

Physical design

The purpose is provide good performances for database applications; due to data independence property is possible change physical design without have effects on applications.

The physical schema of the database is table organization and indices and the physical file organization can be done:

- . unordered (heap);
- . ordered (clustered);
- . hash;
- . clustering of several relations.

Indices

Indices can be realized with different structures:

- . trees;
- . hash.

Indices also are:

- . clustered, primary index when only one index is defined on the table;
- . unclustered, secondary index when many indices can be defined on the table.

It is important notice that all indices must be updated when table is updated.

Criteria

Usually the primary key is used for selections and joins by creating an index on the primary key: this index can be clustered (better) or unclustered; moreover if an attribute is very relevant is useful create index on that attribute.

Most important aspects to improve performances are:

- . no create index in small tables, is better a direct access;
- . no create index on low cardinality attributes (gender);
- . simple predicates: index can be clustered if there is low selectivity (unclustered othewise):

- . equally predicate ($=$) \implies hash;
- . range predicate ($>$, $<$, \geq , \leq) \implies trees;
- . many simple predicates: index can be composed, or is possible create more indices individualizing the proper key order;
- . join operations:
 - . if there is an index on the inner table is possible to use the nested loop;
 - . with merge scan best index is B⁺-tree clustered on the join attribute;
- . group by:
 - . create an index on the grouping attributes;
 - . evaluate possibility *push down* when this operation reduce significantly the number of terms.