12

Creating Views

Objectives

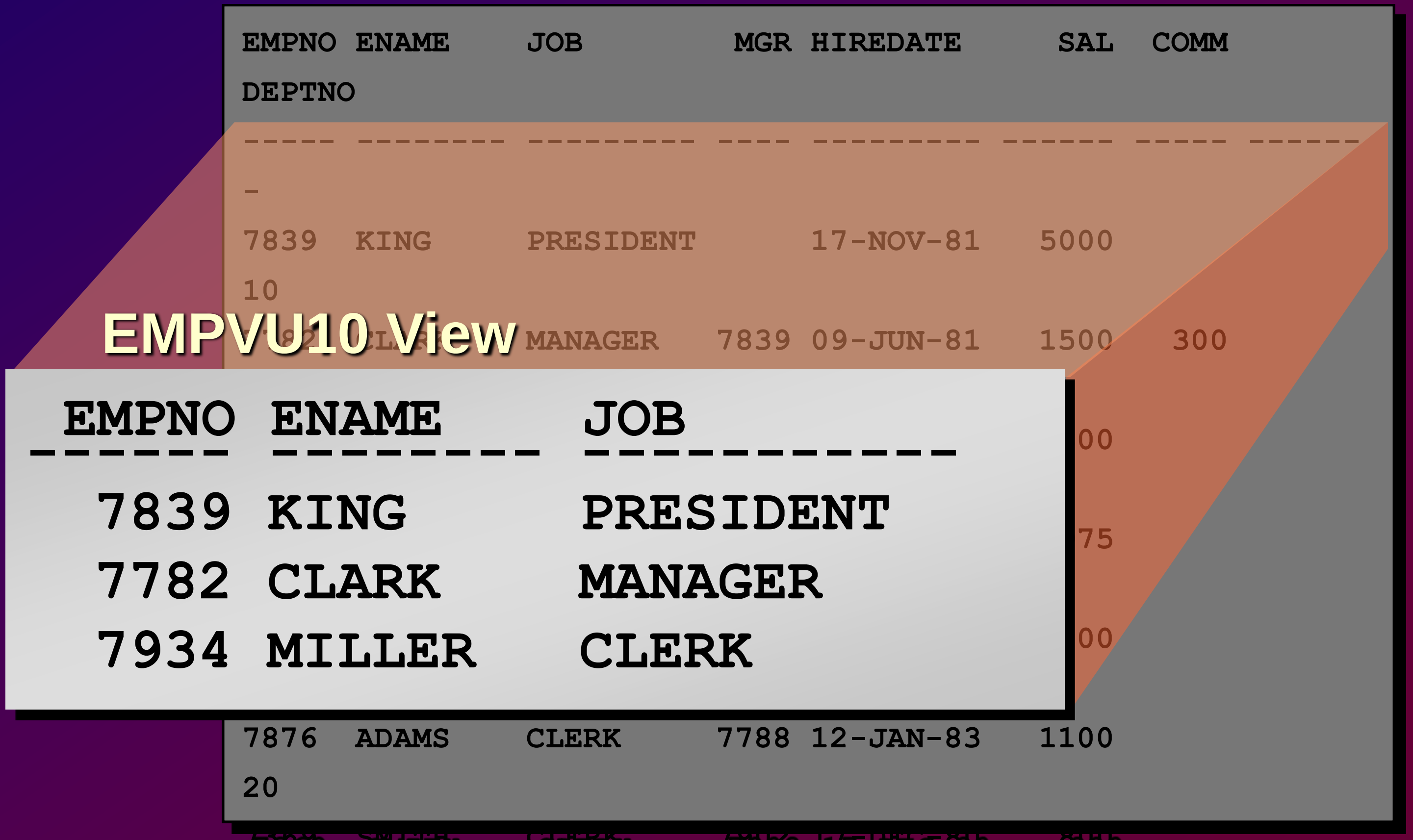
At the end of this lesson, you will be able to:

- Describe a view
- Create a view
- Retrieve data through a view
- Alter the definition of a view
- Insert, update, and delete data through a view
- Drop a view

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	Alternative name for an object

What Is a View?



