Your Name : Desireddy Sai Sankeerthana
Your Email id : saisanke@buffalo.edu
Your UB person number : 50465523

1) Define terms :- Relation Schema: A relation Schema R is denoted by $R(A_1, A_2, \ldots, A_N)$ is made up of a relation name R and a list of attributes $A_1, A_2 \ldots A_N$). the degree of A relation is the number of attributes N of its relation Schema

- Relational database Schema: database Schema is a set of all the relation Schemas in database
- Domain: In domain each tuple should be atomic i.e some elementary type (int)

Ex: Student (Name : String, id : integer)

- Attribute: an attribute describes members of Entity
  Ex: id can be considered as attribute from above Example
- Tuple: Rows in a table with values are considered as tuple

Ex: Student

Name	id
Sam	123
harry	456

} tuples

- Relational instance: A instance of a relation is a set of tuples also called record in which each tuple has some number of fields as the relation Schema
- Relational Cardinality: it refers to the relationships between the data in two tables. types of Cardinality : one to one, one to many, many to one, many to many
- Relation degree : the degree of the relationship represents the number of entity types that associate in a relationship

Reference Lecture Slides (Relational database)
Slide number → 5, 8, 9, 10.

2(1) :- A suitable Schema for relation class

```
CREATE TABLE Classes( class       varchar(4)
                      type        varchar(2)
                      Country     varchar(12)
                      numGuns     integer
                      bore        integer
                      displacement integer );
```

(2) A Suitable Schema for relation Ships

```
CREATE TABLE Ships ( name      cHar(15)
                     class     varchar(14)
                     launched  integer      );
```

(3) A Suitable Schema for relation Battles

```
CREATE TABLE Battles ( name    Varchar(15)
                       date    date        );
```

(4) A Suitable Schema for relation outcomes

```
CREATE TABLE Outcomes ( ship    Varchar(16)
                        battle  Varchar(16)
                        result  String      );
```

(5) An alteration to your Classes relation from (a) to delete the attribute bore

```
ALTER TABLE CLASSES DROP bore;
```

(6) An alteration to your ships relation from (b) to include the attribute yard giving the shipyard where the ship was built

```
ALTER TABLE Ships ADD yard CHARC100);
```

Reference textbook (ulman) page no 70.

3) Briefly Explain the significance of the following in representing information in the real world: (a) data definition language (b) data manipulation language and their differences?

(a) data definition language: DDL is a standardized language with Commands to define the storage group different structures and objects in a database. DDL statements Create, modify and remove database objects such as tables, indexes and stogroups. DDL is also used in a generic sense to refer to any language that describes data

DDL Commands: ALTER, CREATE, DROP, TRUNCATE (or) RENAME

Syn: CREATE TABLE [table name] ([column definitions]) [table parameters];

DROP object type objectname;

(b) Data manipulation language (DML): the SQL commands that deals with manipulation of data present in the database belong to Data manipulation language and it includes most of the SQL statements. it is the Component of the SQL statement that controls access to data and to the database. DML Commands: Insert, update, delete, lock, Call., Select

Syntax: Select: use to fetch selected fields from database

```
Select   column_name1, column_name2, ...
from     table_name
WHERE    Condition;
```

Insert: insert is used to insert data records in database table.

insert into table_name (column_name1, column_name2) values (value1, value2)

Difference between data definition language and data manipulation language as follows

DDL	DML
it stands for data definition langu	it stands for data manipulation language
it is used to create database schema and can be used to define some constraints as well	it is used to add, retrieve or update the data
it basically defines the column of the table	it add or update the row of the table these rows are called tuple
it doesn't have any further classification	it is further classified into procedural & Non procedural DML
Basic command present in DDL Create DROP, rename	DML Commands select, update, insert
DDL doesn't use where clause in its statement	DML uses where clause.

Reference: Lecture slides _ Relational database slid No: 15, 16, 17

4) differentiate between Natural join, theta join and self join (with an Example).

- Natural join: A useful join variant connects two relation by Equating all the common attributes of the same name and projecting out one copy of each pair of equated attributes. a tuple that fails to pair with any tuple of other relations in a join is said to be dangling tuple. for more than one Comman attribute tuples need to agree in all the common attributes denoted as $R_3 := R_1 \bowtie R_2$ in which we pair only those tuples from R_1 & R_2 that agree in whatever attributes are common to the schemas of R_1 & R_2

Example: let relation R be

A	B
1	2
3	4

and relations S be

B	C	D
2	5	6
4	7	8
9	10	11

Now the Natural join of $R \bowtie S$ will be

A	B	C	D
1	2	5	6
3	4	7	8

here the only common attribute of R and S is B. thus, to pair successfully tuples need only to agree in their B Components.

- Theta join: it is a natural join that forces us to pair tuples using one specific condition. while this way equating shared attributes is the most basic on which relations are joined is sometimes desirable to pair tuples from two relations ~~to~~ ~~form~~ on some other basis. the notation of theta join of relations R and S based on condition C is $R \bowtie_C S$ the result of this operation is constructed as follows:

  Take the product R and S

  Select from the product only those tuples that satisfy the condition C

Ex: Relation U

A	B	C
1	2	3
6	7	8
9	7	8

Relation V

B	C	D
2	3	4
2	3	5
7	8	10

$U \bowtie V$

A	U.B	U.C	V.B	V.C	D
1	2	3	2	3	4
6	7	8	2	3	5
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	2	3	4
6	7	8	7	8	10
9	7	8	2	3	4
9	7	8	2	3	5
9	7	8	7	8	10

$U \bowtie_{A<D} V$

A	U.B	U.C	V.B	V.C	D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	7	8	10
9	7	8	7	8	10

here the tuples that doesn't satisfy the condition A<D are removed and rest of the tuples are resulted.

Self join :- it allows you to join a table to itself. In self join each row in a table is joined to itself and Every other row in the table Self join alias is used to assign different names to the same table within the query.

