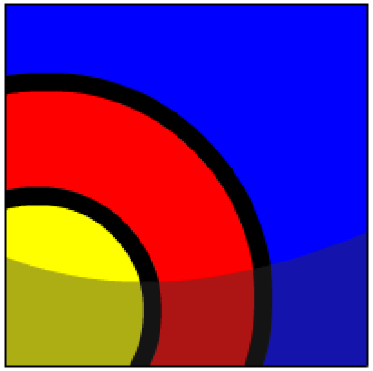
Creating Tables

Copyright © 2009, Oracle. All rights reserved.

What Will I Learn?

In this lesson, you will learn to:

- List and provide an example of each of the number, character, and date data types
- Create a table using the appropriate data type for each column
- Explain the use of external tables
- Use the Data Dictionary to obtain the names and other attributes of database objects



Copyright © 2009, Oracle. All rights reserved.

Why Learn It?

Up until now, you have selected, added, and deleted information in tables in an existing database. If you have a job as a Database Administrator (DBA), however, you will be expected to know how to create tables as well.

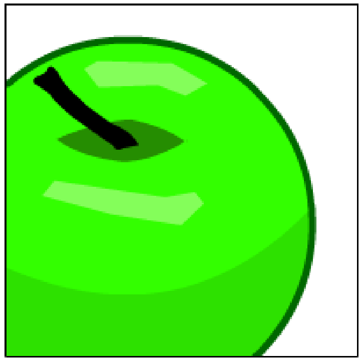
In this lesson, you will learn how to create new tables. Your tables will be small compared to tables that hold millions of rows and hundreds of columns, but creating a small table involves the same SQL statements and syntax as creating a very large one.

You will also learn about external tables, which are similar in structure to the normal Oracle database tables, but the actual data rows are not held inside the database. Instead they are held in a flat file stored externally to the database, and are accessed only when required.

Copyright © 2009, Oracle. All rights reserved.

Tell Me / Show Me

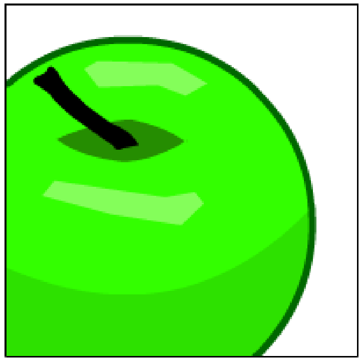
All data in a relational database is stored in tables. When creating a new table, use the following rules for table names and column names:



- Must begin with a letter
- Must be 1 to 30 characters long
- Must contain only A - Z, a - z, 0 - 9, _ (underscore), $, and #
- Must not duplicate the name of another object owned by the same user
- Must not be an Oracle Server reserved word

Copyright © 2009, Oracle. All rights reserved.

Tell Me / Show Me



It is best to use descriptive names for tables and other database objects. If a table will store information about students, name it STUDENTS, not PEOPLE or CHILDREN.

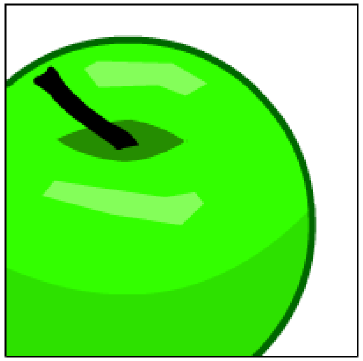
Also, names are not case sensitive. For example, STUDENTS is treated the same as STuDents or students.

Creating tables is part of SQL's data definition language (DDL). Other DDL statements used to set up, change, and remove data structures from tables include ALTER, DROP, RENAME, and TRUNCATE.

Copyright © 2009, Oracle. All rights reserved.

Tell Me / Show Me

CREATE TABLE



To create a new table, you must have the CREATE TABLE privilege and a storage area for it. The database administrator uses data control language (DCL) statements to grant this privilege to users and assign a storage area.

Tables belonging to other users are not in your schema. If you want to use a table that is not in your schema, use the table owner's name as a prefix to the table name:

```
SELECT * FROM mary.students;
```

Copyright © 2009, Oracle. All rights reserved.

Tell Me / Show Me

To create a new table consider the following syntax details:

- **table** is the name of the table
- **column** is the name of the column
- **datatype** is the column's data type and length
- **DEFAULT expression** specifies a default value if a value is omitted in the INSERT statement

```
CREATE TABLE table
(column datatype [DEFAULT expression],
(column datatype [DEFAULT expression],
(……[ ] );
```

For example:

```
CREATE TABLE cd_collection
(cd_number      NUMBER(2),
title           VARCHAR2(14),
artist          VARCHAR2(13),
purchase_date  DATE   DEFAULT SYSDATE);
```

Copyright © 2009, Oracle. All rights reserved.

Tell Me / Show Me

Creating A Table Using A Subquery

A second method for creating a table is to apply the AS subquery clause, which both creates the table and inserts rows returned from the subquery.

