11

Using Flashback Technology II

Objectives

After completing this lesson, you should be able to:

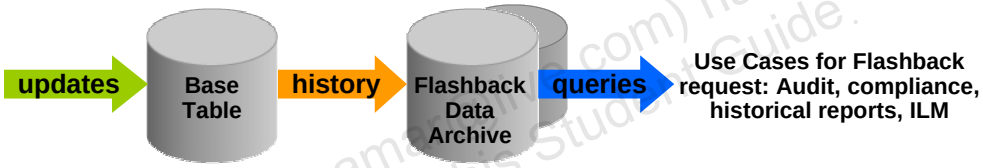
- Describe and use Oracle Total Recall
  - Creating and enabling a Flashback Data Archive (FDA)
  - Managing FDAs
  - Viewing metadata
- Describe and use flashback recycle bins
  - Restore dropped tables from the recycle bin
  - Manage space usage in the recycle bin
  - Query the recycle bin

Oracle Total Recall Overview

Automated tracking of historical database changes:

- Enable at the table level with your specified retention period.
- All subsequent changes are transparently stored and tamper proof.
- Records older than retention period are automatically removed.
- Use Flashback technologies to retrieve history.

```
SELECT … AS OF TIMESTAMP…
SELECT … VERSIONS BETWEEN TIMESTAMP and TIMESTAMP…
```



ORACLE

Oracle Total Recall Overview

The Oracle Total Recall option in Oracle Database 11*g* (also known as Flashback Data Archive) provides a mechanism for tracking changes to production databases that is secure, efficient, easy to use, and application transparent.

With Oracle Total Recall technology, you can automatically track and store the data in tables enabled for Flashback Data Archive. This ensures that flashback queries obtain SQL-level access to the versions of database objects without getting a snapshot-too-old error.

A Flashback Data Archive provides the ability to track and store all transactional changes to a "tracked" table over its life time. It is no longer necessary to build this intelligence into your application. You can use Oracle Total Recall for compliance, audit reports, data analysis, and decision-support systems. The Flashback Data Archive background process starts with the database.

Use case examples:

- Audit support: Find duplicate insurance claims from the last year.
- Compliance support: Monitor stock trading during a quiet period.
- Information Lifecycle Management (ILM): Guarantee immutable access to patient history.
- Retention policy enforcement: Automatically purge records older than five years.
- Historical reporting: Retrieve a client's credit and payment history.
- Error Recovery: Restore records erroneously removed or updated.

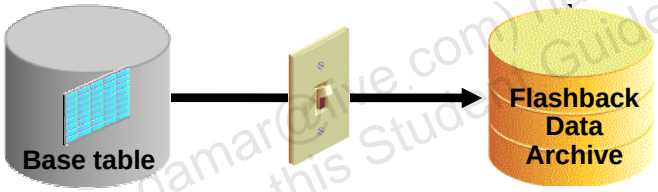
Setup Process

1. Create a new tablespace to hold the FDA.
2. With the FLASHBACK ARCHIVE ADMINISTER system privilege: Create a Flashback Data Archive, assign it to the tablespace, and specify its retention period.

```
CREATE FLASHBACK ARCHIVE fda1
TABLESPACE fda_tbs1 QUOTA 10M RETENTION 1 YEAR;
```

3. With the FLASHBACK ARCHIVE object privilege: Alter the base tables to enable archiving and assign it to a flashback archive.

```
ALTER TABLE HR.EMPLOYEES FLASHBACK ARCHIVE fda1;
```



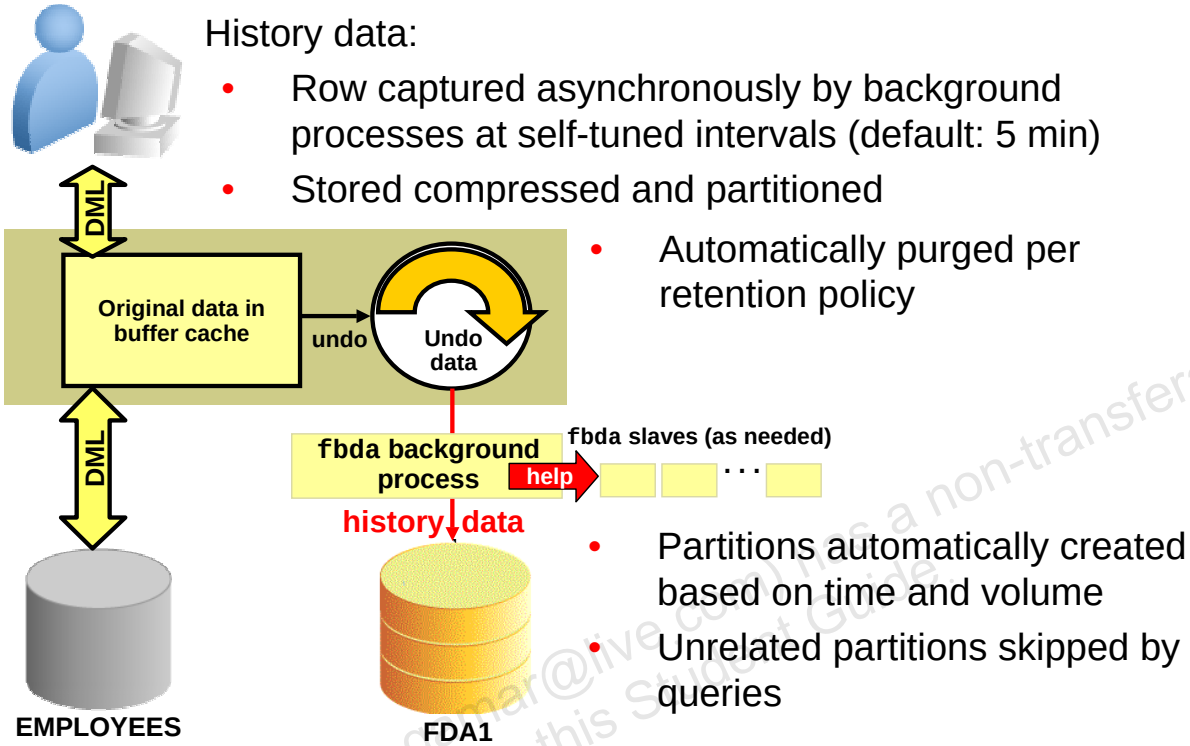
ORACLE

Flashback Data Archive Process

A Flashback Data Archive consists of one or more tablespaces. You can have multiple Flashback Data Archives. They are configured with retention duration. Based on your retention duration requirements, you should create different Flashback Data Archives—for example, one for all records that must be kept for two years, another for all records that must be kept for five years. The database will automatically purge all historical information on the day after the retention period expires.