Example:

id	fullName	Salary	ManagerId
1	John Smith	10000	3
2	Jane Anderson	12000	3
3	Tom Lanon	15000	4
4	Anne Connor	20000	null
5	Jeremy York	9000	1

← Employee table

```
Select employee.FullName, employee.ManagerId, manager.FullName as ManagerName
From Employees employee
JOIN Employees manager
ON employee.ManagerId = manager.Id;
```

Result :-

Id	FullName	ManagerId	ManagerName
1	John Smith	3	Tom Lanon
2	Jane Anderson	3	Tom Lanon
3	Tom Lanon	4	Anne Connor
5	Jeremy York	1	John Smith

Reference : ULMAN Textbook - CHAPTER 2
SUBtopic 2.4.8 (Natural join) Subtopic 2.4.9 (Theta join)

5(1) : what pc models have speed at most 3.00?

$R_1 := \sigma_{speed \leq 3.00}(pc)$

$R_2 := \Pi_{model}(R_1)$

Π_{model}

|

$\sigma_{speed \leq 3.00}$

|

pc

Result

model

1001	1007	1011
1002	1008	1012
1003	1009	
1004	1010	

(2) find those manufacturers that sell PC's, but not laptops.

$R_1 := \sigma_{type=pc}(product)$

maker	model	type
A	1001	PC
A	1002	PC
A	1003	PC
B	1004	PC
B	1005	PC
B	1006	PC
C	1007	PC
D	1008	PC
D	1009	PC
D	1010	PC
E	1011	PC
E	1012	PC
E	1013	PC

$R_2 := \sigma_{type=laptop}(product)$

maker	model	type
A	2004	laptop
A	2005	laptop
A	2006	laptop
B	2007	laptop
E	2001	laptop
E	2002	laptop
E	2003	laptop
F	2008	laptop
F	2009	laptop
G	2010	laptop

$R_3 := \Pi_{maker}(R_1)$

$R_4 := \Pi_{maker}(R_2)$

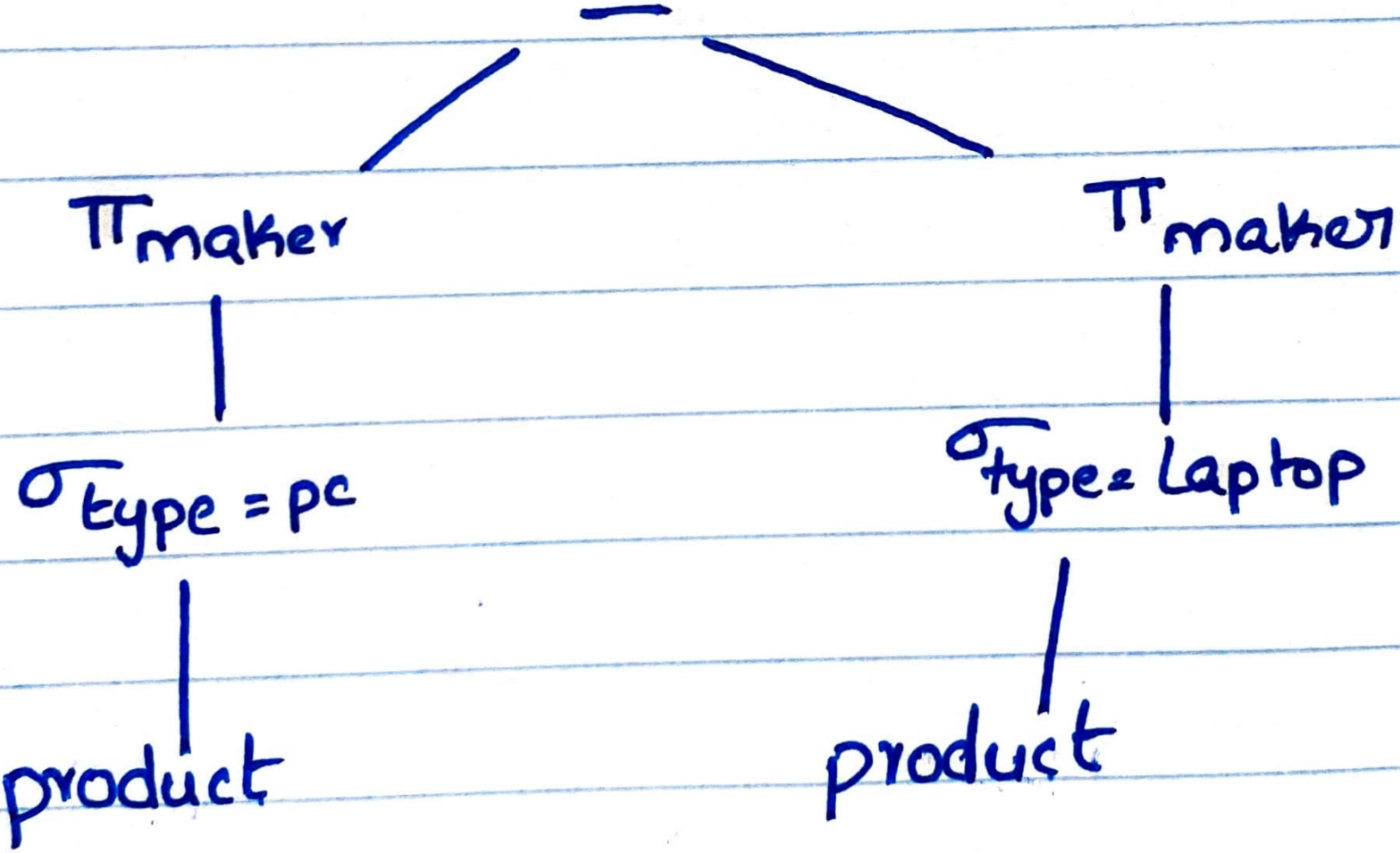
$R_5 := R_3 - R_4$

Result:

B
C
D

maker.