This is an easy way to create a copy of a table to practice SQL statements. Note that you need to create a column alias for those columns in the subquery that contain expressions.

Only datatype definitions and NOT NULL constraints are passed on to a new table created from a subquery. This is because the new table could be used in a different context, in which existing PK-FK relationships may not be relevant.

```
CREATE TABLE tablename
[(column, column, …)]
AS subquery;
```

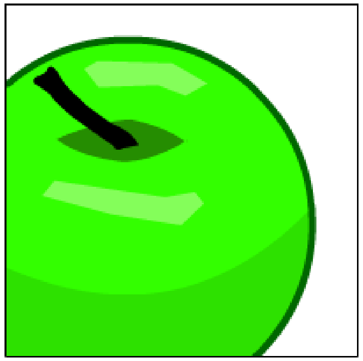
Two examples:

```
CREATE TABLE copy_mytable
AS
(SELECT code, name, start_date,
             end_date, give_away
FROM  f_promotional_menus);
```

```
CREATE TABLE dept80
AS
SELECT employee_id, last_name,
            salary*12 ANNSAL,
            hire_date
FROM  employees
WHERE department_id = 80;
```

Tell Me / Show Me

Creating A Table Using A Subquery



When a copy of a table is made using a subquery, the following rules are important:

- The column names in the new table will be identical to those in the original table, unless column aliases are used
- The column datatypes in the new table will be identical to those in the original table
- Only Not Null constraints are copied to the table, no other constraint types will exist on the new table.

Copyright © 2009, Oracle. All rights reserved.

Tell Me / Show Me

External Tables

Oracle also supports another table type: External table.

In an external table the data rows are not held inside the database files, but are instead found in a flat file, stored externally to the database. There are several advantages in using external tables.

Typically an external table would be used to store data migrated from older versions of the databases used by a company.

When a company is implementing a new application and database, they will typically need to import most of the data from the old systems to the new system for normal read and write access, but perhaps there is some data which is not frequently used and would mainly be required for read access.

That kind of data could be held in an external table. One of the many benefits for Oracle is that data held in external tables only has to be backed up once, and then never again unless the contents of the file change.

Copyright © 2009, Oracle. All rights reserved.

Tell Me / Show Me

External Tables

The syntax to create an external table is very similar to that of creating a standard table, except that it has extra syntax at the end.

An example can be found on the next slide.

Please note the new syntax, not used in standard SQL statements for table creation.

The new syntax is found in red on the next slide.

ORGANIZATION EXTERNAL -- tells Oracle to create an external table

TYPE ORACLE_LOADER -- of type Oracle Loader (an Oracle Product)

DEFAULT DIRECTORY def_dir1 -- what is the name of the directory where the file exists
ACCESS PARAMETERS -- how to read the file

RECORDS DELIMITED BY NEWLINE -- how to identify a new row starts

FIELDS – the start of the external file field name and datatype specification

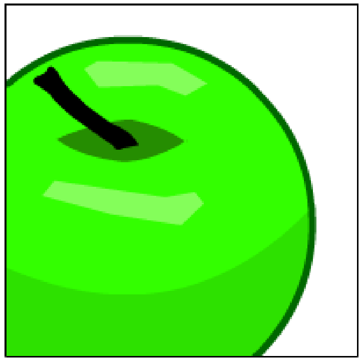
LOCATION – name of the actual file containing the data

Copyright © 2009, Oracle. All rights reserved.

ORACLE Academy

Tell Me / Show Me

External Tables



```
CREATE TABLE emp_load
  (employee_number CHAR(5),
   employee_dob CHAR(20),
   employee_last_name CHAR(20),
   employee_first_name CHAR(15),
   employee_middle_name CHAR(15),
   employee_hire_date DATE)
ORGANIZATION EXTERNAL
   (TYPE ORACLE_LOADER
    DEFAULT DIRECTORY def_dir1
    ACCESS PARAMETERS
   (RECORDS DELIMITED BY NEWLINE
    FIELDS (employee_number CHAR(2),
                employee_dob CHAR(20),
                employee_last_name CHAR(18),
                employee_first_name CHAR(11),
                employee_middle_name CHAR(11),
                employee_hire_date CHAR(10) date_format DATE mask "mm/dd/yyyy“))
    LOCATION ('info.dat'));
```

Copyright © 2009, Oracle. All rights reserved.

Tell Me / Show Me

Data Dictionary

What if you forget the name of one of your tables?

Don’t worry! Every Oracle database contains a Data Dictionary, which is a “master catalog” of all the tables and other objects in the database.

The Data Dictionary itself is a set of tables, because all data in the database is stored in tables. Think of the Data Dictionary as “tables that describe tables.”

We can SELECT from the Dictionary to see the names and other attributes of our tables:

```
SELECT table_name FROM user_tables;
```

Copyright © 2009, Oracle. All rights reserved.

