14. Management of database storage structures.
/repeat file - redo log file, tablespaces /

1.1. Oracle database storage structures.

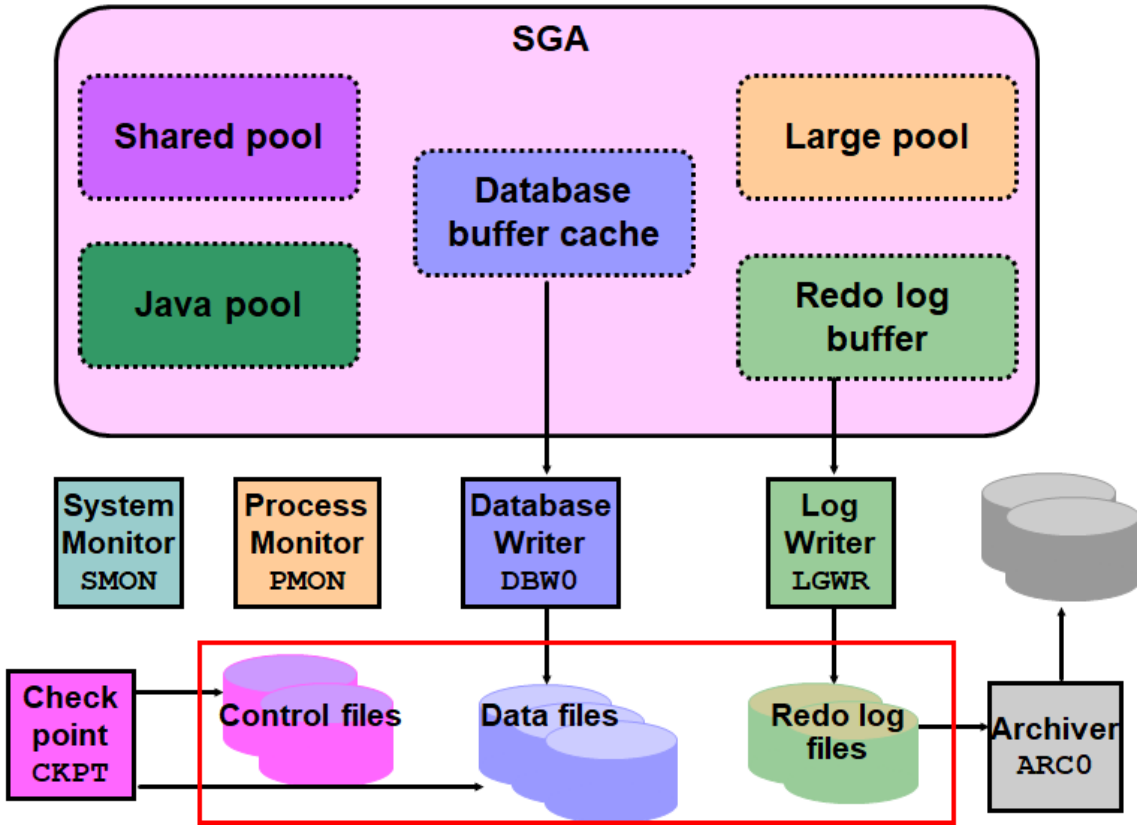


Figure 14.1. Physical structures of databases.

The Oracle database includes the following physical database structures (figure 14.1.):

• Control files: contains records that define the physical structure of the database.

• Data Files: contains all the data from the database (the contents of logical database structures, such as tables and indexes, are physically stored in the data files).

• Redo log files: save all changes made to the data.

Use of the Oracle database storage structure.

Logical data structures are stored in physical database files. You can easily see the logical and physical structures of the user database using Oracle Enterprise Manager.

For detailed information on each structure, the storage links on the server page are selected (fig. 14.2.).

Fig. 14.2. Server page storage area.

1.2. Control files.

When the instance is started and the database is mounted, the control file is read. The entries in the control file indicate the physical files that make up the database.

When additional files are added to the user database, the control file is updated automatically.

The location of the control files is specified in the CONTROL_FILES initialization parameter.

To protect the database from damage due to loss of the control file, it must be multiplexed. When specifying multiple files through the initialization parameter, the user allows the Oracle Database server to maintain multiple copies of the control file.

Access to information about the control files in the user database can be obtained by selecting the Control Files link in the Storage area of the Server page in the Enterprise Manager server (fig. 14.3.). The shared management files page shows the name and location of the control files in the user database. The Advanced page

provides information about creating a control file and identifying the database. The "Record Section" page displays information about the records in the control file.

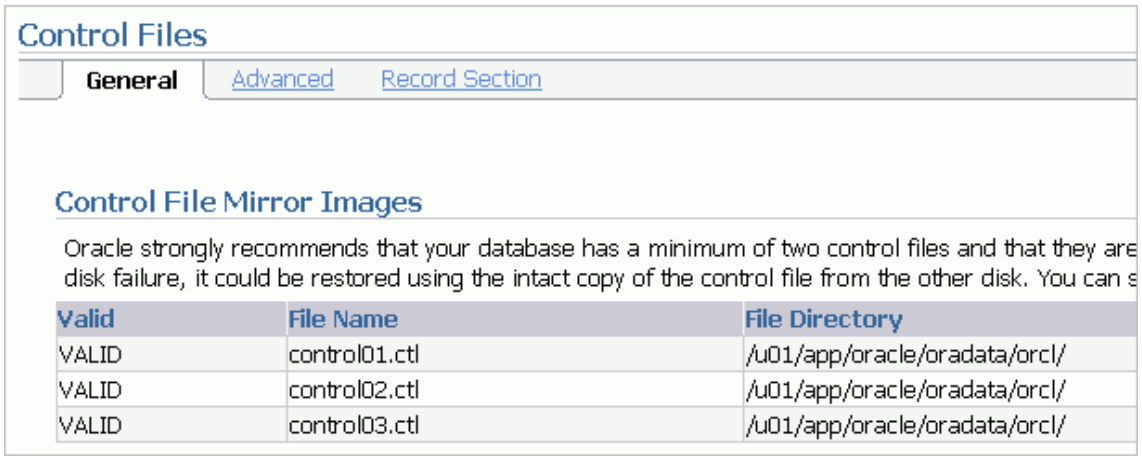


Fig. 14.3. Information about control files in a user database.

1.3. Redo Log Files.

Files for repeat action are used to record changes to the database as a result of transactions and internal actions on the Oracle Database server.

A transaction is a logical unit of work that includes one or more SQL statements managed by a single user. Repeat files protect the database from loss of integrity due to system crashes caused by power outages, disk failure, etc. Reproduction files must be multiplexed to ensure that the information stored in them is not lost in the event of disk corruption.

Redo files consist of groups of overwritten files. The group consists of a repetition log and its multiplexed copies. Each identical copy is a member of this group and each group is identified by a number. The logging process (LGWR) records duplicate entries from the log retry buffer in the replay group until the files in the group are filled or until a log switching operation is required (fig. 14.4). It then switches and saves the files to the next group. Redo groups of logs are used in a circular fashion.

Access to information about the files for re-execution in the user database is done by selecting the Redo Log Groups link in the storage area of the server page (fig. 14.4.). Detailed information can be accessed (including replay file names) by selecting a group and selecting View.

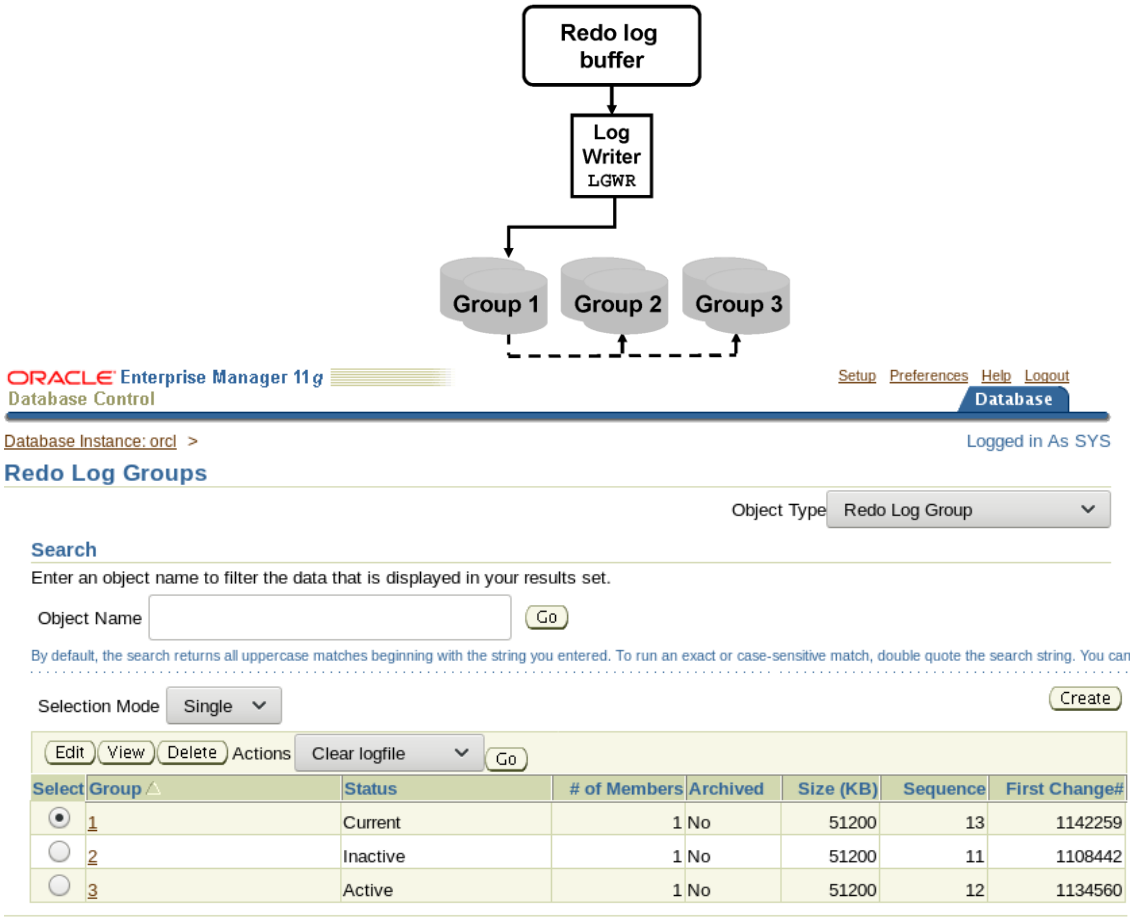


Fig. 14.4. Redo Log Group.

1.4. Redo log multiplexing.

Redo log can be multiplexed by adding a member to an existing group of logs as follows:

1. Select Redo Log Groups in the storage area of the Server page.

The Redo Log Groups page appears.

2. Select a group and Edit.

The Edit Redo Log Group page appears.

3. In the Redo Log Members area, select Add (fig. 14.5.).

The page for adding a new log appears.

4. Enter the file name and file directory. Then select OK.

It is recommended that you store members on separate discs to protect against the loss of rewritable records in the event of a disc failure.

5. Repeat these steps for each existing group.

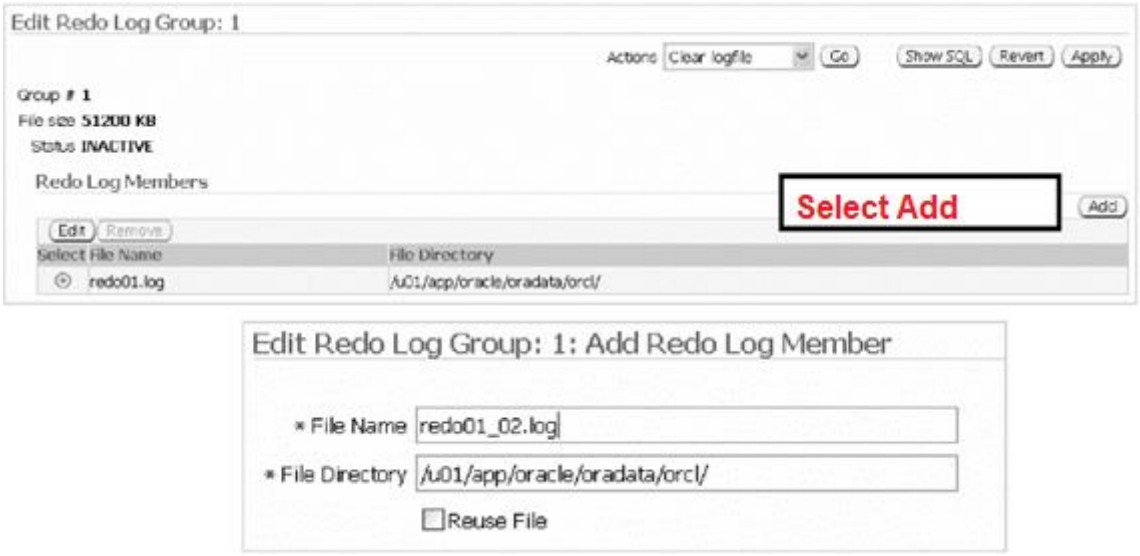


Fig. 14.5. Redo log multiplexing.

1.5. Tablespaces and data files.

The database is divided into logical storage units called tablespaces, which can be used to group connected logical structures together. Each database is logically divided into two or more tablespaces. One or more data files are specifically created for each tablespace to physically store the data of all logical structures in the tablespace.

Database objects, such as tables and indexes, are stored in a tablespace as segments. Each segment contains one or more extensions. The extension consists of adjacent data blocks. Data blocks are the smallest unit of input-output data in the database.

Access to tablespace information in the user database can be obtained by selecting the Tablespaces link in the storage area of the server page (fig. 14.6.). Select the tablespace you are looking for, and then click View to display more detailed information about that tablespace.

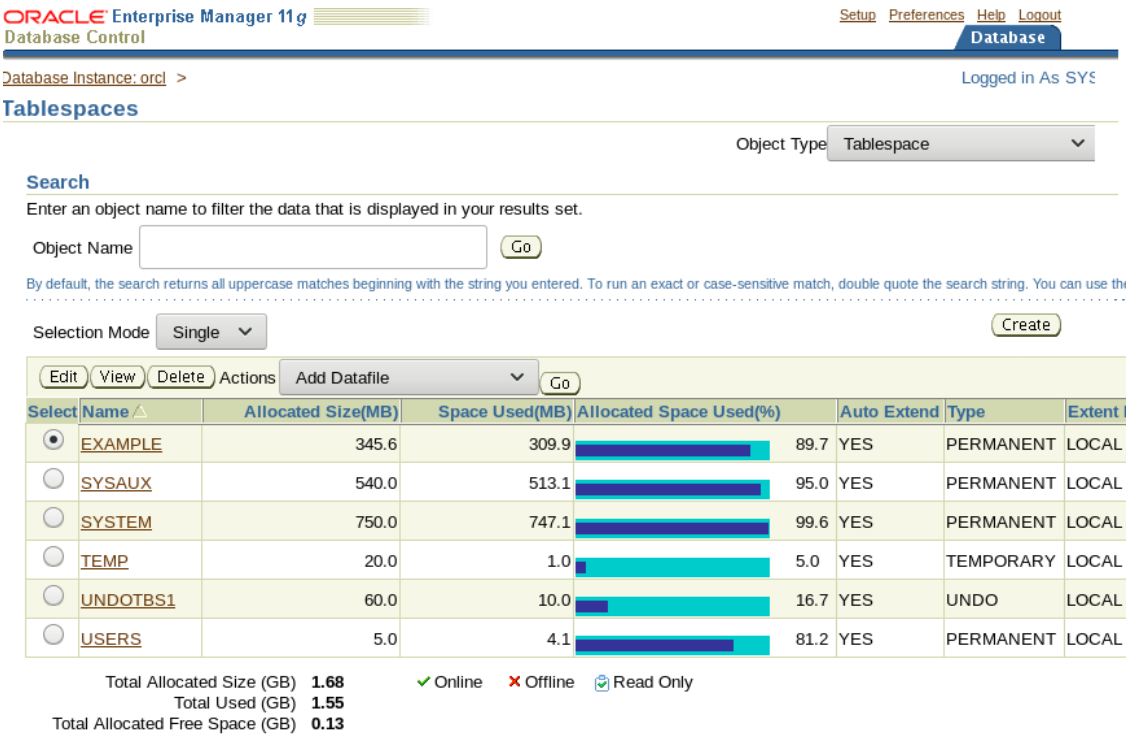


Fig. 14.6. Information about tablespaces.

1.6. Tablespaces as a pre-configured database.

The following tablespaces are created in the pre-configured user database:

• SYSTEM: This tablespace is used by the Oracle Database server to manage the database. It contains tables with the dictionary, which contains administrative information about the database. All of them are contained in the SYS user help schema, which can only be accessed by the SYS user or other administrative users with the required privilege.

• SYSAUX: This is an auxiliary tablespace to the SYSTEM tablespace. Some components and products that use the SYSTEM tablespace (or their own tablespaces) in earlier versions of Oracle now use the SYSAUX tablespace.

• TEMP: This tablespace contains transient data that exists only for the duration of the session. As an example, it can be used for sorting. Each database must have a temporary tablespace that is assigned to users as their temporary space. In the pre-configured database, the TEMP tablespace is specified as the temporary tablespace by default. This means that if no temporary tablespace is specified when creating the user account, Oracle Database assigns this tablespace to the user.

• UNDOTBS1: This is the undo tablespace (Undo) used by the database server to store undo information. Each database must have a tablespace to undo actions, it is created during the creation of the database.

• USERS: This tablespace is used to store persistent user objects and data. In the preconfigured database, USERS is the default tablespace for all objects that are created by non-system users.

• EXAMPLE: This tablespace contains sample schemas that can be installed when creating the database.

Enlarge the database.

The user can increase the size of the database by adding a new tablespace or increase the size of existing tablespaces.

1.7. Creating a new tablespace.

To create a permanent tablespace, follow these steps:

1. Open the Tablespaces page by selecting the Tablespaces link on the Server page.

2. Select Create.

The Create Tables menu appears.

If you need to create a tablespace that is like an existing tablespace, select the existing space and select Create Like from the Actions drop-down list. Then select "Go" (fig. 14.7.).

