quiz on Scans

This set of questions relates to the results of the EXPLAIN query on simple queries.

The quiz is available as a google form https://forms.gle/5fdjpUSr8YaqA9KNA

Q1. What elements are used by the query optimizer to compute the cost function

- CPU cycles and I/O accesses
- query time
- user satisfaction

Q2. What is a Sequential scan

in the query

```SQL
select * from trees;
```

The EXPLAIN diagram uses a sequential scan

What is a sequential scan ?

- the algo goes over all the table rows one by one
- the algo goes over all the table rows in bulk of 8192 blocks
- the algo goes over all the table columns one by one and then rows by rows

Q3. difference between EXPLAIN and EXPLAIN ANALYZE

- same. EX AN outputs out more data like the temperature of the query
- ANALYZE actually executes the query
- ANALZYE gets real data on the data distribution EXPLAIN alone uses default stats

Q4. how does the query planner chooses the algorithms to execute the query

- the nature of the tables and the data
- the type of filter (WHERE) operator : =, <, between, like '%string' etc
- the data types of the columns involved in the filters
- the number of rows of the table, the subqueries etc
- limit, order by
- full text search is a topic by itself
- the performance of the machine
- all of the above

Q5. compare the 2 queries. why has the cost gone up when we add a filter

```sql
epita@airdb=> explain select * from passenger;
                              QUERY PLAN
----------------------------------------------------------------------
 Seq Scan on passenger  (cost=0.00..334838.55 rows=16311855 width=47)
(1 row)
```

so we have

- cost: 334,838
- rows: 16,311,855

and

```sql
epita@airdb=> explain select * from passenger where age > 68;
                                   QUERY PLAN
-------------------------------------------------------------------------
 Seq Scan on passenger  (cost=0.00..375618.19 rows=3613778 width=47)
   Filter: (age > 18)
(5 rows)
```

so we have

- cost: 375,618
- rows: 3,613,778

In the second query the cost has gone up although less rows are returned.

Why ?

- Seq Scan still has to go over all the 16.3M passenger rows and has to check the condition for each row
- Older people travel less so the algorrithm only finds matching records at the end of the table
- Intermediate results have to be stored before the filter can be applied. This extra I/O adds to the cost function

Q6. Index and Bitmap scans

The passenger table has a id primary key

Explain the difference between Index Scan and Index Heap scan in the 2 examples

```sql
 explain select * from passenger where passenger_id = 8888;
                                   QUERY PLAN
------------------------------------------------------------------------------
 Index Scan using passenger_pkey on passenger  (cost=0.43..8.45 rows=1 w
   Index Cond: (passenger_id = 8888)
(2 rows)
```

and

```sql
explain select * from passenger where passenger_id < 8888;
                                   QUERY PLAN
------------------------------------------------------------------------------
 Bitmap Heap Scan on passenger  (cost=184.03..31558.41 rows=9754 width=
   Recheck Cond: (passenger_id < 8888)
   ->  Bitmap Index Scan on passenger_pkey  (cost=0.00..181.59 rows=975
         Index Cond: (passenger_id < 8888)
```

Which if the following statement is true

Bitmap Heap Scans are often more efficient than Index Scan when retrieving a larger number of rows. They are the same thing but bitmap scanning is for boolean data types. When retrieving a small number of rows, Index Scan is faster because it avoids building the bitmap. Bitmap Scanning is a 2 step process: create the bitmap, then accesses the table.

Q7. Index Scans

- Used when: Fetching a small number of rows based on an index condition.
- uses the index to find the location of rows matching the condition.
- Efficient for reading large portions of a table, but can be slow for selective queries on large tables.
- Reads all rows in the table sequentially, one after another.

Q8. Bitmap Heap Scans

- Used when: Fetching a moderate number of rows based on an index condition.
- More efficient than Index Scan when retrieving a larger number of rows, but still selective enough not to warrant a full Sequential Scan.
- Very efficient for retrieving a small number of rows, especially with highly selective conditions.
- Uses an index to create the bitmap, then accesses the table.

Q9. order of operations

In the following query plan

```sql
explain select * from passenger order by random() limit 50;
                                    QUERY PLAN
-----------------------------------------------------------------------
 Limit  (cost=917486.28..917486.41 rows=50 width=55)
   ->  Sort  (cost=917486.28..958265.92 rows=16311855 width=55)
         Sort Key: (random())
         ->  Seq Scan on passenger  (cost=0.00..375618.19 rows=16311855
 JIT:
   Functions: 3
   Options: Inlining true, Optimization true, Expressions true, Deformi
(7 rows)
```

what is the order of operations ?

- bottom to top: Sequential scan on the whole table, then sort by random, then limit to 50 rows
- top to bottom: Limit to 50 rows than sorting by random and finally scanning on the rows
- does not matter. Each step can be done independently