

Creating Tablespaces

Overview:

This document provides details of how to create, alter and delete Tablespaces within the Oracle database. A Tablespace is a storage unit that groups related logical structures together. Database data files are stored in tablespaces. Database data files are part of the tablespace

Restricted Use:

For security reasons, some permissions have not been provided to students to avoid accidental deletion or modification of other user accounts.

Tablespace Types and Characteristics:

There are two different types of tablespaces within Oracle: permanent and temporary. In addition to the permanent tablespaces used by end users, there are additional permanent tablespaces associated with System processes. In permanent tablespaces groups of schema objects persistently and permanently stored. The segments for objects in the tablespace are stored physically in data files. In temporary tablespaces transient data is stored but only persisting for the duration of a session. Temporary tablespaces are often used to improve performance of multiple sort operations that do not fit in memory.

The data dictionary elements reside in the System tablespace. Other components such as system rollback segments are stored in the System tablespace as well. The System tablespace is always created for you when you create a database. It can never be taken off line. It also cannot be dropped or renamed.

The SYSAUX tablespace serves as acts as an auxiliary tablespace to the SYSTEM tablespace, is also always created when you create a database. It contains the schemas used by various Oracle products and features, so that those products do not require their own tablespaces.

The UNDO tablespace is a locally managed tablespace reserved for system-managed undo data. A new installation of Oracle Database automatically creates an undo tablespace.

Tablespace Modes include Read/Write and Read Only. While in Read/Write mode users can read and write to the tablespace. When a tablespaces is initially created it is in read/write mode. SYSTEM and SYSAUX tablespaces and temporary tablespaces are permanently read/write. Tablespaces in Read Only prevent write operations to the data files in the tablespace.

Whenever a database is open, a tablespace can be online or offline. When the tablespace is online the tablespace is accessible. When the tablespace is offline the tablespace is not accessible. You can take a tablespace offline when maintenance or the database is undergoing backup and recovery.

With the appropriate database version, tablespaces can be encrypted to protect data-at-rest. This type of protection is aimed at protecting data from users who try to access database files directly through the operating file system. We will talk in more detail about encryption in future lessons. But be reminded, for an authorized user, who has properly authenticated and has been granted privileges to access the data, can access those objects without providing any kind of additional password or key.

Figure 1, taken directly from Oracle documentation, shows the close relationship between tables, tablespaces and physical data files running on the Oracle database server. Note that tables can span

multiple disks, and a tablespace can be on one or more physical disks. Protecting these files associated with the tablespaces is a critical component to any comprehensive security plan.

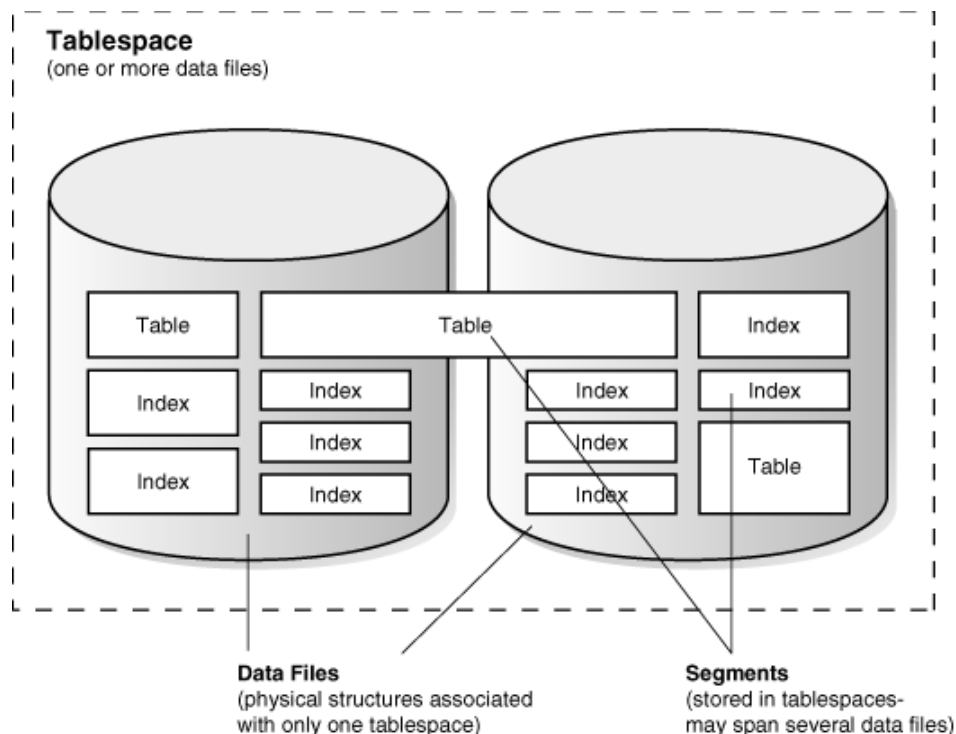


Figure 1 Relationship between Tables, Tablespaces and Data Files

Create Tablespace:

To create a Tablespace, you must have been granted the Create Tablespace system privilege. The following examples provide the critical syntax and possible parameters.

