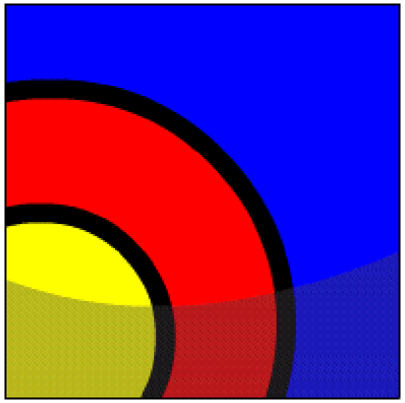
Understanding Dependencies

ORACLE Academy

What Will I Learn?

In this lesson, you will learn to:

- Describe the implications of procedural dependencies
- Contrast dependent objects and referenced objects
- View dependency information in the dictionary views
- Use the UTLDTREE script to create the objects required to display dependencies
- Use the IDEPTREE and DEPTREE views to display dependencies
- Describe when automatic recompilation occurs
- List how to minimize dependency failures



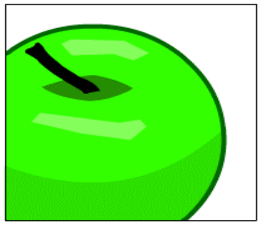
Why Learn It?

A PL/SQL subprogram can execute correctly only if the objects it references exist and are valid. These objects can be tables, views, other PL/SQL subprograms, and other kinds of database object.

So what happens if a referenced object is altered or dropped?

This lesson introduces you to object dependencies and implicit and explicit recompilation of invalid objects.

Tell Me / Show Me



Understanding Dependencies

Dependent and referenced objects:

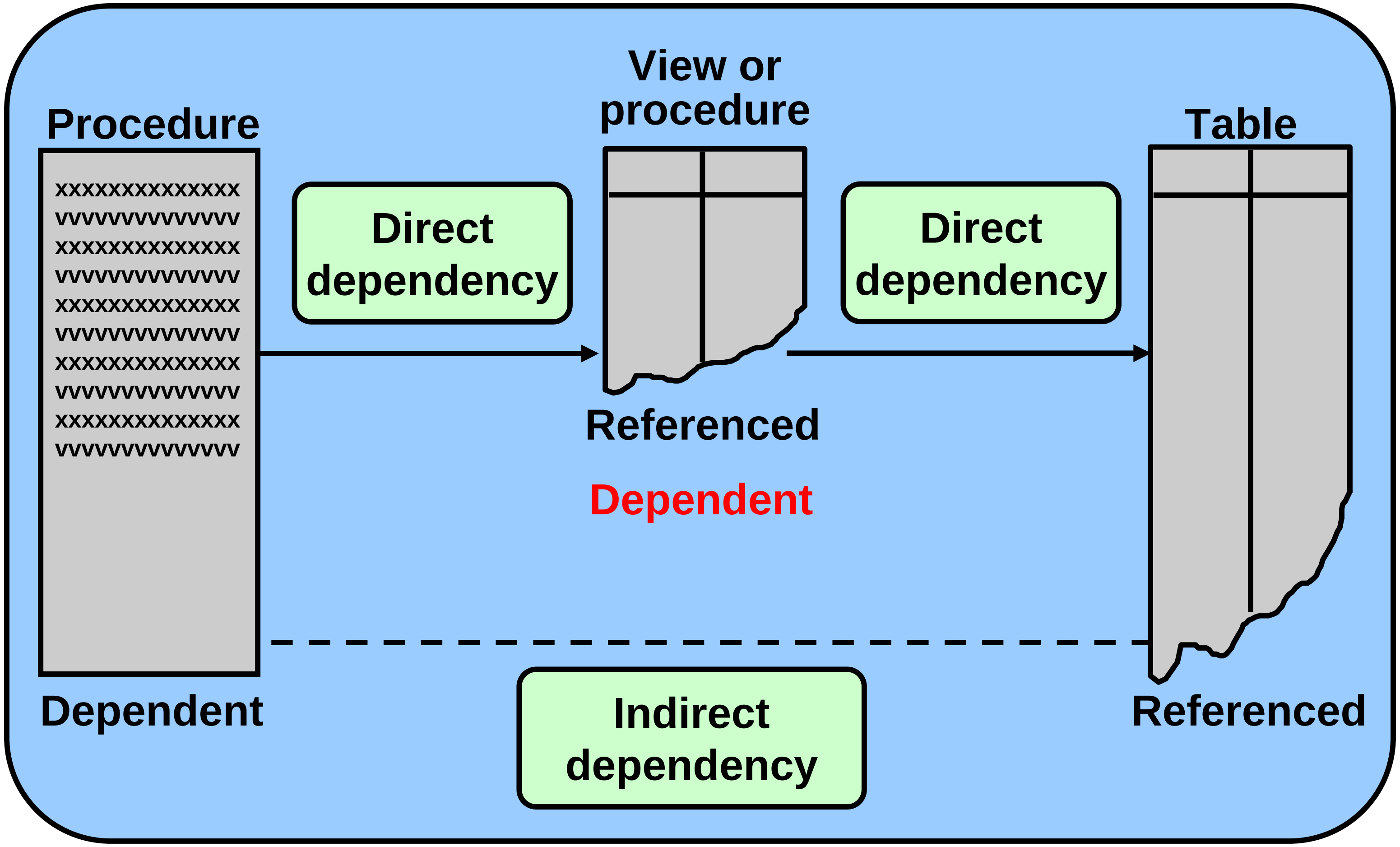
- Some objects reference other objects as part of their definitions. For example, a stored procedure could contain a SELECT statement that selects columns from a table. For this reason, the stored procedure is called a dependent object, whereas the table is called a referenced object.

