PROBLEM 1
Borrowed from https://www.w3resource.com/sql-exercises/subqueries/index.php

DATABASE INSTANCE:

Emp_details

EMP_IDNO	EMP_FNAME	EMP_LNAME	EMP_DEPT
127323	Michale	Robbin	57
526689	Carlos	Snares	63
843795	Enric	Dosio	57
328717	Jhon	Snares	63
444527	Joseph	Dosni	47
659831	Zanifer	Emily	47
847674	Kuleswar	Sitaraman	57
748681	Henrey	Gabriel	47
555935	Alex	Manuel	57
539569	George	Mardy	27
733843	Mario	Saule	63
631548	Alan	Snappy	27
839139	Maria	Foster	57

Emp_department

DPT_CODE	DPT_NAME	DPT_ALLOTMENT
57	IT	65000
63	Finance	15000
47	HR	240000
27	RD	55000
89	QC	75000

QUESTIONS:

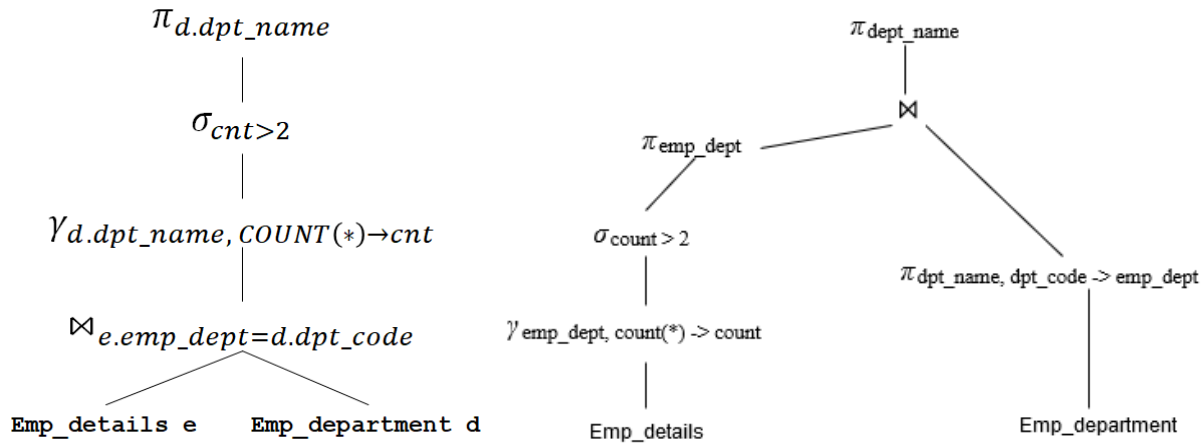
1. Write a query in SQL to find the *names of departments* where *more than two employees* are working.
2. Draw a Relational Algebra plan for your query.
3. Given the following statistics, estimate the cardinality of the output of each operator in your relational algebra plan.
    a. T(Emp_details) = 13, V(Emp_details, emp_dept) = 4
    b. T(Emp_department) = 5, V(Emp_department, dpt_allotment) = 5

PROBLEM 1 SOLUTION
Also see https://www.w3resource.com/sql-exercises/subqueries/sql-subqueries-inventory-exercise-38.php