Figure 2 provides Oracle's diagram showing the major syntax for creating a Tablespace. Each additional tablespace clause provides more and specific options and clauses. By default Tablespaces are smallfile. Bigfile and Smallfile tablespaces are indistinguishable in terms of execution of SQL statements. A smallfile tablespace can contain multiple data files whereas a bigfile tablespace contains one very large data file.

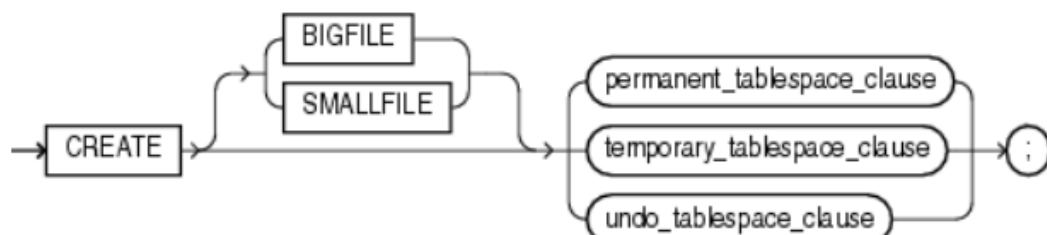


Figure 2 Create User Syntax

Figure 3 shows some of the many components when creating a permanent tablespace. Notice the options for encryption, compression, blocksize and others.

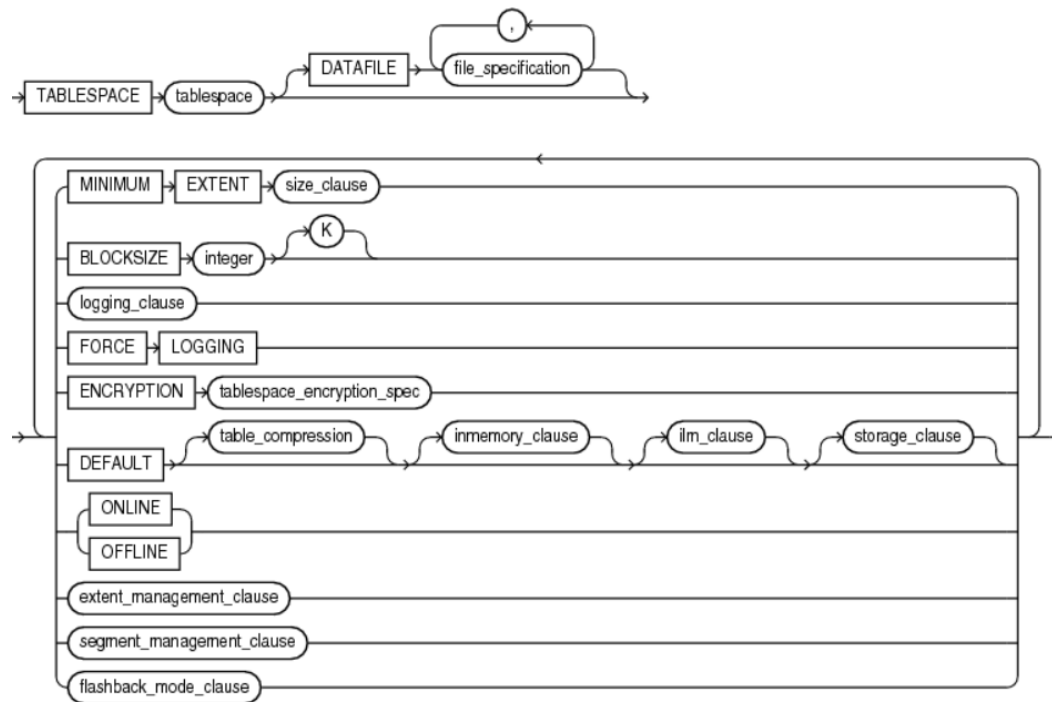


Figure 3 Permanent tablespace clauses

When a tablespace is created in the cloud, the clauses change slightly as they work to keep the physical file structure under their control. For example, Amazon RDS only supports Oracle Managed Files (OMF) for data files, log files and control files. When you create data files and log files, you can't specify the physical file names. However; AWS manages the files (and their names).

The following syntax creates an Oracle Managed File for a tablespace. In this example, a tablespace named Admins is created with autoextend enable with a maximum size of 500M. Notice there is no filename provided:

```
CREATE TABLESPACE Admins DATAFILE AUTOEXTEND ON MAXSIZE 500M;
```

To demonstrate Oracle (via AWS RDS) does create and manage the files for tablespaces, let us query the DBA_DATA_FILES data dictionary to determine the associated filenames and tablespace name for each of the available tablespaces. The following SQL statement was used to return this information.

```
SELECT FILE_NAME, BLOCKS, TABLESPACE_NAME
FROM DBA_DATA_FILES;
```

Figure 4 shows the result after running this query from a student account.