1. Create a tablespace for your Flashback Data Archive. The size depends on the base table and the expected DML and DDL activity.
2. Create a Flashback Data Archive with retention time. Data archived in the Flashback Data Archive is retained for the retention time. This task requires the FLASHBACK ARCHIVE ADMINISTER system privilege. If different retention periods are needed, different archives must be created.
3. Enable flashback archiving (and then disable it again) for a (whole) table. This task requires the FLASHBACK ARCHIVE object privilege. Although flashback archiving is enabled for a table, some DDL statements are not allowed on that table. By default, flashback archiving is off for any table.

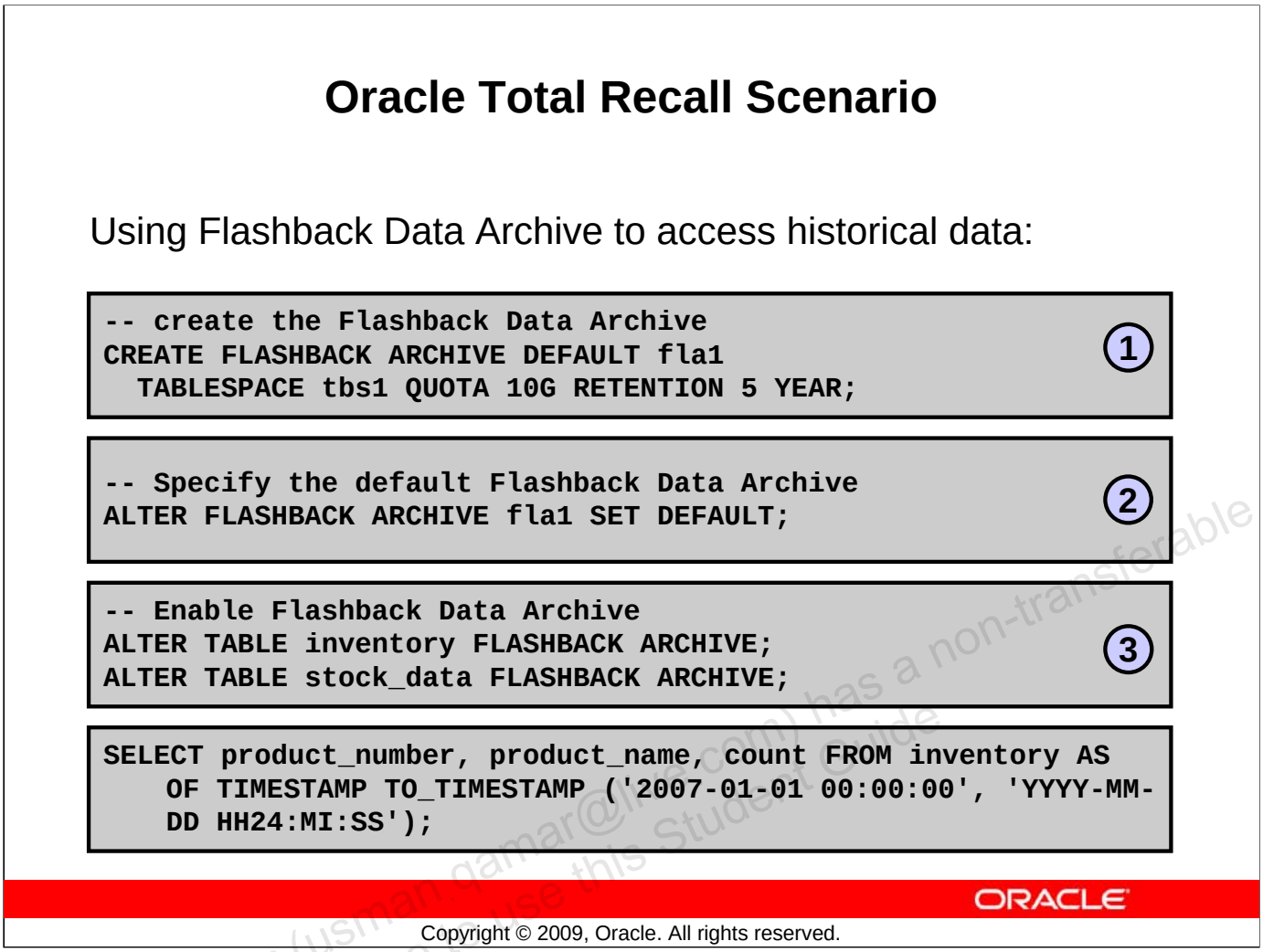
How Total Recall Works



How Total Recall Works

History data is captured from undo (and buffer cache) by the `fdba` background process at self-tuned intervals. The default is every five minutes. The entire base table row that is updated is stored, no matter how many columns are updated.

- History data is compressed using OLTP Table compression, not Hybrid Columnar compression. **Note:** If the base table is compressed with Hybrid Columnar compression, the table cannot be enabled for Flashback Data Archiving.
- Each flashback archive partition is at least 1 day and 1 MB of data, partitioned on ENDSCN. Flashback queries to the archives avoid unrelated partitions.
- Up to ten flashback archiver slaves can be called upon by the `fbda` process.
- If the flashback archive process and slaves are too busy, archiving may be performed inline, which significantly affects the user's response time.



