21

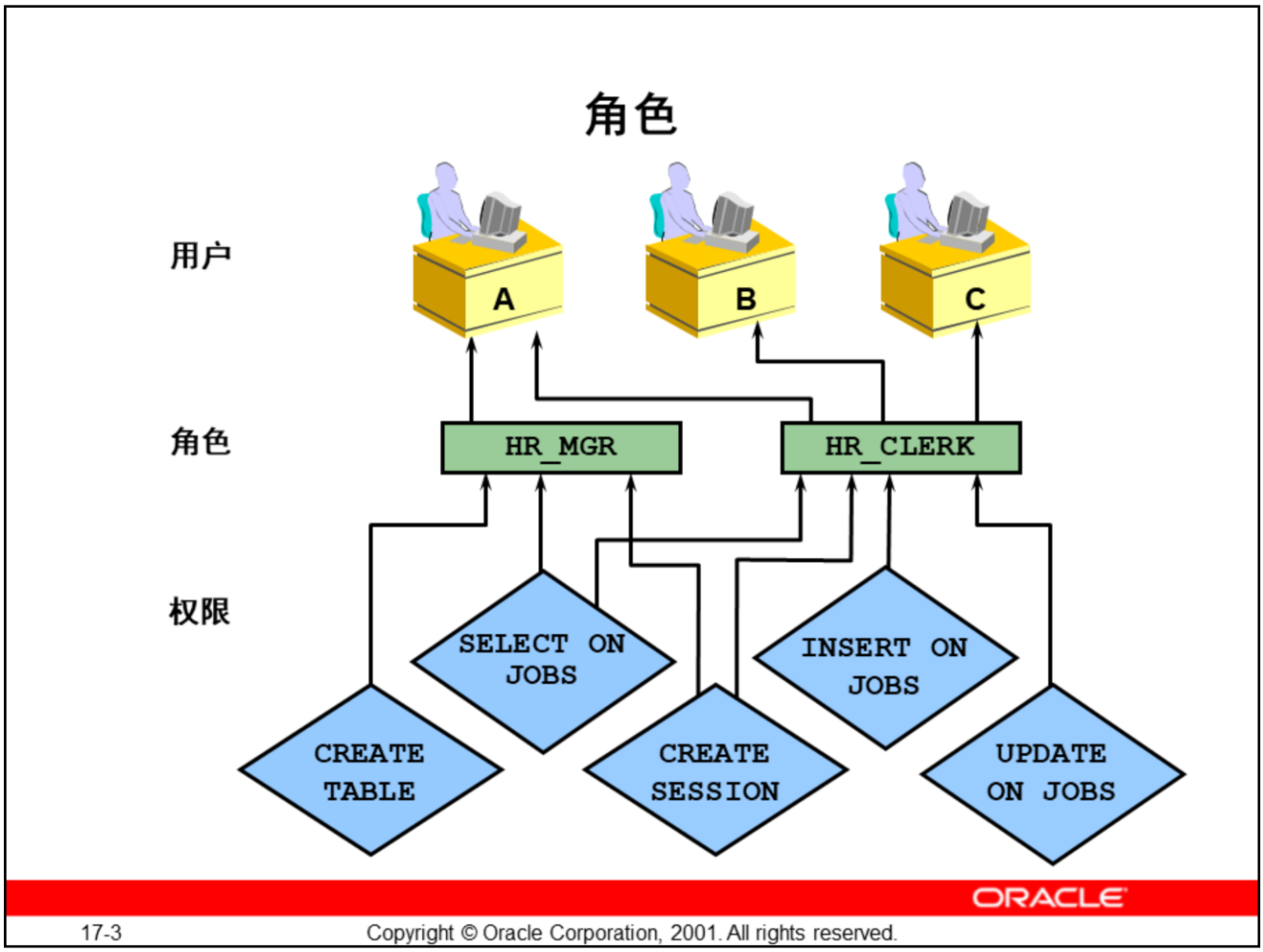
管理角色

目标

完成这一课的学习后，您应该能达到下列目标：

- 创建和修改角色
- 控制角色的可用性
- 删除角色
- 使用预定义角色
- 显示数据字典中的角色信息

ORACLE

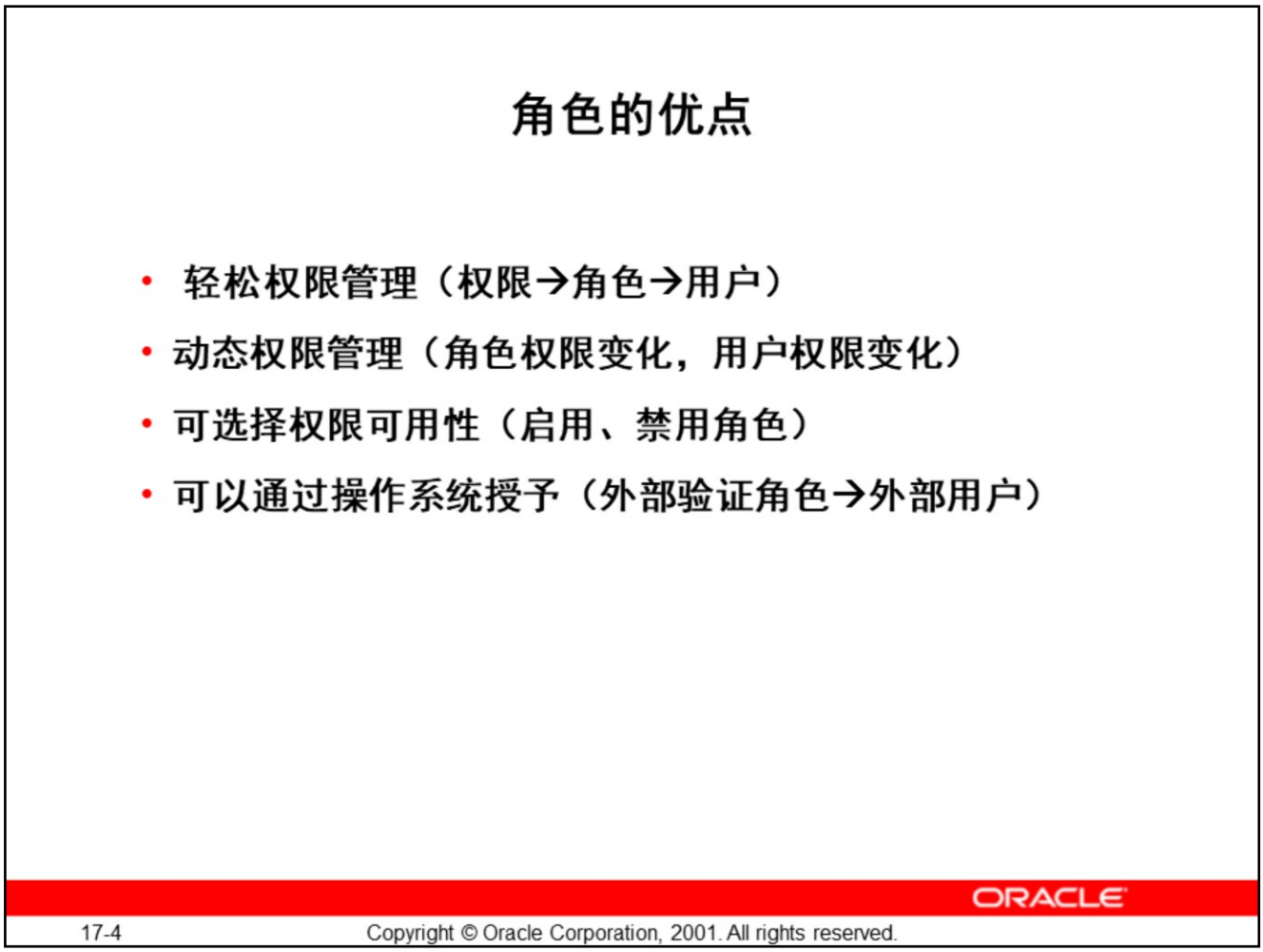


什么是角色？

Oracle 通过角色提供简单且可控制的权限管理。角色是授予用户或其它角色的相关权限的指定组，旨在简化数据库中的权限管理。

角色的特点：

- 可以通过授予和撤消系统权限所用的命令来授予和撤消用户的角色。
- 可以将角色授予任何用户或角色。但是，不能将角色授予它本身，也不能循环授予。
- 角色可以由系统权限和对象权限组成。
- 对于被授予某种角色的每个用户来说，该角色可以启用，也可以禁用。
- 角色可要求通过口令启用。
- 在现有的用户名和角色名中，每个角色名必须唯一。
- 角色不属于任何人，也不存在于任何方案中。
- 在数据字典中存储了有关角色的说明。



角色的优点

轻松权限管理：