EMP Table

```
EMPNO ENAME      JOB          MGR HIREDATE      SAL   COMM
DEPTNO
----- -------- --------- ---- --------- ------ ----- -------
-
7839  KING      PRESIDENT      17-NOV-81    5000
10
                MANAGER   7839 09-JUN-81    1500    300
                                              00
                                              75
                                              00
7876  ADAMS     CLERK     7788 12-JAN-83    1100
20
7369  SMITH     CLERK     7902 17-DEC-80     800
20
7902  FORD      ANALYST   7566 03-DEC-81    3000
20
7698  BLAKE     MANAGER   7839 01-MAY-81    2850
```

EMPVU10 View

```
 EMPNO ENAME      JOB
------- --------- -----------
   7839 KING       PRESIDENT
   7782 CLARK      MANAGER
   7934 MILLER     CLERK
```

Why Use Views?

- To restrict database access
- To make complex queries easy
- To allow data independence
- To present different views of the same data

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 through view	Yes	Not always

Creating a View

- You can embed a subquery within the CREATE VIEW statement.

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

- The subquery can contain complex SELECT syntax.
- The subquery cannot contain an ORDER BY clause.

Creating a View

- Create a view, EMPVU10, that contains details of employees in department 10.

```
SQL> CREATE VIEW     empvu10
  2  AS SELECT       empno, ename, job
  4  FROM            emp
  5  WHERE           deptno = 10;
View created.
```

- Describe the structure of the view by using the SQL*Plus DESCRIBE command.

```
SQL> DESCRIBE empvu10
```

Creating a View

- Create a view by using column aliases in the subquery.

```
SQL> CREATE VIEW     salvu30
  2  AS SELECT       empno EMPLOYEE_NUMBER, ename NAME,
  3                  sal SALARY
  4  FROM            emp
  5  WHERE           deptno = 30;
View created.
```

- Select the columns from this view by the given alias name.

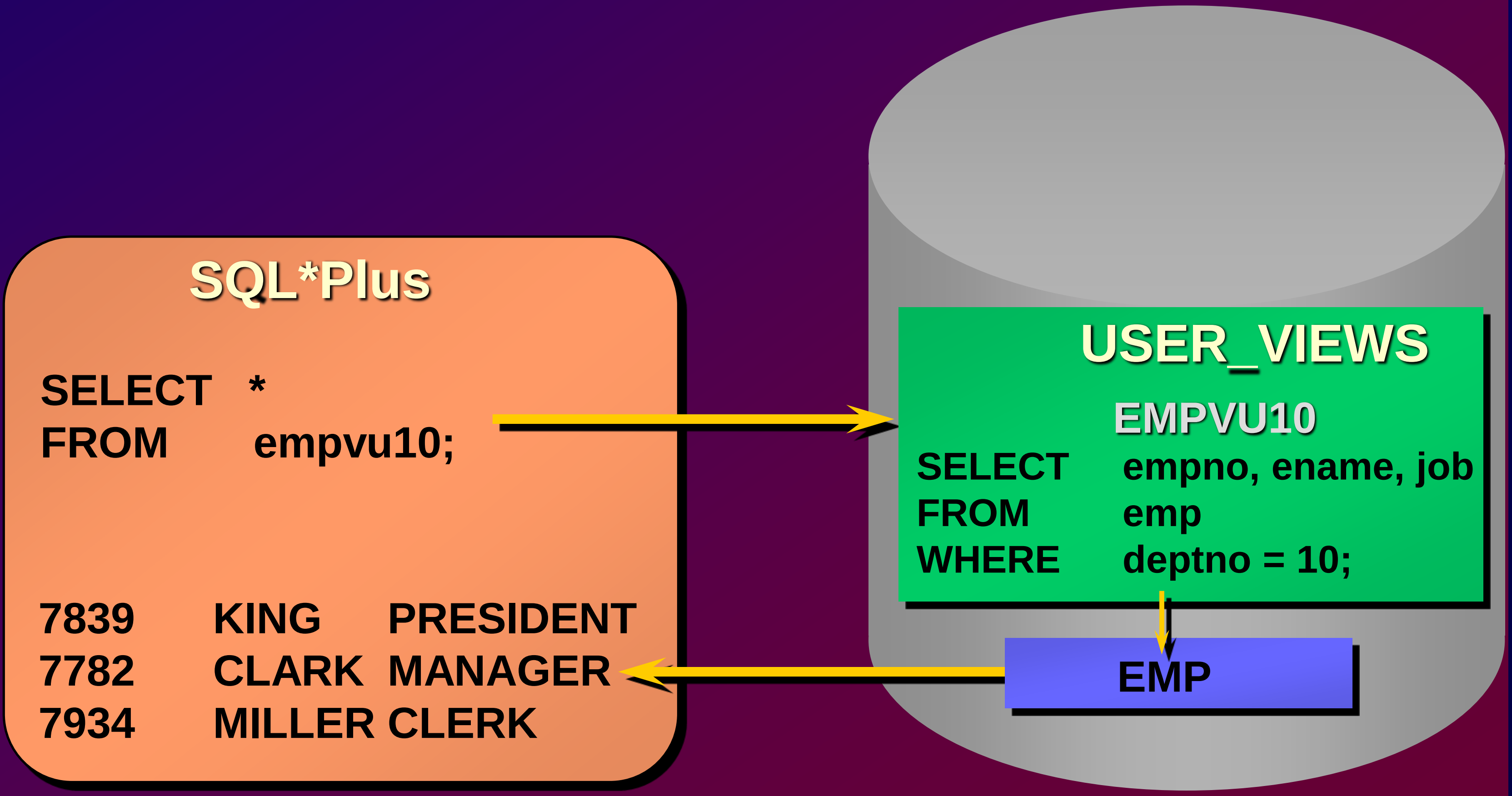
Retrieving Data from a View

```
SQL>  SELECT *
  2   FROM   salvu30;
```

```
EMPLOYEE_NUMBER NAME           SALARY
--------------- ---------- ----------
           7698 BLAKE            2850
           7654 MARTIN           1250
           7499 ALLEN            1600
           7844 TURNER           1500
           7900 JAMES             950
           7521 WARD             1250

6 rows selected.
```

