What is an Index in Oracle?

An index is a performance-tuning method of allowing faster retrieval of records.

An index creates an entry for each value that appears in the indexed columns.

By default, Oracle creates B-tree indexes.

Types of indexes in Oracle

Oracle defines two types of indexes: the B-Tree (Balanced Tree) Index and the Bitmap Index.

B-Tree Index is the default Oracle index created whenever we use the CREATE INDEX command. It compiles a list of values divided into ranges and associates a key with a single row or range of rows. This structure works efficiently in a majority of scenarios, including both the exact match and range of searches.

In its turn, a B-Tree Index is divided into:

- ***Normal Index.*** It is the most common type created if a user does not specify any additional parameters. In particular, Oracle creates it automatically for the primary key column whenever you create a new table with the primary key.

  Note: Oracle won't create an index for the columns with foreign keys.
- ***Function-Based Index.*** It is an index that calculates the result of a function involving one or more table columns (an arithmetic expression, an SQL function, a PL/SQL function, or a package function). The results are stored in the index. It is convenient when you use queries with expressions multiple times. The database must calculate that expression each time, but a Function-Based Index with the same expression lets you avoid those computations.

Bitmap Index is an index type used in scenarios with repetitive values. For instance, a traditional B-Tree index would be too expensive for data warehouses, but Bitmap indexes will save space. In addition, Bitmap indexes work best with complicated queries containing WHERE clauses with multiple conditions, reducing the response type.

Besides, Oracle differentiates unique and non-unique indexes.

- ***Unique Index*****.** Key column(s) can't have duplicate values. The simplest example is the staff database for any organization – two employees can't have the same ID. Then, the row ID is specific for each data value in this index.
- ***Non-unique Index.*** Indexed column(s) can have duplicate values. For instance, several employees can have the same first names. Thus, the respective column may contain duplicates. The row ID will be in the key in sorted order. Non-unique indexes are sorted by the index key and the row ID.

Both the unique and non-unique indexes are the B-Tree index structure versions. By default, the B-Tree index is non-unique. To create unique index in Oracle, you need to use the UNIQUE keyword in the CREATE INDEX statement.

SQL CREATE INDEX Statement

The `CREATE INDEX` statement is used to create indexes in tables.

Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

CREATE INDEX Syntax

Creates an index on a table. Duplicate values are allowed:

```
CREATE INDEX index_name
ON table_name (column1, column2, ...);
```

CREATE UNIQUE INDEX Syntax

Creates a unique index on a table. Duplicate values are not allowed:

```
CREATE UNIQUE INDEX index_name
ON table_name (column1, column2, ...);
```

CREATE INDEX Example

The SQL statement below creates an index named "idx_lastname" on the "LastName" column in the "Persons" table:

```
CREATE INDEX idx_lastname
ON Persons (LastName);
```

If you want to create an index on a combination of columns, you can list the column names within the parentheses, separated by commas:

```
CREATE INDEX idx_pname
ON Persons (LastName, FirstName);
```

DROP INDEX Statement

The `DROP INDEX` statement is used to delete an index in a table.

```
DROP INDEX index_name;
```