0/5 Questions Answered

Vitamin 7

STUDENT NAME

Search students by name or email...

Q1 True or False
4 Points

Q1.1

The order in which we inner join relations in a query can affect the set of tuples that are produced by the query (e.g. R ⋈ S vs. S ⋈ R).

○ True

○ False

Q1.2

Pushing down selections in a query can affect the cost of executing the query.

○ True

○ False

Q1.3

Pushing down selections in a query can affect the set of tuples that are produced by the query.

- True
- False

Q1.4

The order in which we join relations in a query can affect the cost of executing the query (e.g. R ⋈ S vs. S ⋈ R).

- True
- False

Save Answer

Q2 Query Optimizer

4 Points

Q2.1

The primary goal of a query optimizer is to

- translate a relational algebra expression into a relational algebra expression
- translate a relational algebra expression into a query plan
- translate a query plan into a query plan
- translate a query plan into a relational algebra expression

Q2.2

Given n relations, how many ways (i.e. how big in big O notation) are there to join the n relations?

- small (linear in n)
- not that big (polynomial in n)
- pretty big (exponential in n)

- ○ really really big (factorial in n)

Q2.3

How many left-deep query plans are there for n relations?

- ○ small (linear in n)
- ○ not that big (polynomial in n)
- ○ pretty big (exponential in n)
- ○ really really big (factorial in n)

Q2.4

Given n relations, the Selinger Optimizer considers how many plans in pass i?

- ○ small (linear in n)
- ○ not that big (polynomial in n)
- ○ pretty big (exponential in n)
- ○ really really big (more than exponential in n)

Save Answer

Q3 Selinger Algorithm

4 Points

Q3.1

Given a relational algebra expression, the Selinger query optimizer enumerates

- ○ all possible query plans
- ○ all left-deep query plans
- ○ some, but not all, left-deep query plans

Q3.2

The Selinger optimizer outputs a globally optimal query plan.

○ True

○ False

The Selinger optimizer outputs a globally optimal left-deep query plan.

○ True

○ False

Given a perfect query plan cost estimator, the Selinger optimizer would output a globally optimal left-deep query plan.

○ True

○ False

Save Answer

Q4 System R Pass 1

3 Points

Given the following schemas:

```
CREATE TABLE pokemon (
    pokedex INT PRIMARY KEY,
    name VARCHAR[11],
    class VARCHAR[9],
    trainer_id INT
);
```

```
CREATE TABLE trainers (
CREATE TABLE trainers (
    id INT PRIMARY KEY,
    name VARCHAR[14],
    hometown VARCHAR[10],
    journey_time INT,
    cities_visited INT
);
```

Pokemon Table Info

- Alt 1 index of height 3 on `pokedex` column
- 32 data pages with records with `pokedex` <= 250
- 47 data pages with records with `pokedex` > 250
- 17 data pages with records with `pokedex` > 700

Trainers Table Info

- Alt 2 index of height 3 on `journey_time`
- Alt 3 index of height 2 on `cities_visited`

Assume that all nodes in an index fit on a single page.

Suppose we evaluate the following query:

```
SELECT *
FROM pokemon INNER JOIN trainers
ON pokemon.trainerid = trainers.id AND pokemon.pokedex > 250
GROUP BY trainer.cities_visited;
```

Q4.1

How many I/Os will a full scan of the Pokemon table take?

Enter your answer here

Q4.2

How many I/Os will an index scan of the Pokemon table take?

Hint: Remember to push down any single column conditions in pass 1.

Enter your answer here

Q4.3

Based on your answers in 4.1 and 4.2, as well as the following access patterns, which patterns will move on to the next pass of System R?

Access Patterns for the Trainers table:

- Full Scan = 100 I/Os
- Index Scan on journey_time = 70 I/Os
- Index Scan on cities_visited = 140 I/Os.

- ☐ Full Scan – pokemon
- ☐ Index Scan – pokemon.pokedex
- ☐ Full Scan – trainers
- ☐ Index Scan – trainers.journey_time
- ☐ Index Scan – trainers.cities_visited

Save Answer

Q5 System R Pass i

1 Point

Suppose we run the following query:

```
SELECT *
FROM pets INNER JOIN stores
```

```
ON pets.storeid = stores.id
INNER JOIN counties
ON stores.county = counties.name
ORDER BY counties.name
```

Suppose also that we've only implemented the SMJ and BNLJ joins in our database.

Q5.1

In pass 2 of System R, which of the following joins will we consider:

- [ ] Pets and Stores
- [ ] Pets and Counties
- [ ] Counties and Stores

Q5.2

The following are estimates for join costs. In reality, we would make these estimates with selectivity estimation and the joins' cost formulas, but we skip those steps here for convenience:

(For simplicity Pets = P, Stores = S, Counties = C)

- P BNLJ S: 80 I/Os
- S BNLJ P: 100 I/Os
- P SMJ S: 220 I/Os
- P BNLJ C: 230 I/Os
- C BNLJ P: 140 I/Os
- P SMJ C: 300 I/Os
- S BNLJ C: 150 I/Os
- C BNLJ S: 120 I/Os
- S SMJ C: 180 I/Os

Which passes will move on?

Hint: If you're having trouble, check out note 9 section 6 on query optimization on the website.

- ☐ P BNLJ S
- ☐ S BNLJ P
- ☐ P SMJ S
- ☐ P BNLJ C
- ☐ C BNLJ P
- ☐ P SMJ C
- ☐ S BNLJ C
- ☐ C BNLJ S
- ☐ S SMJ C

Save Answer

Save All Answers

Submit & View Submission