Created by Ahsan Arif

Indexes

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.

Create an Index

The syntax for creating a index is:

```
CREATE [UNIQUE] INDEX index_name
  ON table_name (column1, column2, . column_n)
  [ COMPUTE STATISTICS ];
```

UNIQUE indicates that the combination of values in the indexed columns must be unique.

COMPUTE STATISTICS tells Oracle to collect statistics during the creation of the index. The statistics are then used by the optimizer to choose a "plan of execution" when SQL statements are executed.

For example:

```
CREATE INDEX supplier_idx
   ON supplier (supplier_name);
```

In this example, we've created an index on the supplier table called supplier_idx. It consists of only one field - the supplier_name field.

We could also create an index with more than one field as in the example below:

```
CREATE INDEX supplier_idx
   ON supplier (supplier_name, city);
```

We could also choose to collect statistics upon creation of the index as follows:

```
CREATE INDEX supplier_idx
  ON supplier (supplier_name, city)
  COMPUTE STATISTICS;
```

Create a Function-Based Index

In Oracle, you are not restricted to creating indexes on only columns. You can create function-based indexes.

The syntax for creating a function-based index is:

```
CREATE [UNIQUE] INDEX index_name
  ON table_name (function1, function2, . function_n)
  [ COMPUTE STATISTICS ];
```

For example:

```
CREATE INDEX supplier_idx
   ON supplier (UPPER(supplier_name));
```

In this example, we've created an index based on the uppercase evaluation of the *supplier_name* field.

However, to be sure that the Oracle optimizer uses this index when executing your SQL statements, be sure that UPPER(supplier_name) does not evaluate to a NULL value. To ensure this, add **UPPER(supplier_name) IS NOT NULL** to your WHERE clause as follows:

```
SELECT supplier_id, supplier_name, UPPER(supplier_name)
FROM supplier
WHERE UPPER(supplier_name) IS NOT NULL
ORDER BY UPPER(supplier_name);
```

Rename an Index

The syntax for renaming an index is:

```
ALTER INDEX index_name
  RENAME TO new_index_name;
```

For example:

```
ALTER INDEX supplier_idx
  RENAME TO supplier_index_name;
```

In this example, we're renaming the index called *supplier_idx* to *supplier_index_name.*

Collect Statistics on an Index

If you forgot to collect statistics on the index when you first created it or you want to update the statistics, you can always use the ALTER INDEX command to collect statistics at a later date.

The syntax for collecting statistics on an index is:

```
ALTER INDEX index_name
  REBUILD COMPUTE STATISTICS;
```

For example:

```
ALTER INDEX supplier_idx
  REBUILD COMPUTE STATISTICS;
```

In this example, we're collecting statistics for the index called supplier_idx.

Drop an Index

The syntax for dropping an index is:

```
DROP INDEX index_name;
```

For example:

```
DROP INDEX supplier_idx;
```

In this example, we're dropping an index called supplier_idx.