5(2) :


−
π_{maker}
π_{maker}
$\sigma_{type = pc}$
$\sigma_{type = Laptop}$
product
product

(3) find those ram sizes that occur in two (or more laptops

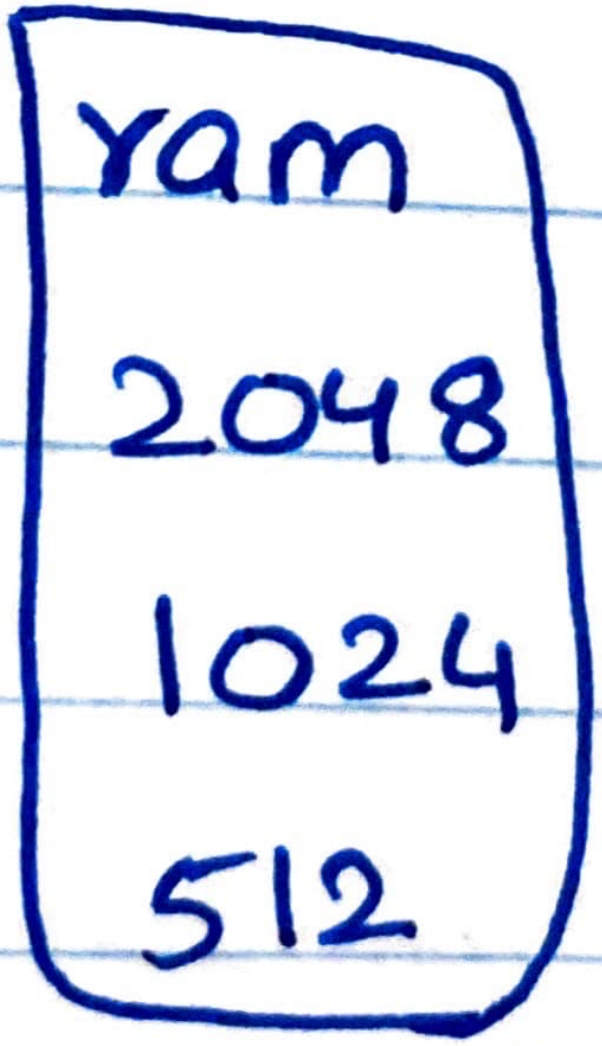
$$R_1 := \rho_{L_1}(\text{Laptop})$$

$$R_2 := \rho_{L_2}(\text{Laptop})$$

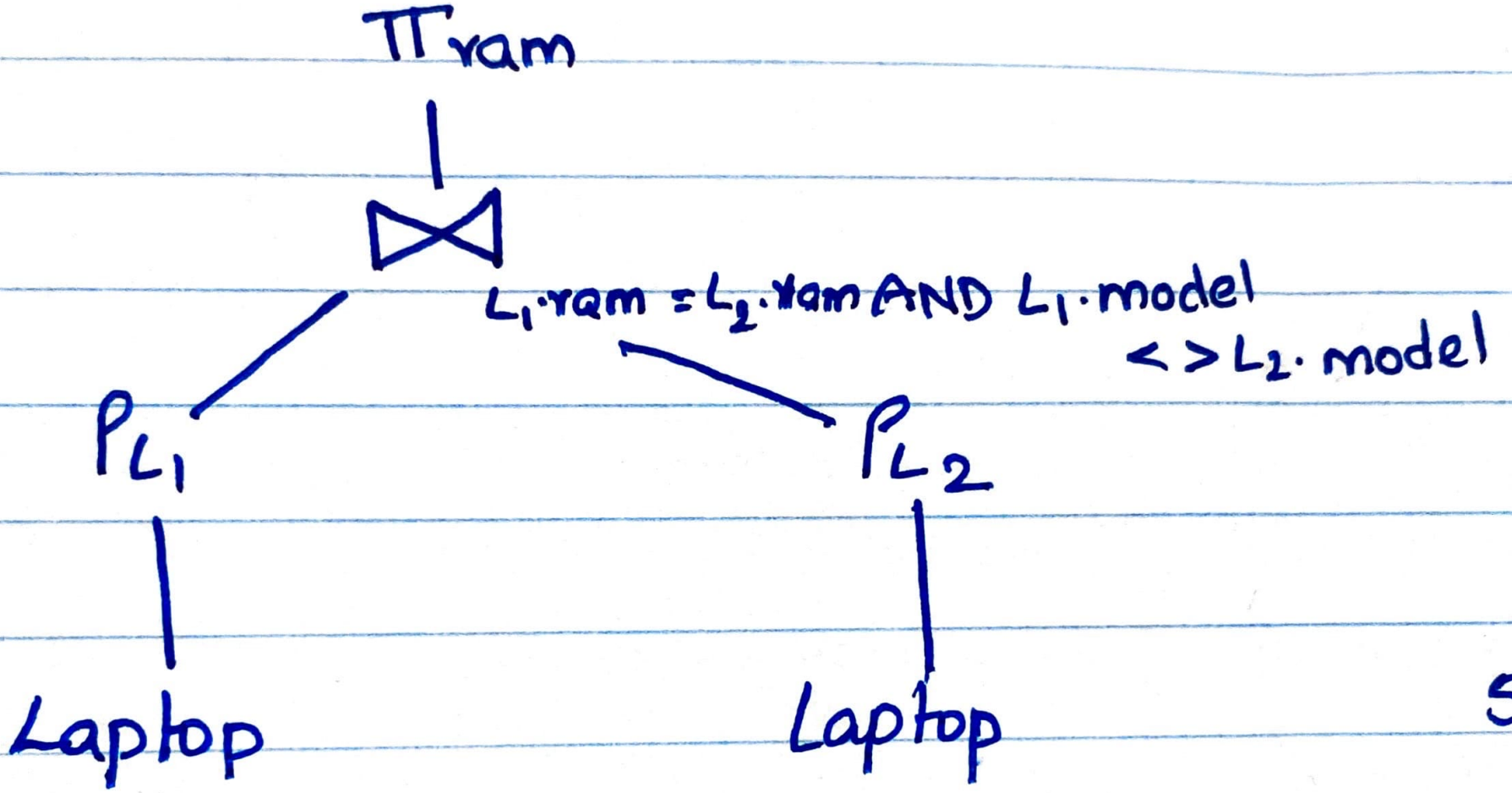
$$R_3 := R_1 \bowtie_{(L_1.\text{ram} = L_2.\text{ram} \text{ AND } L_1.\text{model} <> L_2.\text{model})} R_2$$

$$R_4 := \Pi_{\text{ram}}(R_3)$$

Result :



5(3):



5(4): Find the manufacturer of computer with highest available speed.

