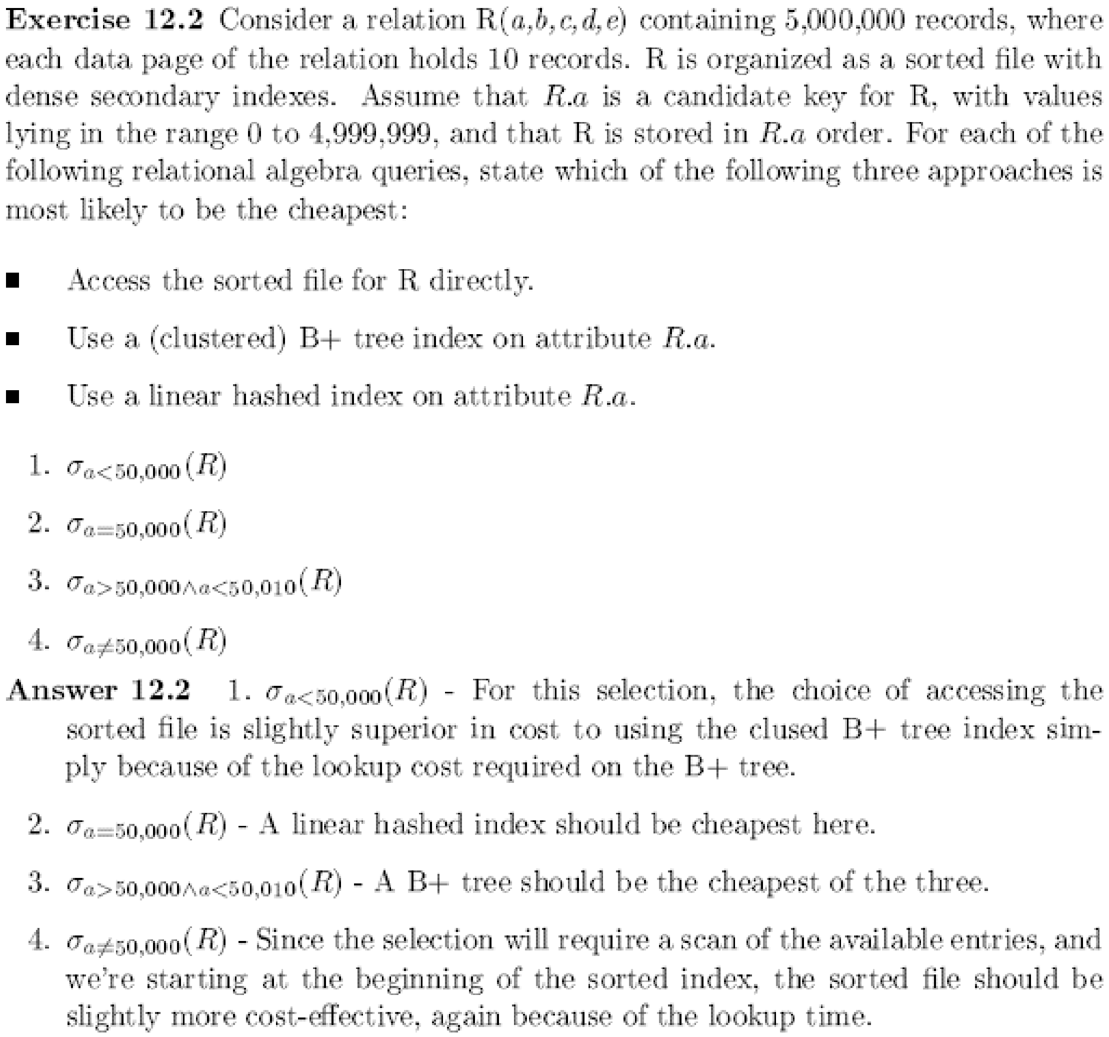
Question 12.2



Exercise 12.2 Consider a relation R(a,b,c,d,e) containing 5,000,000 records, where each data page of the relation holds 10 records. R is organized as a sorted file with dense secondary indexes. Assume that $R.a$ is a candidate key for R, with values lying in the range 0 to 4,999,999, and that R is stored in $R.a$ order. For each of the following relational algebra queries, state which of the following three approaches is most likely to be the cheapest:

