14

Creating Other Schema Objects

Objectives

After completing this lesson, you should be able to do the following:

- Create, maintain, and use sequences
- Create and maintain indexes
- Create private and public synonyms

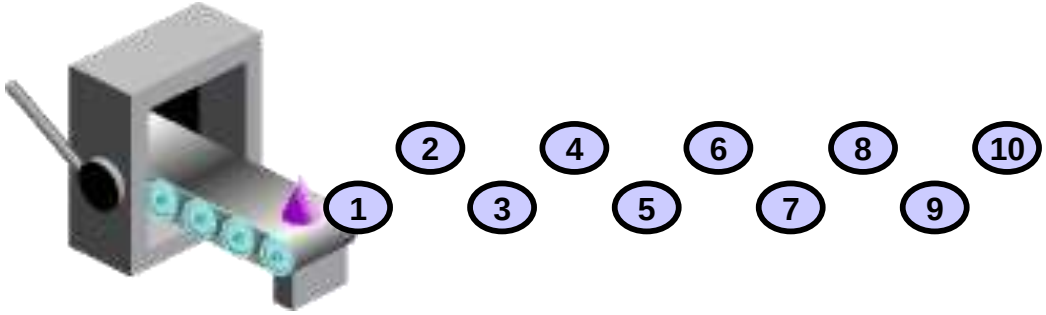
Database Objects

Object	Description
Sequence	Generates numeric values
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

Sequences

A sequence:

- **A sequence is a database object that generates a sequence of integers.**
- **Can automatically generate unique numbers**
- **Is a sharable object**
- **Can be used to create a primary key value**
- **Speeds up the efficiency of accessing sequence values when cached in memory**



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}];
```

Creating a Sequence

- Create a sequence named `DEPT_DEPTID_SEQ` to be used for the primary key of the `DEPARTMENTS` table.
- Do not use the `CYCLE` option.

```
CREATE SEQUENCE dept_deptid_seq
                INCREMENT BY 10
                START WITH 120
                MAXVALUE 9999
                NOCACHE
                NOCYCLE;
Sequence created.
```

NEXTVAL and CURRVAL Pseudocolumns

- 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.
- NEXTVAL must be issued for that sequence before CURRVAL contains a value.

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

Using a Sequence

- Insert a new department named “Support” in location ID 2500:

```
INSERT INTO departments(department_id,
            department_name, location_id)
VALUES      (dept_deptid_seq.NEXTVAL,
            'Support', 2500);
1 row created.
```

- View the current value for the DEPT_DEPTID_SEQ sequence:

```
SELECT   dept_deptid_seq.CURRVAL
FROM     dual;
```

Caching Sequence Values

- Caching sequence values in memory gives faster access to those values.
- Gaps in sequence values can occur when:
  - A rollback occurs
  - The system crashes
  - A sequence is used in another table

Modifying a Sequence

Change the increment value, maximum value, minimum value, cycle option, or cache option:

```
ALTER SEQUENCE dept_deptid_seq
               INCREMENT BY 20
               MAXVALUE 999999
               NOCACHE
               NOCYCLE;
Sequence altered.
```

Guidelines for Modifying a Sequence

- You must be the owner or have the `ALTER` privilege for the sequence.
- Only future sequence numbers are affected.
- The sequence must be dropped and re-created to restart the sequence at a different number.
- Some validation is performed.
- To remove a sequence, use the `DROP` statement:

```
DROP SEQUENCE dept_deptid_seq;
Sequence dropped.
```

Indexes

Object	Description
Table	Basic unit of storage; composed of rows
View	Logically represents subsets of data from one or more tables
Sequence	Generates numeric values
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

Indexes

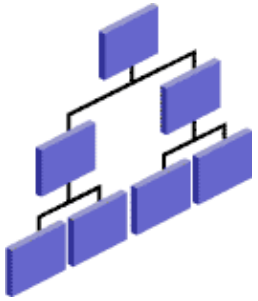
- When looking for a particular topic in a book, you can either scan the whole book looking for your topic, or you can use the book's index to find the exact location of the topic directly.

- An index for a database table is similar in concept to a book index, except that database indexes are used to find specific rows in a table.

Indexes

An index:

- **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**



How Are Indexes Created?

- **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

- **Create an index on one or more columns:**