Oracle Total Recall Scenario

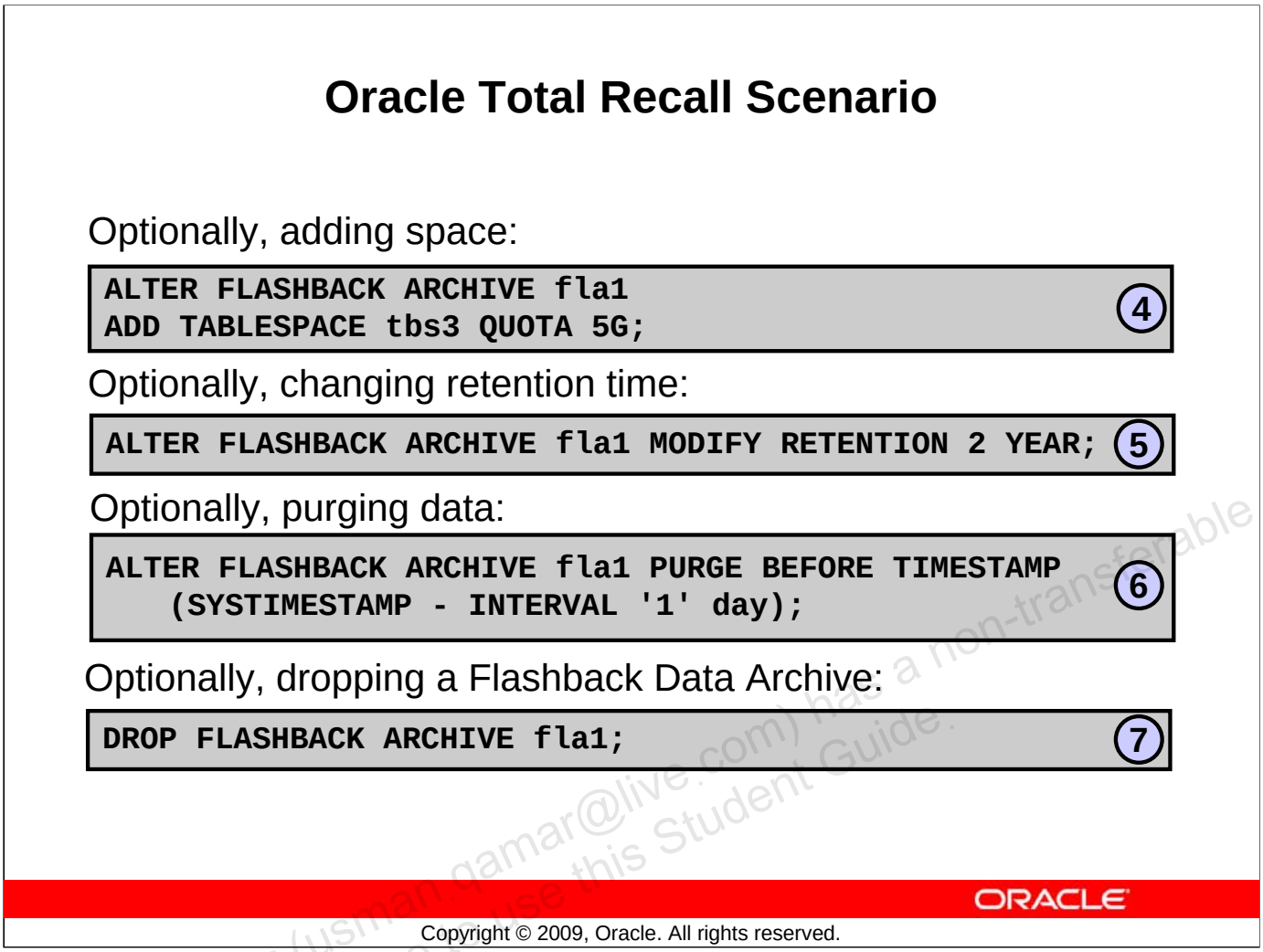
You create a Flashback Data Archive with the `CREATE FLASHBACK ARCHIVE` statement.

- You can optionally specify the default Flashback Data Archive for the system.
- You need to provide the name of the Flashback Data Archive.
- You need to provide the name of the first tablespace of the Flashback Data Archive.
- You can identify the maximum amount of space that the Flashback Data Archive can use in the tablespace. The default is unlimited. Unless your space quota on the first tablespace is unlimited, you must specify this value, or else an `ORA-55621` will ensue.
- You need to provide the retention time (number of days that Flashback Data Archive data for the table is guaranteed to be stored).

In the first example shown in the slide, a default Flashback Data Archive named `fla1` is created that uses up to 10 GB of the `tbs1` tablespace, whose data will be retained for five years. In the second example, the default Flashback Data Archive is specified. By default, the system has no Flashback Data Archive. You can set it in one of two ways:

- Specify the name of an existing Flashback Data Archive in the `SET DEFAULT` clause of the `ALTER FLASHBACK ARCHIVE` statement.
- Include `DEFAULT` in the `CREATE FLASHBACK ARCHIVE` statement when you create a Flashback Data Archive.

In the third example, Flashback Data Archive is enabled. If Automatic Undo Management is disabled, you receive an `ORA-55614` if you try to modify the table.



Oracle Total Recall Scenario (continued)

To enable flashback archiving for a table, include the `FLASHBACK ARCHIVE` clause in either the `CREATE TABLE` or `ALTER TABLE` statement. In the `FLASHBACK ARCHIVE` clause, you can specify the Flashback Data Archive where the historical data for the table will be stored. The default is the default Flashback Data Archive for the system. To disable flashback archiving for a table, specify `NO FLASHBACK ARCHIVE` in the `ALTER TABLE` statement.

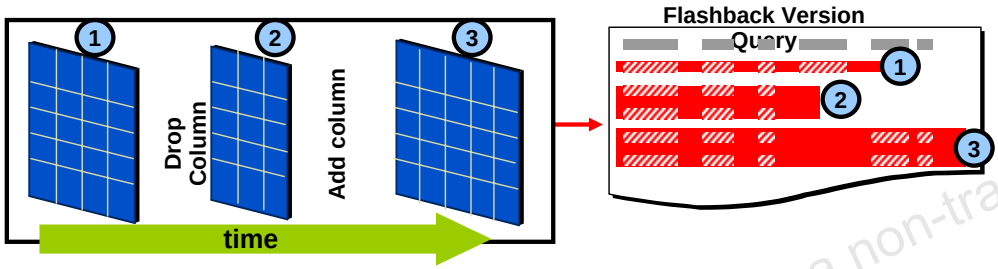
The last statement shown in the previous slide shows how to retrieve the inventory of all items at the beginning of the year 2007. Continuing the previous examples:

- Example 4 adds up to 5 GB of the `tbs3` tablespace to the `fla1` Flashback Data Archive.
- Example 5 changes the retention time for the `fla1` Flashback Data Archive to two years.
- Example 6 purges all historical data older than one day from the `fla1` Flashback Data Archive. Normally, purging is done automatically on the day after your retention time expires. You can also override this for ad hoc clean-up.
- Example 7 drops the `fla1` Flashback Data Archive and historical data, but not its tablespaces. With the `ALTER FLASHBACK ARCHIVE` command, you can:
  - Change the retention time of a Flashback Data Archive
  - Purge some or all of its data
  - Add, modify, and remove tablespaces

Note: Removing all tablespaces of a Flashback Data Archive causes an error.

