1 Question 1

A database stores warehouses and products. Here is the schema:

```sql
CREATE TABLE warehouse(
    name VARCHAR(100) PRIMARY KEY,
    zip_code VARCHAR(20) NOT NULL
);

CREATE TABLE product(
    id int PRIMARY KEY,
    description VARCHAR(600) NOT NULL,
    availableAt VARCHAR(100),
    FOREIGN KEY (availableAt) REFERENCES warehouse(name)
);
```

Here are some facts about the data: - There are 500,000 products and 10,000 warehouses in the database. - The average size of a warehouse record is 50 bytes. - The average size of a product record is 100 bytes. - There are 500 known zip codes, and warehouses are uniformly distributed across zip codes. - Scanning through a table takes 0.1 ms per block. - The disk block size is 8K = 8192 bytes. - Buffer size is 4 blocks.

You want to find all products stored in Providence warehouses at the zip code “02904-2413”. For each of the following query plans, find the approximate runtime. Then, discuss which plan should be selected and why:

1. Select warehouses (02904-2413), and then join with product using a block nested loop join.
2. Select warehouses (02904-2413), and then join with product using a sort-merge join.
3. Join product with warehouse using a block nested loop and then select the merchandise in (02904-2413).

2 Question 2

Consider the relations r1(A, B, C), r2(C, D, E), and r3(E, F), with primary keys A, C, and E, respectively. Assume that r1 has 1000 tuples, r2 has 1500 tuples, and r3 has 750 tuples. Estimate the size of $r_1 \bowtie r_2 \bowtie r_3$, and give an efficient strategy for computing the join. Hint: your strategy should involve the creation of indices.

3 Question 3

Consider a relation r1(A, C) and r2(A, B), under what conditions are the following queries equivalent? Hint: It might help to convert this to relational algebra first. Hint: Think about the size of r1 joined on r2.

```sql
SELECT A, B, SUM(C)
FROM r1 JOIN r2
GROUP BY A, B;

SELECT A, B, C
FROM r2 JOIN
    (SELECT A, SUM(C) as C
     FROM r1
     GROUP BY A)
```
4 Question 4

Given the following relational schema, SQL query, and DB statistics, calculate the approximate number of block transfers for each of the following scenarios.

Schema:
sells (product_id, store_id, price, volume)

SQL Query:
SELECT product_id
FROM sells
WHERE (store_id = 42 
    AND volume < 32);

DB Statistics: - Number of files the relation is contained in: 1 - Number of tuples in the relation: 10M - Number of disk blocks containing the relation tuples: 100K - Distribution of the store_id attribute: Uniform (min=0, max=999) - Distribution of the volume attribute: Normal (μ = 50, σ = 10)

1. There is no index on the database.
2. There is a primary, sparse B+ tree index on the attribute Sells.store_id, and the cost of traversing this index is b1.
3. There is a primary, sparse B+ tree index the attribute Sells.volume, and the cost of traversing this index is b2.
4. There is a secondary, dense B+ tree index on the attribute Sells.store_id, and the cost of traversing this index is b3.
5. There is a secondary, dense B+ tree index on the attribute Sells.volume, and the cost of traversing this index is b4.

Feedback

As this is a new course, we appreciate any feedback you have for us! If you enjoyed this assignment, hated this assignment, or have other thoughts to share, please do so here!