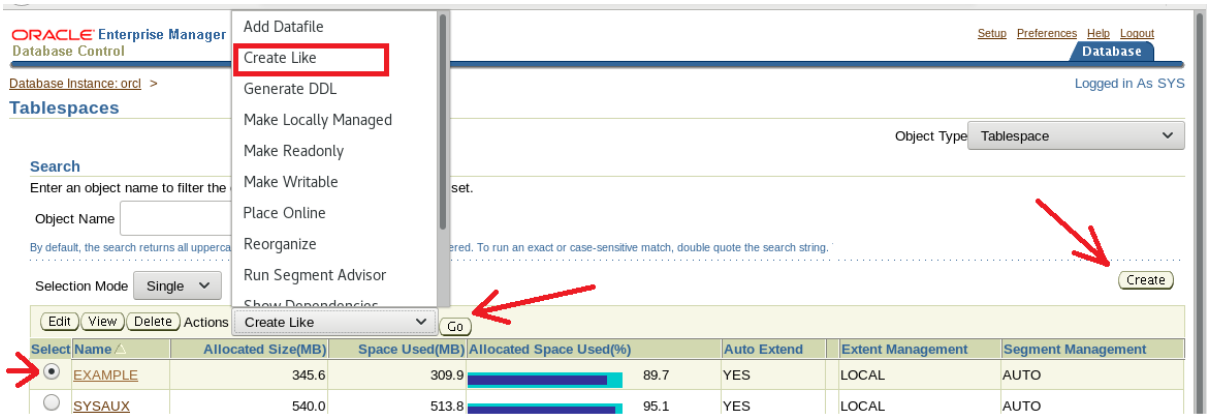


Fig. 14.7. Creating a new tablespace.

3. Enter a tablespace name.

4. Select “Locally Managed” under the heading Extent Management.

Locally managed tablespace extensions are managed efficiently by the Oracle Database server. Vocabulary-managed tabkspac.es use older space management techniques and are not as efficient as locally managed tablespaces. Oracle recommends creating locally managed tablespaces.

5. Select Permanent under the Type heading.

Permanent tablespaces store persistent database objects created by the system or users.

6. Select Read Write under the Status heading.

Read / write status means that users can read and write in the tablespace after it is created. This is the default.

7. Select Add in the Datafiles area of the page (fig. 14.8.).

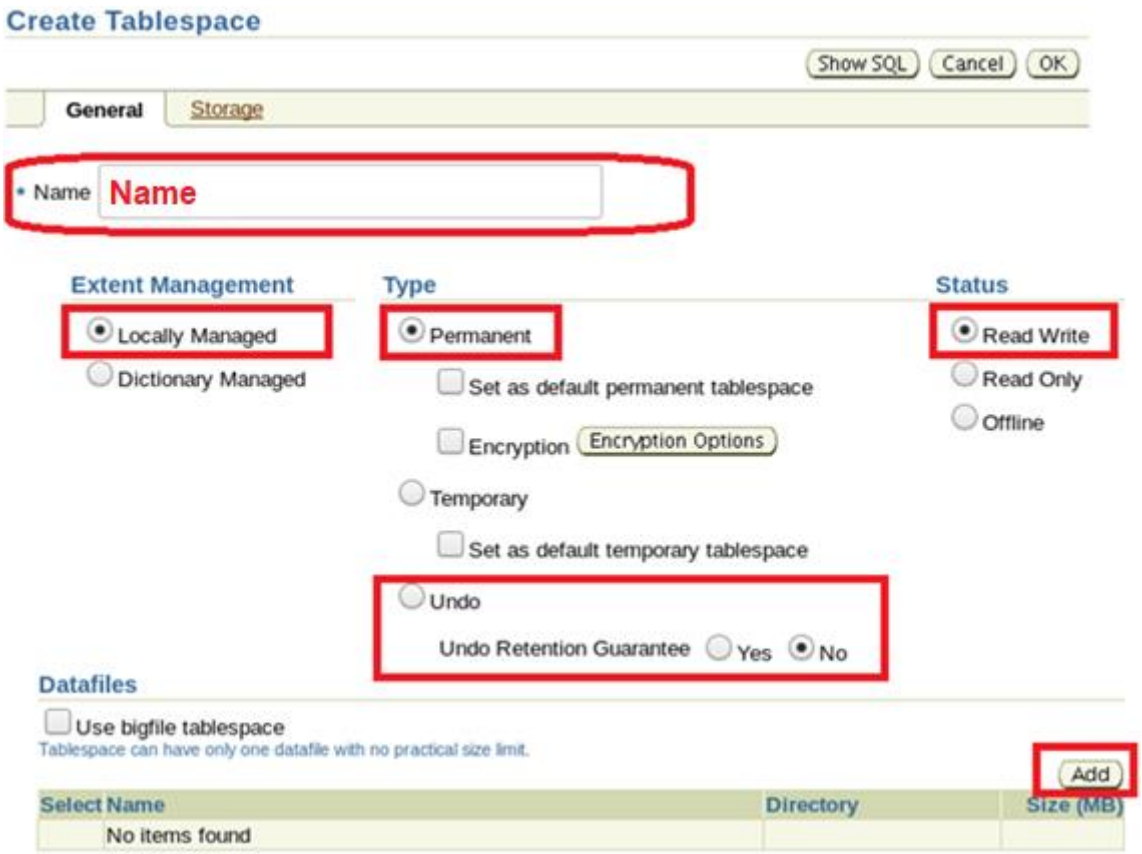


Fig. 14.8. Options when creating a new tablespace.

8. Enter a file name in the "Name" field of the "Add Datafile" page. The default settings for the File Directory and File Size fields are accepted (Fig. 14.9.).

9. In the storage area, select automatically expand data lock when full (AUTOEXTEND) and specify the amount in the "Magnification" field to expand the data file each time it is filled. You can leave the maximum file size set when Unlimited is selected, or you can enter a limit. Then select Continue to go back to the Create Tablespace page.

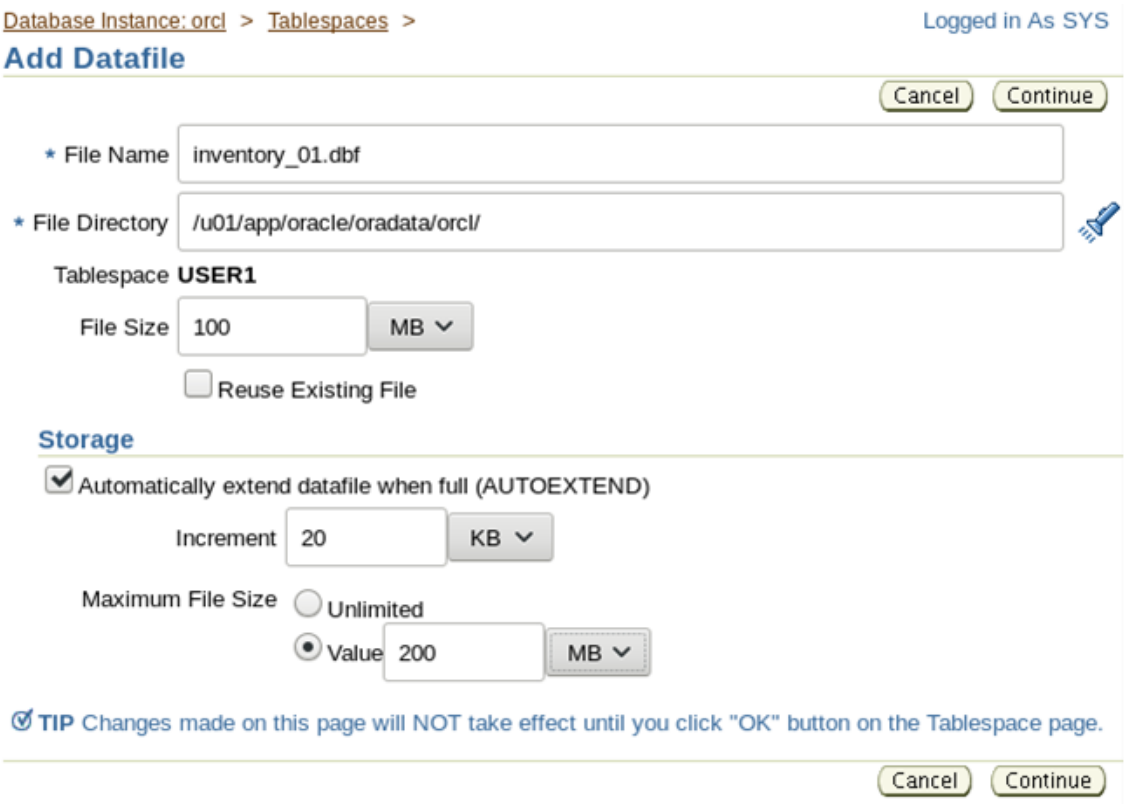


Fig. 14.9. Add a data file.

10. Select the Storage tab.
The page “Create tablespace” appears (fig. 14.10.).

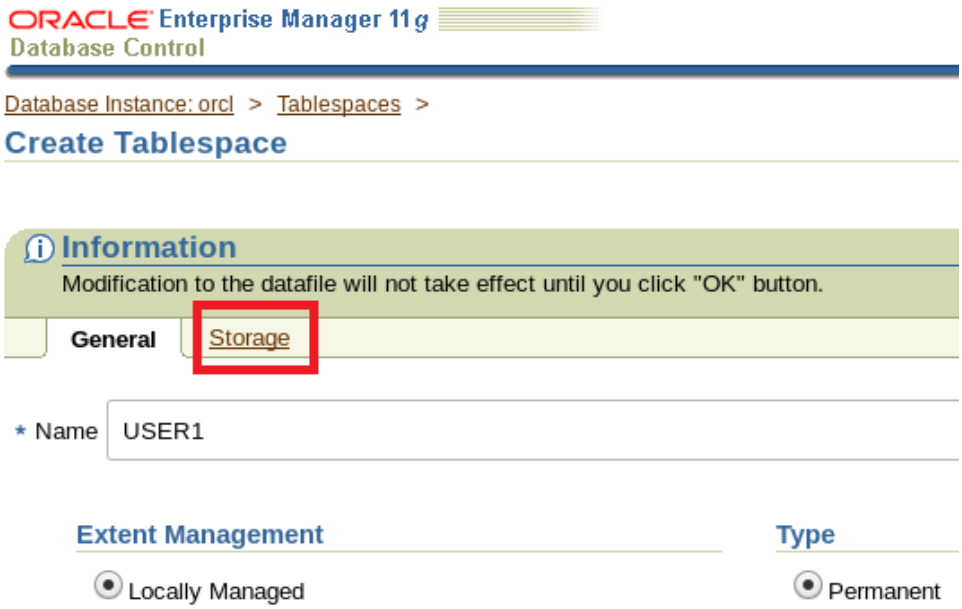


Fig. 14.10. Storage.

11. All default settings on the storage page are accepted.

12. Select OK to add the tablespace.

Then return to the Tablespaces page, where you receive confirmation that the tablespace has been created. You can now see the new tablespace in the tablespace window.

1.8. Change tablespace.

The size of the tablespace can be increased by manually expanding or expanding the data file that belongs to it. The tablespace can be configured so that the data file is automatically expanded when space is needed. If you need to perform an operation to restore a specific tablespace, you may need to transfer it offline.

Add additional space to an existing tablespace by adding a data file (fig. 14.11.):

1. Select the Tablespaces link in the Storage area of the server page.

2. Select a tablespace and use the Add Datafile and Go action options. The Add Data page appears.

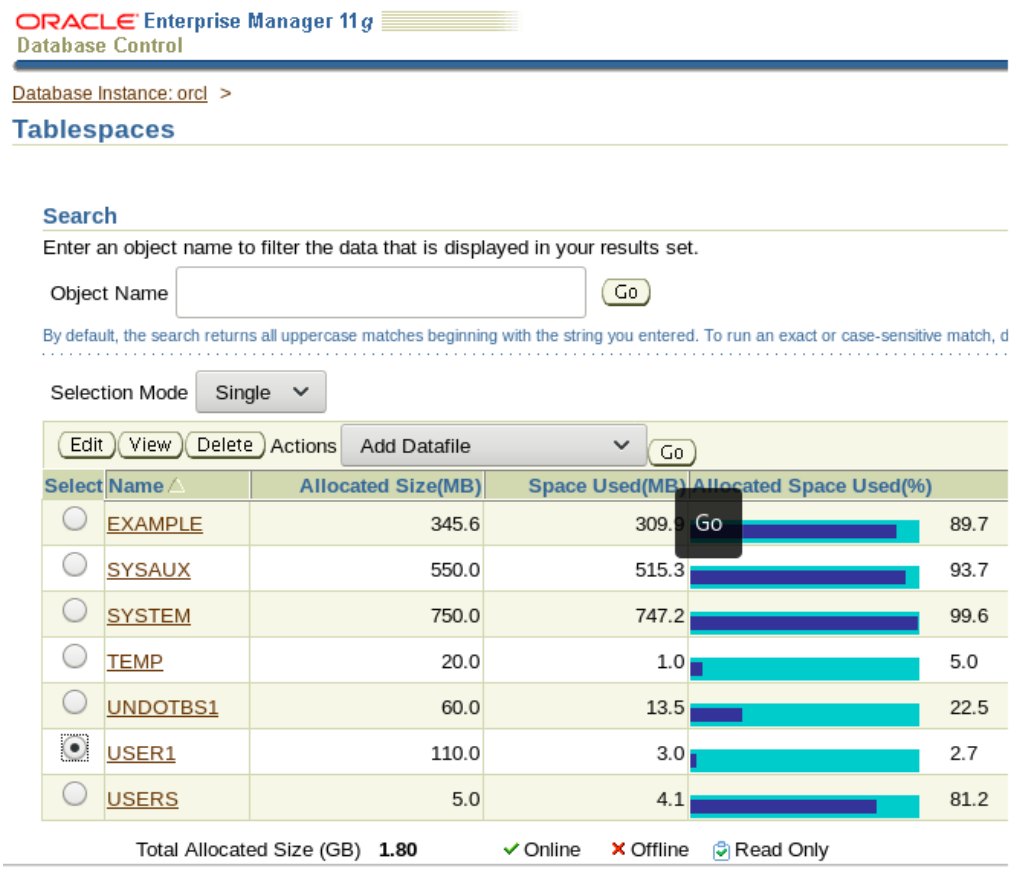


Fig. 14.11. Add a data file to an existing tablespace.

3. Enter the file name, directory and file size. Additional storage information may be provided if desired. Confirm with OK (fig. 14.12.).

4. A confirmation message appears on the Tablespaces page.

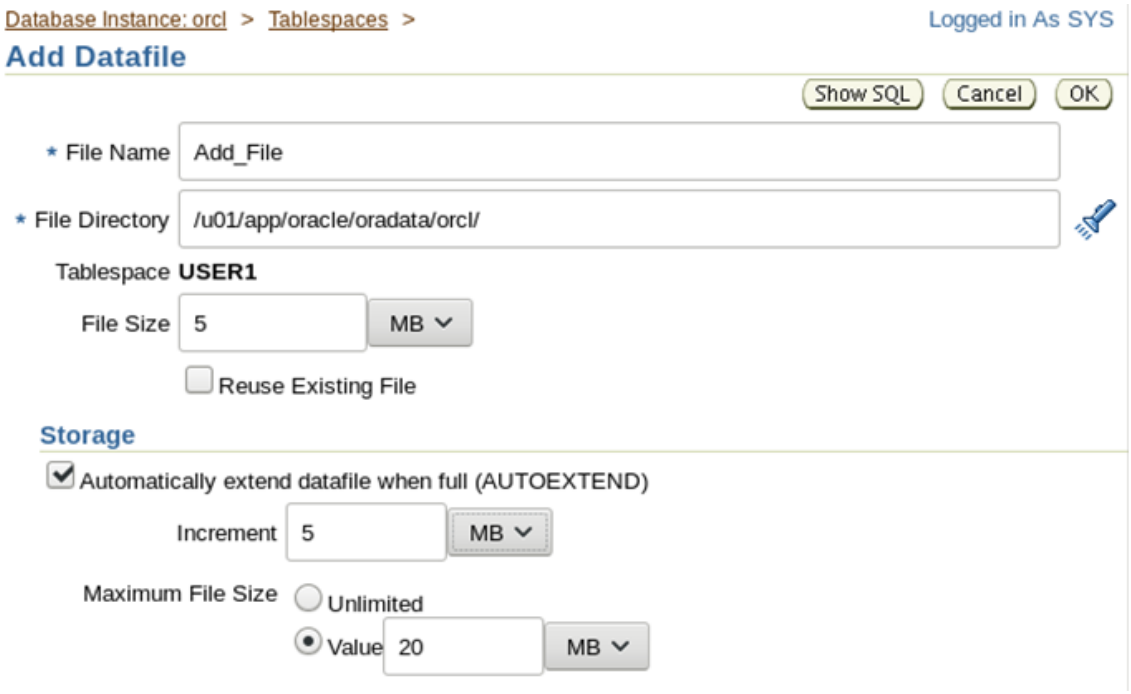


Fig. 14.12. Settings when adding extra space.

Automatic expansion of tablespace when its limit is reached:

1. Select the Tablespaces link again in the storage area of the server page.

2. Select the tablespace to be configured to expand automatically. Then select Edit.

The "Edit Tablespace: NAME" property page appears (fig. 14.13.).

3. Select the data file associated with the tablespace, and then select Edit. The Edit Data File page appears.

Fig. 14.13. Automatic expansion of tablespace.

4. Select to automatically expand the data file when it is full “Autimatically exted datafile when full (AUTOEXTEND).” Enter the appropriate growth step. You can optionally set a maximum file size or set an unlimited file size (fig. 14.14).

