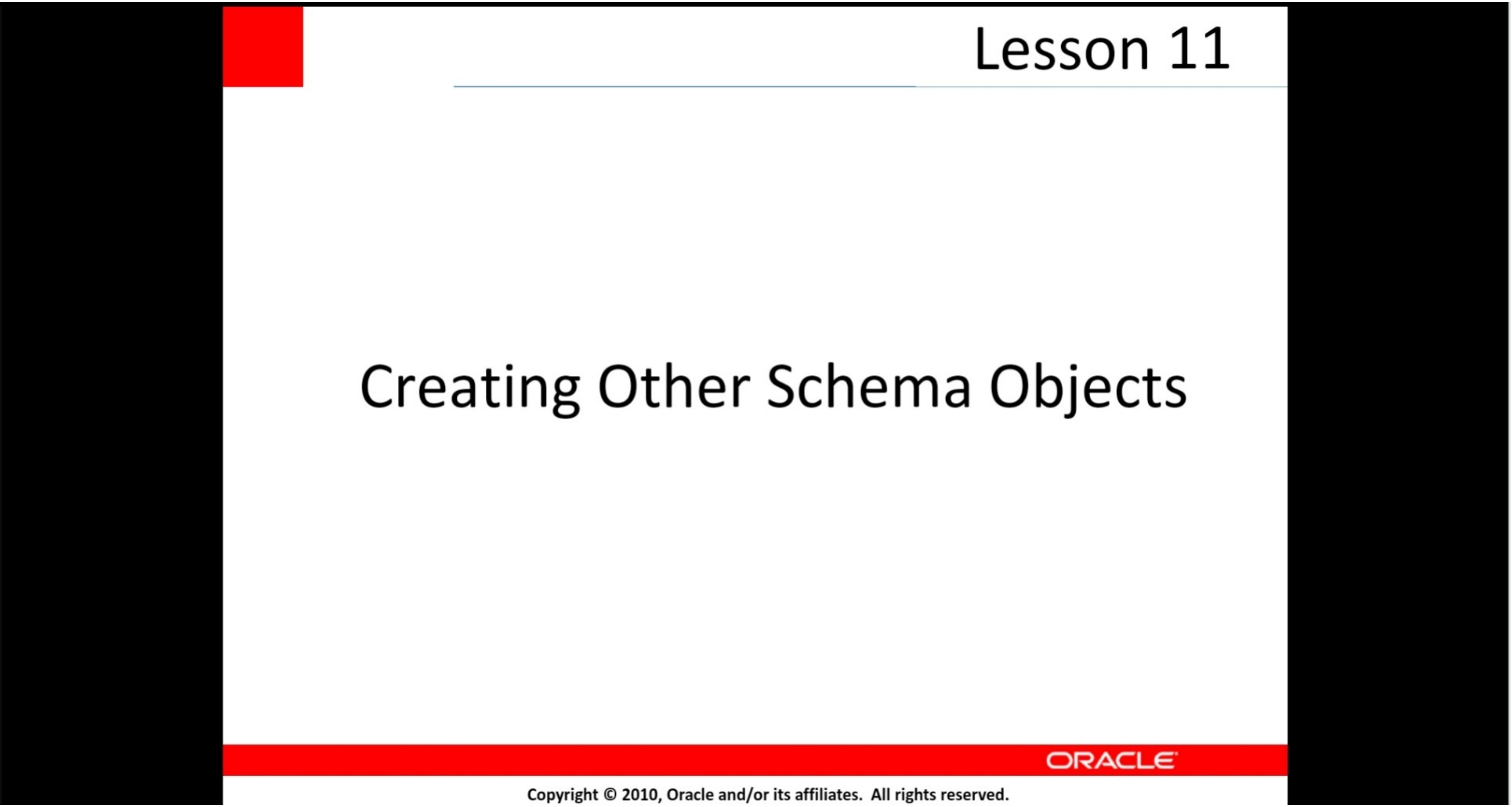
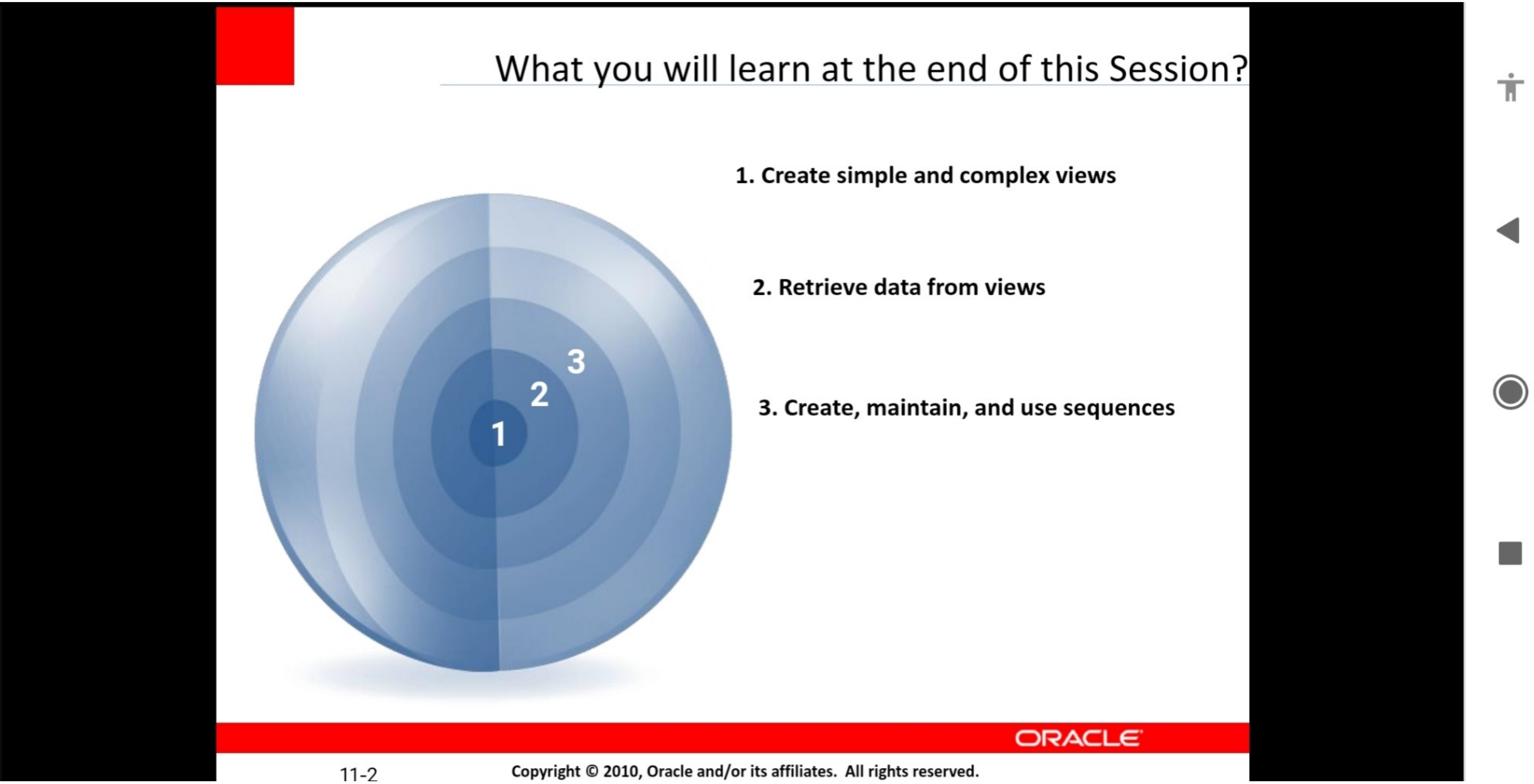


Lesson 11

Creating Other Schema Objects

ORACLE

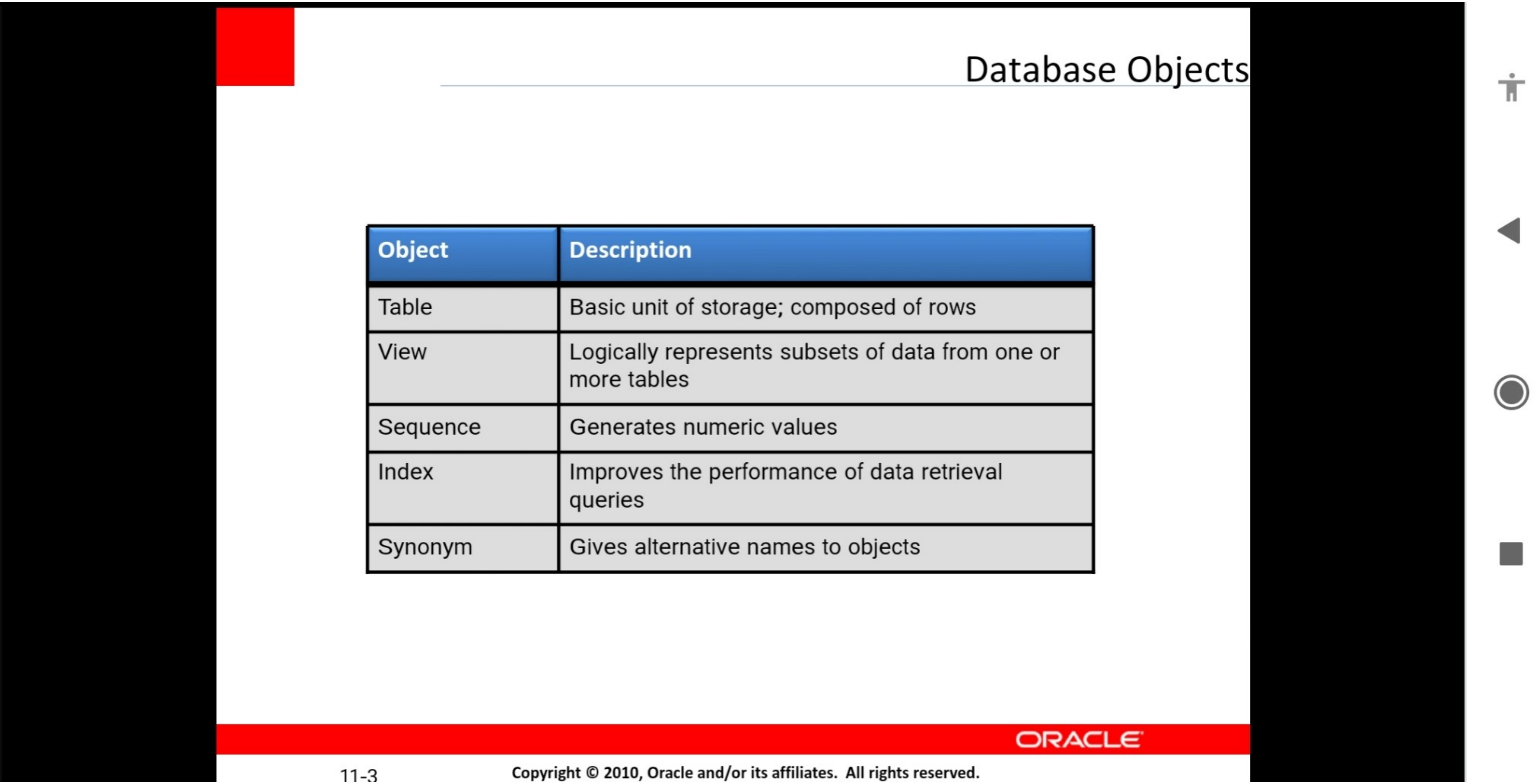


What you will learn at the end of this Session?

1. Create simple and complex views

2. Retrieve data from views

3. Create, maintain, and use sequences

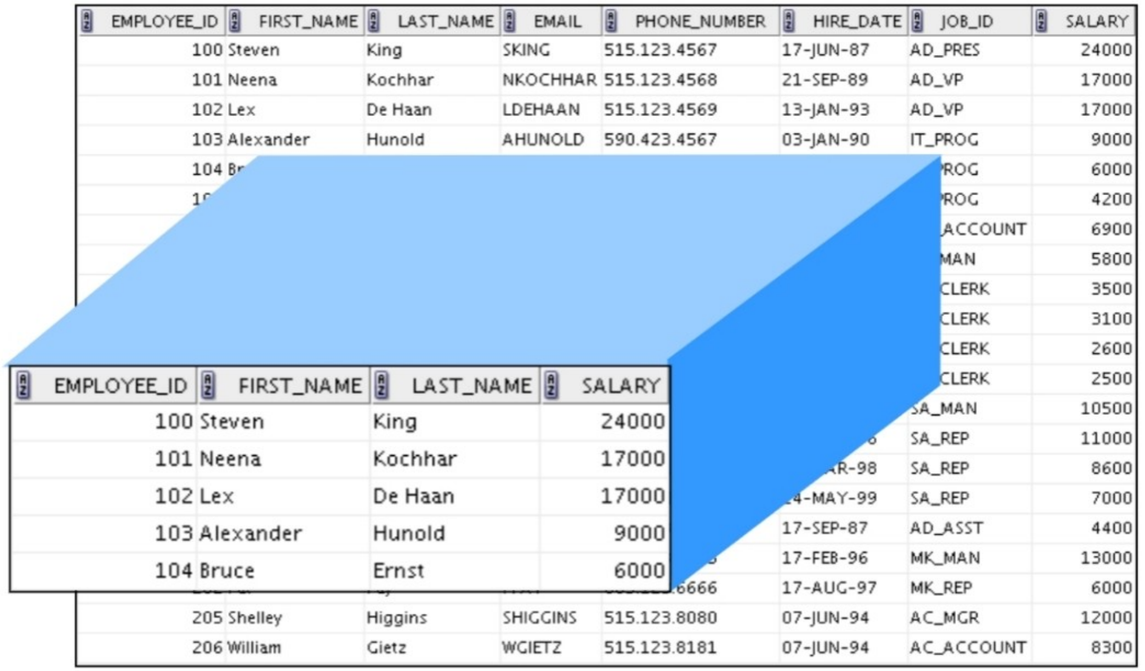


Database Objects

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 data retrieval queries
Synonym	Gives alternative names to objects

