

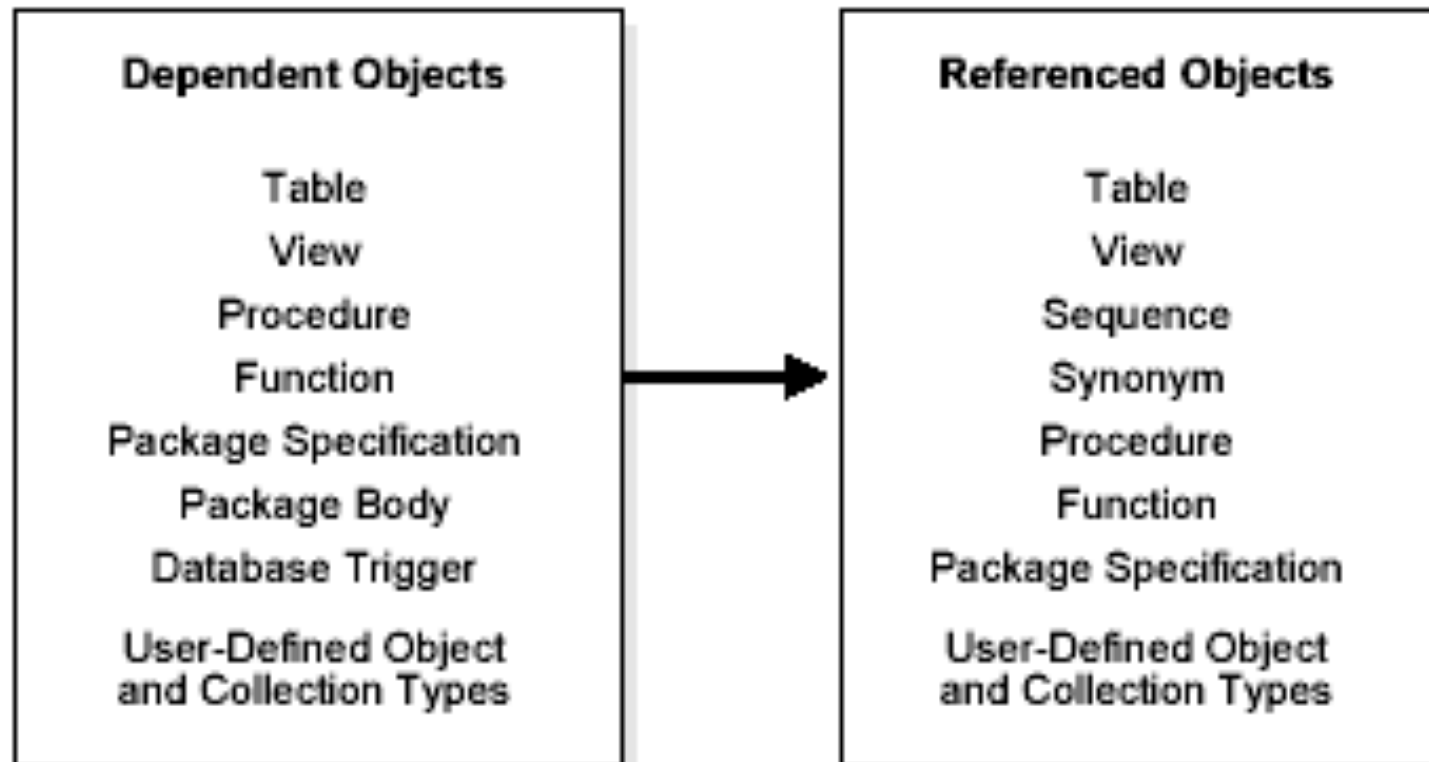
# Dependency Issues

# Dependency Issues

- The definitions of some objects, including views and procedures, reference other objects, such as tables.
- As a result, the objects being defined are dependent on the objects referenced in their definitions.
- For example, a view is defined by a query that references tables or other views.
- An object that references another object as part of its definition is called a ***dependent*** object, while the object being referenced is a ***referenced*** object.

