DDL Statements

Objectives

- Describe the main database objects
- Create tables
- Describe the data types that can be used when specifying column definition
- Alter table definitions
- Drop, rename, and truncate tables

Database Objects

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

Naming Conventions

- Must begin with a letter
- Can be 1–30 characters long
- Must contain only A–Z, a–z, 0–9, _, $, and #
- Must not duplicate the name of another object owned by the same user
- Must not be an Oracle Server reserved word

CREATE TABLE Statement

- You must have :
  - CREATE TABLE privilege
  - A storage area

```
CREATE TABLE [schema.]table
             (column data type [DEFAULT expr];
```

- You specify:
  - Table name
  - Column name, column data type, and column size

Reference other User’s Tables

- Tables belonging to other users are not in the user’s schema.
- You should use the owner’s name as a prefix to the table.

The DEFAULT Option

- Specify a default value for a column during an insert.

```
… hiredate DATE DEFAULT SYSDATE, …
```

- Legal values are literal value, expression, or SQL function.
- Illegal values are another column’s name or pseudo column.
- The default data type must match the column data type.

Creating Tables

- **Create the table**

```
SQL> CREATE TABLE department
  2          (deptno  NUMBER(2),
  3           dname   VARCHAR2(14),
  4           loc     VARCHAR2(13));
Table created.
```

- **Confirm table creation**

```
SQL> DESCRIBE department
```

```
Name                           Null?    Type
------------------------------ -------- ---------
DEPTNO                         NOT NULL NUMBER(2)
DNAME                                   VARCHAR2(14)
LOC                                     VARCHAR2(13)
```

Querying the Data Dictionary

- Describe tables owned by the user.

```
SQL> SELECT   *
  2  FROM     user_tables;
```

- View distinct object types owned by the user.
  - Enclose character and date values within single quotation marks.

```
SQL> SELECT   DISTINCT object_type
  2  FROM     user_objects;
```

- View tables, views, synonyms, and sequences owned by the user.

```
SQL> SELECT   *
  2  FROM     user_catalog;
```

Create Table Using Subquery

- Create a table and insert rows by combining the CREATE TABLE statement and AS *subquery* option.

```
CREATE TABLE table
       [column(, column...)]
AS subquery;
```

- Match the number of specified columns to the number of subquery columns.
- Define columns with column names and default values.

Create Table Using Subquery

```
SQL> CREATE TABLE   dept30
  2  AS
  3     SELECT    empno, ename, sal*12 ANNSAL, hiredate
  4     FROM      employee
  5     WHERE     deptno = 30;
Table created.
```

```
 SQL> DESCRIBE dept30
```

```
 Name                            Null?    Type
 ------------------------------- -------- -----
 EMPNO                           NOT NULL NUMBER(4)
 ENAME                                    VARCHAR2(10)
 ANNSAL                                   NUMBER
 HIREDATE                                 DATE
```

ALTER TABLE Statement

- Add a new column
- Modify an existing column
- Drop an existing column,
- Define a default value for the new column

```
ALTER TABLE table
ADD         (column data type [DEFAULT expr]
            [, column data type]...);
```

```
ALTER TABLE table
MODIFY      (column data type [DEFAULT expr]
            [, column data type]...);
```

```
ALTER TABLE table
DROP column column_name;
```

Adding a Column

DEPT30

EMPNO	ENAME	ANNSAL	HIREDATE
7698	BLAKE	34200	01-MAY-81
7654	MARTIN	15000	28-SEP-81
7499	ALLEN	19200	20-FEB-81
7844	TURNER	18000	08-SEP-81
. . .			

New column

JOB

"...add a new column into DEPT30 table..."

DEPT30

EMPNO	ENAME	ANNSAL	HIREDATE	JOB
7698	BLAKE	34200	01-MAY-81	
7654	MARTIN	15000	28-SEP-81	
7499	ALLEN	19200	20-FEB-81	
7844	TURNER	18000	08-SEP-81	
. . .				

Adding a Column

- You use the ADD clause to add columns.

```
SQL> ALTER TABLE dept30
  2  ADD         (job VARCHAR2(9));
Table altered.
```

- The new column becomes the last column.

```
    EMPNO ENAME             ANNSAL HIREDATE  JOB
--------- ---------- --------- --------- ----
     7698 BLAKE              34200 01-MAY-81
     7654 MARTIN             15000 28-SEP-81
     7499 ALLEN              19200 20-FEB-81
     7844 TURNER             18000 08-SEP-81
...
6 rows selected.
```

Modifying a Column

- You can change a column's data type, size, and default value.

```
ALTER TABLE   dept30
MODIFY        (ename VARCHAR2(15));
Table altered.
```

- A change to the default value affects only subsequent insertions to the table.

Dropping a Column

- You can remove a column and its contents entirely from the table.

```
ALTER TABLE   dept30
DROP COLUMN ename;
Table altered.
```

- You can ignore the column by set unused column

```
SQL>ALTER TABLE dept30 set unused column ename;
Table altered.
SQL> ALTER TABLE dept30 drop unused columns;
Table altered.
```

Dropping a Table

- All data and structure in the table is deleted.
- Any pending transactions are committed.
- All indexes are dropped.
- You *cannot* roll back this statement.

```
SQL> DROP TABLE dept30;
Table dropped.
```

Rename an Object

```
SQL> RENAME dept TO department;
Table renamed.
```

- To change the name of a table, view, sequence, or synonym, you execute the RENAME statement.
- You must be the owner of the object.

Summary

- Creating a Table.
- Naming conventions.
- Modifying the table structure.
- Removing a table from the DB.
- Renaming an object.