

The University of New South Wales

COMP9315 DBMS Implementation

21T1 Final Exam

[\[Instructions\]](#) [\[PostgreSQL\]](#) [\[C\]](#)
[\[Q1\]](#) [\[Q2\]](#) [\[Q3\]](#) [\[Q4\]](#) [\[Q5\]](#) [\[Q6\]](#) **[\[Q7\]](#)** [\[Q8\]](#)

Question 7 (8 marks)

Consider the following database schema:

```

create table People (
    id          integer primary key,
    name        text,
    worksin     integer references Departments(id)
);
create table Departments (
    id          integer primary key,
    name        text,
    manager     integer references People(id)
);
create table Projects (
    id          integer primary key,
    name        text,
    fundedby    integer references Departments(id)
);
create table WorksOn (
    person      integer references People(id),
    project     integer references Projects(id),
    percent     integer
);

```

(Yes, it has a forward reference in the foreign key declarations. Ignore that and the circular dependencies for the purpose of this exercise.)

We have built and populated a database based on this schema. We ran a collection of queries and obtained execution plans by using `EXPLAIN ANALYZE`.

Answer the question associated with each of the execution plans below:

a. How many tuples are there in the `People` table?

```

Aggregate
  (cost=189.00..189.01 rows=1 width=0)
  (actual time=2.899..2.899 rows=1 loops=1)
  -> Seq Scan on people
        (cost=0.00..164.00 rows=10000 width=0)
        (actual time=0.010..1.594 rows=10000 loops=1)
Total runtime: 2.932 ms

```

b. What query produced the following execution plan?

```

Index Scan using people_pkey on people
  (cost=0.00..8.41 rows=9 width=21)
  (actual time=0.017..0.021 rows=9 loops=1)
  Index Cond: (id > 9990)
Total runtime: 0.048 ms

```

c. Give a relational algebra expression corresponding to the query execution plan below. Use the notation that we have been using in lectures for relational algebra operators (σ as `Sel`, π as `Proj`, \bowtie as `Join`) in the tree. Some examples:

- $\sigma_{x>1}R$ would be written as `Sel[x>1](R)`
- $\pi_{a,b}T$ would be written as `Proj[a,b](T)`

- $(R \bowtie_{a=d} S)$ would be written as $(R \text{ Join}[a=d] S)$

There is no need to show operations like Hash which are part of the hash-join operation.

```
Nested Loop
(cost=120.38..644.57 rows=27 width=13)
(actual time=2.786..10.844 rows=29 loops=1)
-> Hash Join
(cost=120.38..636.60 rows=27 width=4)
(actual time=2.770..10.735 rows=29 loops=1)
Hash Cond: (w.project = j.id)
-> Seq Scan on workson w
(cost=0.00..415.06 rows=26906 width=8)
(actual time=0.012..3.833 rows=26906 loops=1)
-> Hash
(cost=120.31..120.31 rows=5 width=4)
(actual time=2.586..2.586 rows=12 loops=1)
-> Hash Join
(cost=19.51..120.31 rows=5 width=4)
(actual time=0.971..2.583 rows=12 loops=1)
Hash Cond: (j.fundedby = d.id)
-> Seq Scan on projects j
(cost=0.00..82.00 rows=5000 width=8)
(actual time=0.007..0.769 rows=5000 loops=1)
-> Hash
(cost=19.50..19.50 rows=1 width=4)
(actual time=0.825..0.825 rows=1 loops=1)
-> Seq Scan on departments d
(cost=0.00..19.50 rows=1 width=4)
(actual time=0.022..0.822 rows=1 loops=1)
Filter: (name = 'Sales Department'::text)
-> Index Scan using people_pkey on people p
(cost=0.00..0.28 rows=1 width=17)
(actual time=0.003..0.003 rows=1 loops=29)
Index Cond: (p.id = w.person)
Total runtime: 42.917 ms
(14 rows)
```

- d. If I run the query that produced the above plan a second time, it takes only 10.912 ms, and subsequent executions of this query are all around 11 ms. Explain why.

Show all working.

Instructions:

- Type your answer to this question into the file called `q7.txt`
- Submit via: **give cs9315 exam_q7 q7.txt**
or via: Webcms3 > exams > Final Exam > Q7 submission > Make Submission

End of Question