Dependency issues:

- If you alter the definition of a referenced object, dependent objects might not continue to work properly. For example, if the table definition is changed, the procedure might not continue to work without error.
- The Oracle server automatically records dependencies among objects. To manage dependencies, all schema objects have a status (valid or invalid) that is recorded in the Data Dictionary, and you can view the status in the USER_OBJECTS Data Dictionary view.

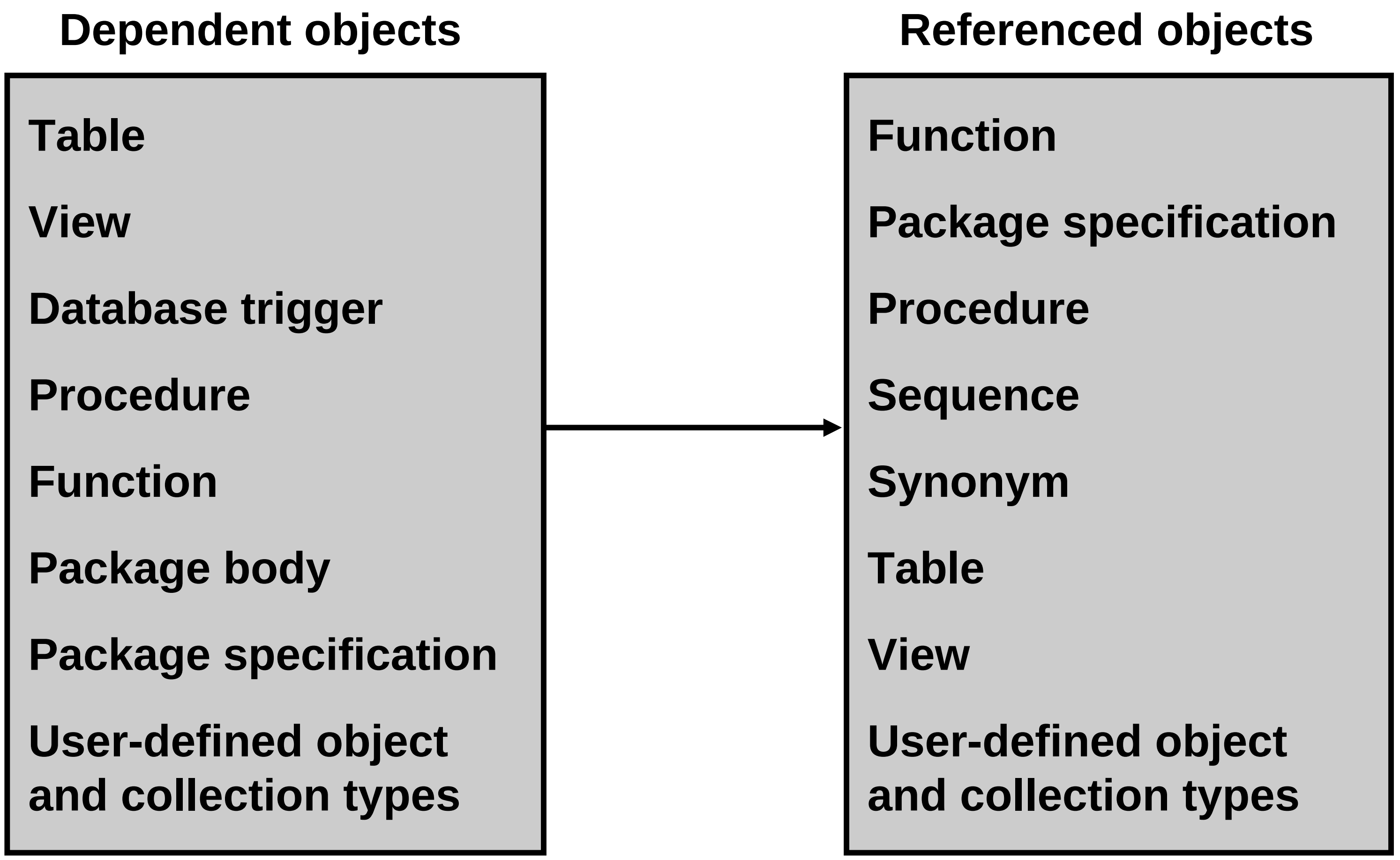
Tell Me / Show Me

Dependencies