Transparent Schema Evolution

- DDL support for:
  - Add, drop, rename, and modify column
  - Drop and truncate partition
  - Rename and truncate table



- Flashback queries work across DDL changes.
- All other DDL is *not* automatically supported (see next slide).

ORACLE

Transparent Schema Evolution

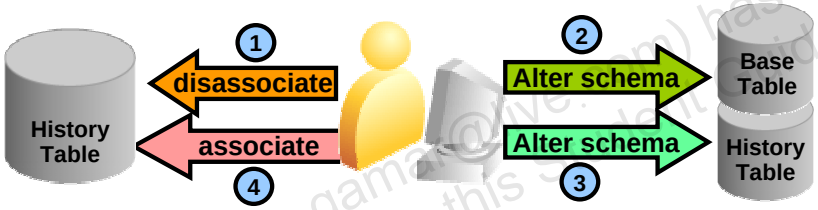
The most common DDL commands are possible with Flashback Data Archives. When a schema has evolved in any of the ways listed in the slide, Total Recall automatically keeps track of the changes. Flashback query appropriately returns the row or rows with the corresponding schema (as shown in the diagram).

Full Schema Evolution

Disassociate or associate procedures in the DBMS_FLASHBACK_ARCHIVE package:

- Disable Total Recall on specified tables and allow more complex DDL (upgrades, split tables, and so on).
- Enforce schema integrity during association. (Base table and history table must be the same schema.)

Note: This function should be used with care and with the understanding that the archive can no longer be guaranteed to be immutable because the history could have been altered during the time of disassociation.



ORACLE

Full Schema Evolution

All DDL changes that are not automatically supported can be executed through the DBMS_FLASHBACK_ARCHIVE package. You can use the DISASSOCIATE_FBA and REASSOCIATE_FBA procedures to disassociate and reassociate a given table from its Flashback Data Archive.

Note: This function should be used with care and with the understanding that the archive can no longer be guaranteed to be immutable, because the history could have been altered during the time of disassociation. The system catalog has a note when the disassociation occurred.

The diagram in the slide shows the following workflow:

- If you have the FLASHBACK ARCHIVE ADMINISTER privilege, you can disassociate the archive from the base table.
- Make the necessary changes to the base table.
- Make the necessary changes to the corresponding archive.
- Then you associate the table with the archive within the same schema. Total Recall validates that the schemas are the same upon association.

Restrictions

- You cannot enable Total Recall for base tables with Hybrid Columnar compression.
- If disassociate is used, immutability of history is no longer guaranteed (but you could always purge history previously anyway with the right privilege).
- There is no transportability of history tables.

Restrictions

Some DDL statements cause error `ORA-55610` when used on a table enabled for Flashback Data Archive. For example:

- `ALTER TABLE statement that includes an UPGRADE TABLE` clause, with or without an `INCLUDING DATA` clause
- `ALTER TABLE` statement that moves or exchanges a partition or subpartition operation
- `DROP TABLE` statement

Guidelines

- Use SCN for precise queries.

  or
- Use Flashback technology for your convenience.
- Flashback uses current system settings.
- Ensure database consistency with a `COMMIT` or `ROLLBACK` operation before querying past data.
- You cannot retrieve past data from a dynamic performance (V$) view. They contain current data.
- However, you can perform queries on past data in static data dictionary views, such as `*_TABLES`.

Guidelines

- Use Flashback Query, Flashback Version Query, or Flashback Transaction Query for SQL code that you write, for convenience.
- Remember that all flashback processing uses the current session settings, such as national language and character set, not the settings that were in effect at the time being queried.
- To query past data at a precise time, use an `SCN`. If you use a time stamp, the actual time queried might be up to 3 seconds earlier than the time you specify. Oracle Database uses SCNs internally and maps them to time stamps at a granularity of 3 seconds.
- To obtain an SCN to use later with a flashback feature, you can use the `DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER` function.
- To compute or retrieve a past time to use in a query, use a function return value as a time-stamp or `SCN` argument. For example, add or subtract an `INTERVAL` value to the value of the `SYSTIMESTAMP` function.
- To ensure database consistency, always perform a `COMMIT` or `ROLLBACK` operation before querying past data.
- You cannot retrieve past data from a dynamic performance (`V$`) view. A query on such a view always returns current data. However, you can perform queries on past data in static data dictionary views, such as `*_TABLES`.

Viewing Flashback Data Archives

Viewing the results:

View Name (DBA/USER)	Description
*_FLASHBACK_ARCHIVE	Displays information about Flashback Data Archives
*_FLASHBACK_ARCHIVE_TS	Displays tablespaces of Flashback Data Archives
*_FLASHBACK_ARCHIVE_TABLES	Displays information about tables that are enabled for flashback archiving

Viewing Flashback Data Archives

You can use the dynamic data dictionary views to view tracked tables and Flashback Data Archive metadata. To access the USER_FLASHBACK_* views, you must have table ownership privileges. To inspect the DBA_FLASHBACK_* views, you need SYSDBA privileges.

For more details, see the *Advanced Application Developer's Guide* and the *PL/SQL Packages and Types Reference*.

Quiz

You cannot drop, but you can truncate, a table that is tracked by Oracle Total Recall.

