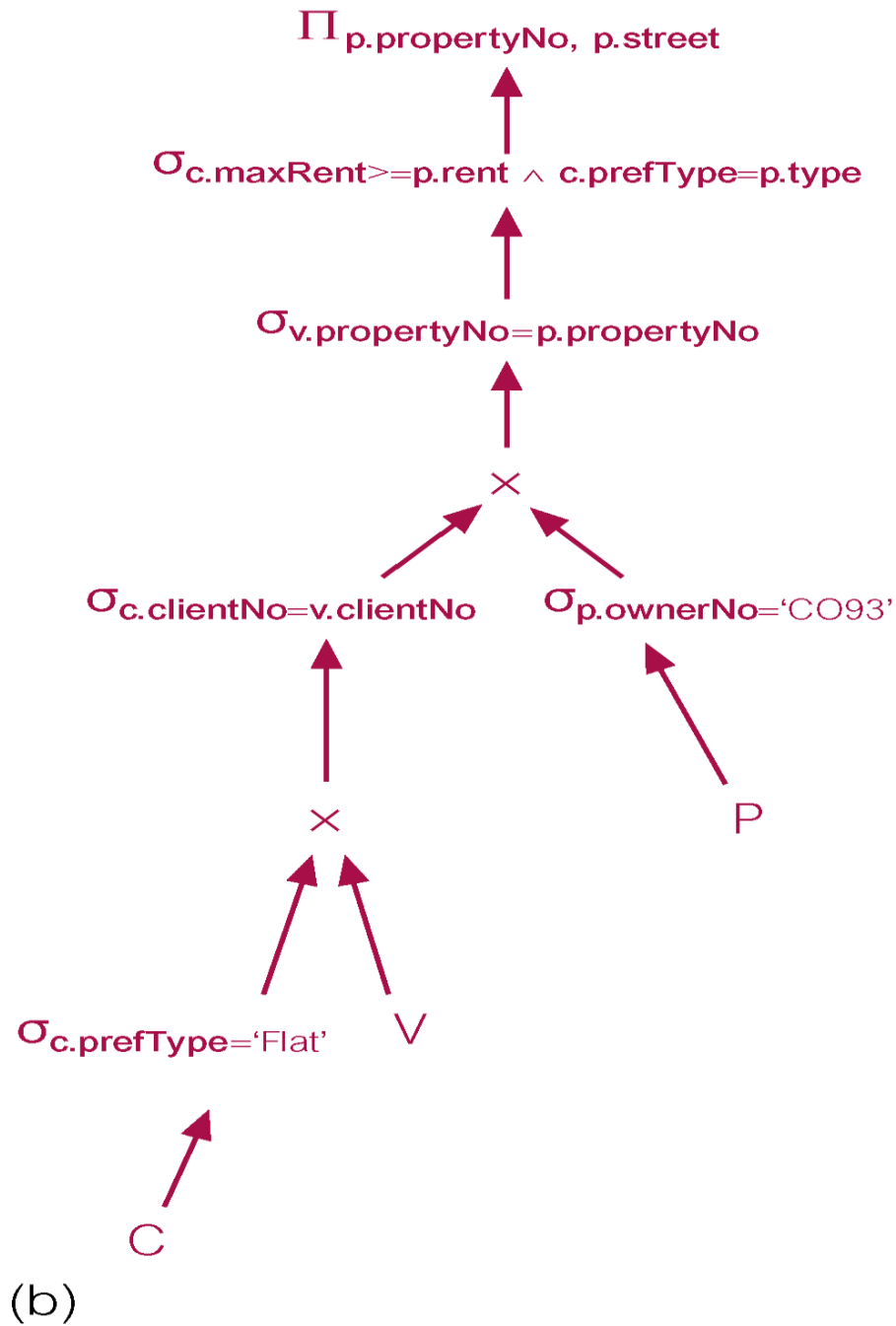