# Different types of dependent and referenced objects

*Figure 19-1 Types of Possible Dependent and Referenced Schema Objects*



# Different types of dependent and referenced objects

- If you alter the definition of a referenced object, dependent objects may or may not continue to function without error, depending on the type of alteration.
- Oracle automatically records dependencies among objects.
- For example, if you alter a table on which several stored procedures depend, Oracle automatically recompiles the dependent procedures the next time the procedures are referenced (executed or compiled against).
- To manage dependencies among schema objects, all of the schema objects in a database have a status:
  - **VALID** The schema object has been compiled and can be immediately used when referenced.
  - **INVALID** The schema object must be compiled before it can be used.
- Oracle automatically tracks specific changes in the database and records the appropriate status for related objects in the data dictionary.
- Status recording is a recursive process. Any change in the status of a referenced object changes the status not only for directly dependent objects, but also for indirectly dependent objects.

# Resolution of Schema Object Dependencies

- When a schema object is referenced directly in a SQL statement or indirectly through a reference to a dependent object, Oracle checks the status of the object explicitly specified in the SQL statement and any referenced objects, as necessary.
- Oracle's action depends on the status of the objects that are directly and indirectly referenced in a SQL statement:
- If every referenced object is valid, then Oracle executes the SQL statement immediately without any additional work.
  - If any referenced view or procedure (including a function or package) is invalid, then Oracle automatically attempts to compile the object.
  - If all invalid referenced objects can be compiled successfully, then they are compiled and Oracle executes the SQL statement.
- If an invalid object cannot be compiled successfully, then it remains invalid. Oracle returns an error and rolls back the transaction containing the SQL statement.

# Compilation of Views and PL/SQL Program Units

- A view or PL/SQL program unit can be compiled and made valid if the following conditions are satisfied:
  1. The definition of the view or program unit must be correct. All of the SQL and PL/SQL statements must be proper constructs.
  2. All referenced objects must be present and of the expected structure. For example, if the defining query of a view includes a column, the column must be present in the base table.
  3. The owner of the view or program unit must have the necessary privileges for the referenced objects.

# Views and Base Tables

- A view depends on the base tables or views referenced in its defining query.
- If the defining query of a view is not explicit about which columns are referenced, for example, `SELECT * FROM table`, then the defining query is expanded when stored in the data dictionary to include all columns in the referenced base table at that time.
- If a base table or view of a view is altered, renamed, or dropped, then the view is invalidated, but its definition remains in the data dictionary along with the privileges, synonyms, other objects, and other views that reference the invalid view.

# Program Units and Referenced Objects

- Oracle automatically invalidates a program unit when the definition of a referenced object is altered.
- For example, assume that a standalone procedure includes several statements that reference a table, a view, another standalone procedure, and a public package procedure. In that case, the following conditions hold:
  - If the referenced table is altered, then the dependent procedure is invalidated.
  - If the base table of the referenced view is altered, then the view and the dependent procedure are invalidated.
  - If the referenced standalone procedure is replaced, then the dependent procedure is invalidated.
  - If the *body* of the referenced package is replaced, then the dependent procedure is not affected. However, if the **specification** of the referenced package is replaced, then the dependent procedure is invalidated.
- Whenever you create a table, index, and view, and then drop the index, all objects dependent on that table are invalidated, including views, packages, package bodies, functions, and procedures.



# Security Authorizations

- Oracle notices when a DML object or system privilege is granted to or revoked from a user and automatically invalidates all the owner's dependent objects.
- Oracle invalidates the dependent objects to verify that an owner of a dependent object continues to have the necessary privileges for all referenced objects.
- Internally, Oracle notes that such objects do not have to be recompiled. Only security authorizations need to be validated, not the structure of any objects.

# Function-Based Index Dependencies

- Function-based indexes depend on functions used in the expression that defines the index. If a PL/SQL function or package function is changed, then the index is marked as *disabled*.