$R_1 := \Pi_{model, Speed}(PC)$

$R_2 := \Pi_{model, Speed}(Laptop)$

$R_3 := R_1 \cup R_2$

$R_4 := \rho_{R_4(model_2, Spd_2)}(R_3)$

$R_5 := R_3 \bowtie_{Speed < Spd_2} (R_4)$

$R_6 := \Pi_{model, Speed}(R_5)$

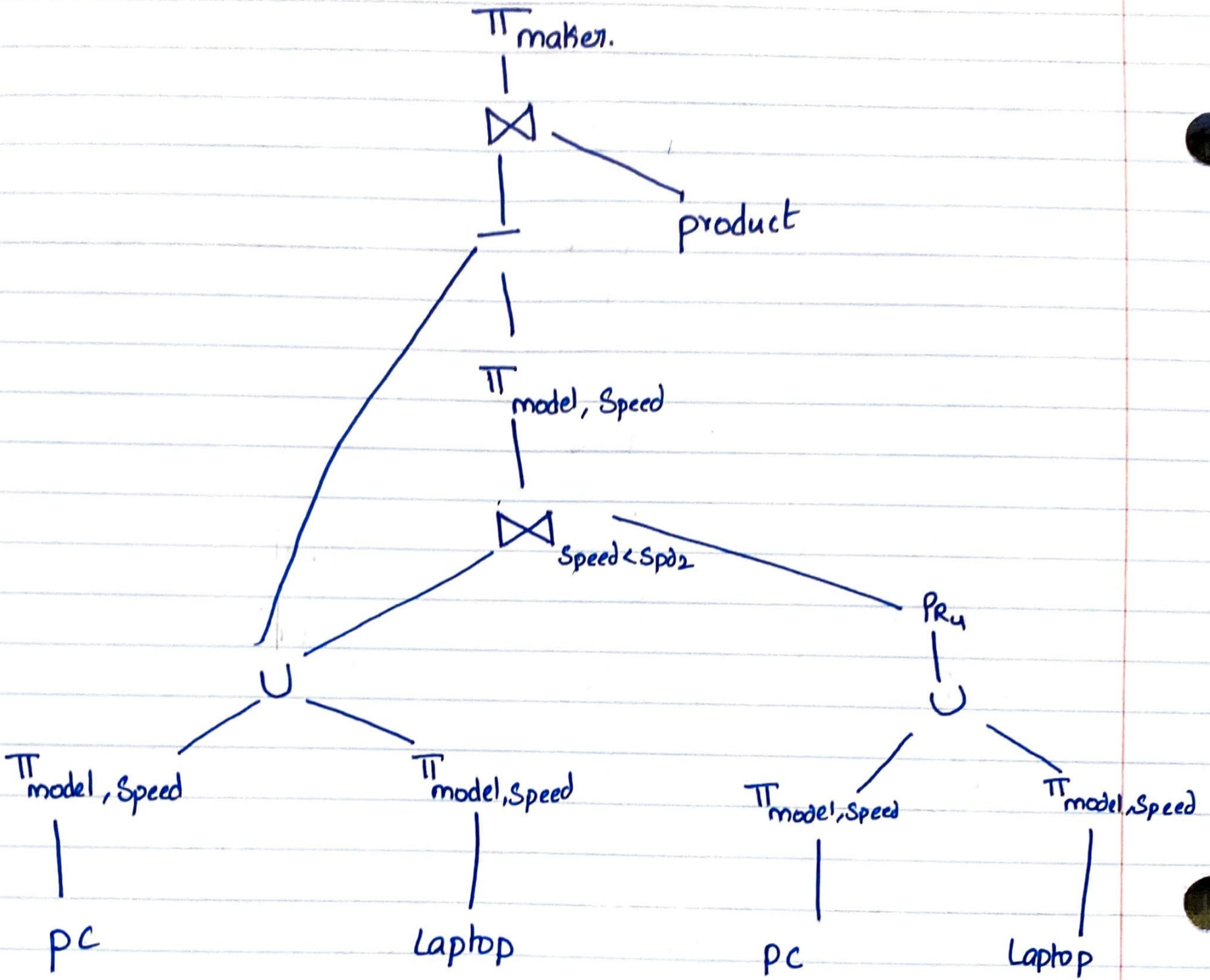
$R_7 := R_3 - R_6$

$R_8 := R_6 \bowtie product$

$R_9 := \Pi_{maker}(R_8)$

Result

maker
B


Π maker.
⋈
product
Π model, Speed
⋈
Speed < Spd2
PR4
∪
∪
Π model, Speed
Π model, Speed
Π model, Speed
Π model, Speed
pc
Laptop
pc
Laptop

(5) Find the manufacturer who sells atleast three different models of PC

$R_1 := \Pi_{maker,\ model}$ (Product ⋈ PC)

$R_2 := \rho_{R_2}(maker2,\ model2)(R_1)$

$R_3 := \rho_{R_3}(maker3,\ model3)(R_1)$

$R_4 := \rho_{R_4}(maker4,\ model4)(R_1)$

$R_5 := R_1 \bowtie (maker = maker2 \text{ AND } model <> model2) R_2$

$R_6 := R_3 \bowtie (maker3 = maker \text{ AND } model3 <> model) R_5$

$R_7 := R_4 \bowtie (maker4 = maker \text{ AND } (model4 = model \text{ OR } model4 = model2 \text{ OR } model4 = model3))$ (R_6)

