

CSE 414 Midterm

April 28, 2017

Name: _____

Question	Points	Score
1	35	
2	15	
3	30	
4	21	
Total	101	

- **Do not open** the test until instructed to do so.
- The test is closed book and electronics. You are allowed only one page of notes (double sided).
- You have **50 minutes** to complete the test.

Problem 1 (SQL)

Part A: Schema

You would like to create a database to store recipes along with information about local grocery stores that can supply the ingredients of those recipes. In more detail, you would like your database to contain the following relations:

- `Recipe(rid, rname)`, where `rid` is a unique integer ID for the recipe and `rname` is a string name.
- `Ingredient(rid, pname, amount)`, where `rid` is the ID of a known recipe, `pname` is the name of the ingredient, and `amount` is the weight in ounces of that ingredient in the recipe.
- `Grocer(gid, gname, city)`, where `gid` is a unique ID for the grocer, `gname` is their name, and `city` is a name of the city where the grocer is located.
- `Sells(gid, pname, price)`, where `gid` is the ID of a known grocer, `pname` is the name of an ingredient, and `price` is the price per ounce of the ingredient at that grocer.

Most of the SQL to create these relations is provided below. However, some important parts are still missing. **Fill in the remaining parts** so that the SQL will create a schema for these relations that properly reflects all parts of the description above.

```
CREATE TABLE Recipe(  
  rid INT,  
  rname VARCHAR(100),
```

```
CREATE TABLE Grocer(  
  gid INT,  
  gname VARCHAR(100),  
  city VARCHAR(100),
```

```
CREATE TABLE Ingredient(  
  rid INT,  
  pname VARCHAR(100),  
  amount FLOAT,
```

```
CREATE TABLE Sells(  
  gid INT,  
  pname VARCHAR(100),  
  price FLOAT,
```

Part B: Queries

Each item below describes, in English, a query to perform on the database using the schema from the previous page. Translate each query into SQL.

1. Get the names of recipes that use 'condensed milk' as an ingredient. You may return them in any order.
2. List each ingredient by name along with the cheapest price at which it is sold by any grocer. Order the results alphabetically by ingredient name.
3. List each recipe by name along with the total cost to purchase all its ingredients, in the required amounts, assuming each ingredient is purchased at the cheapest price at which it is sold by any grocer. You can return the rows in any order. (It is okay to use a subquery in this case.)

Problem 2 (Query Optimization)

Below, we consider possible execution plans for the following query, which returns the names of grocers that sell at least one very inexpensive ingredient:

```
SELECT gname
FROM Grocer G, Sells S
WHERE G.gid = S.gid AND S.price < 0.26
```

Assume the following statistics about the two relations:

Table	# tuples	# blocks
Grocer	1000	25
Sells	10,000	100

Assume the following statistics about the price column of S:

# distinct	Low	High
1000	0.01	2.50

At the bottom of this page, we provide some formulas from lecture that may be useful if you do not already have them in your own notes. The problems start **on the next page**.

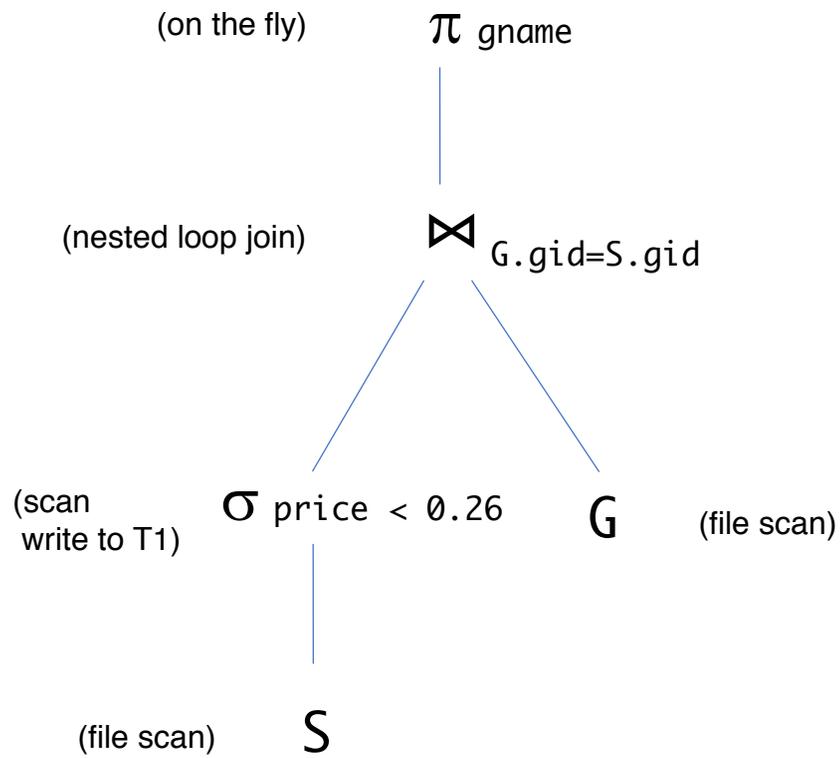
Estimated cost of $X \text{ JOIN } Y$:

- Using a nested loop, the cost is $B(X) + B(X) B(Y)$, where $B(X)$ is # blocks in X .
- Using a clustered index on $Y(A)$, the cost is $B(X) + T(X) B(Y) * E$, where $T(X)$ is # tuples in X , and E is the selectivity of the condition $A = c$.
- Using an unclustered index on $Y(A)$, the cost is $B(X) + T(X) T(Y) * E$.