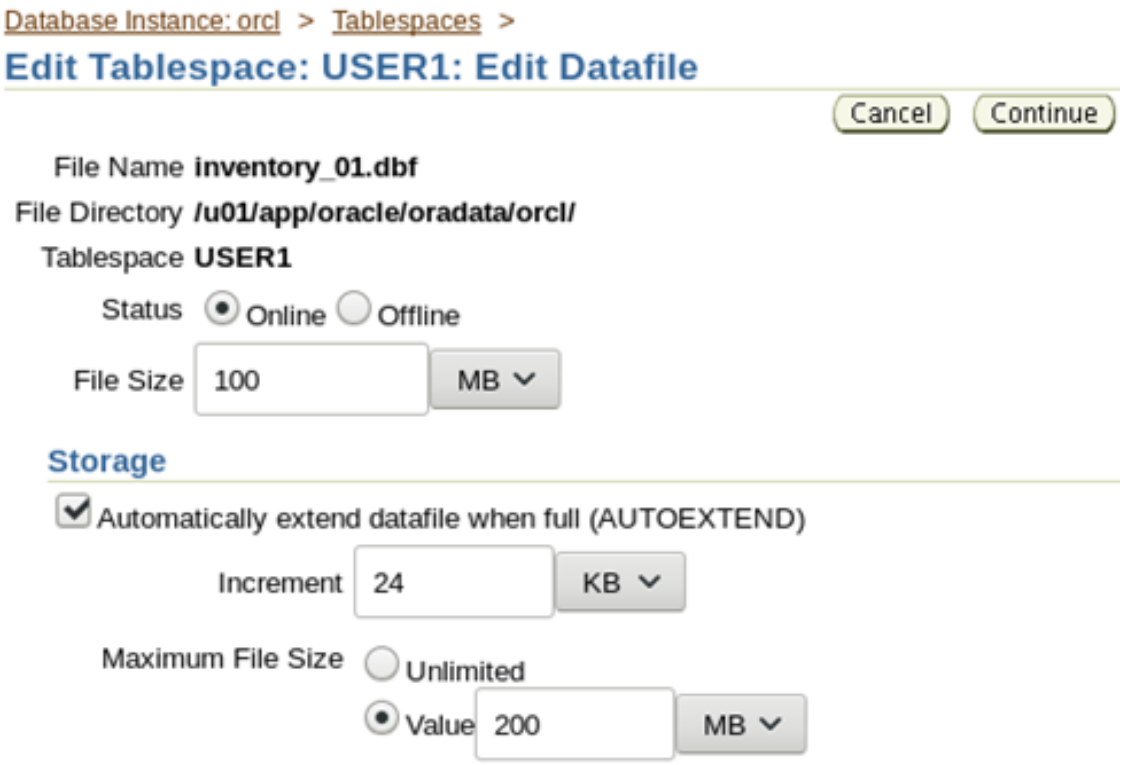


Fig. 14.14. Edit data files.

5. Select Continue, return to the tablespace edit page.

6. Apply is selected. A confirmation message appears.

An SQL statement can be used to manually resize a data file. This allows the user to add space to a database without adding data files, which is advantageous in order not to reach the maximum number of data files allowed in the database. Manually resizing data files allows user to recover unused database space. This is useful when errors in space requirements estimates need to be corrected.

Tablespace thresholds can be changed by selecting a tablespace on the Tablespaces page and going to the Threshold property page. On this page, the thresholds for space generating warnings and critical signals can be changed (fig. 14.15.).

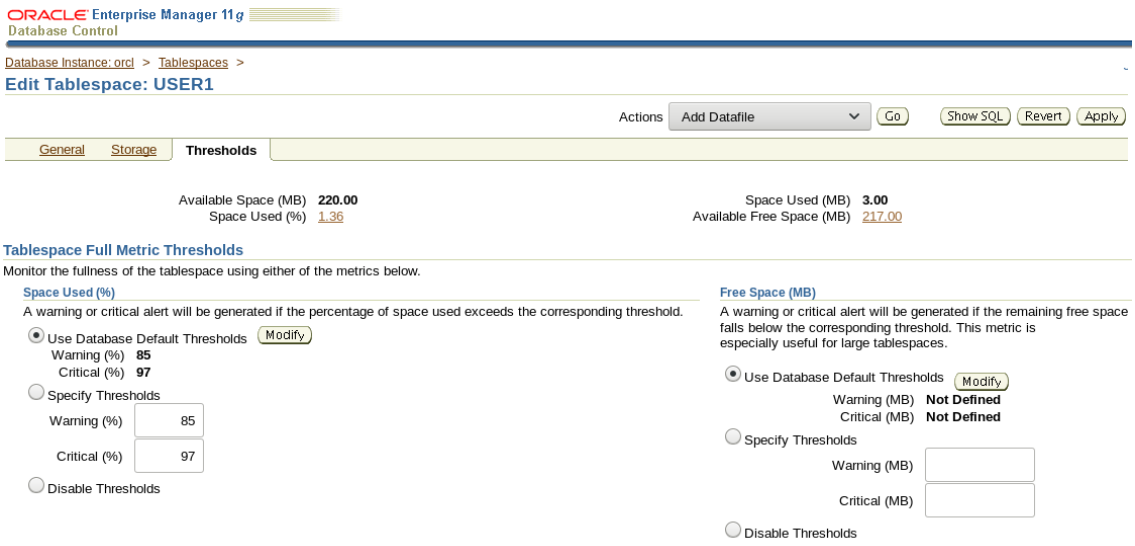


Fig. 14.15. Tablespace thresholds.

Tablespace can be taken offline on the Edit Tablespace page. Select "Offline" in the Status section and then apply (fig. 14.16.).

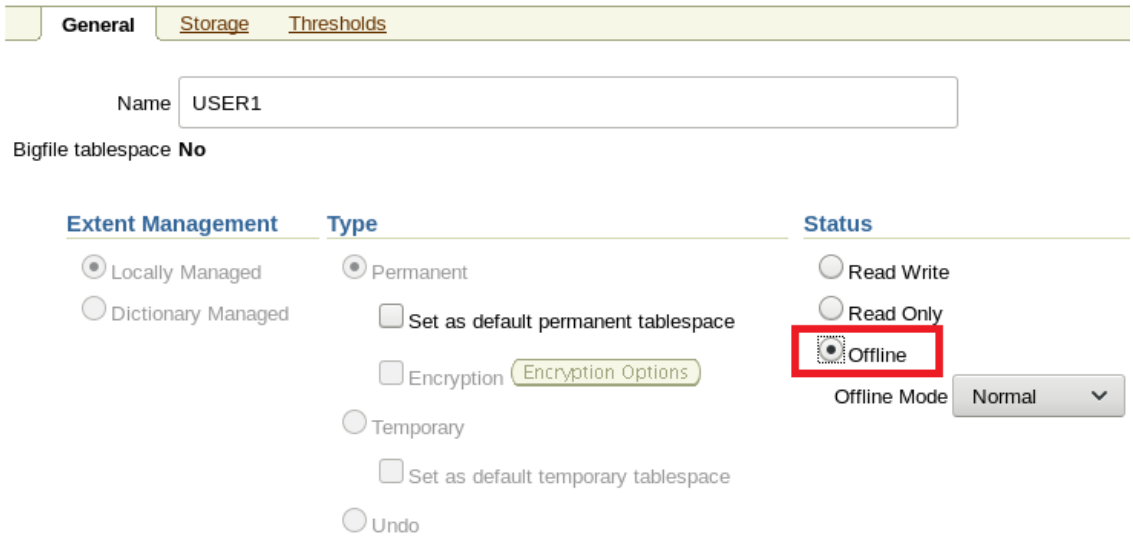


Fig. 14.16. Tablespace offline.

1.9. Delete tablespace.

Once a tablespace is deleted, the objects stored in the tablespace and the data in them are no longer available. In addition, object definitions are removed from the data dictionary. Tablespace-related data files are also deleted from the operating system.

To delete a tablespace using Enterprise Manager:

1. Select Tablespaces in the Storage area of the Server page.

The page for tables appears.

2. Select a tablespace to delete and then select delete.

Enterprise Manager prompts the users to confirm whether they want to delete the tablespace and related data files from the operating system.

1.10. Restore a space in a tablespace.

Over the time, Data Manipulation Language (DML) operations for different segments in a tablespace can create pockets of unused space. These individual pieces are often not large enough to be reused. When all this unused space is added together, there may already be a large amount of total free space in a tablespace.

The Segment Wizard identifies database objects that have unused recoverable space.

It performs its analysis by examining usage and growth statistics and by sampling data at the site. It runs automatically at regular intervals by default. The regular scheduled implementation of the Segment Advisor is known as the Automatic Segment Advisor. The results of the implementation (known as automatic segment wizard results) are summarized in the "Space Summary" section of the database home page as recommendations of the segment wizard.

Each running automatic segmentation wizard analyzes only a subset of segments in the database. If it is considered that certain segments may have recovered unused space, but that these segments do not appear among the recommendations of the segment wizard, the automatic segment wizard may not have selected them for analysis.

This unused space can be restored by manually executing the segment wizard and performing a shrink operation. In this operation, the data is compressed to the front of the table. Free space can be returned to a tablespace or saved in the table for future entries. The table shrink operation does not affect DML table operations.

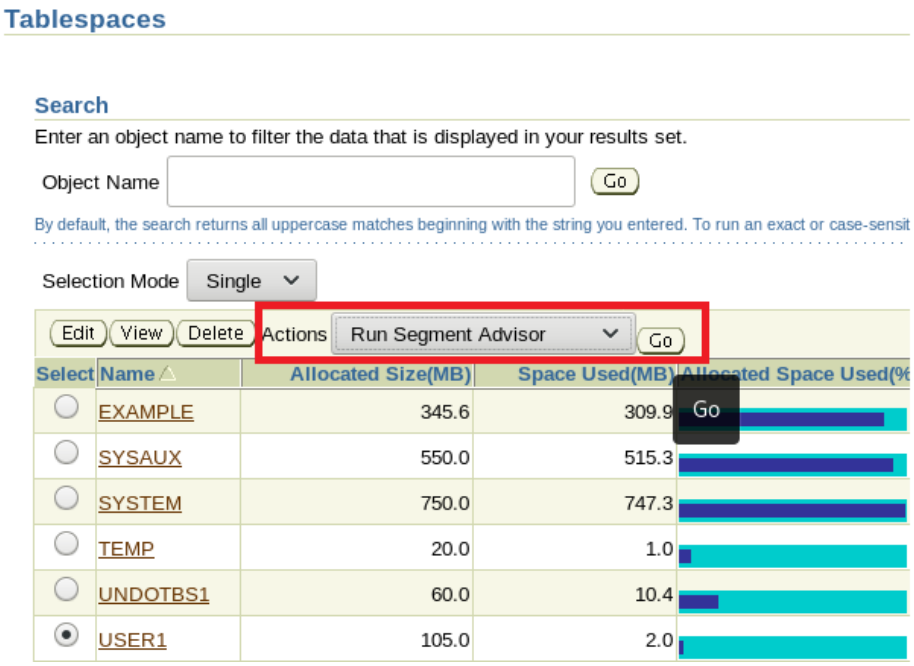


Fig. 14.17. Launch an Automatic Segment Advisor.

Use the segment advisor to analyze a specific tablespace:

1. Select Tablespaces in the Storige area.

2. Select a tablespace and select Run Segment Advisor from the Actions drop-down menu. The space used before the operation is remembered for later comparison. Then select "Go" (fig. 14.17.).

The Automatic Segment Advisor information page appears.

The Segment Advisor can also be launched from the Advisor Central page.

3. Select the link for recommendations (fig. 14.18.).

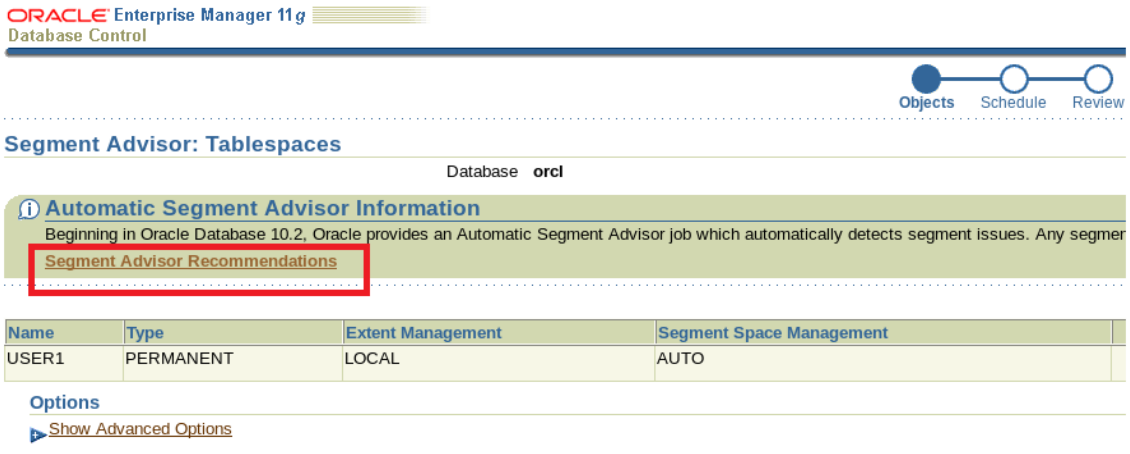


Fig. 14.18. Segment Advisor Recommendations.

The Segment Advisor Recommendations page appears, listing the table space segments that qualify for shrinkage.

4. A shrinkage segment is selected using one of the following techniques:

• Select the Shrink button next to the segment to be shrunk.

• One or more segments are selected and then - execution.

The segment shrink page with options appears.

5. Compact segments are accepted and space is freed up. This action returns the free space in the table space. If you are not looking to assign free space to table space, select "Compact Segments". Select "Implement".

The segment shrink schedule page appears.

6. The name of the task is remembered. Under "Start" it is chosen to start the task immediately.

It should be borne in mind that the shrinkage operation can be resource intensive; Oracle recommends that this operation be performed during non-business hours. You can also select "Later" and plan a better time to perform the operation.

7. Select Submit.

The task scheduling page appears and the status of the task is displayed.

8. Select Refresh to refresh the page until the task status is complete. An overview of the task status can also be selected.

1.11. Undo Data - essence.

Undo is the collective term for records that the Oracle database server writes before changing the data. Undo is used to undo all unauthorized changes made to the database in case an undo operation is required. The undo operation can be:

• the result of the user explicitly issuing a cancellation request to cancel the transaction changes;

• part of a recovery operation in case of cancellation of the session;

• part of a recovery operation, for an instance or work environment.

Undo is also used to ensure reading consistency. Each user receives a consistent and complete view of the data, even while other changes may occur for that data.

The undo information is the basis for several Flashback features of the Oracle database that allow data to be viewed or restored up to a previous point in time. These features include Flashback Query, Flashback Transaction and Flashback Table.

Undo records are stored in the undo table space. Because the space in the tablespace is reused cyclically, undo records are subject to overwriting. For the

success of Flashback functionality (and for complete reading integrity in long transactions), records should not be overwritten too soon.

To control the retention of undo records, an undo hold period can be set, which in turn affects the size of the undo table. Automatic revocation hold setting is enabled by default during installation.

Oracle Database automatically determines how long the undo data should be retained after the transaction based on queries executed on the database. During this time, it is said that the data for cancellation are in a state of indefiniteness. Oracle Database will keep undo data indefinitely for as long as required by queries currently being executed against the database, or for as long as any low undo threshold (which is sometimes longer). After this time, the status of the undo data changes to "expired". Cancellation data is a good candidate for rewriting only when they are in an "expired" state.

If the undo table does not have enough free or expired storage space for active undo data generated by current transactions, Oracle Database may be forced to overwrite the unopened undo data. As this can lead to the failure of user queries, this is not a desirable situation.

1.12. Storage of undo information.

Undo tables exist only in a specialized form of table space called undo tablespace. Other types of segments (such as tables) cannot be created in undo table spaces.

The installation process automatically creates an undo table space. The user can also create a "bigfile" undo table space. Although a database can have many undo table spaces, only one can be defined as the current undo table space at a time.

Undo table spaces are permanent, locally managed table spaces with automatic extension allocation. The Oracle Database server manages them automatically.

Undo segments are automatically created and are always owned by the SYS user. Because undo segments act as a round buffer, each segment has at least two extensions. The maximum number of default extensions depends on the size of the database block.

Because undo data is needed to recover from failed transactions (such as those that can occur when an instance crashes), undo table spaces can only be restored while the instance is in the MOUNT state.

1.13. Comparison between Undo data and Redo data.

Table. 5.1. Undo data and Redo data.

	Undo	Redo
Record of…	How to undo a change	How to reproduce change
Used for ...	Go-back, reading integrity and Flashback	Recovery to redirect forward if any changes to the database
Stored in…	Undo segments	Undo logs
Protects against…	Inconsistent readings in multiuser systems	Data loss
Archives	Not required (Undo can be discarded after the transaction is completed.)	It is crucial for recovery

The undo and Redo data sounds similar, but they serve many different purposes. Undo exists in case there is a need to undo the change and this happens for completeness of read, return and Flashback, as described above. Re-execution is necessary if the changes need to be made again (for example, in case they are lost for some reason) (Table 5.1.).

The process of performing the action involves checking whether the changes in the transaction have been saved in the Redo file, which is permanently stored on disk, unlike memory. In addition, it is usually multiplexed, or has multiple copies of the redo data on disks. Although the changes may not have been saved in the data files where the table blocks are actually stored, confirmation that the changes were saved in the redo file is sufficient. If there is a power outage just before the changes are reflected in the data files, this is not a problem. Because the transaction is completed when the system restarts, it is able to forward all redo records that have not yet been reflected in data files during the interrupt.

Redo data must be archived for recovery purposes from the last backup. Undo does not require such archiving, because once the transaction is complete, these changes no longer need to be undone.

1.14. Use Enterprise Manager to view undo information.

The Enterprise Manager Automatic Undo Management page can be used to view the undo configuration, including the automatically set undo retention period and the name and size of the Undo tablespace.

The Automatic Undo Management page is accessed by selecting "Automatic Undo Management" in the database configuration area of the "Server" page (fig. 14.19.).

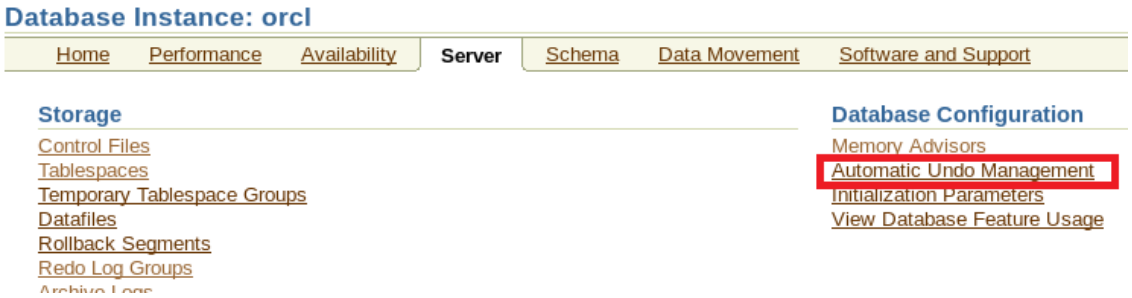


Fig. 14.19. Access the Automatic Undo Management page.

1.15. Undo Management.

Oracle's database server provides automatic undo management, which is a fully automated mechanism for managing undo information and space in a dedicated undo space for all sessions. The system automatically adjusts to ensure the best possible retention of undo information. Specifically, to automatically expand table spaces, the undo hold period is set to be slightly longer than the longest active query. For fixed undo table spaces, the server is dynamically tuned for the best possible retention.