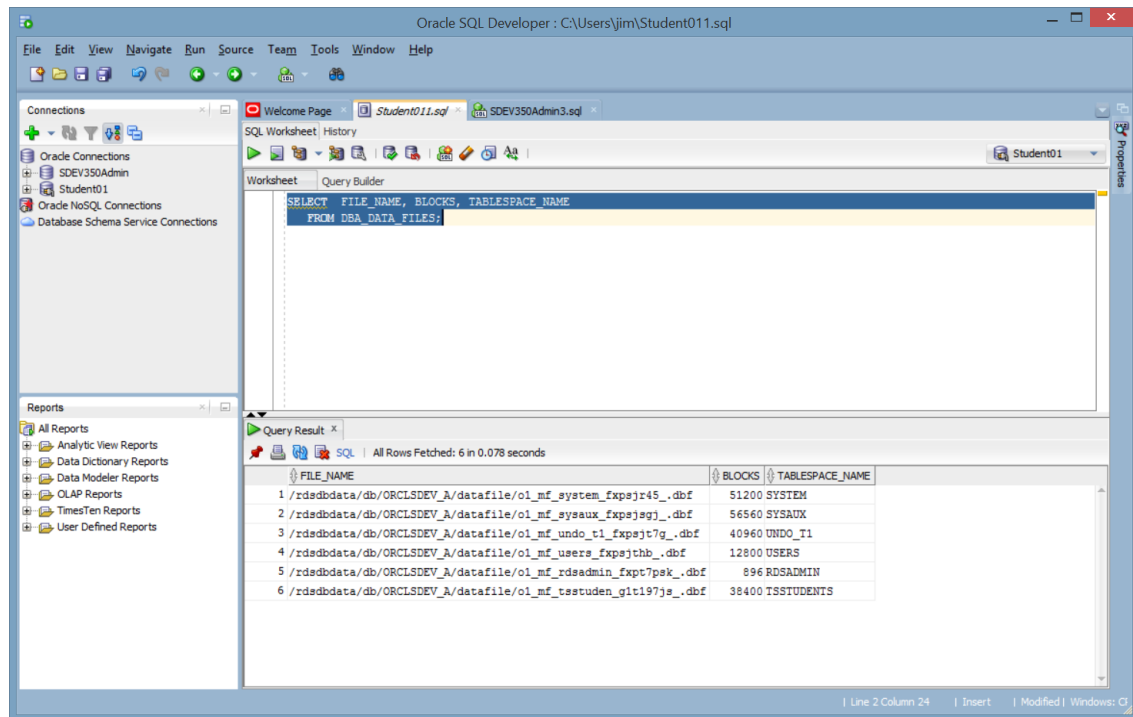


Figure 4 Determining the Physical Data Files for each Tablespace

The following are additional details about most of the Create Tablespaces clauses:

- **Minimum_Extent** - The minimum size of an extent in the tablespace. One extent for a segment may be stored in one file, while another can be stored in a separate. A single extent can never span data files.
- **BlockSize** - Specify a nonstandard block size for the tablespace. The default blocksize is 8192. Options for BlockSize range from 2048 to 32768.
- **Force_Logging** - Use this clause to put the tablespace into `FORCE LOGGING` mode. Logging changes to objects may be attractive for security audits. Be aware that logging all changes can slow performance.
- **Encryption** - Use this clause to specify the encryption properties of the tablespace.
- **Table_Compression** - specify default compression of data for all tables created in the tablespace
- **OFFLINE/ONLINE** – Set the tablespace online or offline. Temporary tablespaces cannot use this clause.
- **Autoextend On** – This clause allows Oracle Managed Files to automatically extend their size when needed. This is handy for systems that have unknown workloads or peak writing periods.

Figure 5 shows the results of successfully executing the previous Create Tablespace statement.