1. True
2. False

Answer: 1

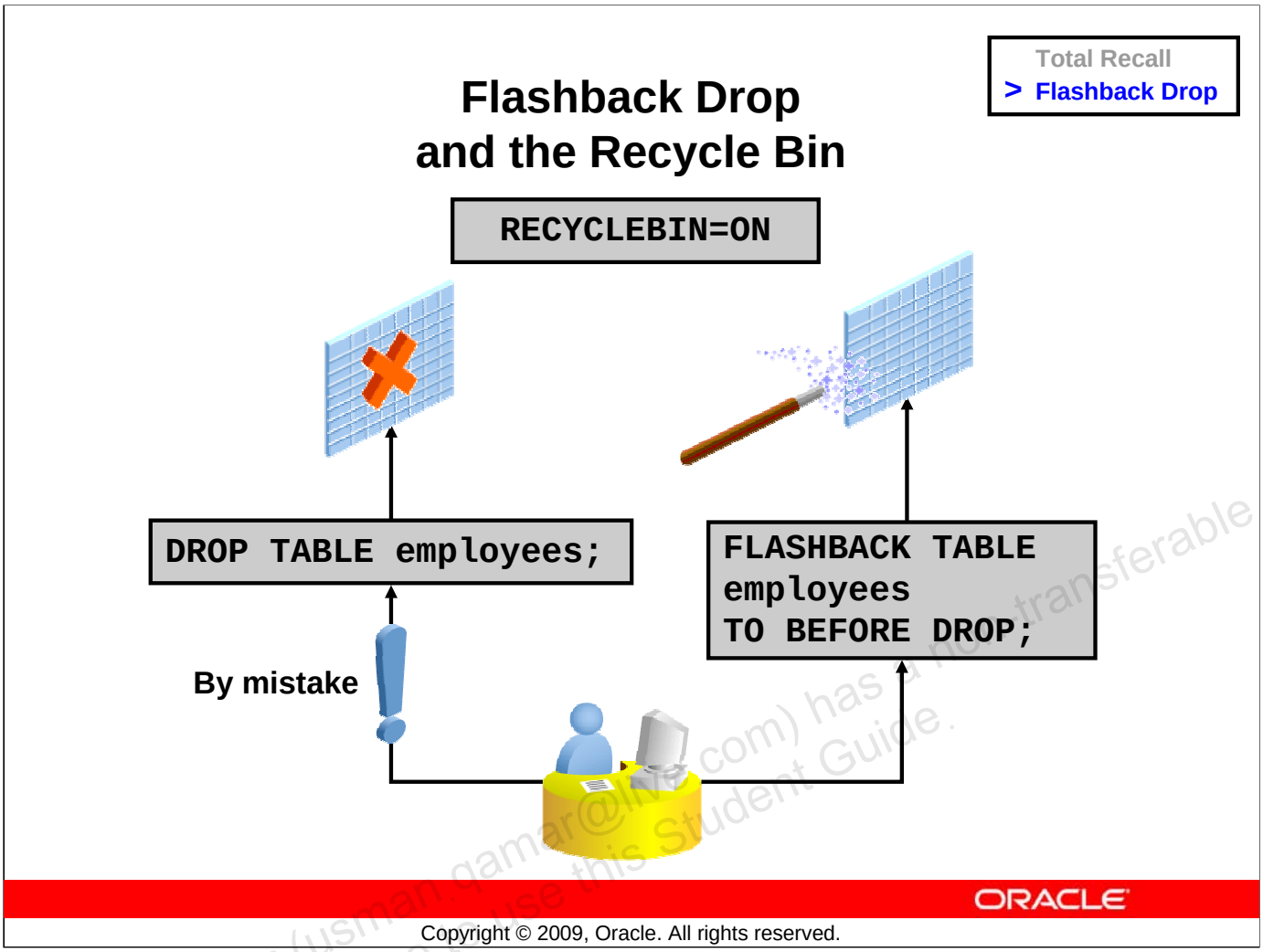
Quiz

Select all correct statements about Oracle Total Recall:

1. Oracle Total Recall is enabled by default.
2. A Flashback Data Archive provides the ability to track and store all transactional changes to a “tracked” table over its lifetime.
3. Dropping a column in a table enabled for Flashback Data Archive causes an error.
4. Flashback processing always uses the settings that were in effect at the time of being queried.
5. Flashback uses the current session settings, such as national language and character set.