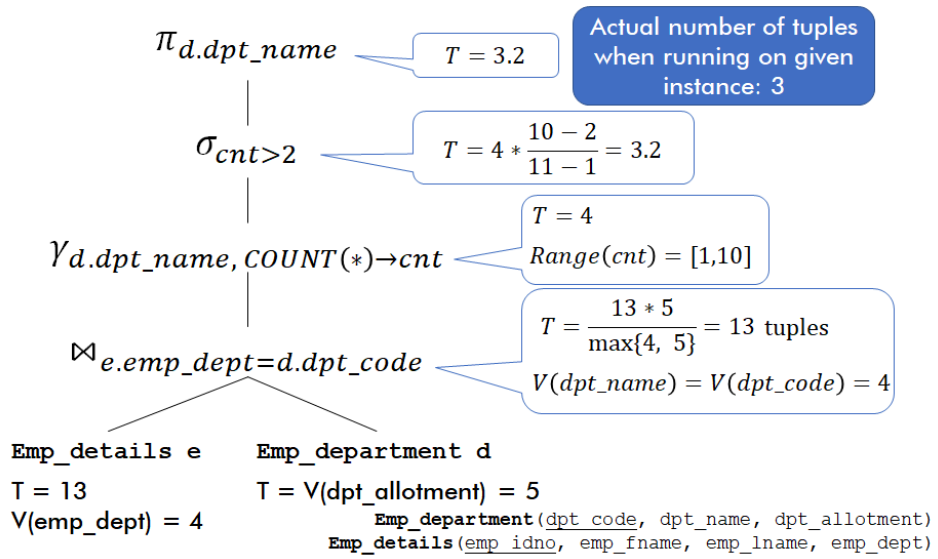
Here is the output when running the query on the given instance:

```
dpt_name
--------
IT
HR
Finance
```

Here are two possible RA plans. There are many other correct answers.
Challenge: can you prove that these two RA plans are semantically equivalent?
(use the RA Equivalences we covered in lecture)



Here is an annotated version with cardinality estimation:



Explanation: For the join we assume that e.emp_dept is contained in d.dpt_code. The cardinality estimate is given by formula. We also estimate the number of distinct values of d.dpt_name. To do this we assume that no two departments have the same name, V(dpt_name) = V(dpt_code). By the join condition this is also equal to V(emp_dept). The estimate number of unique values is 4, because that is the minimum between V(Emp_details, emp_dept) and V(Emp_department, dpt_code).

For the aggregate, the cardinality estimate is equal to V(dpt_name) of the input, which is 4. We also estimate the range of the COUNT(*) aggregate. There are 4 groups in the aggregate. Each group has a non-zero count (or else that department would not appear in the input), so the minimum count is 1. The maximum count is 10, which would be achieved if 3 groups had a count of 1 and the last group has a count of 10, since the total number of tuples in the input is 13.

For the selection, we use the range selection formula. In the total range, there are 10 numbers between and including 1 and 10.

Projection does not affect cardinality.

Here are a couple possible SQL queries.

(Shana's soln)

```
SELECT d.dpt_name
  FROM Emp_details e, Emp_department d
 WHERE e.emp_dept = d.dpt_code
 GROUP BY d.dpt_name
HAVING COUNT(*) > 2;
```

(Soln from w3resource.com)

```
SELECT dpt_name FROM emp_department
  WHERE dpt_code IN
  (
    SELECT emp_dept
      FROM emp_details
      GROUP BY emp_dept
      HAVING COUNT(*) >2
  );
```

PROBLEM 2
Borrowed from https://www.w3resource.com/sql-exercises/subqueries/index.php

DATABASE INSTANCE:

Emp_details

EMP_IDNO	EMP_FNAME	EMP_LNAME	EMP_DEPT
127323	Michale	Robbin	57
526689	Carlos	Snares	63
843795	Enric	Dosio	57
328717	Jhon	Snares	63
444527	Joseph	Dosni	47
659831	Zanifer	Emily	47
847674	Kuleswar	Sitaraman	57
748681	Henrey	Gabriel	47
555935	Alex	Manuel	57
539569	George	Mardy	27
733843	Mario	Saule	63
631548	Alan	Snappy	27
839139	Maria	Foster	57

Emp_department

DPT_CODE	DPT_NAME	DPT_ALLOTMENT
57	IT	65000
63	Finance	15000
47	HR	240000
27	RD	55000
89	QC	75000

QUESTIONS:

1. Write a query in SQL to find the *first name* and *last name* of employees working for departments whose allotment is *second lowest*.
2. Draw a Relational Algebra plan for your query.
3. Given the following statistics, estimate the cardinality of the output of each operator in your relational algebra plan.
    a. T(Emp_details) = 13, V(Emp_details, emp_dept) = 4
    b. T(Emp_department) = 5, V(Emp_department, dpt_allotment) = 5

HINT: First write the query / RA to find the *lowest* department allotment.
Next write the query / RA for the department with *second lowest* allotment.
Next write the full query / RA.