Although the Oracle Database server automatically manages data and undo space by default, some tasks may need to be performed if you use Flashback operations. Revocation administration must prevent errors in space, the use of too much space and errors from "too old" information.

On the page for management of the automatic undo it is chosen to edit the table Undo Tablespace (fig. 14.20.). Then choose to edit in the "Datafile" section to change the fixed size undo table space.

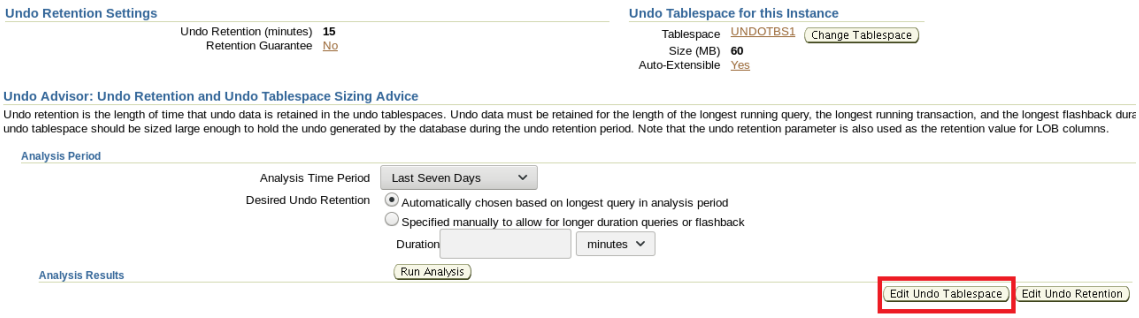


Fig. 14.20. Edit Undo table space.

1.16. Use the undo wizard to calculate the minimum amount of the undo table space.

Undo Advisor can be used by selecting "Undo Advisor" on the undo management page (fig. 14.21.).

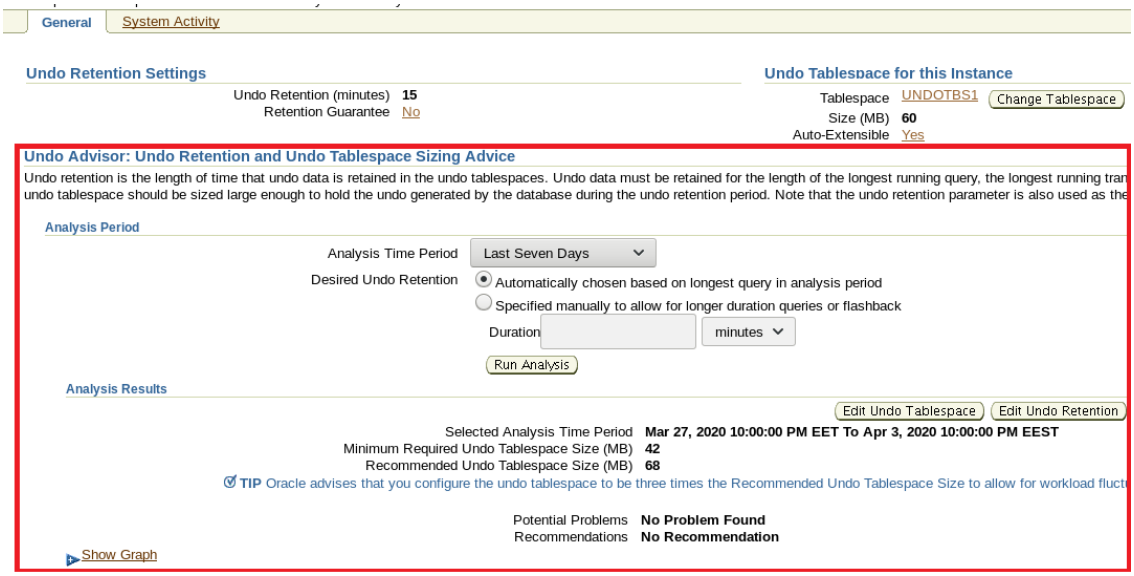


FIG. 14.21. Undo Advisor.

The minimum size of the revocation tablespace can be calculated based on statistics collected for a certain period of time or on the basis of a certain retention undo period.

To calculate the minimum size of the undo table based on statistics collected over a period of time, the following steps are performed:

1. Open the page “Automatic Undo Management”.

2. In section “Analysis Time Period” select the desired time period from the drop-down list “Time Period”.

3. Select „Automatically chosen based on longest query in analysis period in the field “Desired Undo Retention”.

4. Select „Run Analysis “.

The minimum required amount of undo table space is displayed in the section „ Analysis Results“.

5. (Optional) View a graphical result of the analysis by selecting "Show Graph".

To calculate the minimum size of the undo table based on a specified duration, the following steps are performed:

1. Open the Automatic Undo Management page.

2. In the "Analysis Period" section, "Specified manually to allow for longer duration queries or flashback" is selected.

3. Enter the desired duration of the cancellation delay in seconds, minutes, hours or days in the "Duration" field.

To determine this value, the expected duration of the longest request for a specific workload is considered and / or the longest expected duration for the Oracle Flashback operation is analyzed.

4. Select Run Analysis.

The minimum required amount of undo table space is displayed in the "Analysis Results" section.

5. (Optional) To display a graphical result of the analysis, select Show Graph.

View system activity information.

"System Activity" is selected to view detailed information during the user analysis period. The upper part of the page provides information about the activity during the selected period (Fig. 14.22.).

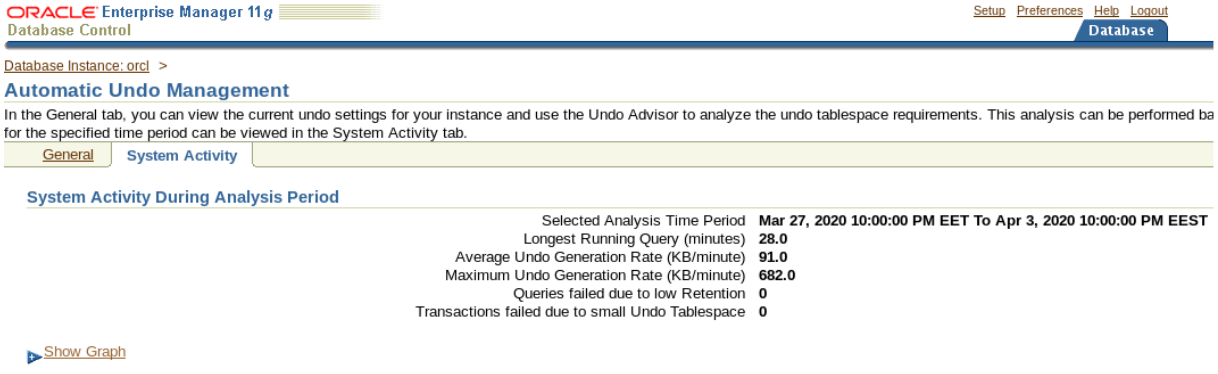


Fig. 14.22. View system activity information.

"Show Graph" is selected to display the information in graphical format.

The undo tablespace usage graph shows the amount of space available for reuse, the amount of space reconfigured for query or retransmission, and the amount of space used for active transactions. The vertical axis shows the size of the undo tablespace in megabytes; the horizontal axis shows the date and time (fig. 14.23.).

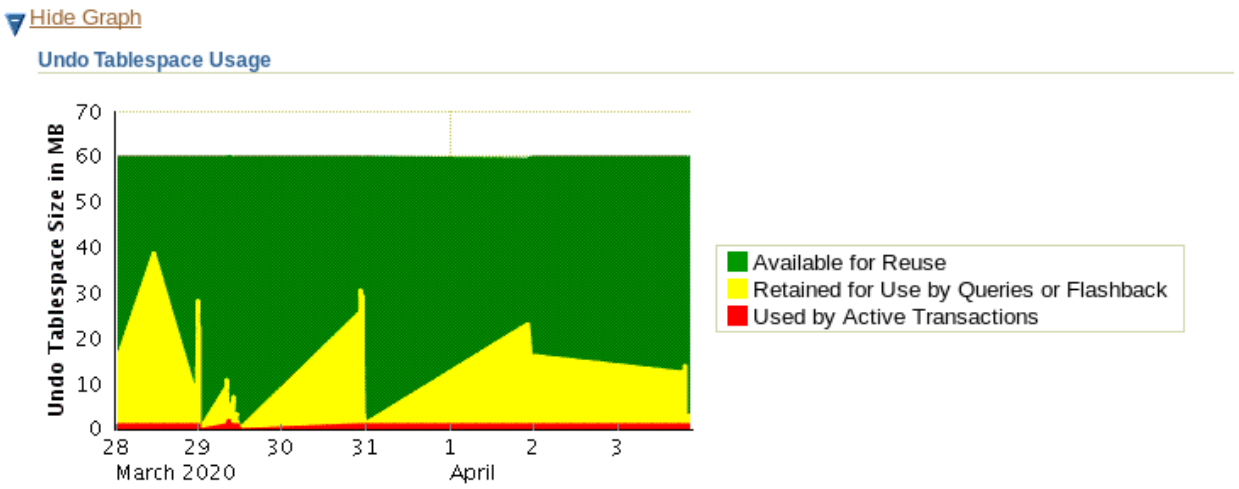


Fig. 14.23. Usage of the undo tablespace.

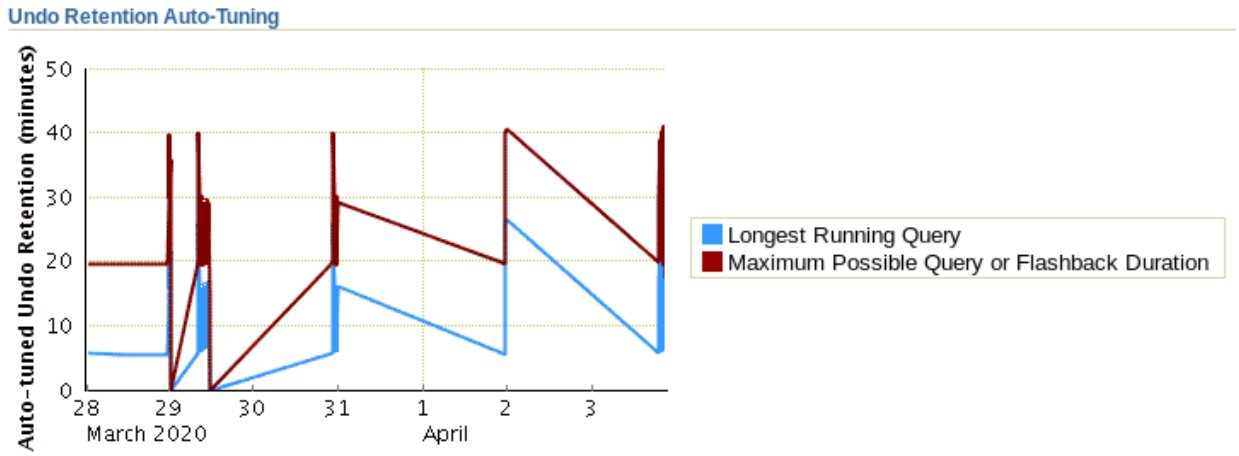


Fig. 14.24. Undo retention auto-tuning graph.

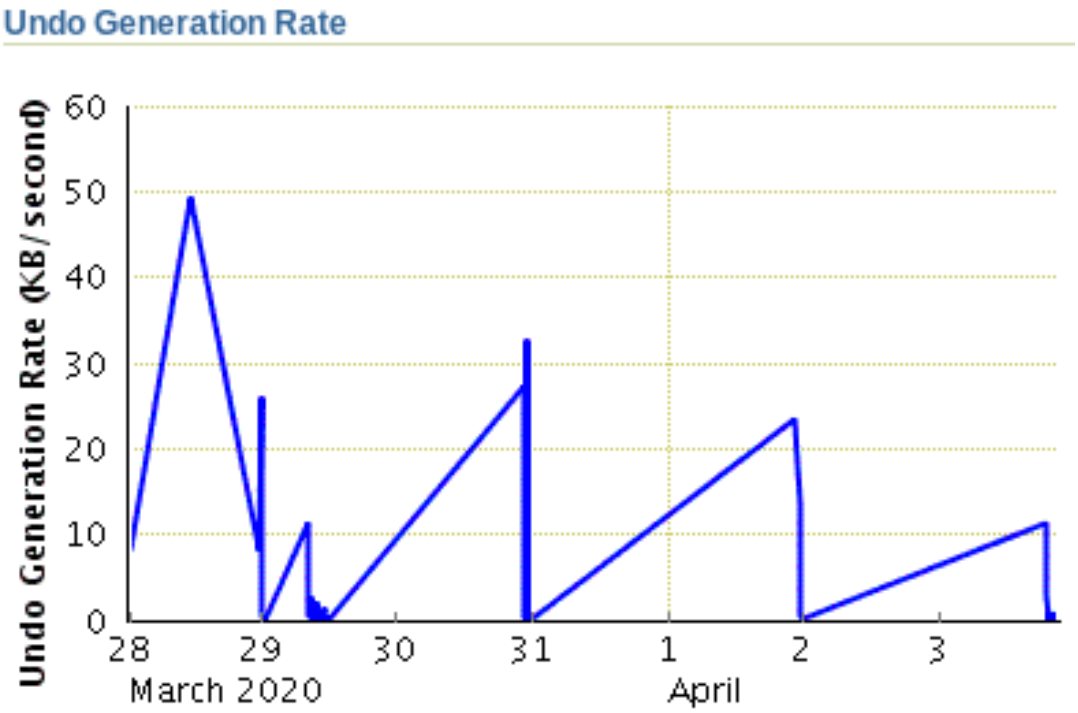


Fig. 14.25. Undo Generation. Frequency of generating undo operations.

The Undo Retention Auto-Tuning graph shows the automatically set retention delay (Fig. 14.24.). Shows the maximum possible duration of the request or return, together with the longest request. The vertical axis shows time and date information; the horizontal axis shows the automatically set undo hold.

The undo generate graph shows the frequency of generating undo operations. The vertical axis shows the rate of cancellation generation in kilobytes per second; the horizontal axis shows the date and time.

7. Managing database schema objects.

The concept of "scheme".

A schema is a collection of database objects. The schema is owned by a database user and has the same name as that user. The objects of the schema are the logical structures that directly relate to the data in the database. Schema objects include structures such as tables, views, and indexes.

There is no connection between table space and schema. Objects in the same schema can be in different tablespaces, and a tablespace can contain objects from different schemas.

It is possible to create and manipulate schema objects using SQL or Oracle Enterprise Manager. When Enterprise Manager is used, basic SQL is generated for

the user. Schematics created as part of the database creation process include: SYS, SYSTEM, and sample schemas.

7.1.1. Schemes.

When creating databases, a number of schemas are created for the user. The following two schemes are of particular importance:

• SYS schema: All base tables and views that make up the database dictionary in the database are created in the SYS schema. The Data Dictionary is a collection of tables that describe the Oracle database. When the database is created, the data dictionary is located in the SYSTEM tablespace, it is updated by the Oracle database server when running a Data Definition Language Operator (DDL). The data dictionary contains information about users, schema objects, and storage structures. The data dictionary can also be used only as a reference for reading information about the database. When using Enterprise Manager, data dictionary tables can be accessed through views. SYS schema objects should never be modified by a user or database administrator, and no one should create tables in the SYS schema.

• SYSTEM Schema: Contains additional tables and views that store administrative information and internal tables and views used by various Oracle options and tools. No additional objects need to be created in the SYSTEM schema.

• Sample schemas: During a full installation of Oracle Database, sample schemas are installed automatically with the initial database. Sample diagrams serve to provide a common platform for examples in Oracle's documentation and curriculum. They are a set of interconnected schemes aimed at providing a multi-layered approach to complexity and include the following:

- HR (Human Resources): The Human Resources Scheme is a simple scheme for introducing key topics. An extension of this scheme supports Oracle Internet Directory (OID) demos.

- OE (Order Entry): The order entry scheme is for dealing with issues of medium complexity. Many types of data are available in the OE schema. The OC subcircuit (online catalog) is a collection of object-relational database structures built into the OE schema.

- PM (Product Media): The scheme is dedicated to multimedia data types.

- QS (Queued Shipping): The supply queue scheme contains a set of schemes used to demonstrate the capabilities of Oracle Advanced Queuing.

- SH (Sales History): The sales history scheme is designed to allow demonstrations with larger amounts of data. An extension of this scheme provides support for advanced analytical processing.

7.2. Access to schema objects.

The user can quickly access many types of schema objects from the Database objects menu on the Schema page (Fig. 14.26.). After selecting one of the links, the page for this type of object is displayed. In the search area of the page, you can enter a schema name and an object name to search for a specific object. User can also search for other types of objects from the search section by selecting the object type from the drop-down menu.

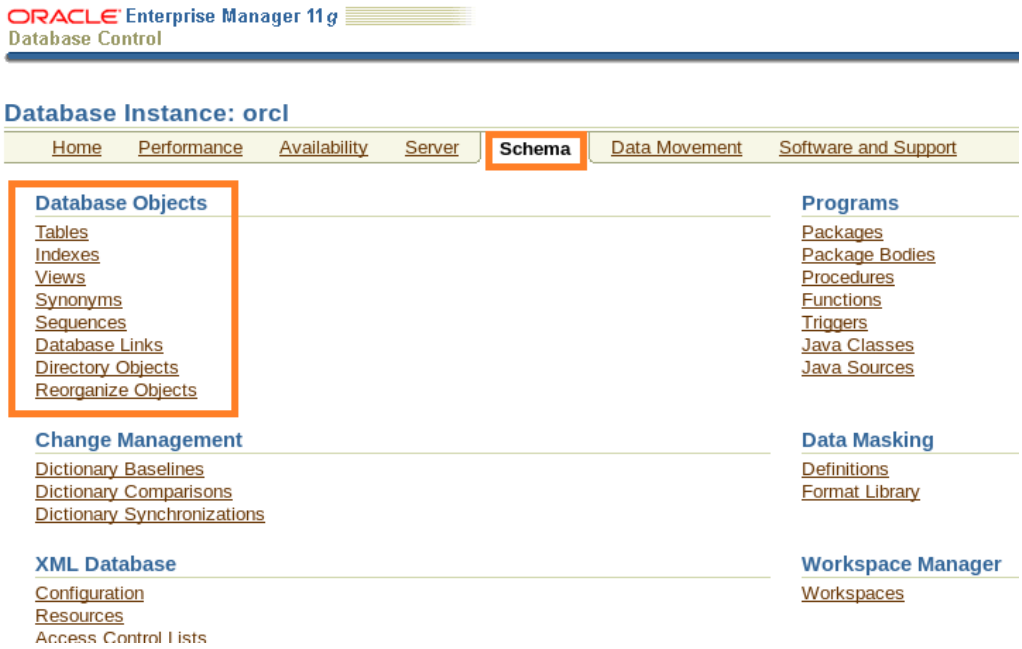


Fig. 14.26. Database objects.