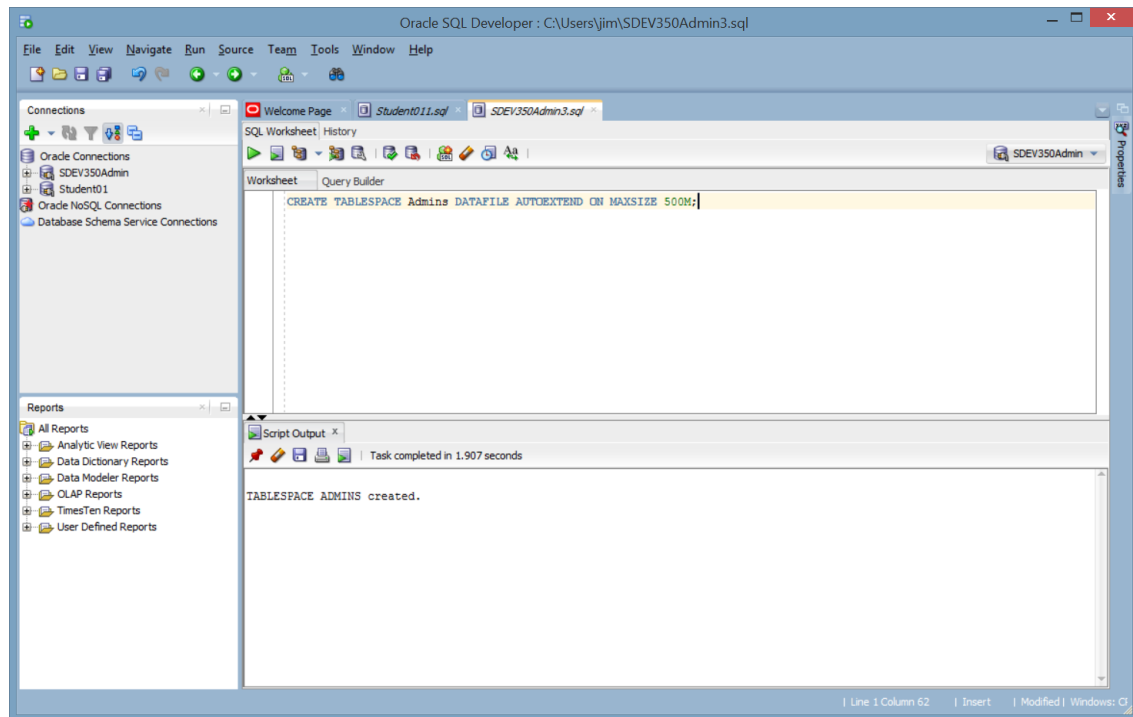


Figure 5 Creating a Tablespace

You can use the Data dictionary views to verify the tablespace was created. For example, the following query will show the filename, number of blocks and tablespace name for all tablespaces in the database.

```
SELECT  FILE_NAME, BLOCKS, TABLESPACE_NAME
        FROM DBA_DATA_FILES;
```

Figure 6 shows the results of running this query using a student account in SQL Developer.

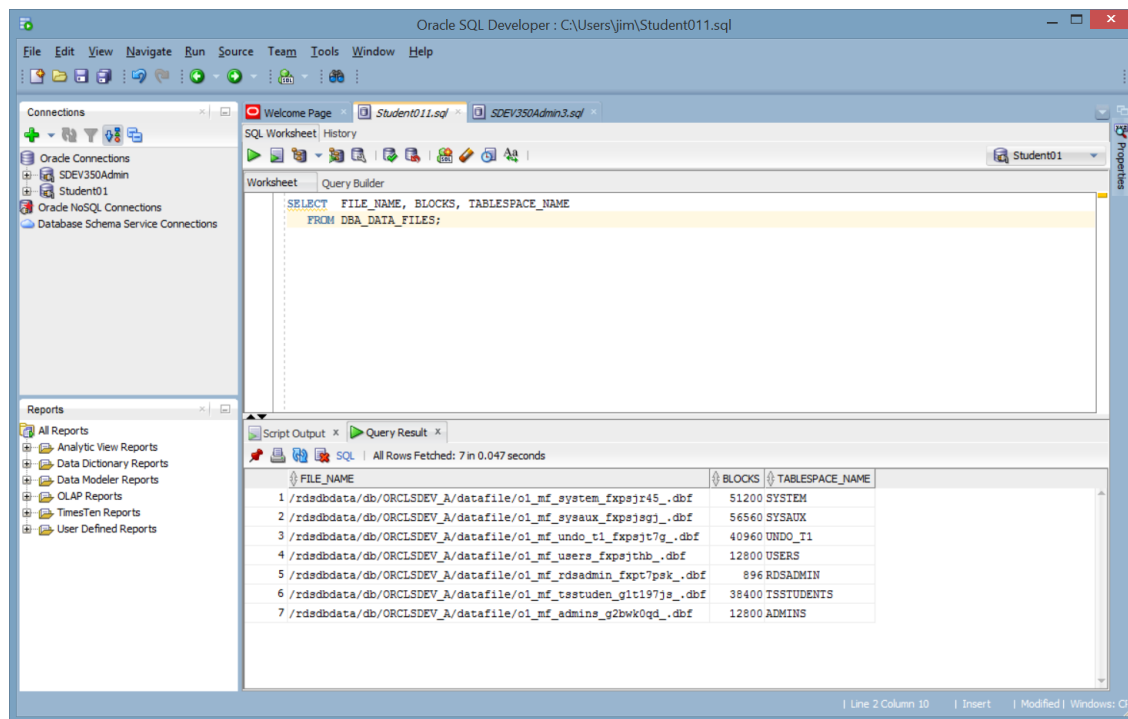


Figure 6 Verifying the User Creation with the Data dictionary

You can also use the DBA_TABLESPACES data dictionary view to list tablespaces and other information. For example, the following query will list the Tablespace name, block size and status for all tablespaces.

```
select tablespace_name, Block_Size, status from DBA_Tablespaces;
```

Executing the query from the student account in SQL Developer will provide the output as shown in figure 7.

