Experiment No. 8

Environment: Microsoft Windows
Tools/ Language: Oracle

Objective:- To create

Introduction – An **index** is a schema object that contains an entry for each value that appears in the indexed column(s) of the table or cluster and provides direct, fast access to rows. Oracle Database supports several types of index:

- **Normal indexes**. (By default, Oracle Database creates B-tree indexes.)
- **Bitmap indexes**, which store rowids associated with a key value as a bitmap
- **Partitioned indexes**, which consist of partitions containing an entry for each value that appears in the indexed column(s) of the table
- **Function-based indexes**, which are based on expressions. They enable you to construct queries that evaluate the value returned by an expression, which in turn may include built-in or user-defined functions.
- **Domain indexes**, which are instances of an application-specific index of type ***indextype***

Syntax :-

Create Simple Index:

```
CREATE INDEX index_name
        ON table_name(column_name);
```

Create Composite Index:

```
CREATE INDEX index_name
        ON table_name(column_name1,column_name2,...);
```

Create Unique Index: it will be created on column having unique values

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

Create Bitmap Index: *useful for columns in which number of distinct values are small as compared to the number of rows in the table such as GENDER, MARITAL_STATUS for example:*

- *GPA has few distinct values as compared to 10,000 student as GPA lies between ranges of 1 to 10*
- *We maintain data of all the students in B.Tech course than among those 2000 students every student has age between 16 to 24* (*most probably*) *then there are only 9 distinct value of column* ***Age*** *for* ***2000 students rows.***

```
CREATE BITMAP INDEX index_name
        ON table_name(column_name);
```

Drop Index:

```
DROP INDEX index_name;
```

Sequences

In Oracle, you can create an autonumber field by using sequences. A sequence is an object in Oracle that is used to generate a number sequence. This can be useful when you need to create a unique number to act as a primary key.

The syntax to create a sequence in Oracle is:

```
CREATE SEQUENCE sequence_name
  INCREMENT BY integer
  START WITH integer
  MAXVALUE integer | NOMAXVALUE
  MINVALUE integer | NOMINVALUE
  CYCLE | NOCYCLE
  CACHE integer | NOCAHE
  ORDER | NOORDER;
```

For example:

```
CREATE SEQUENCE RollNo_seq
  MINVALUE 1
  MAXVALUE 999999999999999999999999999
  START WITH 1
  INCREMENT BY 1
  CACHE 20;
```

This would create a sequence object called RollNo_seq. The first sequence number that it would use is 1 and each subsequent number would increment by 1 (ie: 2,3,4,...}. It will cache up to 20 values for performance.

Now that you've created a sequence object to simulate an autonumber field, we'll cover how to retrieve a value from this sequence object. To retrieve the next value in the sequence order, you need to use nextval.

For example:

```
RollNo_seq.NEXTVAL;
```

This would retrieve the next value from supplier_seq. The nextval statement needs to be used in a SQL statement. For example:

```
INSERT INTO ClassDetails
(RollNo, Course_name)
VALUES
(RollNo_seq.NEXTVAL, 'Introduction to DBMS');
```

This insert statement would insert a new record into the ClassDetails table. The RollNo field would be assigned the next number from the RollNo_seq sequence. The Course_name field would be set to 'Introduction to DBMS'.

Sequence: Specify the name of the sequence to be created.

If you specify none of the following clauses, then you create an ascending sequence that starts with 1 and increases by 1 with no upper limit. Specifying only `INCREMENT BY -1` creates a descending sequence that starts with -1 and decreases with no lower limit.

- To create a sequence that increments without bound, for ascending sequences, omit the `MAXVALUE` parameter or specify `NOMAXVALUE`. For descending sequences, omit the `MINVALUE` parameter or specify the`NOMINVALUE`.
- To create a sequence that stops at a predefined limit, for an ascending sequence, specify a value for the `MAXVALUE` parameter. For a descending sequence, specify a value for the `MINVALUE` parameter. Also specify `NOCYCLE`. Any attempt to

generate a sequence number once the sequence has reached its limit results in an error.

- To create a sequence that restarts after reaching a predefined limit, specify values for both the MAXVALUE and MINVALUE parameters. Also specify CYCLE. If you do not specify MINVALUE, then it defaults to NOMINVALUE, which is the value 1.

INCREMENT BY

Specify the interval between sequence numbers. This integer value can be any positive or negative integer, but it cannot be 0. This value can have 28 or fewer digits. The absolute of this value must be less than the difference of MAXVALUE and MINVALUE. If this value is negative, then the sequence descends. If the value is positive, then the sequence ascends. If you omit this clause, then the interval defaults to 1.

START WITH

Specify the first sequence number to be generated. Use this clause to start an ascending sequence at a value greater than its minimum or to start a descending sequence at a value less than its maximum. For ascending sequences, the default value is the minimum value of the sequence. For descending sequences, the default value is the maximum value of the sequence. This integer value can have 28 or fewer digits.

MAXVALUE

Specify the maximum value the sequence can generate. This integer value can have 28 or fewer digits. MAXVALUE must be equal to or greater than START WITH and must be greater than MINVALUE.

NOMAXVALUE

Specify NOMAXVALUE to indicate a maximum value of 10^{27} for an ascending sequence or -1 for a descending sequence. This is the default.

MINVALUE

Specify the minimum value of the sequence. This integer value can have 28 or fewer digits. MINVALUE must be less than or equal to START WITH and must be less than MAXVALUE.

NOMINVALUE

Specify NOMINVALUE to indicate a minimum value of 1 for an ascending sequence or -10^{26} for a descending sequence. This is the default.

CYCLE

Specify CYCLE to indicate that the sequence continues to generate values after reaching either its maximum or minimum value. After an ascending sequence reaches its maximum value, it generates its minimum value. After a descending sequence reaches its minimum, it generates its maximum value.