7.2.1. Set data types in tables.

When creating a table, the data type for each of its columns must be specified. When creating a procedure or stored function, the data type for each of its arguments must be specified.

These data types determine the domain of the values that each column can contain.

The built-in data types of the Oracle database are:

• CHAR (n) - Character data with a fixed length of n bytes. The maximum size is 2000 bytes. The default minimum size is 1 byte.

• DATE - Valid period from January 1, 4712 (BC) to December 31, 9999 (AD). The DATE type stores date and time information.

• NUMBER (p, s) - Number with accuracy p and scale s. Accuracy is the total number of digits. The scale is the number of digits to the right of the decimal point.

• VARCHAR2 (n) - Variable length string with maximum length n in bytes or characters. The maximum size is 4000 bytes and the minimum is 1 byte or 1 character. For the VARCHAR2 type, the size in brackets must be indicated.

7.3. Creating and modifying tables.

Tables are the basic unit for storing data in an Oracle database. They store all data available to the user. Each table has columns and rows. Create a table using Enterprise Manager:

1. Select Tables in the Database Objects area of the Schema page. The Tables page appears (Fig. 14.27.).
2. Select Create. The Create Table: Organization of Tables page appears (Fig. 14.28.).
3. Selected by default - "Standard (Heap Organized)", followed by Continue.
4. The Create table page appears (Fig. 14.29.).
5. Enter the name of the table in the Name field.
6. Enter the schema name in the Schema field or click on the flashlight icon to call up the search function.
7. Enter the name of the table space in the Tablespace field or click on the flashlight icon to call up the search function.
8. In the Columns region, enter the column name and data types.
9. Confirmation follows - OK.
10. The update message indicates that the table was created successfully (Fig. 14.30.).

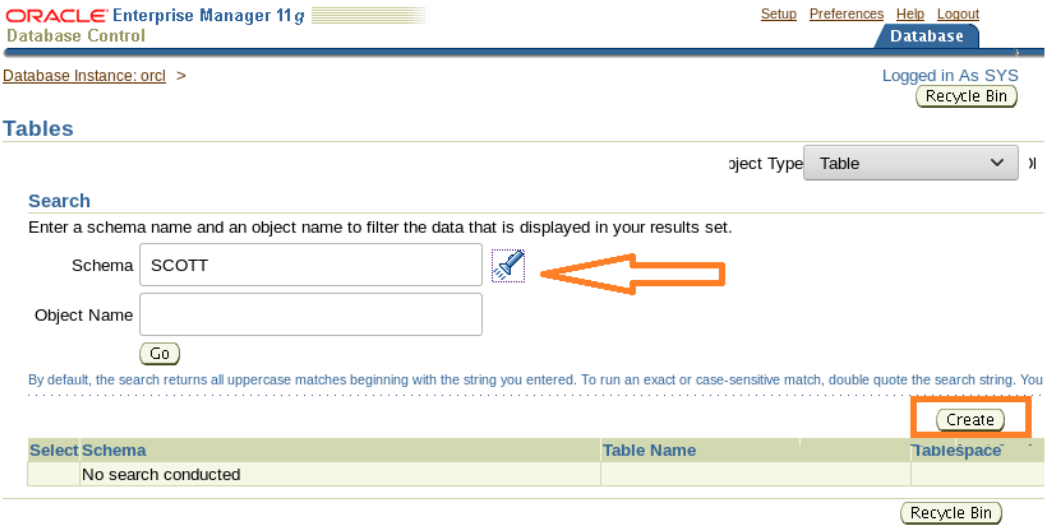


Fig. 14.27. The page Tables.

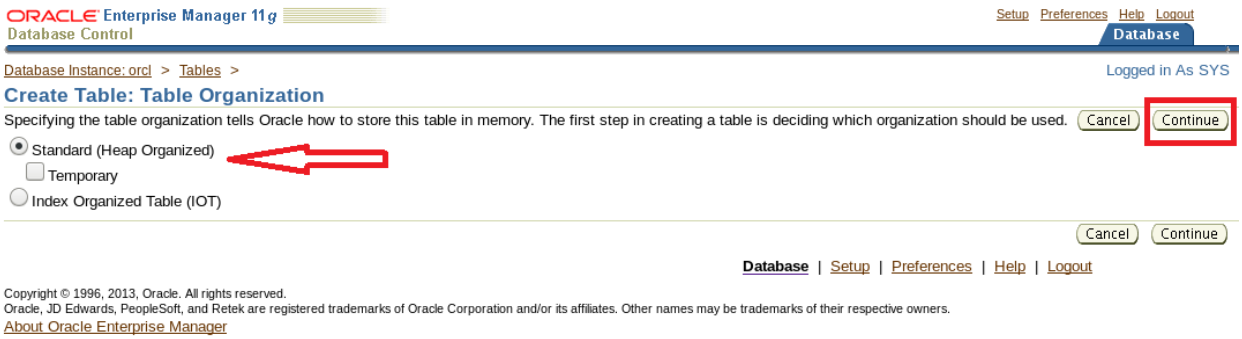


Fig. 14.28. Page Create a table: Table Organization.

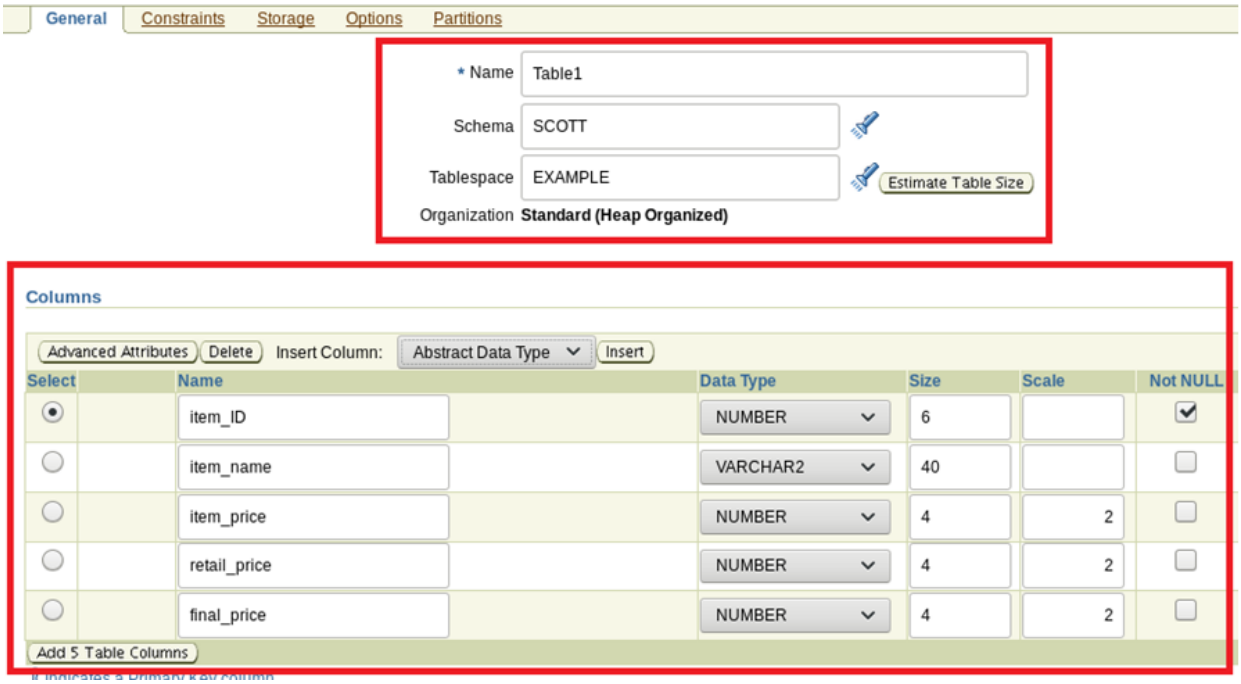


Fig. 14.29. Creating a table.

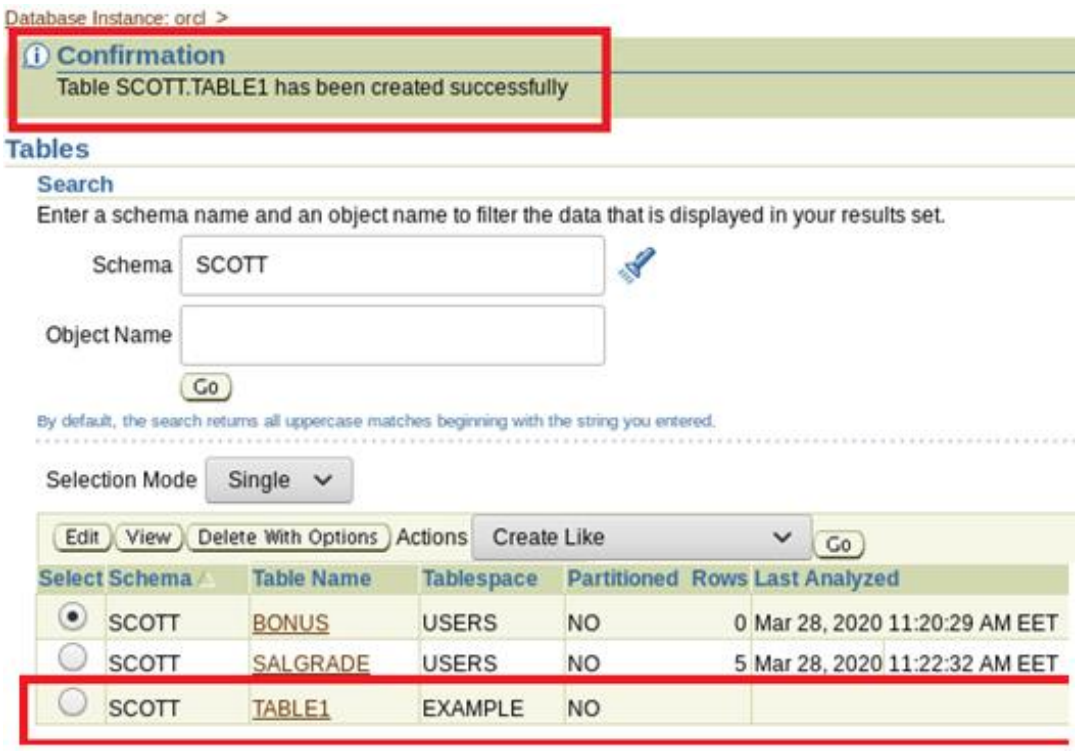


Fig. 14.30. Successfully creating a table.

7.3.1. Change table.

In the following example, Enterprise Manager is used to change a table - additional columns are added (Fig. 14.31.).

1. On the Tables page, select the table in the list of results (after selecting a scheme and GO) and Edit - Edit. In addition to all attributes and metadata, as created, columns can be added and deleted.

2. If there are no empty fields in the Columns section, select the "Add 5 columns per table" button. A list of editable columns appears.

3. Enter the name of the new column, data type and size. Select Apply. An update message indicates that the table has changed successfully.

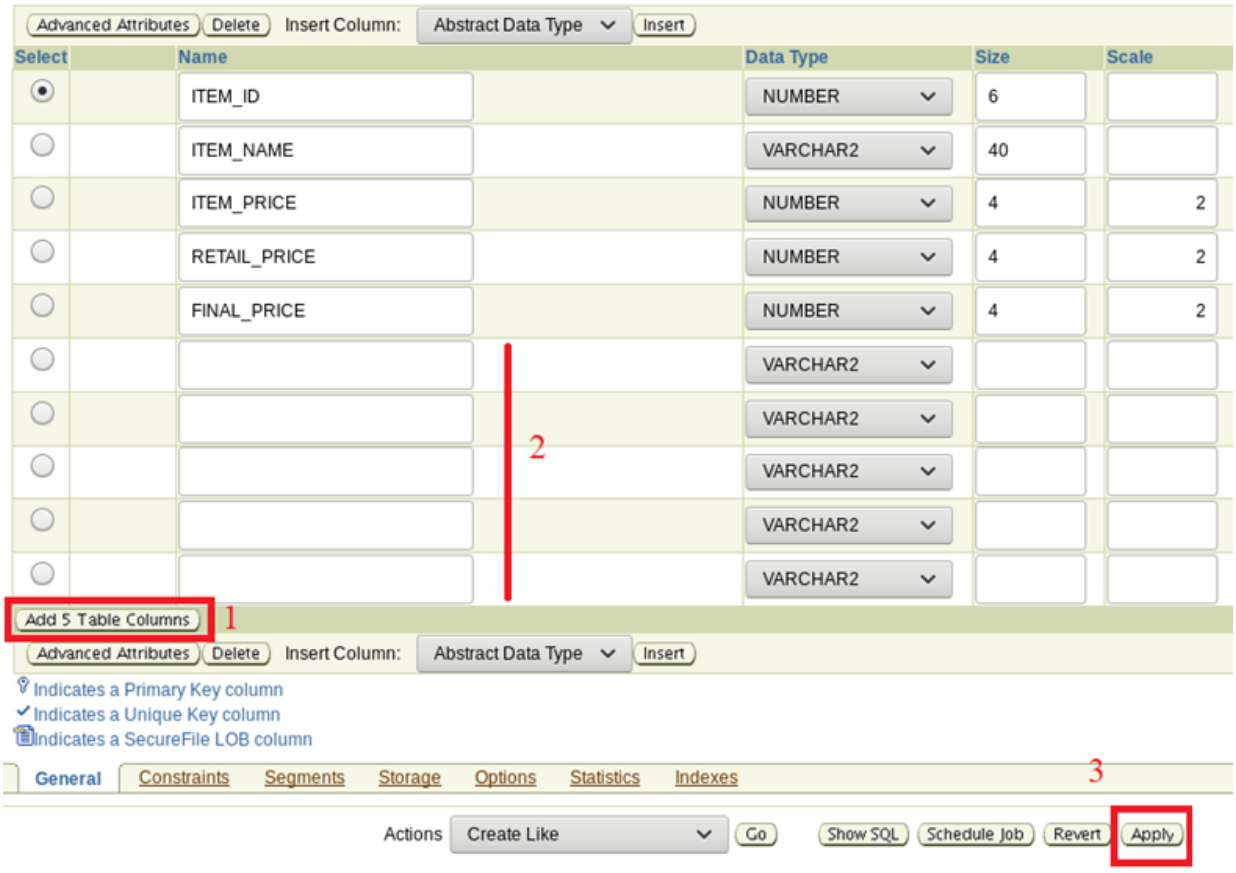


Fig. 14.31. Editing table. Add additional columns.

7.4. Understanding data integrity.

The Entity Relationship diagram in the slide depicts the data integrity constraints set out in the sample HR schema (Fig. 14.32.).

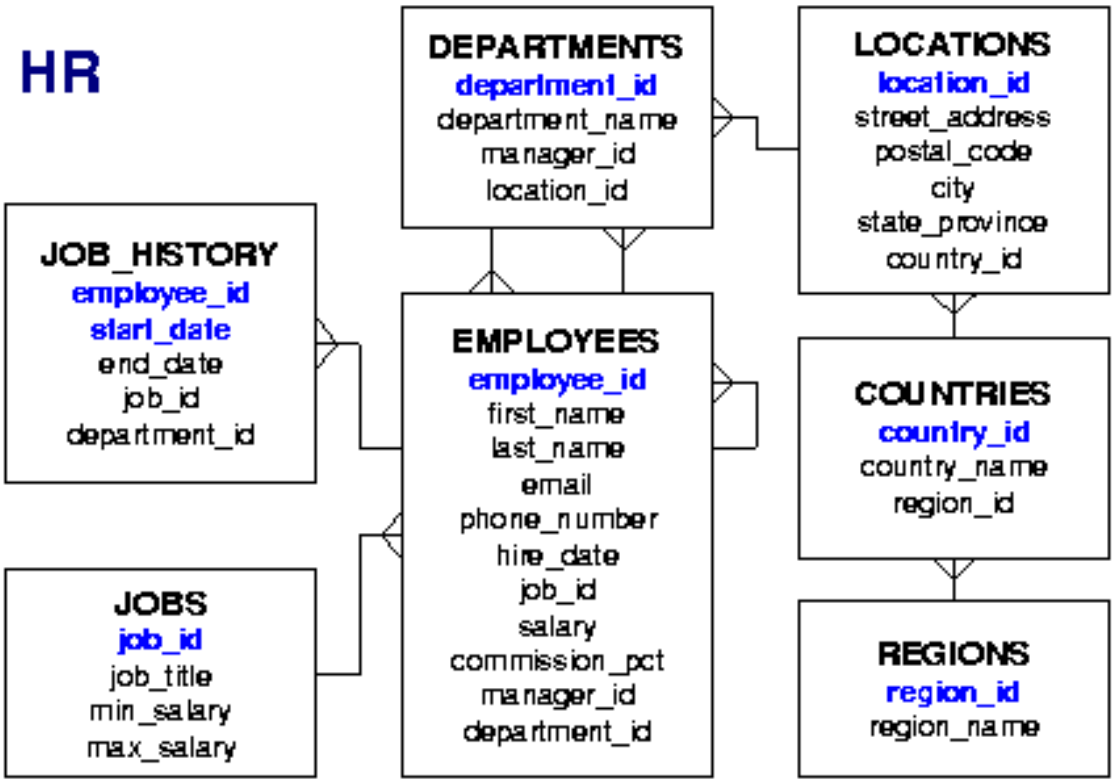


Fig. 14.32. Integrity of data in the HR scheme.

To impose constraints on entering values in columns, the following integrity constraints can be used:

• NOT NULL: By default, all columns in a table allow empty fields. NULL means no value. A NOT NULL constraint requires that a table column does not contain empty fields. For example, a NOT NULL constraint can be defined when entering a value in the LAST_NAME column for each row of the EMPLOYEES table.

• UNIQUE KEY: The UNIQUE KEY integrity constraint requires that each value in a column or set of columns (key) be unique - that is, no two rows in a table can have duplicate values in a particular column or set of columns. For example, a UNIQUE KEY constraint is defined in the DEPARTMENT_NAME column of the DEPARTMENTS table to disable rows with duplicate department names.