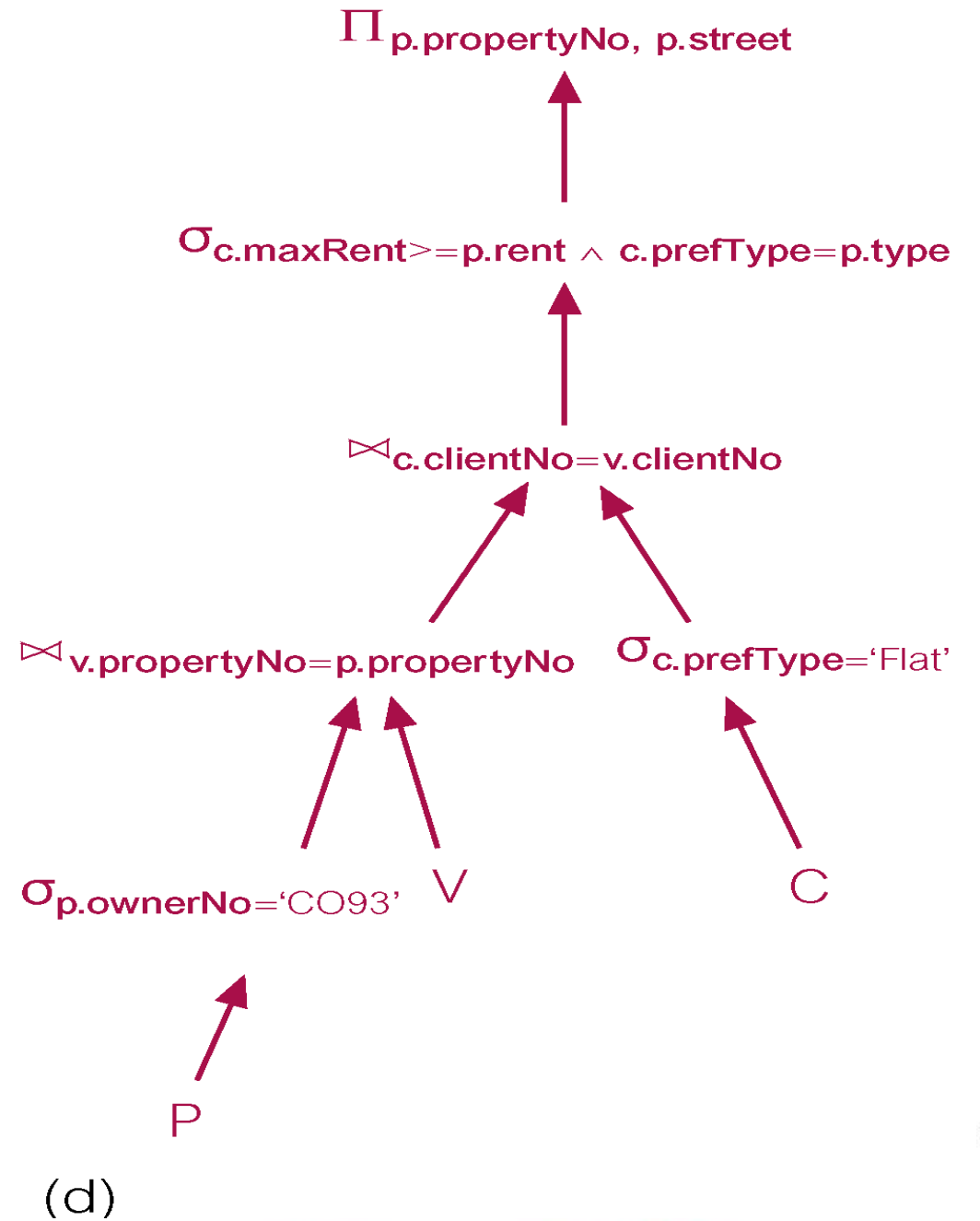
