1. Consider the *following relational schema. An employee can work in more than one department; the pct_time field of the Works relation shows the percentage of time that a given employee works in a given department.*

> *Emp(eno, ename, age, salary)*
> *Works (eno, deptid, pct_time)*
> *Dept (deptid, deptname, budget, manageid)*

(1) Give a SQL statement to find the employees with salaries over 50,000, ages under 40 and working hours in software development departments over 50% ，list their NOs and *names*.
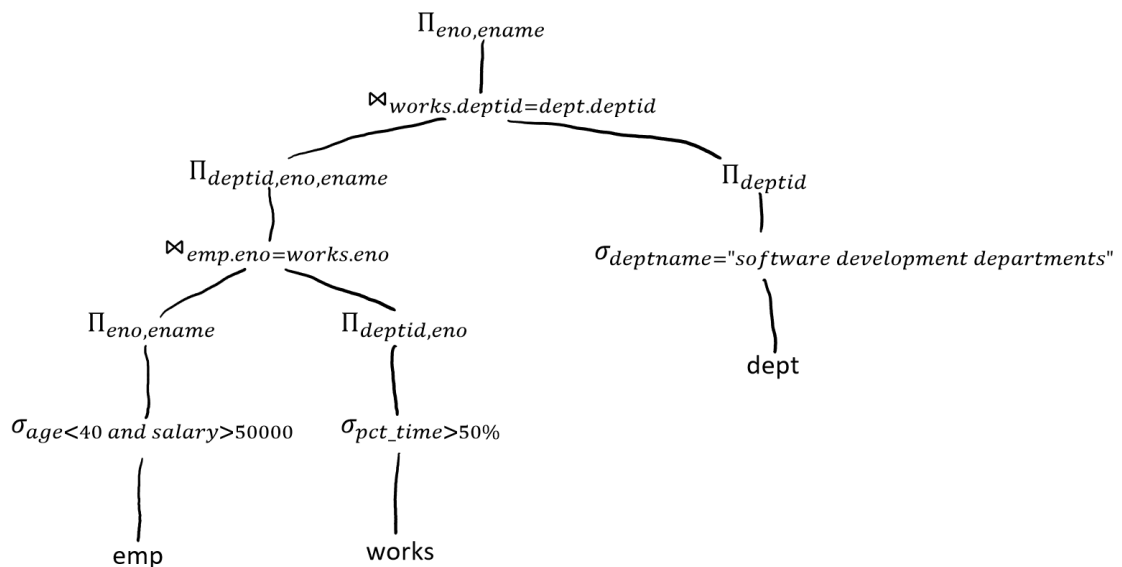
> Select Emp.eno, ename
>
> From emp, works, Dept
>
> Where emp.eno=works.eno and works.deptid=dept.deptid
>
> And salary>50000 and age<40 and pct_time>50 and deptname= "software development"

(2) For the SQL statement in (1), give an optimized query tree.

$$\Pi_{eno,ename}$$

$$\bowtie_{works.deptid=dept.deptid}$$

$$\Pi_{deptid,eno,ename} \qquad \Pi_{deptid}$$

$$\bowtie_{emp.eno=works.eno} \qquad \sigma_{deptname="software\ development\ departments"}$$