$R_8 := \Pi_{maker}(R_7)$

maker
A
B
D
E

(6) Find those hd sizes that occur in two or more PC's

$R_1 := \rho_{P_1}(PC)$

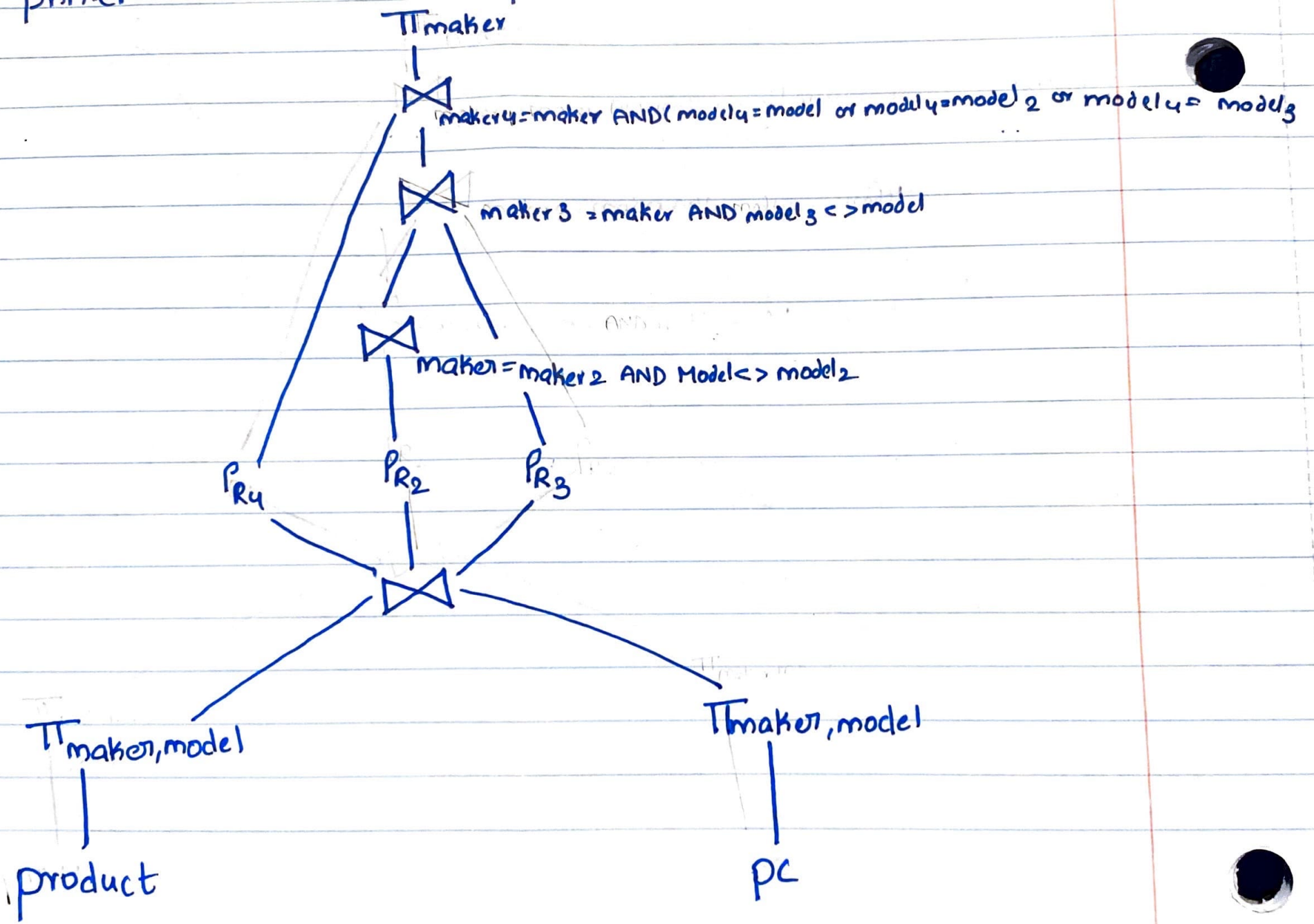
$R_2 := \rho_{P_2}(PC)$

$R_3 := R_1 \bowtie_{(P_1.hd = P_2.hd \text{ AND } P_1.model <> P_2.model)} R_2$

$R_4 := \Pi_{hd}(R_3)$

Result:

hd


Πmaker
maker4=maker AND(model4=model or model4=model2 or model4= model3
maker3 =maker AND model3 <>model
maker=maker2 AND Model<> model2
PR4
PR2
PR3
Πmaker,model
Πmaker,model
product
pc