```
CREATE INDEX index
ON table (column[, column]...);
```

- **Improve the speed of query access to the `LAST_NAME` column in the `EMPLOYEES` table:**

```
CREATE INDEX  emp_last_name_idx
ON            employees(last_name);
Index created.
```

Index Creation Guidelines

Create an index when:	
✓	A column contains a wide range of values
✓	A column contains a large number of null values
✓	One or more columns are frequently used together in a `WHERE` clause or a join condition
✓	The table is large and most queries are expected to retrieve less than 2% to 4% of the rows in the table
Do not create an index when:	
✗	The columns are not often used as a condition in the query
✗	The table is small or most queries are expected to retrieve more than 2% to 4% of the rows in the table
✗	The table is updated frequently
✗	The indexed columns are referenced as part of an expression

CREATE INDEX with CREATE TABLE Statement

```
CREATE TABLE NEW_EMP
(employee_id NUMBER(6)
              PRIMARY KEY USING INDEX
             (CREATE INDEX emp_id_idx ON
             NEW_EMP(employee_id)),
first_name   VARCHAR2(20),
last_name    VARCHAR2(25));
Table created.
```

```
SELECT INDEX_NAME, TABLE_NAME
FROM   USER_INDEXES
WHERE  TABLE_NAME = 'NEW_EMP';
```

INDEX_NAME	TABLE_NAME
EMP_ID_IDX	NEW_EMP

Types of Indexes

- B-tree indexes (Balanced Trees)
- Function-based indexes
- Bitmap and bitmap join indexes
- Application domain indexes

B-tree indexes (Balanced Trees)

- The most common type of database index.
- A B-tree index is an ordered list of values divided into ranges.
- By associating a key with a row or range of rows, B-trees provide excellent retrieval performance for a wide range of queries, including exact match and range searches.
- A B-tree index has two types of blocks: branch blocks for searching and leaf blocks that store values.
- The upper-level branch blocks of a B-tree index contain index data that points to lower-level index blocks.

Function-Based Indexes

- A function-based index is based on expressions.
- The index expression is built from table columns, constants, SQL functions, and user-defined functions.

```
CREATE INDEX upper_dept_name_idx
ON dept2(UPPER(department_name));

Index created.

SELECT *
FROM   dept2
WHERE  UPPER(department_name) = 'SALES';
```

Removing an Index

- Remove an index from the data dictionary by using the `DROP INDEX` command:

```
DROP INDEX index;
```

- Remove the `UPPER_LAST_NAME_IDX` index from the data dictionary:

```
DROP INDEX emp_last_name_idx;
Index dropped.
```

- To drop an index, you must be the owner of the index or have the `DROP ANY INDEX` privilege.

Synonyms

Object	Description
Table	Basic unit of storage; composed of rows
View	Logically represents subsets of data from one or more tables
Sequence	Generates numeric values
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

Synonyms

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**

```
CREATE [PUBLIC] SYNONYM synonym
FOR    object;
```

Creating and Removing Synonyms

- Create a shortened name for the `DEPT_SUM_VU` view:

```
CREATE SYNONYM  d_sum
FOR  dept_sum_vu;
Synonym Created.
```

- Drop a synonym:

```
DROP SYNONYM d_sum;
Synonym dropped.
```

Summary

In this lesson, you should have learned how to:

- Create, use, and remove views
- Automatically generate sequence numbers by using a sequence generator
- Create indexes to improve query retrieval speed
- Use synonyms to provide alternative names for objects