Recovering from Human Error using Flashback

Purpose

This lesson describes the Oracle Database 10 *g* Flashback features that support self-service error correction.

Topics

This lesson discusses the following:

- [Overview](#)
- [Filling the Recycle Bin](#)
- [Querying Tables in the Recycle Bin](#)

- [Recovering Tables from the Recycle Bin](#)

- [Emptying Tables from the Recycle Bin](#)

- [Using Flashback Table to Recover a Table](#)

Overview

[Back to List of Topics](#)

In Oracle Database 10 *g* , flashback functionality has been extended. It is now possible to flash back an entire table and quickly recover a table that has been inadvertently dropped.

What is the Recycle Bin?

In previous releases of the Oracle server, the `DROP` command permanently removed objects from the database. Now when you drop a table and its dependent objects it is placed in the recycle bin. It is not removed from the database until you empty the recycle bin. The extents allocated to the segment are not deallocated until you purge the object. You can restore these dropped objects from the recycle bin at any time.
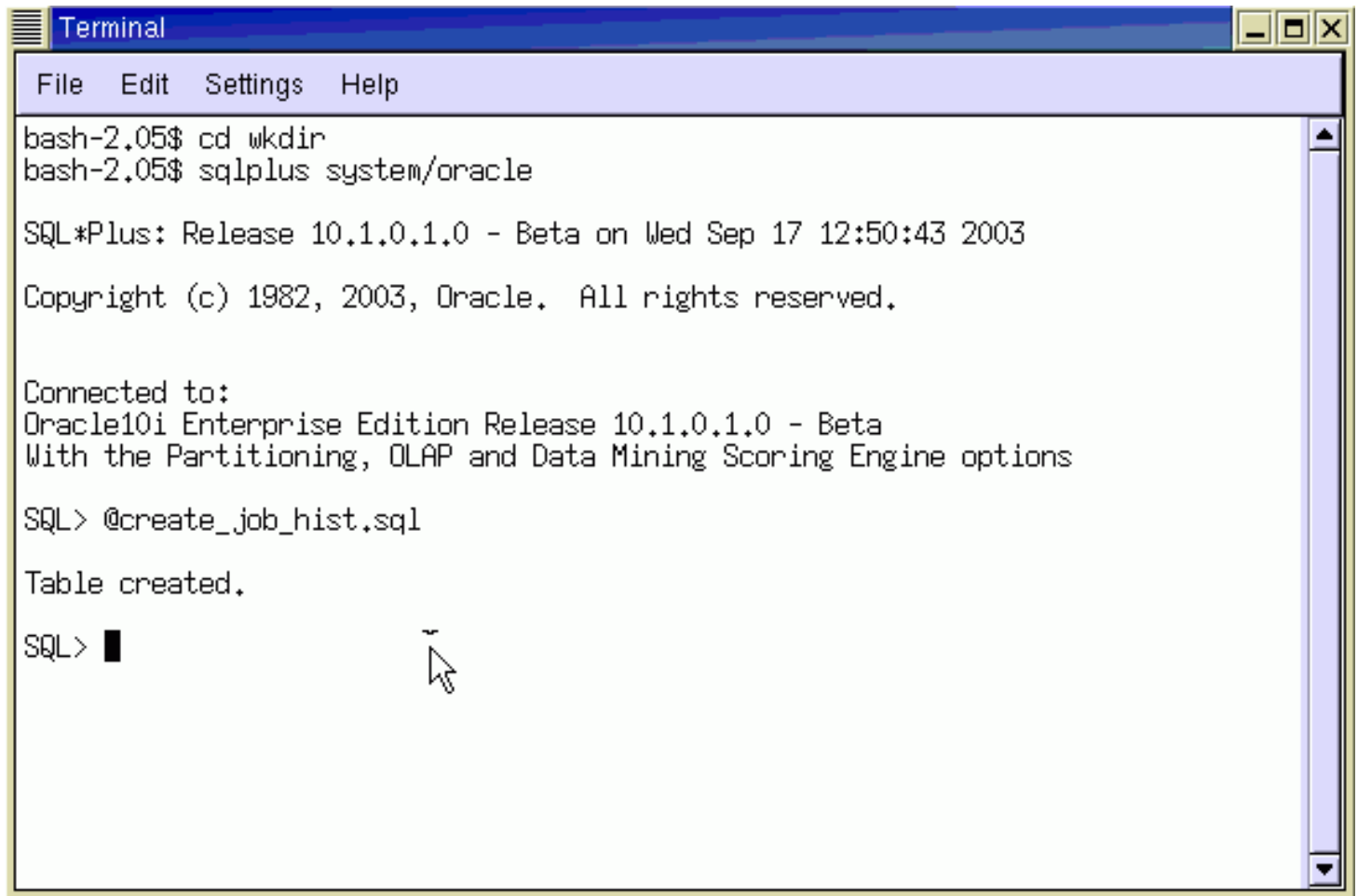
Filling the Recycle Bin

[Back to List of Topics](#)

First, you will need to add an object to the Recycle Bin. Perform the following steps:

1. Open **SQL*Plus** and enter the following commands to create a new table:

connect system/<password>
@create_job_hist

```
Terminal                                                    _ □ ×

File   Edit   Settings   Help

bash-2.05$ cd wkdir
bash-2.05$ sqlplus system/oracle

SQL*Plus: Release 10.1.0.1.0 - Beta on Wed Sep 17 12:50:43 2003

Copyright (c) 1982, 2003, Oracle.  All rights reserved.


Connected to:
Oracle10i Enterprise Edition Release 10.1.0.1.0 - Beta
With the Partitioning, OLAP and Data Mining Scoring Engine options

SQL> @create_job_hist.sql

Table created.

SQL>
```

2. Now enter the following SQL statement to drop the table you just created.

drop table hr.jobs_hist;

```
 ▤  Terminal                                                    _ □ X
  File   Edit   Settings   Help

 bash-2.05$ cd wkdir
 bash-2.05$ sqlplus system/oracle

 SQL*Plus: Release 10.1.0.1.0 - Beta on Wed Sep 17 12:50:43 2003

 Copyright (c) 1982, 2003, Oracle.  All rights reserved.


 Connected to:
 Oracle10i Enterprise Edition Release 10.1.0.1.0 - Beta
 With the Partitioning, OLAP and Data Mining Scoring Engine options

 SQL> @create_job_hist.sql

 Table created.

 SQL> drop table hr.jobs_hist;

 Table dropped.

 SQL> █      I          ▷
```

The table has been dropped. You have just added this table to the recycle bin.

Querying Tables in the Recycle Bin

You can still query tables after you have dropped them. You can issue a SELECT statement against the objects in the recycle bin using the name of the object in the recycle bin. You can also issue a Flashback Query against objects in the recycle bin to query past data, as long as the undo information required to satisfy the query is available. You can only issue these queries if you are the original owner of the object. Perform the following

1. At this point, you can query the recycle bin for the table you just dropped. Open another browser window and enter the following URL:

http://<hostname>:5500/em

Enter **sys/<password>** as **SYSDBA** and click **Login** .
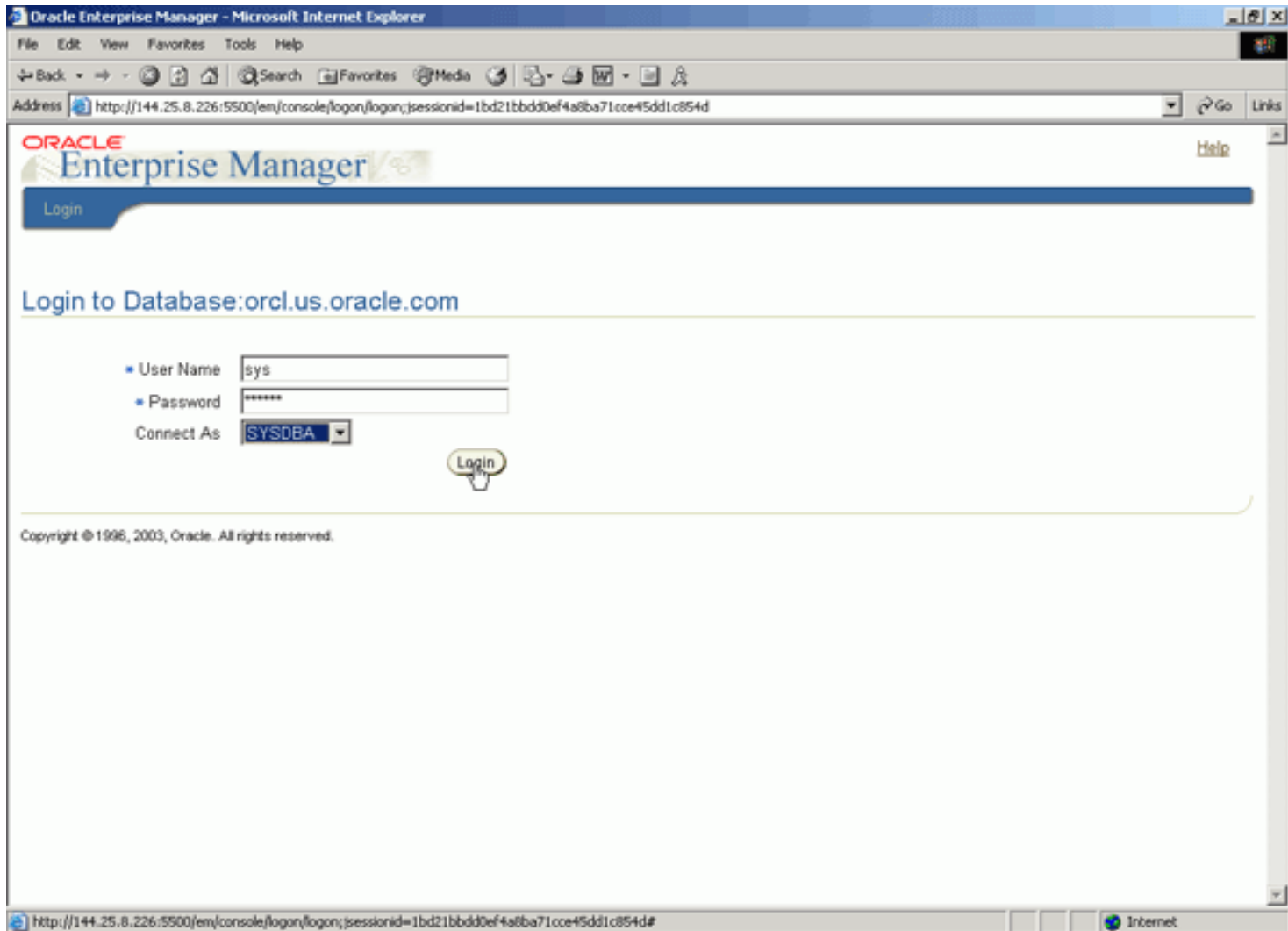
2. Click the **Administration** tab.
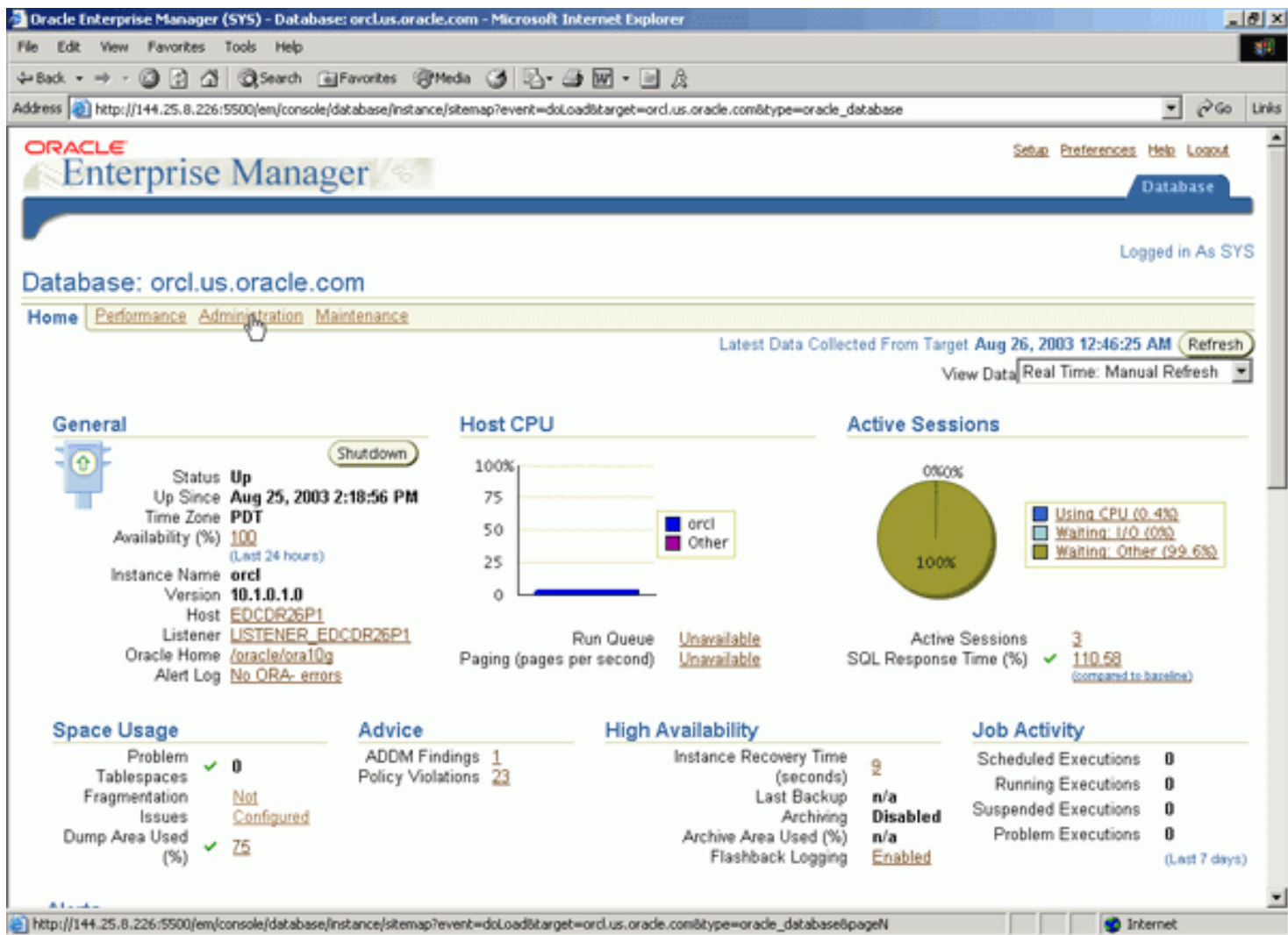
3. Click **Tables** .
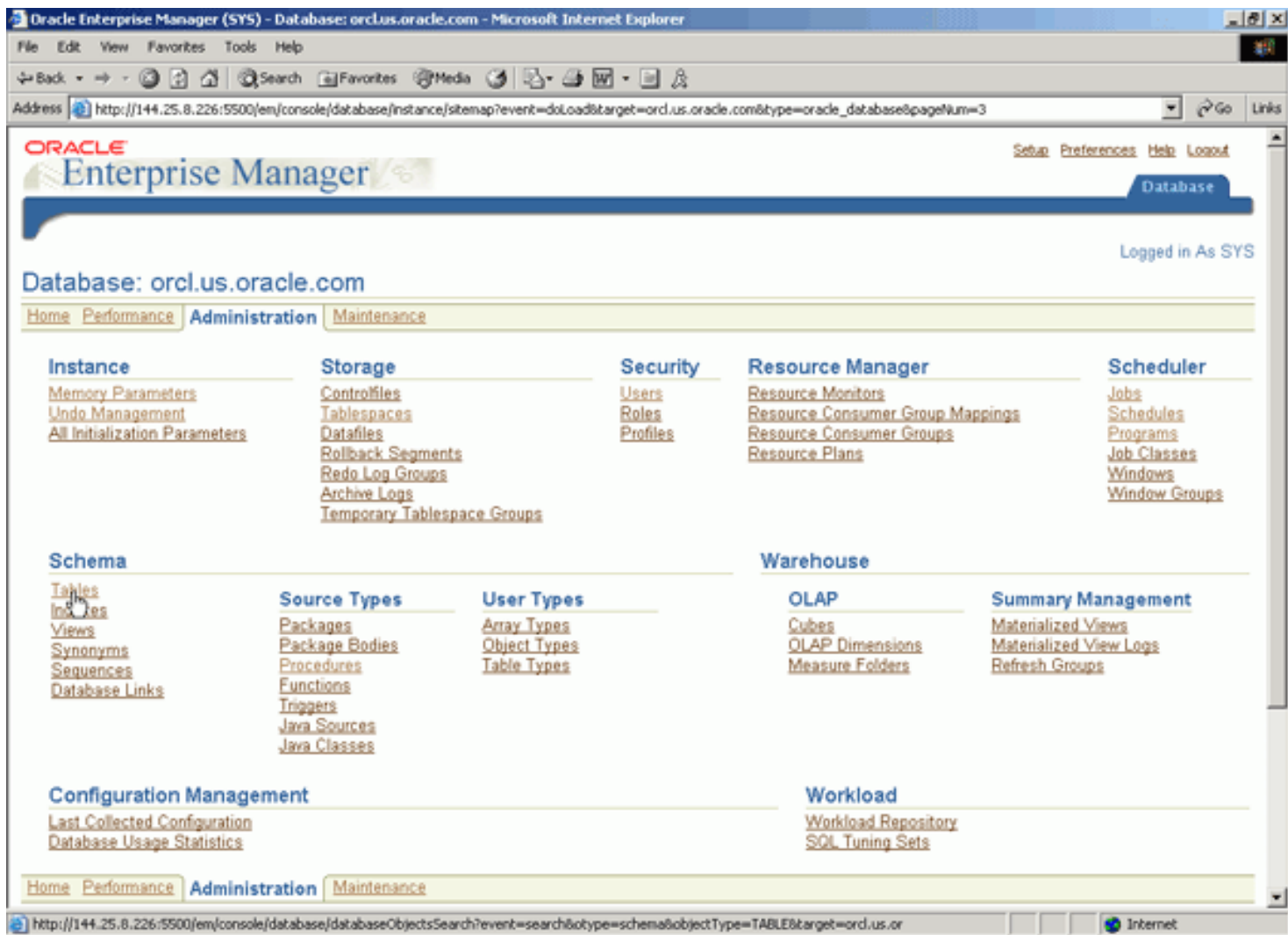
4. Enter **HR** for the Schema and click **GO** .