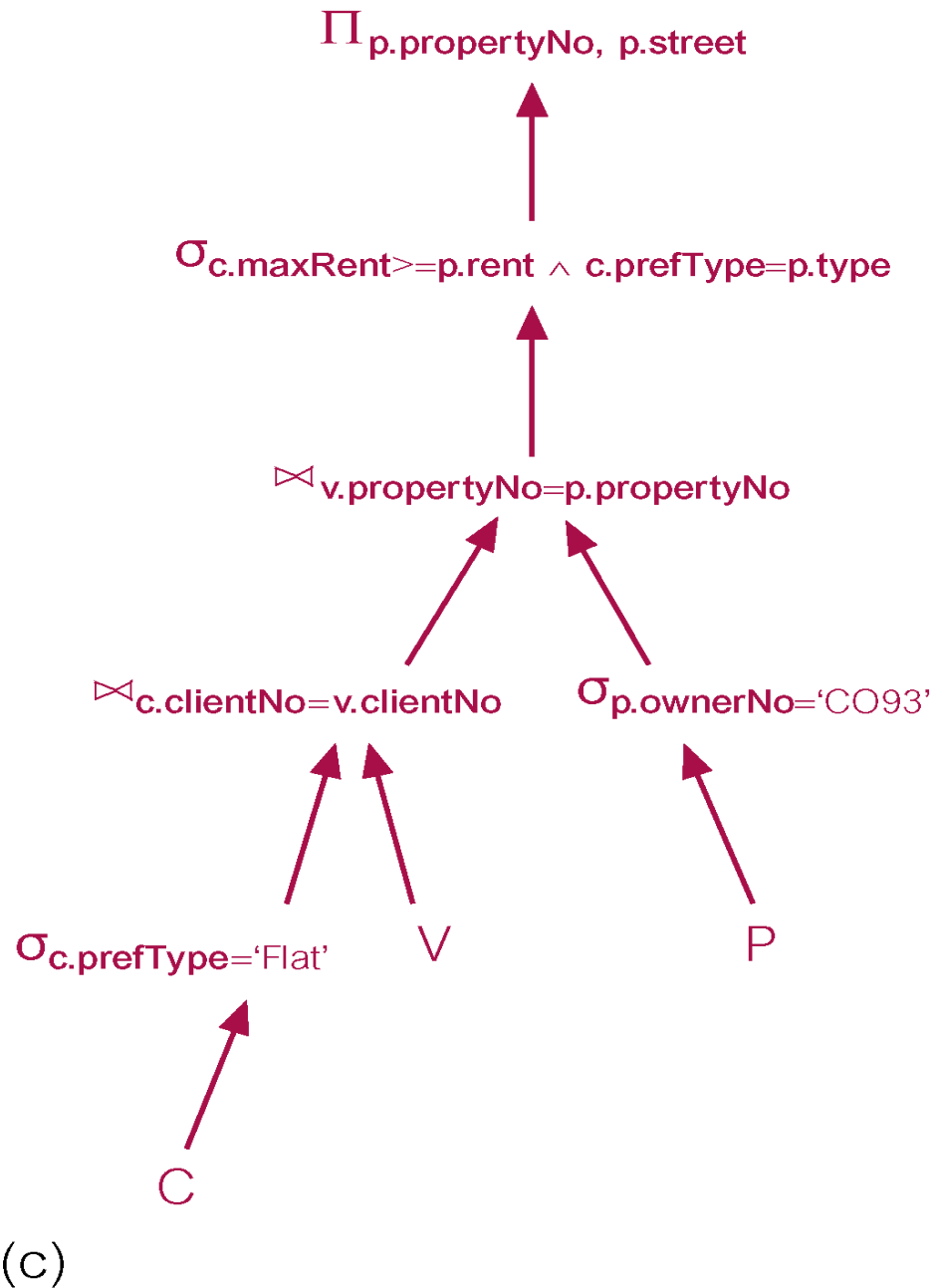