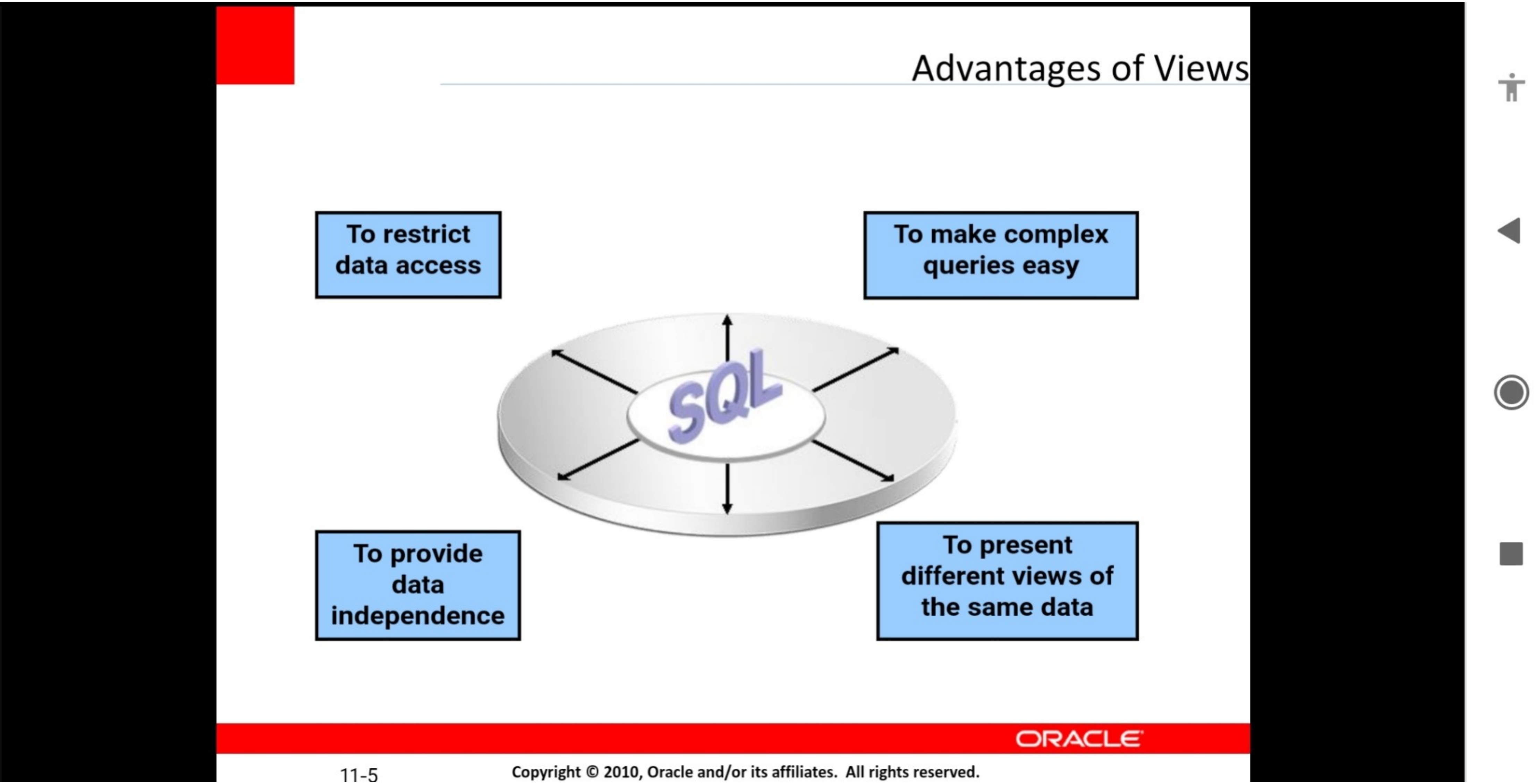
What Is a View?

EMPLOYEES table



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000
104	Br					ROG	6000
10						ROG	4200
						ACCOUNT	6900
						MAN	5800
						CLERK	3500
						CLERK	3100
						CLERK	2600
						CLERK	2500
						SA_MAN	10500
					6	SA_REP	11000
					AR-98	SA_REP	8600
					4-MAY-99	SA_REP	7000
					17-SEP-87	AD_ASST	4400
					17-FEB-96	MK_MAN	13000
				6666	17-AUG-97	MK_REP	6000
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	12000
206	William	Gietz	WGIETZ	515.123.8181	07-JUN-94	AC_ACCOUNT	8300

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
100	Steven	King	24000
101	Neena	Kochhar	17000
102	Lex	De Haan	17000
103	Alexander	Hunold	9000
104	Bruce	Ernst	6000



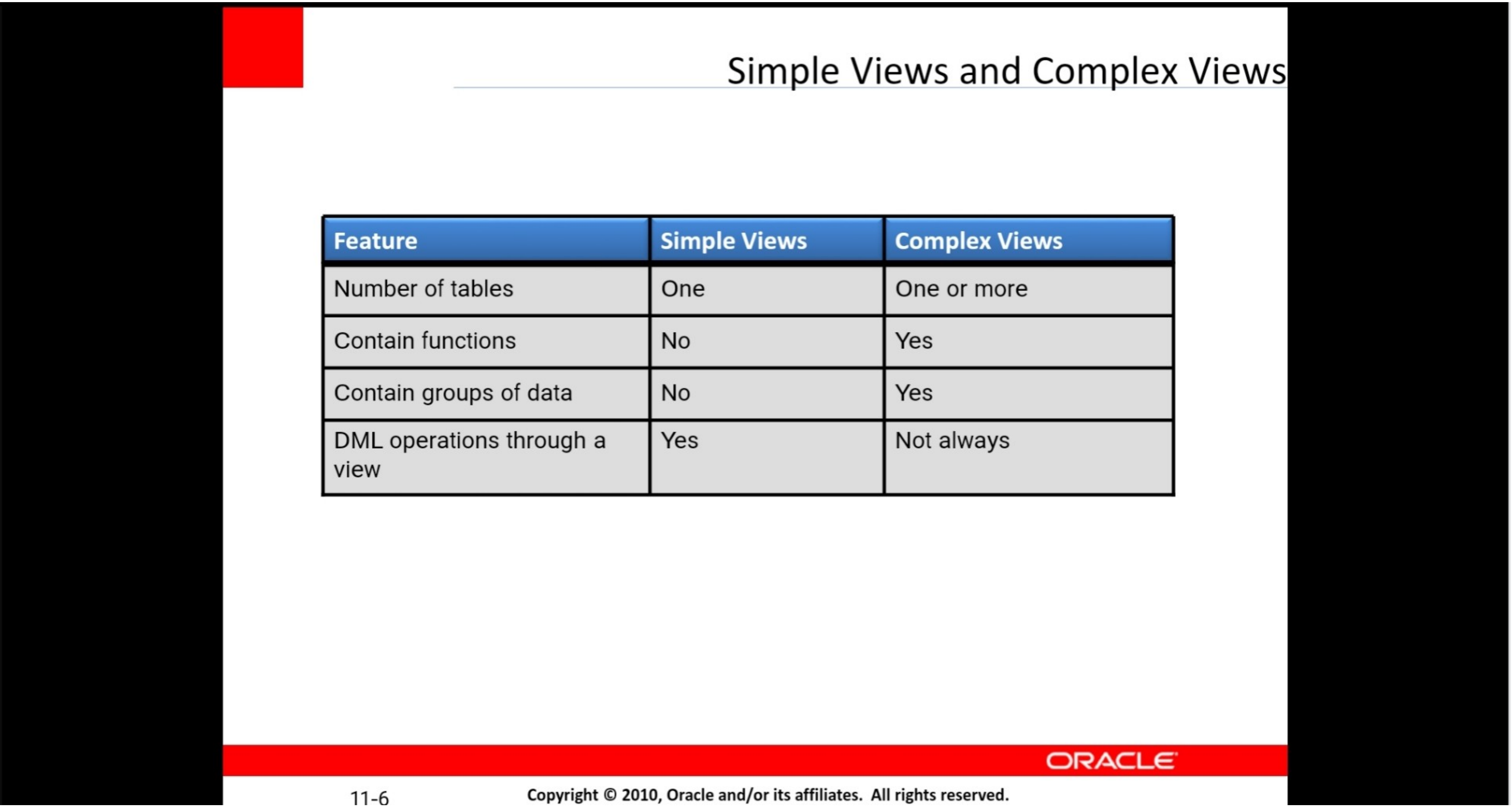
Advantages of Views

To restrict data access

To make complex queries easy

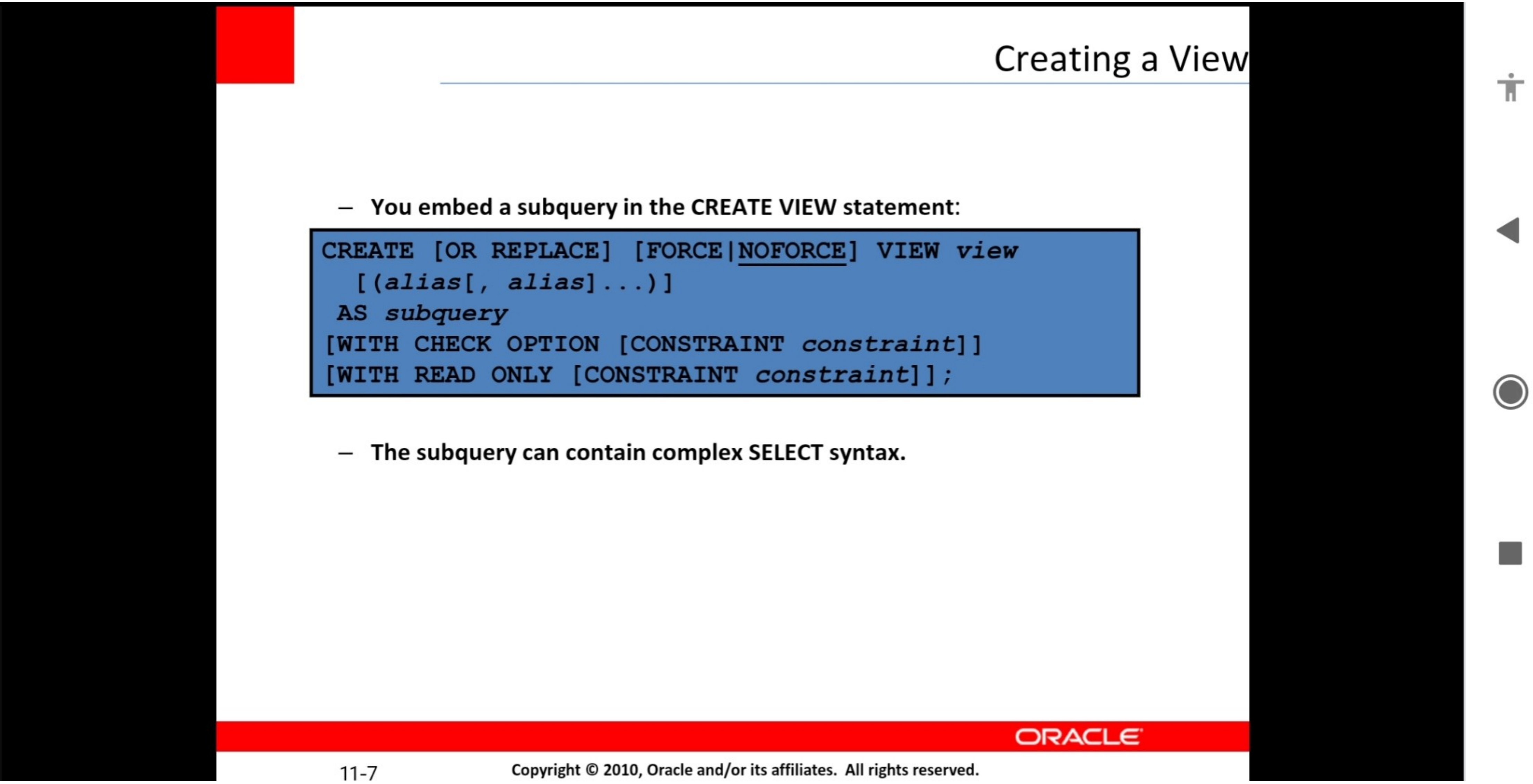
To provide data independence

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 operations through a view	Yes	Not always

