Solution 4

Solved by Aaron Colin Foote, acf502, 4258770

- Width (total number of blocks at leaf level) of B*-tree implementing an index on manuf is equal to $\mathbf{w_m}$.
  $\mathbf{w_m}$ is equal to 50.
  - “an attribute manuf is indexed”
  - “an attribute manuf has 50 distinct values”

- Each block contains 5000/1000 rows.
  This means each block records 5 rows (**blocking factor** = 5)
  - “a relational table REPAIR occupies 1000 data blocks”
  - “a relational table REPAIR contains 5000 rows”

- Fanout = 20
  - “all indexes are implemented as B*-trees with a fanout equal to 20”

- $\log_{20}50 \approx 1.306$
- $\log_{20}5000 \approx 2.843$

Query 1:

SELECT DISTINCT manuf
FROM REPAIR;

We have to read w_m blocks.

Index processing:

w_m blocks (horizontal traversal through leaf level on an index on manuf)
50 index values read.

Query 2:

SELECT rego
FROM REPAIR
WHERE manuf = 'Toyota' AND model = 'Corolla';

We have to read $[\log_{20}50 +1] + ((5000/50) + 5000/(50*5))/2$ blocks

Index processing:

$[\log_{20}50 +1] \approx [1.306 + 1] \approx 2.306$ blocks
Total number of row identifiers found 5000/50 = 100 row identifiers

Table processing:

The best case (full clustering): 5000/(50*5) = 20 blocks
The worst case (each row in a different block): 5000/50 = 100 blocks

Query 3:

SELECT rego
FROM REPAIR
WHERE TO_CHAR(rdate,'YYYY') = '2012';

We have to read 1000 data blocks because the entire relational table REPAIR has to be scanned.

Table processing:

1000 blocks (total block capacity of table REPAIR (5000/5 (blocking factor)))

Query 4:

SELECT COUNT(*)
FROM REPAIR
WHERE manuf = 'Honda';

We have to read floor($\log_{20}$50 +1) = 2 blocks.

Index processing:

floor($\log_{20}$50 +1) ≈ floor(1.306 + 1) = 2 index blocks

Counting row identifiers:

0 data blocks

Query 5:

*SELECT **
FROM REPAIR
WHERE rego = 'PKR856' AND address ='15 Station St.'
AND rdate = '15-DEC-2010';

We have to read [$\log_{20}$5000 +1] + 1 blocks because the primary key (rego, address, rdate) is automatically indexed.

≈ [2.843 + 1] + 1
≈ 4.843 blocks

Query 6:

SELECT COUNT(model)
FROM REPAIR;

We have to read 1000 blocks because the entire relational table REPAIR has to be scanned.

Table processing:

1000 blocks (total block capacity of table REPAIR (5000/5 (blocking factor)))

Query 7:

SELECT COUNT(DISTINCT model)
FROM REPAIR;

We have to read 1000 blocks because the entire relational table REPAIR has to be scanned.

Table processing:

1000 blocks (total block capacity of table REPAIR (5000/5 (blocking factor)))

Query 8:

SELECT manuf, COUNT(*)
FROM REPAIR
GROUP BY manuf;

We have to read w_m (w_m = 50) blocks

Index processing:

w_m (w_m = 50) blocks

Query 9:

*SELECT **
FROM REPAIR
ORDER BY rego;

We have to read 1000 blocks because the entire relational table REPAIR has to be scanned.

Table processing:

1000 blocks (total block capacity of table REPAIR (5000/5 (blocking factor)))

Query 10:

*SELECT **
FROM REPAIR
WHERE rego= 'PKR856' AND manuf = 'Honda';

We have to read floor($\log_{20}$50 +1) = 2 blocks.

Index processing:

floor($\log_{20}$50 +1) ≈ floor(1.306 + 1) = 2 blocks

Table processing:

The best case (full clustering): 5000/(50*5) = 20 blocks
The worst case (each row in a different block): 5000/50 = 100 blocks