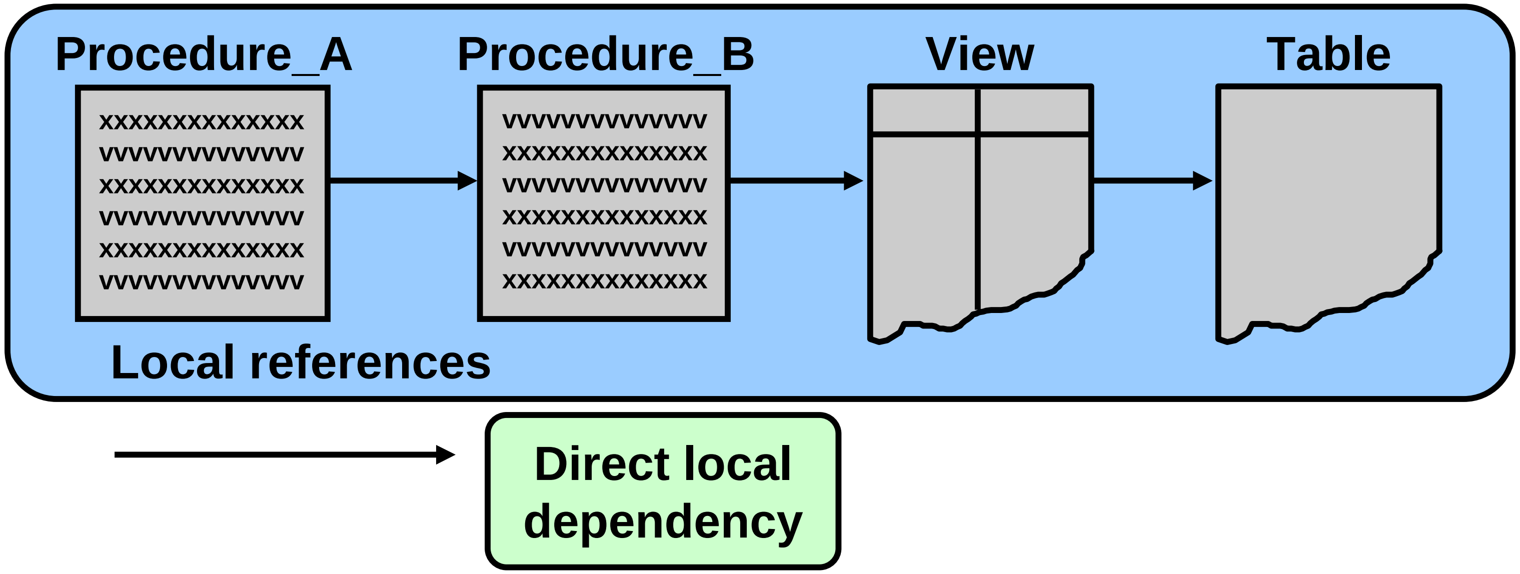
Tell Me / Show Me

Dependencies Summarized



Tell Me / Show Me

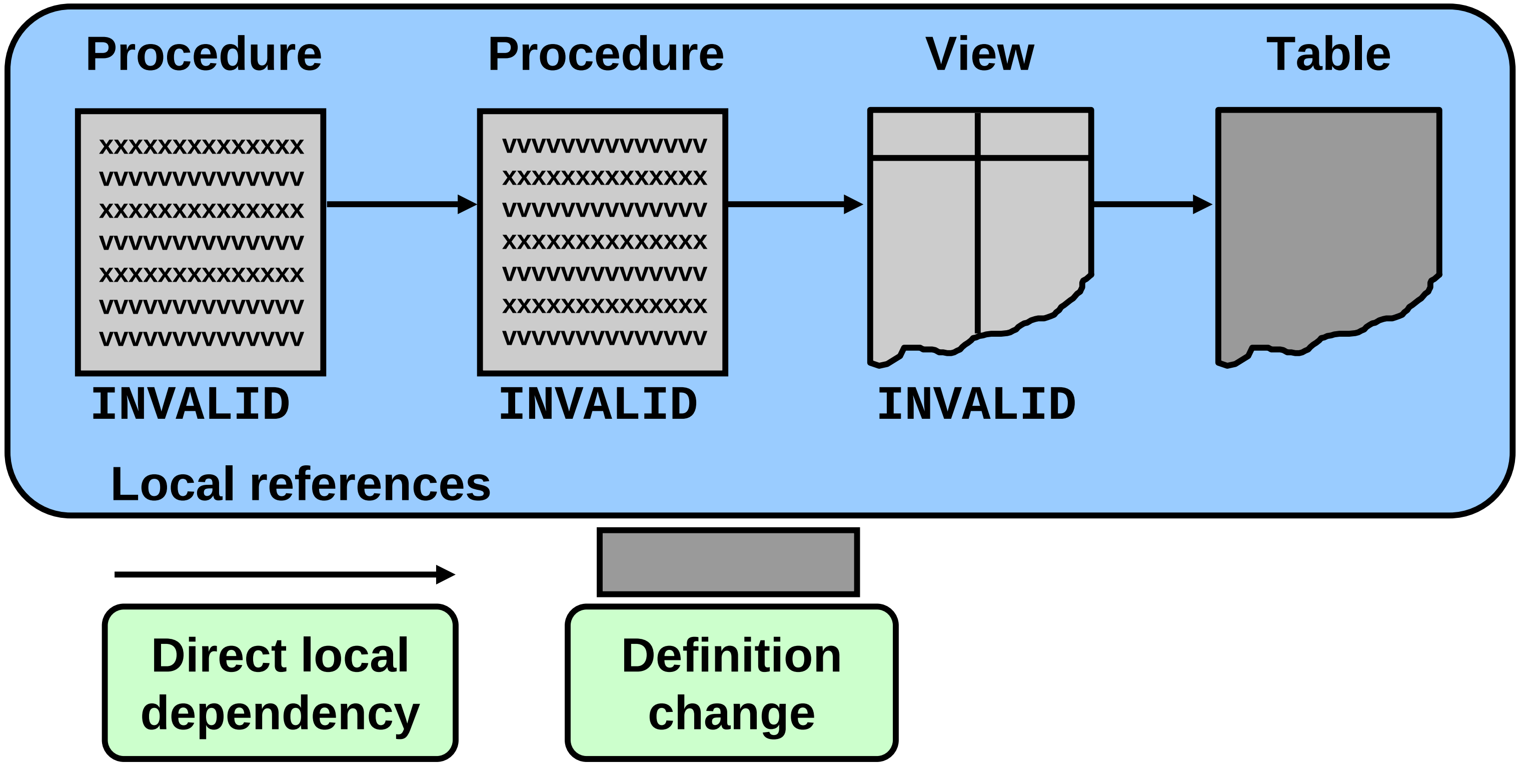
Local Dependencies



In the case of local dependencies, the objects are on the same node in the same database. The Oracle server automatically manages all local dependencies, using the database's internal "depends-on" table. When a referenced object is modified, the dependent objects are invalidated. The next time an invalidated object is called, the Oracle server automatically tries to recompile it.