PROBLEM 2 SOLUTION
Also see https://www.w3resource.com/sql-exercises/subqueries/sql-subqueries-inventory-exercise-39.php

Here is the output when running the query on the given instance:

```
emp_fname  emp_lname
---------  ---------
Alan       Snappy
George     Mardy
```

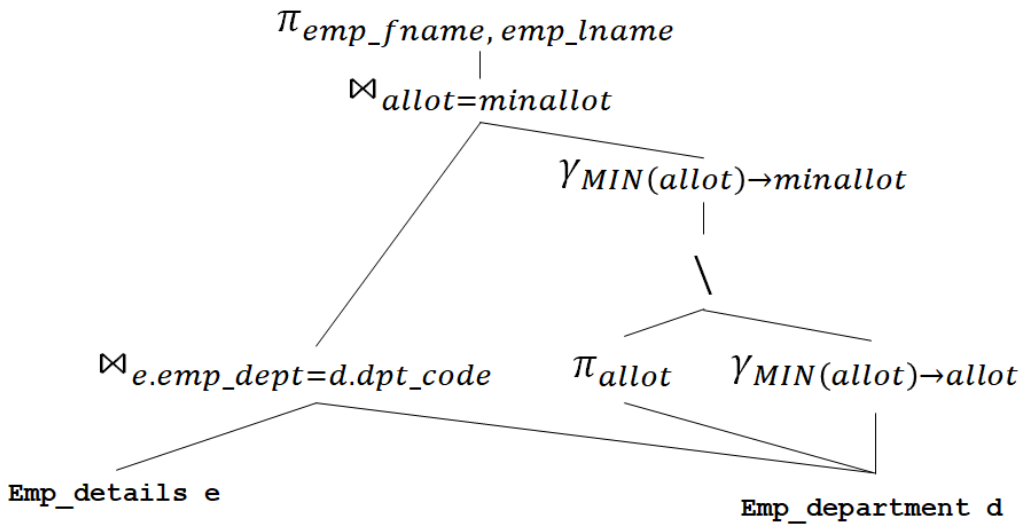
Here is some SQL. This initial solution heavily uses WHERE subqueries to filter out the allocations that are not the second lowest. Pro: may be easier to write. Con: cannot draw RA directly from this solution.

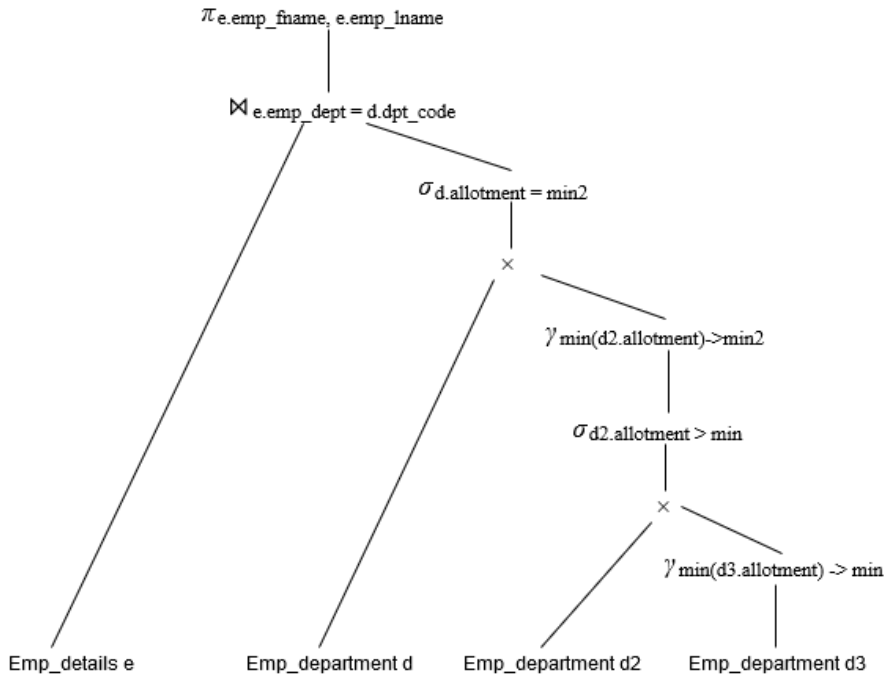
```
SELECT e.emp_fname, e.emp_lname
  FROM emp_details e, emp_department d
 WHERE e.emp_dept = d.dpt_code
   AND d.dpt_allotment = (
       SELECT MIN(d2.dpt_allotment)
         FROM emp_department d2
        WHERE d2.dpt_allotment > (
              SELECT MIN(d3.dpt_allotment)
              FROM emp_department d3));
```