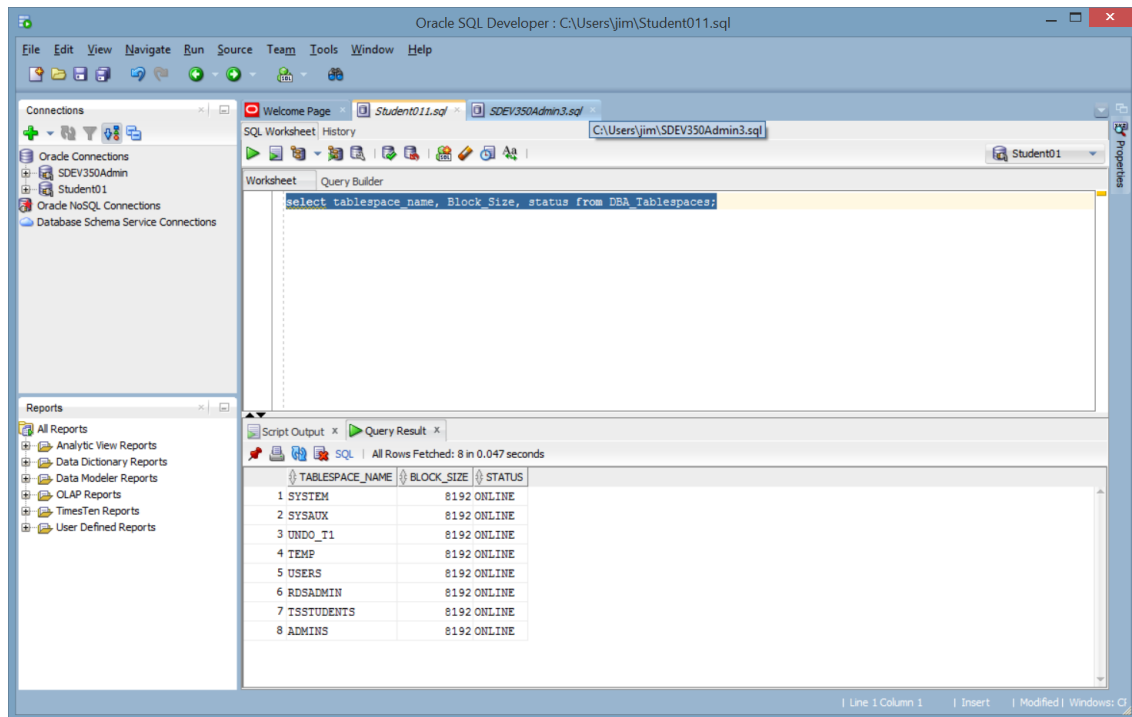


Figure 7 Listing the Tablespaces in Oracle

Altering a Tablespace:

The Alter Tablespace statement can be used to alter an existing tablespace. With the Alter tablespace statement, multiple actions can take place including, move a tablespace online or offline, start or end a backup, make a tablespace read only or read write, change the state to PERMANENT or TEMPORARY, set the default logging to LOGGING or NOLOGGING, rename a tablespace and more.

You do need Alter Tablespace permissions for a successful Alter call. If provided with the Alter User privileges, running this statement would yield a successful response as shown in figure 8.

```
ALTER TABLESPACE Admins RENAME TO AdTools;
```