$$\Pi_{eno,ename} \qquad \Pi_{deptid,eno} \qquad \text{dept}$$

$$\sigma_{age<40\ and\ salary>50000} \qquad \sigma_{pct\_time>50\%}$$

$$\text{emp} \qquad \text{works}$$

- Consider the following schema, where the primary keys are underlined ,
  Suppliers (supplier-id, supplier-name, city, telephone, address)
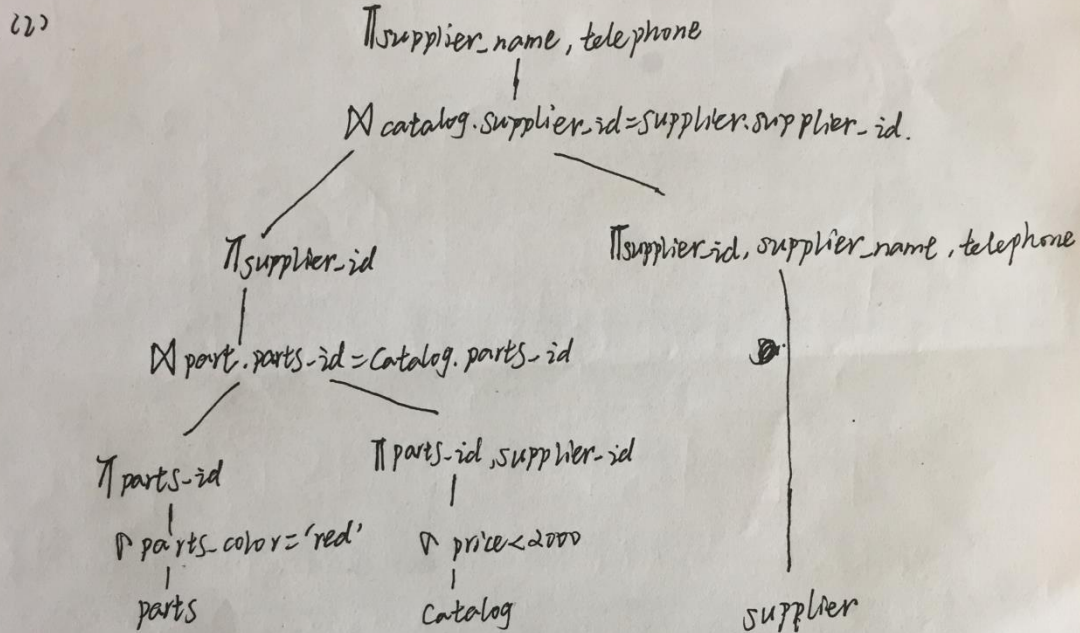 Parts (parts-id, parts-name, parts-color)
  Catalog (supplier-id, parts-id, price)
- (1) Give an SQL statement to find out the *name* and *telephone* of the suppliers who

supply a red *part* whose *price* is below $2000.
- (2) Translate this SQL statement into an initial query tree, and give an optimized query *tree* for it, by means of heuristic query optimization.

(1). Select supplier. supplier_name, supplier. telephone
from Supplier, Parts, Catalog.
where supplier. supplier_id = Catalog. supplier_id. and
Catalog. parts_id = Parts. parts_id. and
Parts. parts_color = 'red' and Catalog. price < 2000.

(2)
Π supplier_name, telephone
|
$\bowtie$ catalog. supplier_id = supplier. supplier_id.

Π supplier_id
|
$\bowtie$ part. parts_id = Catalog. parts_id

Π parts_id
|
σ parts_color = 'red'
|
parts

Π parts_id, supplier_id
|
σ price < 2000
|
Catalog

Π supplier_id, supplier_name, telephone
|
σ
|
supplier

2. (1) Select Student. name, Student. dept_name
from Student, department, takes
where Student. ID = takes. ID and Student. dept_name = department. dept
name
and Student. tot_cred > 40 and department. building = 'Building3'
and takes. course_id = '2016 CSDBS'.