• PRIMARY KEY: Each table in the database can have at most one PRIMARY KEY constraint. The values in the group of one or more columns that are subject to this constraint represent the unique row identifier. Virtually every row is named after its PRIMARY KEY values.

Oracle's compliance with the PRIMARY KEY integrity constraint ensures that both of the following are true:

- There are no two rows in a table with duplicate values in the specified column or set of columns.

- The PRIMARY KEY columns do not allow a lack of value, ie the value must exist for the PRIMARY KEY columns in each row.

Oracle imposes all restrictions on PRIMARY KEY using indexes. The PRIMARY KEY constraint created for the DEPARTMENT_ID column in the EMPLOYEES table is imposed by the implicit creation of:

- unique index in this column;

- NOT NULL constraint for this column.

• FOREIGN KEY Restrictions: FOREIGN KEY restrictions are also called link integrity restrictions. The various tables in a relational database can be linked by common columns, and the rules that govern the connection of the columns must be maintained. The rules on reference integrity ensure that these relationships are maintained.

The limit on the integrity of the references requires that each row of the table match the value of FOREIGN KEY to a value in the "parent" key.

As an example, FOREIGN KEY is defined in the DEPARTMENT_ID column of the EMPLOYEES table. It ensures that each value in this column must match a value in the PRIMARY KEY of the DEPARTMENTS table (also the DEPARTMENT_ID column). Therefore, there can be no wrong department numbers in the DEPARTMENT_ID column of the EMPLOYEES table.

Another type of limitation of the integrity of references is called limitation of the integrity of own data. This type FOREIGN KEY refers to a parent key in the same table.

• CHECK constraints (condition check): The CHECK integrity constraint in a column or set of columns requires a certain condition to be true or unknown for each row in the table. If the Data Manipulation Language (DML) expression causes the CHECK constraint condition to be estimated to be incorrect, the expression is returned. For example, a CHECK constraint can be created in the EMPLOYEES table to ensure that the salary * Commission_pct <= 5000.

7.4.1. Set constraints.

Constraints can be added to a table:

1. Select the table in which you want to create constraints from the Tables page, then select Edit.

2. Constraints is selected (Fig. 14.33.).

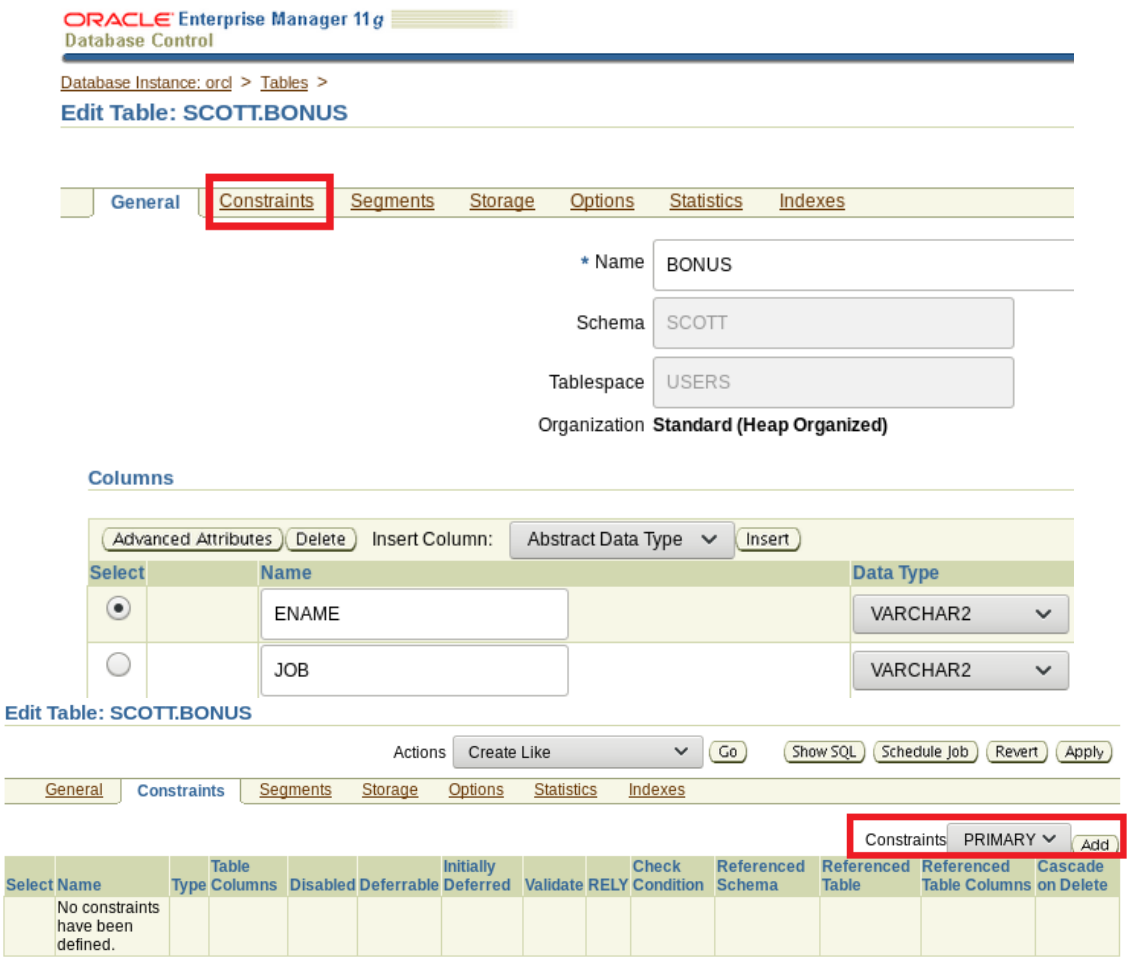


Fig. 14.33. Constraints.

The constraints page shows all the constraints that are defined in the table.

3. Select the type of constraint to add from the Constraints drop-down menu and add it by selecting Add.

4. Enter the appropriate information for the type of restriction to be defined. It is then confirmed (Continue) (Fig. 14.34.).

5. Select Apply.

A confirmation message is received that the table has changed.

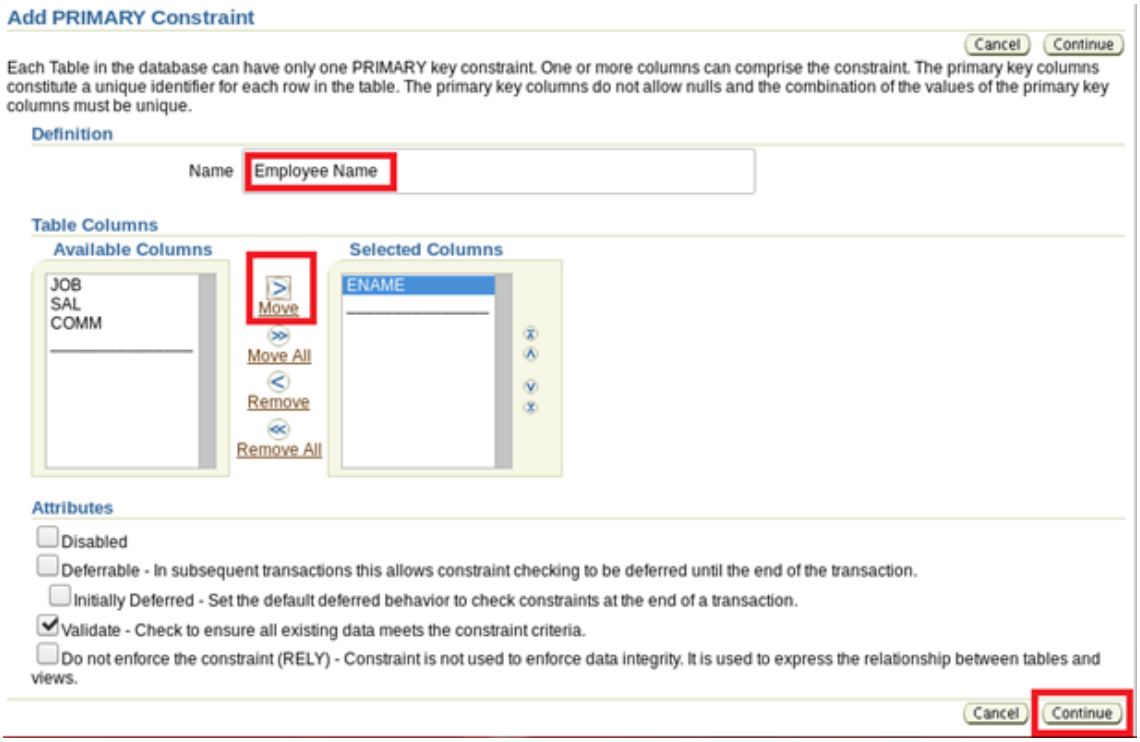


Fig. 14.34. Adding constraints.

7.5. View table attributes.

To view the attributes of a table using Enterprise Manager:

1. Select the Tables link in the Database Objects area of the Schema page.
2. Enter a schema name and start browsing with Go.
3. Select a table from the list and the View button to view the table's attributes (fig. 14.10).

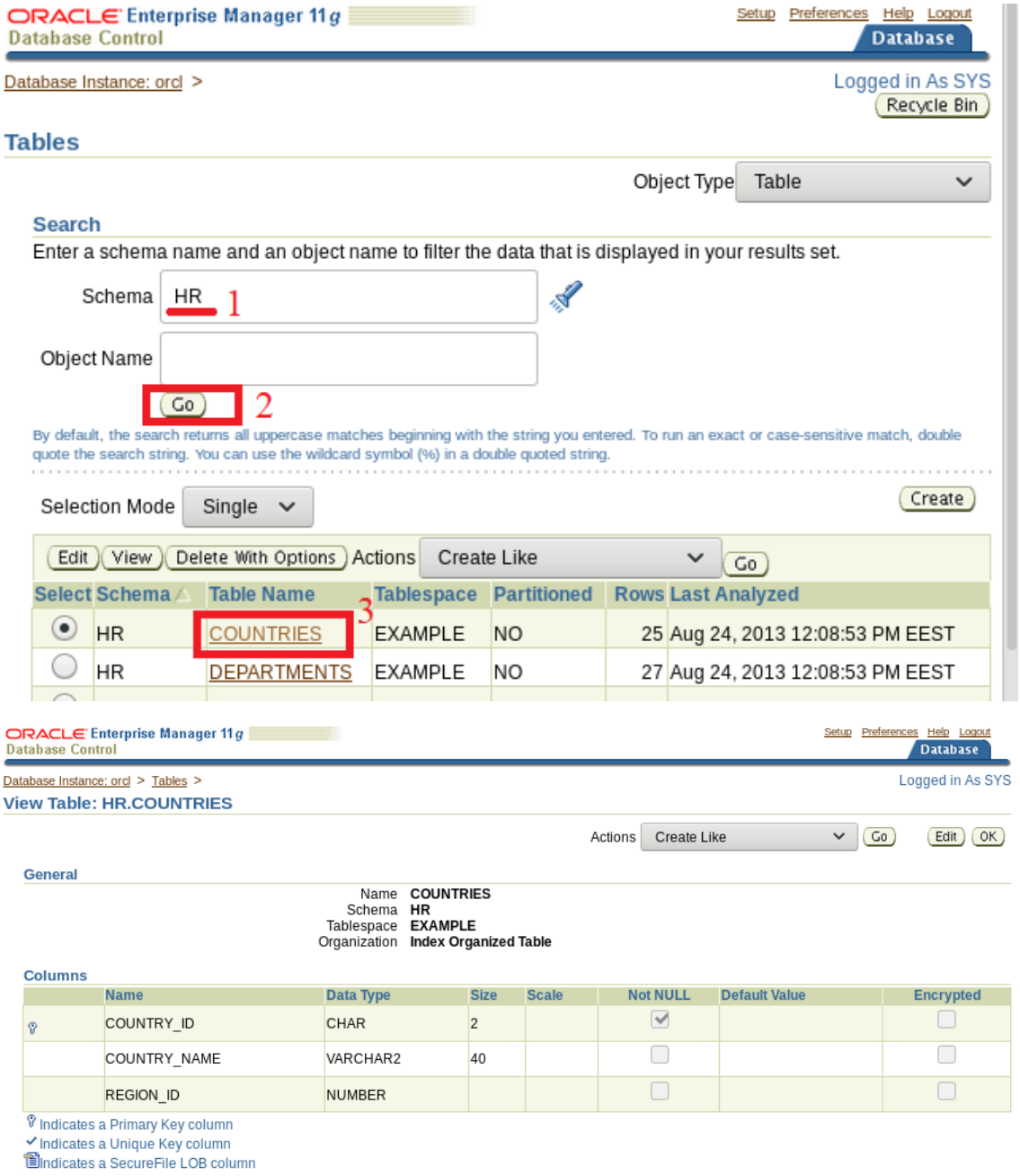


Fig. 14.10. View table attributes.

7.6. View the contents of a table.

View the contents of a table using Enterprise Manager:

1. Select a table to view from the Table page. Then select View Data from the Actions drop-down list and “Go” (fig. 14.11.).

The View Table Data page appears. The data for the table rows is displayed in the Result area. The Query box shows the SQL query that was executed to get the results (fig. 14.12).

2. On this page, when selecting a column, the data can be sorted in ascending or descending order. To change the query, select Refine Query.

3. On the "Refine Query for Table" page, you can select the columns to be displayed and specify a WHERE clause for the SQL statement to filter the results (fig. 14.13.).

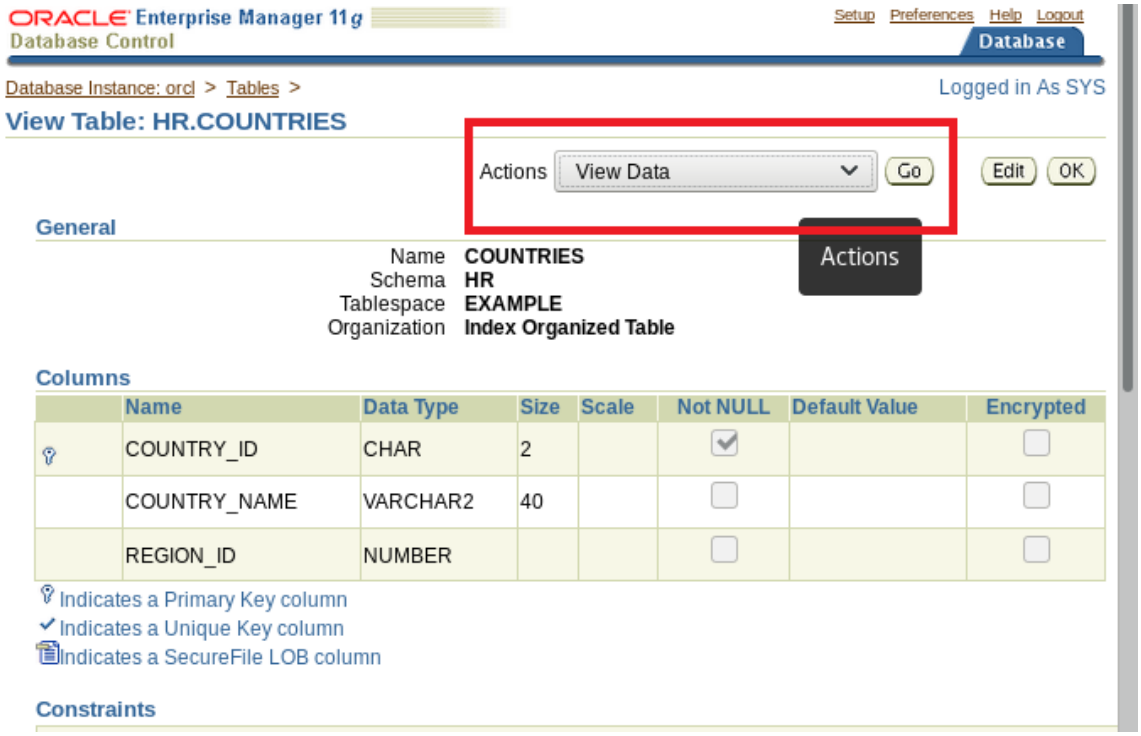


Fig. 14.11. Select a table to view its contents.

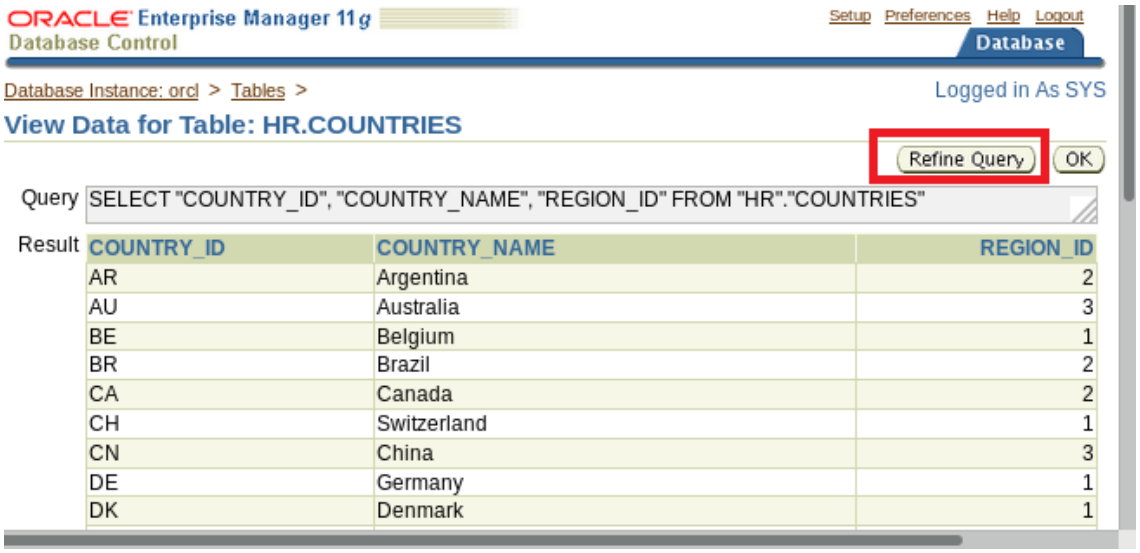


Fig. 14.12. View data in a table.