(6) Find those hd sizes that occur in two (or) more PC's

$R_1 := \rho_{P_1}(PC)$

$R_2 := \rho_{R_2}(PC)$

$R_3 := R_1 \bowtie_{(P_1.hd = P_2.hd \text{ AND } P_1.model <> P_2.model)} R_2$

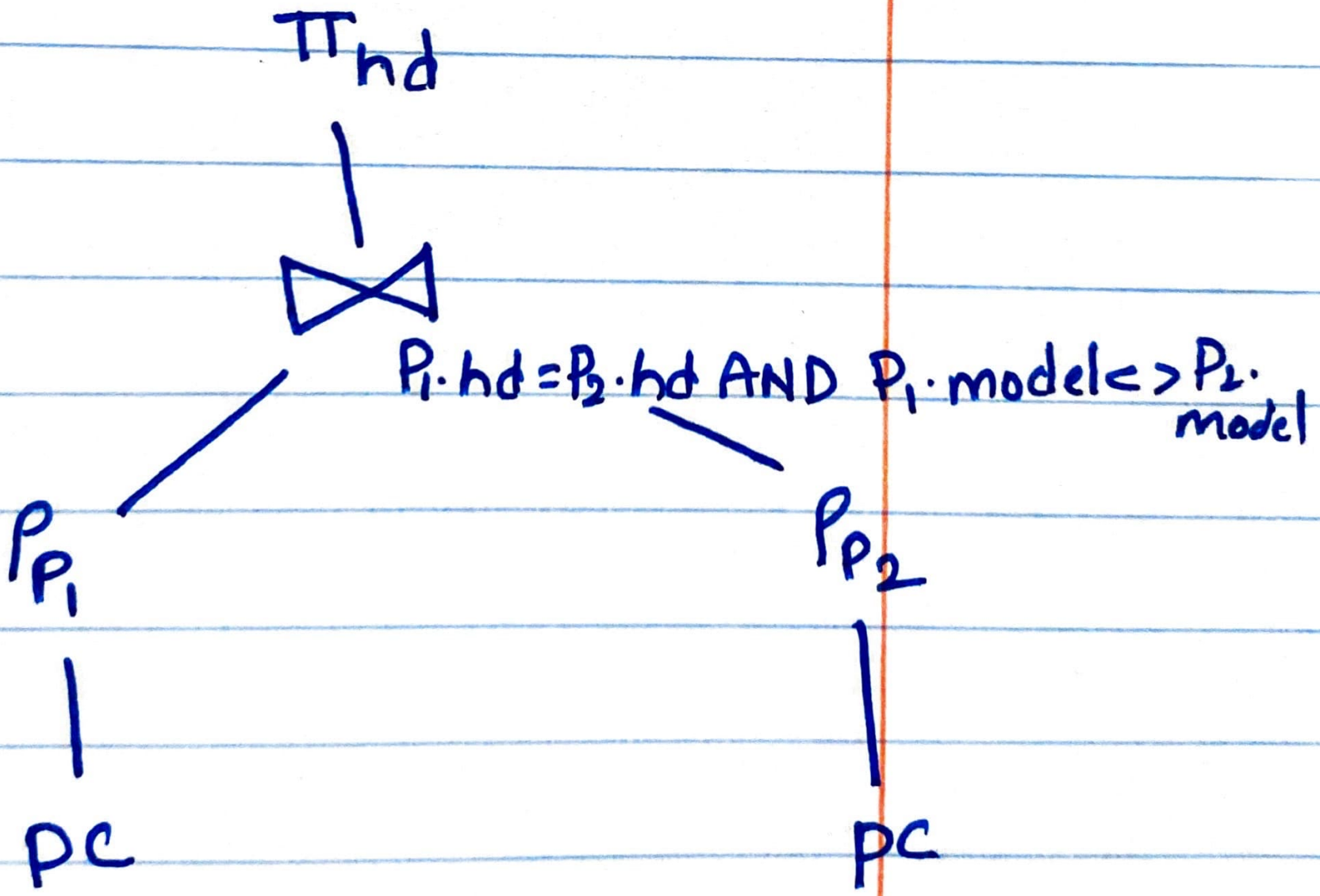
$R_4 := \Pi_{hd}(R_3)$

Result :

hd
250
80
160

$<> P_2.model$

5(6):



↦ Find those pairs of PC model that have both same speed and RAM. A pair should be listed only once

$R_1 := \rho_{PC_1}(PC)$

$R_2 := \rho_{PC_2}(PC)$

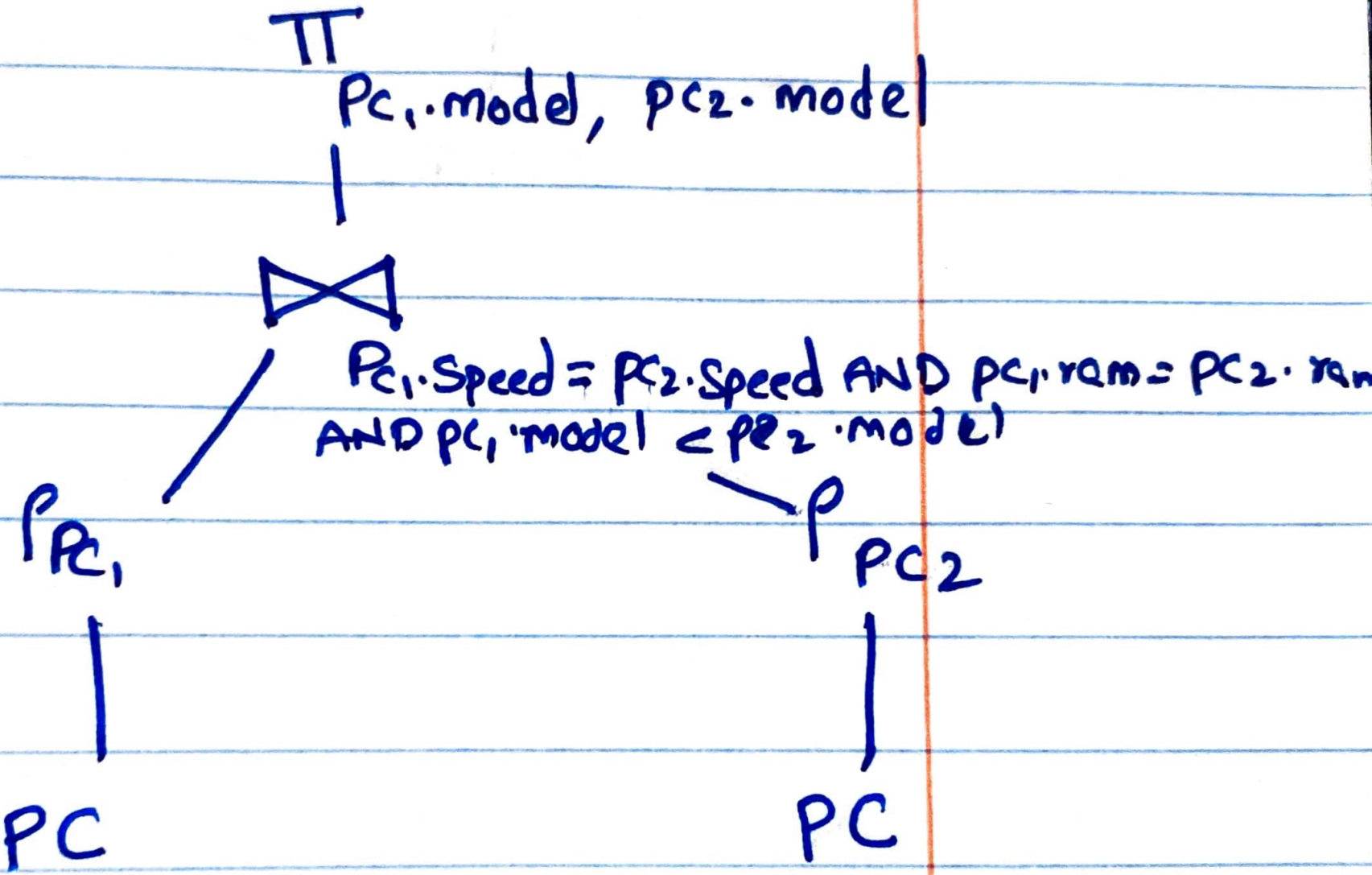
$R_3 := R_1 \bowtie_{(PC_1.speed = PC_2.Speed \text{ AND } PC_1.ram = PC_2.ram \text{ AND } PC_1.model < PC_2.model)} R_2$