5. Click the **Recycle Bin** link.

6. Select **View Recycle Bin of all users** from the View Options list box and enter the **HR** schema and click **GO** .

7. You notice that the table you dropped is listed. Click **View Content** .

8. You see the contents of the table you dropped. Click **OK** .

Recovering Tables From the Recycle Bin

You can recover the table from the recycle bin by performing the following:

1. In the recycle bin, select the checkbox next to the JOBS_HIST object and click **Flashback Drop.**
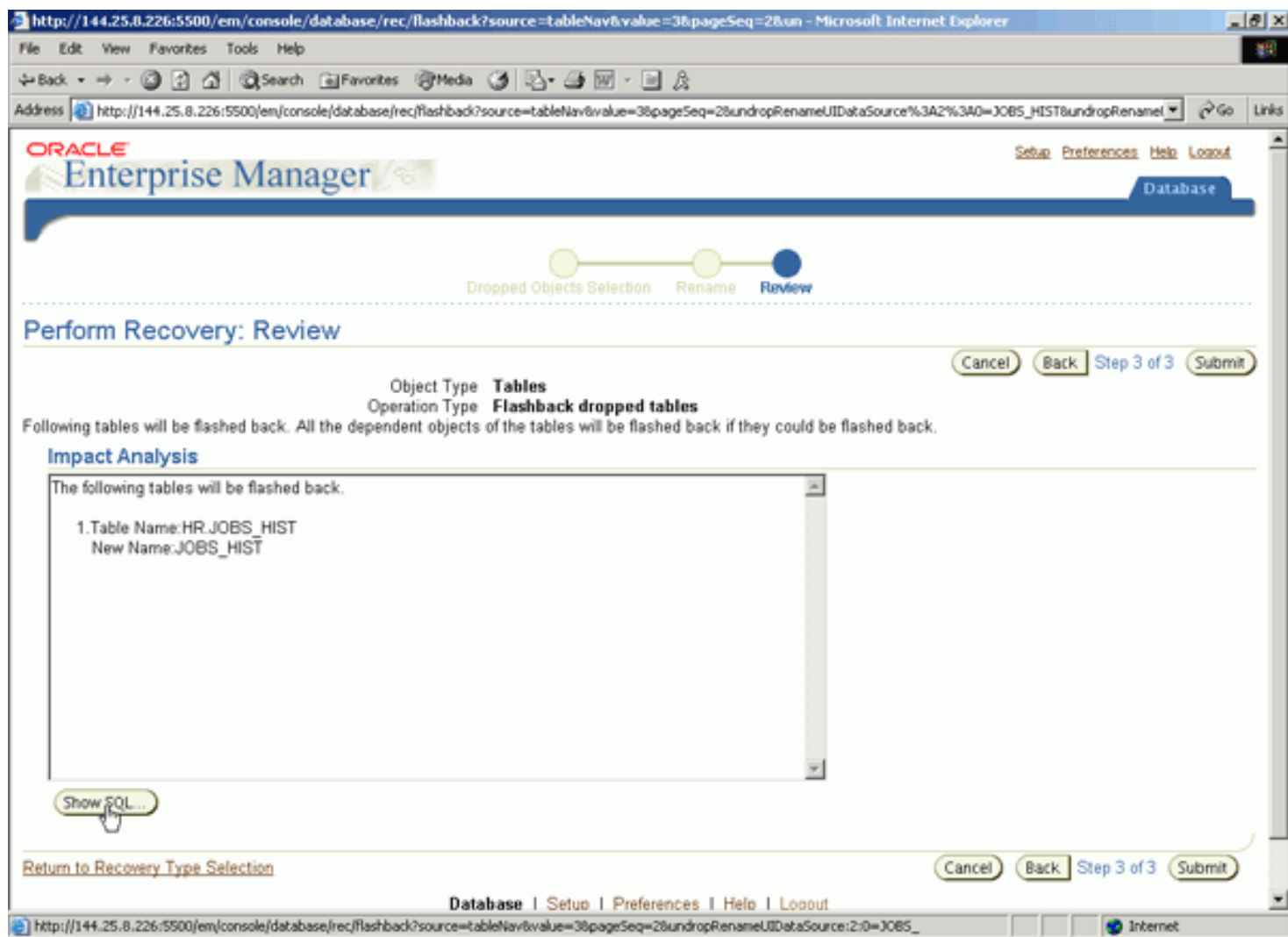

