Implementation of Databases

Assignment 2

Participants:
(sorted in last name order)
Ulfet CETIN
Shreya KAR
Samuel ROY

Exercise 2.1 (Relational Calculus)

1. Find the names of employees who are certified to fly aircraft manufactured by Boeing.

   - TRC

     $$
     \begin{array}{l}
     \{\\
     \quad t \mid \exists e \in employee \; \exists c \in certified \; \exists a \in aircraft\\
     \quad (\\
     \qquad t.name = e.name \land a.aircraft\_id = c.aircraft\_id \land\\
     \qquad e.emp\_id = c.emp\_id \land a.manufacturer = ''Boeing''\\
     \quad)\\
     \}
     \end{array}
     $$

   - DRC

     $$
     \begin{array}{l}
     \{\\
     \quad < n > \; \mid \exists a, e, m(< e, n > \in employee \land < a, m > \in aircraft \land < a, e > \in certified \land m = ''Boeing''\,)\}
     \end{array}
     $$

2. Find the aircraft ids of all aircrafts that can be used on non-stop flights (i.e. where the aircraft.range > flights.distance) from Vancouver to Tokyo.

   - TRC

     $$
     \begin{array}{l}
     \{\\
     \quad t \mid \exists a \in aircraft \; \exists f \in flights\\
     \quad (\\
     \qquad t.aircraft\_id = a.aircraft\_id \land f.from =' Vancouver' \land\\
     \qquad f.to =' Tokyo' \land a.range \; > \; f.distance\\
     \quad)\\
     \}
     \end{array}
     $$

   - DRC

     $$
     \begin{array}{l}
     \{\\
     \quad < a > \; \mid \exists r, f, t, d(< f, t, d > \in flights \land < a, r > \in aircraft \land\\
     \quad f =' Vancouver' \land t =' Tokyo' \land r > dist\\
     \}
     \end{array}
     $$

3. Find the names of pilots who can operate planes with a range greater than 3000 miles but are not certified on any aircraft manufactured by Boeing.

   - TRC

     $$
     \begin{array}{l}
     \{\\
     \quad t \mid \exists e \in employee \; \exists c \in certified \; \exists a \in aircraft\\
     \quad (\\
     \qquad t.name = e.name \land a.aircraft\_id = c.aircraft\_id \land\\
     \qquad e.emp\_id = c.emp\_id \land a.range > 3000 \land\\
     \qquad \forall a1 \in aircraft\\
     \qquad\quad (\\
     \qquad\qquad a1.manufacturer = ''Boeing'' \land\\
     \qquad\qquad \exists c1 \in certified \; \exists e1 \in employee\\
     \qquad\qquad (\\
     \qquad\qquad\quad (a1.aircraft\_id = c1.aircraft\_id \land c1.emp\_id = e1.emp\_id) \Rightarrow\\
     \qquad\qquad\quad e1.emp\_id \neq e.emp\_id\\
     \qquad\qquad)\\
     \qquad\quad)\\
     \quad)\\
     \}
     \end{array}
     $$

- DRC

$$\begin{array}{l}
\{ \\
\quad NAME \mid \exists\ EID, NAME, SALARY, AID, MANUFACTURER, MODEL, RANGE \\
\quad (\\
\quad\quad \langle EID, NAME, SALARY\rangle \in employee\ \wedge \\
\quad\quad \langle EID, AID\rangle \in certified\ \wedge \\
\quad\quad \langle AID, MANUFACTURER, MODEL, RANGE\rangle \in aircraft\ \wedge \\
\quad\quad RANGE > 30000 \wedge \\
\quad\quad \forall AID_x, MODEL_x, RANGE_x \\
\quad\quad (\\
\quad\quad\quad \langle AID_x, ''BOEING'', MODEL_x, RANGE_x\rangle \in aircraft \wedge \\
\quad\quad\quad \exists EID_x, NAME_x, SALARY_x \\
\quad\quad\quad (\\
\quad\quad\quad\quad (\\
\quad\quad\quad\quad\quad \langle EID_x, AID_x\rangle \in certified\ \wedge \\
\quad\quad\quad\quad\quad \langle EID_x, NAME_x, SALARY_x, \rangle \in employee \\
\quad\quad\quad\quad) \Rightarrow EID_x \neq EID \\
\quad\quad\quad) \\
\quad\quad) \\
\quad) \\
\}
\end{array}$$