Tell Me / Show Me

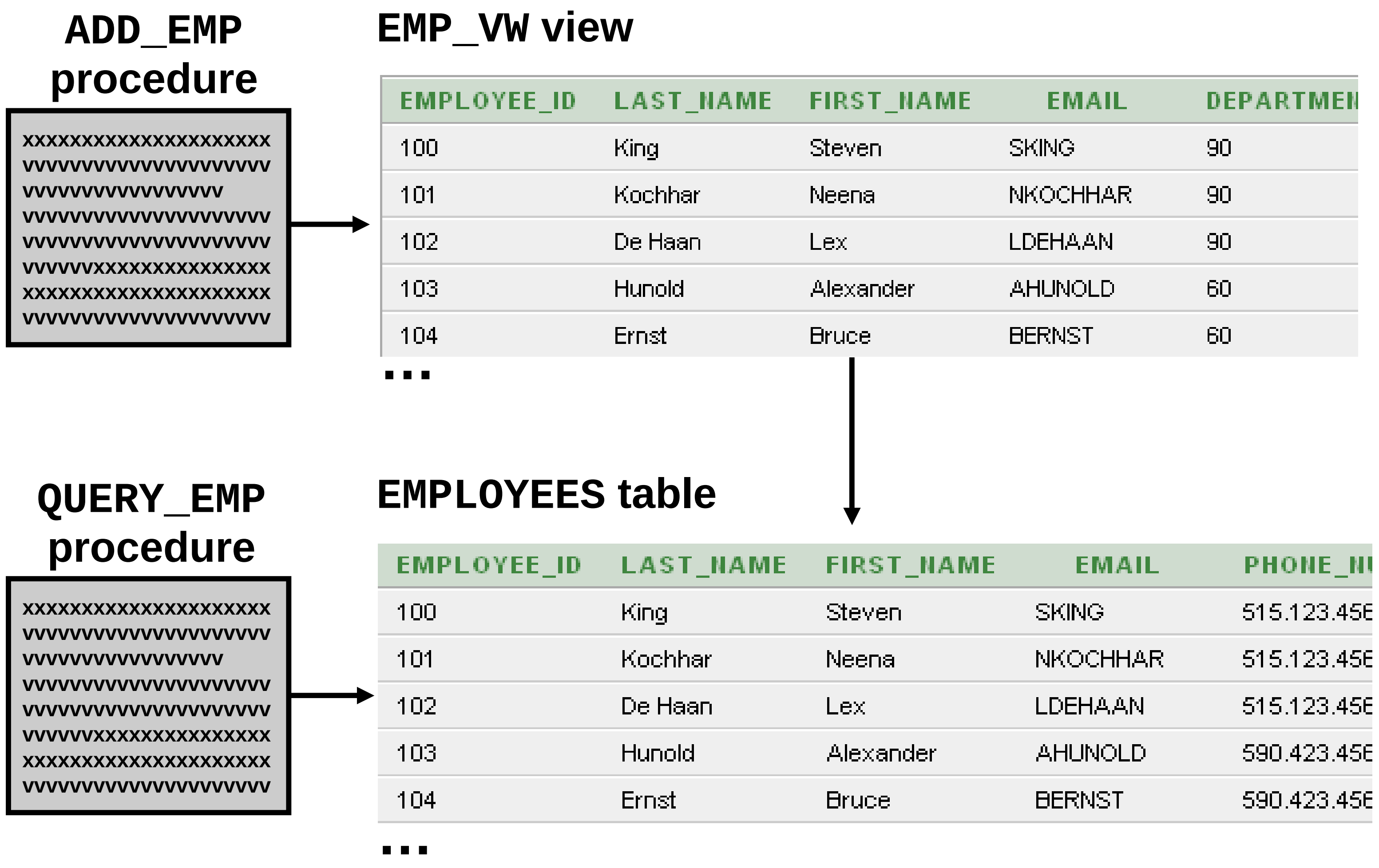
Local Dependencies (continued)



The Oracle server implicitly attempts to recompile any `INVALID` object when the object is next called.

Tell Me / Show Me

A Scenario of Local Dependencies



EMPLOYEE_ID	LAST_NAME	FIRST_NAME	EMAIL	DEPARTMEN
100	King	Steven	SKING	90
101	Kochhar	Neena	NKOCHHAR	90
102	De Haan	Lex	LDEHAAN	90
103	Hunold	Alexander	AHUNOLD	60
104	Ernst	Bruce	BERNST	60

...

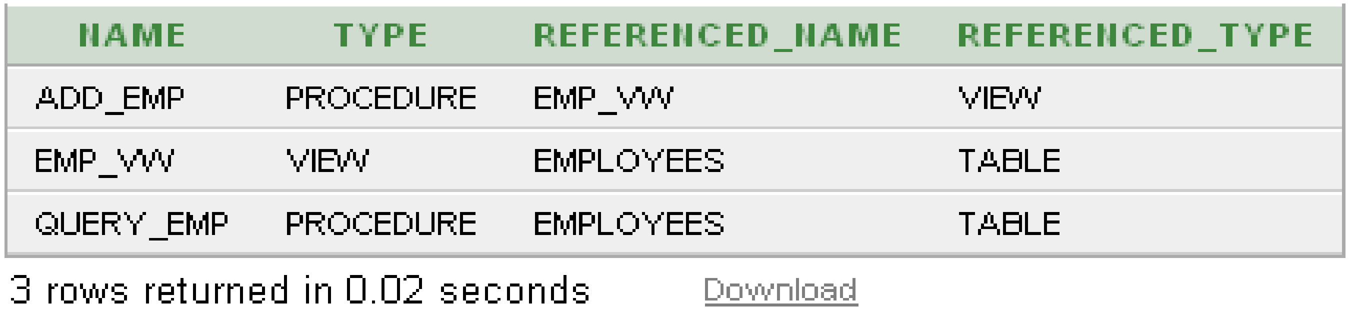
EMPLOYEE_ID	LAST_NAME	FIRST_NAME	EMAIL	PHONE_NU
100	King	Steven	SKING	515.123.456
101	Kochhar	Neena	NKOCHHAR	515.123.456
102	De Haan	Lex	LDEHAAN	515.123.456
103	Hunold	Alexander	AHUNOLD	590.423.456
104	Ernst	Bruce	BERNST	590.423.456

...