2. To recover a table from the recycle bin, you are required to recover all dependent objects as well. In this case there are no dependencies. Click **Continue** .
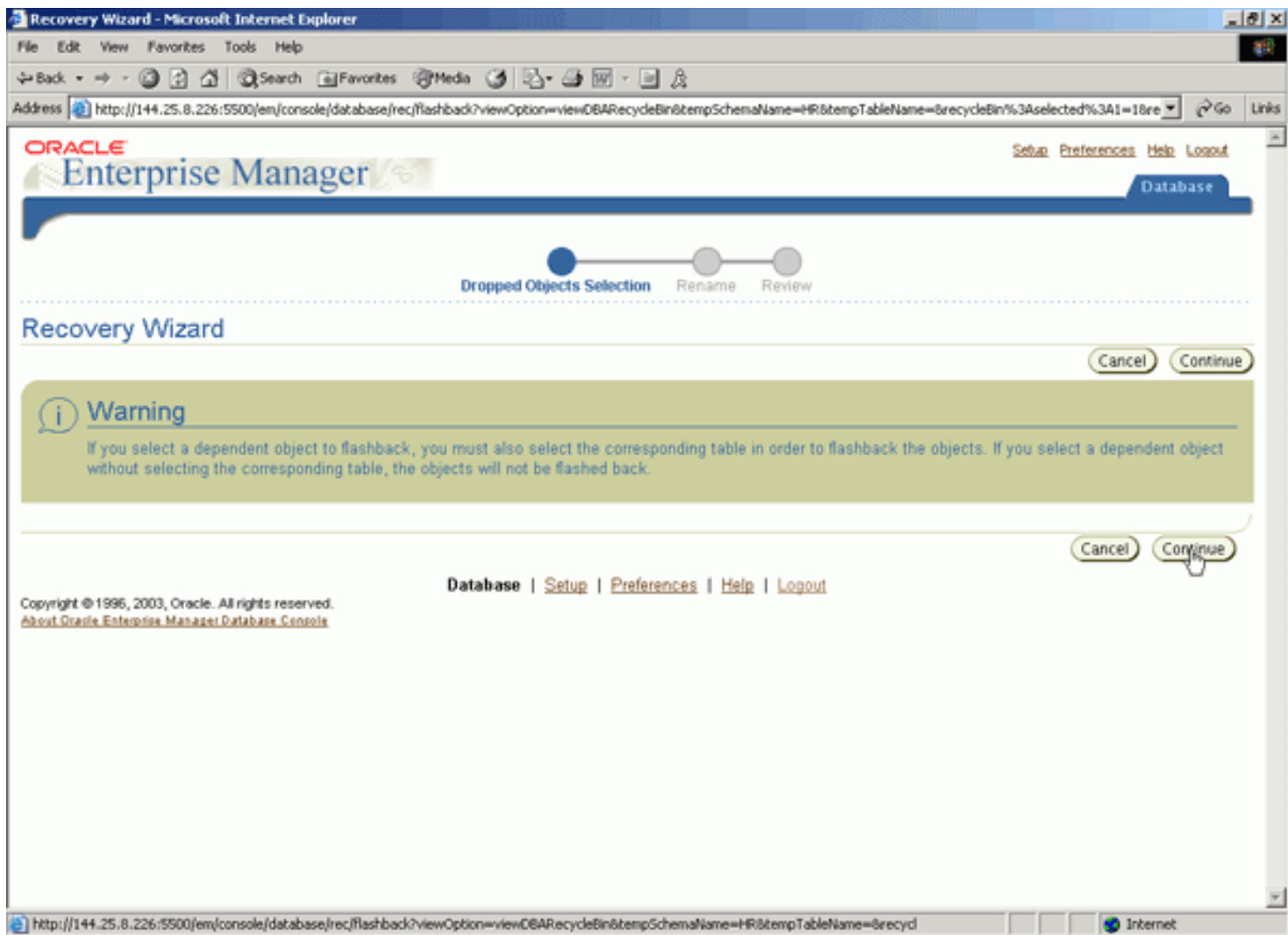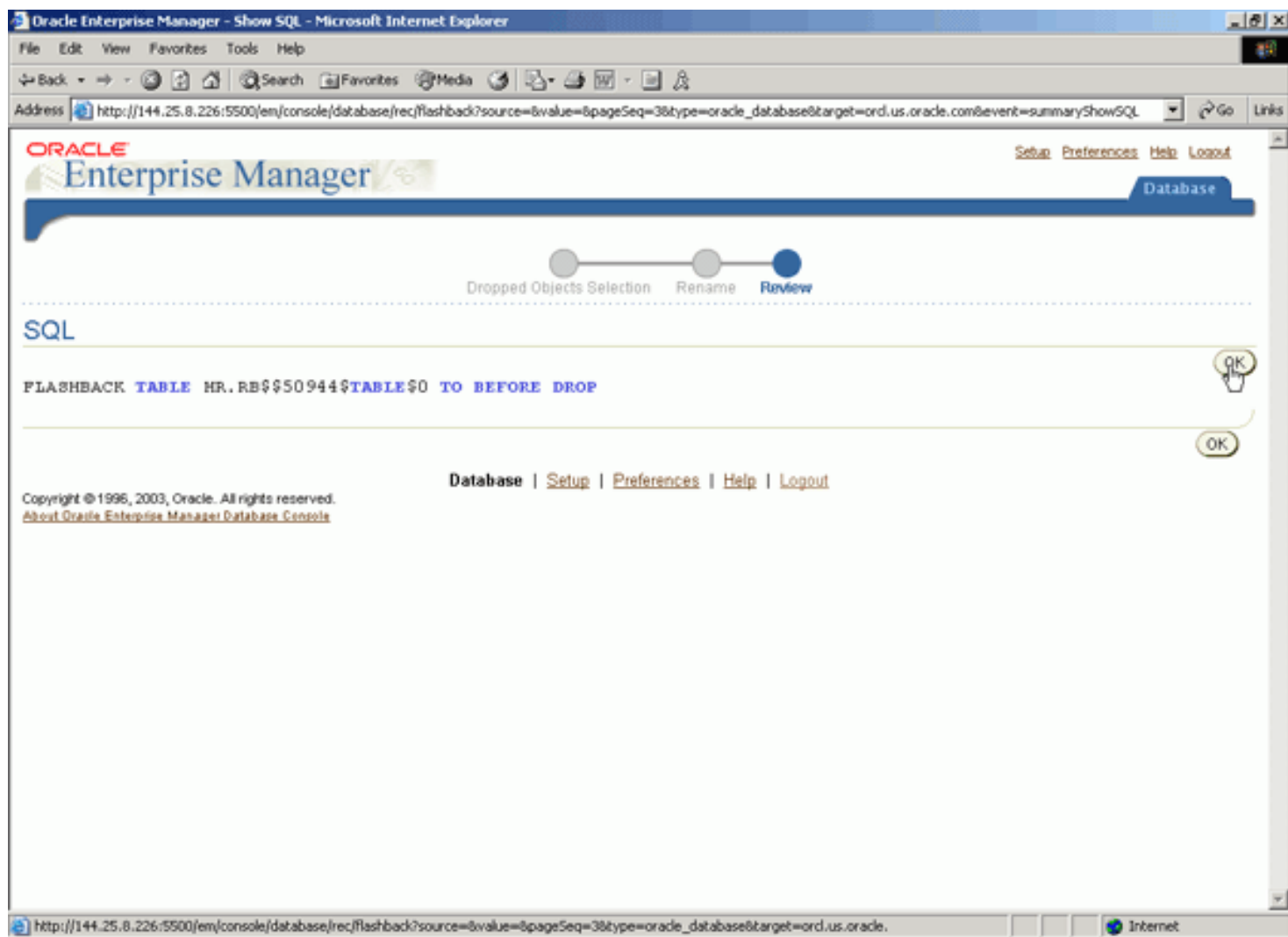
3. Click **Next** .
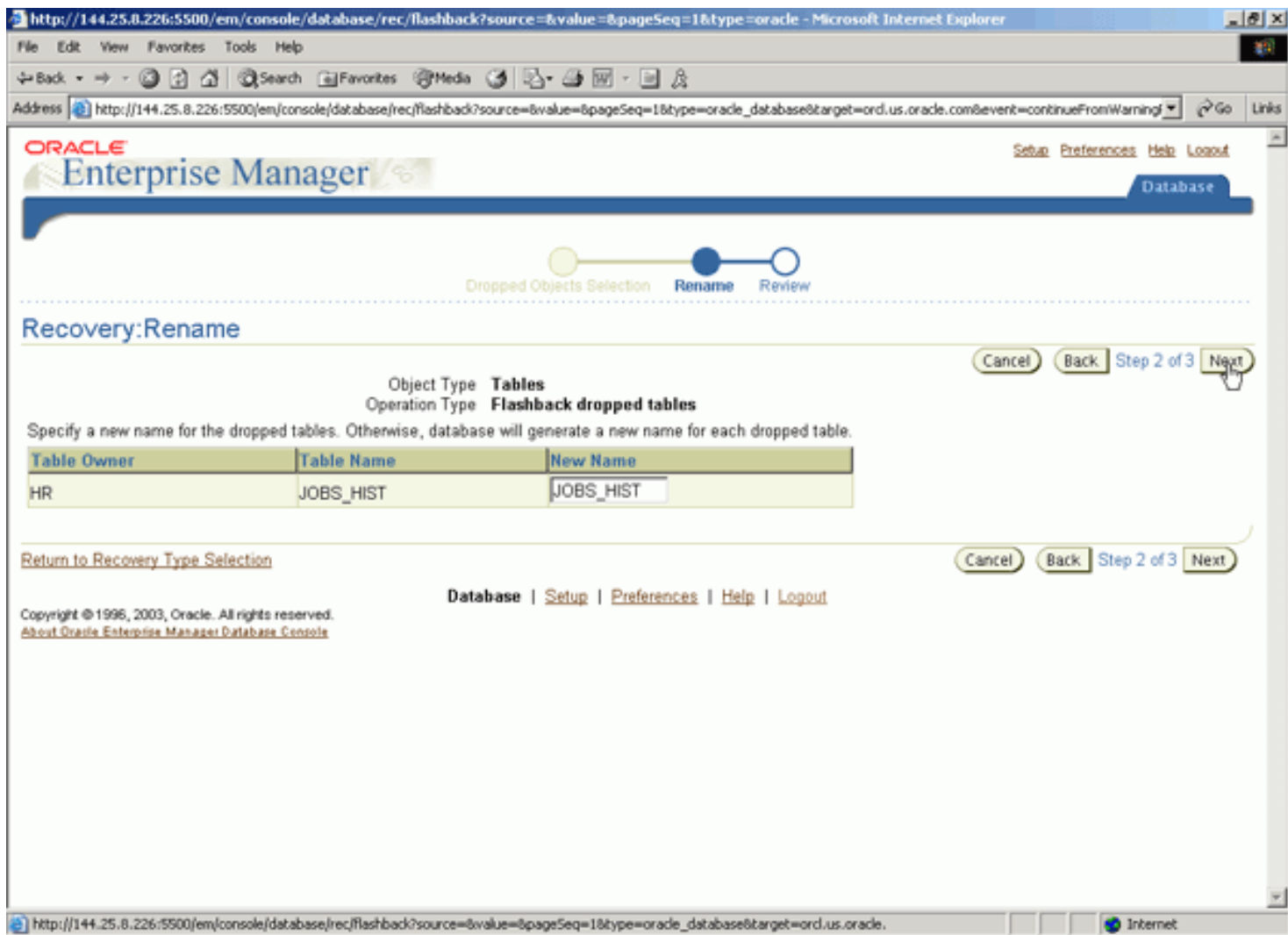
4. Before you submit the job to recover the table, click **Show SQL...** to see what will be executed.
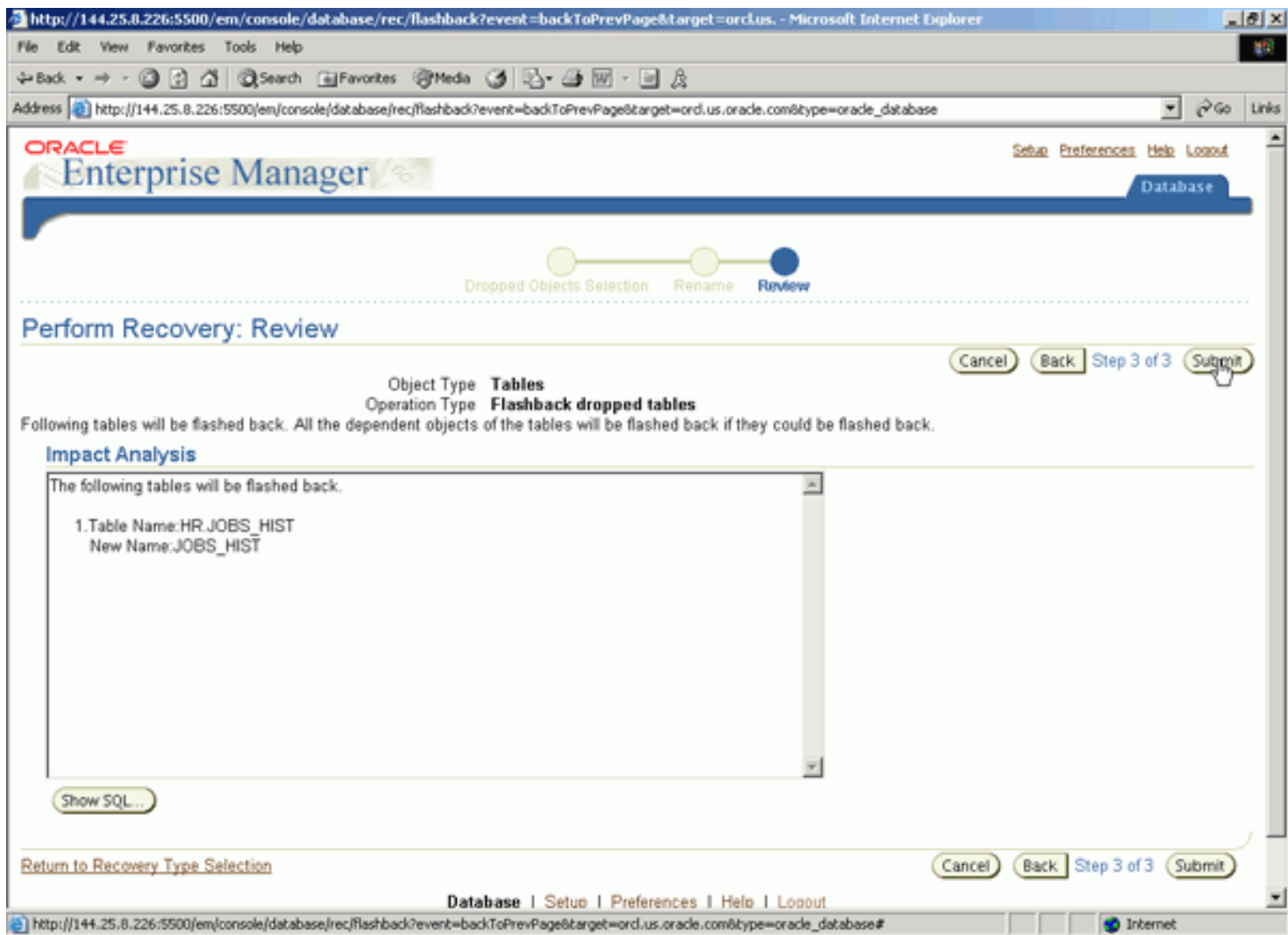
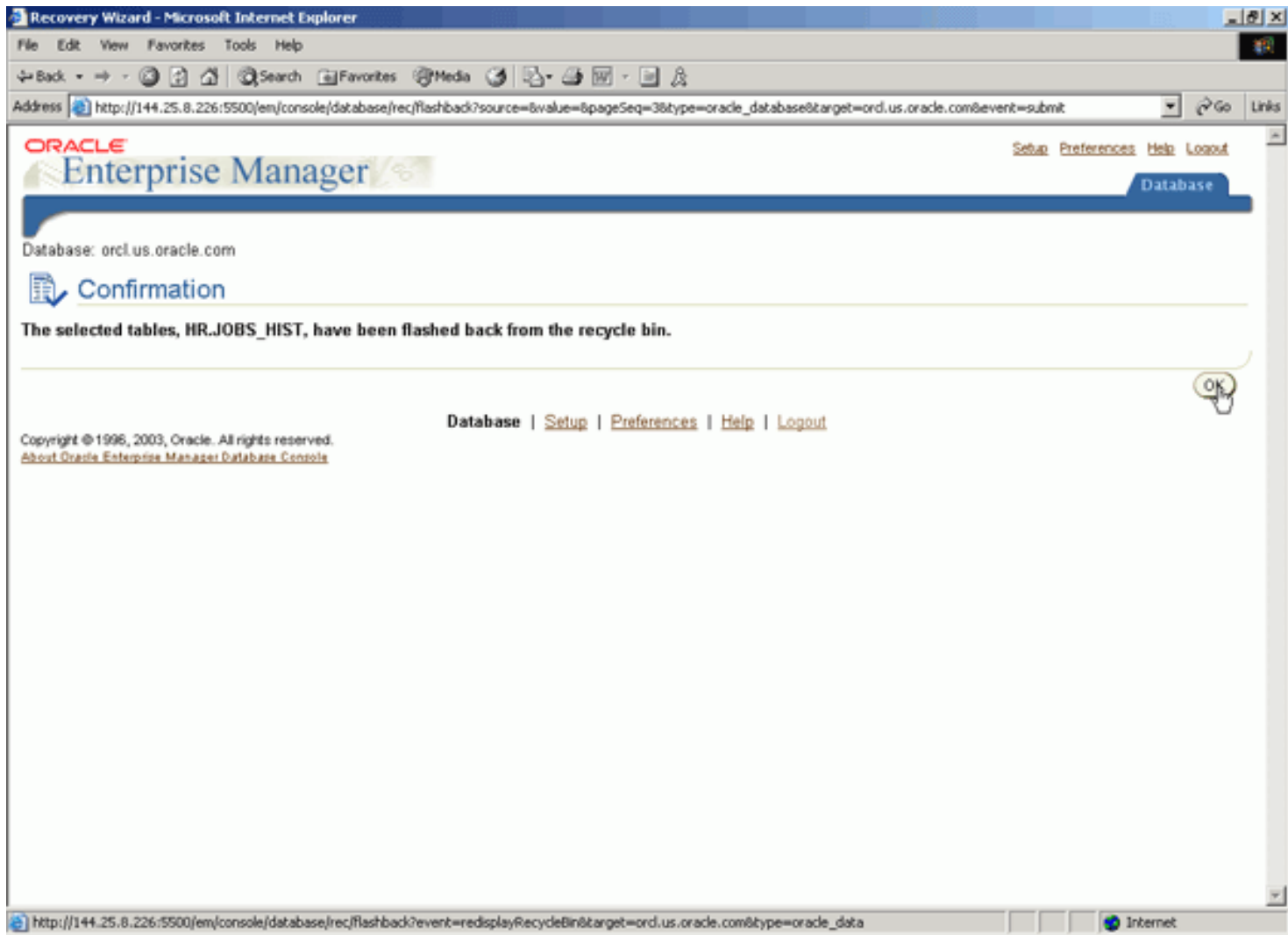**5.** Review the SQL statement that will run. When done, click **OK** .
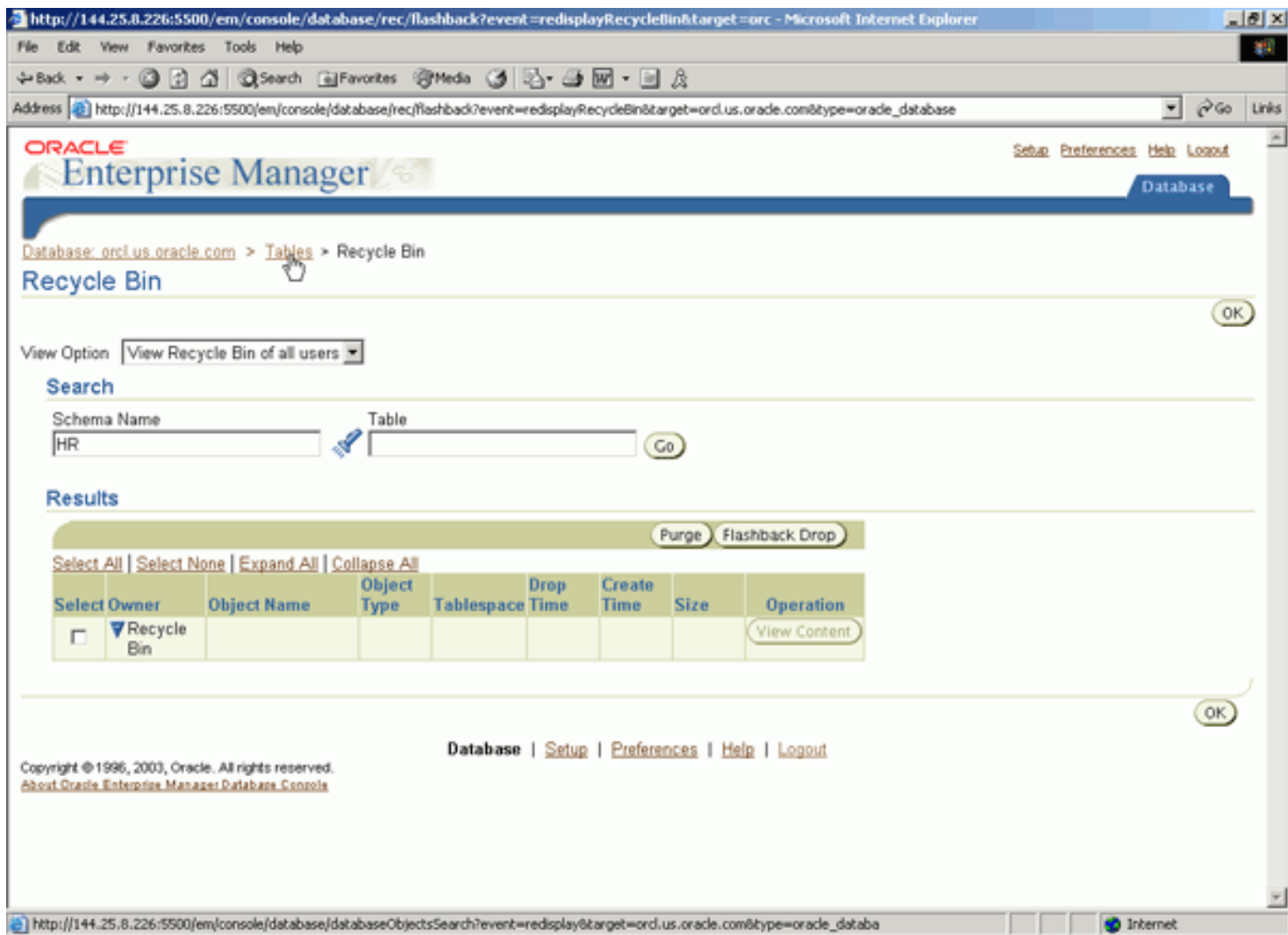
6. Click **Submit** .