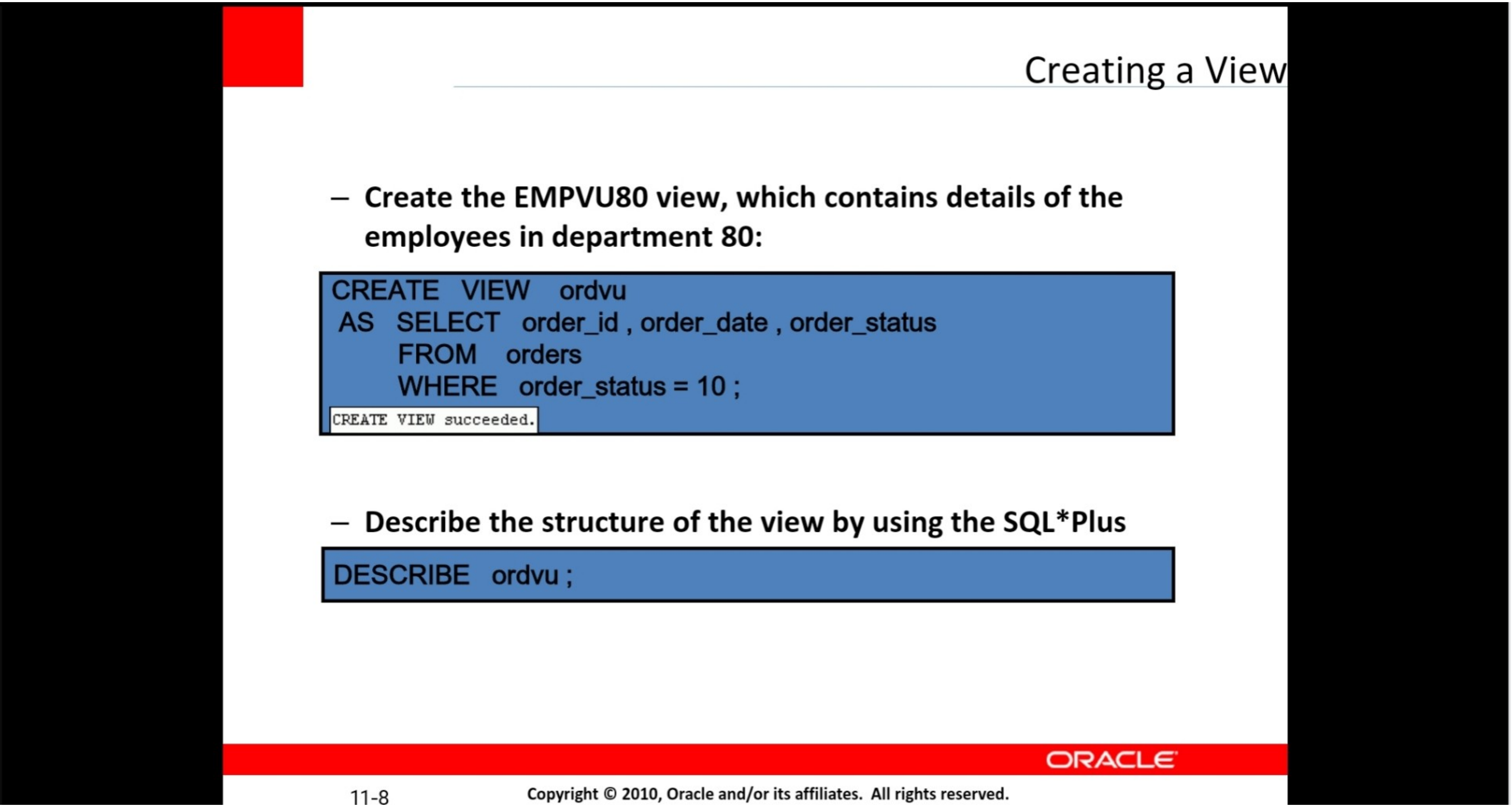


Creating a View

- 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

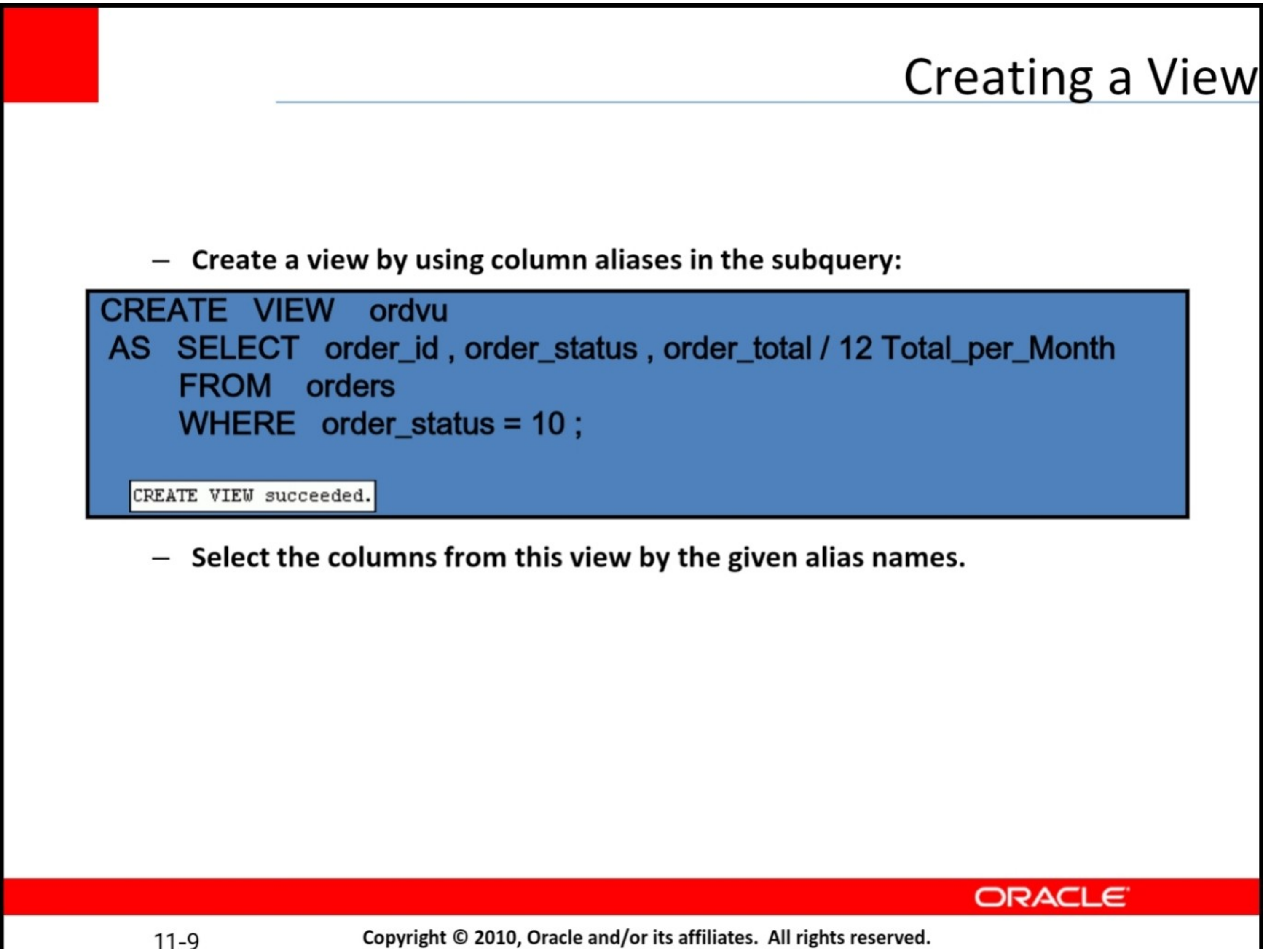
- Create the EMPVU80 view, which contains details of the employees in department 80:

```
CREATE   VIEW    ordvu
 AS   SELECT   order_id , order_date , order_status
      FROM    orders
      WHERE   order_status = 10 ;
```

```
CREATE VIEW succeeded.
```

- Describe the structure of the view by using the SQL*Plus

```
DESCRIBE   ordvu ;
```



Creating a View

– **Create a view by using column aliases in the subquery:**

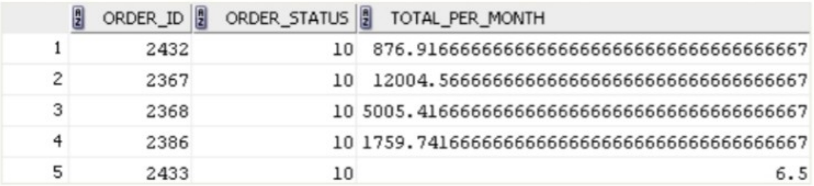
```
CREATE  VIEW   ordvu
 AS  SELECT  order_id , order_status , order_total / 12 Total_per_Month
     FROM   orders
     WHERE  order_status = 10 ;
```

```
CREATE VIEW succeeded.
```

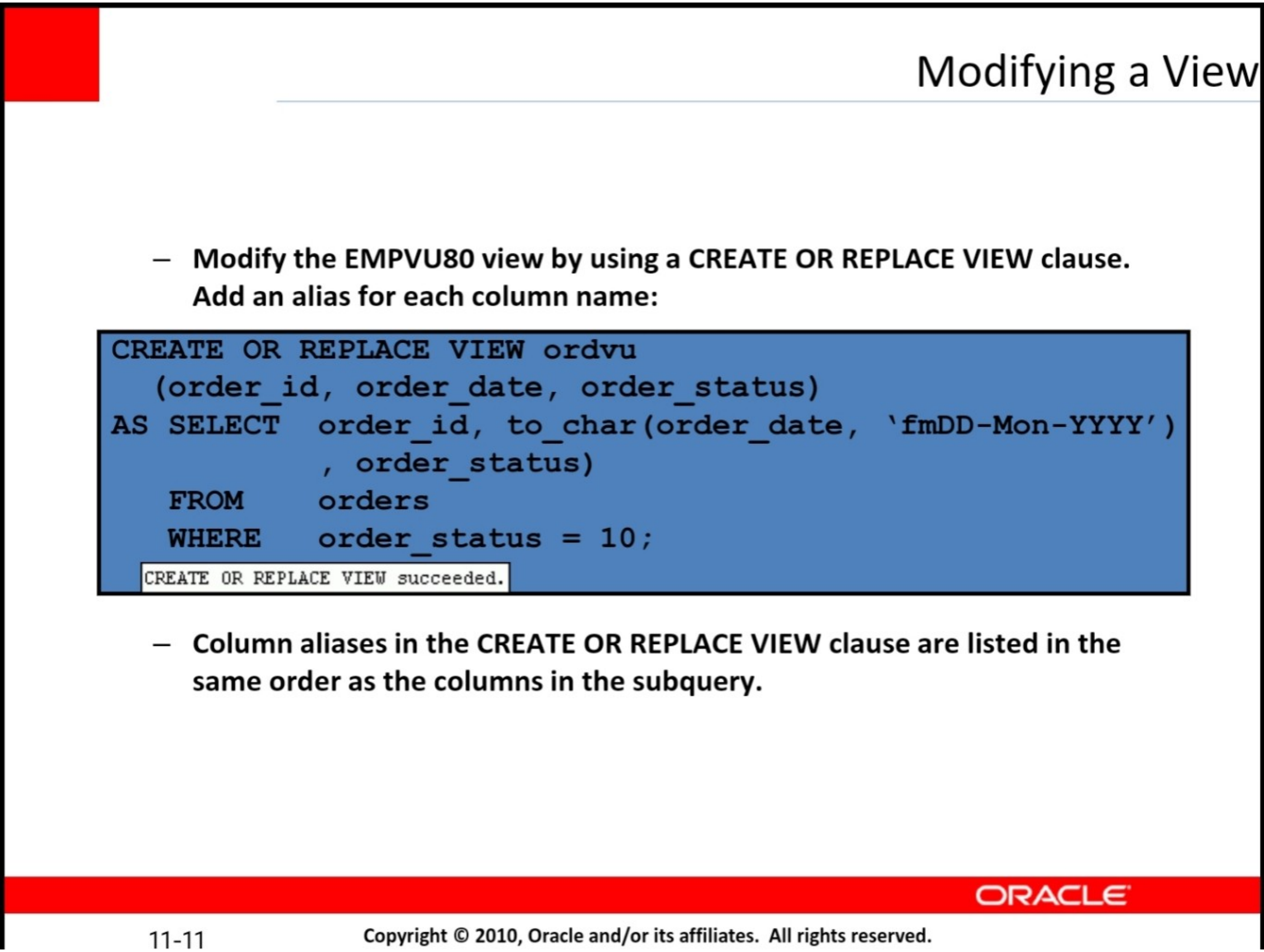
– **Select the columns from this view by the given alias names.**

Retrieving Data from a View

```
SELECT  *
FROM  ordvu ;
```



	ORDER_ID	ORDER_STATUS	TOTAL_PER_MONTH
1	2432	10	876.91666666666666666666666666666666667
2	2367	10	12004.56666666666666666666666666666667
3	2368	10	5005.41666666666666666666666666666666667
4	2386	10	1759.74166666666666666666666666666666667
5	2433	10	6.5



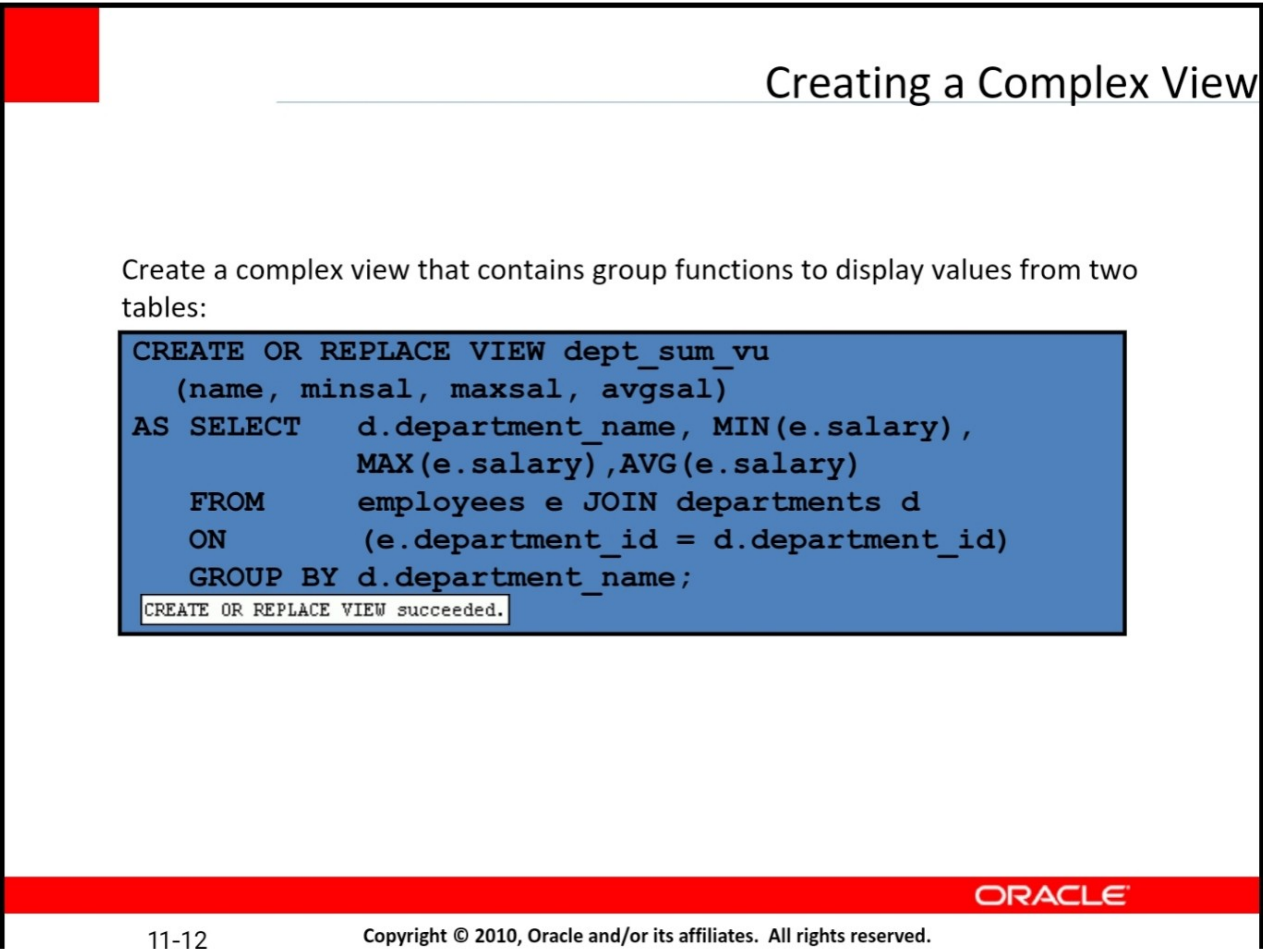
Modifying a View

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

```
CREATE OR REPLACE VIEW ordvu
  (order_id, order_date, order_status)
AS SELECT  order_id, to_char(order_date, ‘fmDD-Mon-YYYY’)
           , order_status)
   FROM    orders
   WHERE   order_status = 10;
```

```
CREATE OR REPLACE VIEW succeeded.
```