NOCYCLE

Specify NOCYCLE to indicate that the sequence cannot generate more values after reaching its maximum or minimum value. This is the default.

CACHE

Specify how many values of the sequence the database preallocates and keeps in memory for faster access. This integer value can have 28 or fewer digits. The minimum value for this parameter is 2. For sequences that cycle, this value must be less than the number of values in the cycle. You cannot cache more values than will fit in a given cycle of sequence numbers. Therefore, the maximum value allowed for `CACHE` must be less than the value determined by the following formula:

```
(CEIL (MAXVALUE - MINVALUE)) / ABS (INCREMENT)
```

If a system failure occurs, all cached sequence values that have not been used in committed DML statements are lost. The potential number of lost values is equal to the value of the `CACHE` parameter.

Note: Oracle recommends using the `CACHE` setting to enhance performance if you are using sequences in a Real Application Clusters environment.

NOCACHE

Specify `NOCACHE` to indicate that values of the sequence are not preallocated. If you omit both `CACHE` and `NOCACHE`, the database caches 20 sequence numbers by default.

ORDER

Specify `ORDER` to guarantee that sequence numbers are generated in order of request. This clause is useful if you are using the sequence numbers as timestamps. Guaranteeing order is usually not important for sequences used to generate primary keys.

`ORDER` is necessary only to guarantee ordered generation if you are using Oracle Database with Real Application Clusters. If you are using exclusive mode, sequence numbers are always generated in order.

NOORDER

Specify `NOORDER` if you do not want to guarantee sequence numbers are generated in order of request. This is the default.

DROP SEQUENCE

Once you have created your sequence in Oracle, you might find that you need to remove it from the database.

Syntax

The syntax to a drop a sequence in Oracle is:

```
DROP SEQUENCE sequence_name;
```

sequence_name

The name of the sequence that you wish to drop.

Example

Let's look at an example of how to drop a sequence in Oracle.

For example:

```
DROP SEQUENCE RollNo_seq;
```

This example would drop the sequence called *RollNo_seq*.

Practical Assignment - 8

Department: Computer Engineering & Applications

Course: B.Tech. (CSE)

Subject: Database Management System Lab (CSE3083)

Year: 2nd **Semester:** 3rd

SQL Script for this Experiment

Student

sID	sName	GPA	sizeHS	DoB
123	Amy	3.9	1000	26-JUN-96
234	Bob	3.6	1500	7-Apr-95
345	Craig	3.5	500	4-Feb-95
456	Doris	3.9	1000	24-Jul-97
567	Edward	2.9	2000	21-Dec-96
678	Fay	3.8	200	27-Aug-96
789	Gary	3.4	800	8-Oct-96
987	Helen	3.7	800	27-Mar-97
876	Irene	3.9	400	7-Mar-96
765	Jay	2.9	1500	8-Aug-98
654	Amy	3.9	1000	26-May-96
543	Craig	3.4	2000	27-Aug-98

College

cName	state	enrollment
Stanford	CA	15000
Berkeley	CA	36000
MIT	MA	10000
Cornell	NY	21000
Harvard	MA	50040

Apply

sID	cName	major	decision
123	Stanford	CS	Y
123	Stanford	EE	N
123	Berkeley	CS	Y
123	Cornell	EE	Y
234	Berkeley	biology	N
345	MIT	bioengineering	Y
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
678	Stanford	history	Y
987	Stanford	CS	Y
987	Berkeley	CS	Y
876	Stanford	CS	N
876	MIT	biology	Y
876	MIT	marine biology	N
765	Stanford	history	Y
765	Cornell	history	N
765	Cornell	psychology	Y
543	MIT	CS	N

Write SQL queries for the following:

Q1. As we need to notify in system birthday of each student so kindly create an index DoBindex on column DoB of Student table.

Q2. Which index be more suitable for major in Apply? Create Bitmap Index name MAJORindex.

Q3. Remove index on DoB column.

Q4. Create an Unique index ENRindex on enrollment.

Q5. Create an composite Unique index SCMindex on Apply using columns sID, cName, major.

Q6. Create composite index on Apply using columns cName and major. Name this index as CMapplyINDX.

Q7. Create sequence sID_seq with following parameters, increment by 3, cycle, cache 4 and which will generate the numbers among 988 to 999.
Q8. Display next value of Sequence *sID_seq.*
Q9. A new student entered the database named Eric with next sID from sequence *sID_seq* having GPA 9.9 , sizeHS 9999, DoB as '23-Apr-98' to table Student.
Q10. Now, another boy registered to our system named Troy with next sID from sequence *sID_seq* having GPA 9.8 and sizeHS 989 and Dob as '25-Nov-99'.
Q11. Display details of Student table and observe newly inserted Eric and Troy sID.
Q12. Create sequence *GPA_seq* having maximum value as 5 and min value as 3 you are supposed to start sequence with 5 and decrement the sequence with -1, cycle and no cache.
Q13. Update GPA of Eric to next value of sequence GPA_seq.
Q14. Insert student Jack with sId from sID_seq , GPA from GPA_seq, sizeHS as 1500 and DoB as '22- OCT-97'.
Q15. Display detail of Student Table and observe GPA and sID of Jack.
Q16. Display next value of sequence GPA_seq.
Q17. Drop sequence sID_seq.
Q18. Drop sequence GPA_seq.

Pre-Experiment Questions:

1. What are indexes?
2. How they improve performance of retrieval in DBMS?
3. What is surrogate key?

Post Experiment Questions:

1. Can we create index on view?
2. Difference between Bitmap index and B Tree Indexes?
3. Why sequence are important?
4. How they help in implementing surrogate keys?