

10

Creating Other Schema Objects

Objectives

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

- **Create simple and complex views**
- **Retrieve data from views**
- **Create, maintain, and use sequences**
- **Create and maintain indexes**
- **Create private and public synonyms**

Database Objects

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

Simple Views and Complex Views

Feature	Simple Views	Complex Views
Number of tables	One	One or more
Contain functions	No	Yes
Contain groups of data	No	Yes
DML operations through a view	Yes	Not always

Creating a View

- You embed a subquery in the CREATE VIEW statement:

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view
  [(alias[, alias]...)]
  AS subquery
  [WITH CHECK OPTION [CONSTRAINT constraint]]
  [WITH READ ONLY [CONSTRAINT constraint]];
```

- The subquery can contain complex SELECT syntax.

Creating a View

- **Create the EMPVU80 view, which contains details of employees in department 80:**

```
CREATE VIEW empvu80
AS SELECT employee_id, last_name, salary
FROM employees
WHERE department_id = 80;
```

View created.

- **Describe the structure of the view by using the *iSQL*Plus* DESCRIBE command:**

```
DESCRIBE empvu80
```

Modifying a View

- **Modify the EMPVU80 view by using a CREATE OR REPLACE VIEW clause. Add an alias for each column name:**

```
CREATE OR REPLACE VIEW empvu80
  (id_number, name, sal, department_id)
AS SELECT  employee_id, first_name || ' '
           || last_name, salary, department_id
FROM      employees
WHERE     department_id = 80;
```

View created.

- **Column aliases in the CREATE OR REPLACE VIEW clause are listed in the same order as the columns in the subquery.**

Creating a Complex View

Create a complex view that contains group functions to display values from two tables:

```
CREATE OR REPLACE VIEW dept_sum_vu
  (name, minsal, maxsal, avgsal)
AS SELECT  d.department_name, MIN(e.salary),
           MAX(e.salary), AVG(e.salary)
FROM      employees e JOIN departments d
ON        (e.department_id = d.department_id)
GROUP BY d.department_name;
```

View created.

Removing a View

You can remove a view without losing data because a view is based on underlying tables in the database.

```
DROP VIEW view;
```

```
DROP VIEW empvu80;  
View dropped.
```