ORACLE

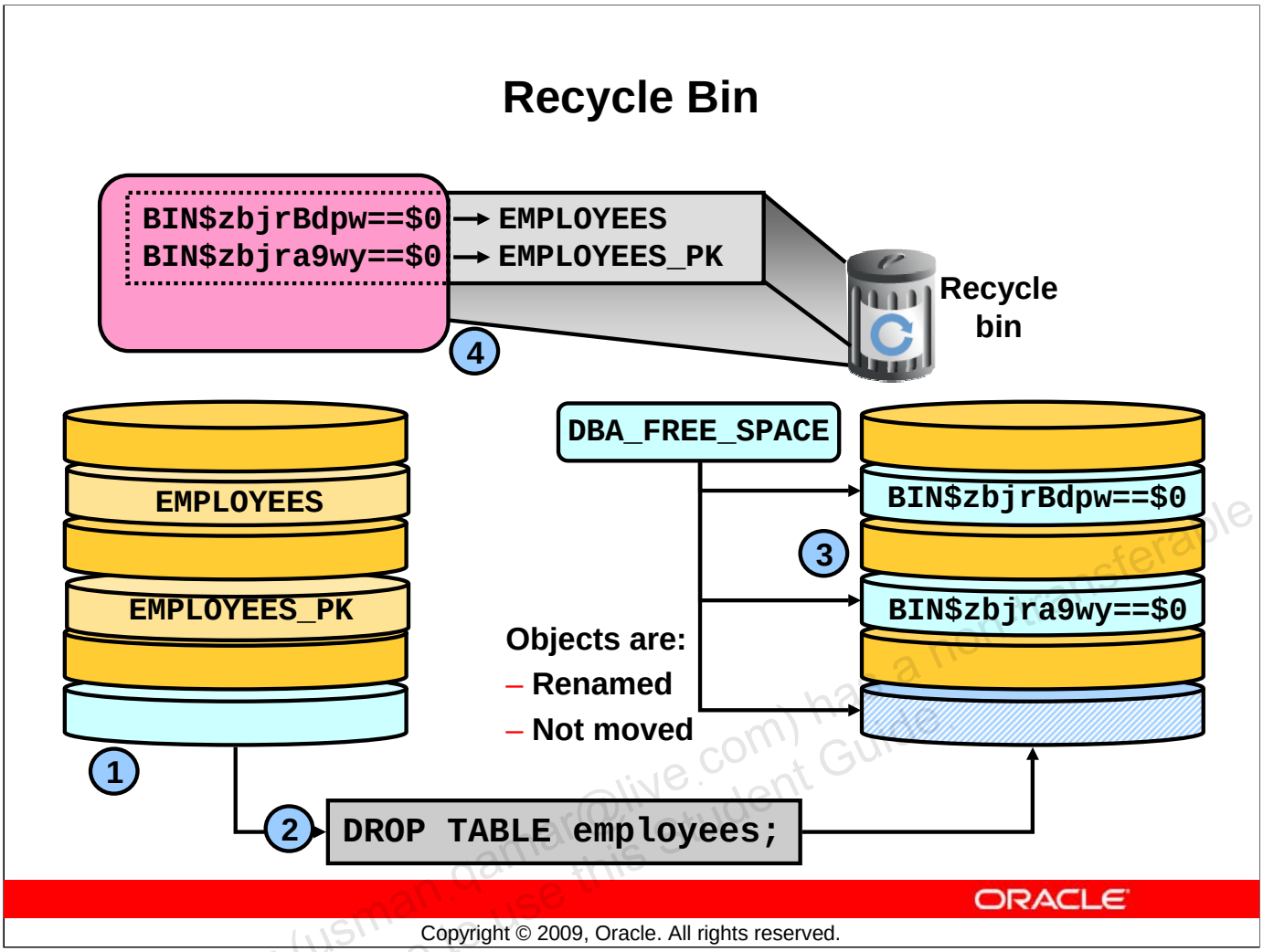
Answer: 2, 5



Flashback Drop and the Recycle Bin

Using the FLASHBACK TABLE command, you can undo the effects of a DROP TABLE statement without having to use point-in-time recovery.

Note: The RECYCLEBIN initialization parameter is used to control whether the Flashback Drop capability is turned ON or OFF. If the parameter is set to OFF, then dropped tables do not go into the recycle bin. If this parameter is set to ON, the dropped tables go into the recycle bin and can be recovered. By default, RECYCLEBIN is set to ON.



Recycle Bin

Without the recycle bin enabled, when you drop a table, the space associated with the table and its dependent objects is immediately reclaimable (that is, it can be used for other objects).

If the recycle bin is enabled, when you drop a table, then the space associated with the table and its dependent objects is not immediately reclaimable, even though it does appear in `DBA_FREE_SPACE`. Instead, the dropped objects are referenced in the recycle bin and still belong to their owner. The space used by recycle bin objects is never automatically reclaimed unless there is space pressure. This enables you to recover recycle bin objects for the maximum possible duration.

When a dropped table is “moved” to the recycle bin, the table and its associated objects and constraints are renamed using system-generated names. The renaming convention is as follows:

```
BIN$unique_id$version
```

where `unique_id` is a 26-character globally unique identifier for this object making the recycle bin name unique across all databases and `version` is a version number assigned by the database.

Recycle Bin (continued)

The recycle bin itself is a data dictionary table that maintains the relationships between the original names of dropped objects and their system-generated names. You can query the recycle bin by using the `DBA_RECYCLEBIN` view. The diagram in the previous slide illustrates this behavior:

1. You have created a table called `EMPLOYEES` in your tablespace.
2. You drop the `EMPLOYEES` table.
3. The extents occupied by `EMPLOYEES` are now considered as free space.
4. `EMPLOYEES` is renamed and the new name is recorded into the recycle bin.

Restoring Tables from the Recycle Bin

- Restore dropped tables and dependent objects.
- If multiple recycle bin entries have the same original name:
  - Use unique, system-generated names to restore a particular version
  - When using original names, the restored table is last in, first out (LIFO)
- Rename the original name if that name is currently used.

```
FLASHBACK TABLE <table_name> TO BEFORE DROP
[RENAME TO <new_name>];
```


Restoring Tables from the Recycle Bin

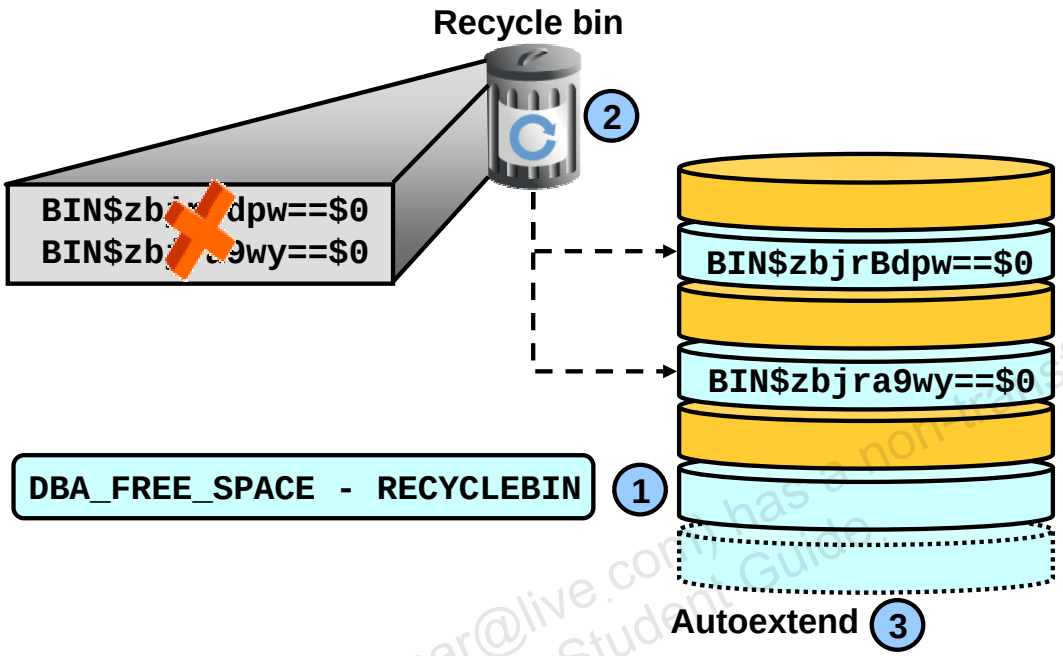
Use the FLASHBACK TABLE . . . TO BEFORE DROP command to recover a table and all of its possible dependent objects from the recycle bin. You can specify either the original name of the table or the system-generated name assigned to the object when it was dropped.