使用角色可以简化权限管理。不是将同一组权限授予多个用户，而是将这些权限授予一个角色，然后将该角色授予每个用户。

动态权限管理：

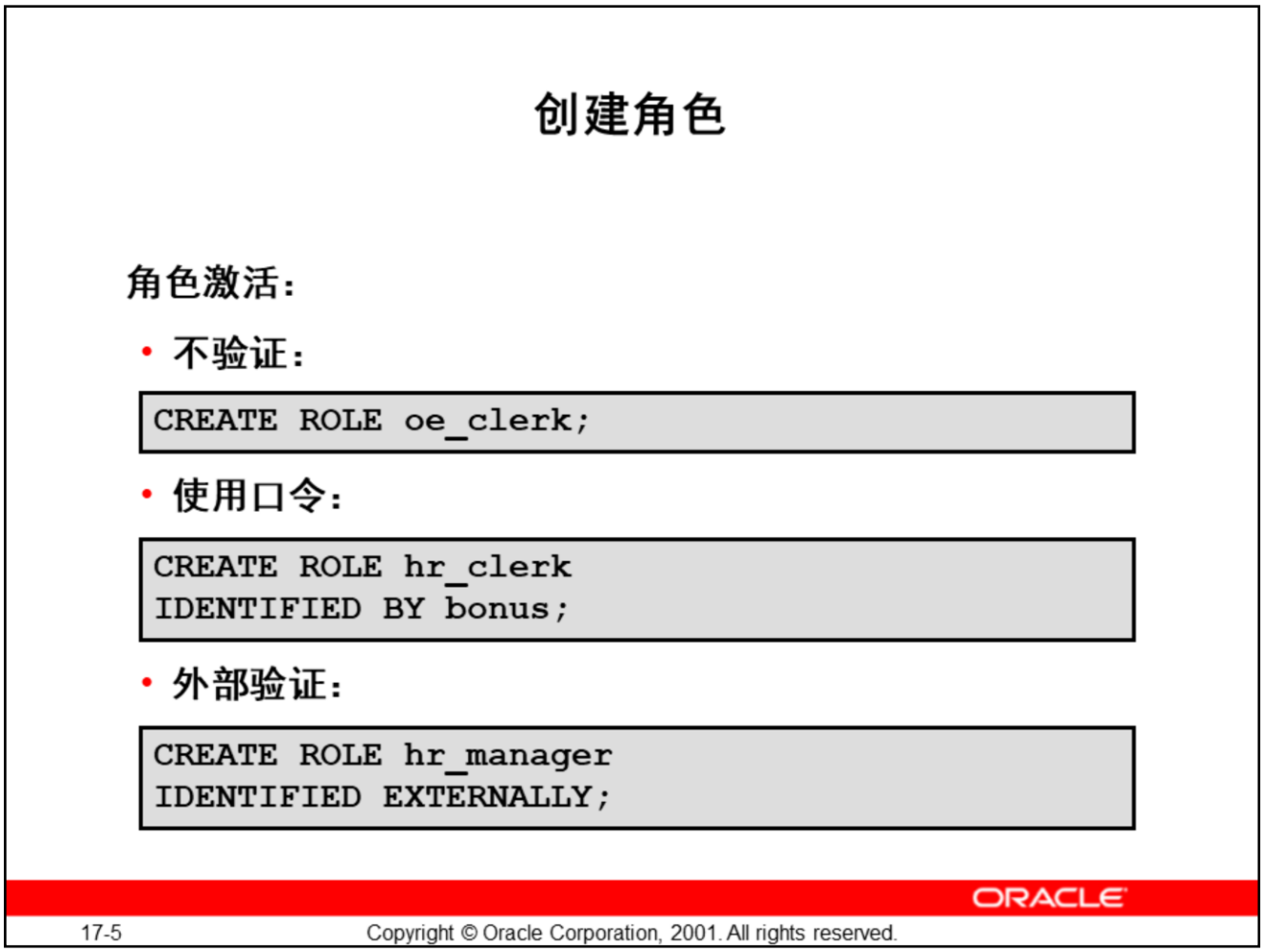
如果修改了与角色关联的权限，被授予该角色的所有用户都将立即自动获得修改后的权限。

可选择权限可用性：

可启用和禁用角色以暂时打开和关闭权限。还可以通过启用角色验证用户是否已被授予该角色。

可以通过操作系统授予：

可以使用操作系统命令或实用程序向数据库中的用户分配角色。



创建角色

使用 CREATE ROLE 语句可创建角色。必须具有 CREATE ROLE 系统权限才能创建角色。创建类型为 NOT IDENTIFIED、IDENTIFIED EXTERNALLY 或 BY password 的角色时，通过 ADMIN 选项为该角色授权。

使用下列命令创建角色：

```
CREATE ROLE role [NOT IDENTIFIED | IDENTIFIED
    {BY password | EXTERNALLY | GLOBALLY | USING package}]
```

