

## Exercise 2

1. Discuss which query optimisation approach(es) (enumerating all plans, heuristic based, adaptive plans) can be suitable for the following scenarios:

- **Scenario A:** Given a table with 1 million tuples, run the following query:

```
SELECT customer
FROM Table
WHERE spend BETWEEN 100 AND 200
AND birth_year > 2000;
```

- **Scenario B:** Given 5 tables with 1 million tuples in each table, run a query:

```
SELECT T1.name, T2.salary, T3.qualification, T4.phone, T5.leader
FROM Table1 T1
    INNER JOIN Table2 T2 ON T2.id = T1.id
    INNER JOIN Table3 T3 ON T3.id = T1.id
    INNER JOIN Table4 T4 ON T4.id = T1.id
    INNER JOIN Table5 T5 ON T5.department = T1.department
WHERE T1.age > 50;
```

2. A particular query on a table A used to run quite efficiently in a DBMS. After inserting many records and deleting many other records from table A, that same query is now taking more time to run, even when the total number of records has not changed. What can be the reason for that? What can you do as the user/database administrator of that DBMS to improve the performance of this query?
3. When a database needs to join two tables using a page-oriented nested loop join algorithm, both tables are found to be in the main memory. Will it change how the cost of a cost-based optimiser calculates the cost of the query plan?
4. What are the possible approaches that you can take to improve query performance in practice?
5. In the worst case, if there is enough memory only to hold one page/block of each table, what are the estimated cost of simple nested loop join and block nested loop join? What are the number of seeks for simple nested loop join and block nested loop join?