Figure 21.4.
Relational
algebra tree
for Example
21.3



Example 21.3 Use of Transformation Rules

- Aturan 4 dan 7, untuk memindahkan Projection ke bagian bawah setelah Equijoins, dan buat operasi Projection baru yang sesuai. Hasil ditunjukkan di Gambar 21.4 (e).
- Optimasi tambahan dalam contoh khusus ini bahwa operasi Selection ($c.prefType = p.type$) dapat disederhanakan menjadi ($p.type = 'Flat'$), $c.prefType = 'Flat'$ diketahui dari predikat klausa pertama. Menggunakan substitusi ini, kita tempatkan Selection pada tree paling bawah.
- Hasilnya adalah RA tree yang paling optimal seperti Gambar 21.4 (f).



Figure 21.4.
Relational
algebra tree
for Example
21.3

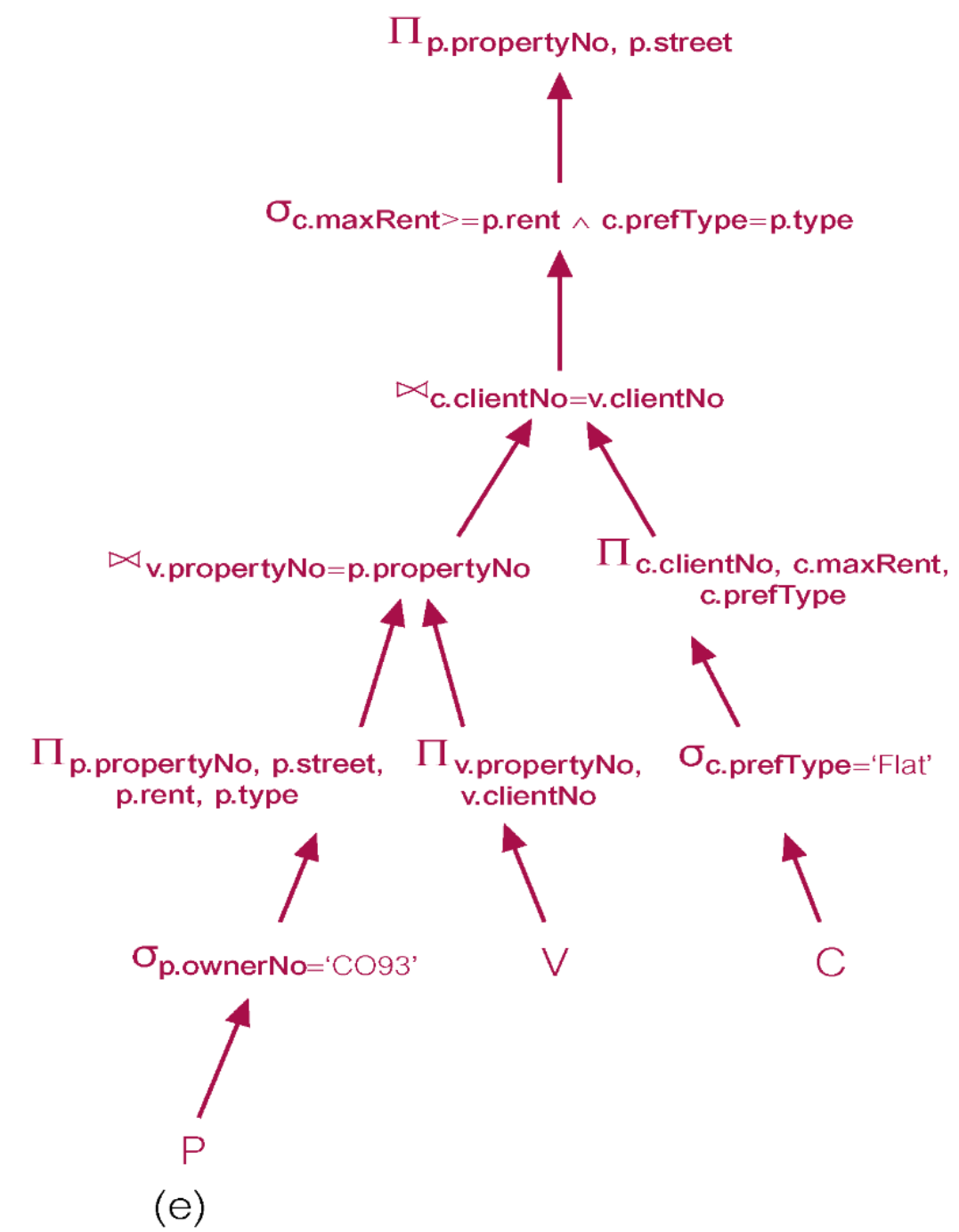
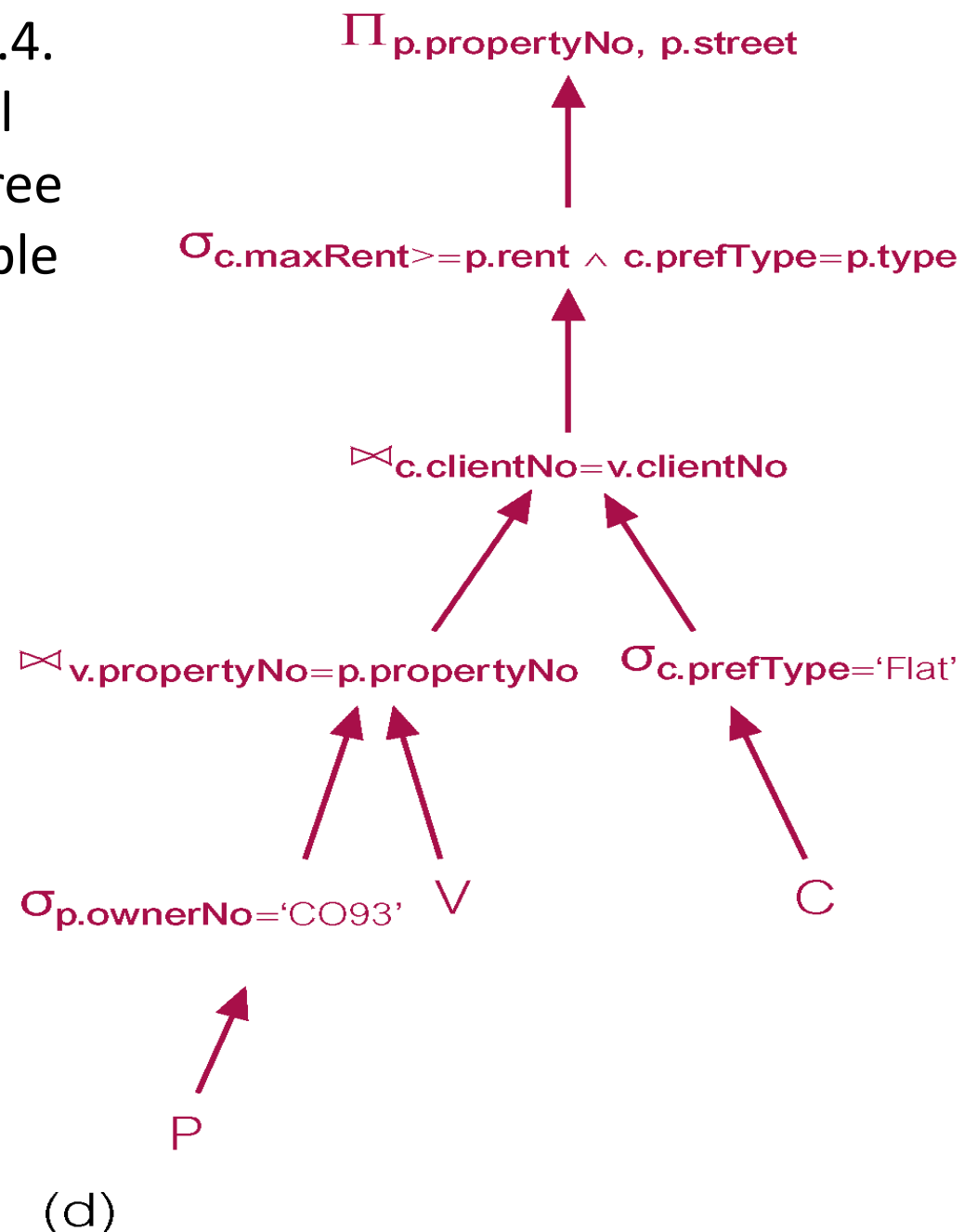
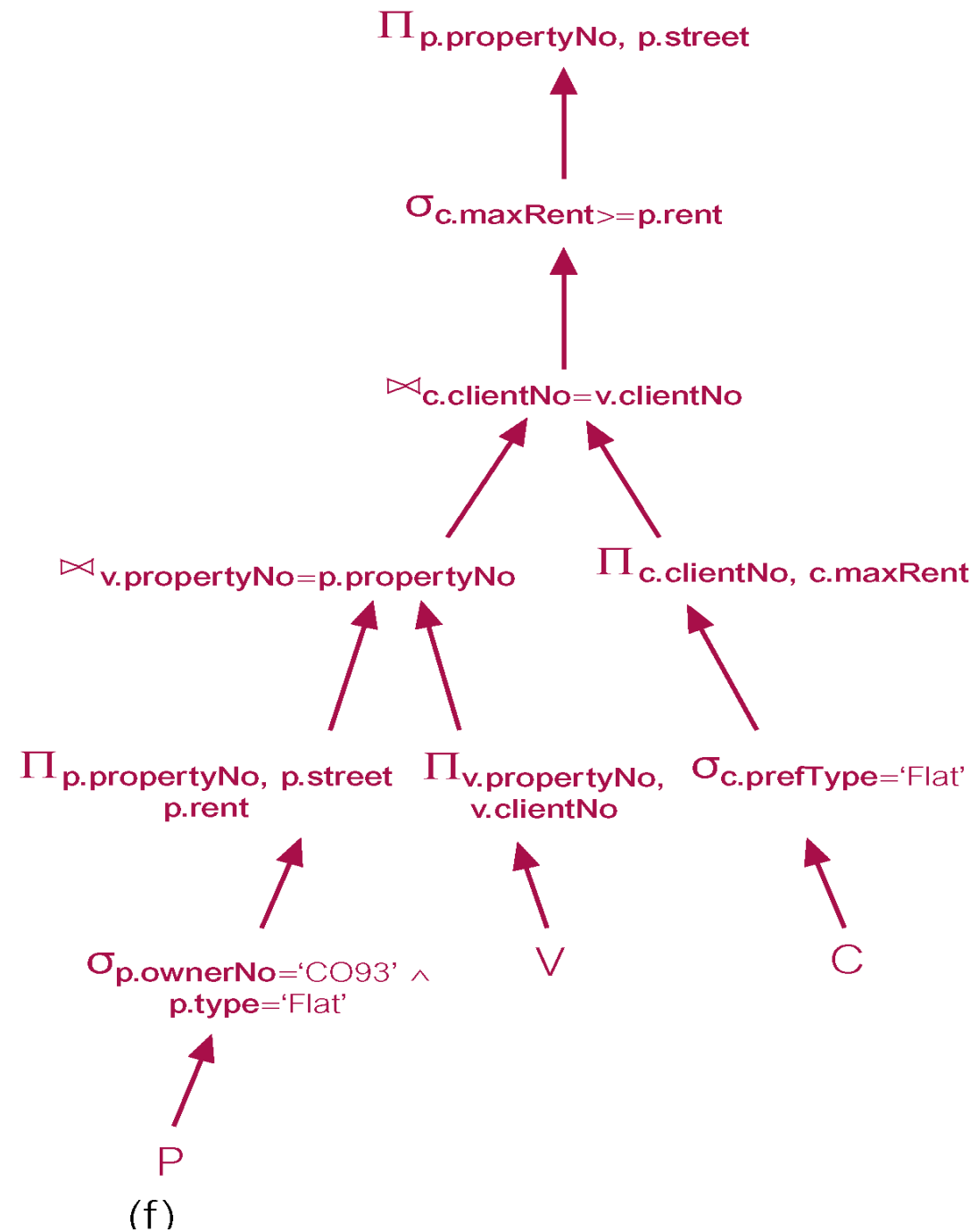
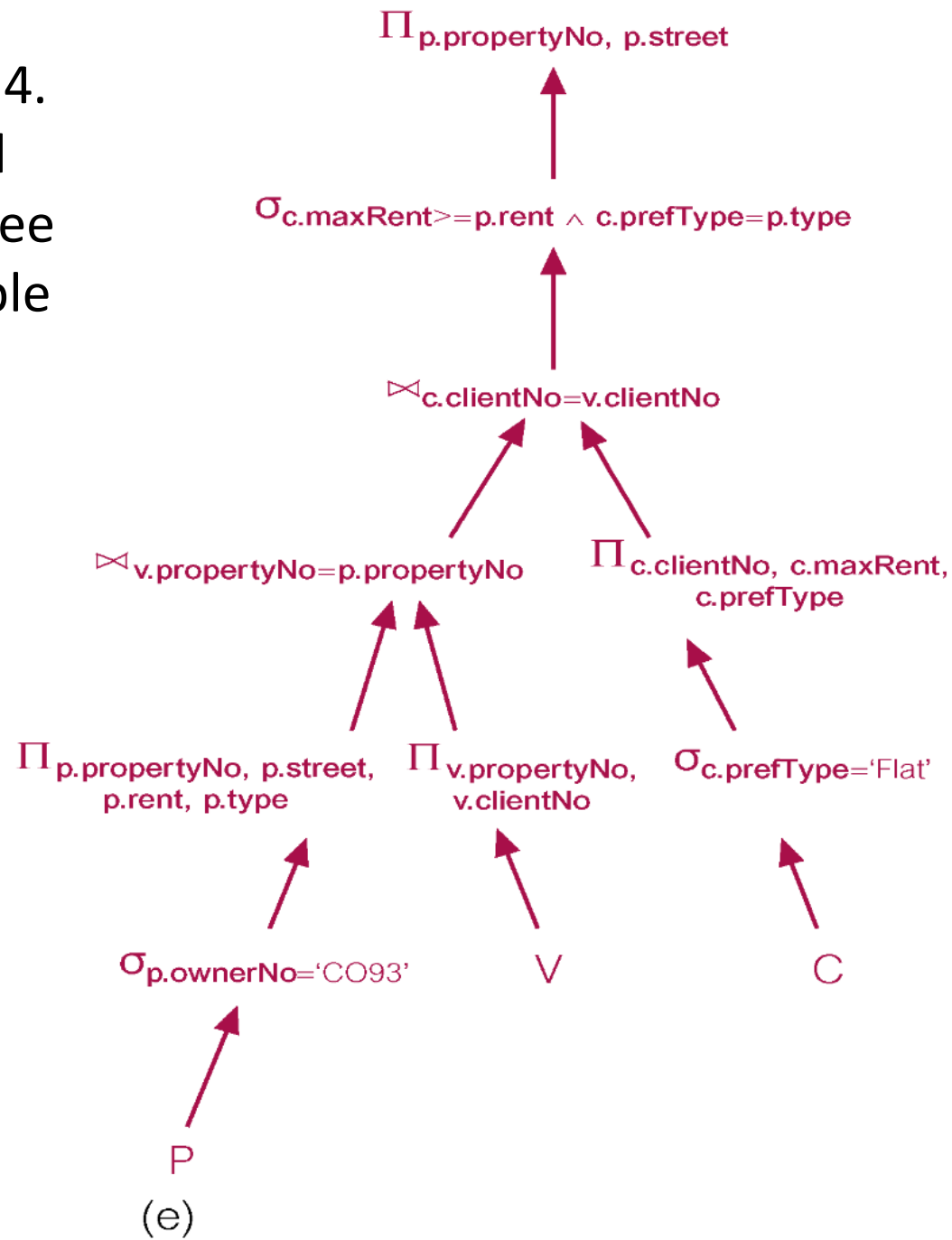