## Requirements to use a function-based index:

- The table must be analyzed after the index is created.
- The query must be guaranteed not to need any NULL values from the indexed expression, because NULL values are not stored in indexes.
- Any user-written function used in a function-based index must have been declared with the DETERMINISTIC keyword to indicate that the function will always return the same output return value for any given set of input argument values.
- The index owner needs the EXECUTE privilege on the function used to define a function-based index. If the EXECUTE privilege is revoked, Oracle marks the index DISABLED.
- If the function or the specification of a package containing the function is redefined (or if the index owner's EXECUTE privilege is revoked), then the following conditions hold:
  - The index is marked as DISABLED.
  - Queries on a DISABLED index fail if the optimizer chooses to use the index.

# Dependency Management

When a dependent object is created, Oracle takes the following steps:

1. Oracle attempts to resolve all references by first searching in the current schema.
2. If a referenced object is not found in the current schema, Oracle attempts to resolve the reference by searching for a private synonym in the same schema.
3. If a private synonym is not found, Oracle looks for a public synonym.
4. If a public synonym is not found, Oracle searches for a schema name that matches the first portion of the object name.
5. If a matching schema name is found, Oracle attempts to find the object in that schema.
6. If no schema is found, an error is returned.

# Shared SQL Dependency Management

- In addition to managing dependencies among schema objects, Oracle also manages dependencies of each shared SQL area in the shared pool.
- If a table, view, synonym, or sequence is created, altered, or dropped, or a procedure or package specification is recompiled, all dependent shared SQL areas are invalidated.
- At a subsequent execution of the cursor that corresponds to an invalidated shared SQL area, Oracle reparses the SQL statement to regenerate the shared SQL area.

# Local and Remote Dependency Management

- Tracking dependencies and completing necessary recompilations are performed automatically by Oracle.
- **Local dependency management** occurs when Oracle manages dependencies among the objects in a single database. For example, a statement in a procedure can reference a table in the same database.
- Oracle manages all local dependencies using the database's internal dependency table, which keeps track of each schema object's dependent objects.
- When a referenced object is modified, Oracle uses the depends-on table to identify dependent objects, which are then invalidated.
- **Remote dependency management** occurs when Oracle manages dependencies in distributed environments across a network. For example, an Oracle Forms trigger can depend on a schema object in the database.

# Management of Remote Dependencies

- Oracle uses different mechanisms to manage remote dependencies.

## Dependencies Among Local and Remote Database Procedures

- Dependencies among stored procedures in a distributed database system are managed using **time stamp checking** or **signature checking**.
  - **Time stamp Checking** : Here, whenever a procedure is compiled or recompiled its time stamp (time it is created, altered, or replaced) is recorded in the data dictionary.
  - The compiled version of the procedure contains information about each remote procedure that it references, including the remote procedure's schema, package name, procedure name, and time stamp.
  - When a dependent procedure is used, Oracle compares the remote time stamps recorded at compile time with the current time stamps of the remotely referenced procedures.
  - Depending on the result of this comparison, two situations can occur:
    - Local and remote procedures execute without compilation if the time stamps match.
    - Local procedure is invalidated if any time stamps of remotely referenced procedures do not match, and an error is returned to the calling environment. Furthermore, all other local procedures that depend on the remote procedure with the new time stamp are also invalidated.

# Management of Remote Dependencies

## Dependencies Among Other Remote Schema Objects

- Oracle does not manage dependencies among remote schema objects other than local-procedure-to-remote-procedure dependencies.
- For example, assume that a local view is created and defined by a query that references a remote table. Also assume that a local procedure includes a SQL statement that references the same remote table. Later, the definition of the table is altered. As a result, the local view and procedure are never invalidated, even if the view or procedure is used after the table is altered, and even if the view or procedure now returns errors when used. In this case, the view or procedure must be altered manually so errors are not returned.

## Dependencies of Applications

- Code in database applications can reference objects in the connected database.
- Such applications are dependent on the schema objects they reference.
- Dependency management techniques vary, depending on the development environment.