This SQL came from the RA below.

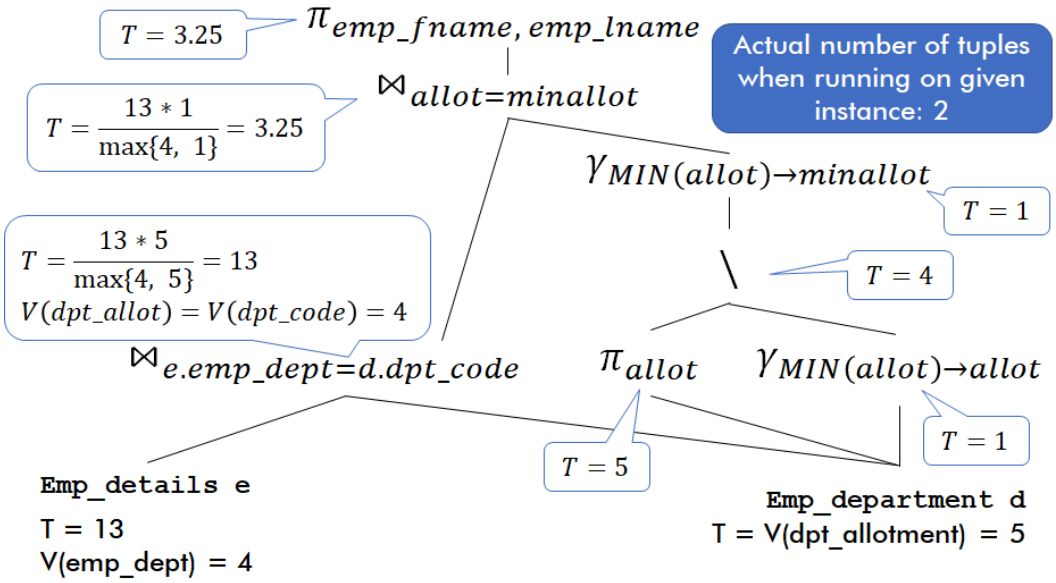
```
SELECT e.emp_fname, e.emp_lname
FROM emp_details e, emp_department d,
   (SELECT MIN(x.dpt_allotment) AS minallot2
      FROM (SELECT dpt_allotment
              FROM emp_department
            EXCEPT
            SELECT MIN(dpt_allotment) AS dpt_allotment
              FROM emp_department) x
   ) m
WHERE e.emp_dept = d.dpt_code
  AND d.dpt_allotment = m.minallot2
```

Here's a couple RA solutions.





Here is an annotated version with cardinality estimation:



Explanation: aggregation without groups results in a single tuple. Projection does not change cardinality.

The EXCEPT operator removes one tuple, which is guaranteed to be present in the input on the left. We know that it removes a single tuple because every department has a unique allocation (due to T = V(dpt_allotment) in Emp_department).

The first join is the same as in problem 1.

The second join joins 13 estimated tuples on the left with 1 tuple on the right. This is like a selection statement, where we are selecting those tuples from the left whose dpt_allotment is equal to the minallot from the right. The calculation works the same when we apply the join cardinality estimation formula.