4. Find the employee id's of the employees who make the highest salary.

- TRC

$$\begin{array}{l}
\{ \\
\quad t \mid \exists e1 \in employee \\
\quad\quad (\\
\quad\quad\quad t.emp\_id = e1.emp\_id \wedge \\
\quad\quad\quad \neg(\exists e2 \in employee(e2.salary > e1.salary \wedge e1.emp\_id \neq e2.emp\_id)) \\
\quad\quad) \\
\}
\end{array}$$

- DRC

$$\begin{array}{l}
\{ \\
\quad < e1 > \ \mid \exists n1, s1 \\
\quad (\\
\quad\quad < e1, n1, s1 > \in employees \wedge \\
\quad\quad \neg(\exists e2, n2, s2 \\
\quad\quad\quad\quad (\\
\quad\quad\quad\quad\quad < e2, n2, s2 > \in employees \wedge e2 \neq e1 \wedge s2 > s1 \\
\quad\quad\quad\quad) \\
\quad\quad\quad) \\
\quad) \\
\}
\end{array}$$

Exercise 2.2 (Sorting)

Suppose you have a file of 20,000 pages and five buffer pages and you are sorting it using general (external) merge-sort.

1. How many runs will you produce? Remark: When a file is sorted, in intermediate steps subfiles are created. Each sorted subfile is called a run.

Question does not specify which #runs is required to mention for which step (zeroth pass, first pass, ...).

We assume what is asked is the total number of runs produced during the whole sorting procedure.

In the following table, you can find how many runs are produced in the individual steps, and the total sum.

Pass No	How We Calculate It	Number of Runs
0	$\lceil 20000/5 \rceil = \lceil 4000 \rceil$	4000
1	$\lceil 4000/4 \rceil = \lceil 1000 \rceil$	1000
2	$\lceil 1000/4 \rceil = \lceil 250 \rceil$	250
3	$\lceil 250/4 \rceil = \lceil 62.5 \rceil$	63
4	$\lceil 63/4 \rceil = \lceil 15.75 \rceil$	16
5	$\lceil 16/4 \rceil = \lceil 4 \rceil$	4
6	$\lceil 4/4 \rceil = \lceil 1 \rceil$	1
Total		5334

2. How many passes will it take to sort the file completely?

$$\begin{aligned} \text{Number of Passes} &= 1 + \log_{B-1}\lceil N/B \rceil \\ &= 1 + \log_4\lceil 20000/5 \rceil \\ &= 1 + \lceil 5.9828921423310435 \rceil \\ &= 7 \end{aligned}$$

3. How many buffer pages do you need at least to sort the file in two passes?

   Let B denote the number of buffer pages available in the system.

If it is assumed that the zeroth pass IS NOT counted as one of the two passes,

- Number of Passes $= 1 + \log_{B-1}\lceil 20000/B \rceil = 3$
- $\log_{B-1}\lceil 20000/B \rceil = 2$
- if B is chosen as 28:

  $\log_{27}\lceil 20000/28 \rceil \overset{?}{=} 2$

  $\lceil 1.9938131981415528 \rceil \overset{?}{=} 2$
- The answer is 28.

If it is assumed that the zeroth pass IS counted as one of the two passes,

- Number of Passes $= 1 + \log_{B-1}\lceil 20000/B \rceil = 2$
- $\log_{B-1}\lceil 20000/B \rceil = 1$
- if B is chosen as 142:

  $\log_{141}\lceil 20000/142 \rceil \overset{?}{=} 1$

  $\lceil 0.9997778442543881 \rceil \overset{?}{=} 1$
- The answer is 142.

