Advanced Topics in Databases
Exercise 6

1. Name and describe the different stages of query processing.
   Why is an optimization step during the query processing needed?

2. Algebraic Optimization
   (a) Which algebraic rules exists? Give an example for the rules.
   (b) The following tables are given:
      
      PERSONS (PNo, Name, PC, Town, Street)
      TELEPHONE (PNo, TelephoneNo)
      EMPLOYEE (PNo, PersonalNo, Department, Salary, Room)
      STUDENT (PNo, MatrNo, CourseId)
      LECTURE (LID, LName, Credits, CourseId, Lecturer)
      EXAMINATION_REGISTRATION (LID, MatrNo, Date)
      
      Furthermore, the following views exist:
      
      PHONE_DIRECTORY: select * from Persons natural join TELEPHONE
      STUDENT_DATA: select PNo, Name, MatrNo from PERSONS natural join STUDENTS
      EXAMINATION_LIST: select * from EXAMINATION_REGISTRATION natural join STUDENT_DATA natural join PERSONS

      Transform the following queries into relational algebra and optimize the queries algebraically:
      
      i. select TelephoneNo
         from PERSONS natural join TELEPHONE natural join EMPLOYEE
         where Name= ‘Paul Dietrich ’or ROOM= ‘G59-311 ’
      
      ii. select TelephoneNo
          from STUDENT_DATA natural join PHONE_DIRECTORY
          where Name like ’%Meier’and CourseId=5
       
      iii. select Name
           from STUDENT_DATA natural join EXAMINATION_LIST natural join LECTURE natural join EMPLOYEE natural join PHONE_DIRECTORY
           where Name=’Gunter Saake ’and Lecturer=PersonalNo

   (c) Given is the following additional optimization rule for the algebraic optimization:
   \[ \pi_x (r_1 - r_2) \leftrightarrow \pi_x (r_1) - \pi_x (r_2) \]
Discuss the given rule: Is the given rule correct? Under which condition is the given rule usable?

3. Describe the cost-based optimization approach.
   What is the difference between cost-based and algebraic optimization?

4. Physical Optimization

From the lecture the following relations with corresponding constraints are known:

- **CUSTOMER**: (CNo, Cadr, Last_Name)
- **ORDER**: (CNo→Customer, Product, Amount)

- Table **CUSTOMER**: 1000 rows; one page: 5 rows
- Table **ORDER**: 100,000 rows; one page: 10 rows
- 500 orders include coffee
- 50 rows of (CNo, Last_Name) on one page
- 3 rows of CUSTOMER × ORDER on one page
- Puffer for every relation size: 1, no clamping sets

Query:

```sql
select c.CNo, Last_Name
from CUSTOMER c, ORDER o
where c.CNo = o.CNo and product = 'coffee'
```

What is the optimal plan under this requirements?
How does the number of required read and write access changes under the following assumptions:

(a) The relation **CUSTOMER** is a view. The root relation **PERSONS** has 10,000 entries with 2 tuples per page
   ```sql
   create view CUSTOMER as
   select CNo, Cadr, Last_Name
   from PERSONS
   where purchase > 0;
   ```

(b) The relation **CUSTOMER** is organized as B+-tree?
(c) A hash-index exists using the Product attribute of the relation **ORDER**
(d) The puffer has the size of 10 pages
(e) ‘purchase >0 ’ is replaced by ‘Product < >’coffee ”

Good Luck!