Figure 21.4.
Relational
algebra tree
for Example
21.3



Exercises

1. 21.16 Calculate the cost of the three strategies cited in Example 21.1 if the Staff relation has 10 000 tuples, Branch has 500 tuples, there are 500 Managers (one for each branch), and there are 10 London branches.
2. Menggunakan Hotel schema seperti berikut ini (Chapter 3 matakuliah Database Systems), tentukan apakah query berikut ini benar secara semantic.



Tugas dipresentasikan (bagian nilai UAS)

1. Menggunakan schema Hotel seperti berikut ini (Chapter 3 matakuliah Database Systems), tentukan apakah query berikut ini benar secara semantic.

- (b) **SELECT** g.guestNo, g.name
FROM Hotel h, Booking b, Guest g
WHERE h.hotelNo = b.hotelNo **AND** h.hotelName = 'Grosvenor Hotel';
- (c) **SELECT** r.roomNo, h.hotelNo
FROM Hotel h, Booking b, Room r
WHERE h.hotelNo = b.hotelNo **AND** h.hotelNo = 'H21' **AND** b.roomNo = r.roomNo **AND**
type = 'S' **AND** b.hotelNo = 'H22';



Tugas dipresentasikan (bagian nilai UAS)

The following tables form part of a database held in a relational DBMS:

Hotel (hotelNo, hotelName, city)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

Guest (guestNo, guestName, guestAddress)

where Hotel contains hotel details and hotelNo is the primary key;

Room contains room details for each hotel and (roomNo, hotelNo) forms the primary key;

Booking contains details of bookings and (hotelNo, guestNo, dateFrom) forms the primary key;

Guest contains guest details and guestNo is the primary key.



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Tugas dipresentasikan (bagian nilai UAS)

2. Menggunakan schema Hotel, gambarkan pohon aljabar relasi (relational algebra tree) untuk setiap query berikut dengan mentransformasikan query ke bentuk yang lebih efisien. Berikan penjelasan setiap langkahnya.

(a) **SELECT** r.roomNo, r.type, r.price
FROM Room r, Booking b, Hotel h
WHERE r.roomNo = b.roomNo **AND** b.hotelNo = h.hotelNo **AND**
h.hotelName = 'Grosvenor Hotel' **AND** r.price > 100;

(b) **SELECT** g.guestNo, g.guestName
FROM Room r, Hotel h, Booking b, Guest g
WHERE h.hotelNo = b.hotelNo **AND** g.guestNo = b.guestNo **AND** h.hotelNo = r.hotelNo **AND**
h.hotelName = 'Grosvenor Hotel' **AND** dateFrom >= '1-Jan-04' **AND** dateTo <= '31-Dec-04';



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA

Thank You

Reference: Database Systems A Practical Approach to Design, Implementation, and Management Fourth Edition.

Thomas M. Connolly and Carolyn E. Begg



UNTAR
Universitas Tarumanagara



UNTAR untuk INDONESIA