10. Formal Query Languages

10.5.1 Selection and Projection Query

Consider the Relation Schema:

Employee(<u>Eid</u>, Ename, Profession, Rate)

and consider the query: Find the employee Eid and Ename for employees with Profession 'Carpenter', and whose Rate is less than 100 $/Hour.

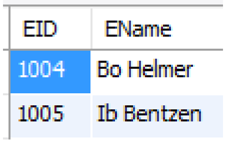
a) Make the query in SQL.
Create the relation, populate with some example data, and make the query.

```
CREATE DATABASE Formal_Query_Languages;
USE Formal_Query_Languages;

CREATE TABLE Employee(
Eid INT PRIMARY KEY,
Ename VARCHAR(30),
Profession VARCHAR(30),
Rate DECIMAL(8,2));


INSERT Employee VALUES
(1001, 'Johan Jensen', 'Painter',125.00),
(1002, 'Thomas Koberg', 'Painter', 80.00),
(1003, 'Jesper Hansen', 'Carpenter', 300.00),
(1004, 'Bo Helmer', 'Carpenter', 85.00),
(1005, 'Ib Bentzen', 'Carpenter', 95.00);

SELECT Eid, Ename FROM Employee
WHERE Profession = 'Carpenter'  AND Rate < 100;
```



EID	EName
1004	Bo Helmer
1005	Ib Bentzen

b) Make the query in Relational Algebra:

$\Pi_{\text{Eid, Ename}}(\sigma_{\text{Profession='Carpenter'} \wedge \text{Rate} < 100}(\text{Employee}))$

c) Make the query in Domain Calculus:

{<Eid, Ename> | ∃ Profession, Rate (<Eid, Ename, Profession, Rate>∈Employee ^ Profession='Carpenter' ^ Rate<100) }

10.5.2 Join Query

As continuation of 10.5.1, consider the Relation Schemas:

Employee(<u>Eid</u>, Ename, Profession, Rate)
Projects(<u>Pid</u>, Pname)
Staffing(<u>Pid</u>, <u>Eid</u>, Hours)

and consider the query: Find for each staffed project and each of the employees in its staff: the Pid, Pname, Eid, Ename, Hours and Rate.

a) Make the query in SQL.
Create the relations, populate with some example data, and make the query.

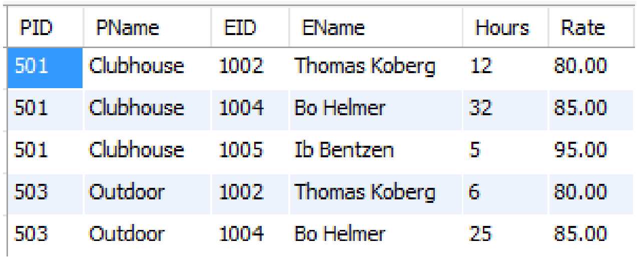
```
CREATE TABLE Projects (
Pid        INT PRIMARY KEY,
Pname VARCHAR(30));

CREATE TABLE Staffing (
Pid INT,
Eid INT,
Hours INT,
PRIMARY KEY (Pid, Eid),
FOREIGN KEY (Pid) REFERENCES Projects(Pid),
FOREIGN KEY (Eid) REFERENCES Employee( Eid));

INSERT Projects VALUES
(501, 'Clubhouse'), (502, 'Lightning'), (503, 'Outdoor'),
(504, 'Indoor');

INSERT Staffing VALUES
(501, 1002, 12), (501, 1004, 32), (501, 1005, 5),
(503, 1002, 6), (503, 1004, 25);

SELECT Pid, Pname, Eid, Ename, Hours, Rate FROM
Employee NATURAL JOIN Staffing NATURAL JOIN Projects;
```



PID	PName	EID	EName	Hours	Rate
501	Clubhouse	1002	Thomas Koberg	12	80.00
501	Clubhouse	1004	Bo Helmer	32	85.00
501	Clubhouse	1005	Ib Bentzen	5	95.00
503	Outdoor	1002	Thomas Koberg	6	80.00
503	Outdoor	1004	Bo Helmer	25	85.00

b) Make the query in Relational Algebra:

$\Pi_{\text{Pid, Pname, Eid, Ename, Hours, Rate}}$ (Employee ⋈ Staffing ⋈ Projects)

c) Make the query in Domain Calculus:

{<Pid, Pname, Eid, Ename, Hours, Rate>
|∃ Profession (<Eid, Ename, Profession, Rate>∈Employee
˄ <Pid, Eid, Hours>∈Staffing
˄ <Pid, Pname>∈Projects) }

10.5.3 Aggregation and Grouping

Consider the University database. Find for each course offered in autumn 2009 the number of students who have taken that course.

a) Make the query in Relational Algebra:

${}_{\text{CourseID}}G_{\text{COUNT(StudID)}}\ (\sigma_{\text{StudyYear =2009} \wedge \text{Semester = 'Fall'}}(\text{Takes}))$

b) Make the query in SQL.

```
SELECT CourseID,  COUNT(StudID)
FROM Takes
WHERE StudyYear =2009 AND Semester
= 'Fall'          GROUP BY CourseID;
```