其中：

role：是角色的名称

NOT IDENTIFIED：表明启用该角色时，不需要进行验证

IDENTIFIED：表明启用该角色时，需要进行验证

BY password：提供用户在启用角色时必须指定的口令

USING package：创建应用程序角色，该角色只能由使用授权的程序包的应用程序启用

EXTERNALLY：表明在启用该角色之前，用户必须由外部服务（例如操作系统或第三方服务）授权

需在操作系统里定义组ora_<sid>_<role>[_[d][a]]

d:指示<role>部分指定的角色为用户的默认角色

a：指示可以使用with admin option为用户授予<role>部分所指定的角色

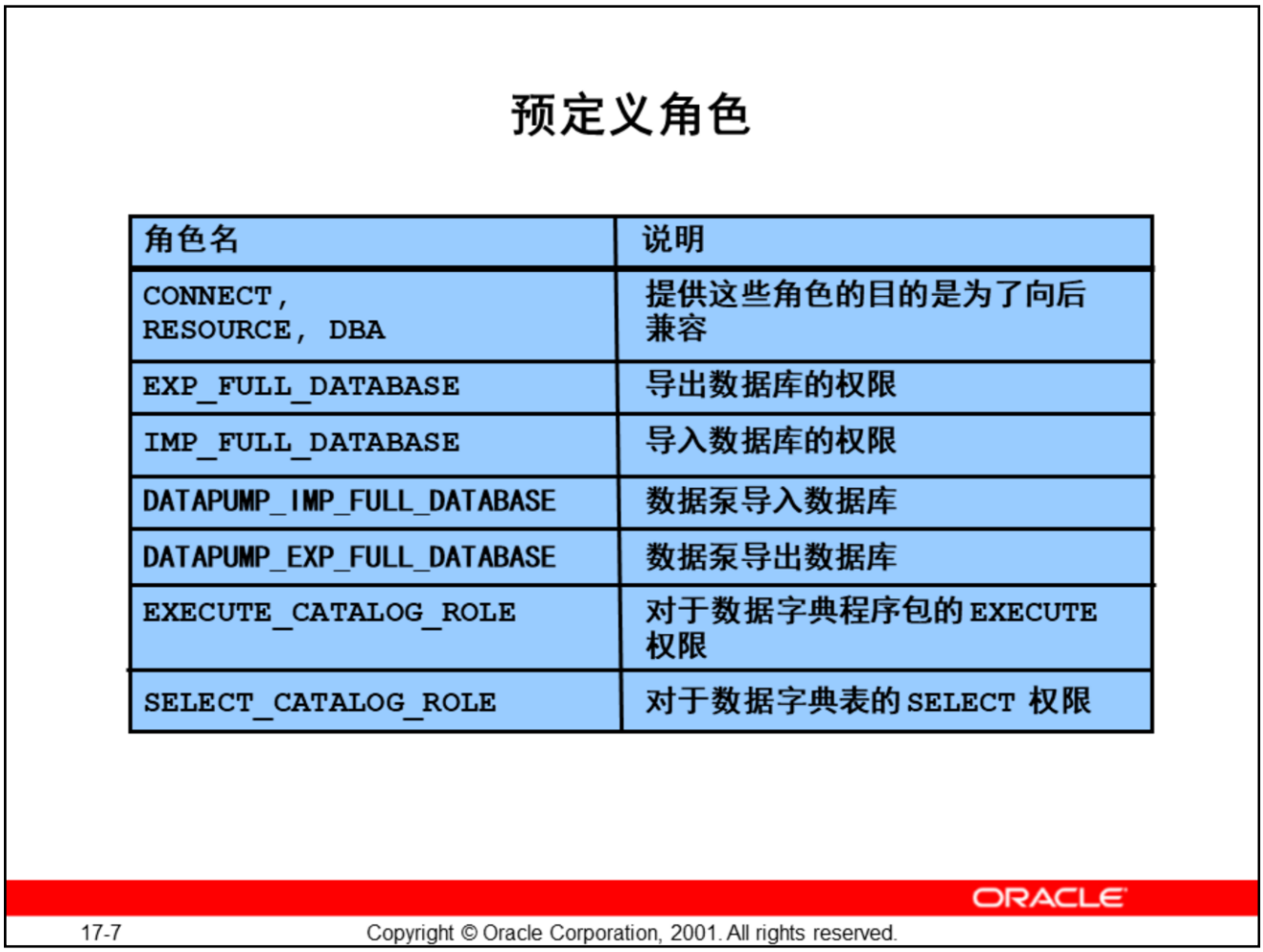
系统参数os_role必须为true，为某用户授予某角色会失败。

GLOBALLY：表明通过 SET ROLE 语句启用角色之前或登录时，必须由企业目录服务授权用户使用该角色

角色的外部验证

- 一般与用户外部验证配合使用
- 系统参数**os_role**必须为**true**，为某用户授予某角色会失败。
- 需在操作系统里定义组**ora_<sid>_<role>[_[d][a]]**
- **d:**指示**<role>**部分指定的角色为用户的默认角色
- **a**：指示可以使用**with admin option**为用户授予**<role>**部分所指定的角色

ORACLE



预定义角色

运行数据库创建脚本时，系统列出的角色是为 Oracle 数据库自动定义的角色。提供 CONNECT、 RESOURCE 和 DBA 角色的目的是为了向后与 Oracle 服务器的早期版本兼容。

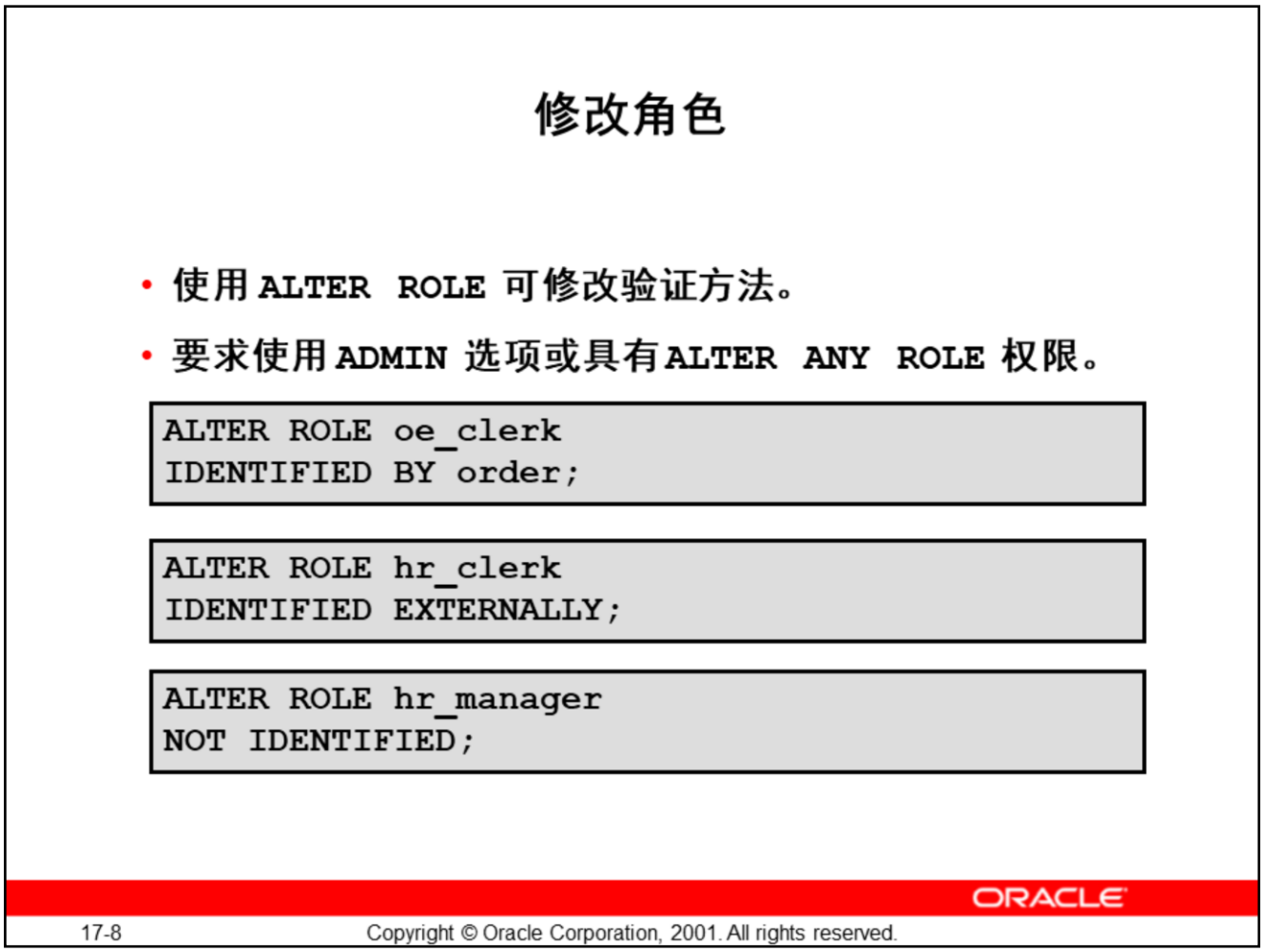
提供了 EXP_FULL_DATABASE 和 IMP_FULL_DATABASE 角色以便于使用导入和导出实用程序。