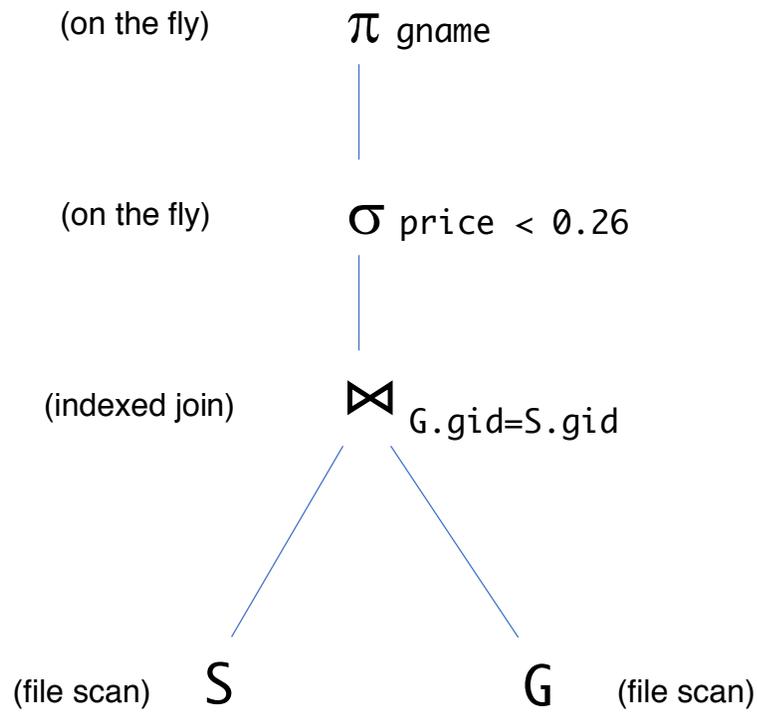
Estimated selectivity of conditions:

- For $A = c$, the selectivity is $1 / (\# \text{ distinct values of } A)$
- For $A < c$, the selectivity is $(c - \text{lowest value of } A) / (\text{highest} - \text{lowest value of } A)$

1. The total cost of the plan below is _____ disk block I/Os.



2. The total cost of the plan below is _____ disk block I/Os.



3. Circle the plan that is probably faster in this case:

nested loop join (1)

indexed join (2)

2. Consider the following query, which finds the IDs of grocers that only sell cheap items (i.e., grocers that do not sell any extremely expensive items):

```
SELECT DISTINCT gid
FROM Grocer
WHERE gid NOT IN
  (SELECT DISTINCT gid
   FROM Grocer NATURAL JOIN Sells
   WHERE price > 2.00)
```

- a. Write an extended relational algebra expression that computes the result.

- b. Write a datalog rule that computes the same result or explain why it is impossible to do so.

3. Consider the following query, which shows each recipe along with the total amount of ingredients (in ounces) used in that recipe:

```
SELECT rname, sum(amount)
FROM Recipe NATURAL JOIN Ingredient
GROUP BY rid, rname
```

- a. Write an extended relational algebra expression that computes the result.

- b. Write a datalog rule that computes the same result or explain why it is impossible to do so.

Problem 4 (Multiple-Choice / Short Answer)

1. Consider the following SQL query using relations $S(A, B, C)$ and $T(C, D)$:

```
SELECT *
FROM S, T
WHERE _____
```

Circle those columns that can be legally used in the WHERE clause:

A B C D

2. Consider the following SQL query using relation $T(A, B)$:

```
SELECT _____
FROM T
GROUP BY A
```

Circle those expressions that can be legally used in the SELECT clause:

A B sum(A) sum(B)

3. Give an example of an error (illegal SQL) that will not be caught by SQLite. (Limit your answer to one sentence.)

4. Suppose that $S(A, \dots)$ is a relation containing m rows and $T(F, \dots)$ is a relation containing n rows. Assume that A is the primary key of S and that F is a foreign key reference to $S(A)$. Suppose that we perform an inner join between S and T on $S.A = T.F$. Circle the maximum number of rows that can be output:

m n m+n m*n

5. Circle the one that best describes how many logical and physical query plans will be considered by a cost-based query optimizer:
- a. one logical plan, one physical plan
 - b. one logical plan, multiple physical plan
 - c. multiple logical plans, one physical plan
 - d. multiple logical plans, multiple physical plans
6. Order the following four query languages by expressive power: standard RA, extended RA, SELECT-FROM-WHERE (SFW), and SQL. Between each two in the ordering, you may write $<$ or $=$ to indicate strictly more powerful or equal power.
7. Suppose we have a relation $T(A, B, C)$ and we create an index on (A, B) . For each type of index listed below, circle those columns that can searched for an exact match using an index of that type on (A, B) .

Hash index on (A, B) : A B C (A, B) (B, C)

B+ tree index on (A, B) : A B C (A, B) (B, C)