# CHAPTER 5

1. **Sequence**

* Syntax :

**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

INCREMENT BY *n* specifies the interval between sequence numbers, where *n* is an integer (If this clause is omitted, the sequence increments by 1.)

START WITH *n* specifies the first sequence number to be generated (If this clause is omitted, the sequence starts with 1.)

MAXVALUE *n* specifies the maximum value the sequence can generate

NOMAXVALUE specifies a maximum value of 10^27 for an ascending

sequence and –1 for a descending sequence (This is the default option.)

MINVALUE *n* specifies the minimum sequence value

NOMINVALUE specifies a minimum value of 1 for an ascending sequence and –(10^26) for a descending sequence (This is the default option.)

CYCLE | NOCYCLE specifies whether the sequence continues to generate values after reaching its maximum or minimum value

(NOCYCLE is the default option.)

CACHE n | NOCACHE specifies how many values the Oracle server preallocates and keeps in memory (By default, the Oracle server caches 20 values.)

* Example:

**CREATE** **SEQUENCE** dept\_deptid\_seq

**INCREMENT** **BY** 10

START **WITH** 120

**MAXVALUE** 9999

**NOCACHE**

**NOCYCLE**;

**ALTER** **SEQUENCE** dept\_deptid\_seq

**INCREMENT** **BY** 20

**MAXVALUE** 999999

**NOCACHE**

**NOCYCLE**;

**DROP** **SEQUENCE** dept\_deptid\_seq;

* Using:

NEXTVAL returns the next available sequence value. It returns a unique value every time it is referenced, even for different users.

CURRVAL obtains the current sequence value.

Example:

**INSERT** **INTO** departments(department\_id,department\_name, location\_id)

**VALUES** (dept\_deptid\_seq.**NEXTVAL**,'Support', 2500);

**SELECT** dept\_deptid\_seq.**CURRVAL** **FROM** dual;

* Note:

Rules for Using NEXTVAL and CURRVAL

\* You can use NEXTVAL and CURRVAL in the following contexts:

• The SELECT list of a SELECT statement that is not part of a subquery

• The SELECT list of a subquery in an INSERT statement

• The VALUES clause of an INSERT statement

• The SET clause of an UPDATE statement

\* You cannot use NEXTVAL and CURRVAL in the following contexts:

• The SELECT list of a view

• A SELECT statement with the DISTINCT keyword

• A SELECT statement with GROUP BY, HAVING, or ORDER BY clauses

• A subquery in a SELECT, DELETE, or UPDATE statement

• The DEFAULT expression in a CREATE TABLE or ALTER TABLE statement

1. **Index**

* Introduction

Is a schema object

• Can be used by the Oracle server to speed up the

retrieval of rows by using a pointer

• Can reduce disk I/O by using a rapid path access

method to locate data quickly

• Is independent of the table that it indexes

• Is used and maintained automatically by the Oracle server

* Types of Indexes

Automatically: A unique index is created automatically when you define a PRIMARY KEY or UNIQUE constraint in a table definition.

Manually: Users can create nonunique indexes on columns to speed up access to the rows.

* Creating an Index

Syntax :

**CREATE INDEX *index***

**ON *table* (*column*[, *column*]...);**

Example

**CREATE** **INDEX** emp\_last\_name\_idx

**ON** employees(last\_name);

* Remove an Index

Syntax:

**DROP INDEX *index*;**

Example:

**DROP** **INDEX** emp\_last\_name\_idx;

1. **Synonyms**

Synonyms are database objects that enable you to call a table by another name. You can create synonyms to give an alternate name to a table.

Simplify access to objects by creating a synonym

(another name for an object). With synonyms, you can:

• Create an easier reference to a table that is owned

by another user

• Shorten lengthy object names

Syntax:

**CREATE [PUBLIC] SYNONYM *synonym***

**FOR *object*;**

In the syntax:

PUBLIC creates a synonym that is accessible to all users

*synonym* is the name of the synonym to be created

*object* identifies the object for which the synonym is created

Example:

**CREATE** **SYNONYM** emp

**FOR** employees;

**select** \* **from** emp;

**DROP** **SYNONYM** emp;

1. **Practices:**
   1. You need a sequence that can be used with the primary key column of the departments table. The sequence should start at 200 and have a maximum value of 1,000. Have your sequence increment by 10
   2. To test your sequence, write a script to insert two rows in the departments table
   3. Create a nonunique index on the NAME column in the departments table.
   4. Create a synonym for your EMPLOYEES table. Call it EMP