Tell Me / Show Me

Data Dictionary

The tables and views in the data dictionary contain information about:

- Users and their privileges
- Tables, columns and their data types, integrity constraints, indexes
- Privileges granted on database objects
- Storage structures of the database
- In fact, everything in the database.

The data stored in the data dictionary is also often called "metadata." It consists of two levels: internal and external. The internal level contains tables that are used by the various DBMS software components. These are normally not visible to end users. The external level provides many views on these base tables to access information about objects and structures at different levels of detail.

Copyright © 2009, Oracle. All rights reserved.

Tell Me / Show Me

Data Dictionary

There are hundreds of views in the Dictionary. For us, the most important ones are the USER_* views, which describe the objects in your schema.

There are two other types of dictionary views:

- ALL_* views describe objects which you have been given privileges to use, for example tables in other users' schemas
- DBA_* views describe everything in the database, and can be used only by the DBA.

NAME	COMMENTS
USER_TABLES	Description of the user's own relational tables
USER_TAB_COLUMNS	Columns of user's tables, views and clusters
ALL_TABLES	Description of relational tables accessible to the user
DBA_TABLES	Description of all relational tables in the database
DBA_USERS	Information about all users of the database
….	….

Copyright © 2009, Oracle. All rights reserved.

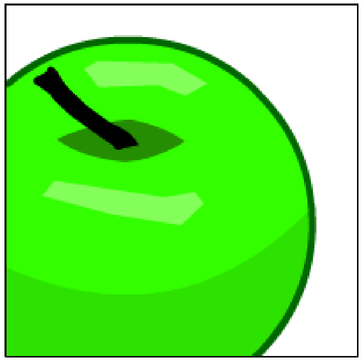
Tell Me / Show Me

You can query the data dictionary to view various database objects owned by you. To see all the views available in the data dictionary, use this SQL command:

```
SELECT *
FROM DICTIONARY;
```

The data dictionary tables frequently used are:

- USER_TABLES
- USER_OBJECTS
- USER_CATALOG or USER_CAT



Tell Me / Show Me

To access these tables, use the following SQL commands.

To see the names of tables owned by the user (you):

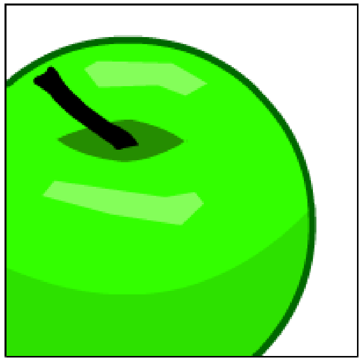
```
SELECT table_name
FROM user_tables;
```

To view distinct object types owned by the user:

```
SELECT DISTINCT object_type
FROM  user_objects;
```

To view all objects owned by the user:

```
SELECT *
FROM user_catalog;
```



Copyright © 2009, Oracle. All rights reserved.

Tell Me / Show Me

Terminology

Key terms used in this lesson include:

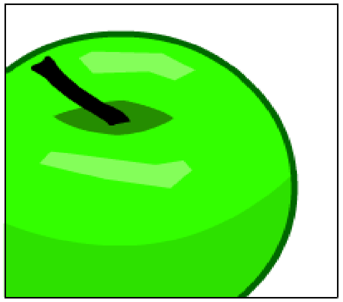
CREATE TABLE

Data dictionary

Table

Schema

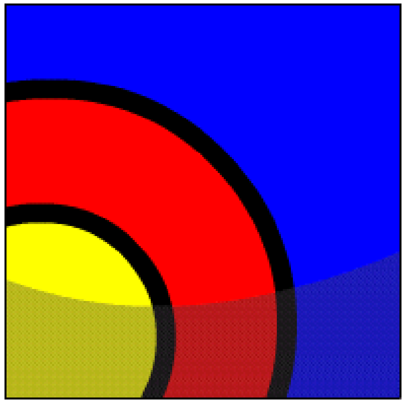
DEFAULT



Summary

In this lesson you have learned to:

- List and provide an example of each of the number, character, and date data types
- Create a table using the appropriate data type for each column
- Explain the use of external tables
- Use the Data Dictionary to obtain the names and other attributes of database objects

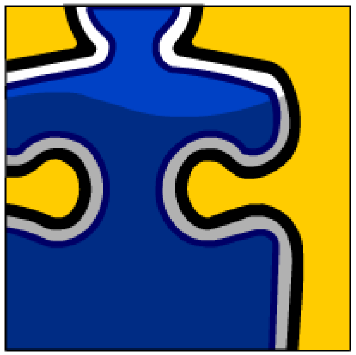


Copyright © 2009, Oracle. All rights reserved.

Summary

Practice Guide

The link for the lesson practice guide can be found in the course resources in Section 0.



Copyright © 2009, Oracle. All rights reserved.