7. You receive a message saying the flashback drop was successful. Click **OK** .

8. The JOBS_HIST table no longer appears in the Recycle Bin. You want to make sure that the table now appears in the list. Click the **Tables** breadcrumb.

9. You now see the JOBS_HIST back in the list.

Emptying the Recycle Bin

At some point, you may want to permanently remove the object from the database and release the space allocated to it after you have dropped the object. To remove an object from the recycle bin, perform the following:

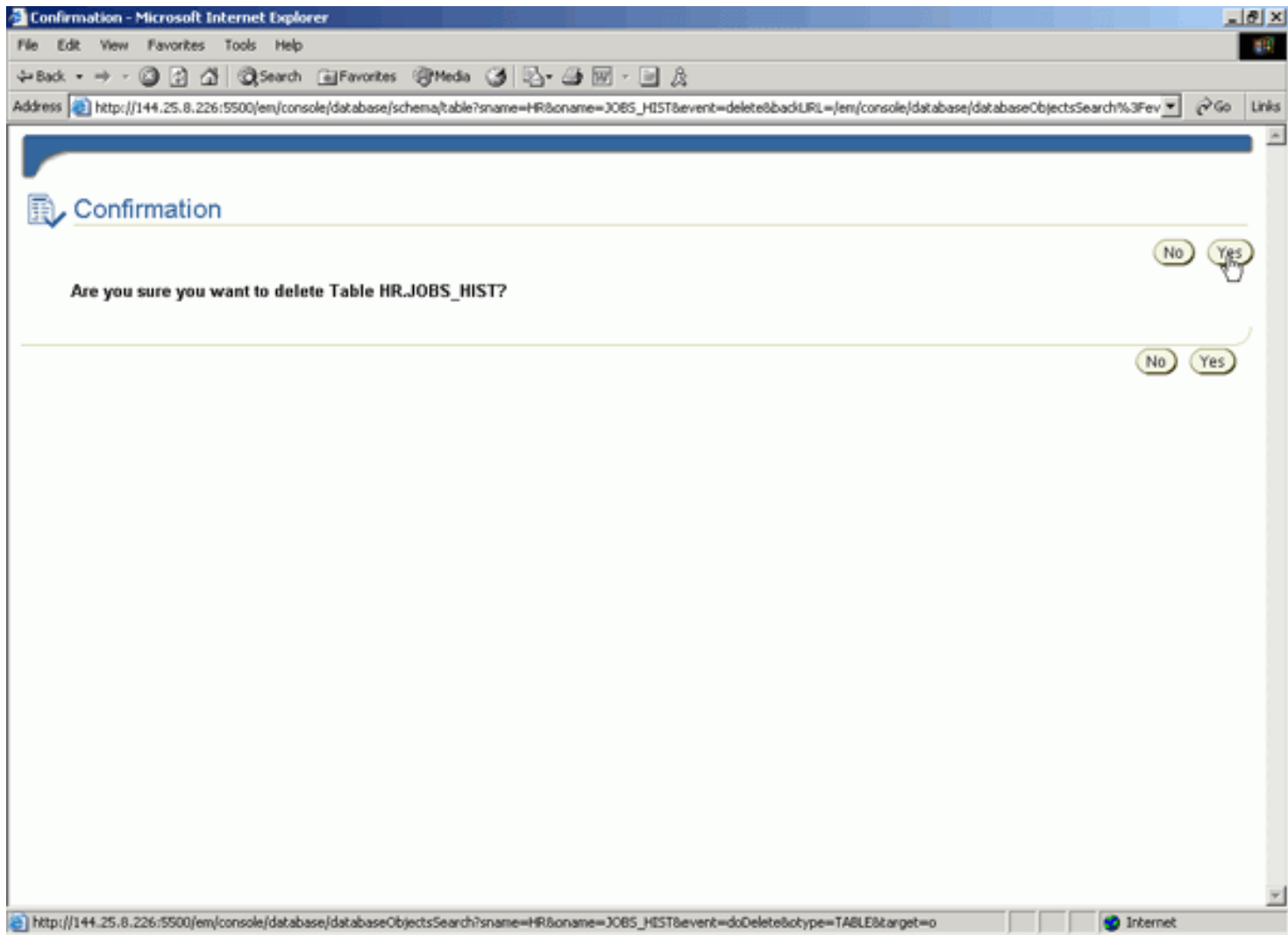1. Select the radio button in front of the **JOBS_HIST** table and click **Delete.**

2. Click **Yes** to confirm deletion.

3. Click the **Recycle Bin** link.

4. Select **View Recycle Bin of all users** from the View Options list box and enter the **HR** schema and click **GO** .

5. In the recycle bin, select the checkbox next to the JOBS_HIST object and click **Purge.**
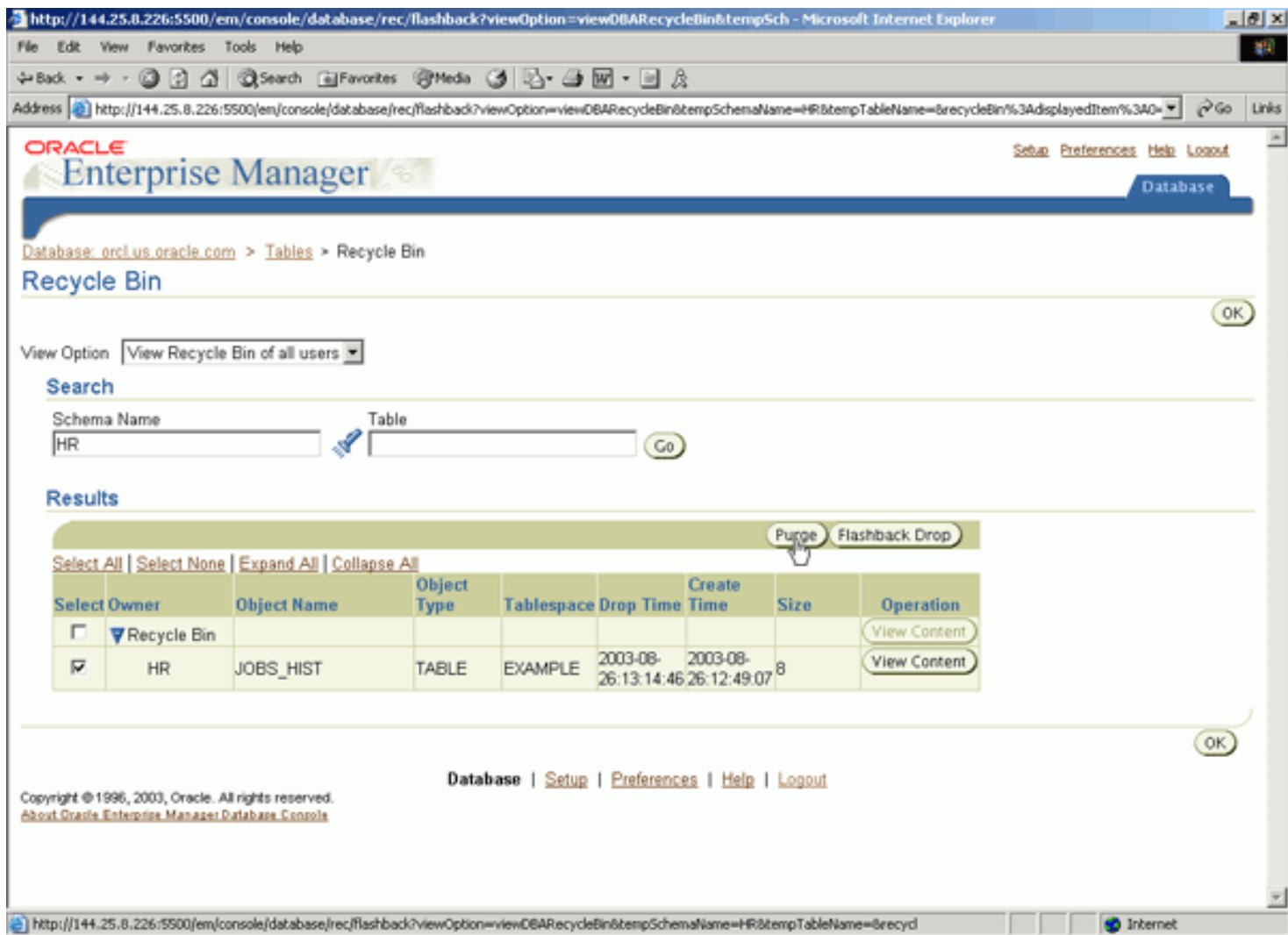
6. Click **Continue** to confirm deletion.