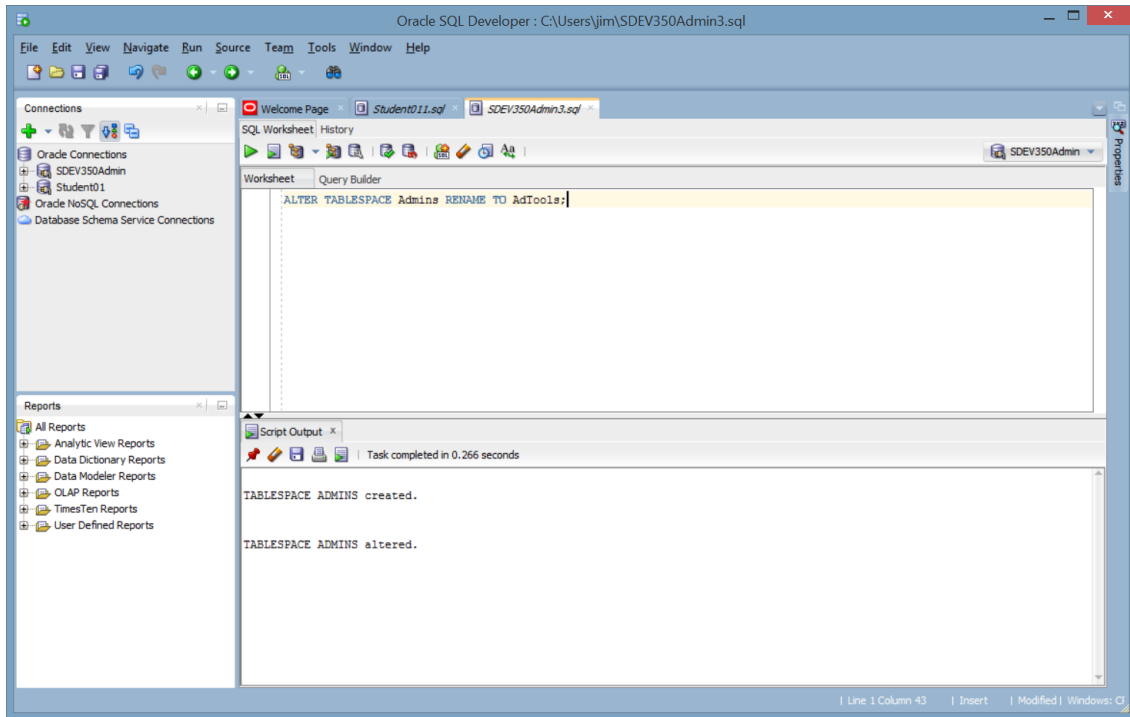


Figure 8 Altering a Tablespace to Rename it

Executing the following data dictionary query confirms the tablespace was altered (See figure 9).

```
select tablespace_name, Block_Size, status from DBA_Tablespaces;
```

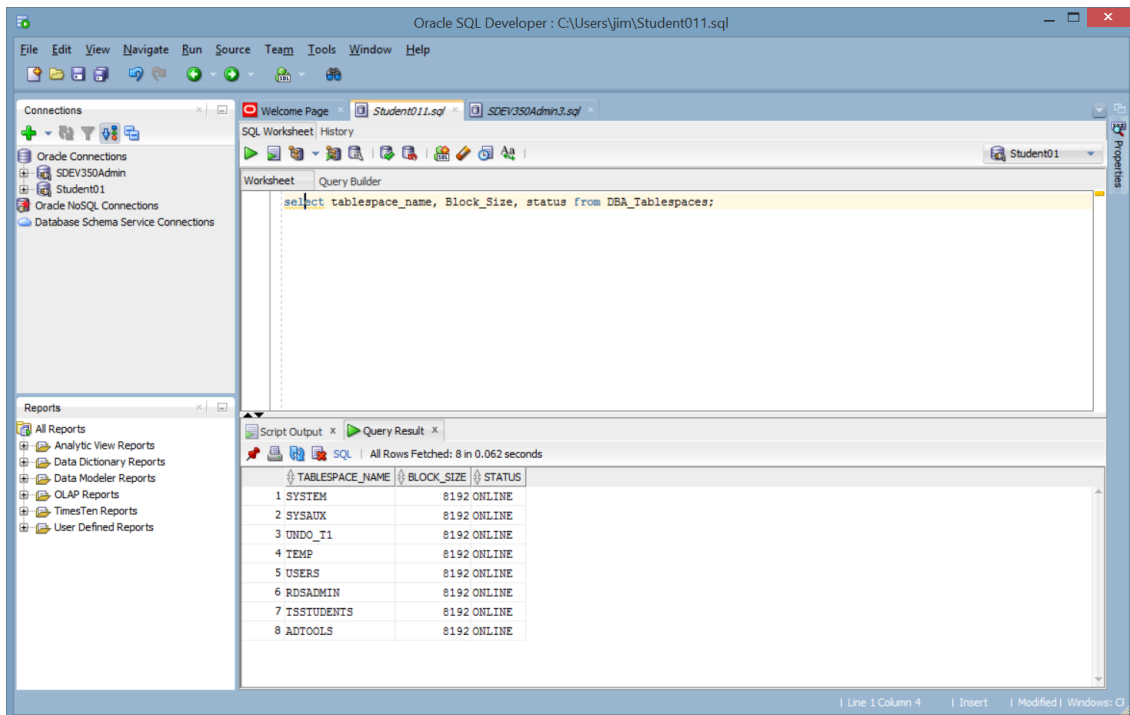


Figure 9 Renaming the Tablespace

Additional options available for the Alter Tablespace syntax are shown in figure 10.

