

Guidelines for choosing indexes.

INT206 2/2020

Guidelines

1. Do not index small relations.
2. Index the primary key if it is not a key of file organization.
3. Add an index to a foreign key that is accessed frequently.
4. Add a secondary index to any attribute that is heavily used.
5. Add an index on attributes that are frequently involved in:
 - a. Selection (WHERE clause) and join criteria
 - b. ORDER BY
 - c. GROUP BY
 - d. Other operations involving sorting (UNION or DISTINCT)
6. Add an index on attributes used in built-in aggregate functions or built-in functions (*for example*)
Select branchno, avg(salary)
From staff
Group by branchno ;
Create index staff_sal_idx on staff(branchno,salary) ;
This may allow DBMS to perform the query data in the index alone (called index-only plan)
7. Add an index on attributes for an index-only plan.
8. Do not index an attribute or relation frequently updated.
9. Do not index an attribute that query returns a lot of rows.
10. Do not index attributes of long character strings.
11. A combination of columns used together in query conditions may be good candidates for indexes if the joint conditions return few rows.
12. Tables with a lot of insertions and deletions should not have many indexes.
13. Stable columns with few values (low cardinality) are good candidates for bitmap indexes if the columns appear in WHERE conditions.
14. Avoid indexes on combinations of columns. Most optimization components can use multiple indexes on the same table. An index on a combination of columns is not as flexible as multiple indexes on individual columns of the table.