提供 EXECUTE_CATALOG_ROLE 和 SELECT_CATALOG_ROLE 角色，用于访问数据字典视图和程序包。这些角色可以授予不具有 DBA 角色、但要求访问数据字典中的视图和表的用户。

其它特殊角色：

Oracle 服务器还创建授权您管理数据库的其它角色。在许多操作系统中，这些角色称为 OSOPER 和 OSDBA。它们的名称可能因操作系统而异。

其它角色是由数据库附带的 SQL 脚本定义的。例如，AQ_ADMINISTRATOR_ROLE 提供管理高级排队的权限。AQ_USER_ROLE 已废弃，继续使用的目的主要是为了与版本 8.0 兼容。



修改角色

修改角色时，只能更改其验证方法。而且，您的角色必须通过 ADMIN 选项进行授予，或者您必须具有 ALTER ANY ROLE 系统权限。

使用下列命令修改角色：

```
ALTER ROLE role {NOT IDENTIFIED | IDENTIFIED
 {BY password |USING package| EXTERNALLY | GLOBALLY }};
```

其中：

role：是角色的名称

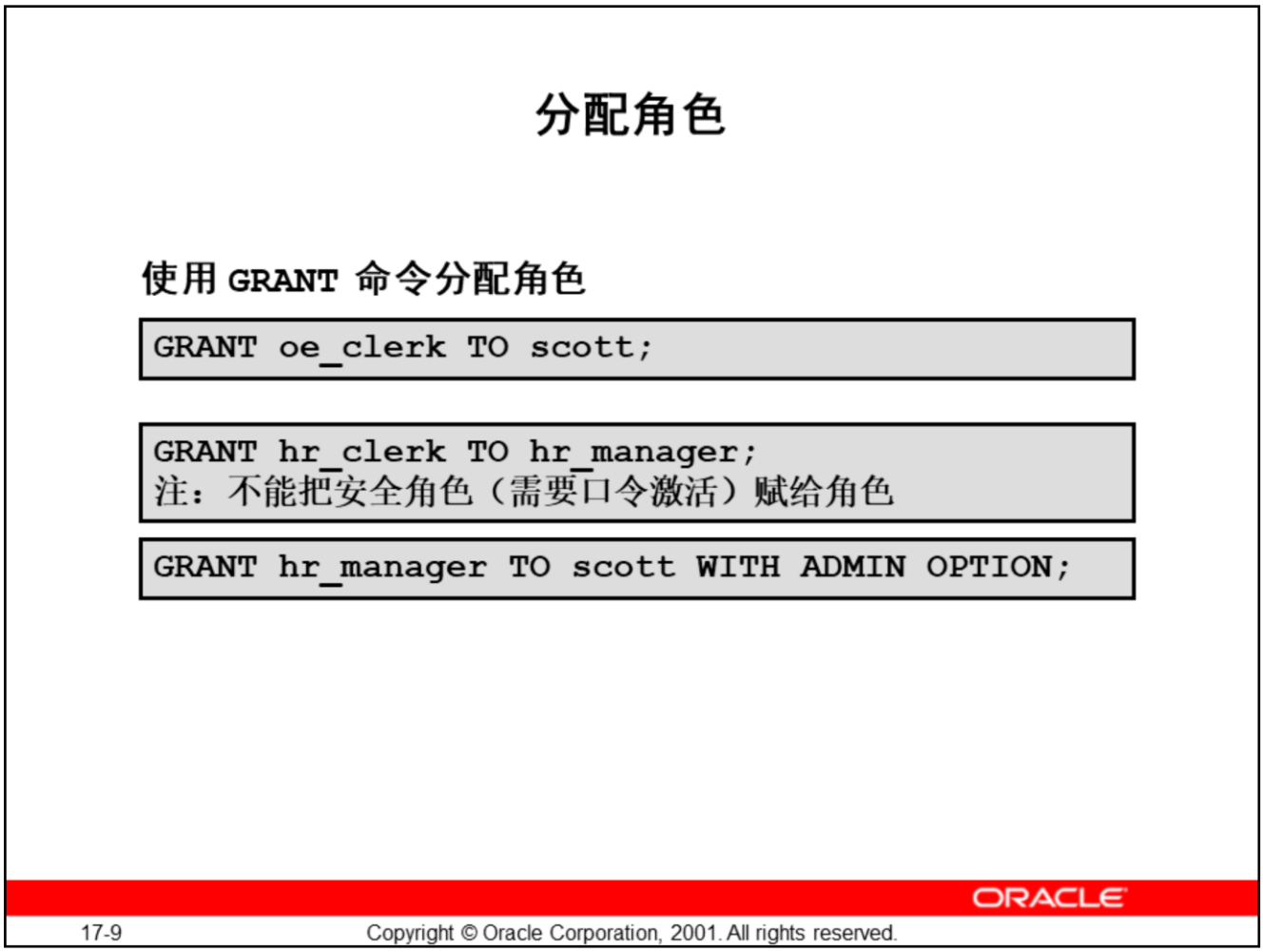
NOT IDENTIFIED：表明启用该角色时，不需要进行验证

IDENTIFIED：表明启用该角色时，需要进行验证

BY password：提供启用角色时所使用的口令

EXTERNALLY：表明在启用该角色之前，用户必须由外部服务（例如操作系统或第三方服务）授权

GLOBALLY：表明通过 SET ROLE 语句启用角色之前或登录时，必须由企业目录服务授权用户使用该角色



分配角色

可使用为用户授予系统权限所用的语法命令为用户授予角色：

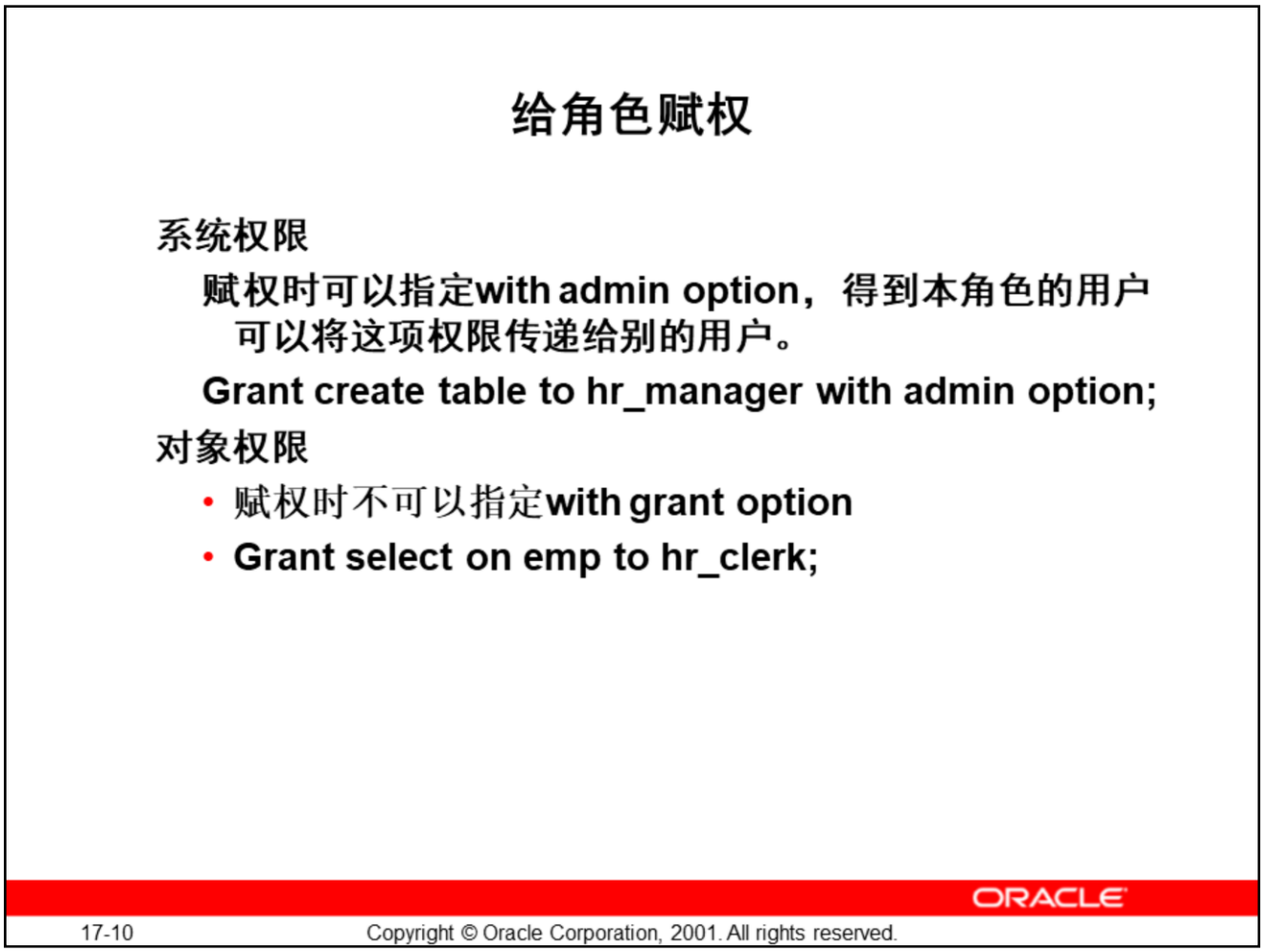
```
GRANT role [, role ]...
TO {user|role|PUBLIC}
  [, {user|role|PUBLIC} ]...
[WITH ADMIN OPTION]
```

其中：

role：是要授予的角色集合

PUBLIC：将角色授予所有用户

WITH ADMIN OPTION：使被授予者能够为其他用户或角色授予角色。（如果使用该选项授予角色，被授予者将能够授予和撤消其他用户的角色，并可改变或删除角色。）



分配角色

可使用为用户授予系统权限所用的语法命令为用户授予角色：

```
GRANT role [, role ]...
TO {user|role|PUBLIC}
 [, {user|role|PUBLIC} ]...
[WITH ADMIN OPTION]
```

其中：

role：是要授予的角色集合