$R_4 := \pi_{PC_1.model,\ PC_2.model}(R_3)$

Result:

PC_1.model	PC_2.model
1004	1012

5(7) :


π
pc1.model, pc2.model
pc1.speed = pc2.speed AND pc1.ram = pc2.ram
AND pc1.model < pc2.model
ρ
pc1
ρ
pc2
PC
PC

(8) find those manufacturers of atleast two different computer with speed of atleast 2.80

$R_1 := \pi_{model}(\sigma_{speed \geq 2.80(PC)}) \cup \pi_{model}(\sigma_{speed \geq 2.80(laptop)})$ $R_2 := \pi_{maker, model}(R_1 \text{ product})$

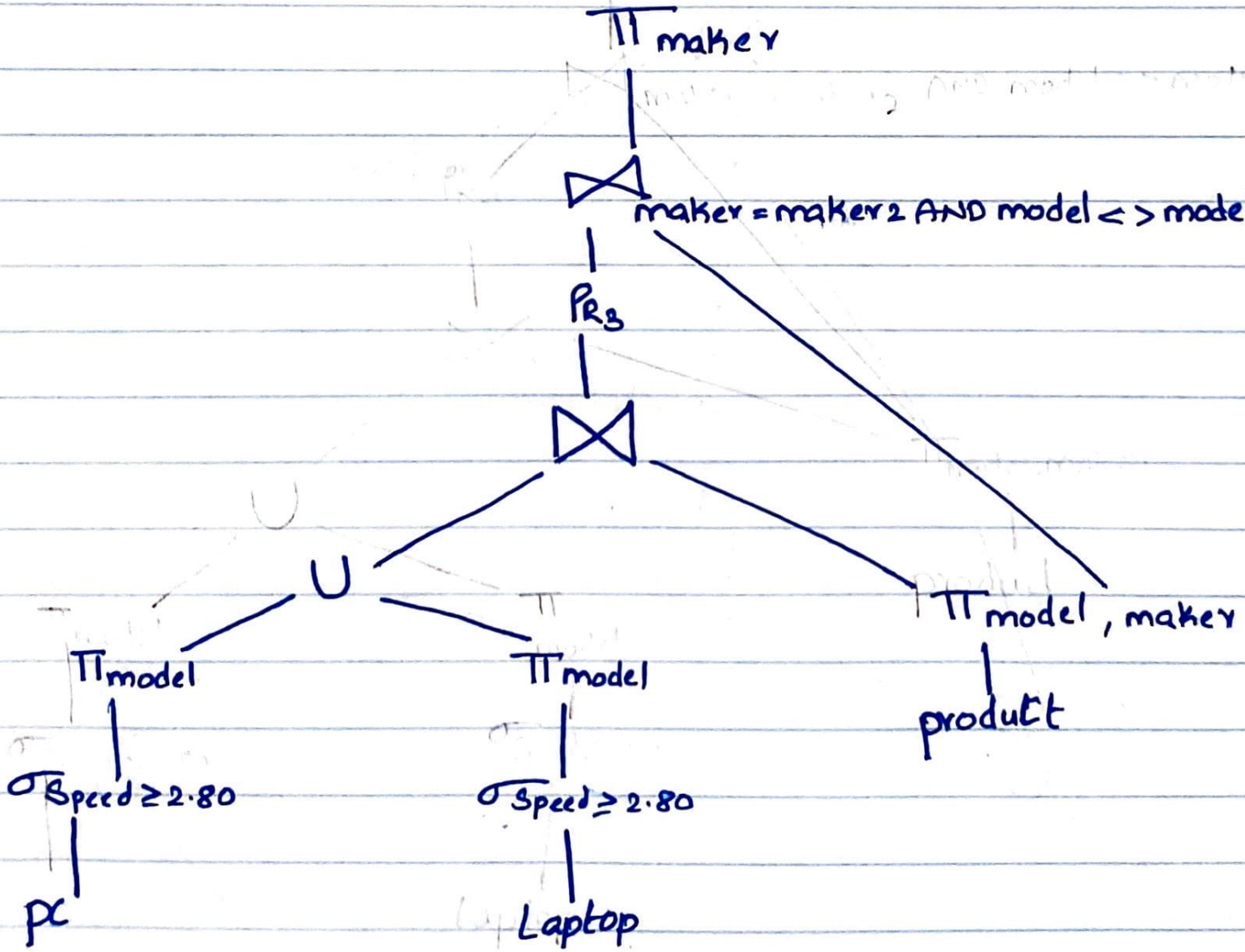
$R_3 := \rho_{R_3(maker_2, model_2)}(R_2)$

$R_4 := R_2 \bowtie_{(maker = maker_2 \text{ AND } model <> model_2)} R_3$

$R_5 := \pi_{maker}(R_4)$

maker
B
E

5(8)


Π maker
maker = maker2 AND model <> model2
PR3
U
Π model
Π model
Π model , maker
σ Speed ≥ 2.80
σ Speed ≥ 2.80
product
PC
Laptop