7. The table has been purged from the recycle bin. Click your **Tables** breadcrumb **. .**

Using Flashback Versions Query to Recover a Table to a Point in Time

In Oracle9 *i* , Flashback Query was introduced. This feature provides a way for you to view and repair historical data. You can perform queries on the database as of a certain wall clock time or user-specified system change number (SCN).
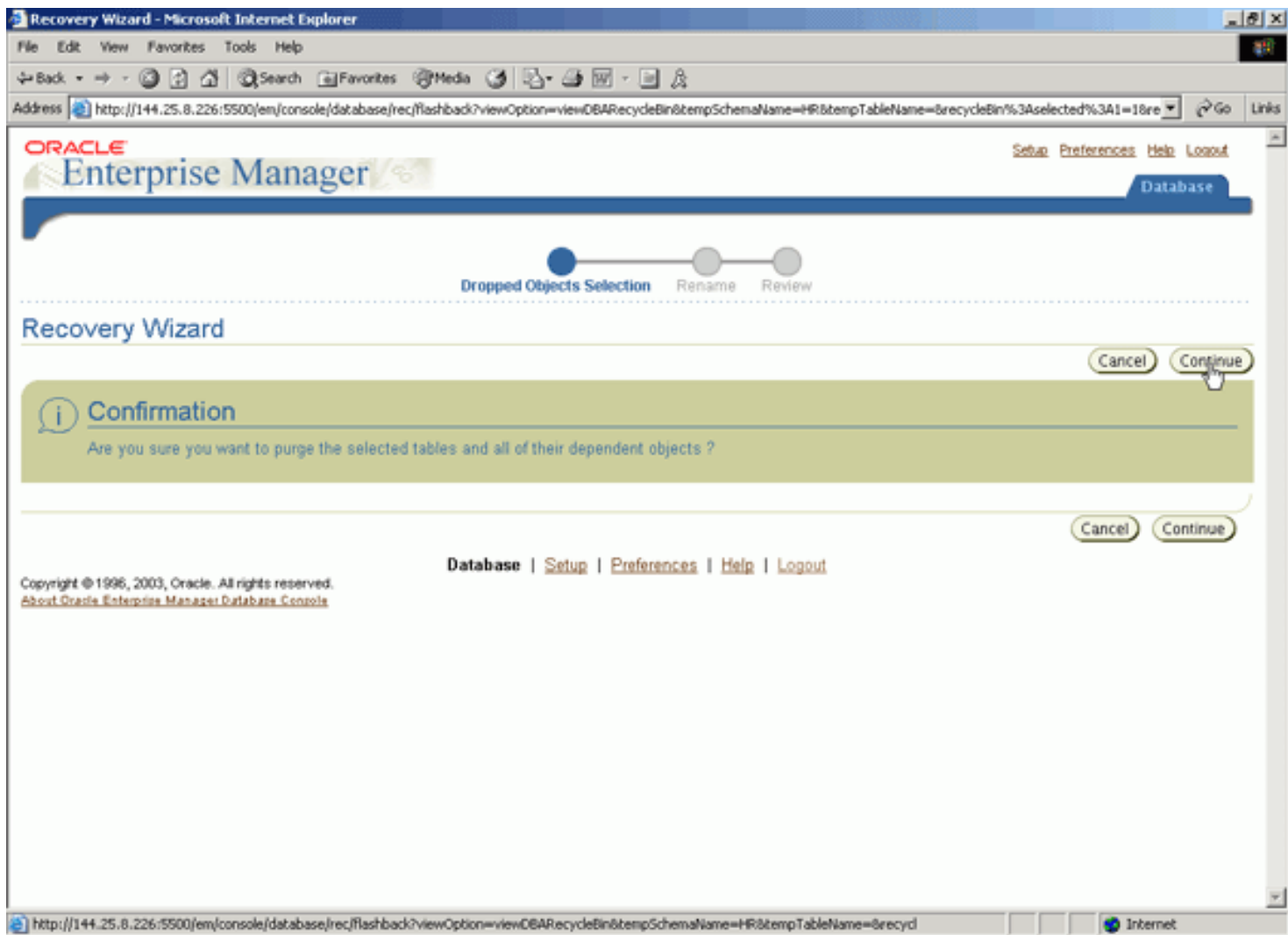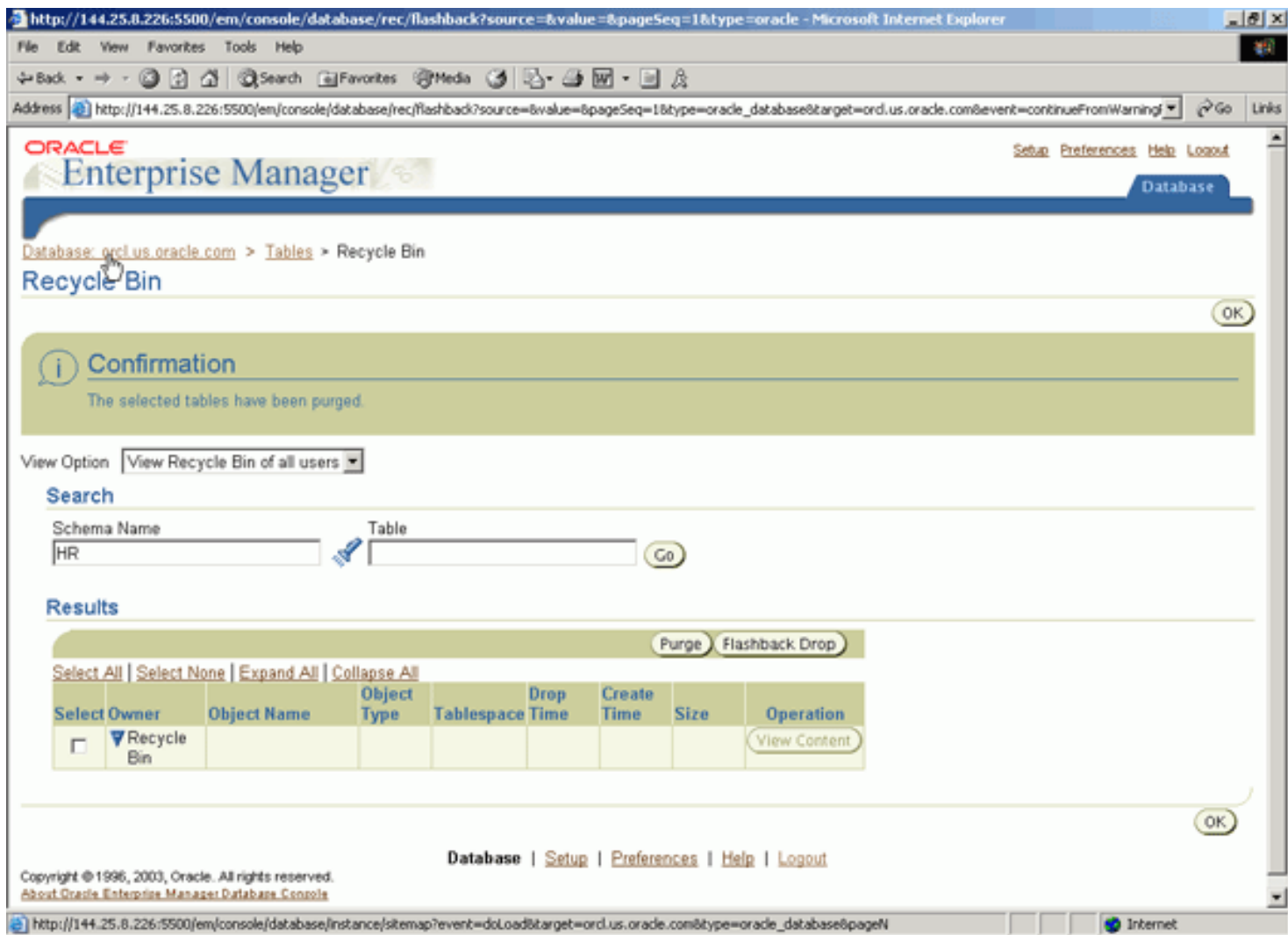
In Oracle Database 10 *G* , Flashback functionality has been expanded. You can use the `VERSIONS BETWEEN` clause to retrieve all of the versions of the rows that exist or ever existed between the time the query was issued and a point back in time. This type of query is referred to as a Flashback Versions Query.

The rows returned by the Flashback Versions Query are a history of changes of the rows across transactions.

The Flashback Versions Query retrieves all committed occurrences of the rows. Changes made by the current active transaction are not returned. In other words, uncommitted row versions within a transaction will not be shown.

You can use Flashback Versions Query to retrieve row history. It provides you with a way to audit the rows of a table and

retrieve information about the transactions that changed the rows. You can use the transaction ID obtained from Flashback Versions Query to perform transaction mining using LogMiner or Flashback Transaction Query to obtain additional information about the transaction.

You will perform the following tasks:

- [Update the JOBS Table](#)
- [Enable Row Movement on the JOBS Table](#)
- [Recover the JOBS Table to a Particular Point in Time](#)

Update the JOBS Table

[Back to List](#)

To show how flashback works, you need to make a change to a table. Perform the following:

1. From your **SQL*Plus** session, execute the following script logged on as **system** .

   updatejobs.sql

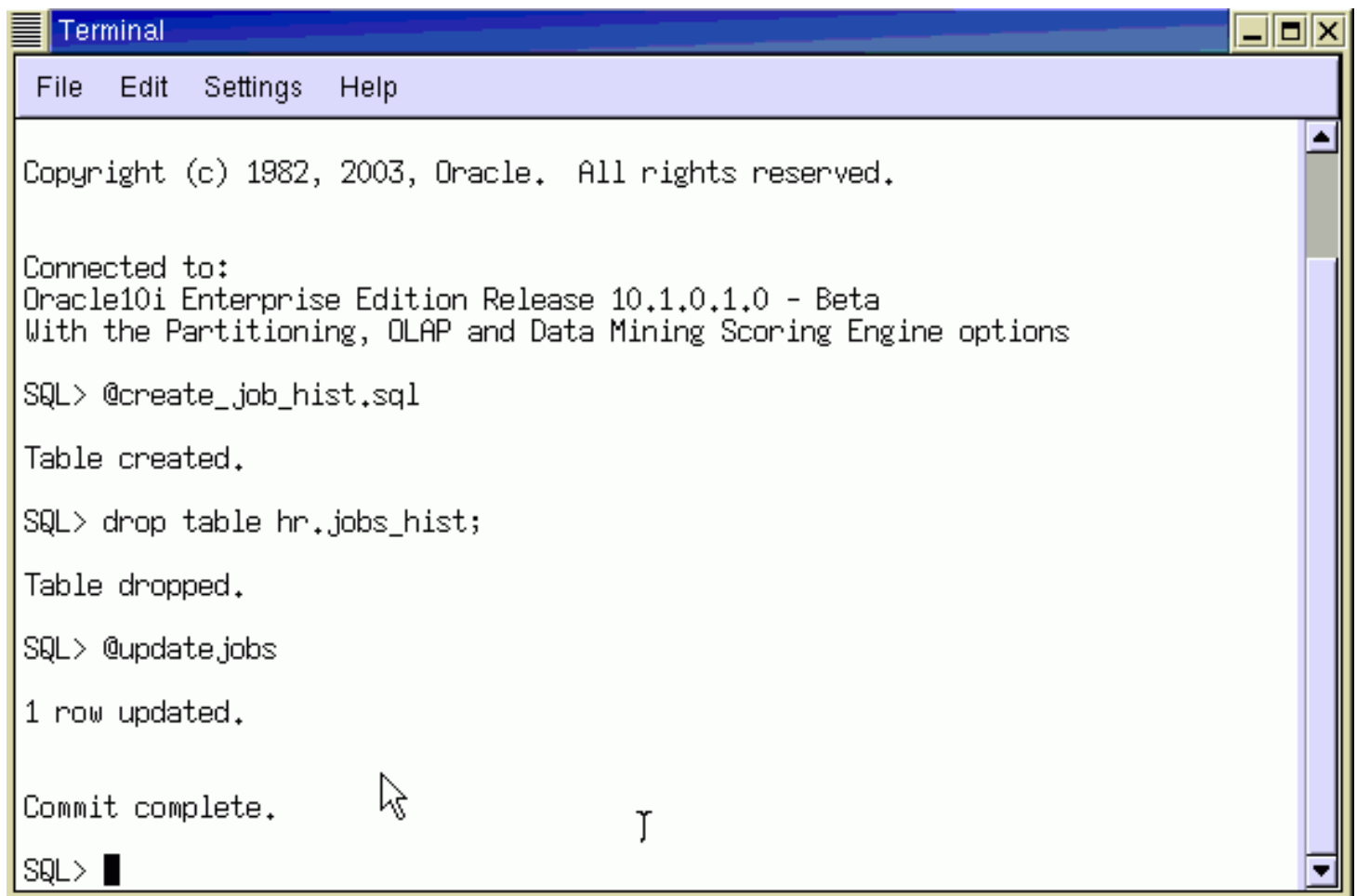   The updatejobs.sql contains the following:

   UPDATE hr.jobs

   SET max_salary = max_salary + 2000

   WHERE job_id = 'IT_PROG'

   /

   COMMIT

   /

```
Terminal                                                    [_][□][X]

  File   Edit   Settings   Help

Copyright (c) 1982, 2003, Oracle.  All rights reserved.


Connected to:
Oracle10i Enterprise Edition Release 10.1.0.1.0 - Beta
With the Partitioning, OLAP and Data Mining Scoring Engine options

SQL> @create_job_hist.sql

Table created.

SQL> drop table hr.jobs_hist;

Table dropped.

SQL> @update.jobs

1 row updated.


Commit complete.

SQL> █
```
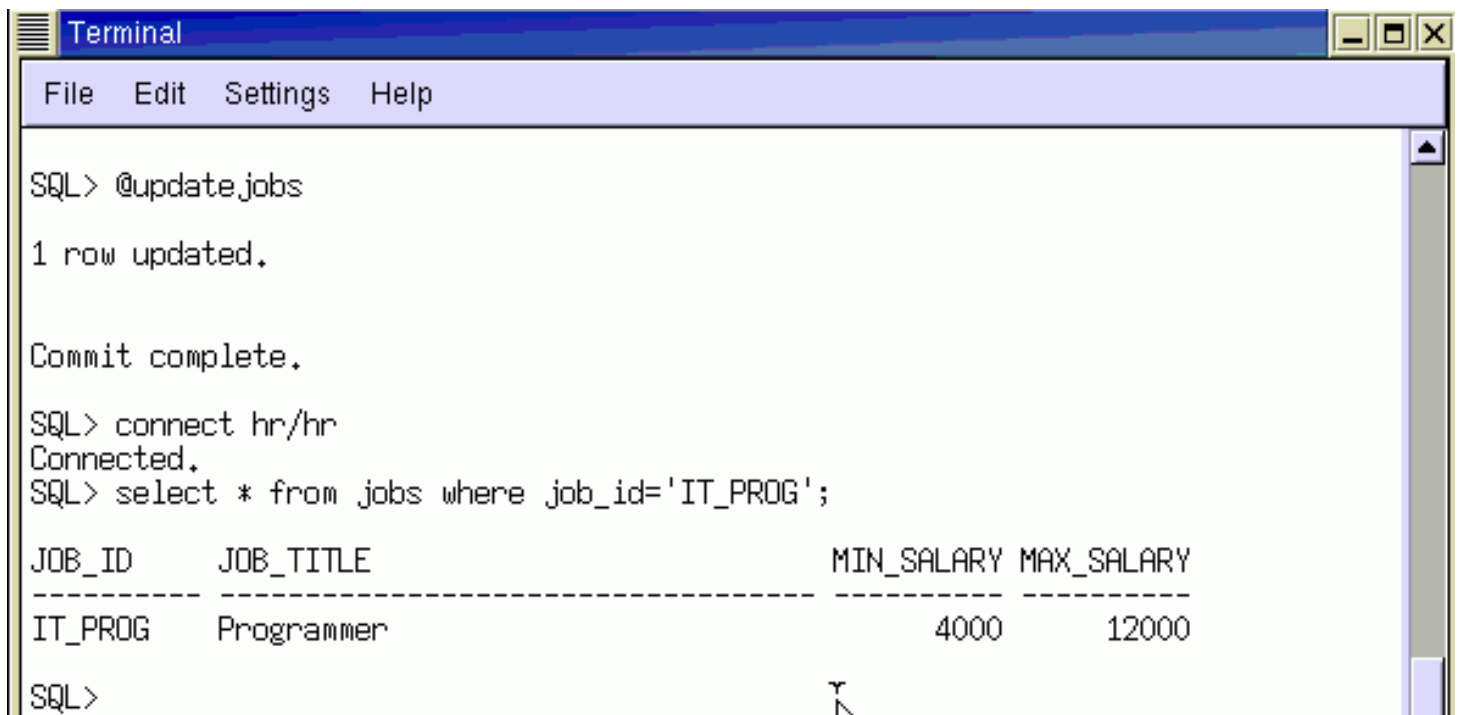
2. To see the current value, enter the following SQL statements:

```
connect hr/hr
select * from jobs where job_id='IT_PROG';
```

```
Terminal                                                    _ □ X

File   Edit   Settings   Help

SQL> @update.jobs

1 row updated.


Commit complete.

SQL> connect hr/hr
Connected.
SQL> select * from jobs where job_id='IT_PROG';

JOB_ID     JOB_TITLE                          MIN_SALARY MAX_SALARY
---------- ---------------------------------- ---------- ----------
IT_PROG    Programmer                               4000      12000

SQL>
```

You see the current value after your update is 12000.

