

Q) Create an aggregation query where PostgreSQL uses (in-memory) hash-aggregation.

Solution:

Here's an example query that uses hash-aggregation in PostgreSQL:

```
SELECT dept_name, AVG(salary) as avg_salary
FROM instructor
GROUP BY dept_name;
```

In this query, we are calculating the average salary for each department. To do this, we group the records by the dept_name column using the GROUP BY clause, and then calculate the average salary using the AVG function. Since we are using an aggregation function, PostgreSQL needs to group the records and compute the aggregate values. In this case, PostgreSQL can use hash-aggregation to perform this task more efficiently. The hash-aggregation algorithm works by hashing the grouping keys (in this case, the dept_name column), and then computing the aggregate values for each group in-memory.

```
explain SELECT dept_name, AVG(salary) as avg_salary
FROM instructor
GROUP BY dept_name;
```

QUERY PLAN

```
-----
HashAggregate  (cost=16.60..19.10 rows=200 width=90)
  Group Key: dept_name
    -> Seq Scan on instructor  (cost=0.00..14.40 rows=440 width=72)
(3 rows)
```

Time: 0.972 ms

```
explain analyze SELECT dept_name, AVG(salary) as avg_salary
FROM instructor
GROUP BY dept_name;
```

QUERY PLAN

```
-----
HashAggregate  (cost=16.60..19.10 rows=200 width=90) (actual
time=0.115..0.132 rows=17 loops=1)
  Group Key: dept_name
    Batches: 1  Memory Usage: 40kB
    -> Seq Scan on instructor  (cost=0.00..14.40 rows=440 width=72)
(actual time=0.020..0.027 rows=50 loops=1)
Planning Time: 0.160 ms
Execution Time: 0.188 ms
(6 rows)
```

Time: 0.810 ms