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

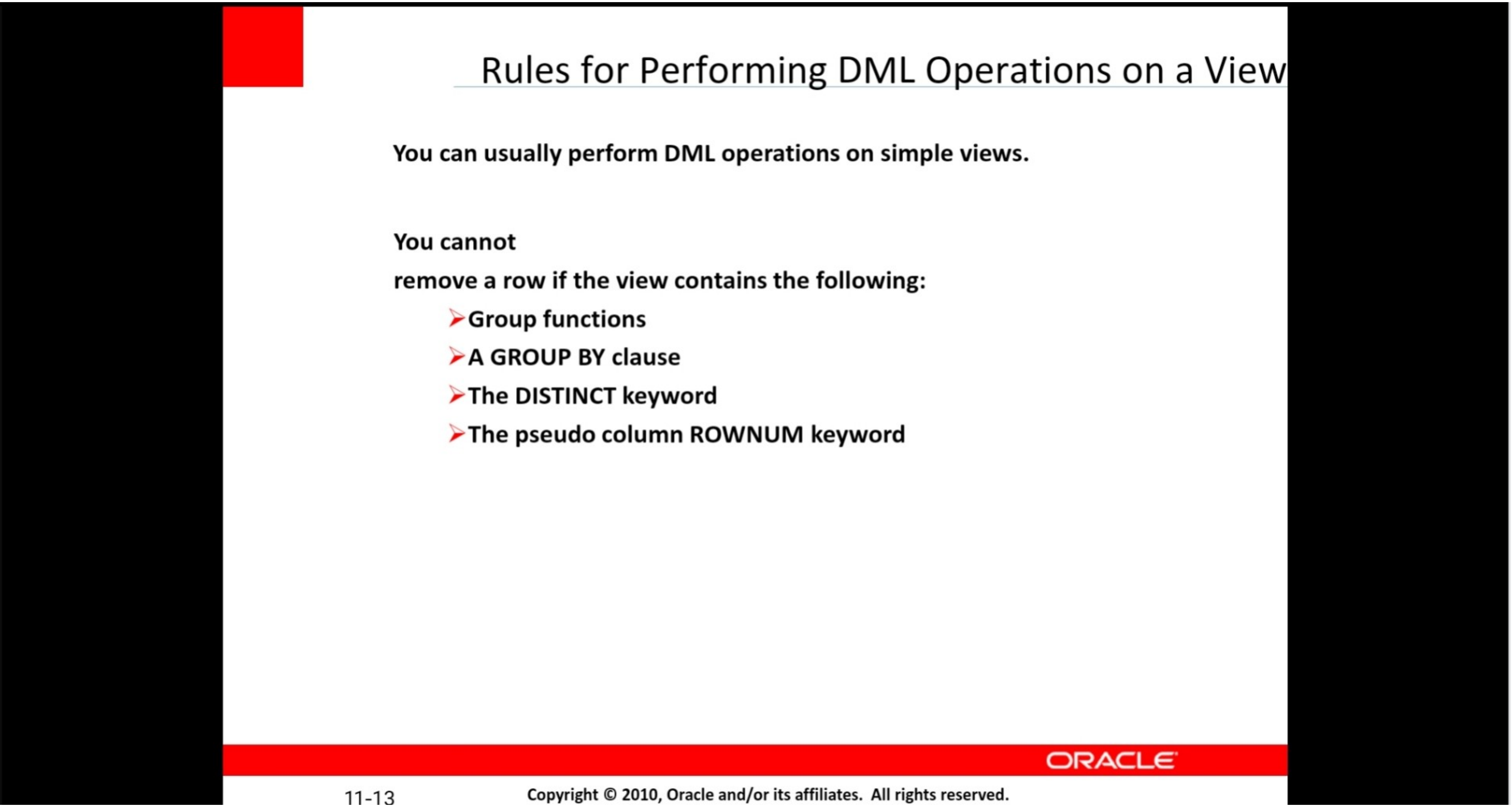


Creating a Complex View

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;
```

```
CREATE OR REPLACE VIEW succeeded.
```



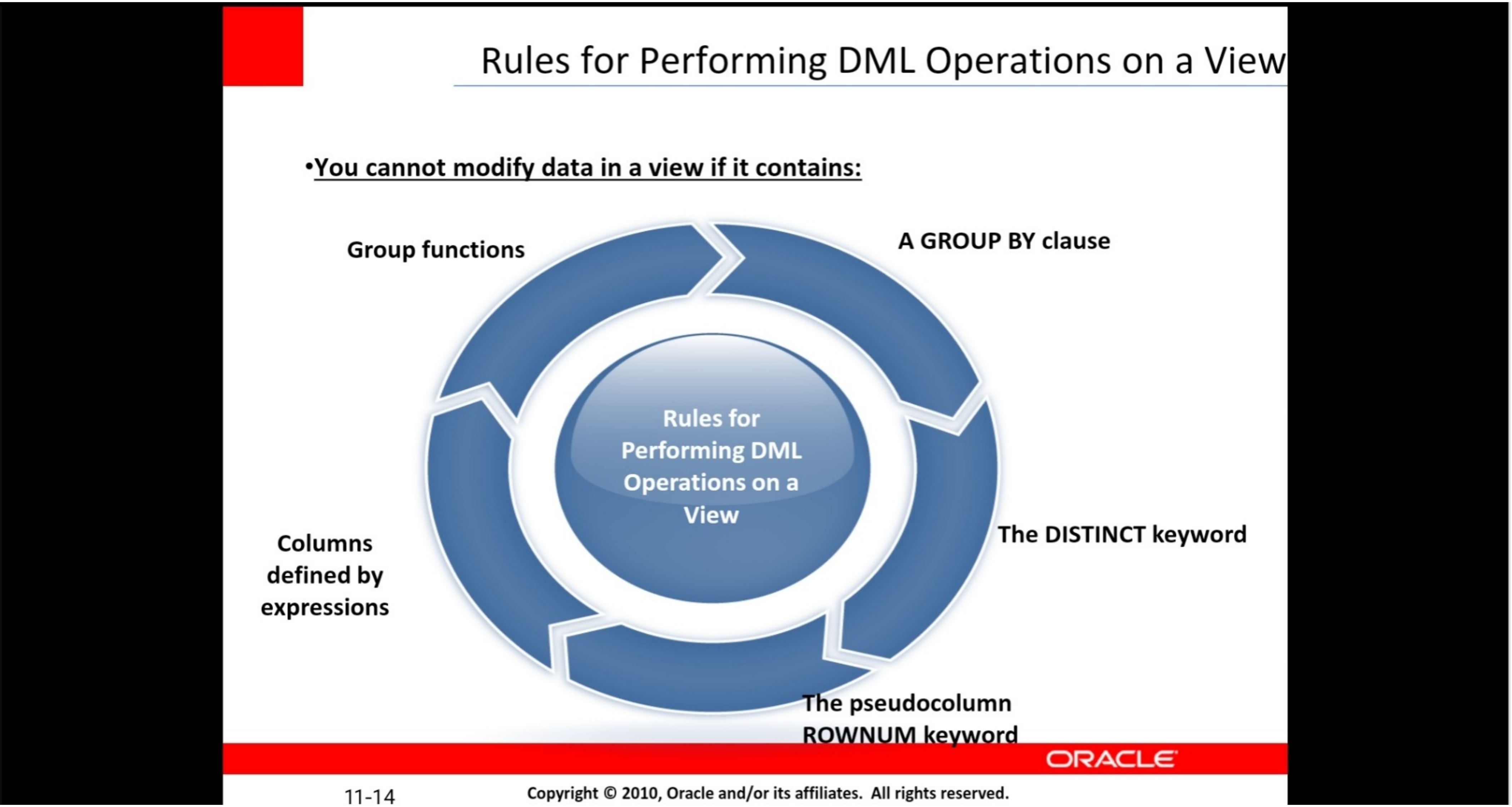
Rules for Performing DML Operations on a View

You can usually 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**
- **The pseudo column ROWNUM keyword**

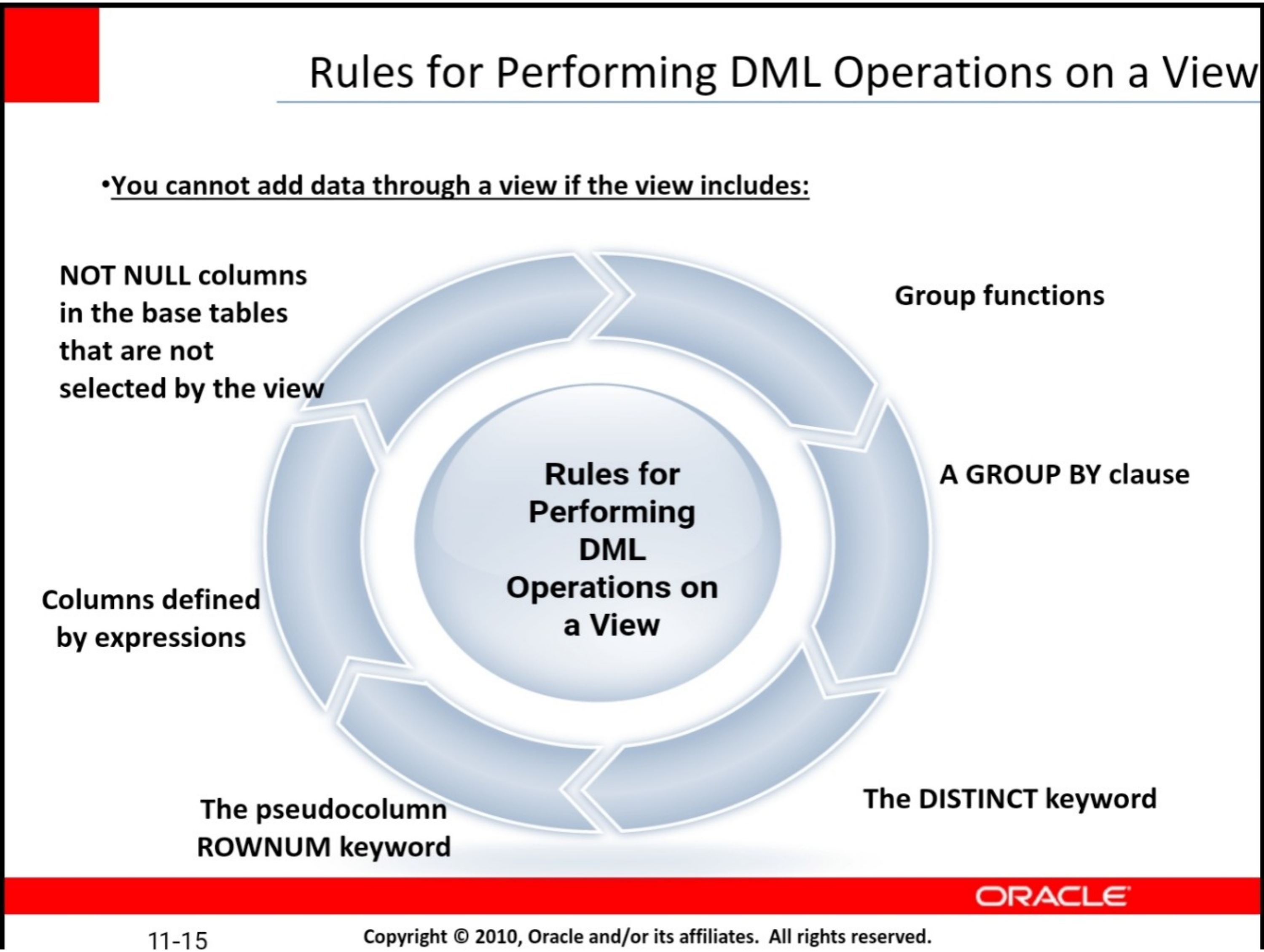


Rules for Performing DML Operations on a View

•**You cannot modify data in a view if it contains:**

Rules for Performing DML Operations on a View

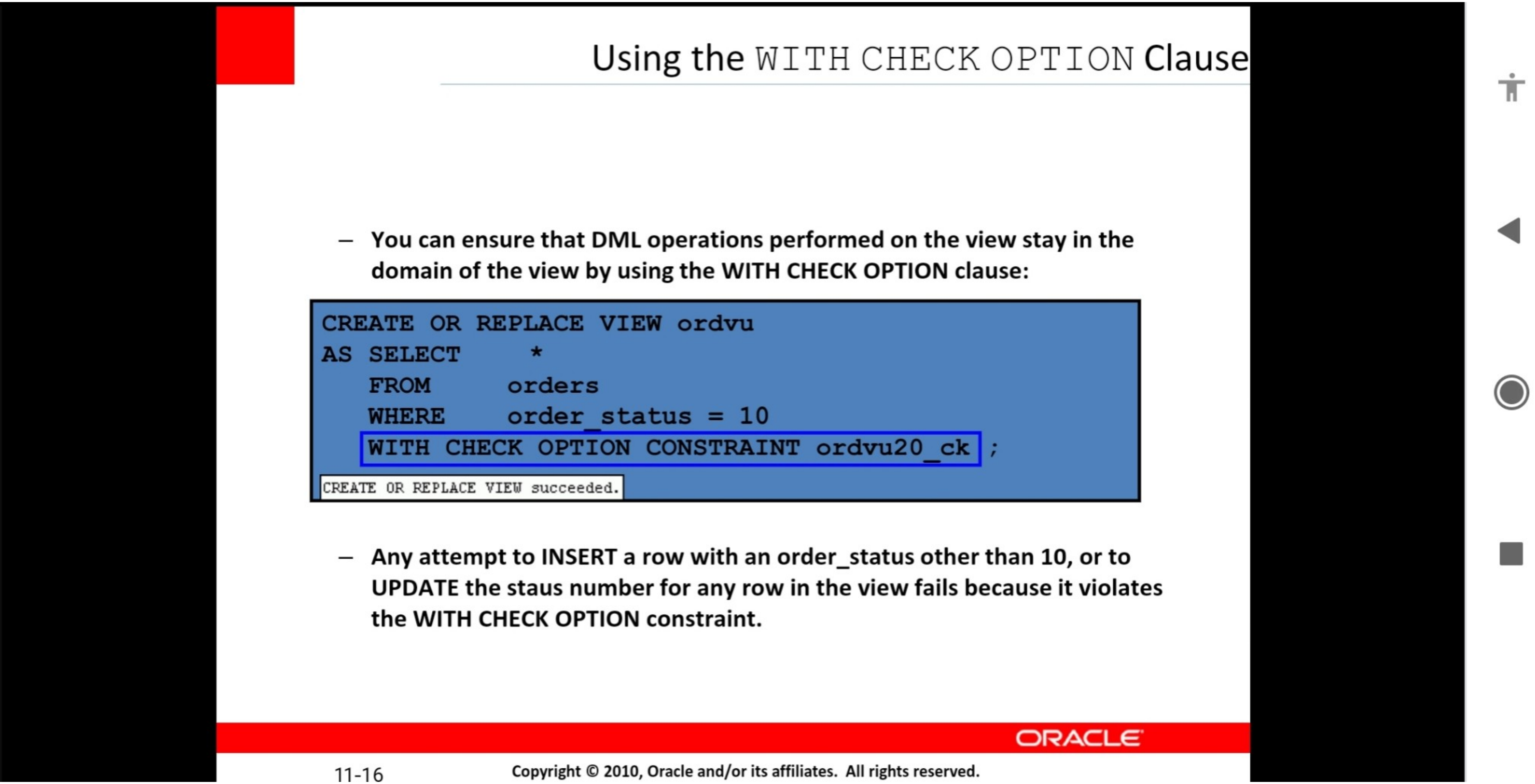
- Group functions
- A GROUP BY clause
- The DISTINCT keyword
- The pseudocolumn ROWNUM keyword
- Columns defined by expressions



Rules for Performing DML Operations on a View

- **You cannot add data through a view if the view includes:**

Rules for Performing DML Operations on a View
NOT NULL columns in the base tables that are not selected by the view
Group functions
A GROUP BY clause
The DISTINCT keyword
The pseudocolumn ROWNUM keyword
Columns defined by expressions



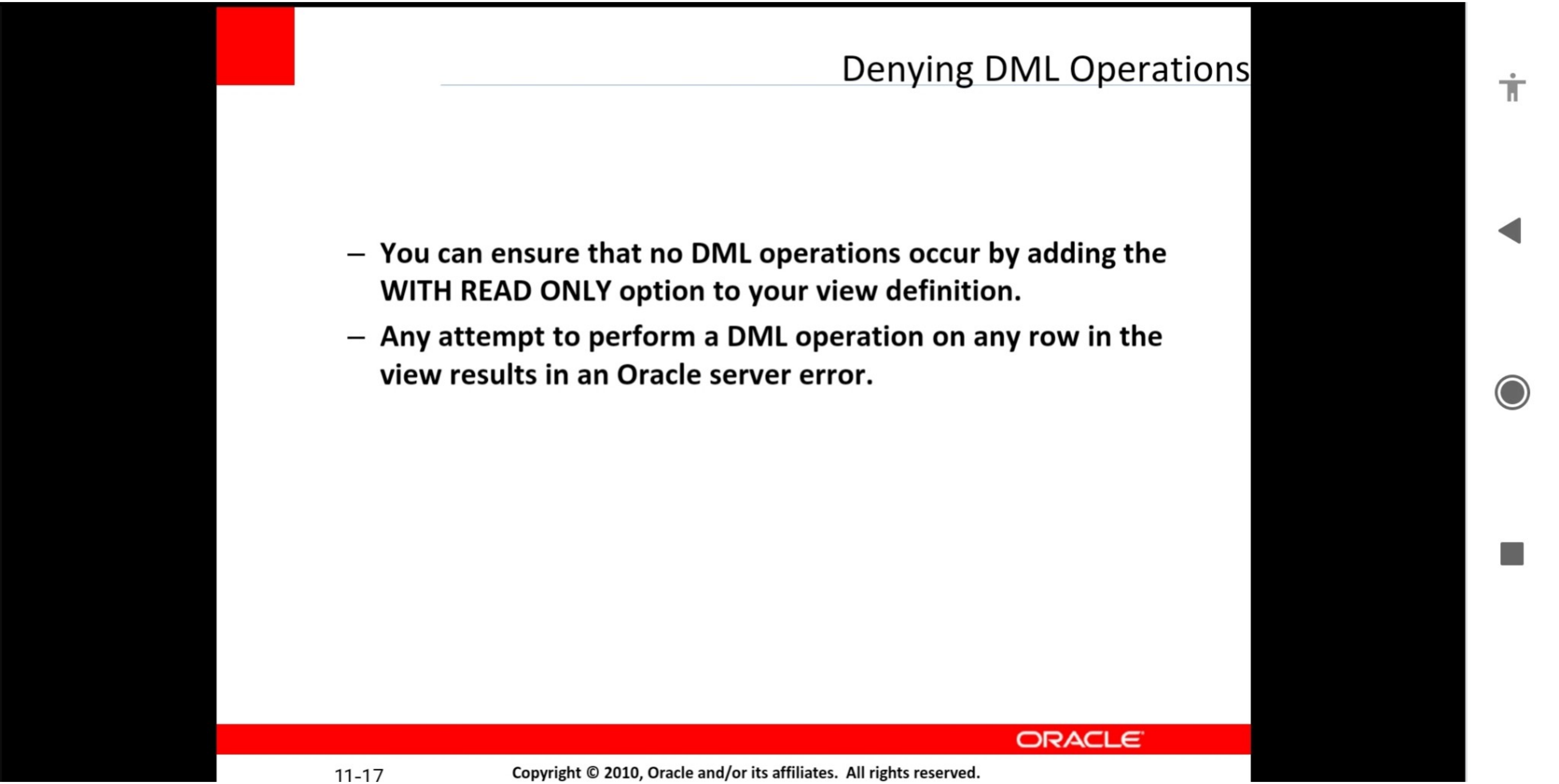
Using the WITH CHECK OPTION Clause

– You can ensure that DML operations performed on the view stay in the domain of the view by using the WITH CHECK OPTION clause:

```
CREATE OR REPLACE VIEW ordvu
AS SELECT    *
   FROM     orders
   WHERE    order_status = 10
   WITH CHECK OPTION CONSTRAINT ordvu20_ck ;
```

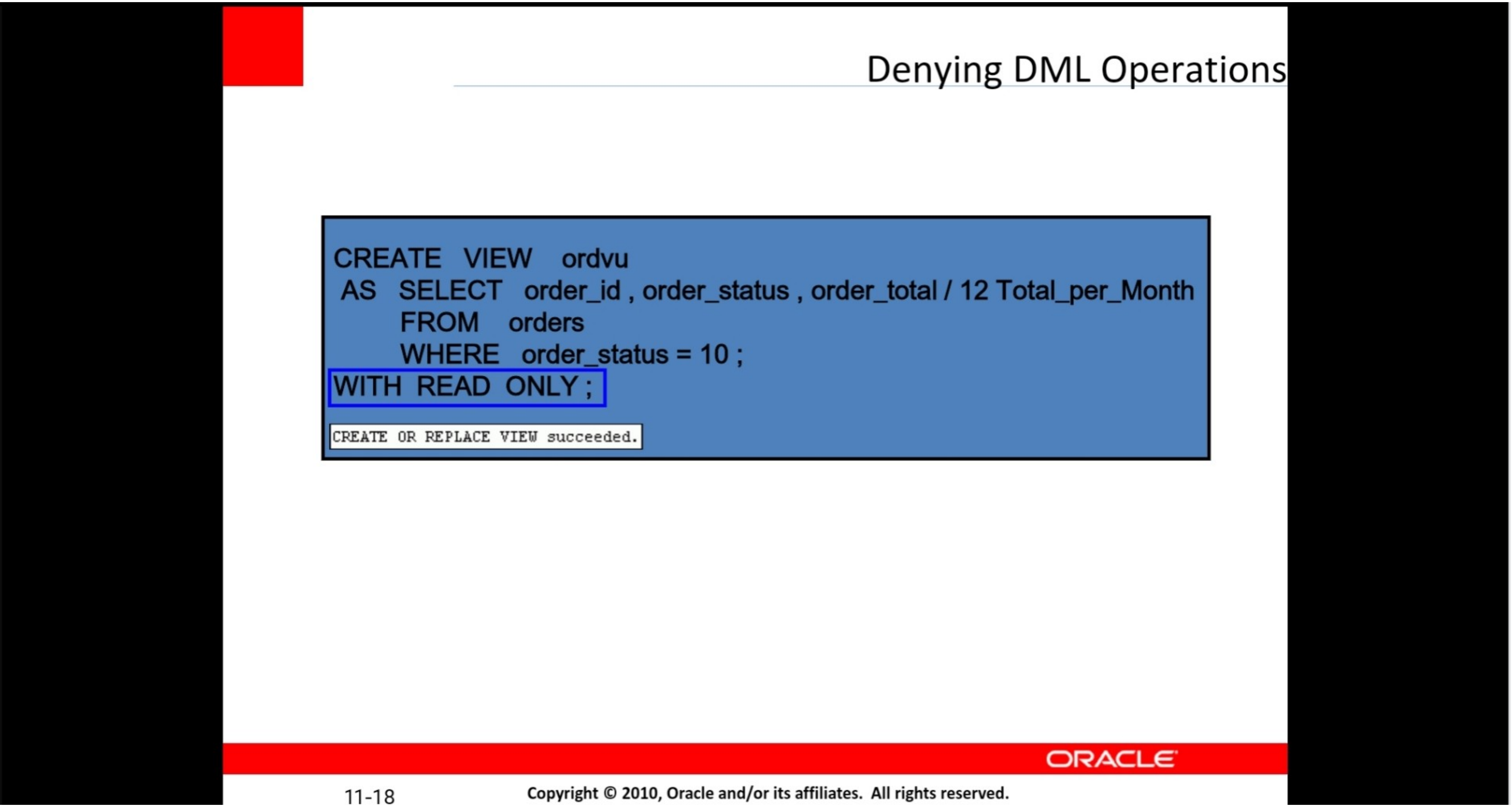
```
CREATE OR REPLACE VIEW succeeded.
```

– Any attempt to INSERT a row with an order_status other than 10, or to UPDATE the staus number for any row in the view fails because it violates the WITH CHECK OPTION constraint.



Denying DML Operations

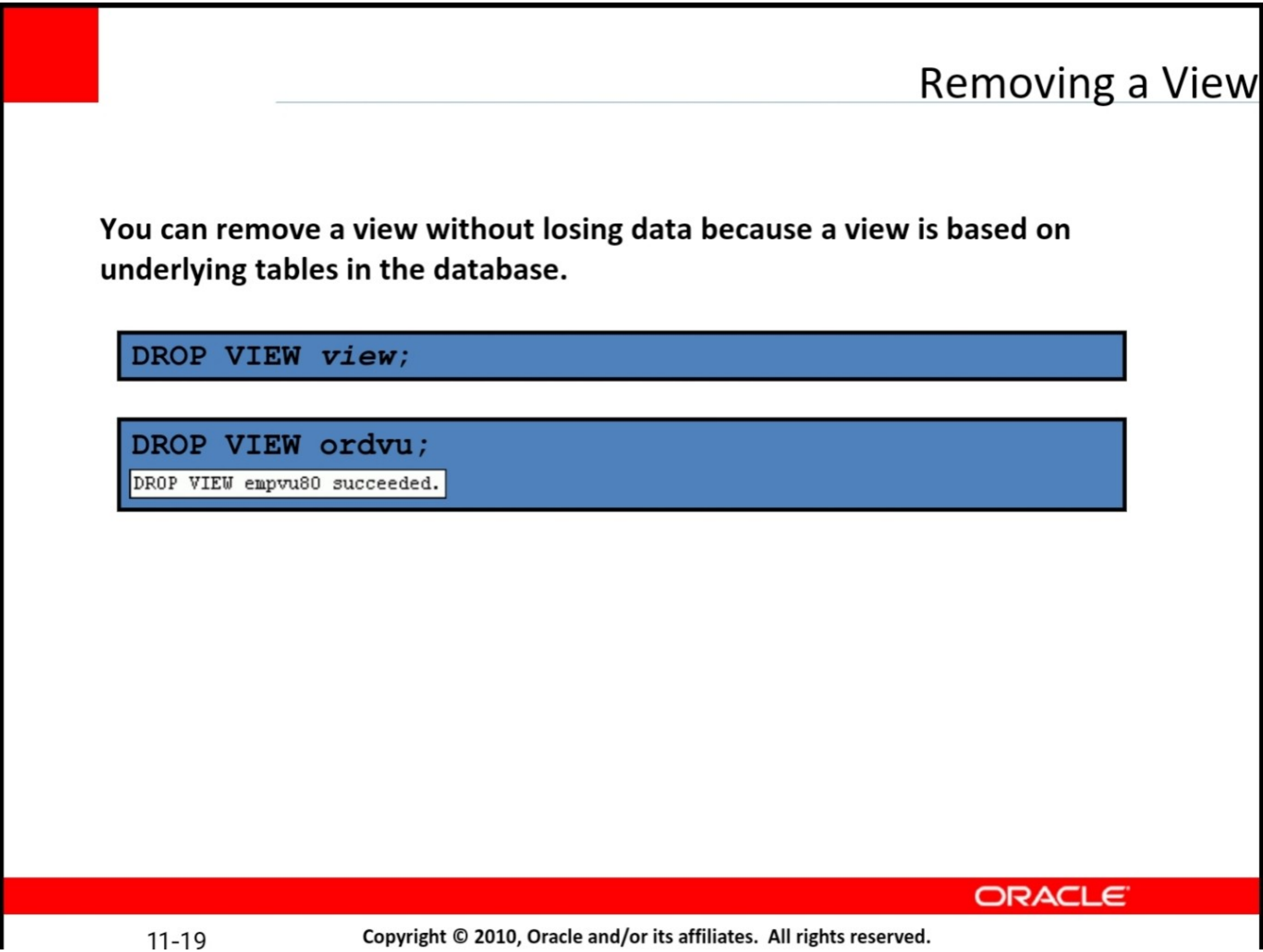
- You can ensure that no DML operations occur by adding the WITH READ ONLY option to your view definition.
- Any attempt to perform a DML operation on any row in the view results in an Oracle server error.



Denying DML Operations

```
CREATE   VIEW    ordvu
 AS   SELECT   order_id , order_status , order_total / 12 Total_per_Month
      FROM    orders
      WHERE   order_status = 10 ;
WITH  READ  ONLY ;
```