Enable Row Movement on the JOBS Table

[Back to List](#)

You must first enable row movement on the table you want to perform flashback table on.Enabling row movement lets you specify whether Oracle can move a table row. It is possible for a row to move, for example, during table compression or an update operation on partitioned data. Perform the following:

1. Switch to your Enterprise Manager browser session, click on the **JOBS** table.

2. Select the **Options** tab.

3. Select **Yes** for Enable Row Movement and click **Apply** .
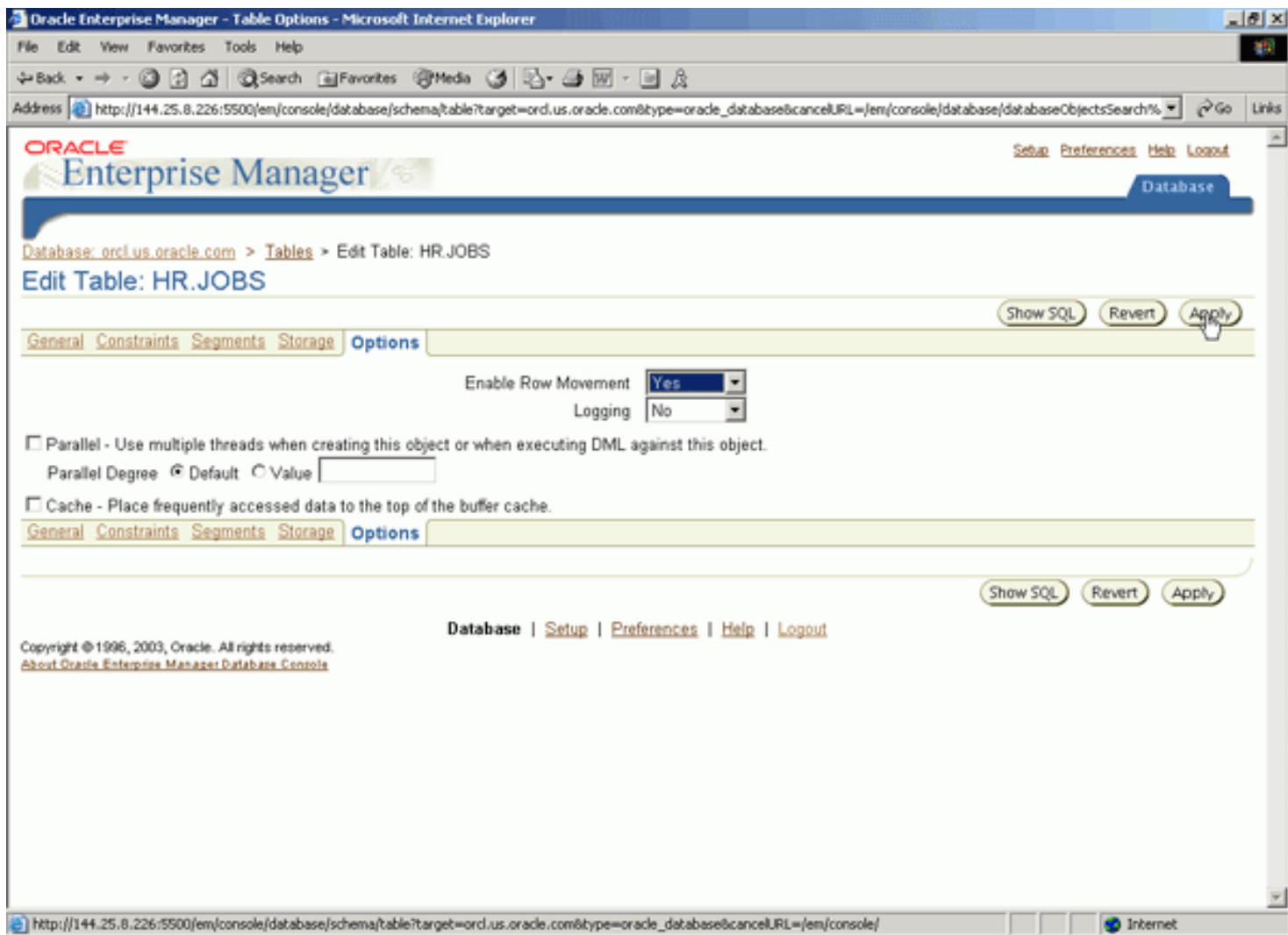
4. Now you can perform a flashback table. Click the **Tables** breadcrumb.
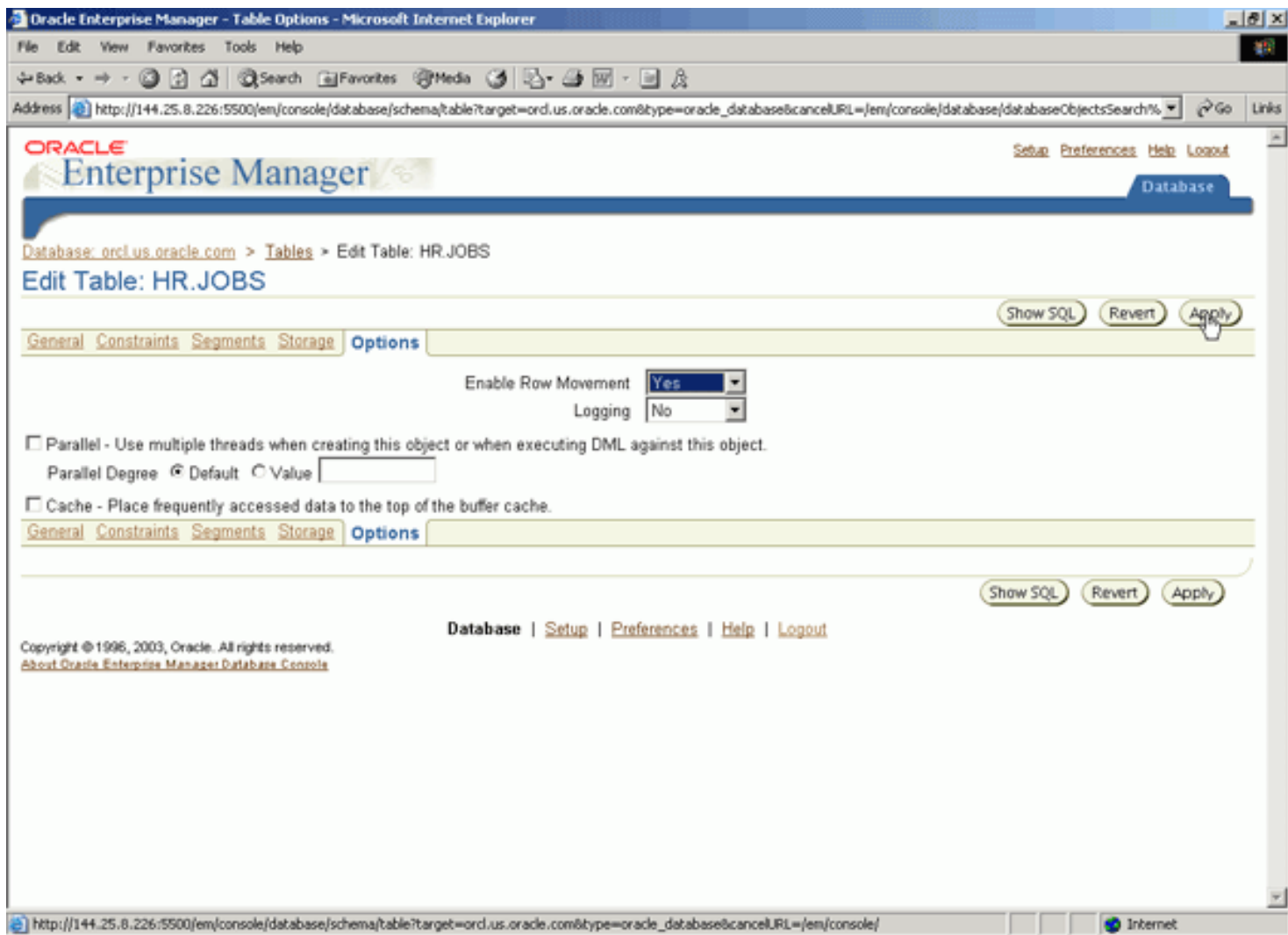
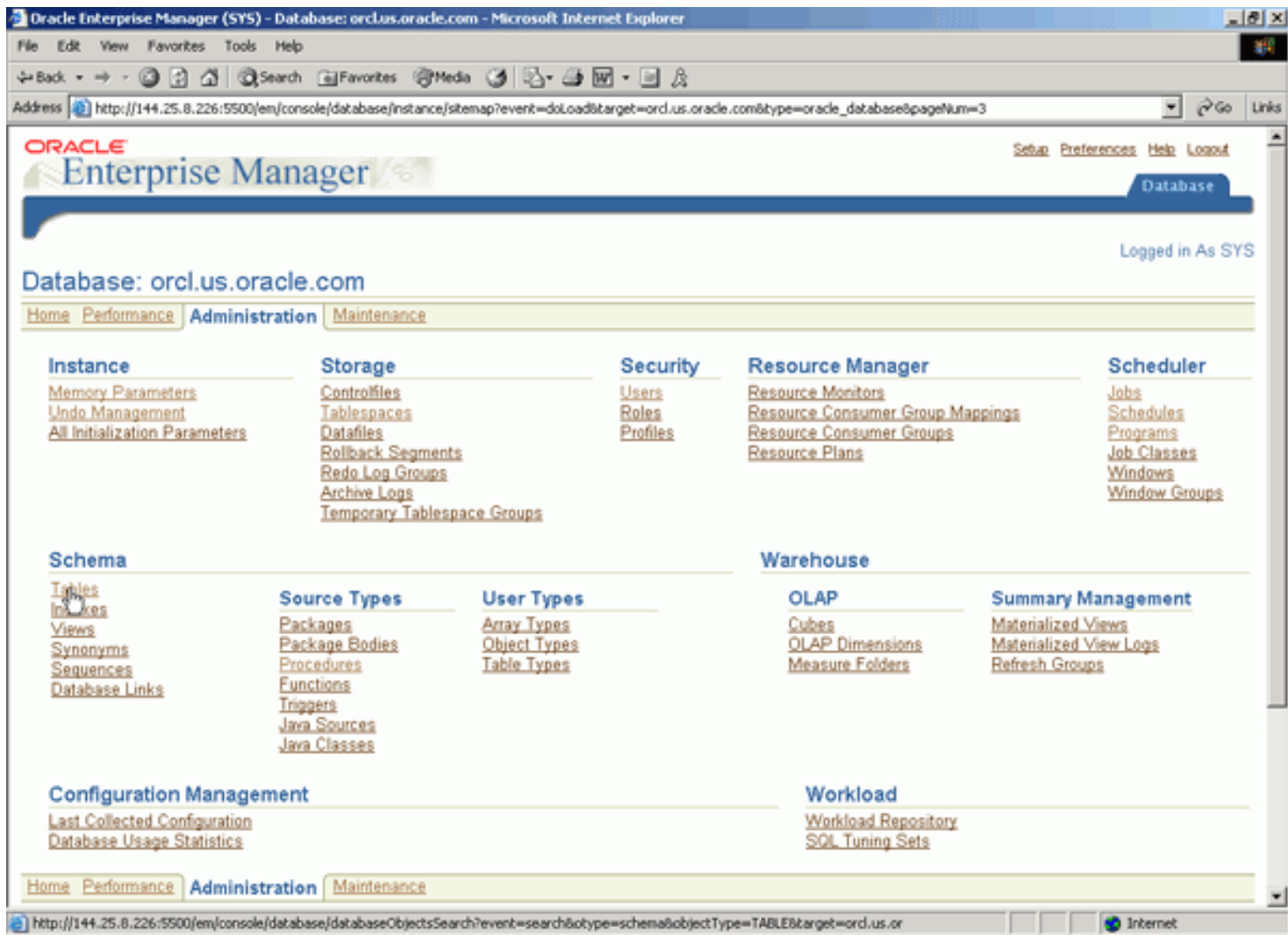**Recover the JOBS Table to a Particular Point in Time**

Now you are ready to recover the JOBS table to a particular point in time. Perform the following:

1. From your Enterprise Manager browser session, click the **Tables** link.

2. Enter **HR** as the Schema and click **GO** .

3. Select the radio button in front of the **JOBS** table and select **Flashback Row History** from the Actions list box and click **GO** .