Querying the USER_VIEWS Data Dictionary Table



SQL*Plus

```
SELECT   *
FROM      empvu10;
```

```
7839     KING     PRESIDENT
7782     CLARK  MANAGER
7934     MILLER CLERK
```

USER_VIEWS

EMPVU10

```
SELECT     empno, ename, job
FROM       emp
WHERE      deptno = 10;
```

Modifying a View

- Modify the EMPVU10 view by using CREATE OR REPLACE VIEW clause. Add an alias for each column name.

```
SQL> CREATE OR REPLACE VIEW empvu10
  2          (employee_number, employee_name, job_title)
  3   AS SELECT       empno, ename, job
  4   FROM            emp
  5   WHERE           deptno = 10;
View created.
```

- Column aliases in the CREATE 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.

```
SQL> CREATE VIEW    dept_sum_vu
  2               (name, minsal, maxsal, avgsal)
  3  AS SELECT    d.dname, MIN(e.sal), MAX(e.sal),
  4               AVG(e.sal)
  5  FROM         emp e, dept d
  6  WHERE        e.deptno = d.deptno
  7  GROUP BY     d.dname;
View created.
```

Rules for Performing DML Operations on a View

- You can perform DML operations on simple views.
- You cannot remove a row if the view contains the following:
  - Group functions
  - A GROUP BY clause
  - The DISTINCT keyword

Rules for Performing DML Operations on a View

- You cannot modify data in a view if it contains:
  - Any of the conditions mentioned in the previous slide
  - Columns defined by expressions
  - The ROWNUM pseudocolumn
- You cannot add data if:
  - The view contains any of the conditions mentioned in the previous slide
  - There are NOT NULL columns in the base tables that are not selected by the view

Using the WITH CHECK OPTION Clause

- You can ensure that DML on the view stays within the domain of the view by using the WITH CHECK OPTION.

```
SQL> CREATE OR REPLACE VIEW empvu20
  2  AS SELECT      *
  3  FROM   emp
  4  WHERE  deptno = 20
  5  WITH CHECK OPTION CONSTRAINT empvu20_ck;
View created.
```

- Any attempt to change the department number for any row in the view will fail because it violates the WITH CHECK OPTION constraint.

Denying DML Operations

- You can ensure that no DML operations occur by adding the WITH READ ONLY option to your view definition.

```
SQL> CREATE OR REPLACE VIEW empvu10
  2         (employee_number, employee_name, job_title)
  3   AS SELECT      empno, ename, job
  4   FROM           emp
  5   WHERE          deptno = 10
  6   WITH READ ONLY;
View created.
```

- Any attempt to perform a DML on any row in the view will result in Oracle8 Server error ORA-01732.

Removing a View

- Remove a view without losing data because a view is based on underlying tables in the database.

```
DROP VIEW view;
```

```
SQL> DROP VIEW empvu10;
View dropped.
```

Summary

- A view is derived from data in other tables or other views.
- A view provides the following advantages:
  - Restricts database access
  - Simplifies queries
  - Provides data independence
  - Allows multiple views of the same data
  - Can be dropped without removing the underlying data

Practice Overview

- Creating a simple view
- Creating a complex view
- Creating a view with a check constraint
- Attempting to modify data in the view
- Displaying view definitions
- Removing views