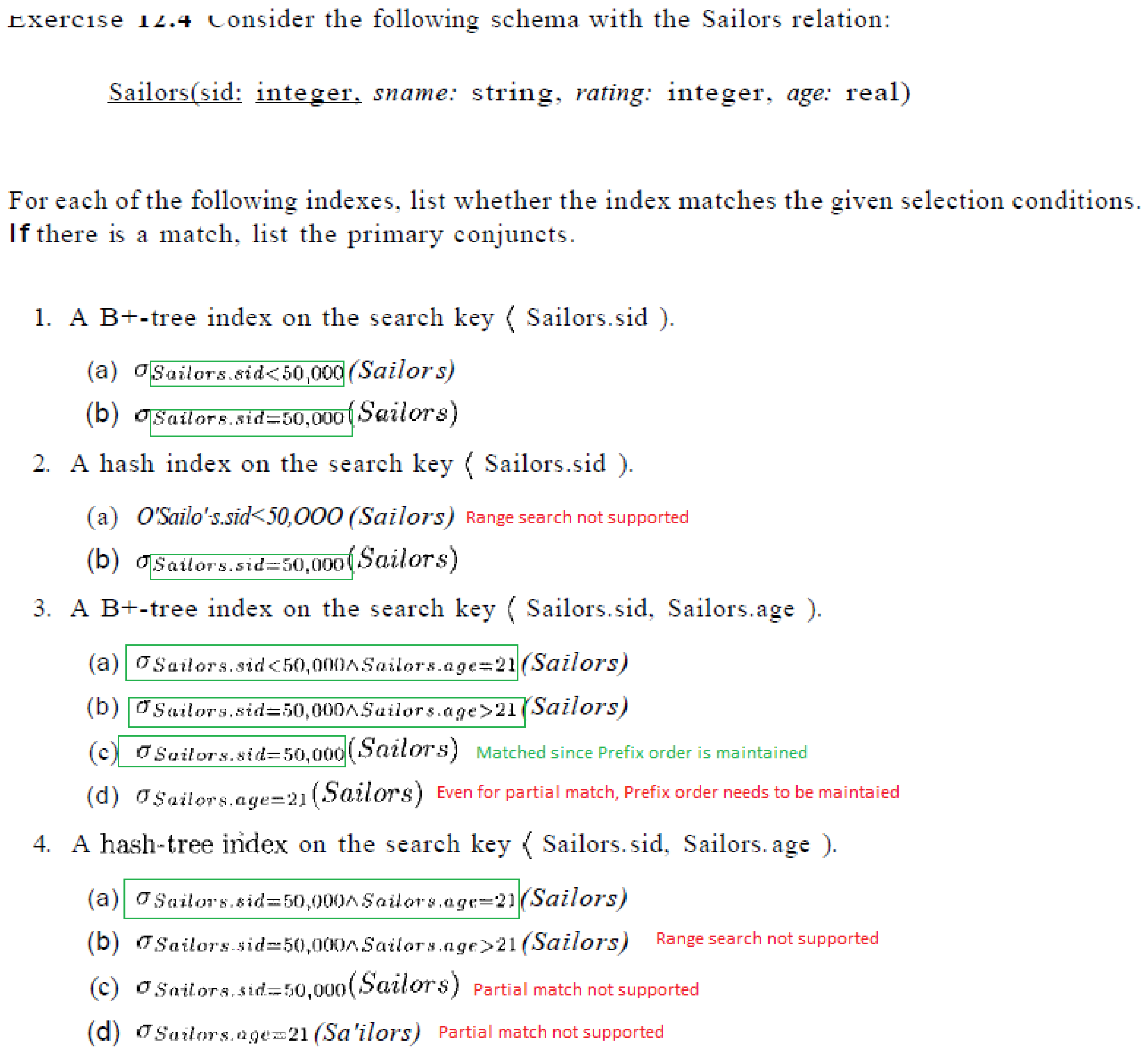
- Access the sorted file for R directly.
- Use a (clustered) B+ tree index on attribute $R.a$.
- Use a linear hashed index on attribute $R.a$.

1. $\sigma_{a<50,000}(R)$
2. $\sigma_{a=50,000}(R)$
3. $\sigma_{a>50,000 \wedge a<50,010}(R)$
4. $\sigma_{a\neq 50,000}(R)$

Answer 12.2

1. $\sigma_{a<50,000}(R)$ - For this selection, the choice of accessing the sorted file is slightly superior in cost to using the clused B+ tree index simply because of the lookup cost required on the B+ tree.
2. $\sigma_{a=50,000}(R)$ - A linear hashed index should be cheapest here.
3. $\sigma_{a>50,000 \wedge a<50,010}(R)$ - A B+ tree should be the cheapest of the three.
4. $\sigma_{a\neq 50,000}(R)$ - Since the selection will require a scan of the available entries, and we're starting at the beginning of the sorted index, the sorted file should be slightly more cost-effective, again because of the lookup time.

Question 12.4



Exercise 12.4 Consider the following schema with the Sailors relation:

Sailors(sid: integer, *sname:* string, *rating:* integer, *age:* real)

For each of the following indexes, list whether the index matches the given selection conditions. If there is a match, list the primary conjuncts.

1. A B+-tree index on the search key ⟨ Sailors.sid ⟩.
   - (a) $\sigma_{\boxed{Sailors.sid<50,000}}(Sailors)$
   - (b) $\sigma_{\boxed{Sailors.sid=50,000}}(Sailors)$
2. A hash index on the search key ⟨ Sailors.sid ⟩.
   - (a) *O'Sailo's.sid<50,OOO (Sailors)* Range search not supported
   - (b) $\sigma_{\boxed{Sailors.sid=50,000}}(Sailors)$
3. A B+-tree index on the search key ⟨ Sailors.sid, Sailors.age ⟩.
   - (a) $\boxed{\sigma_{Sailors.sid<50,000 \wedge Sailors.age=21}}(Sailors)$
   - (b) $\boxed{\sigma_{Sailors.sid=50,000 \wedge Sailors.age>21}}(Sailors)$
   - (c) $\boxed{\sigma_{Sailors.sid=50,000}}(Sailors)$ Matched since Prefix order is maintained
   - (d) $\sigma_{Sailors.age=21}(Sailors)$ Even for partial match, Prefix order needs to be maintaied
4. A hash-tree index on the search key ⟨ Sailors.sid, Sailors.age ⟩.
   - (a) $\boxed{\sigma_{Sailors.sid=50,000 \wedge Sailors.age=21}}(Sailors)$
   - (b) $\sigma_{Sailors.sid=50,000 \wedge Sailors.age>21}(Sailors)$ Range search not supported
   - (c) $\sigma_{Sailors.sid=50,000}(Sailors)$ Partial match not supported
   - (d) $\sigma_{Sailors.age=21}(Sa'ilors)$ Partial match not supported

Solution

Marked in question.