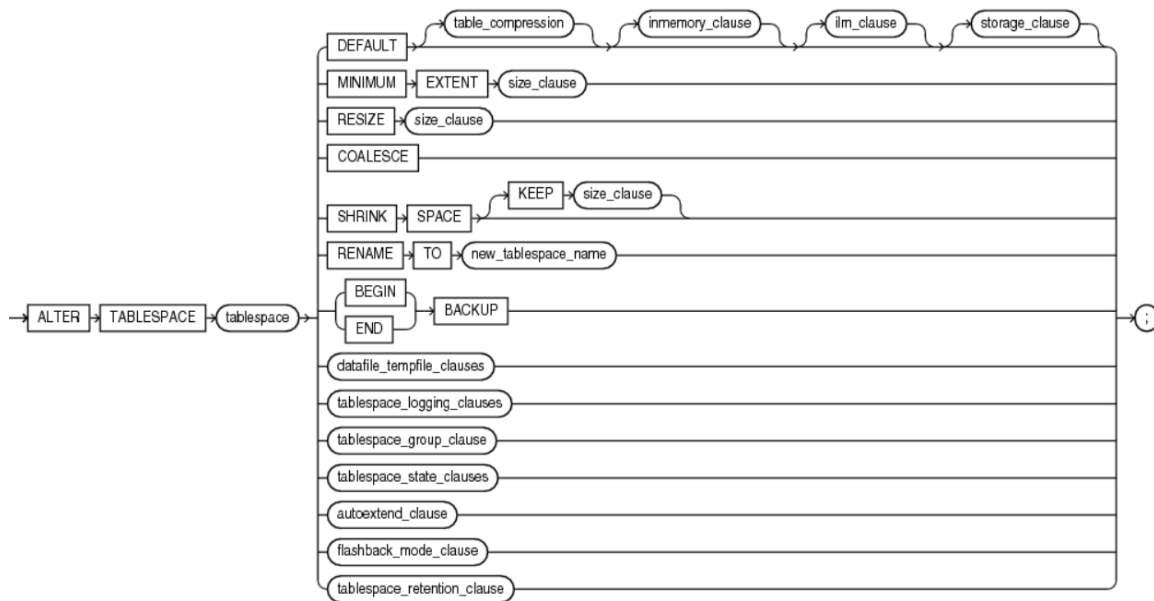


Figure 10 Alter Tablespace options

Drop a Tablespace:

With the proper Drop Tablespace permissions, a tablespace can be dropped (deleted). You do need the Drop Tablespace permissions for a successful Drop call.

As shown in figure 11, there are multiple options for dropping a tablespace that include the ability to cascade constraints and discard contents or keep or discard datafiles.

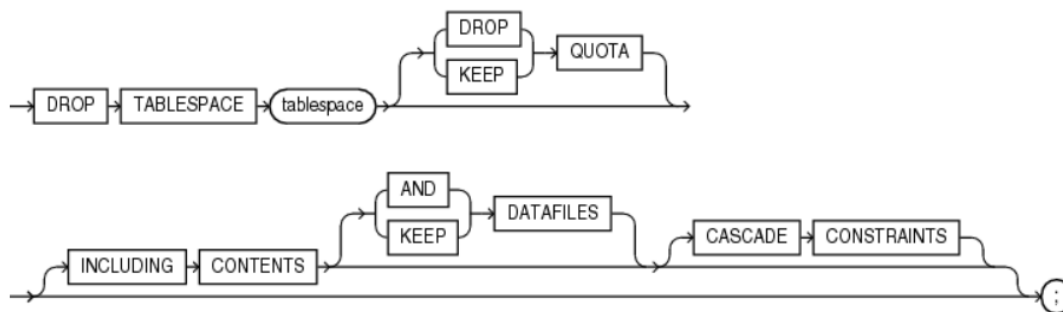


Figure 11 Drop Tablespace Syntax

Executing the following SQL statement will drop the AdTools tablespace along with the contents and datafiles. (See figure 12).

```

DROP TABLESPACE AdTools
INCLUDING CONTENTS AND DATAFILES;

