Ex.No.: 15		OTHER DATABASE OBJECTS
Date:		

OTHER DATABASE OBJECTS

Objectives

After the completion of this exercise, the students will be able to do the following:
- Create, maintain, and use sequences
- Create and maintain indexes

Database Objects

Many applications require the use of unique numbers as primary key values. You can either build code into the application to handle this requirement or use a sequence to generate unique numbers.
If you want to improve the performance of some queries, you should consider creating an index. You
can also use indexes to enforce uniqueness on a column or a collection of columns.
You can provide alternative names for objects by using synonyms.

What Is a Sequence?

A sequence:
- Automatically generates unique numbers
- Is a sharable object
- Is typically used to create a primary key value
- Replaces application code
- Speeds up the efficiency of accessing sequence values when cached in memory

The CREATE SEQUENCE Statement Syntax

Define a sequence to generate sequential numbers automatically:

```
CREATE SEQUENCE sequence
[INCREMENT BY n]
[START WITH n]
[{MAXVALUE n | NOMAXVALUE}]
[{MINVALUE n | NOMINVALUE}]
[{CYCLE | NOCYCLE}]
[{CACHE n | NOCACHE}];
```

In the syntax:

sequence is the name of the sequence generator

1) Create Sequence DEPT-ID-SEQ

```
Increment By 10
START WITH 200
MAX VALUE 1000
NO CACHE
NOCYCLE;
```

2)
```
SELECT
    sequence_name,
    nextval,
    increament_by,
    last_number.
FROM
    user_sequences
WHERE
    sequence_name='DEPT_ID_SEQ';
```

3)
```
INSERT INTO DEPT(DEPT_ID, DEPT_NAME)
VALUES(Dept_ID, SEQ.NEXTVAL, 'Education');
```

~~Insert into DEPT (DEPT_ID, DEPT_NAME)~~
~~VALUES(Dept-Id_SEQ_NEXT VAL, 'Healthcare');~~

4)
```
CREATE INDEX emp_Dept_id_idx
ON EMP(Dept_id);
```

~~5)~~

5) SELECT
    indesc_name,
    uniqueness
FROM
    user_indexes
WHERE
    table_name = 'EMP';