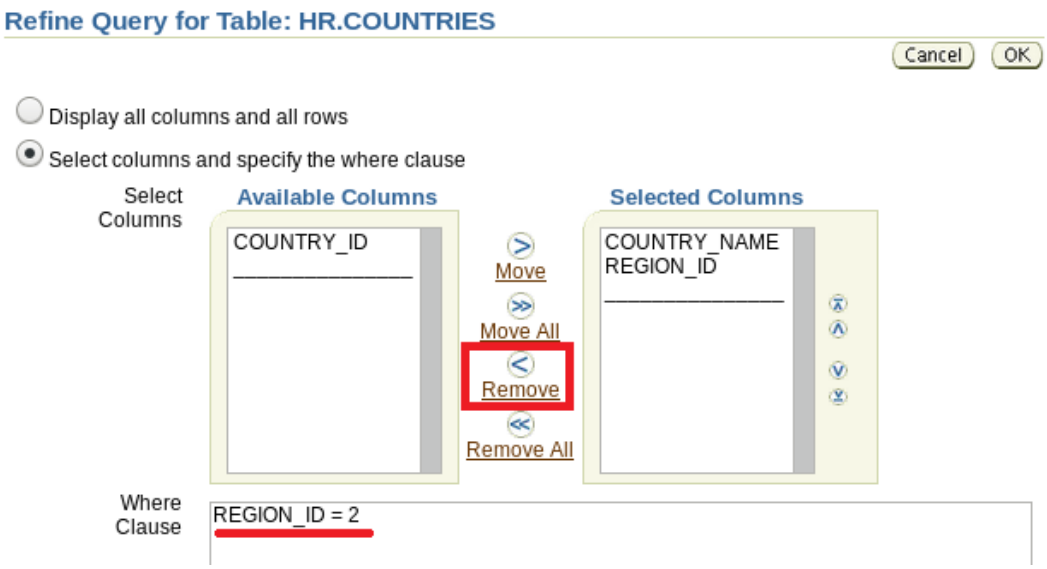


Fig. 14.13. Filter view of table contents.

7.7. Delete a table.

When a table is no longer needed in a user database and it is deleted, the following events occur:

- the table definition is removed from the data dictionary;
- all rows of the table are removed;
- all indexes defined on the table are removed.

Delete a table using Enterprise Manager to delete a table:

1. Select Tables in the Database Objects area of the Schema page.
2. Enter the schema name and table name or use the search tool to select the table. Then select Delete (fig. 14.14.).

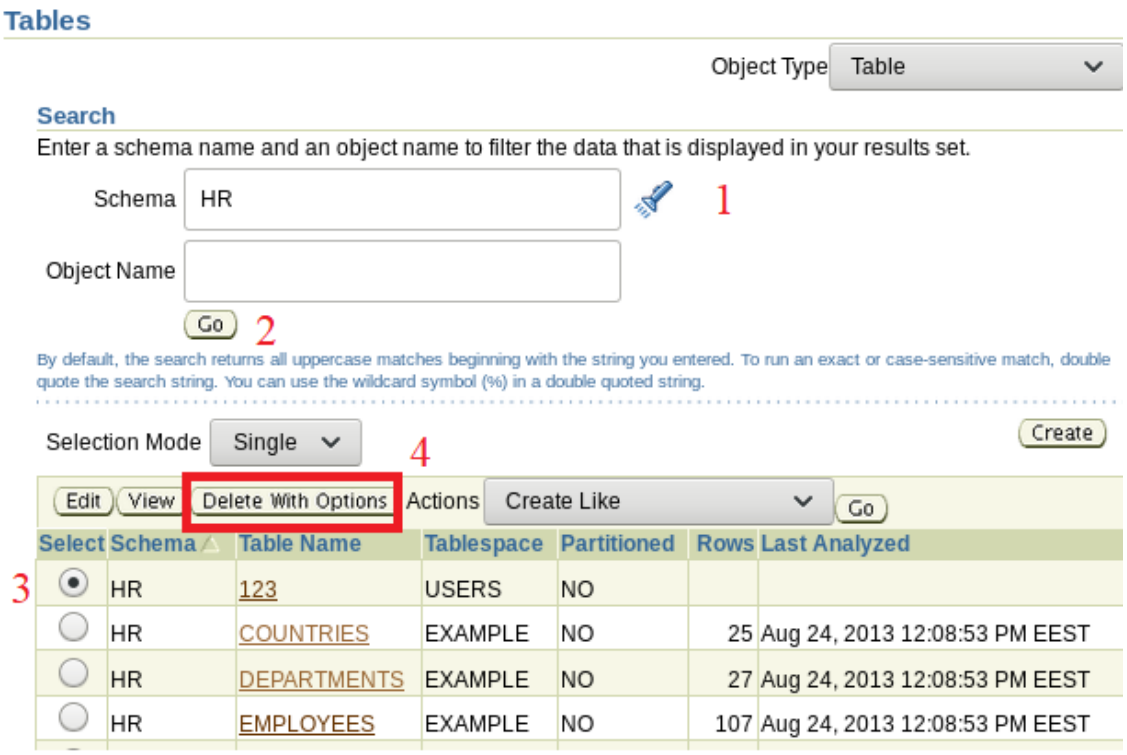


Fig. 14.14. Selecting a table to delete.

3. The confirmation page allows the user to choose whether to proceed with the operation. Validates if the correct table to delete is identified.

A confirmation message appears if the table has been successfully deleted (fig. 14.15.).

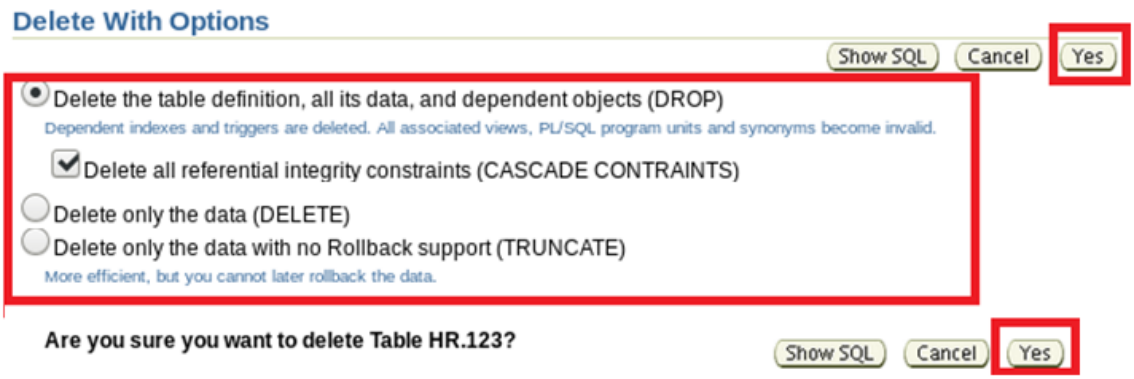
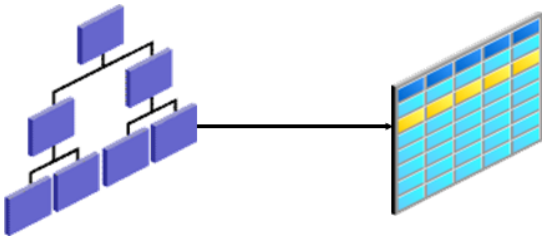


Fig. 14.15. Delete a table.

If a table is later found to have been deleted by mistake, it can be successfully recovered by applying the Flashback Drop feature.

7.8. Index management.



Indexes are optional structures that are associated with tables. They can be created to increase the performance of data retrieval. An Oracle index provides a direct access path to table data.

Indexes can be created on one or more columns per table. Once an index is created, it is automatically maintained and used by the Oracle server. When the table is updated, changes to the relevant indexes are made automatically with full transparency to the user.

The Indexes page is from Database Objects on the Schema page. Index attributes can be viewed or the Actions menu can be used to view dependencies for an index.

Indexes can be created explicitly, or they can be created implicitly through constraints that are placed on a table.

The index in short:

- Usually provides faster access to rows in a table;
- Automatically used to access data;
- It is physically independent from the table;
- It is maintained automatically when updating the table.

7.8.1. Creating and removing indexes.

To create a standard index (B-tree) on a single column, the following should be executed:

1. Select the Tables menu in the Database Objects area of the Schema page. The Tables page appears.

2. Enter the name of the table or use the search tool to select the table. Select Create Index from the action drop-down list and then click Go. The Create index page appears (fig. 14.16.).

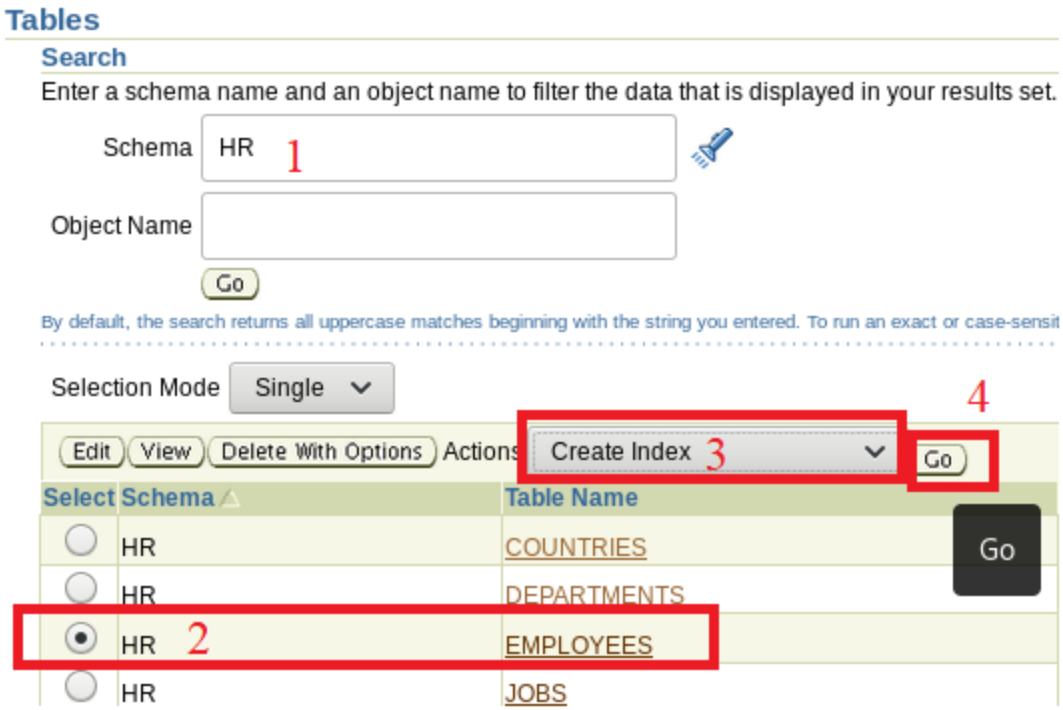


Fig. 14.16. Selecting a table to create indexes.

3. Enter a name for the new index. The name of the tablespace is entered and "Standard - B-tree" is selected as the index type (fig. 14.17.).

4. Select the column in the table column list by entering 1 in the Order column. Accept ASC as sort order. Then - OK to create the index. After the index is created, the Indexes page appears with a confirmation message. The new index is listed in the Result section.

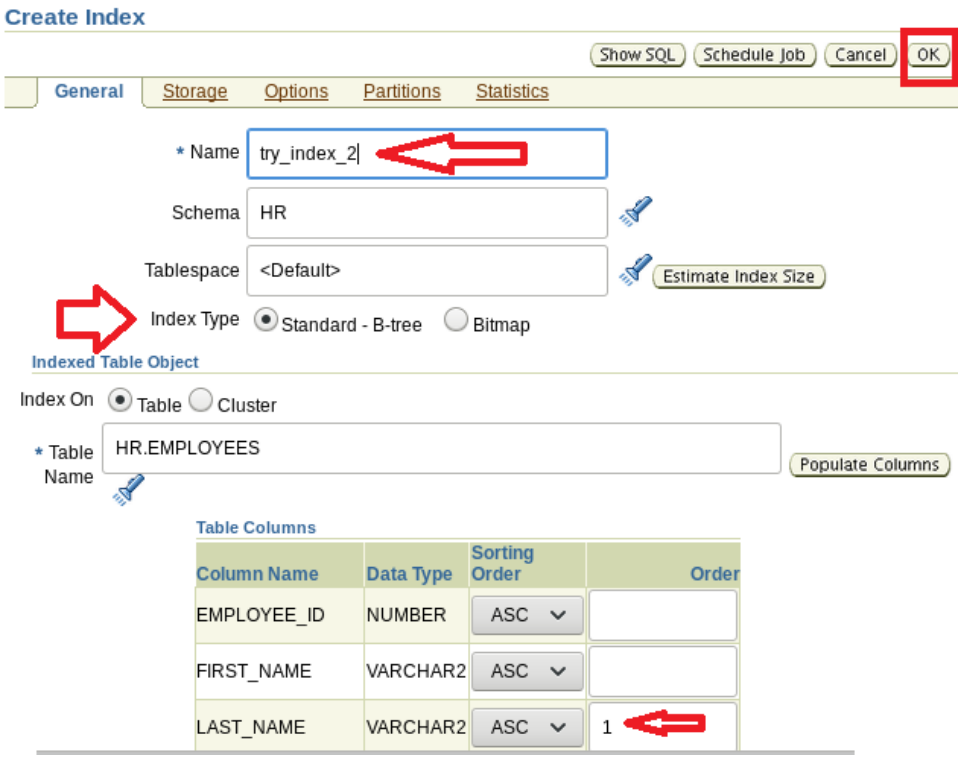


Fig. 14.17. Create an index.

To delete an index:

1. Select Indexes in the Schema area of the administration page. The Indexes page appears.

2. Type the name of the index or use the search tool to find the index. Then select Delete.

3. A confirmation window appears, and after confirming that the index has been successfully deleted (fig. 14.18.).

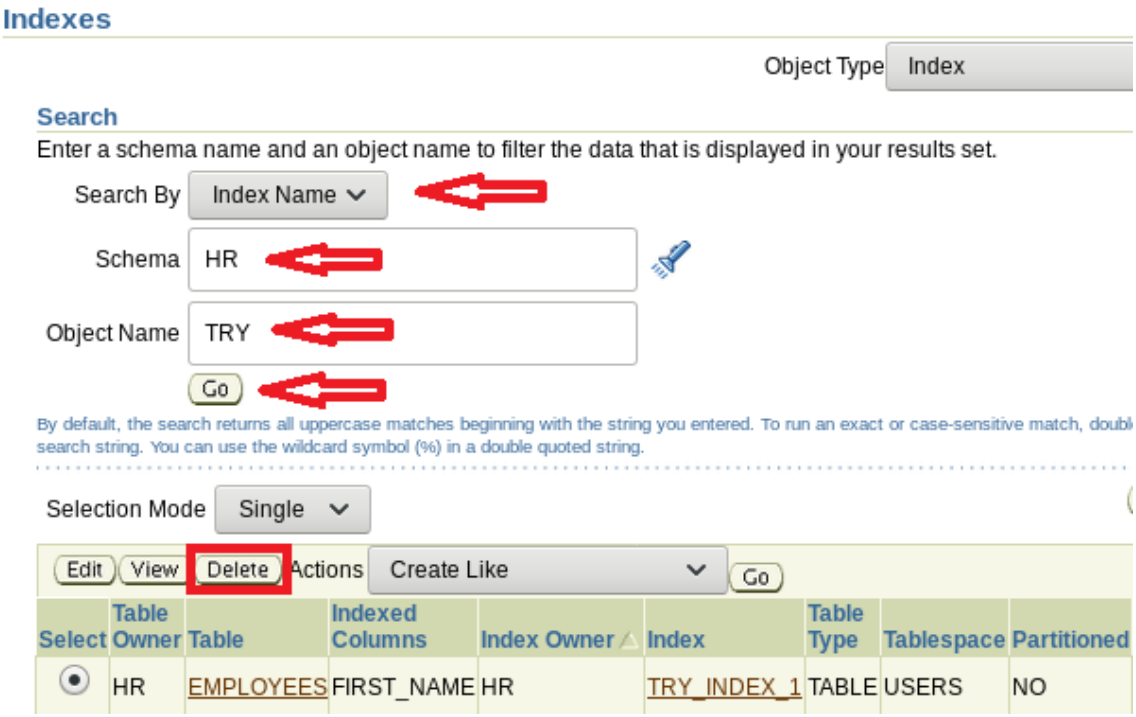


Fig. 14.18. Delete an index.

7.9. Manage views.

Views are customized representations of data from one or more tables or other views. These can be viewed as stored queries. Views don't actually contain data; instead, they derive their data from the tables on which they are based. These tables are called the base tables of the view.

Like tables, views can be queried, updated, inserted, and deleted (with some limitations). All operations performed on a view actually affect the underlying tables of the view. Views provide an additional level of security by restricting access to a predefined set of table rows and columns. They also hide data complexity and store complex queries.

To see the views defined in the database, select the Views link in the Database Objects area of the Schema page. With the help of Enterprise Manager views can be created (fig. 14.19.).

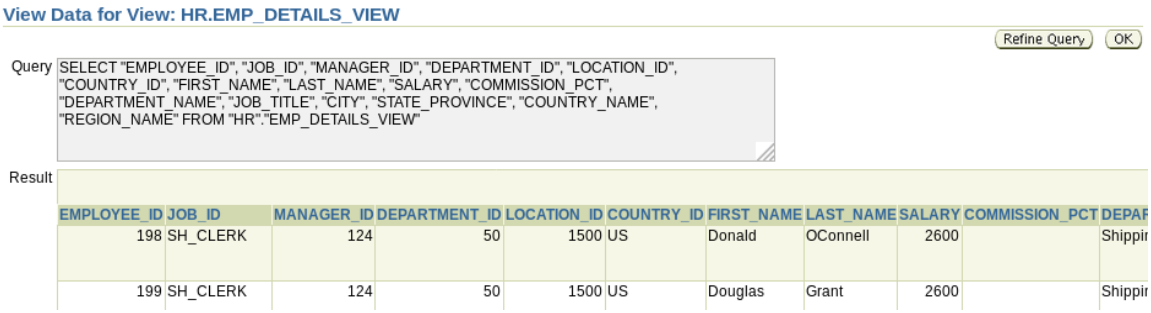


Fig. 14.19. Manage views.

7.10. Manage the program code stored in the database.

The following types of code objects can be managed through Enterprise Manager:

PL/SQL

- Packages;
- Procedures;
- Functions;
- Database triggers.

Java

- Program code;
- Compiled classes.

Database-resident programming units are programs written in PL/SQL or Java that are stored in the Oracle database. With database-resident programming units, frequently used code can be written once, tested, and called from any application program. This ensures consistent application of business rules and facilitates application development.

