

Q) Create a query where PostgreSQL uses bitmap index scan on relation takes. Explain why PostgreSQL may have chosen this plan.

Solution:

Here's an example query that could use a bitmap index scan:

```
SELECT *
FROM takes
WHERE course_id = 'CS101';
```

PostgreSQL may choose a bitmap index scan in this case because it is an efficient way to search for values in a large table using an index.

When the query is executed, PostgreSQL can use the bitmap index to create a bitmap (a series of 1s and 0s) indicating which rows in the table match the query condition (`student_id = 123`). The bitmap index scan retrieves this bitmap directly from the index and uses it to read only the necessary data pages from the table. This can be much faster than scanning the entire table.

A bitmap index scan is particularly useful when the query condition matches a relatively small fraction of the rows in the table. In such cases, the overhead of using the index and the bitmap can be more than offset by the savings in I/O and CPU time.

```
explain SELECT * FROM takes WHERE course_id = 'CS101';
               QUERY PLAN
```

```
-----
Seq Scan on takes  (cost=0.00..595.00 rows=1 width=24)
  Filter: ((course_id)::text = 'CS101'::text)
(2 rows)
```

```
explain analyze SELECT * FROM takes WHERE course_id = 'CS101';
               QUERY PLAN
```

```
-----
Seq Scan on takes  (cost=0.00..595.00 rows=1 width=24) (actual
time=7.606..7.607 rows=0 loops=1)
  Filter: ((course_id)::text = 'CS101'::text)
  Rows Removed by Filter: 30000
  Planning Time: 0.192 ms
  Execution Time: 7.641 ms
(5 rows)
```