```
CREATE OR REPLACE VIEW succeeded.
```



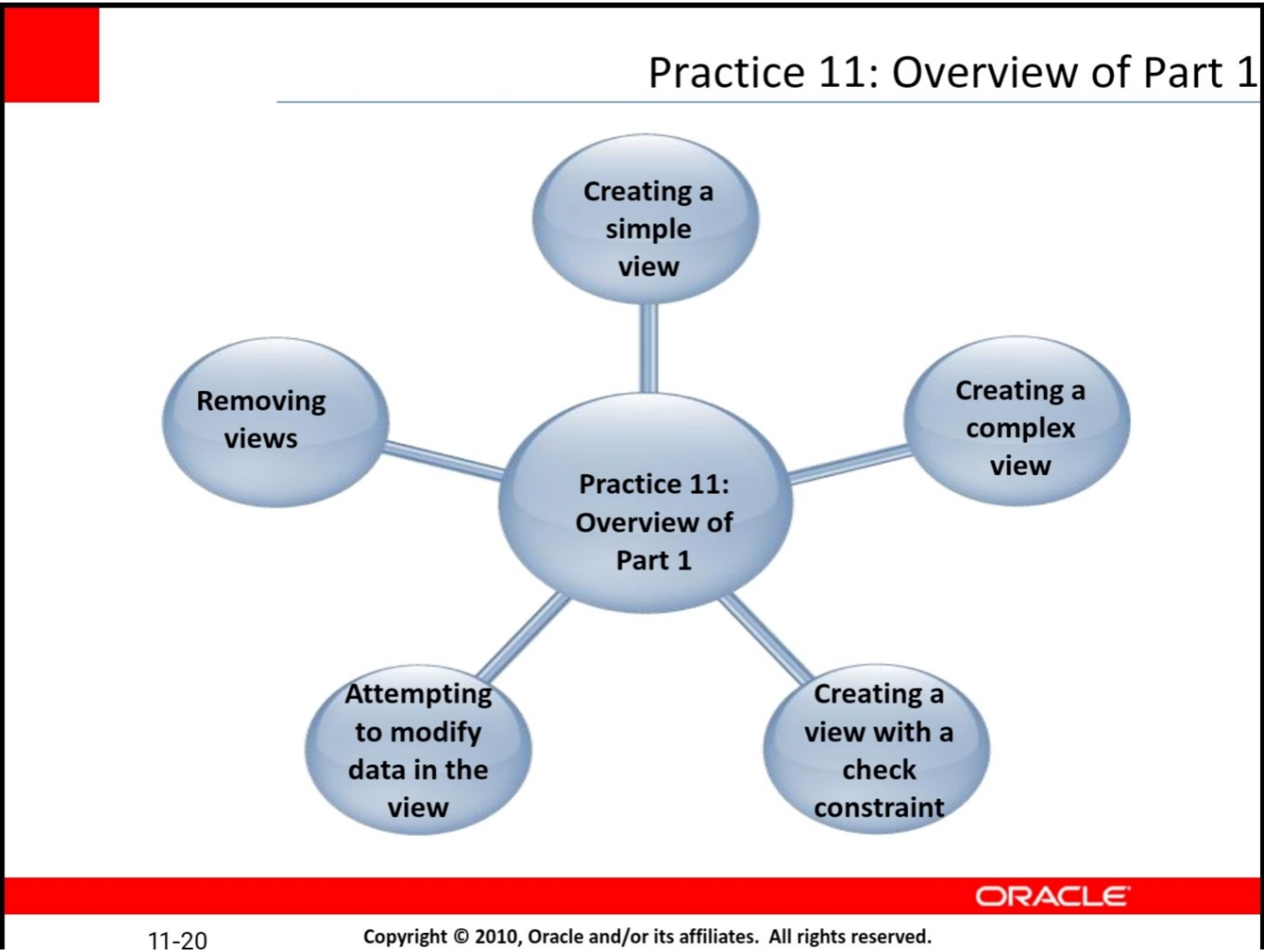
Removing a View

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

```
DROP VIEW view;
```

```
DROP VIEW ordvu;
```

```
DROP VIEW empvu80 succeeded.
```



Practice 11: Overview of Part 1

Practice 11: Overview of Part 1

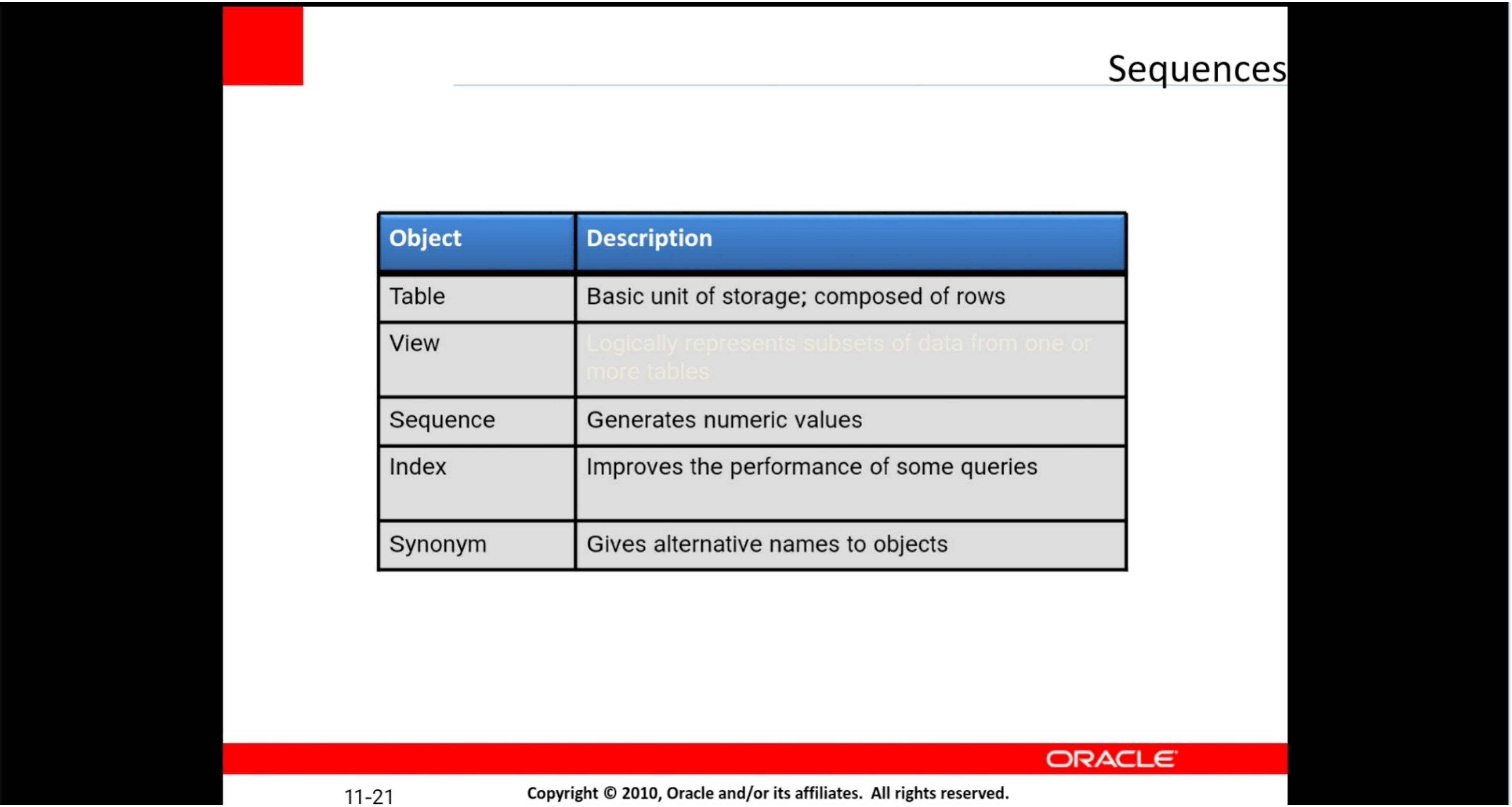
Creating a simple view

Creating a complex view

Creating a view with a check constraint

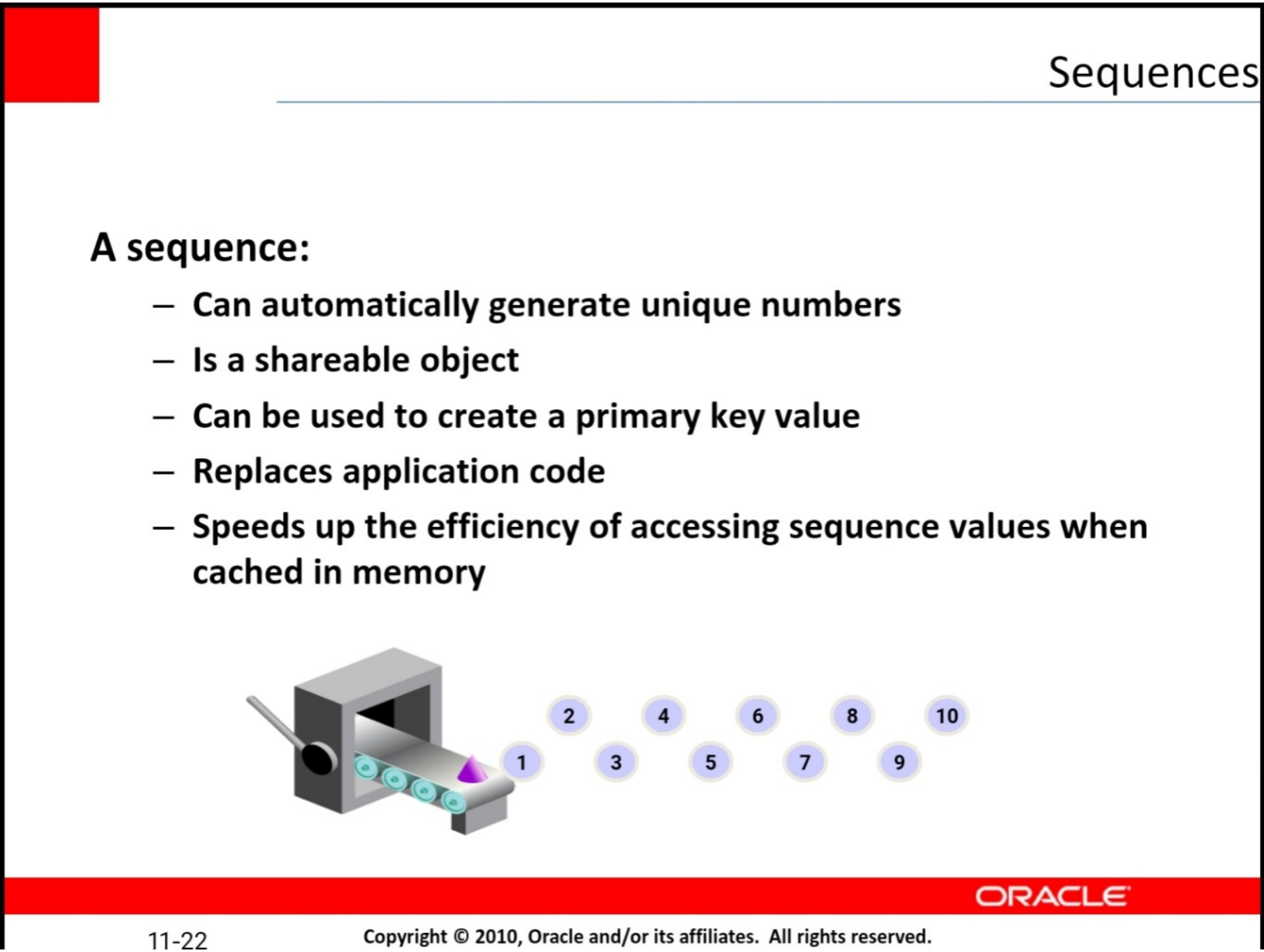
Attempting to modify data in the view

Removing views



Sequences

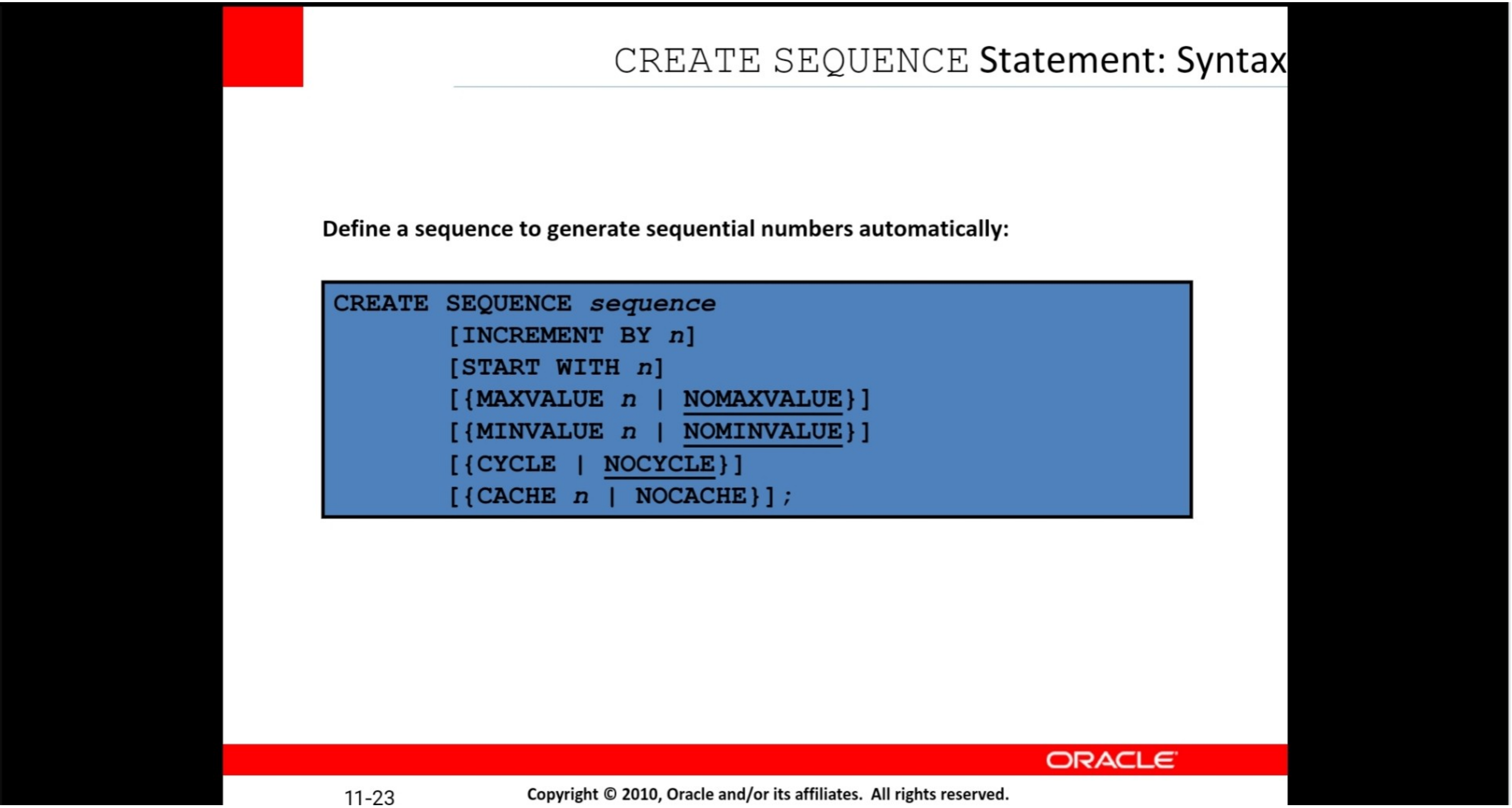
Object	Description
Table	Basic unit of storage; composed of rows
View	Logically represents subsets of data from one or more tables
Sequence	Generates numeric values
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects



Sequences

A sequence:

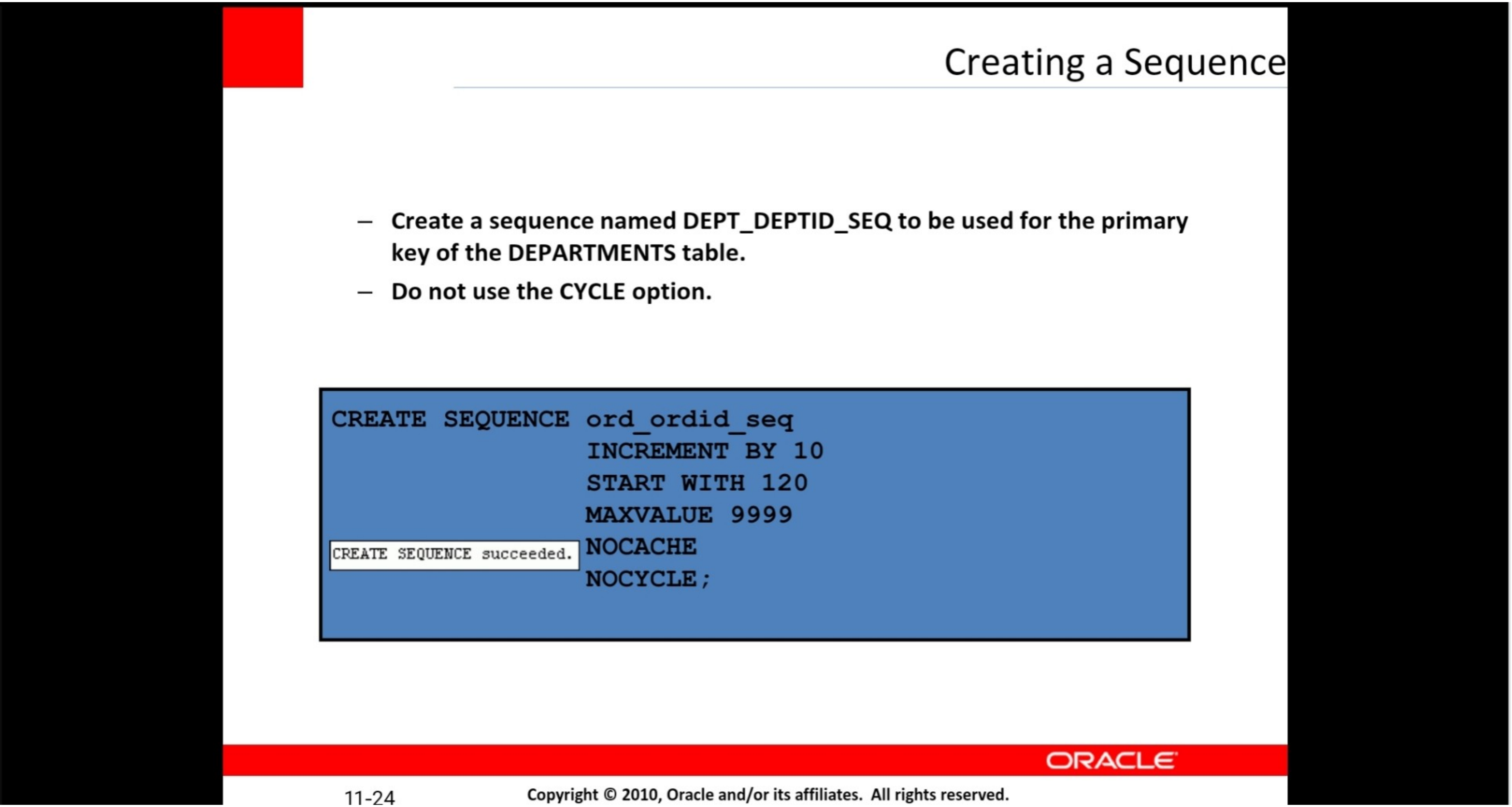
- **Can automatically generate unique numbers**
- **Is a shareable object**
- **Can be used to create a primary key value**
- **Replaces application code**
- **Speeds up the efficiency of accessing sequence values when cached in memory**



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}];
```