```

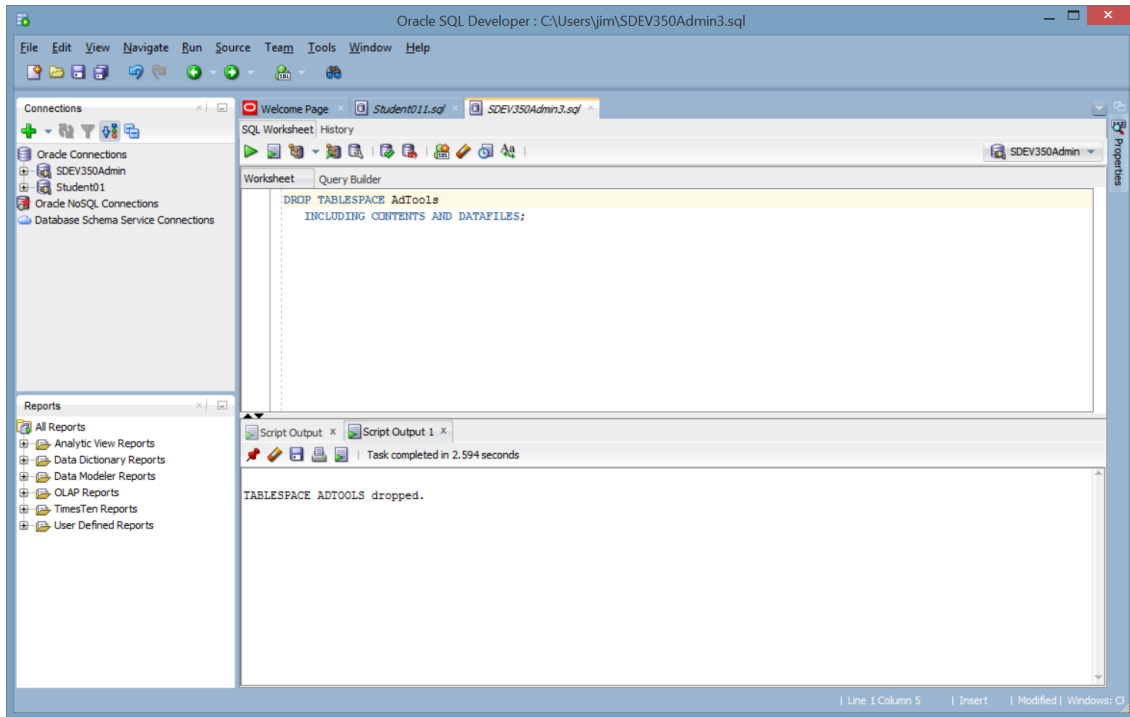


Figure 12 Dropping a Tablespace and the Contents and DataFiles

Successfully dropping the Tablespace will remove it from the DBA_Tablespaces view as shown in Figure 13.

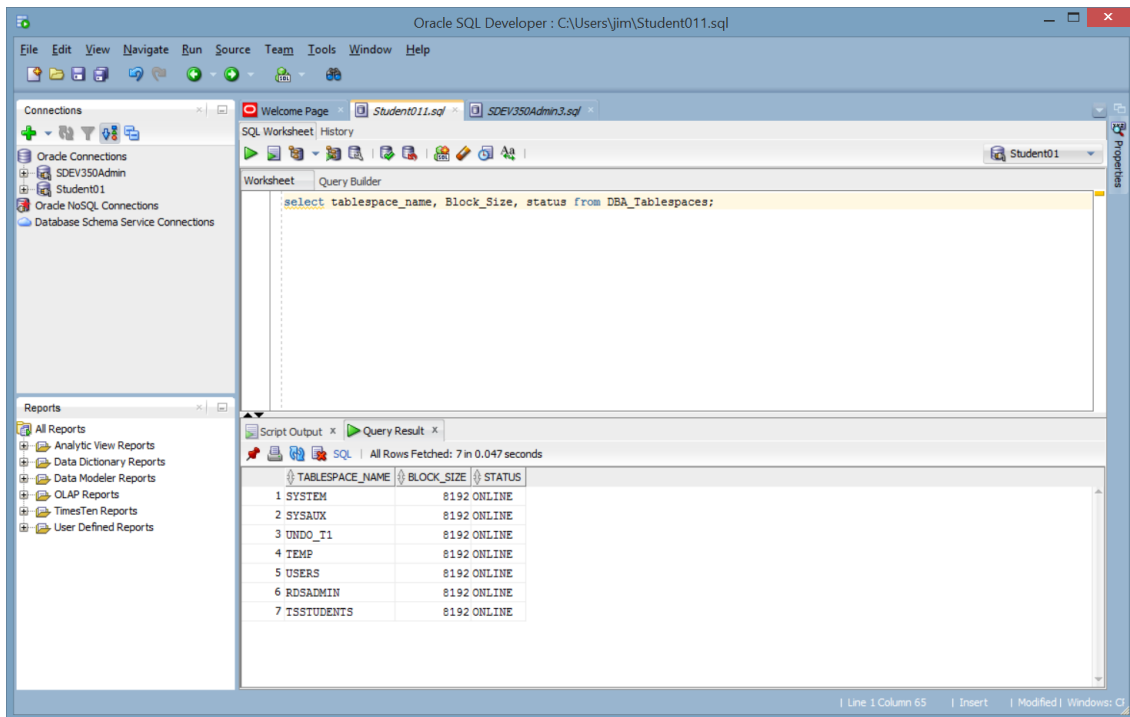


Figure 13 Verifying the Tablespace was Dropped

Once a tablespace has been dropped, the data in the tablespace is not recoverable. Therefore, ensure that all data contained in a tablespace to be dropped will not be required in the future. Also, immediately before and after dropping a tablespace from a database, back up the database completely.

As you can see, the Alter and Drop tablespace privileges can be very dangerous in the wrong hands. Many organizations have privileges such as these team controlled and very restricted. This ensures one person doesn't destroy an entire set of data stored in a database.

Tablespace Security Summary:

Some tablespace security best practices are worth noting. In a typical database, multiple tablespaces should be used. For example separate tables should be used to store user data and each major application. If possible, data files should be stored across different disks minimizing the possibility of data loss. Back-up tablespaces on a regular basis.

Finally, be sure to assign tablespace quotas for users. Quotas allocate space allowance to limit how much space a user is provided. You can always add more if the need arises. With Unlimited Tablespace rights, a user could accidentally or willfully take over a tablespace, fill up your disk and impact most other users of the database.