Tell Me / Show Me

Displaying Direct Dependencies by Using USER_DEPENDENCIES

```
SELECT name, type, referenced_name, referenced_type
  FROM   user_dependencies
  WHERE  referenced_name IN ('EMPLOYEES','EMP_VW' );
```



NAME	TYPE	REFERENCED_NAME	REFERENCED_TYPE
ADD_EMP	PROCEDURE	EMP_VW	VIEW
EMP_VW	VIEW	EMPLOYEES	TABLE
QUERY_EMP	PROCEDURE	EMPLOYEES	TABLE

3 rows returned in 0.02 seconds

You can also view direct dependencies in Application Express: SQL Workshop -> Object Browser -> choose an object, then click the Dependencies tab.

Tell Me / Show Me

Displaying Direct and Indirect Dependencies

Run the script `utldtree.sql` that creates the objects that enable you to display the direct and indirect dependencies.

This script creates four objects:

- A table `deptree_temptab` to hold dependency data
- A procedure `deptree_fill` to populate the table
- Two views `deptree` and `ideptree` to select and format dependency data from the populated table.

For each object whose dependencies you want to see:

1. Execute the `DEPTREE_FILL` procedure.

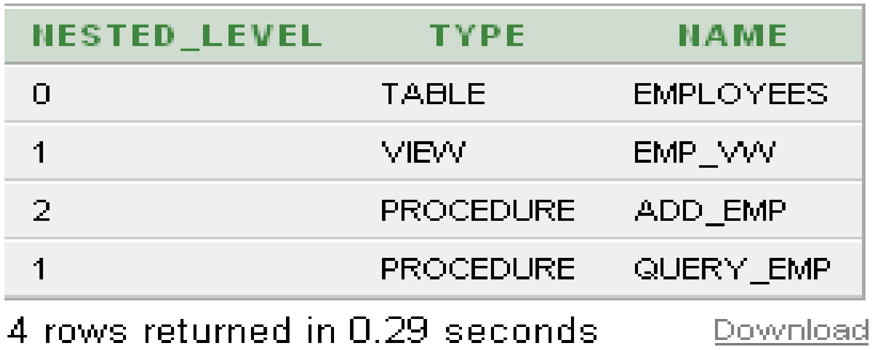
```
BEGIN  deptree_fill('TABLE','SCOTT','EMPLOYEES');  END;
```

Tell Me / Show Me

Displaying Direct and Indirect Dependencies (continued)

2. Display the dependency data using the `DEPTREE` view

```
SELECT   nested_level, type, name
  FROM      deptree
  ORDER BY seq#;
```



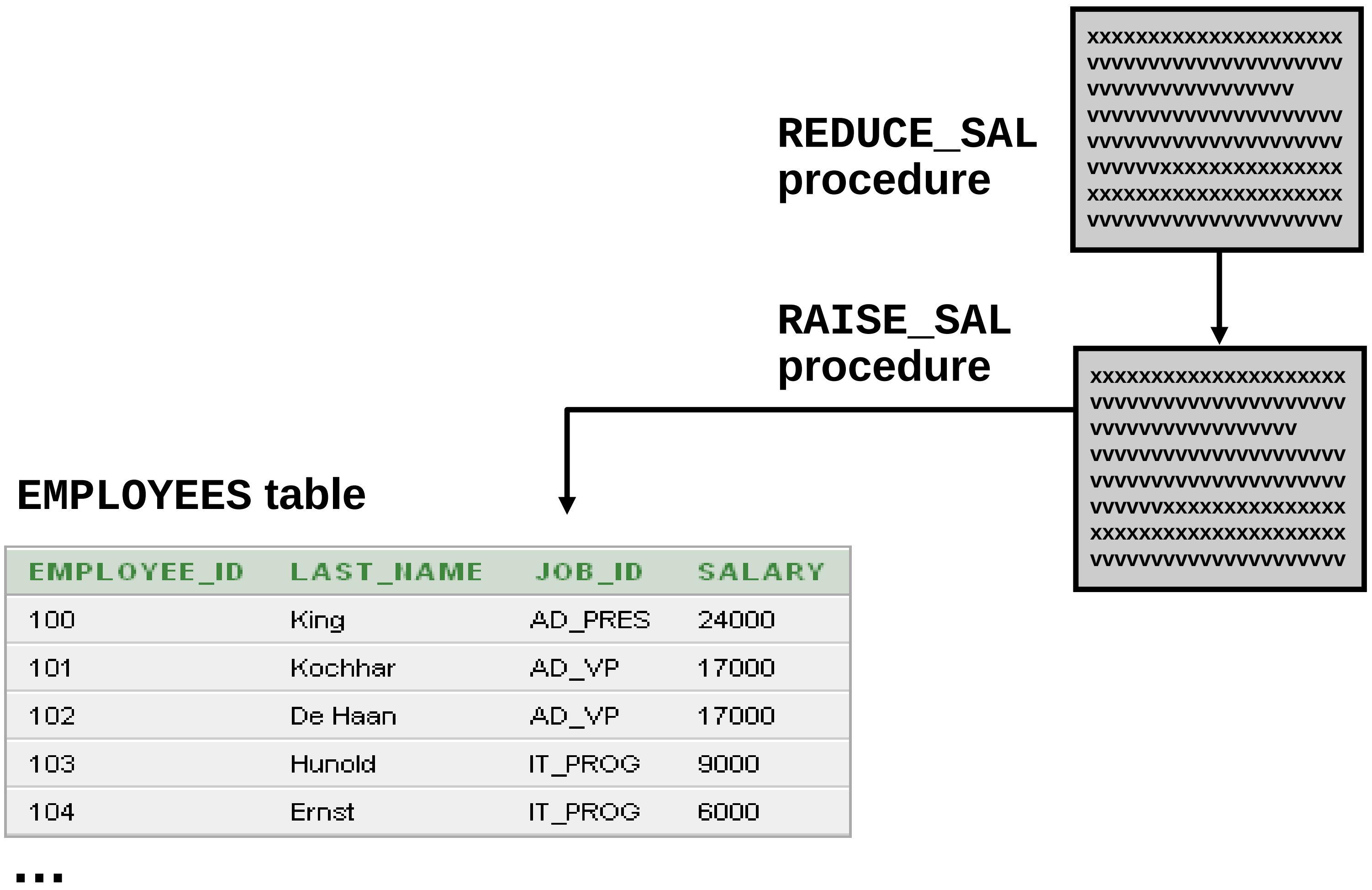
NESTED_LEVEL	TYPE	NAME
0	TABLE	EMPLOYEES
1	VIEW	EMP_VW
2	PROCEDURE	ADD_EMP
1	PROCEDURE	QUERY_EMP