If you specify the original name, and if the recycle bin contains more than one object of that name, then the object that was moved to the recycle bin most recently is recovered first (LIFO: last in, first out). If you want to retrieve an older version of the table, you can specify the system-generated name of the table that you want to retrieve, or issue additional FLASHBACK TABLE . . . TO BEFORE DROP statements until you retrieve the table you want.

If a new table of the same name has been created in the same schema since the original table was dropped, then an error is returned unless you also specify the RENAME TO clause.

Note: When you flash back a dropped table, the recovered indexes, triggers, and constraints keep their recycle bin names. Therefore, it is advisable to query the recycle bin and DBA_CONSTRAINTS before flashing back a dropped table. In this way, you can rename the recovered indexes, triggers, and constraints to more usable names.

Recycle Bin: Automatic Space Reclamation



Recycle Bin: Automatic Space Reclamation

As long as the space used by recycle bin objects is not reclaimed, you can recover those objects by using Flashback Drop. The following are recycle bin object reclamation policies:

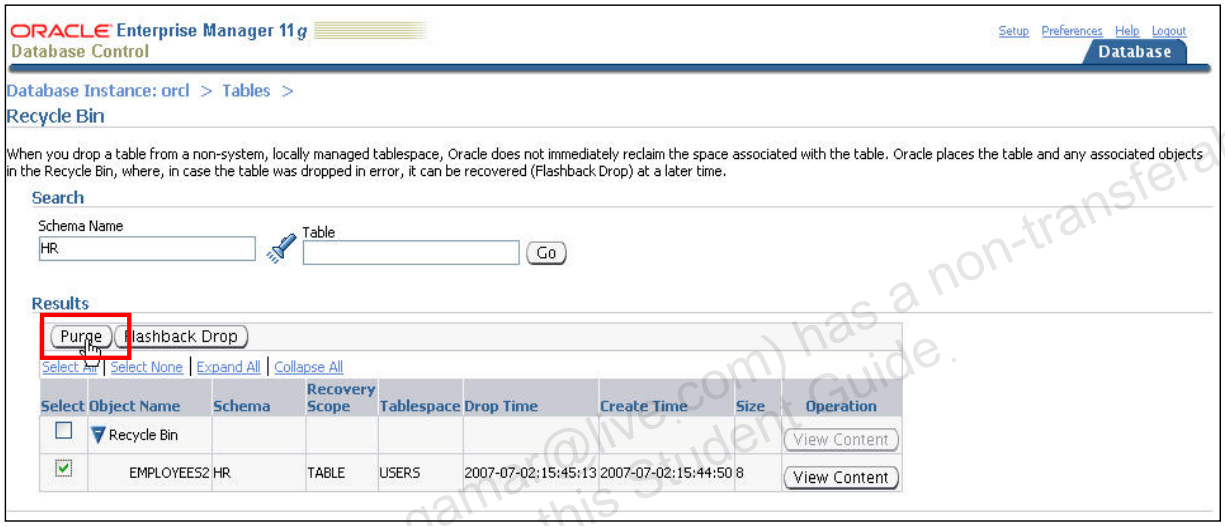
- Manual cleanup when you explicitly issue a PURGE command
- Automatic cleanup under space pressure: While objects are in the recycle bin, their corresponding space is also reported in DBA_FREE_SPACE because their space is automatically reclaimable. The free space in a particular tablespace is then consumed in the following order:
  1. Free space not corresponding to recycle bin objects
  2. Free space corresponding to recycle bin objects. In this case, recycle bin objects are automatically purged from the recycle bin using a first in, first out (FIFO) algorithm.
  3. Free space automatically allocated if the tablespace is auto-extensible. Suppose that you create a new table inside the TBS1 tablespace. If there is free space allocated to this tablespace that does not correspond to a recycle bin object, this free space is used as a first step. If this is not enough, free space is used that corresponds to recycle bin objects that reside inside TBS1. If the free space of some recycle bin objects is used, these objects are purged automatically from the recycle bin. At this time, you can no longer recover these objects by using the Flashback Drop feature. As a last resort, the TBS1 tablespace is extended (if possible) if the space requirement is not yet satisfied.

Recycle Bin: Manual Space Reclamation

```
PURGE {TABLE <table_name>|INDEX <index_name>}
```

```
PURGE TABLESPACE <ts_name> [USER <user_name>]
```

```
PURGE [USER_|DBA_]RECYCLEBIN
```



Recycle Bin: Manual Space Reclamation

Use the PURGE command to permanently remove objects from the recycle bin. When an object is purged from the recycle bin, the object and dependent objects are permanently removed from the database. As a consequence, objects purged from the recycle bin are no longer recoverable by using Flashback Drop. The following are possible uses of PURGE:

- PURGE TABLE purges the specified table.
- PURGE INDEX purges the specified index.
- PURGE TABLESPACE purges all the objects residing in the specified tablespace. In addition, objects residing in other tablespaces may get purged if they are dependent.
- PURGE RECYCLEBIN purges all the objects that belong to the current user. RECYCLEBIN and USER_RECYCLEBIN are synonymous.
- PURGE DBA_RECYCLEBIN purges all the objects. You must have enough system privileges or the SYSDBA system privilege to issue this command.

Tables can also be purged from the recycle bin using Enterprise Manager. On the Schema folder tab, click Tables, then select the schema the dropped object resided in and click the Recycle Bin button. Select the table from the results list and click the Purge button.

Note: For PURGE TABLE and PURGE INDEX commands, if you specify an original name and if the recycle bin contains more than one object of that name, then the object that has been in the recycle bin the longest is purged first (FIFO).

Bypassing the Recycle Bin

```
DROP TABLE <table_name> [PURGE] ;
```

```
DROP TABLESPACE <ts_name>
[INCLUDING CONTENTS] ;
```

```
DROP USER <user_name> [CASCADE] ;
```

Security considerations for the recycle bin:

```
ALTER SYSTEM SET RECYCLEBIN=OFF SCOPE=SPFILE;
```

ORACLE

Bypassing the Recycle Bin

You can use the DROP TABLE PURGE command to permanently drop a table and its dependent objects from the database. When you use this command, the corresponding objects are not moved to the recycle bin. This command provides the same functionality that the DROP TABLE command provided in previous releases.