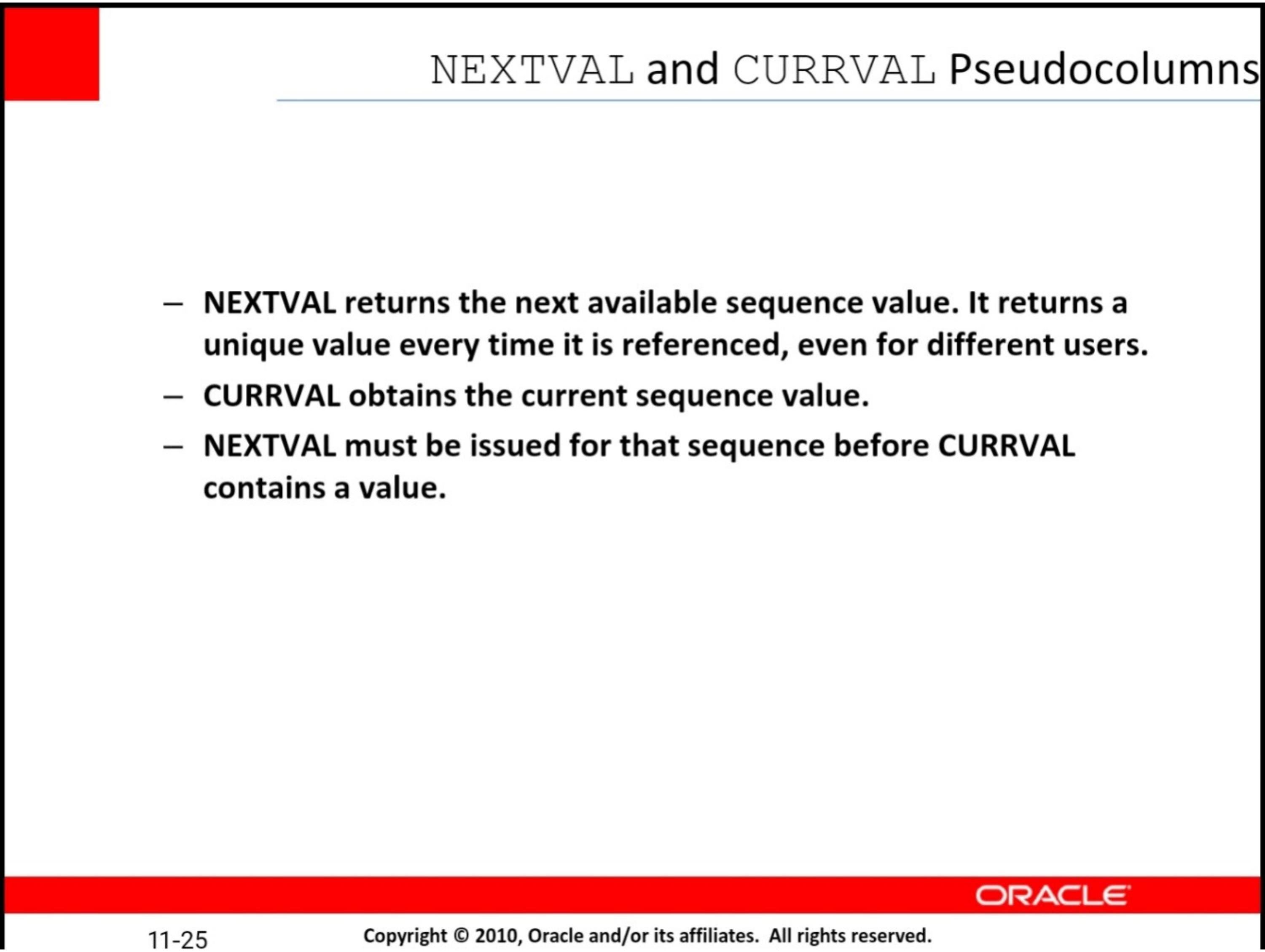


Creating a Sequence

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

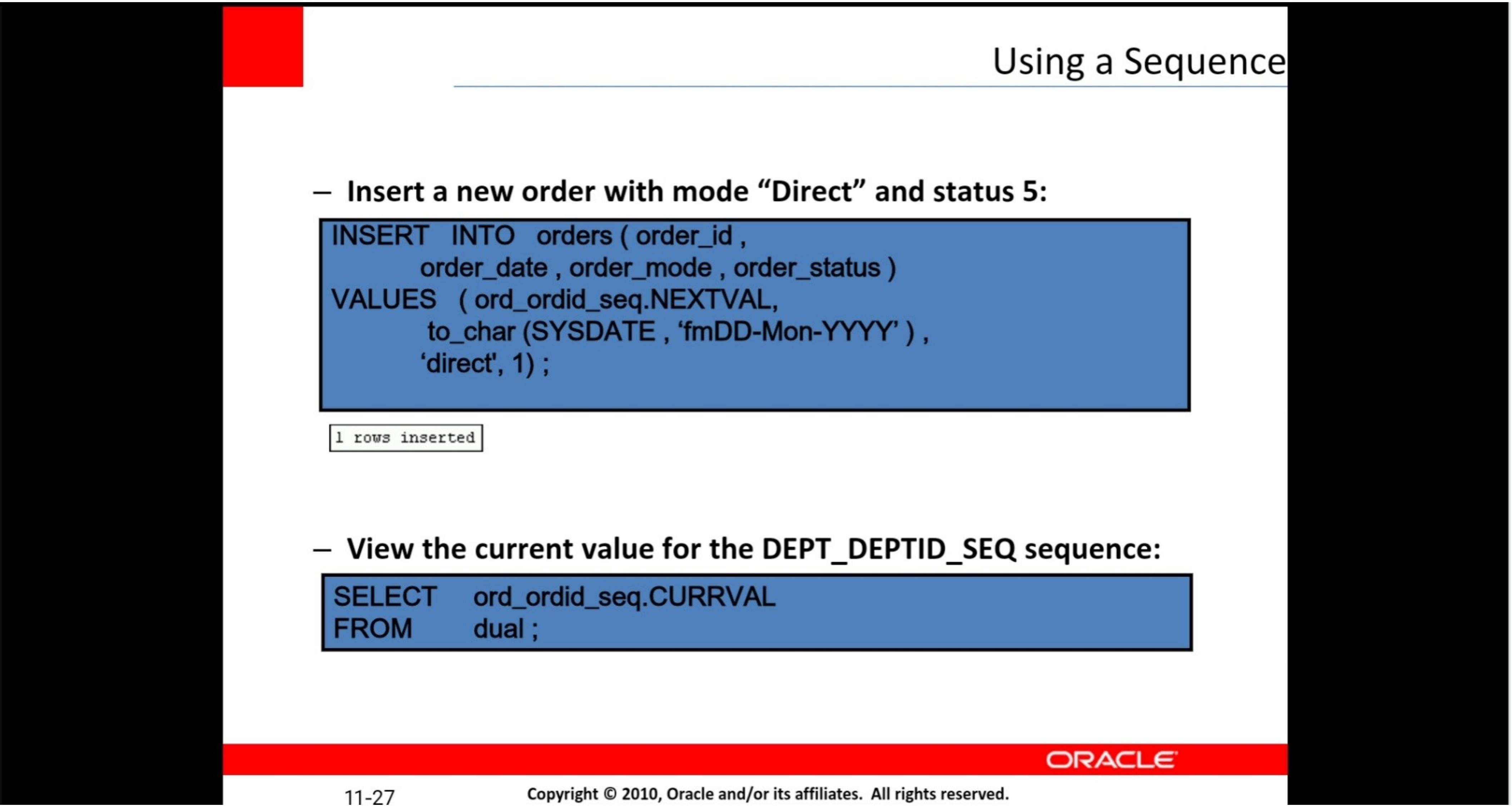
```
CREATE SEQUENCE ord_ordid_seq
                INCREMENT BY 10
                START WITH 120
                MAXVALUE 9999
                NOCACHE
                NOCYCLE;
```

```
CREATE SEQUENCE succeeded.
```



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.



