

Exercise 4

- Build the DB having the following tables

Employees

<u>Code</u>	Name	Age	Salary
101	Mario Rossi	34	4000
103	Mario Bianchi	23	3500
104	Luigi Neri	38	6100
105	Nico Bini	44	3800
210	Marco Celli	49	6000
231	Siro Bisi	50	6000
252	Nico Bini	44	7000
301	Sergio Rossi	34	7000
375	Mario Rossi	50	6500

Supervision

Head	<u>Employee</u>
210	101
210	103
210	104
231	105
301	210
301	231
375	252

Exercise 4

- Create a new database

```
CREATE DATABASE employeesSupervision;
```

- Create Tables

```
CREATE TABLE employees (name VARCHAR(20) NOT  
NULL, surname VARCHAR(20) NOT NULL, age  
smallint UNSIGNED, salary double(7,2), code  
smallint UNSIGNED NOT NULL AUTO_INCREMENT  
PRIMARY KEY );
```

```
CREATE TABLE supervision (employee smallint(5)  
unsigned primary key references  
employees(code), head smallint(5) unsigned  
references employees(code) );
```

Exercise 4

Populating tables (without auto_increment)

```
INSERT INTO employees VALUES ('mario', 'rossi', 34,  
    4000, 101), ('mario', 'bianchi', 23, 3500, 103),  
    ('luigi', 'neri', 38, 6100, 104), ('nico', 'bini',  
    44, 3800, 105), ('marco', 'celli', 49, 6000, 210),  
    ('Siro', 'Bisi', 50, 6000, 231), ('Nico', 'Bini',  
    44, 7000, 252), ('Sergio', 'Rossi', 34, 7000, 301),  
    ('Mario', 'Rossi', 50, 6500, 375);  
  
INSERT INTO supervision VALUES (101, 210), (103, 210),  
    (104, 210), (105, 231), (210, 301), (231, 301),  
    (252, 375);
```

Exercise 4: Queries

Q: *find code, name, surname, age and salary of the employees earning more than 4000 Euros*

```
select * from employees where salary > 4000;
```

Tuple Relational Calculus:

$$\{ e.* \mid e(\text{Employees}) \mid e.\text{salary} > 4000 \}$$

Exercise 4: Queries

Q: find code, name, surname and age of the employees earning more than 4000 Euros

```
select code, name, surname, age
from employees
where salary > 4000;
```

Tuple Relational Calculus:

$$\{ e.(code, name, surname, age) \mid e(employees) \mid e.salary > 4000 \}$$

Exercise 4: Queries

Q: find the codes of the heads of the employees which earn more than 4000 Euros

```
select distinct head
from supervision s join employees e on
    e.code=s.employee
where salary > 4000;
```

Tuple Relational Calculus:

$$\{ s.head \mid e(\text{Employees}), s(\text{Supervision}) \mid e.code = s.employee \wedge e.salary > 4000 \}$$

Exercise 4: Queries

Q: *find name, surname and salary of the heads of the employees which earn more than 4000 Euros*

```
select distinct e2.name, e2.surname,  
e2.salary  
from employees e1 join supervision s on  
e1.code=s.employee join employees e2 on  
s.head = e2.code  
where e1.salary > 4000;
```

Tuple Relational Calculus:

$$\{ \text{NameH}, \text{SalaryH} : e'.(\text{name}, \text{salary}) | \\ e'(\text{Employees}), e(\text{Employees}), s(\text{Supervision}) \mid e.\text{code} = \\ s.\text{employee} \wedge s.\text{head} = e'.\text{code} \wedge e.\text{salary} > 4000 \}$$

Exercise 4: Queries

Q: find the employees which earn more than their respective heads. Show: code, name, surname and salary of such employees and their heads

```
select distinct e1.code, e1.name, e1.surname,
  e1.salary, e2.code as headCode, e2.name as
  headName, e2.surname as headSurname, e2.salary
as headSalary
from employees e1 join supervision s on
  e1.code=s.employee join employees e2 on s.head
  = e2.code
where e1.salary > e2.salary;
```

Tuple Relational Calculus:

$$\{e.(name, code, salary), nameH, codeH, salarH : e'.(name, code, salary) | \\ e'(Employees), e(Employees), s(Supervision) | e.code = s.employee \wedge \\ s.head = e'.head \wedge e.Stipendio > e'.Stipendio \}$$

Exercise 4: Queries

Q: find code, name and surname of the heads whose employees all earn more than 4000 euros

```
select distinct e.code, e.name, e.surname
from employees e join supervision s on
    s.head=e.code
where e.code not in (select head
                    from employees e1 join
                    supervision s1 on e1.code =
                        s1.employee
                    where e1.salary <= 4000);
```

Tuple Relational Calculus:

$$\{e.(code, name) \mid e(Employees), s(Supervision) \mid e.code = s.head \wedge \neg(\exists e'(Employees) (\exists s'(Supervision) (s.head = s'.head \wedge s'.employee = e'.code \wedge e'.salary \leq 4000)))\}$$