

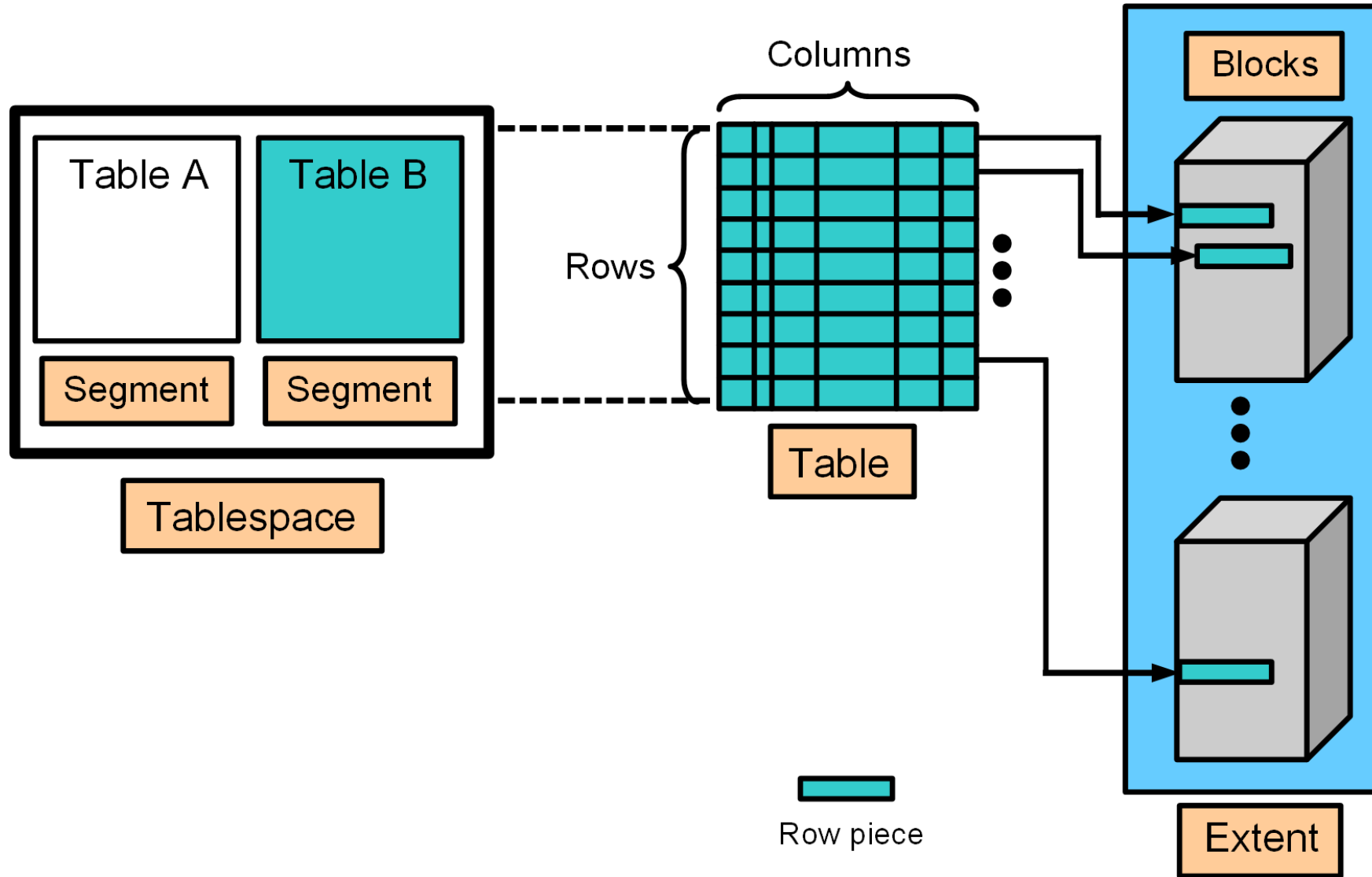
Creating and Managing Tablespaces

Objectives

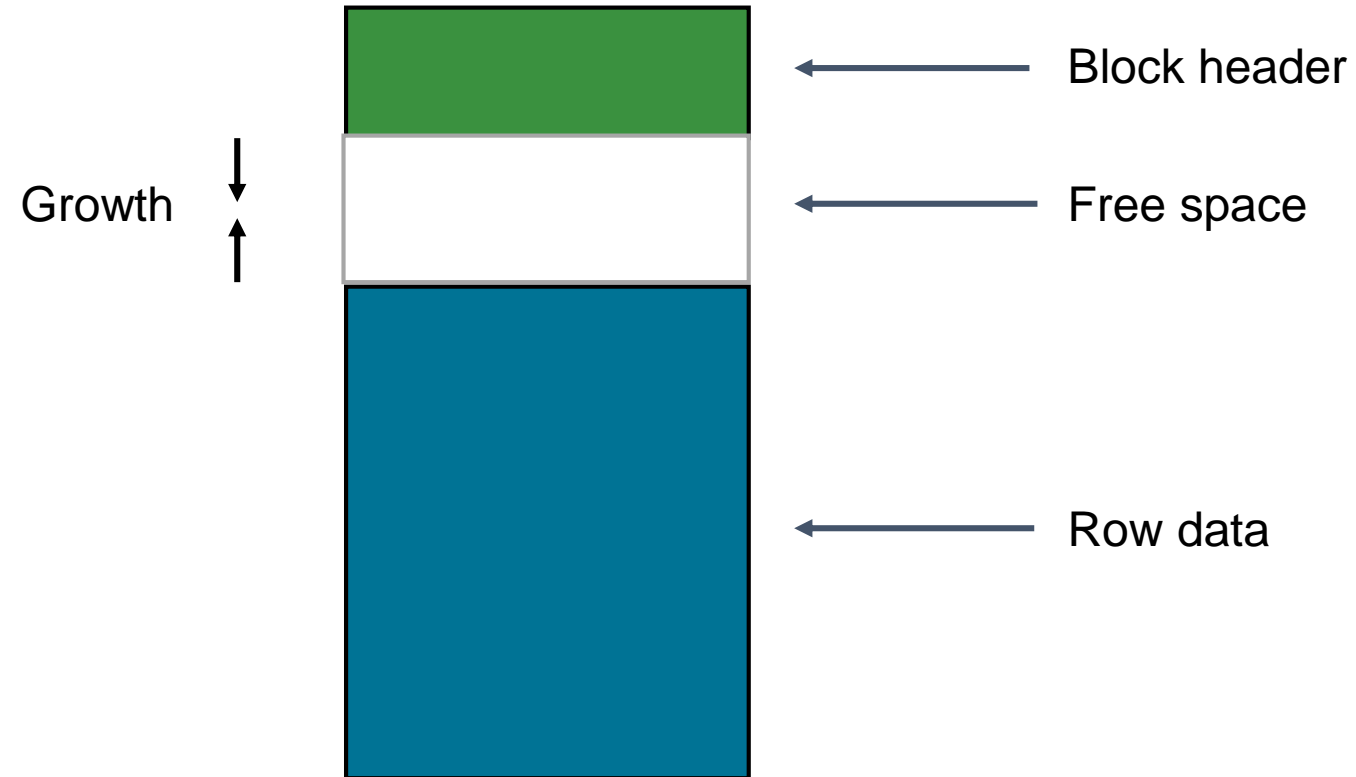
- After completing this lesson, you should be able to:
 - Explain how table data is stored in the database
 - Use SQL*Plus to:
 - Create and drop tablespaces
 - Alter tablespaces
 - View tablespace information
 - Implement Oracle Managed Files (OMF)
 - Use SQL*Plus to move and rename online data files
 - Implement tablespace encryption



How Table Data Is Stored



Database Block Content



Creating Tablespaces

- A tablespace is an allocation of space in the database that can contain schema objects.
- Create a tablespace with the `CREATE TABLESPACE` statement or a graphical interface, such as EM Express.
- You can create three types of tablespaces:
 - Permanent tablespace: Contains persistent schema objects. Objects in permanent tablespaces are stored in data files.
 - Undo tablespace: Is a type of permanent tablespace used by Oracle Database to manage undo data if you are running your database in automatic undo management mode. Oracle strongly recommends that you use automatic undo management mode rather than using rollback segments for undo.
 - Temporary tablespace: Contains schema objects only for the duration of a session. Objects in temporary tablespaces are stored in temp files.