4. Select **JOB_TITLE** and **MIN_SALARY** and click **Remove** . Then enter **WHERE job_id = 'IT_PROG'** in the Where Clause to Filter Rows field and click **Next** .
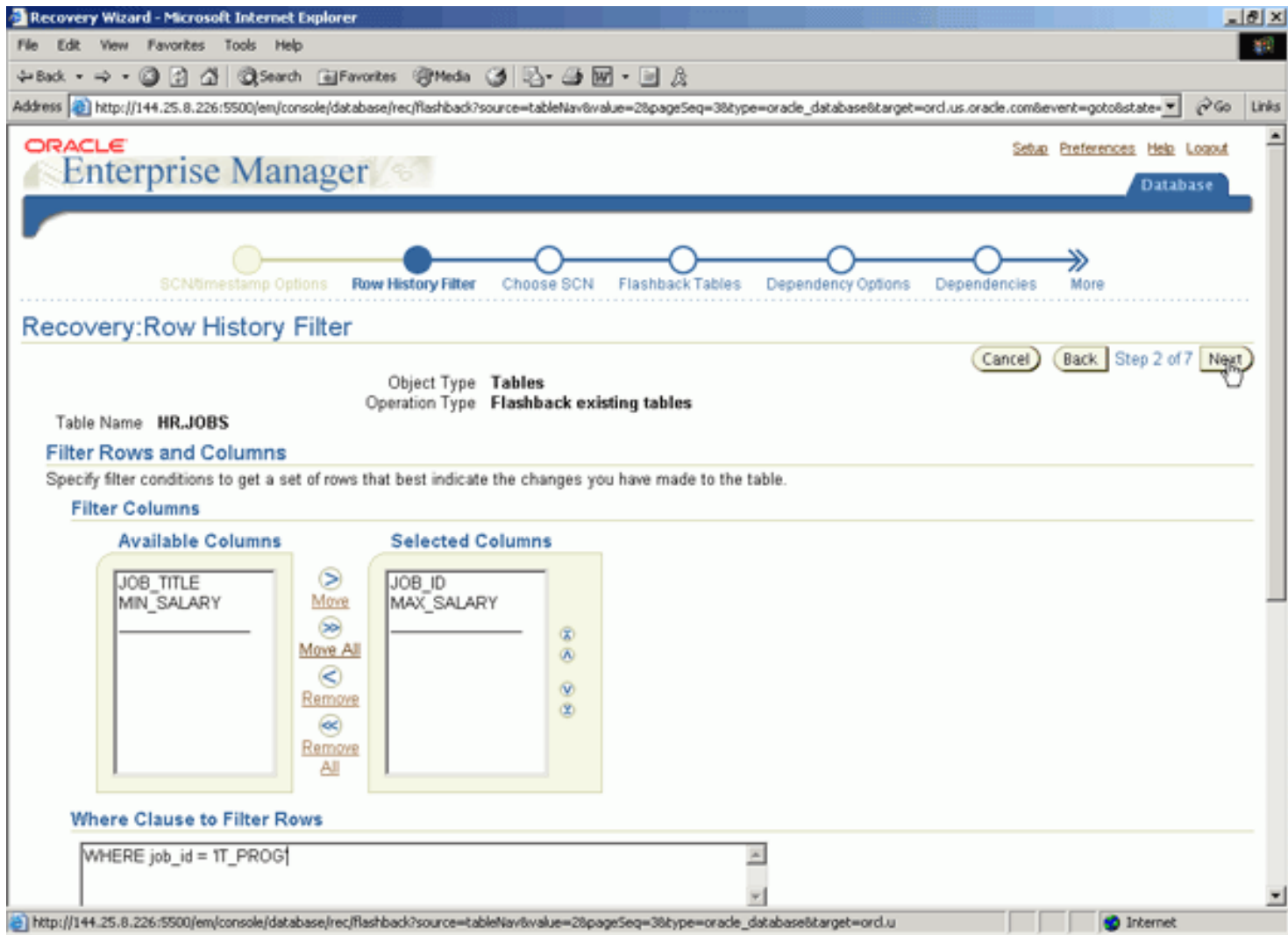
5. The update you made previously is in the list. Click **Next** .

6. Click **Next** .

7. You may want to see what will change. Click **Show Row Changes** .

8. Click **Continue** .

9. You will see the objects that will change as a result of the flashback operation. Click **OK** .

You are now ready to perform the flashback operation. Click **Submit** .

10.

