Creating and Managing Tables

Objectives

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

- 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	Numeric value generator
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

Naming Rules

Table names and column names:

- Must begin with a letter

- Must 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

The `CREATE TABLE` Statement

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

```
CREATE TABLE [schema.]table
              (column datatype [DEFAULT expr][, ...]);
```

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

Data Types

Data Type	Description
`VARCHAR2(size)`	Variable-length character data
`CHAR(size)`	Fixed-length character data
`NUMBER(p,s)`	Variable-length numeric data
`DATE`	Date and time values
`LONG`	Variable-length character data up to 2 gigabytes
`CLOB`	Character data up to 4 gigabytes
`RAW and LONG RAW`	Raw binary data
`BLOB`	Binary data up to 4 gigabytes
`BFILE`	Binary data stored in an external file; up to 4 gigabytes
`ROWID`	A 64 base number system representing the unique address of a row in its table.

The `DEFAULT` Option

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

```
... hire_date DATE DEFAULT CURRENT_TIMESTAMP, ...
```

- Literal values, expressions, or SQL functions are legal values.
- Another column's name or a pseudocolumn are illegal values.
- The default data type must match the column data type.

Creating Tables

- Create the table.

```
CREATE TABLE dept
          (deptno  NUMBER(2),
           dname   VARCHAR2(14),
           loc     VARCHAR2(13));
Table created.
```

Creating a Table
by Using a Subquery Syntax

- Create a table and insert rows by combining the `CREATE TABLE` statement and the `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.

Creating a Table by Using a Subquery

```
CREATE TABLE  dept80
  AS
    SELECT   employee_id, last_name,
             salary*12 ANNSAL,
             hire_date
    FROM     employees
    WHERE    department_id = 80;
Table created.
```

```
DESCRIBE dept80
```

Name	Null?	Type
EMPLOYEE_ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
ANNSAL		NUMBER
HIRE_DATE	NOT NULL	DATE

TECHNION
Azrieli Continuing Education and
External Studies Division

The ALTER TABLE Statement

Use the ALTER TABLE statement to:

- Add a new column

- Modify an existing column

- Define a default value for the new column

- Drop a column

The ALTER TABLE Statement

Use the ALTER TABLE statement to add or drop columns.

```
ALTER TABLE table_name
ADD COLUMN column_name data_type [constraints];
```

```
ALTER TABLE table
DROP COLUMN  (column);
```

The ALTER TABLE Statement

Use the ALTER TABLE statement to modify columns.*

```
ALTER TABLE table_name
ALTER COLUMN column_name SET DATA TYPE new_data_type;
```

```
ALTER TABLE table
ALTER COLUMN column_name SET DEFAULT new_default_value
```

```
ALTER TABLE table
ALTER COLUMN column_name DROP DEFAULT;
```

* In postgarsql

The `ALTER TABLE` Statement

Use the `ALTER  TABLE` statement to rename of a table or a column.*

```
ALTER TABLE table_name
RENAME COLUMN old_column_name TO new_column_name;
```

```
ALTER TABLE table
RENAME TO new_table_name;
```

* In postgarsql

Adding a Column

New column

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE
149	Zlotkey	126000	29-JAN-00
174	Abel	132000	11-MAY-96
176	Taylor	103200	24-MAR-98

JOB_ID

"Add a new column to the DEPT80 table."

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

Adding a Column

- You use the `ADD` clause to add columns.

```
ALTER TABLE dept80
ADD job_id VARCHAR(9);
Table altered.
```

- The new column becomes the last column.

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

Modifying a Column

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

```
ALTER TABLE    dept80
ALTER COLUMN  last_name SET DATA TYPE VARCHAR(30);
Table altered.
```

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

Dropping a Column

Use the `DROP COLUMN` clause to drop columns you no longer need from the table.

```
ALTER TABLE   dept80
DROP COLUMN   job_id;
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 the `DROP TABLE` statement.

```
DROP TABLE dept80;
Table dropped.
```

Truncating a Table

- The `TRUNCATE TABLE` statement:
  - Removes all rows from a table
  - Releases the storage space used by that table

```
TRUNCATE TABLE detail_dept;
Table truncated.
```

- You cannot roll back row removal when using `TRUNCATE`.
- Alternatively, you can remove rows by using the `DELETE` statement.

Summary

In this lesson, you should have learned how to use DDL statements to create, alter, drop, and rename tables.

Statement	Description
`CREATE TABLE`	Creates a table
`ALTER TABLE`	Modifies table structures
`DROP TABLE`	Removes the rows and table structure
`TRUNCATE`	Removes all rows from a table and releases the storage space

Practice 9 Overview

This practice covers the following topics:

- Creating new tables
- Creating a new table by using the `CREATE TABLE AS` syntax
- Modifying column definitions
- Verifying that the tables exist
- Adding comments to tables
- Dropping tables
- Altering tables