Creating Permanent Tablespaces in a CDB

- Tablespace creation during CDB creation:
 - With DBCA: USERS tablespace created in the CDB root
 - With CREATE DATABASE statement with USER_DATA TABLESPACE clause: Your defined tablespace created in the CDB root
- Create a permanent tablespace in the CDB root:

```
SQL> CONNECT system@cdb1
SQL> CREATE TABLESPACE tbs_CDB_users
        DATAFILE '/u1/app/oracle/oradata/cdb/cdb_users01.dbf' SIZE 100M;
```

- Create a permanent tablespace in a PDB:

```
SQL> CONNECT system@PDB1
SQL> CREATE TABLESPACE tbs_PDB1_users
        DATAFILE '/u1/app/oracle/oradata/cdb/pdb1/users01.dbf' SIZE 100M;
```

Altering and Dropping Tablespaces

- When you create a tablespace, it is initially a read/write tablespace.
- Use the `ALTER TABLESPACE` statement to take a tablespace offline or online, add data files or temp files to it, or make it a read-only tablespace.
- A tablespace can be in one of three different statuses or states:
 - Read Write
 - Read Only
 - Offline with one of the following options:
 - `NORMAL`
 - `TEMPORARY`
 - `IMMEDIATE`
- Add space to an existing tablespace by either adding data files to the tablespace or changing the size of an existing data file.
- Use the `DROP TABLESPACE` statement to drop a tablespace and its contents from the database if you no longer need its content.

Viewing Tablespace Information

- Tablespace and data file information can be obtained by querying the following views:
 - Tablespace information:
 - CDB_TABLESPACES and DBA_TABLESPACES
 - V\$TABLESPACE
 - Data file information:
 - CDB_DATA_FILES and DBA_DATA_FILES
 - V\$DATAFILE
 - Temp file information:
 - CDB_TEMP_FILES and DBA_TEMP_FILES
 - V\$TEMPFILE
 - Tables in a tablespace:
 - ALL_TABLES

Review: Implementing Oracle Managed Files (OMF)

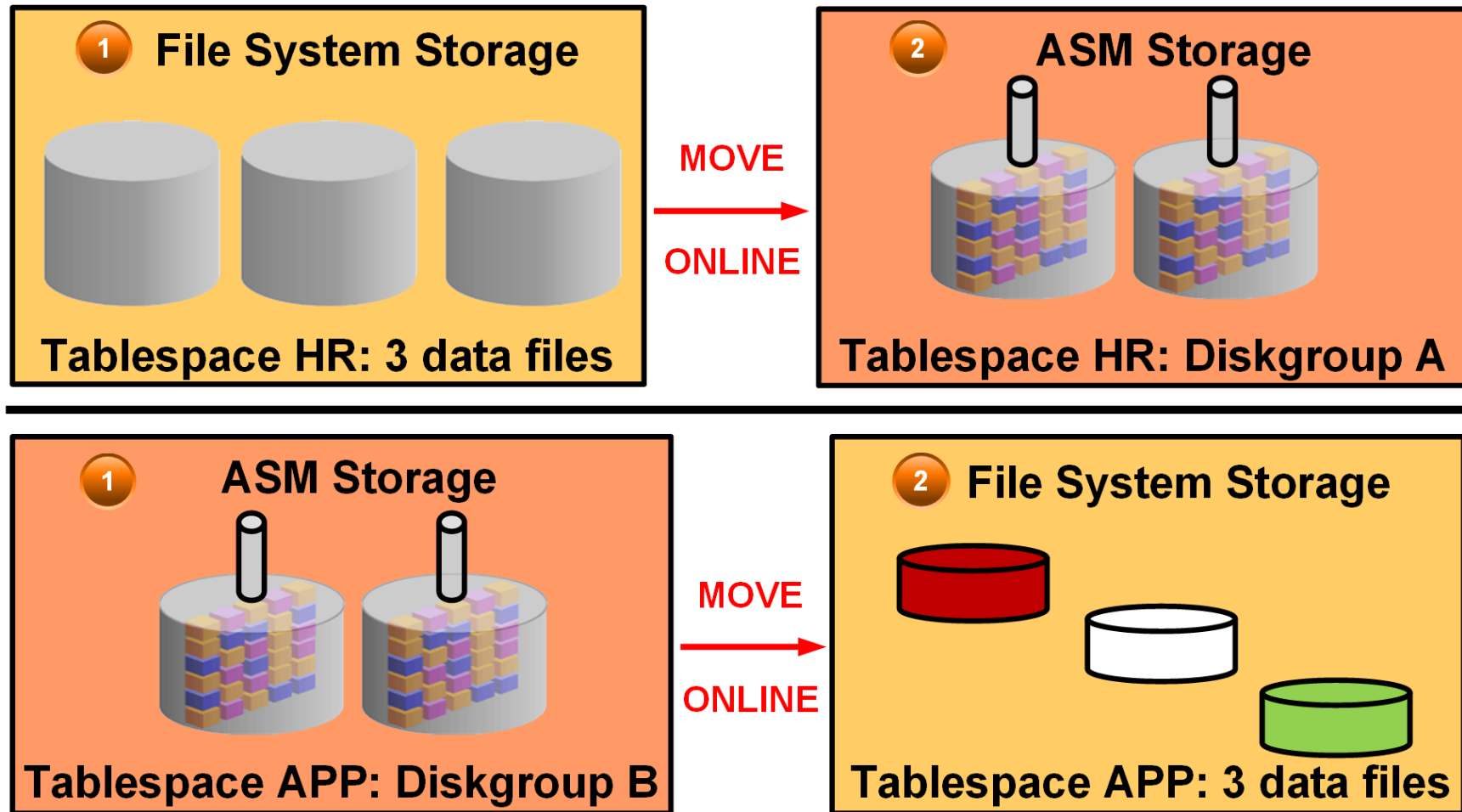
- Specify file operations in terms of database objects rather than file names.