PUBLIC：将角色授予所有用户

WITH ADMIN OPTION：使被授予者能够为其他用户或角色授予角色。（如果使用该选项授予角色，被授予者将能够授予和撤消其他用户的角色，并可改变或删除角色。）

分配角色（续）

缺省情况下，对于创建角色的用户，系统将通过 ADMIN OPTION 为其分配角色。没有通过 ADMIN OPTION 被授予角色的用户需要具有 GRANT ANY ROLE 系统权限才能授予和撤消其它角色。

设置缺省角色

- 可以为一个用户分配多个角色。
- 可以为用户分配一个或多个缺省角色（非口令验证角色）
- 限制用户的缺省角色的数目。

```
ALTER USER scott
     DEFAULT ROLE hr_clerk, oe_clerk;
```

```
ALTER USER scott DEFAULT ROLE ALL;
```

```
ALTER USER scott DEFAULT ROLE ALL EXCEPT
     hr_clerk;
```

```
ALTER USER scott DEFAULT ROLE NONE;
```

ORACLE

缺省角色

可以为一个用户分配多个角色。缺省角色是用户登录时自动启用的角色的子集。缺省情况下，登录时自动启用分配给用户的所有角色，而无需口令。使用 ALTER USER 命令可以限制用户的缺省角色的数目。

DEFAULT ROLE 子句只适用于已通过 GRANT 语句直接授予用户的角色。DEFAULT ROLE 子句不能用来启用下列角色：

- 未授予用户的角色
- 通过其它角色授予的角色
- 由外部服务（如操作系统）管理的角色

使用下列语法分配用户的缺省角色：

```
ALTER USER user DEFAULT ROLE
{role [,role]... | ALL [EXCEPT role [,role]... ] | NONE}
```

其中：

user：是被授予角色的用户的名称

role：是作为用户缺省角色的角色

缺省角色（续）

ALL：除 EXCEPT 子句中列出的角色外，将授予用户的所有角色都设置为缺省角色（这是缺省设置。）

EXCEPT：表明后随角色不应包括在缺省角色中

NONE：授予用户的任何角色都不作为缺省角色（用户登录时拥有的唯一权限是直接分配给该用户的权限。）

由于角色必须被授予后才有可能作为缺省角色，所以不能使用 CREATE USER 命令设置缺省角色。

应用程序角色

- 应用程序角色只能由授权的 PL/SQL 程序包启用。
- USING package 子句用来创建应用程序角色。

```
CREATE ROLE admin_role
IDENTIFIED USING hr.package_emp;
```