Enterprise Manager can be used to manage source types such as PL/SQL packages, procedures, triggers, functions, and Java sources and classes. Actions include creating, compiling, creating synonyms for granting privileges, and showing dependencies for these source types.

The following types of resident program units can be created in a database:

• Package: A structure that contains definitions, blocks of PL/SQL code, or both in a single device;

• Package bodies: Contains the PL/SQL code for the procedures and functions defined in the package definitions to which they belong;

• Self-contained subprogrames: Procedures and functions are self-contained subprograms. Functions must return a value to the calling program;

• Database Triggers: Stored subprogrames that are associated with a database table, view, or event.

7.10.1. Using Enterprise Manager to manage stored program code.

Packages can be viewed if they already exist in the database by selecting the Packages link on the Schema/Programs property page. The package properties page opens. Packages can then be searched by schema name, or object name, or neither. (The latter option will include every package in the result set.)

To view a package definition, select its name in the results list, or select the package from the results list, and then select Edit. The Edit Package property page contains the package name, schema, status, and source. From the Edit Package page, the package definition and package body can be modified and the package can be compiled. New packages can be created by selecting "Create" on the Packages property page. Packages can also be deleted there.

The other types of database resident programming units are similarly managed using Enterprise Manager.

7.11. Control access to schema objects.

By default, only the object owner or users with elevated system privileges can access a schema object. For any object that needs to be made available to other users, privileges must be granted to that object. Privileges can be granted directly to a user or to a role (fig. 14.20.).

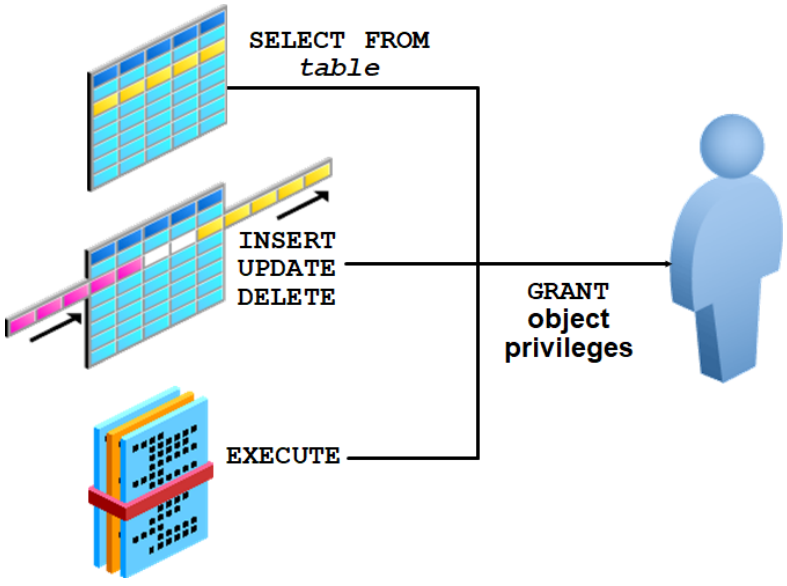


Fig. 14.20. Control access to schema objects.

7.11.1. Loading data.

Enterprise Manager can be used to load data into tables - in batches. Batch loading is useful when dealing with a lot of data. Data from files can be loaded on the operating system or from other databases. It is also possible to export data to files.

To load data with Enterprise Manager, one method is to use control (.ctl) and data (.dat) files. These files are formatted as standard SQL*Loader files.

The SQL*Loader control file should not be mistaken with the database control file. The SQL*Loader control file is used to describe the data file that is used when loading packages.

Data can be loaded in the following sequence:

1. Log in to Enterprise Manager as a user with administrative rights.

2. Select "Load Data from User Files" in the "Move Row Data" region of the "Data Movement" page to invoke a six-step loading wizard (fig. 14.21.).

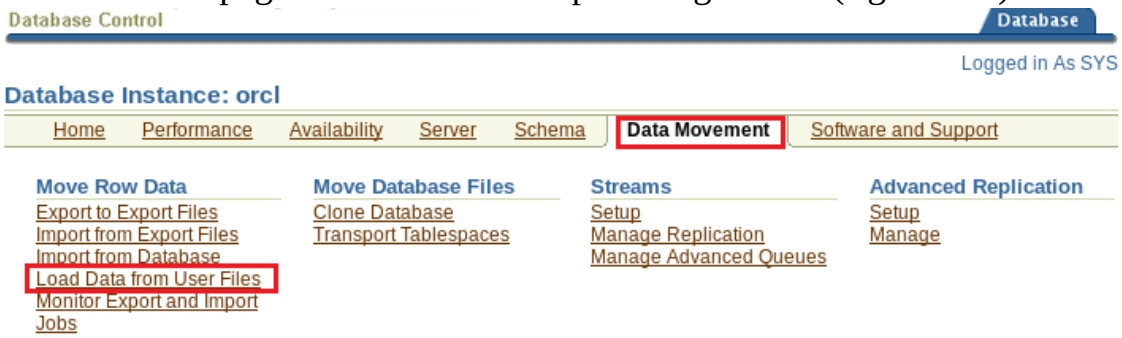


Fig. 14.21. Invoke a six-step data loading wizard.

3. Select Use Existing Control File on the Load Data: Control File page.

Here, the necessary files for the exercise are copied in advance to the specified default directory.

4. Enter the username and password for the machine. Continued /Next/.

5. Continue.

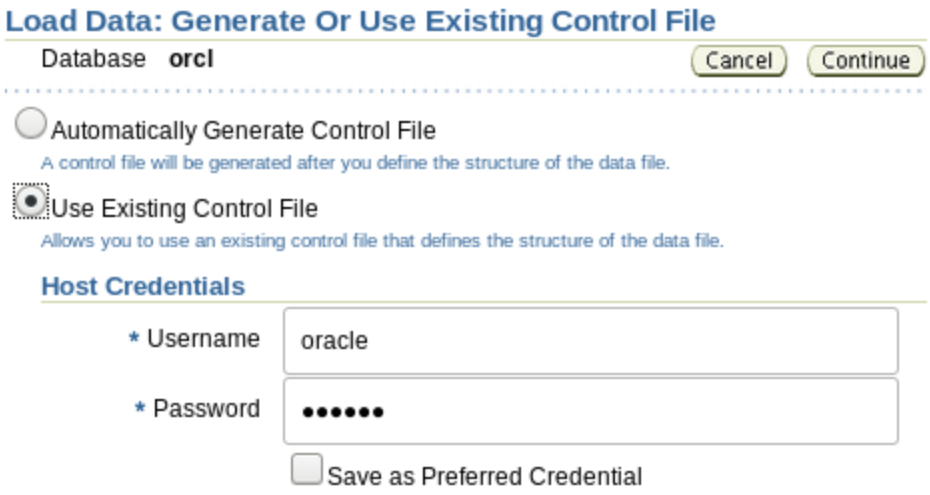


Fig. 14.22. Load data with an existing control file.

6. The location of the data file is specified on the Load Data: Data File page. The file location can be viewed with the flashlight button. A data file located on the machine can also be selected. With an existing control file, usually the data file is described in the control file (fig. 14.23.).

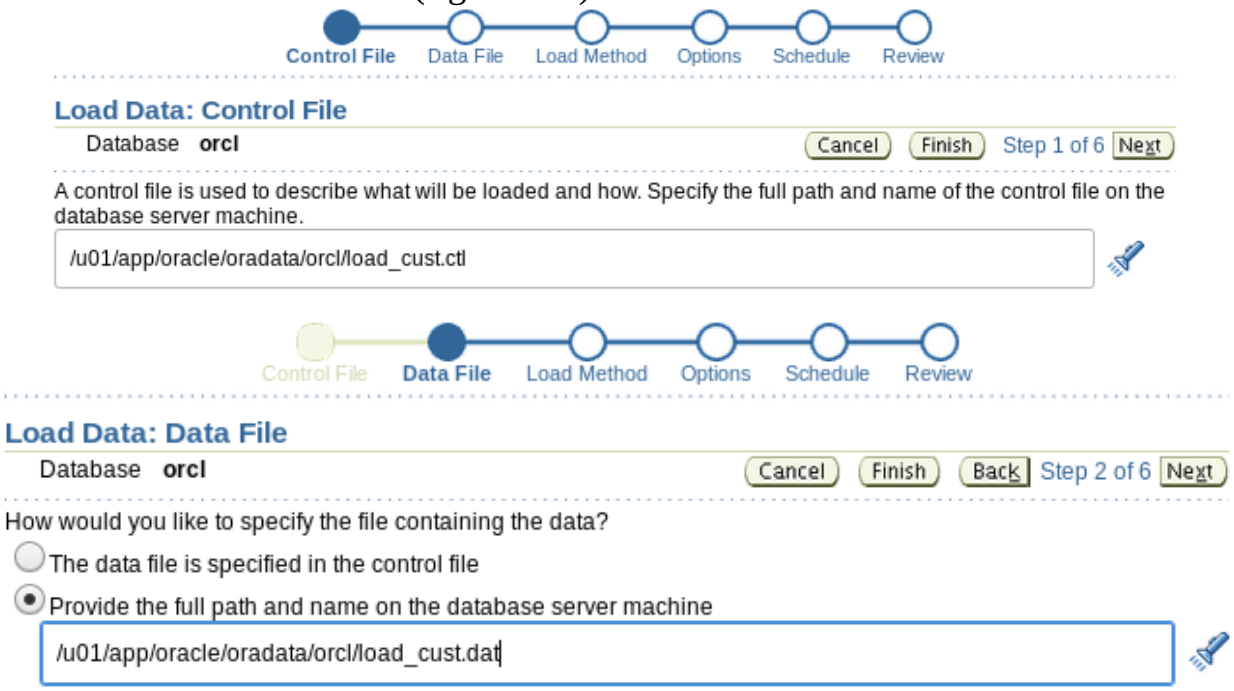


Fig. 14.23. Loading data. Control file and data file.

7. Select Next.

8. The charging method options are conventional path, direct path and parallel direct path. Both direct path options are faster than the conventional method.

On the "Data loading: loading method" page, Conventional Path is selected (fig. 14.24.).

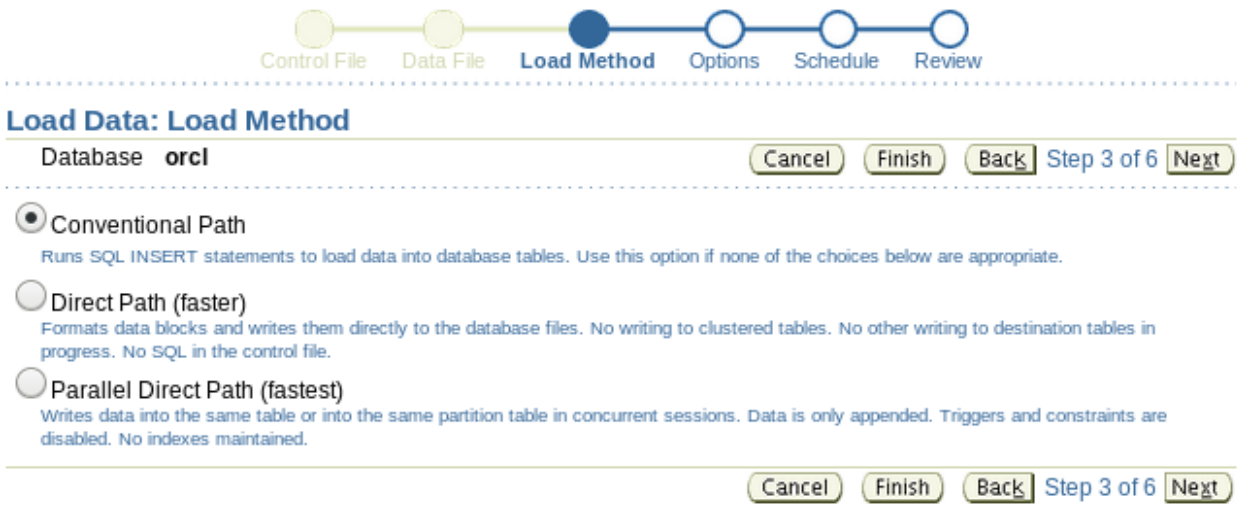


Fig. 14.24. Loading data. The loading method.

9. On the "Load Data: Options" page, confirm the selection of options and click Next.

10. On the Load Data: Schedule page, the information about the schedule is entered - when it is appropriate to perform the action, continue /Next/ (fig. 14.25.).

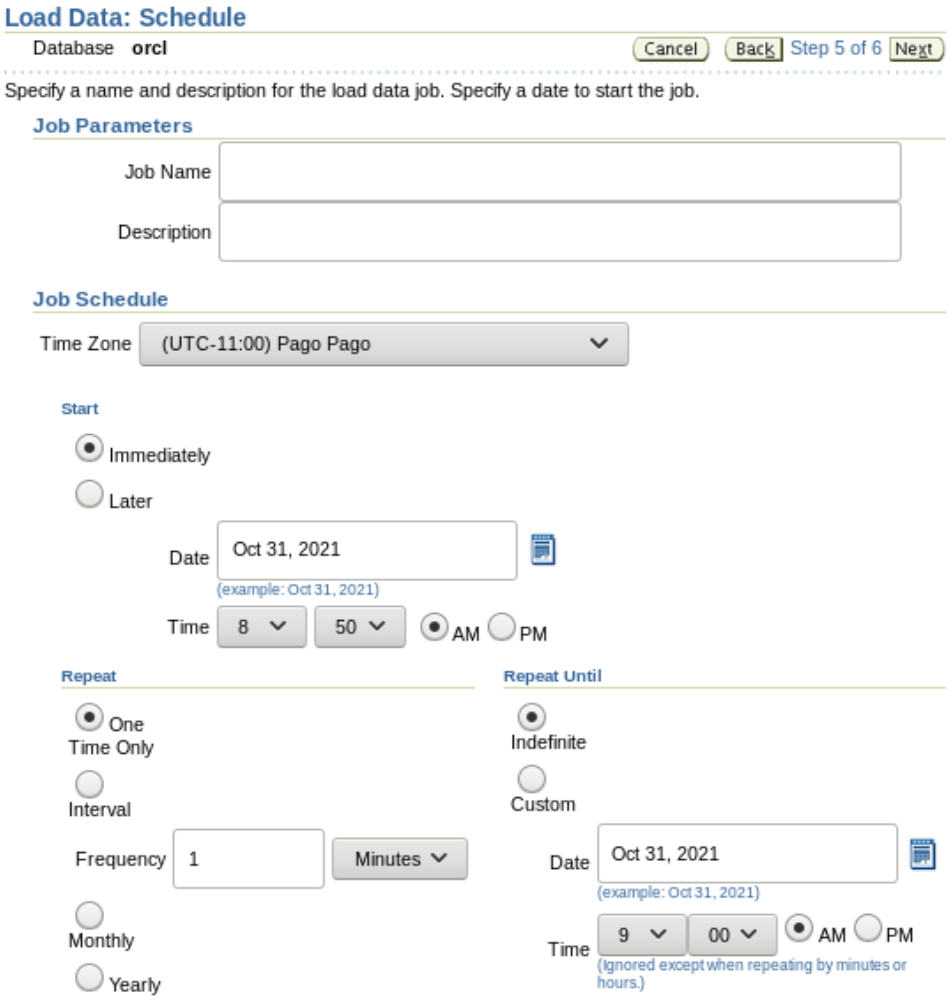


Fig. 14.25. Data loading schedule.

11. The Load Data View page checks the selection so far.
12. Select Submit Job.
13. Confirmation is received that the job was created successfully.
14. Via View Job, a summary of the job is viewed (fig. 14.26.).

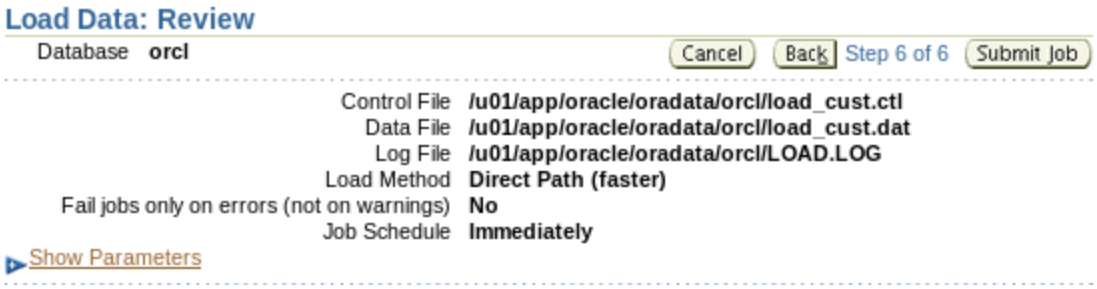


Fig. 14.26. Review of the data load job.

15. The summary page should show whether the task was successfully created. The log file can be viewed by clicking on the task under the Logs heading or by viewing the log file directly (fig. 14.27.).

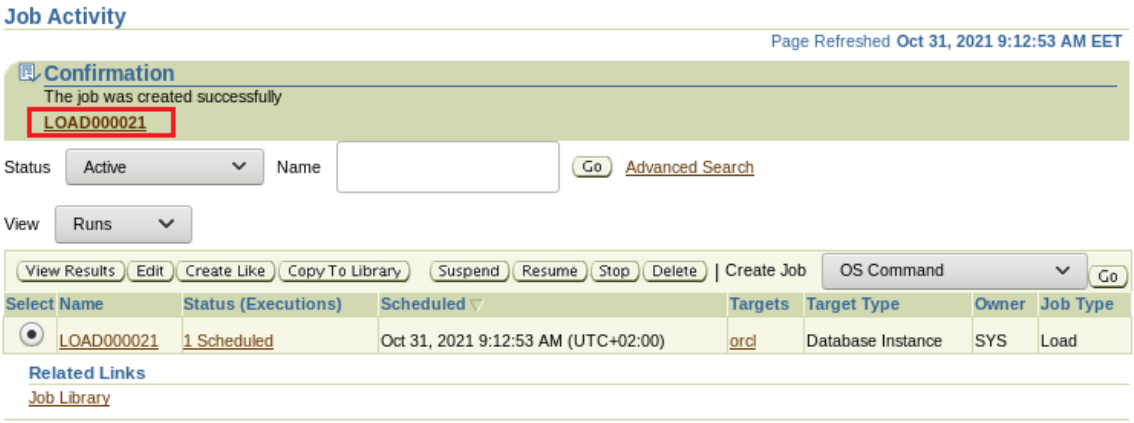


Fig. 14.27. Registering a data load job.

16. Loaded data can be verified by going to the Tables page, selecting the specific table and 'View Data'.