Parameter	Description
DB_CREATE_FILE_DEST	Defines the location of the default file system directory for data files and temporary files
DB_CREATE_ONLINE_LOG_DEST_n	Defines the location for redo log files and control file creation
DB_RECOVERY_FILE_DEST	Gives the default location for the fast recovery area

- Example:

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST='/u01/app/oracle/oradata';  
SQL> CREATE TABLESPACE tbs_1;
```

Moving or Renaming Online Data Files



Examples: Moving and Renaming Online Data Files

- Relocating an online data file:

```
SQL> ALTER DATABASE MOVE DATAFILE '/disk1/myexample01.dbf'  
2 TO '/disk2/myexample01.dbf';
```

- Copying a data file from a file system to Automatic Storage Management (ASM):

```
SQL> ALTER DATABASE MOVE DATAFILE '/disk1/myexample01.dbf'  
2 TO '+DiskGroup2' KEEP;
```

- Renaming an online data file:

```
SQL> ALTER DATABASE MOVE DATAFILE '/disk1/myexample01.dbf'  
2 TO '/disk1/myexample02.dbf';
```

Tablespace Encryption by Default in DBCS



- In Oracle Database Cloud Service, user-created tablespaces are encrypted by default.
- Tablespaces created when the database is first created (in the root container, PDB seed, and `PDB1`) are NOT encrypted.
- The default encryption algorithm is AES128.
- The underlying architecture supporting this feature is Transparent Data Encryption (TDE).

Controlling Tablespace Encryption by Default

Parameter Value	Description
ALWAYS	Any tablespace created will be transparently encrypted with the AES128 algorithm unless a different algorithm is specified in the ENCRYPTION clause.
CLOUD_ONLY (Default value)	Tablespaces created in a Database Cloud Service database will be transparently encrypted with the AES128 algorithm unless a different algorithm is specified in the ENCRYPTION clause. For non-Database Cloud Service databases, tablespaces will only be encrypted if the ENCRYPTION clause is specified.
DDL	Tablespaces are not transparently encrypted and are only encrypted if the ENCRYPTION clause is specified.

Managing the Software Keystore and Master Encryption Key



- When the Database Cloud Service instance is created, a local auto-login software keystore is also created.
- The keystore is local to the compute node and is protected by a system-generated password.
- The auto-login software keystore is automatically opened when accessed.
- Use the `dbaascli` utility to change (rotate) the master encryption key.

```
DBAAS> tde rotate masterkey
Executing command tde rotate masterkey
Enter keystore password:
Successfully rotated TDE masterkey
```

Creating an Encrypted Tablespace by Using a Nondefault Algorithm

- Use the `ENCRYPTION USING` clause to specify the algorithm:

```
SQL> CREATE TABLESPACE encrypt_ts  
      DATAFILE '$ORACLE_HOME/dbs/encrypt.dat' SIZE 100M  
      ENCRYPTION USING '3DES168'  
      DEFAULT STORAGE (ENCRYPT) ;
```

- Restrictions:
 - Temporary and undo tablespaces cannot be encrypted.
 - `BFILE` data type and external tables are not encrypted.
 - The key for an encrypted tablespace cannot be changed.
 - The `SYSTEM` tablespace cannot be encrypted.

Summary

- In this lesson, you should have learned how to:
 - Explain how table data is stored in the database
 - Use SQL*Plus to:
 - Create and drop tablespaces
 - Alter tablespaces
 - View tablespace information
 - Implement Oracle Managed Files (OMF)
 - Use SQL*Plus to move and rename online data files
 - Implement tablespace encryption



Practice 12: Overview

- 12-1: Viewing Tablespace Information
- 12-2: Creating a Tablespace
- 12-3: Creating a Tablespace that is Encrypted by Default