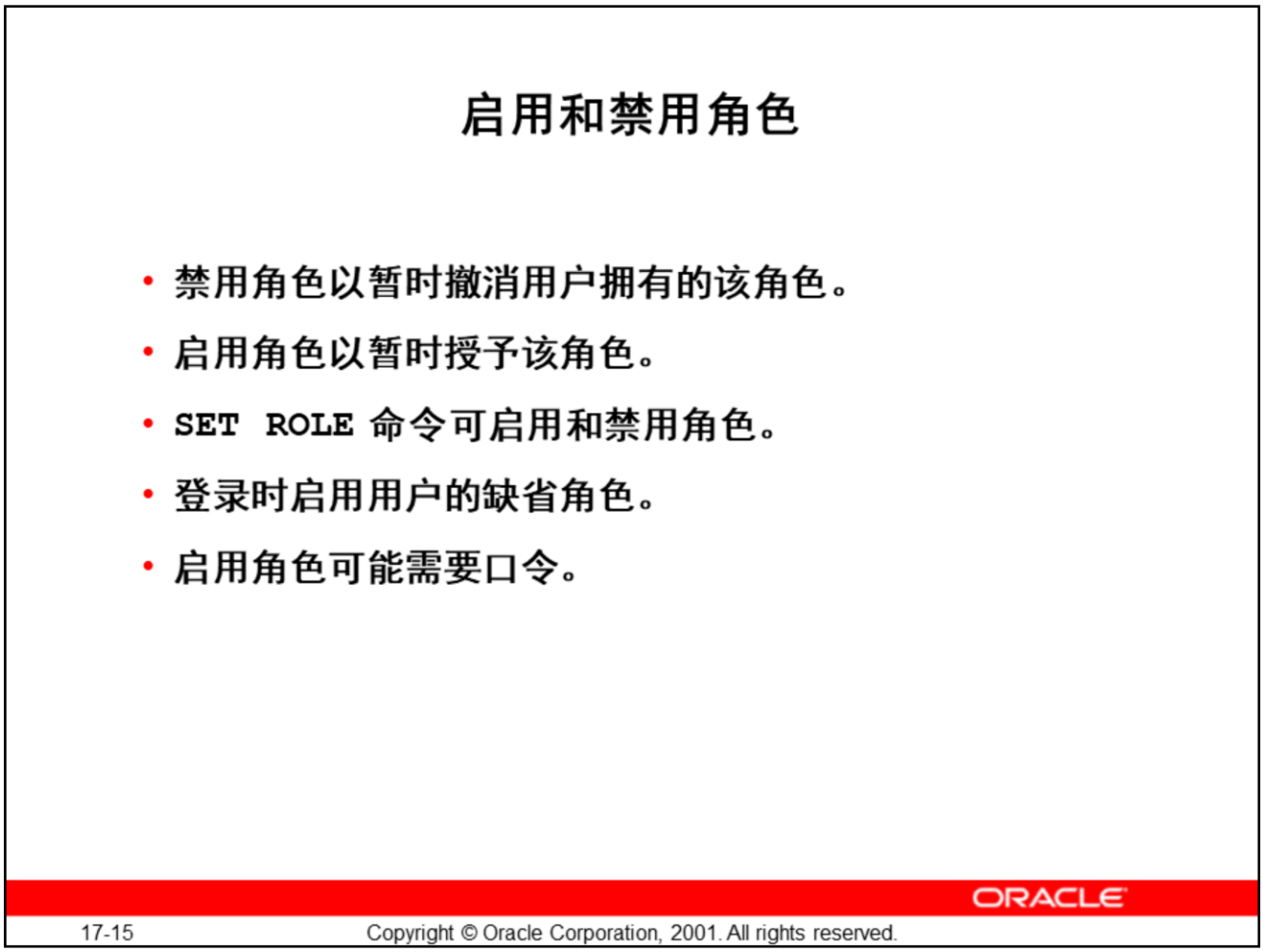
ORACLE

应用程序角色

CREATE ROLE 语句中的 USING package 子句用来创建应用程序角色。应用程序角色只能由使用授权的 PL/SQL 程序包的应用程序启用。应用程序开发人员无需在应用程序内嵌入口令来保护角色。他们可创建应用程序角色并指定授权哪个 PL/SQL 程序包可以启用该角色。

```
SQL> CREATE ROLE admin_role IDENTIFIED USING hr.package_emp;
```

本例中，admin_role 是一个应用程序角色，只有在 hr.package_emp PL/SQL 程序包内定义的模块才能启用该角色。



启用和禁用角色

启用或禁用角色可暂时激活和不激活与角色关联的权限。必须首先为用户授予角色，然后才能启用该角色。

启用角色时，用户可以使用授予该角色的权限。如果禁用角色，用户将不能使用与该角色关联的权限，除非将该权限直接授予用户或授予为该用户启用的另一个角色。角色的启用针对的是一个会话。在下一个会话中，用户的活动角色将恢复为缺省角色。

指定要启用的角色：

SET ROLE 命令和 DBMS_SESSION.SET_ROLE 过程将启用包含在命令中的全部角色，并禁用所有其它角色。可以通过任何允许使用 PL/SQL 命令的工具或程序启用角色；但不能在存储过程中启用角色。

可以使用 ALTER USER...DEFAULT ROLE 命令指出用户登录时将启用的角色。所有其它角色都将被禁用。

启用角色时可能需要口令。SET ROLE 命令中必须包含口令才能启用角色。分配给用户的缺省角色不需要口令，这些角色同没有口令的角色一样在登录时启用。

启用和禁用角色（续）

限制：

不能在存储过程中启用角色，因为该操作可能会改变安全域（权限集），安全域允许首先调用存储过程。因此，在 PL/SQL 中，可以在匿名块和应用程序过程（如 Oracle Forms 过程）中启用和禁用角色，但不能在存储过程中执行该操作。

如果存储过程包含了 SET ROLE 命令，运行时会产生下列错误：

```
ORA-06565: cannot execute SET ROLE from within stored procedure
```

启用和禁用角色

SET ROLE 命令关闭授予用户的任何其它角色。

```
SET ROLE {role [ IDENTIFIED BY password ]
         [, role [ IDENTIFIED BY password ]]...
         | ALL [ EXCEPT role [, role ] ...]
         | NONE }
```