(9) find the manufacturer of cheapest selling colour printer

$R_1 := \Pi_{model, price}(Printer)$

$R_2 := \rho_{R_2(model_2, price_2)}(R_1)$

$R_3 := \Pi_{model, speed}(R_1 \bowtie_{price > price_2} R_2)$

$R_4 := R_1 - R_3$

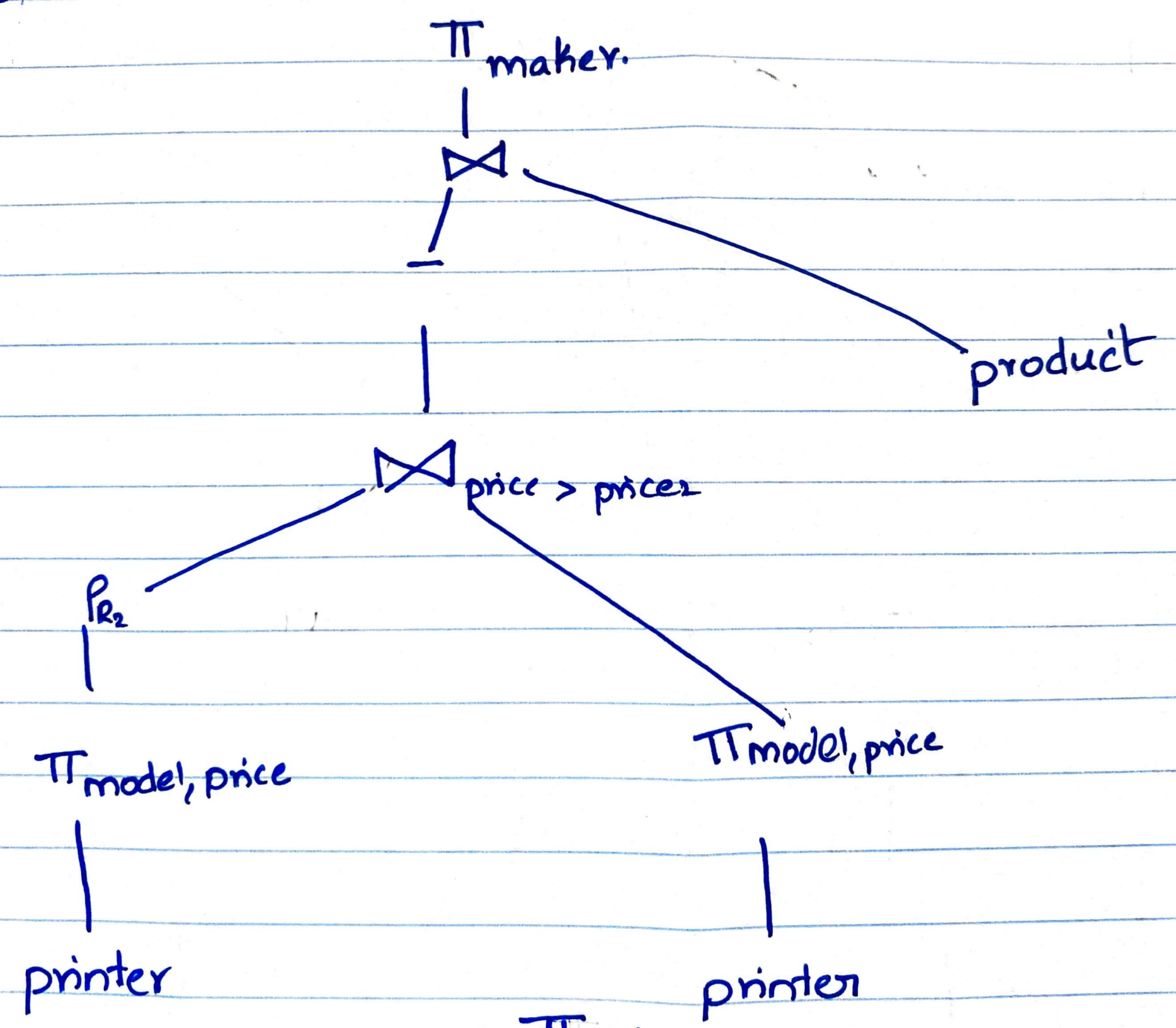
$R_5 := R_4 \bowtie product$

$R_6 := \Pi_{maker}(R_5)$

Result:

maker
E

Reference: Set operations on relations (2.4.4)
projection (2.4.5)
Selection (2.4.6)
Natural joins (2.4.8)
Combining operations to form (2.4.10)


Π maker.
product
price > price2
ρR2
Π model, price
Π model, price
printer
printer

6) what is the difference between the natural join $R \bowtie S$ and the theta join $R \bowtie_C S$ where the Condition C is that $R.A = S.A$ for each attribute A appearing in the Schemas of both R and S? you can describe your understanding with an Example?

$R \bowtie_C S$ theta join of relation R and S can also be represented as

$$R \bowtie_C S = \sigma_C (R \times S)$$

the natural join of R and S can be expressed by starting with the product $R \times S$. we then apply the selection operator with a Condition C of the form

$$R.A = S.A \text{ and } R.A_1 = S.A_1 \text{ and } R.A_2 = S.A_2 \cdots R.A_n = S.A_n$$

where $A_1, A_2, \ldots A_n$ are all the attributes appearing in the schemas of both R and S. finally, we project one of copy each of equated attributes. Let L be List of attributes in schema R followed by those attributes in Schema S of that are not also in schema R

$$R \bowtie S = \pi_L (\sigma_C (R \times S))$$

Reference : ULMAN Textbook chapter 2 SUBTOPIC 2.4.12 Relationship Among operations

7) A general form of relational-algebra query is

$$\pi_L (\sigma_C (R_1 \times R_2 \times \ldots \times R_n))$$

here L is an arbibary list of attributes and c is an arbitrary condition. the list of relations $R_1, R_2 \ldots R_n$ may include the same relation repeated several times in which case appropriate renaming may be assumed applied to the R_i's. Show how to express any query of this form in SQL

Relational algebra query is

$$\pi_L (\sigma_C (R_1 \times R_2 \times \ldots \times R_n))$$

the SQL query is

```
SELECT L
FROM R1, R2 ..... Rn
WHERE C is TRUE;
```

Reference : ULMAN Textbook - CHAPTER 2
SUBTOPIC 2.4.12 Relationships Among operations.