2. Consider the database University given in the textbook,
   - (1) Give a SQL statement to find some students and list their names and departments that they belong to. It is required that their total credits (presented by tot_cred) are more than 40, their departments are located in Building 3, and they take the course identified by '2016CSDBS'.
   - (2) For the SQL statement in (1), give an optimized query tree.
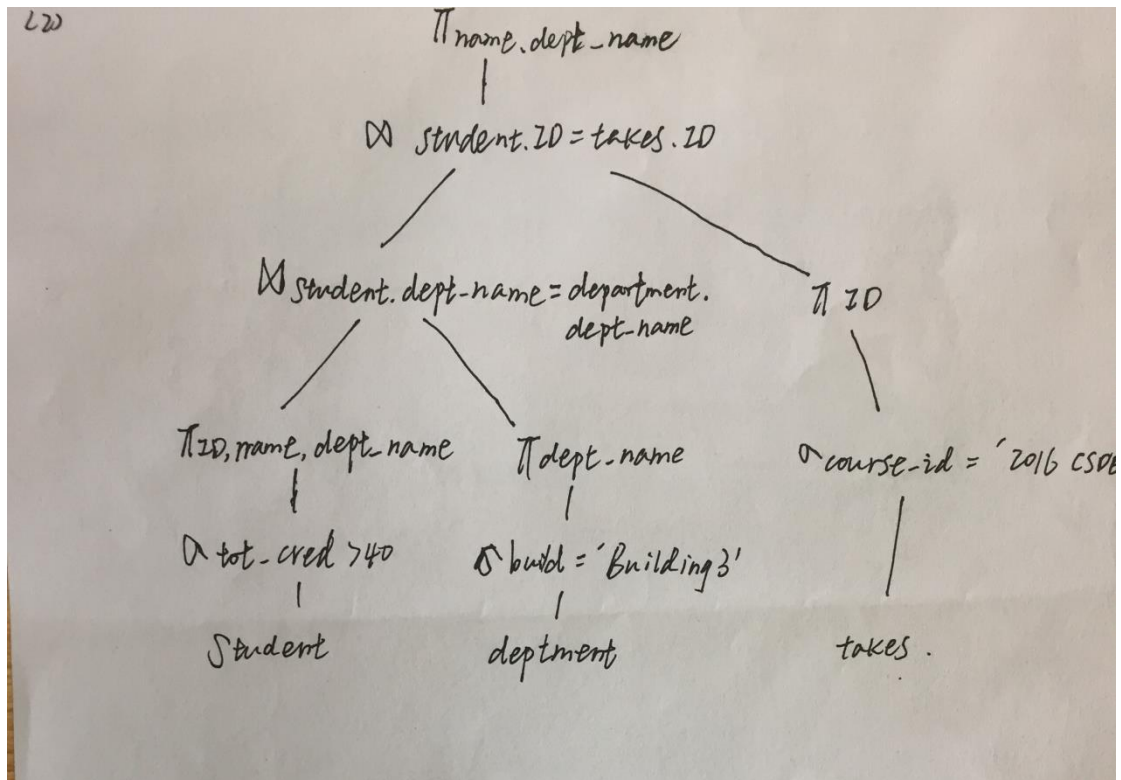
3. Consider the **Student-Project** management database given below.

     Student (Sno, Sname, class, major, dept_name)

     Project (Pno, Pname, requirement, start_time, end_time)

     Teacher (Tno, Tname, phone, email, dept_name)

     Department (dept_name, building, budget)

     Participation (Sno, Pno, Tno, grade)

(1) Give a SQL statement to find some students and list their *names* and *department*s that they belong to and list the *names* of the teachers who guide their projects. It is required that the students' major is '*Big data major*', they participate in the projects named as '*Big data analysis and mining*', their teachers who guide them are in *Computer Science Department*, and their *grades* of projects are more than 85.

    select   *Sname*, dept_name*, Tname*

    from   *Student*, *Participation, Project, Teacher*

    where *Student.Sno=Participation.Sno*

        and *Teacher.Tno=Participation.Tno*

        and *Project.Pno=Participation.Pno*

        and major=' *Big data major*'

        and *Project.Pname=' Big data analysis and mining*'

        and *Teacher. dept_name=' Computer Science Department*'

        and *grade>85*

或者：

    select   *Sname*, *major, Tname*

    from   *Student* **natural join** *Participation* **natural join** *Project* **natural join** *Teacher*

    where major=' *Big data major*' and *Project.Pname=' Big data analysis and mining*'

        and *Teacher. dept_name=' Computer Science Department*' and *grade>85*

(2) For the SQL statement in (1), give an optimized query tree.   (7 points)

Π Sname, dept_name, Tname

$\bowtie$ Participation.Tno=Teacher.Tno

Π Sname, dept_name, Tno

$\bowtie$ Participation.Pno=Project.Pno

Π Sname, dept_name, Pno, Tno

$\bowtie$ Student.Sno=Participation.Sno

Π Sno, Sname, dept_name

σ major=' Big data major'

Student
(Sno, Sname, class, major, dept_name)

Π Sno, Pno, Tno

σ grade>85

Participation
(Sno, Pno, Tno, grade)

Π Pno

σ *Pname=' Big data analysis and mining'*

Project
(Pno, Pname, requirement, start_time, end_time)

Π Tno, Tname

σ *dept_name=' Computer Science Department'*

Teacher
(Tno, Tname, phone, email, dept_name)