4 rows returned in 0.29 seconds   Download

In this example, `ADD_EMP` is directly dependent on `EMP_VW`, which in turn is directly dependent on `EMPLOYEES` (look at the `NESTED-LEVEL` column).

Tell Me / Show Me

Another Scenario of Local Dependencies



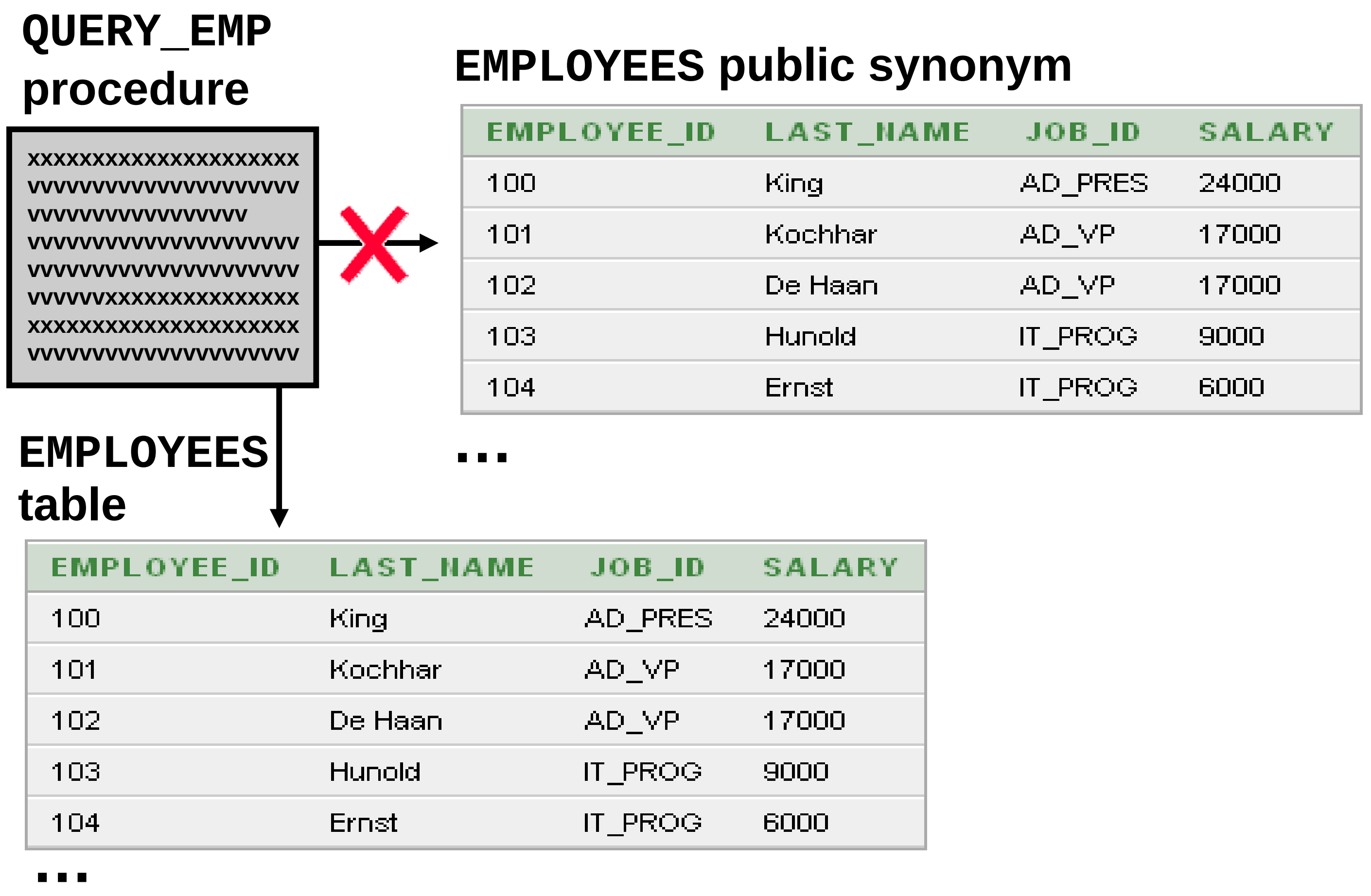
EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
103	Hunold	IT_PROG	9000
104	Ernst	IT_PROG	6000

...