Exercise 2.3 (Indexing)

On the relation Cities(Name, Province, Population, KilometresFromAachen) we generate the following 2 queries:

- Q1:

```
SELECT Name , Province
FROM Cities
WHERE Population > 100000
```

- Q2:

```
SELECT Province , count ( city ) , sum ( Population )
FROM Cities
group by Province
```

1. Briefly explain how a B+-tree index on Population could be used during processing of Q1.

   Solution: B+ Trees are used to organize the data in a database, such that data can be retrieved in an effective way. In the query example given above, we can build a b+ tree index on the attribute 'population'. The advantage of doing this will be that we only need to traverse the right subtree of the node with population 10,000 to fetch data records with a population greater than 10,000. The This will reduce the costs and access time in comparison to a sequential selection algorithm.

2. Briefly explain why using a clustered index on Province would be more efficient than either a hash-table or B+-tree index during processing of Q2.

   Solution:Indexing on a group by clause can be done by two methods-

   (a) sorting (clustered index)

   (b) hashing

   Implementation of clustered index on the group by attribute i.e. 'province' sort records in an alphabetical order by province name leading to faster execution time, as the data records in a clustered index are located close to each other. If we use a hash table to index province name, it will demand more query processing, since the records are not actually close(worst case will be each being in the different page in the harddisk). Hence clustered index is a more effective way of indexing.

Exercise 2.4 (Short Questions)

1. Give two examples of SQL constructs/semantics not expressible in relational algebra (RA).
   SQL is more expressive than relational algebra. These operations cannot be expressed in Relational Algebra:
   (a) Order by
   (b) Group by

2. Explain the difference between DRC and TRC.

TRC:

- variables are tuple variables
- format: $\{S \mid p(S)\}$
- one can NOT directly see how many fields are there in S without looking into p(S), as assignments have to be done in p(S) explicitly (unless S is an element of one of relational schemas)

DRC:

- variables are domain variables
- format: $\{(x_1, x_2, ..., x_n) \mid p((x_1, x_2, ..., x_n))\}$
- one can look at $\{(x_1, x_2, ..., x_n)$ part and see how many variables are there, no need to explicitly assign values to x_i variables

3. What does "relational completeness" mean (in your own words, please)? Show that SQL is relationally complete by enumerating SQL constructs corresponding to selection, projection, cartesian product, union, and difference.
   Solution: A query language is said to be relationally complete if it can be expressed with relational algebra. Since SQL is a superset of relational algebra i.e. a more powerful language than relational algebra it is also called a relationally complete query language. SQL constructs that show its relational completeness are as follows:
   (a) Selection :
   Relational Algebra: $\sigma EmployeeId = 6500(Employee)$
   Query: select * from Employee where EmployeId=6500;
   (b) Projection :
   Relational Algebra: $\pi Name, Age(Employee)$
   Query: select Name,Age from Employee;
   (c) Cartesian Product
   Relational Algebra: $EmployeeXManager$
   Query : select * from Employee,Manager;
   (d) Union:
   Relational Algebra: $\pi Name(Employee)U\pi Name(Manager)$
   Query: select * from Employee or select * from Manager;
   (e) Difference:
   Relational Algebra: Employee-Manager
   Query: select * from Employee where EmployeeId not in (select EmployeeId from Manager);

4. What is an unsafe query? Considering the schema given in exercise 1.1, give an example and explain why it is important to disallow such queries.
   Solution : An unsafe query is a query in relational calculus which returns an infinite. number of results. It is important to disallow such queries because a database should return results for a query in a finite amount of time. Based on the schema in exercise 1.1, an example of an unsafe query on relation Employee is:
   $\{e|\neg(e \in employees)\}$
   This query returns all objects which are not employees. The results of this query is infinite and thus the query is unsafe.