其中：

role：是角色的名称

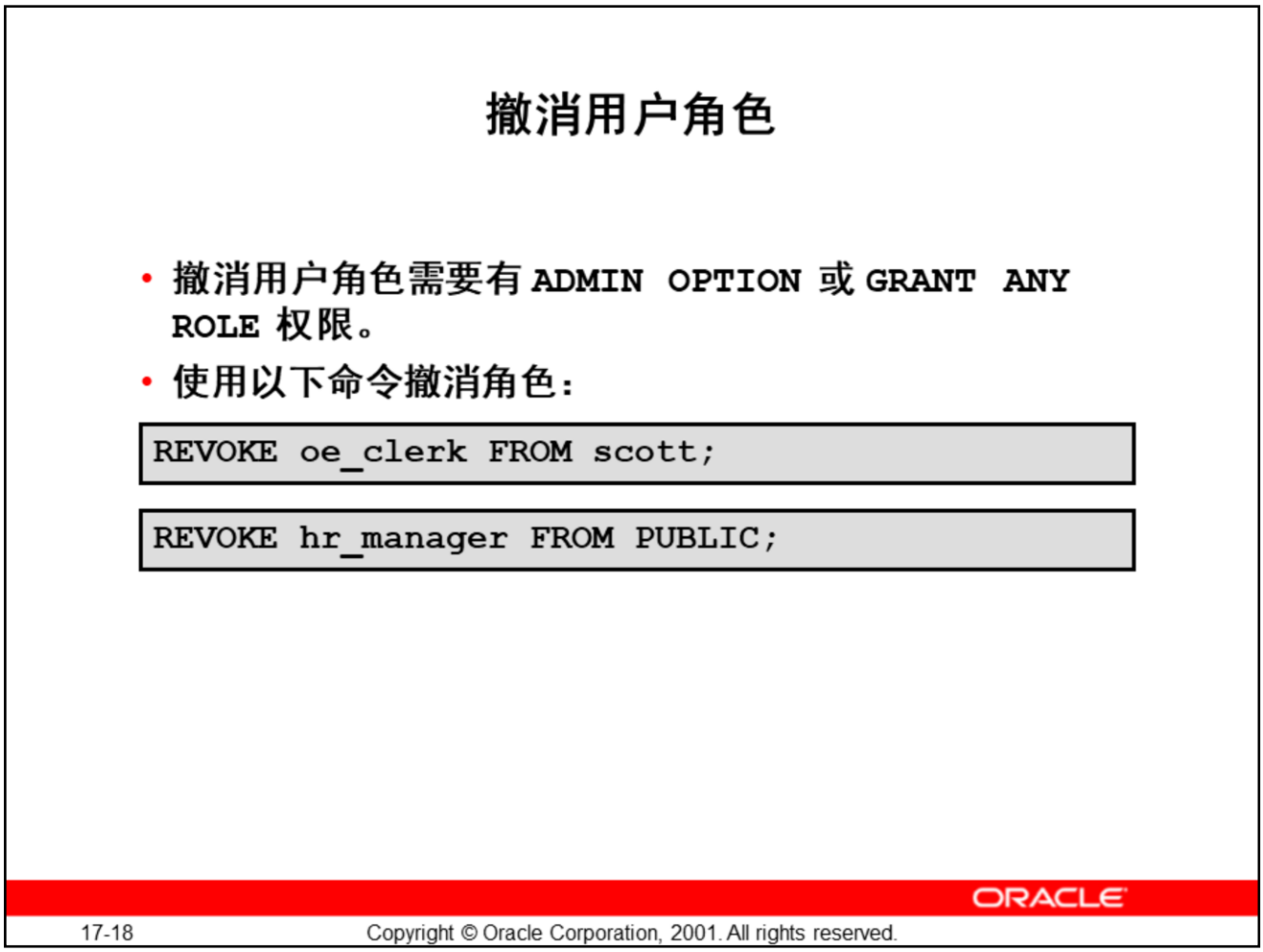
IDENTIFIED BY password：提供启用角色时所需的口令

ALL：除了 EXCEPT 子句中列出的角色外，启用授予当前用户的全部角色（不能使用该选项启用带口令的角色。）

EXCEPT role：不启用这些角色

NONE：禁用当前会话的全部角色（只有直接授予用户的权限是活动的。）

只有在启用的每个角色都没有口令时，不带 EXCEPT 子句的 ALL 选项才有效。



撤消用户角色

使用 SQL 语句 REVOKE 可撤消用户角色。通过 ADMIN 选项获取角色的任何用户都可撤消任何其他数据库用户或角色的角色。此外，具有 GRANT ANY ROLE 权限的用户也可以撤消任何角色。

```
REVOKE role [, role ]
    FROM {user|role|PUBLIC}
     [, {user|role|PUBLIC} ]
```

其中

role：是要撤消的角色或从其撤消角色的角色

user：要撤消其系统权限或角色的用户

PUBLIC：撤消所有用户的权限或角色

删除角色

- 删除角色：
  - 删除授予所有用户和角色的角色
  - 删除数据库角色
- 需要 ADMIN OPTION 或 DROP ANY ROLE 权限
- 使用以下命令删除角色：

```
DROP ROLE hr_manager;
```