ORACLE Academy

Tell Me / Show Me

A Third Scenario of Local Naming Dependencies



EMPLOYEES public synonym

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
103	Hunold	IT_PROG	9000
104	Ernst	IT_PROG	6000

...

EMPLOYEES table

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
103	Hunold	IT_PROG	9000
104	Ernst	IT_PROG	6000

...

Tell Me / Show Me

Recompiling a PL/SQL Program Unit

Recompilation:

- Is handled automatically through implicit run-time recompilation
- Is handled through explicit recompilation with the ALTER statement:

```
ALTER PROCEDURE [SCHEMA.]procedure_name COMPILE;
```

```
ALTER FUNCTION  [SCHEMA.]function_name  COMPILE;
```

```
ALTER PACKAGE [SCHEMA.]package_name
   COMPILE [PACKAGE | SPECIFICATION | BODY];
```

```
ALTER TRIGGER trigger_name [COMPILE[DEBUG]];
```

Tell Me / Show Me

Unsuccessful Recompilation

Recompiling dependent procedures and functions is unsuccessful when:

- The referenced object is dropped or renamed
- The data type of the referenced column is changed
- The referenced column is dropped
- A referenced view is replaced by a view with different columns
- The parameter list of a referenced procedure is modified

Tell Me / Show Me

Successful Recompilation

Recompiling dependent procedures and functions is successful if:

- The referenced table has new columns
- The data type of referenced columns has not changed
- A private table is dropped, but a public table that has the same name and structure exists
- The PL/SQL body of a referenced procedure has been modified and recompiled successfully

Tell Me / Show Me

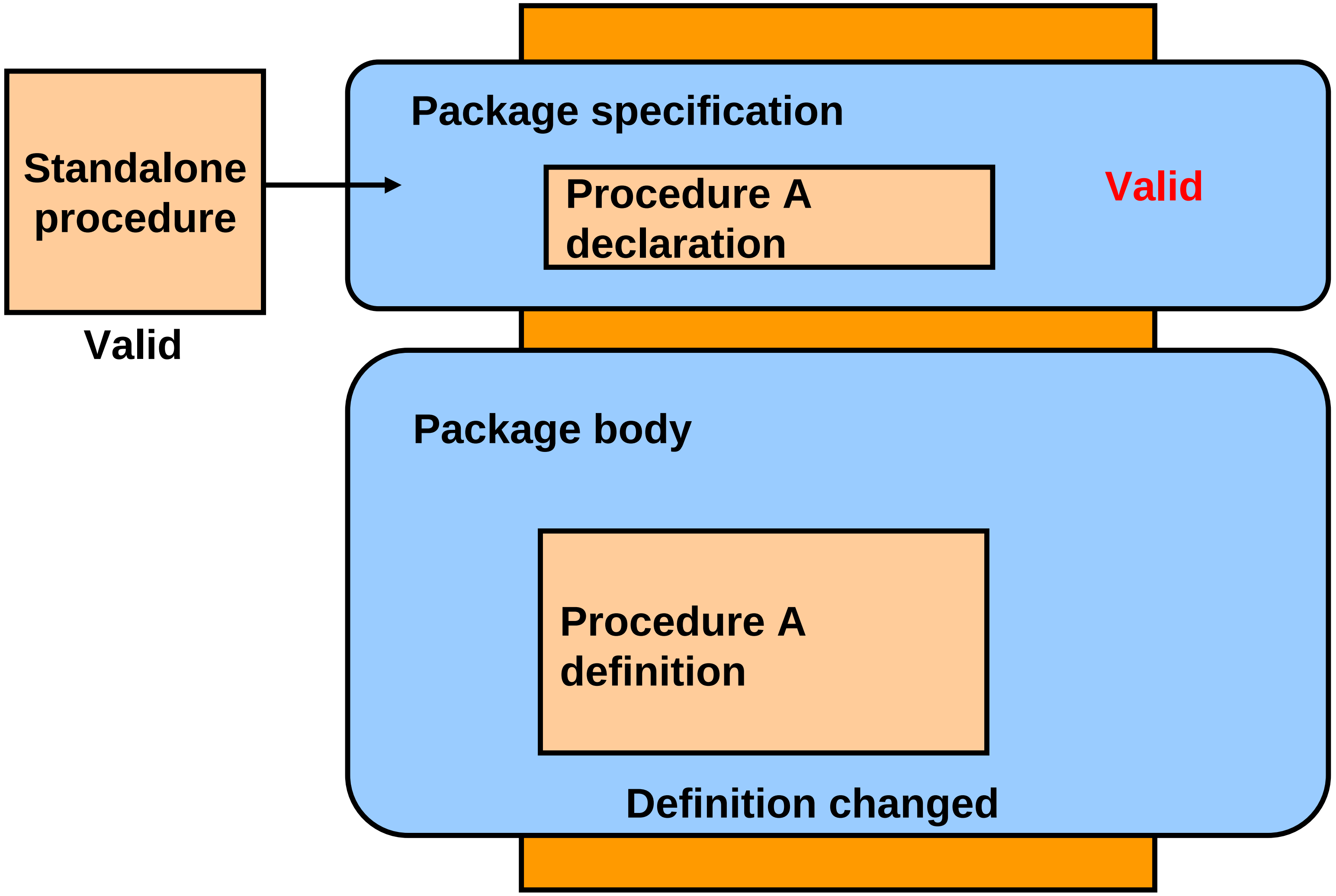
Recompilation of Procedures

Minimize dependency failures by:

- Declaring records with the `%ROWTYPE` attribute
- Declaring variables with the `%TYPE` attribute
- Querying with the `SELECT *` notation
- Including a column list with `INSERT` statements

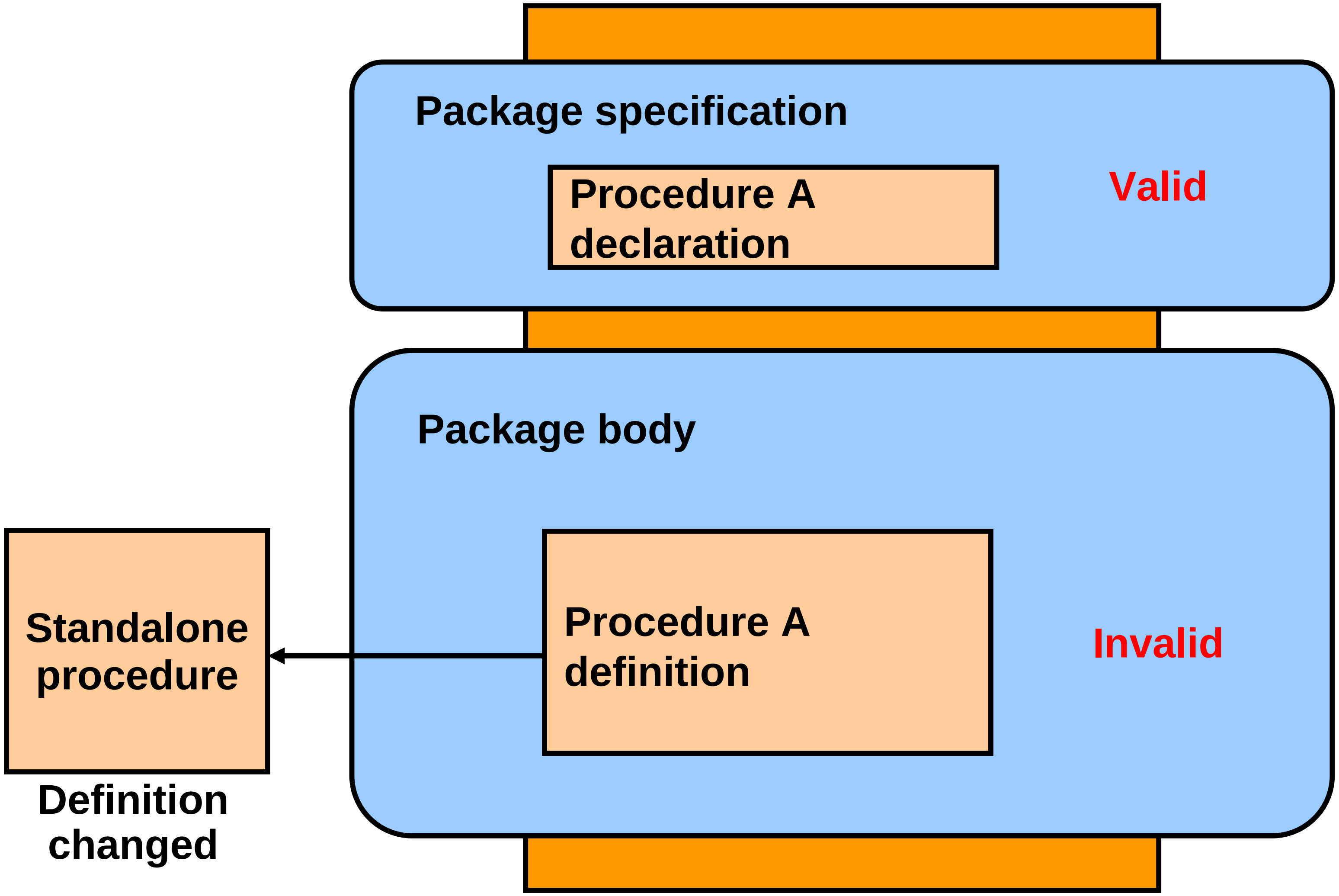
Tell Me / Show Me

Packages and Dependencies



Tell Me / Show Me

Packages and Dependencies (continued)

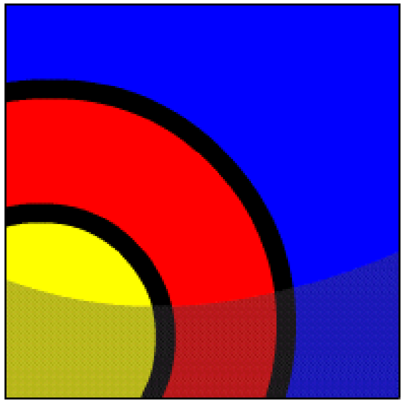


ORACLE Academy

Summary

In this lesson, you learned to:

- Describe the implications of procedural dependencies
- Contrast dependent objects and referenced objects
- View dependency information in the dictionary views
- Use the UTLDTREE script to create the objects required to display dependencies
- Use the IDEPTREE and DEPTREE views to display dependencies
- Describe when automatic recompilation occurs
- List how to minimize dependency failures



ORACLE Academy

Try It / Solve It

The exercises in this lesson cover the following topics:

- Describing the implications of procedural dependencies
- Describing dependent objects and referenced objects
- Viewing dependency information in the dictionary views
- Using the UTLDTREE script
- Using the IDEPTREE and DEPTREE views
- Listing how to minimize dependency failures