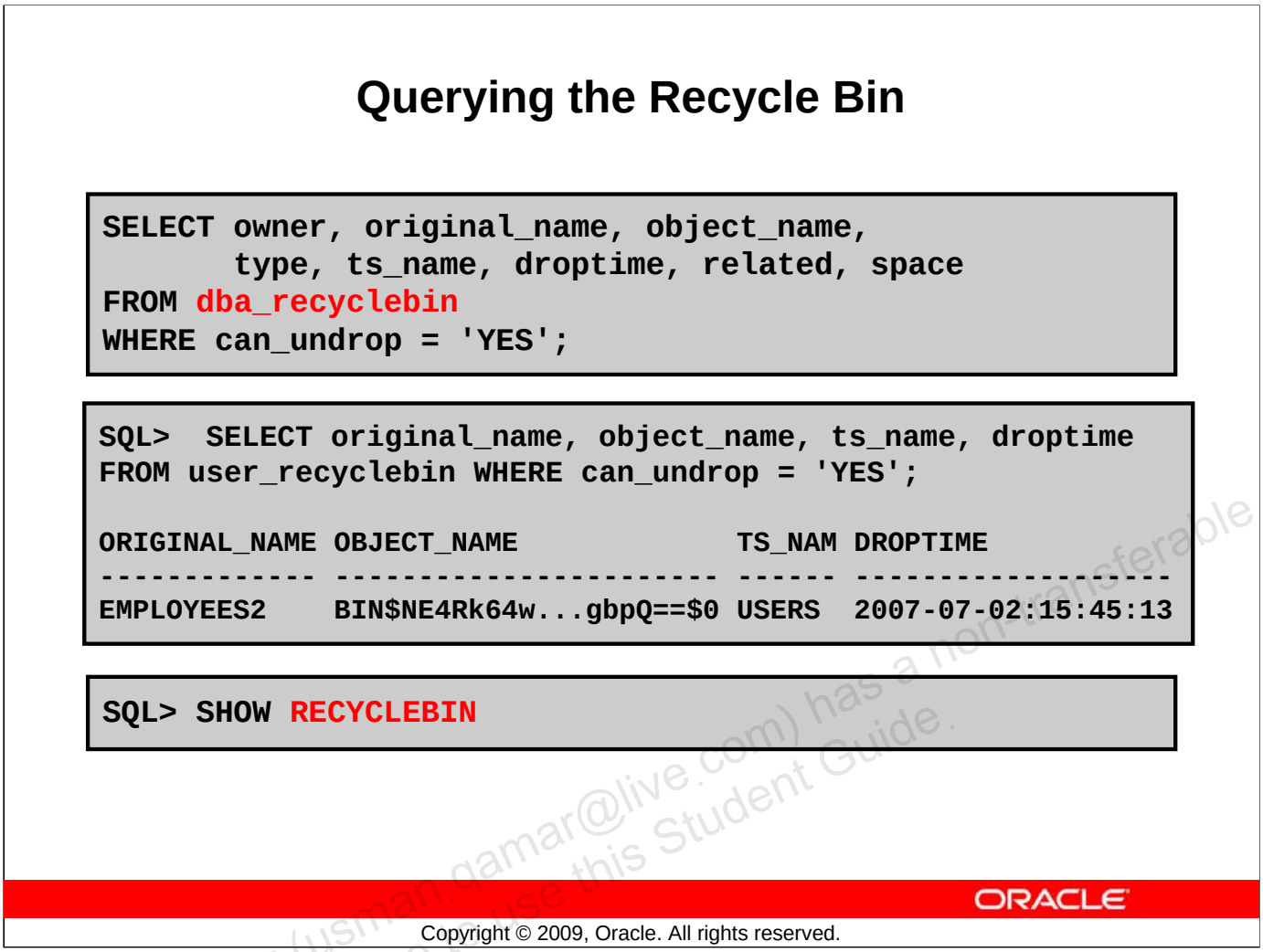
When you issue the DROP TABLESPACE . . . INCLUDING CONTENTS command, the objects in the tablespace are not placed in the recycle bin. Moreover, objects in the recycle bin belonging to the tablespace are purged. When you issue the same command without the INCLUDING CONTENTS clause, the tablespace must be empty for the command to succeed. However, there can be objects belonging to the tablespace in the recycle bin. In this case, these objects are purged.

When you issue the DROP USER . . . CASCADE command, the user and all the objects owned by the user are permanently dropped from the database. Any objects in the recycle bin belonging to the dropped user are purged.

For increased security, you may decide to not allow the use of the recycle bin. Connected as SYSDBA, you can:

- View the recycle bin status with:
  ```
  SHOW PARAMETER RECYCLEBIN
  ```
- Disable the use of the recycle bin with:
  ```
  ALTER SYSTEM SET RECYCLEBIN=OFF SCOPE=SPFILE;
  ```

After issuing this command, you need to restart the database.



Querying the Recycle Bin

You can view all the objects that you have dropped by querying `user_recyclebin` or `RECYCLEBIN`. It has a synonym `RECYCLEBIN`, for ease of use.

The `dba_recyclebin` view shows you all the objects that have been dropped by all users and that are still in the recycle bin.

You can also use the SQL*Plus `SHOW RECYCLEBIN` command. This command shows you only those objects that can be "undropped."

The examples show how to extract important information from the recycle bin:

- `original_name` is the name of the object before it is dropped.
- `object_name` is the system-generated name of the object after it is dropped.
- `type` is the object's type.
- `ts_name` is the name of the tablespace to which the object belongs.
- `droptime` is the date at which the object was dropped.
- `related` is the object identifier of the dropped object.
- `space` is the number of blocks currently used by the object.

You can also see the content of the recycle bin by using Database Control.

Note: For detailed information about the `DBA_RECYCLEBIN` view, see the *Oracle Database Reference* guide.

Quiz

When you flash back a dropped table, the recovered indexes, triggers, and constraints keep their recycle bin names.

1. True
2. False

Answer: 1

Summary

In this lesson, you should have learned how to:

- Set up and use Total Recall
- Restore dropped tables from the recycle bin
- Query the recycle bin

Practice 11 Overview: Using Flashback Technology

This practice covers the following topics:

- Using Total Recall
- Recycle bin activities (*optional*)