ORACLE

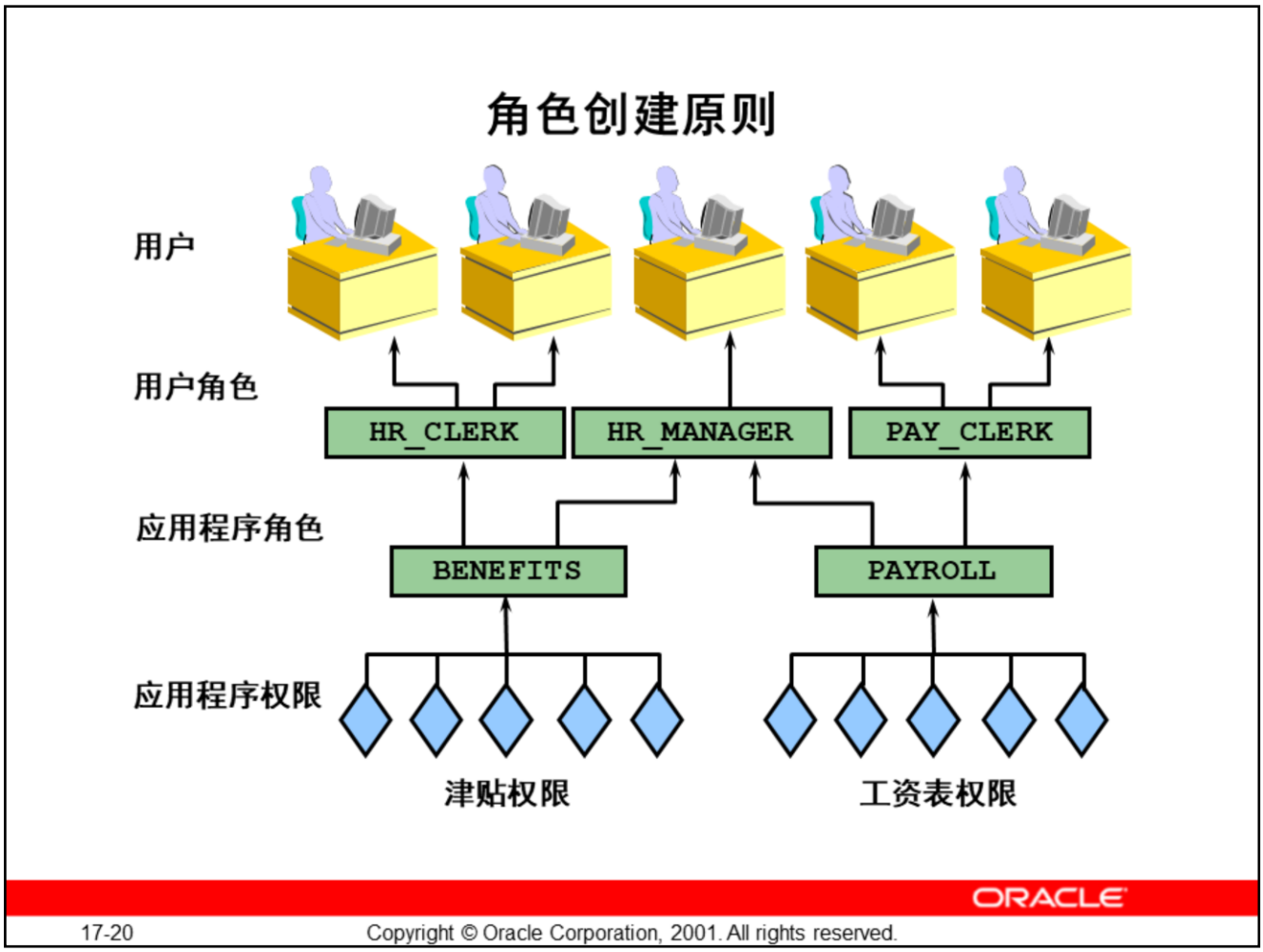
删除角色

使用下列语法从数据库中删除角色：

```
DROP ROLE role
```

删除角色时，Oracle 服务器从所有用户和被授予该角色的角色中及数据库中撤消该角色。

必须通过 ADMIN OPTION 被授予了角色或具有 DROP ANY ROLE 系统权限才能删除角色。

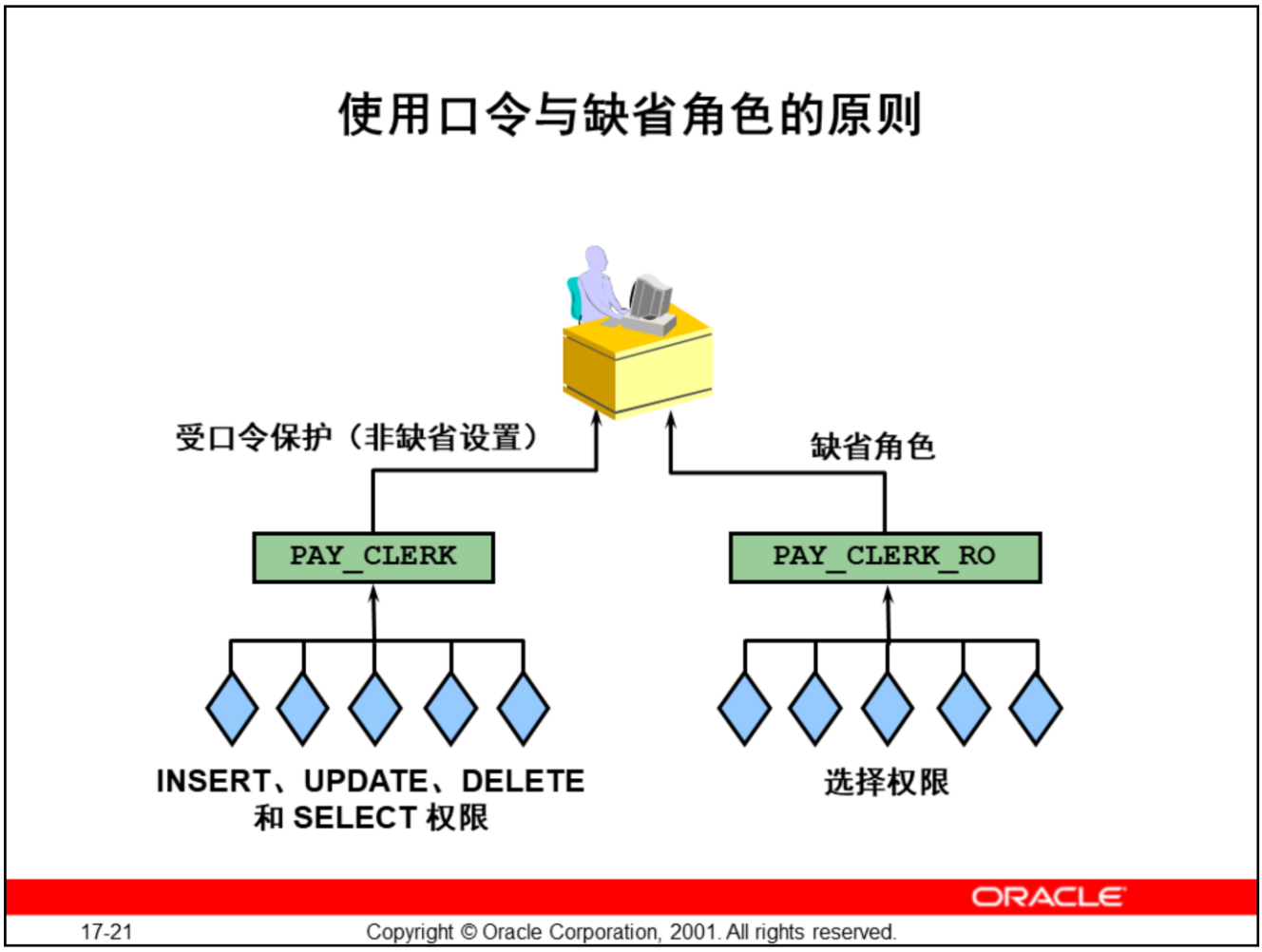


角色创建原则

因为角色包含执行任务所需的权限，因此，角色名通常就是应用程序任务或职称。幻灯片中的示例就使用应用程序任务与职称作为角色名。请按下列步骤创建、分配和授予用户角色：

1. 为每个应用程序任务创建一个角色。应用程序角色名对应于应用程序任务，如 PAYROLL。
2. 为应用程序角色分配执行任务所需的权限。
3. 为每种用户类型创建一个角色。用户角色名对应于职称，如 PAY_CLERK。
4. 将应用程序角色授予用户角色。
5. 将用户角色授予用户。

如果应用程序的修改要求使用新的权限执行工资表任务，则 DBA 只需向应用程序角色 PAYROLL 分配新的权限。当前正在执行该任务的所有用户都将接收到新的权限。



使用口令与缺省角色的原则

启用角色时口令提供了额外的安全级。例如，在启用 PAY_CLERK 角色时，应用程序可能要求用户输入口令，因为可以使用该角色发出支票。

通过使用口令，只能通过应用程序来启用角色。在幻灯片的示例中对这种技术做了说明。

- DBA 授予了用户两个角色：PAY_CLERK 和 PAY_CLERK_RO。
- PAY_CLERK 已被授予执行工资表职员功能所需的全部权限。
- PAY_CLERK_RO（RO 表示只读）仅被授予了在表上执行工资表职员功能所需的 SELECT 权限。

- 用户可以登录到 SQL* Plus 进行查询，但是不能修改任何数据，这是由于 PAY_CLERK 不是一个缺省角色，用户并不知道 PAY_CLERK 的口令。
- 当用户登录到工资表应用程序时，该程序提供口令以启用 PAY_CLERK。口令已编入程序中，系统不会提示用户输入口令。

获取角色信息

可以通过查询以下视图来获取有关角色的信息：

- DBA_ROLES：数据库中存在的所有角色
- DBA_ROLE_PRIVS：授予用户和角色的角色
- ROLE_ROLE_PRIVS：授予角色的角色
- DBA_SYS_PRIVS：授予用户和角色的系统权限
- DBA_TAB_PRIVS：授予用户和角色的对象权限
- DBA_COL_PRIVS：授予用户和角色的字段级对象权限
- ROLE_SYS_PRIVS：当前用户所具有的角色的系统权限
  ROLE_TAB_PRIVS：当前用户所具有的角色的对象权限
- SESSION_ROLES：当前用户启用的角色

ORACLE

查询角色信息

许多数据字典视图不仅包含授予用户的权限信息，还包含有关角色是否需要口令的信息。

```
SQL> SELECT role, password_required
  2   FROM dba_roles;

ROLE                           PASSWORD
------------------------------ -----------
CONNECT                        NO
RESOURCE                       NO
DBA                            NO
SELECT_CATALOG_ROLE            NO
EXECUTE_CATALOG_ROLE           NO
IMP_FULL_DATABASE              NO
EXP_FULL_DATABASE              NO
SALES_CLERK                    YES
HR_CLERK                       EXTERNAL
```

小结

在这一课中，您应该能够掌握：

- 创建角色
- 为角色分配权限
- 为用户或角色分配角色
- 设置缺省角色

ORACLE