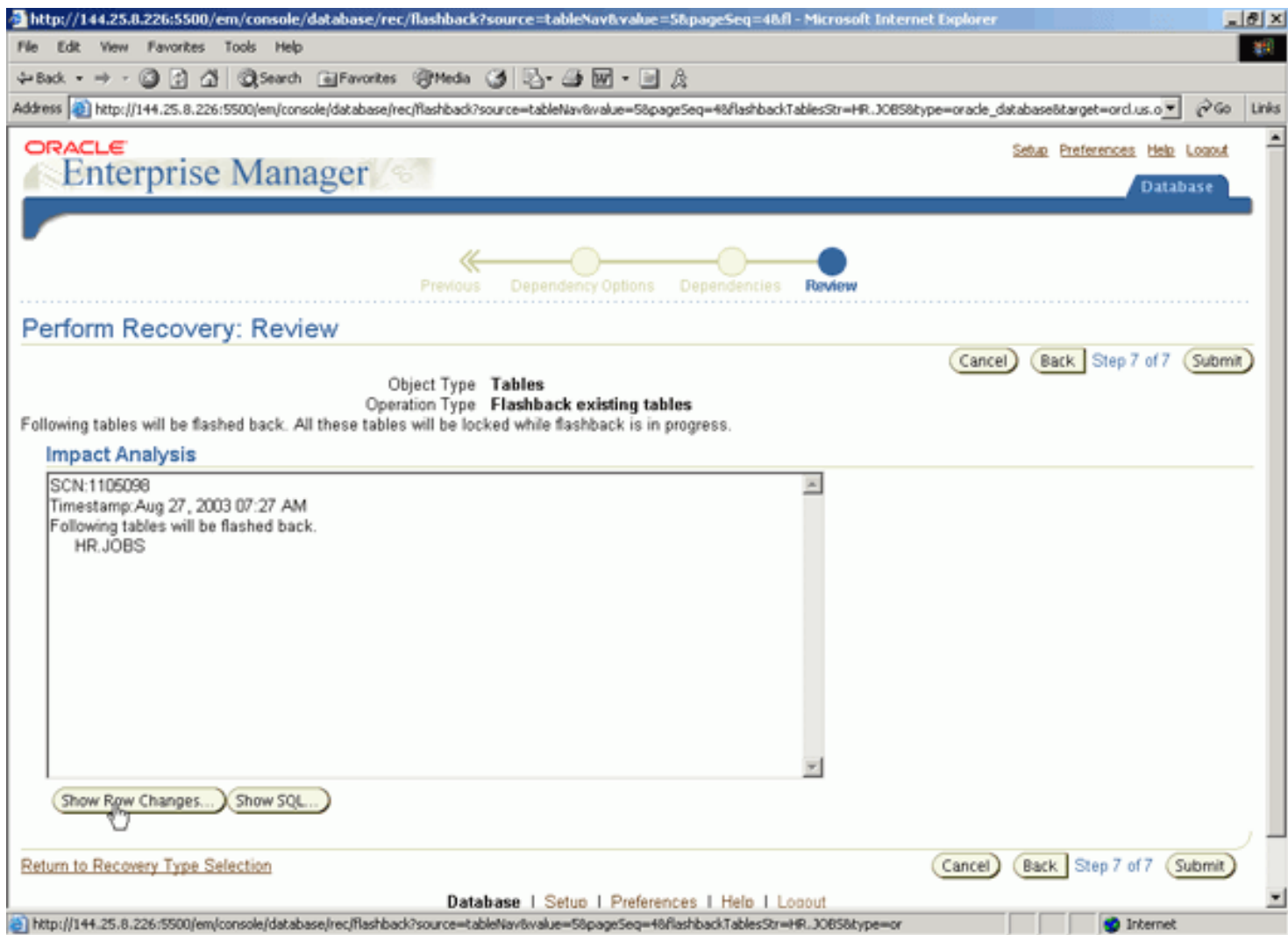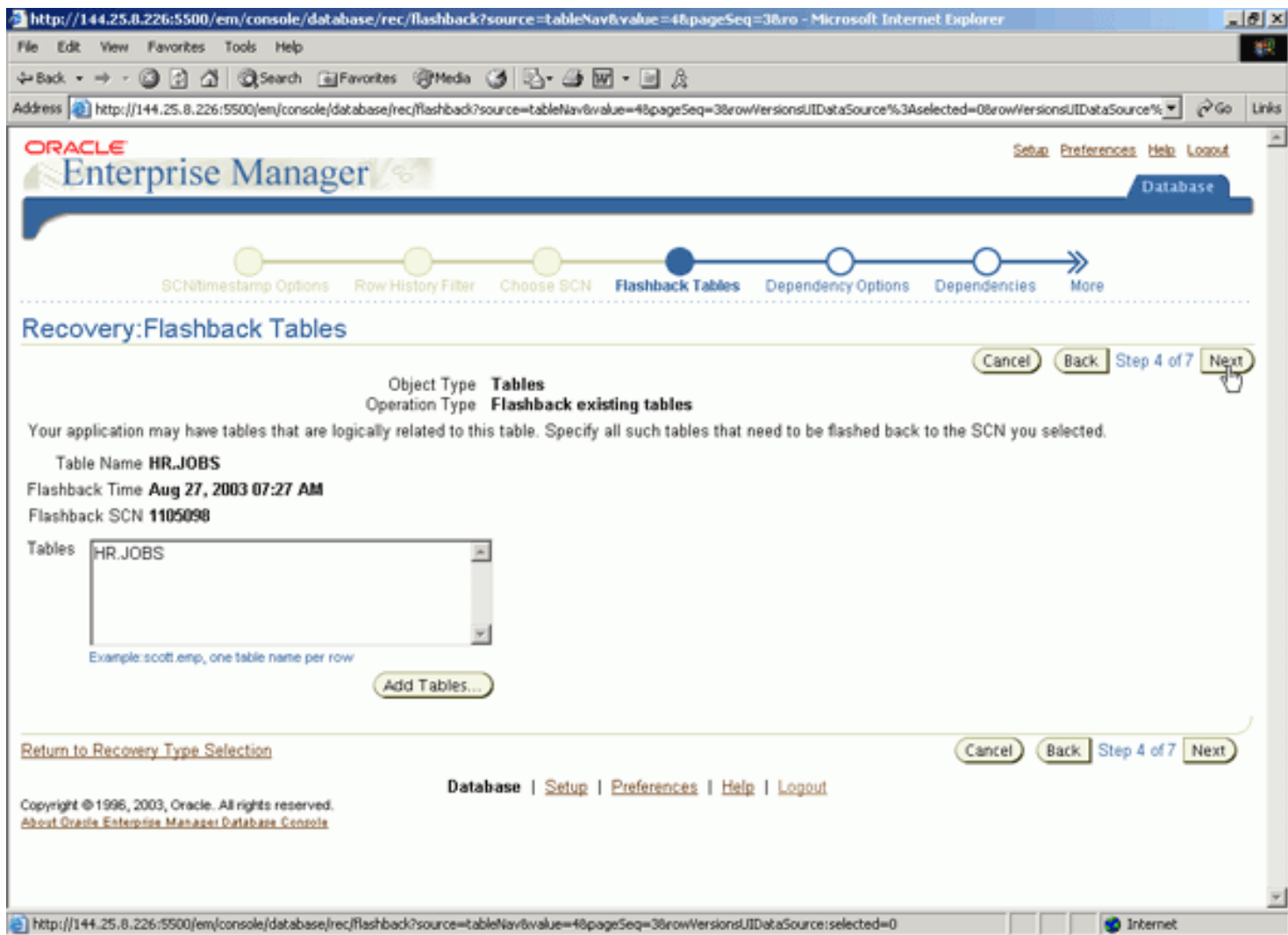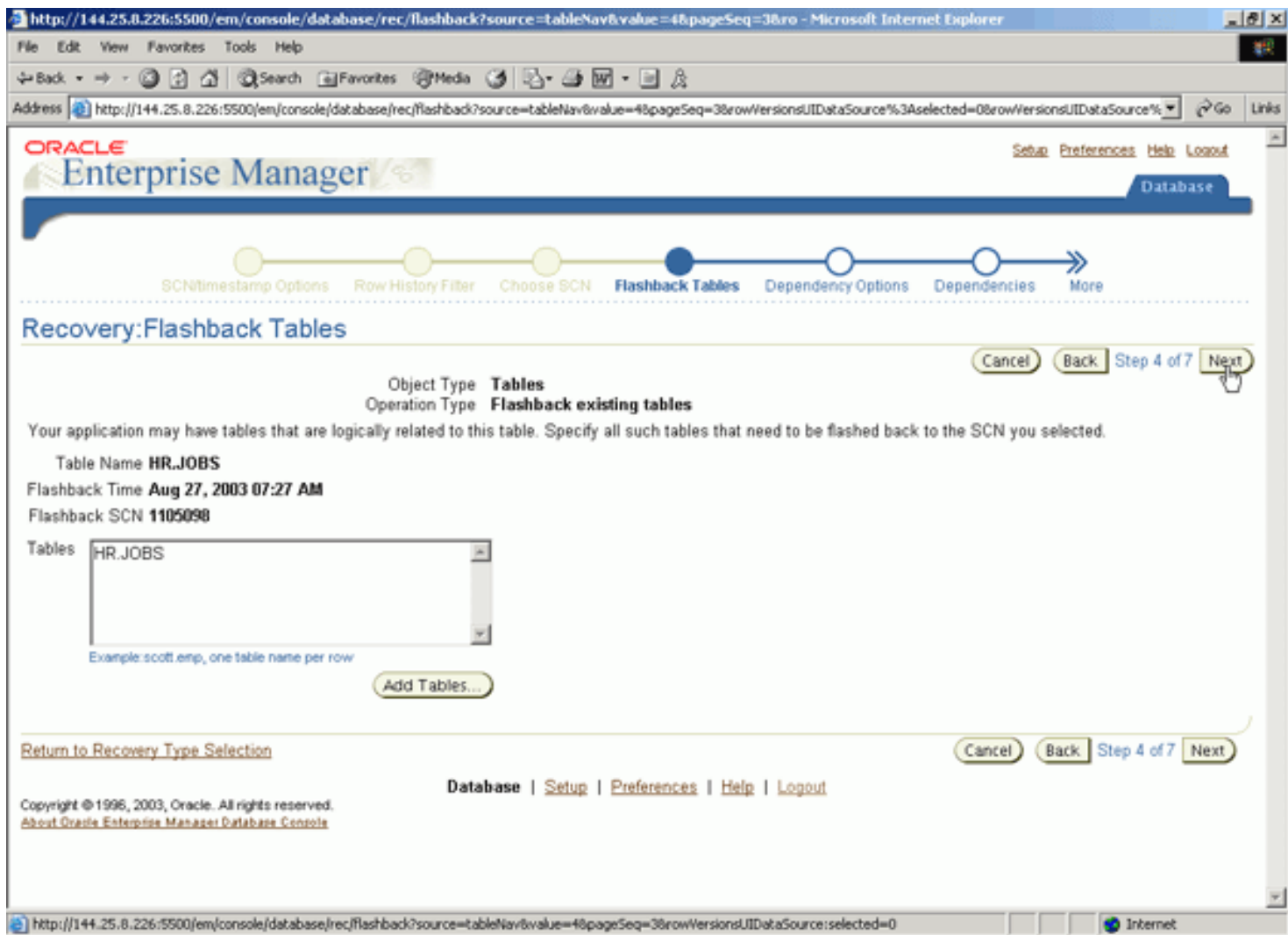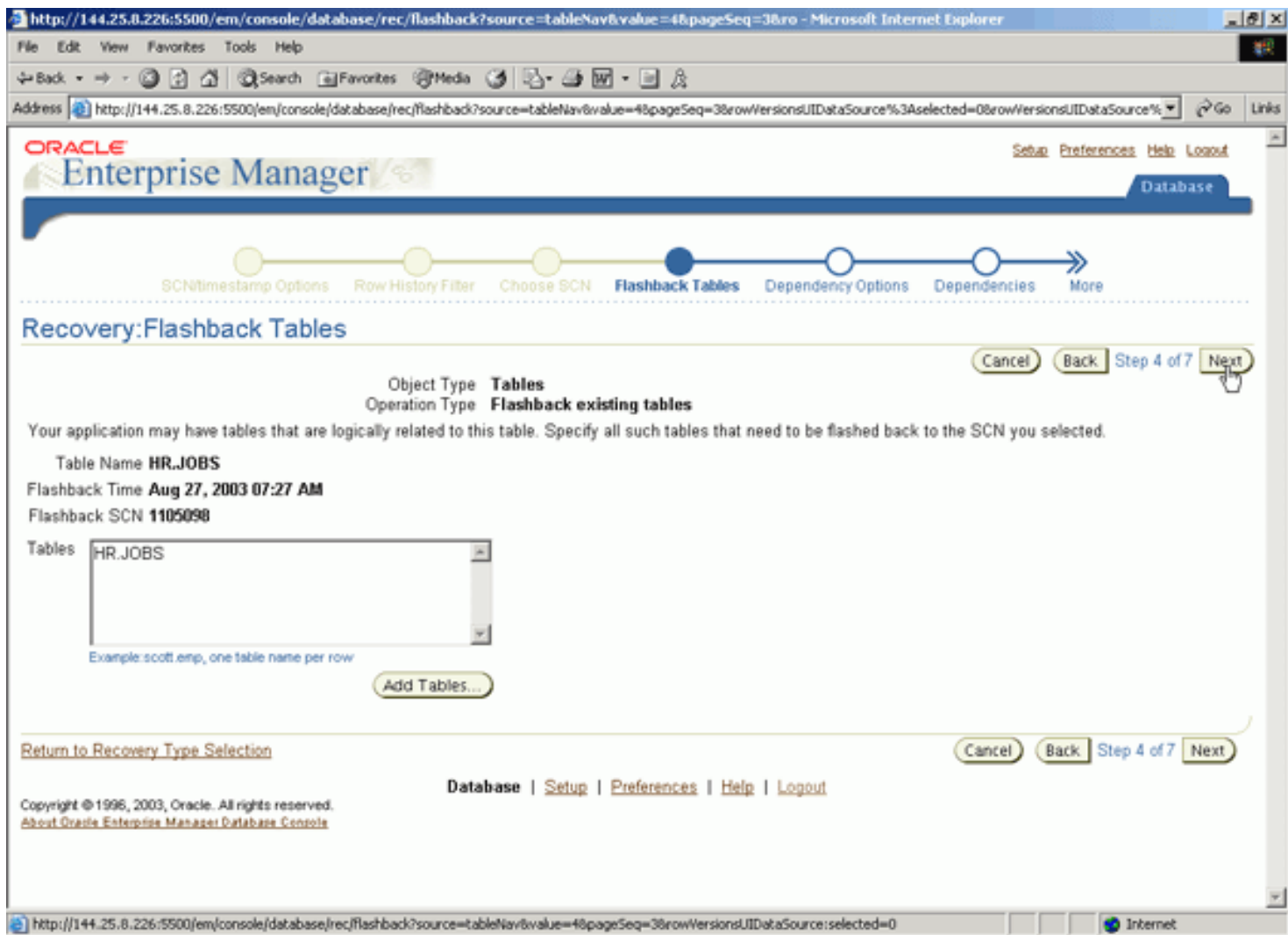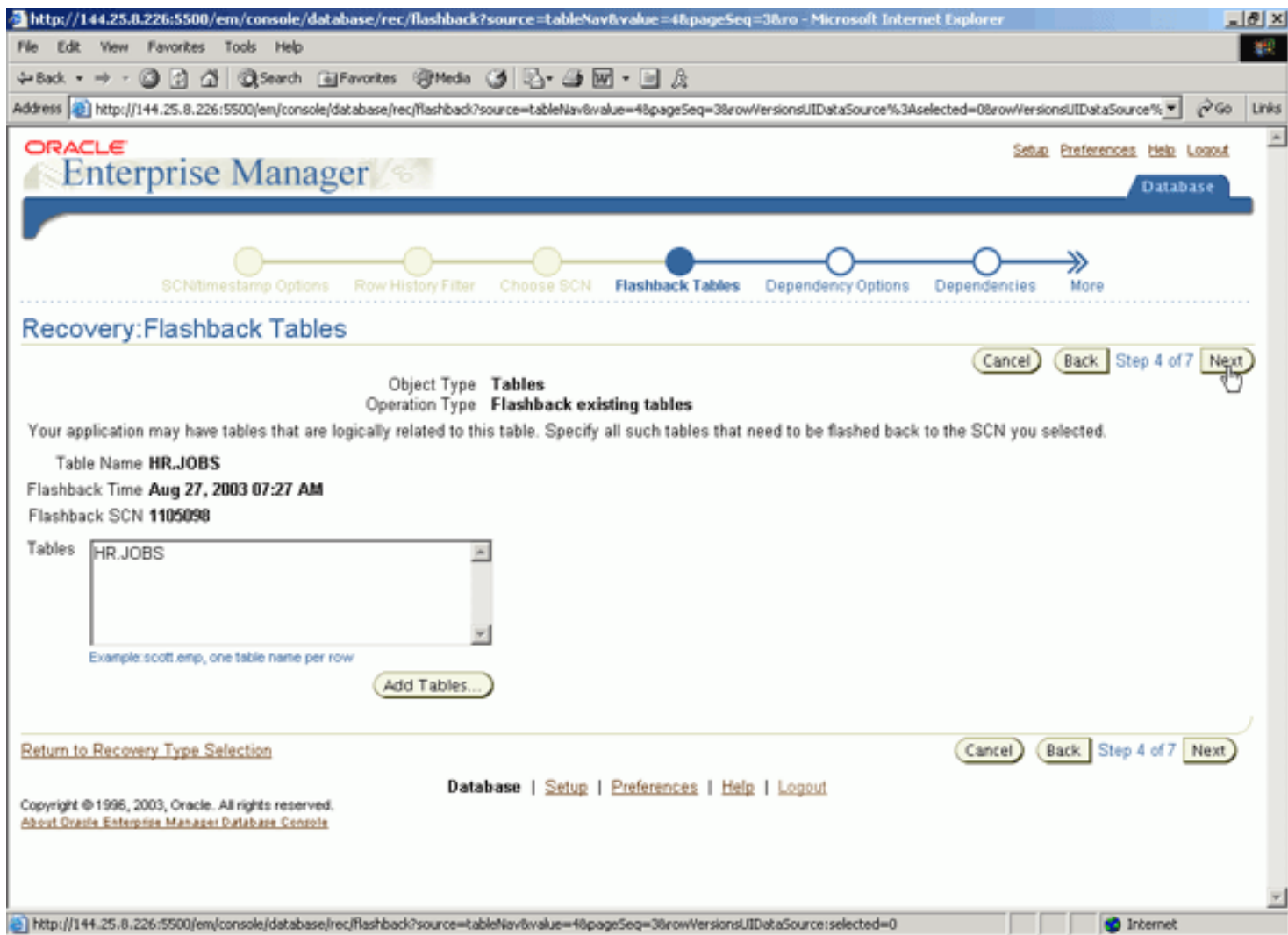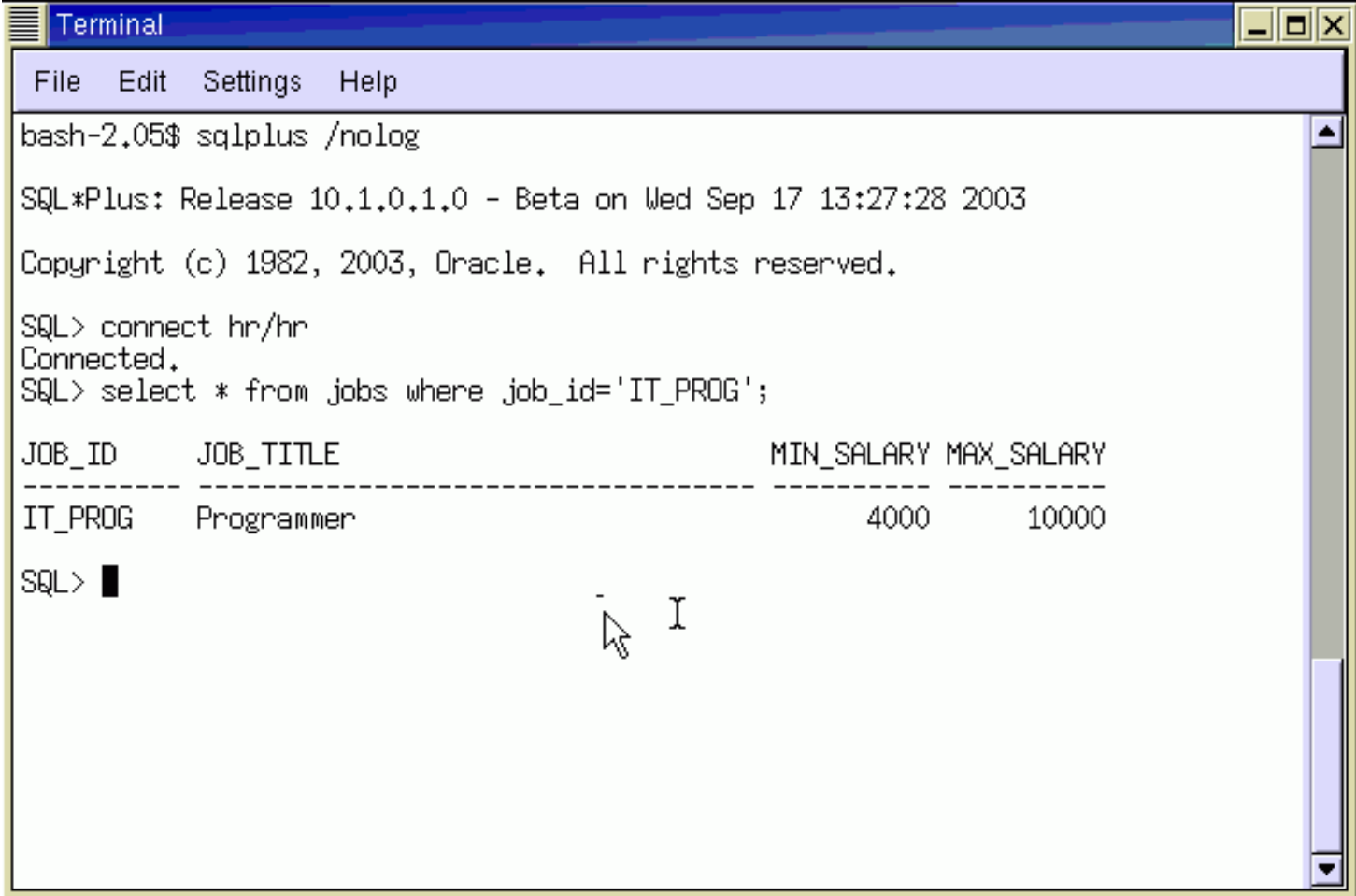The flashback operation has completed successfully. Click **OK** .

11.

12. Switch to your SQL*Plus session again and perform the following SQL Statement to check to see that the salary value has changed back.

```
select * from jobs where job_id='IT_PROG';
```

```
Terminal                                                    _ □ ×

 File   Edit   Settings   Help

bash-2.05$ sqlplus /nolog

SQL*Plus: Release 10.1.0.1.0 - Beta on Wed Sep 17 13:27:28 2003

Copyright (c) 1982, 2003, Oracle.  All rights reserved.

SQL> connect hr/hr
Connected.
SQL> select * from jobs where job_id='IT_PROG';

JOB_ID     JOB_TITLE                              MIN_SALARY MAX_SALARY
---------- ------------------------------------- ---------- ----------
IT_PROG    Programmer                                  4000      10000

SQL>
```

You see that the value has been changed back to 10000.