Using a Sequence

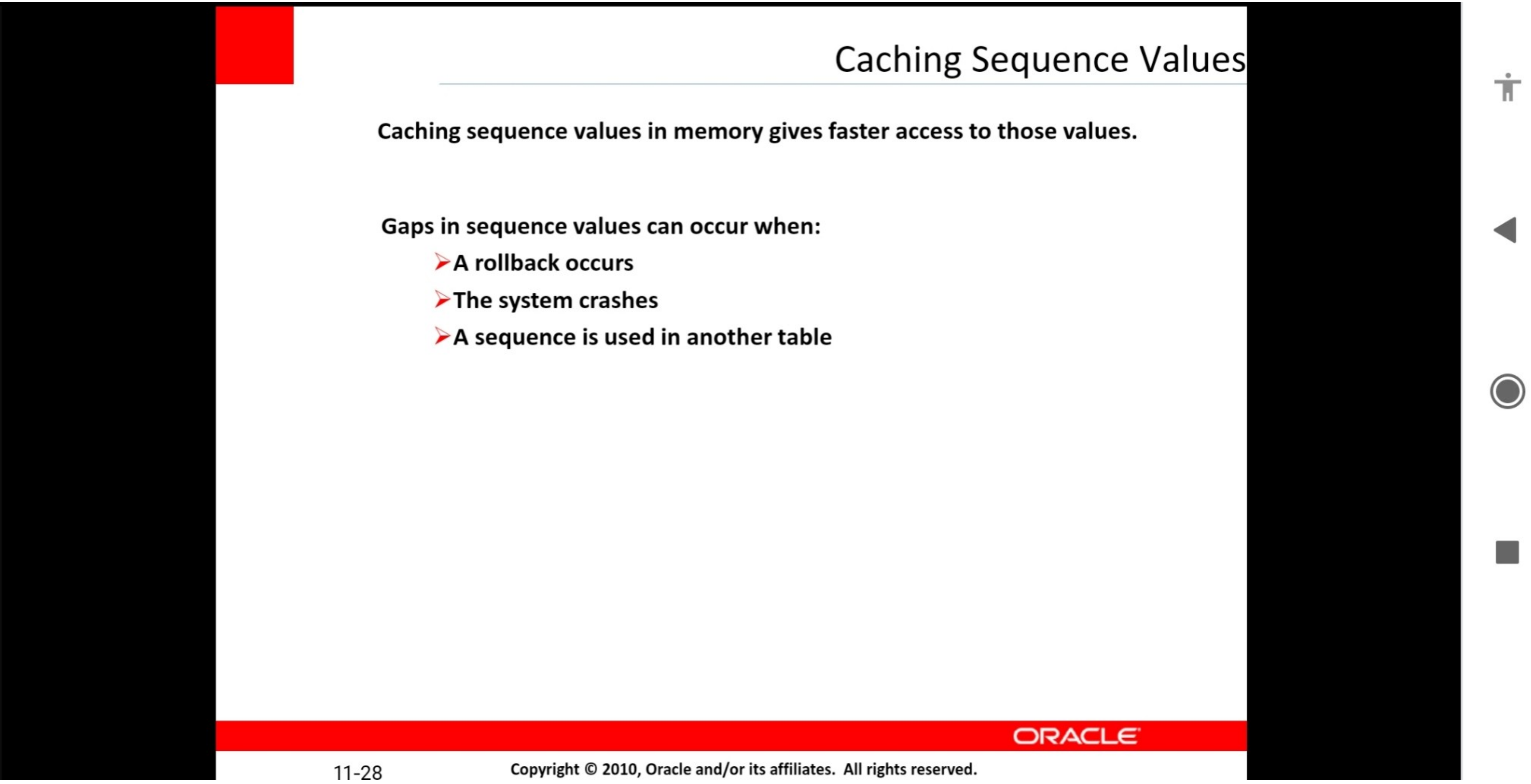
– **Insert a new order with mode “Direct” and status 5:**

```
INSERT   INTO   orders ( order_id ,
          order_date , order_mode , order_status )
VALUES   ( ord_ordid_seq.NEXTVAL,
           to_char (SYSDATE , ‘fmDD-Mon-YYYY’ ) ,
          ‘direct’, 1) ;
```

```
1 rows inserted
```

– **View the current value for the DEPT_DEPTID_SEQ sequence:**

```
SELECT     ord_ordid_seq.CURRVAL
FROM       dual ;
```

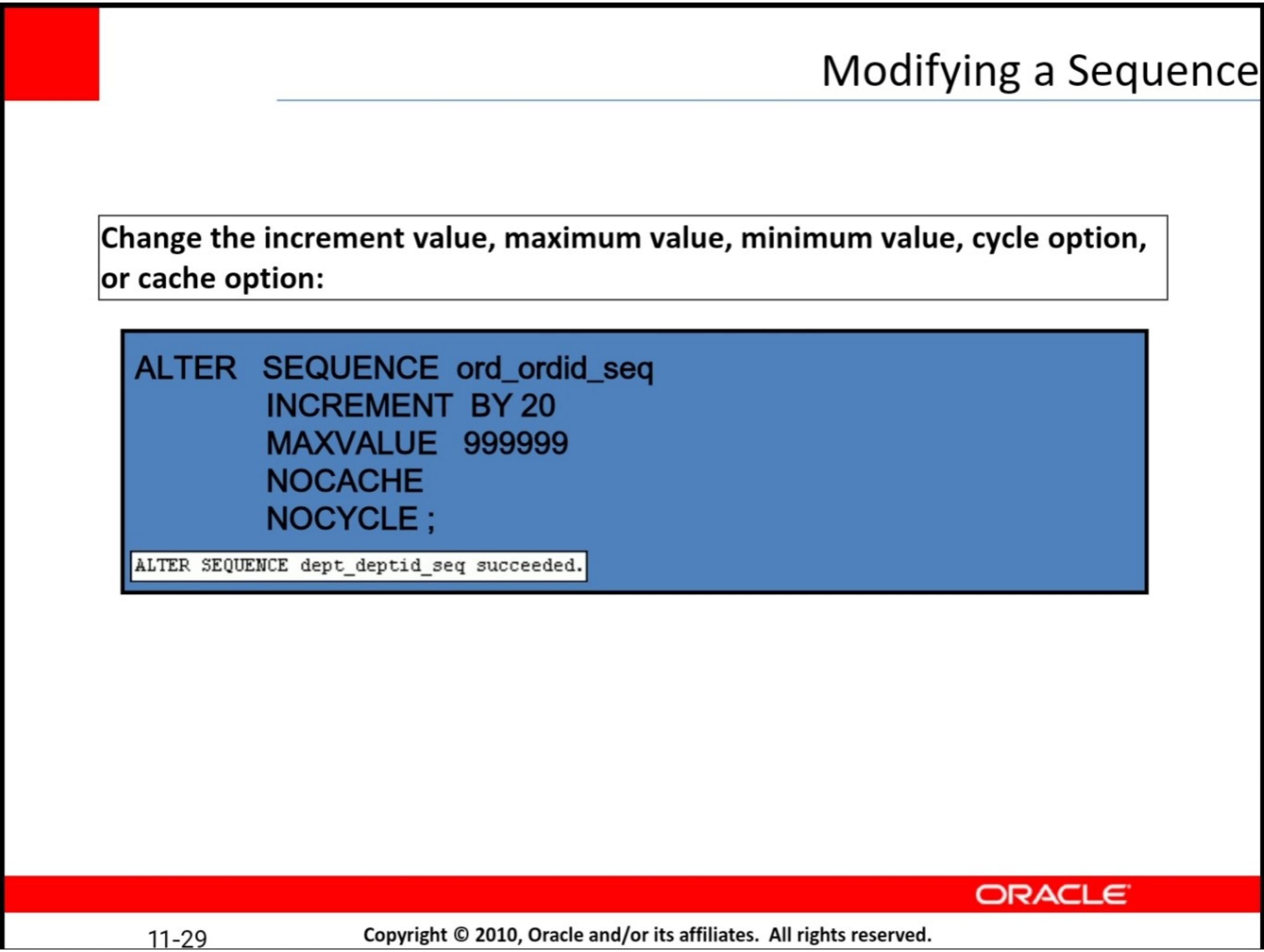


Caching Sequence Values

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**

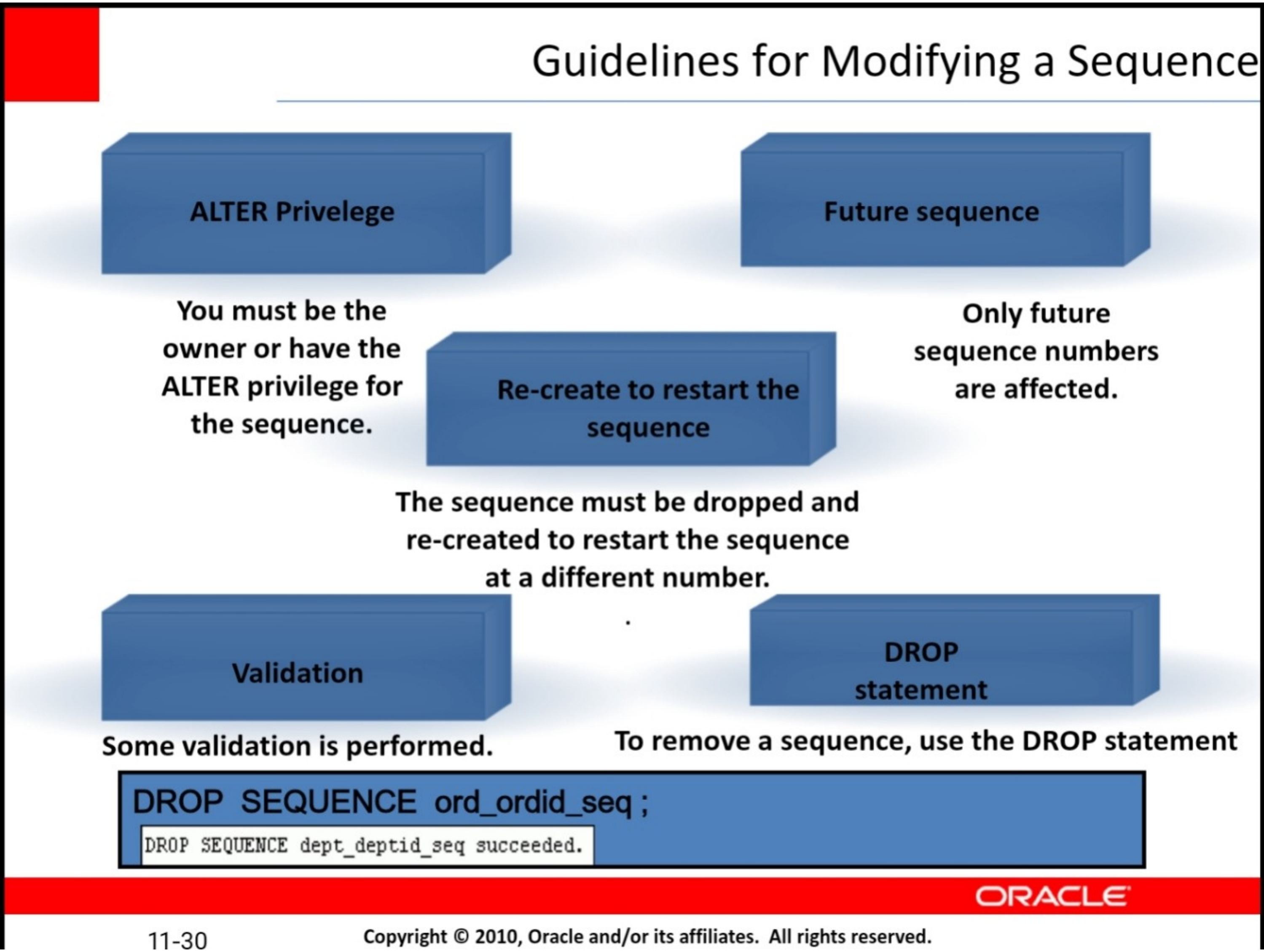


Modifying a Sequence

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

```
ALTER   SEQUENCE  ord_ordid_seq
          INCREMENT  BY 20
          MAXVALUE   999999
          NOCACHE
          NOCYCLE ;
```

```
ALTER SEQUENCE dept_deptid_seq succeeded.
```



Guidelines for Modifying a Sequence

ALTER Privelege

You must be the owner or have the ALTER privilege for the sequence.

Future sequence

Only future sequence numbers are affected.

Re-create to restart the sequence

The sequence must be dropped and re-created to restart the sequence at a different number.

.

Validation

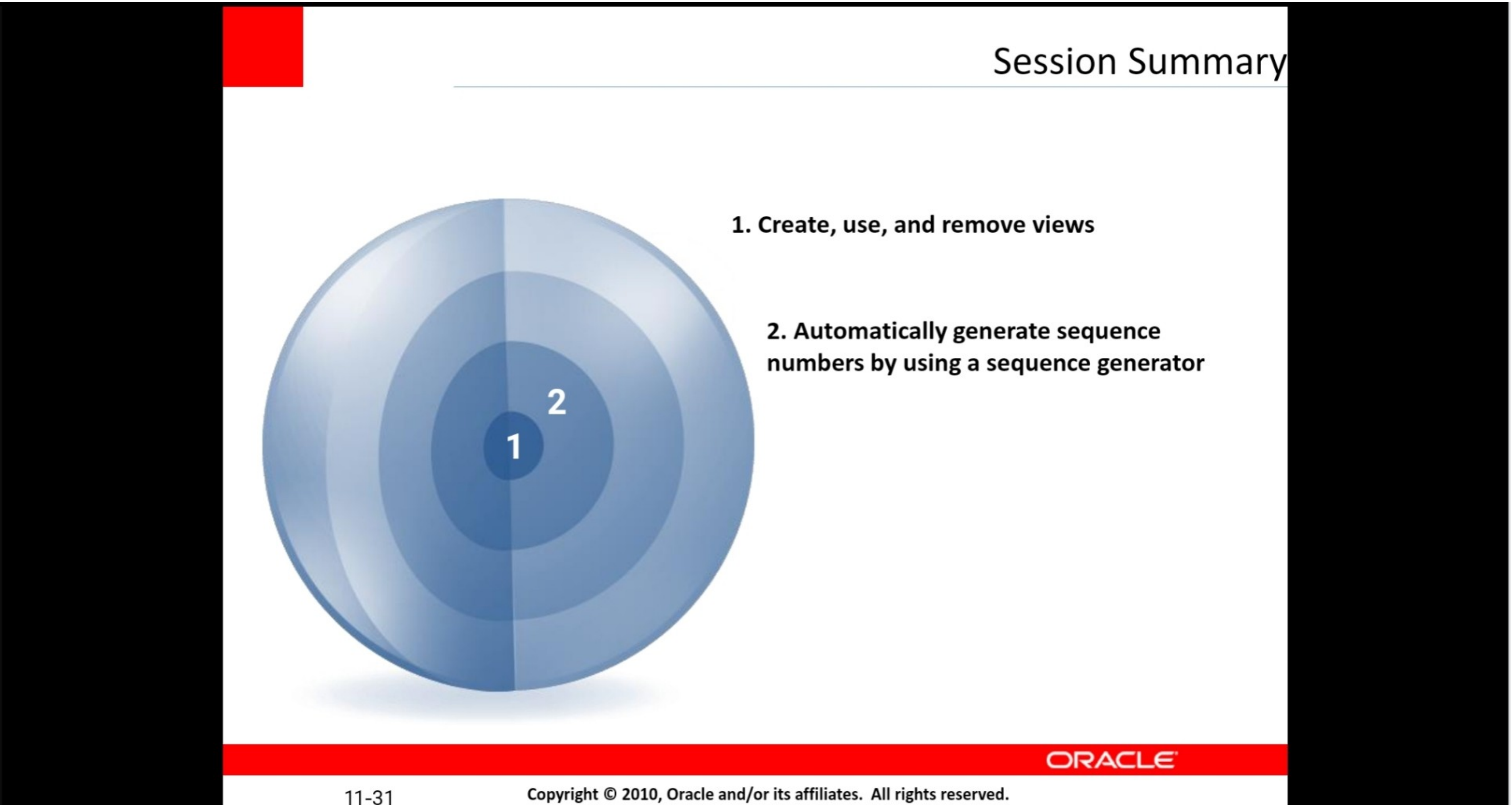
Some validation is performed.

DROP statement

To remove a sequence, use the DROP statement

```
DROP  SEQUENCE  ord_ordid_seq ;
```

```
DROP SEQUENCE dept_deptid_seq succeeded.
```



Session Summary

1. Create, use, and remove views

2. Automatically generate sequence numbers by using a sequence generator