 



Database Programming with SQL 13-1: Creating Tables

Practice Activities

# Objectives

* List and categorize the main database objects
* Review a table structure
* Describe how database schema objects are used by the Oracle database

# Vocabulary

Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **Data dictionary** | Created and maintained by the Oracle Server and contains information about the database |
| **Schema** | A collection of objects that are the logical structures that directly refer to the data in the database |
| **DEFAULT** | Specifies a preset value if a value is omitted in the INSERT statement |
| **TABLE** | Stores data; basic unit of storage composed of rows and columns |
| **CREATE TABLE** | Command use to make a new table |

# Try It / Solve It

1. Complete the GRADUATE CANDIDATE table instance chart. Credits is a foreign-key column referencing the requirements table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | student\_id | last\_name | first\_name | credits | graduation\_date |
| Key Type | Primary key |  |  | Foreign key |  |
| Nulls/Unique | **NO/YES** | **NO/NO** | **NO/NO** | **NO/NO** | **YES/NO** |
| FK Column |  |  |  |  |  |
| Datatype | NUMBER | VARCHAR2 | VARCHAR2 | NUMBER | DATE |
| Length | 6 | 75 | 75 | 6 |  |

For credits and student\_id it could have been precision and scale rather mentioned here. I assume that when 6 is written for student\_id it means NUMBER(6,0) and for credits NUMBER(5, 2)

1. Write the syntax to create the grad\_candidates table.

**CREATE TABLE  graduate\_candidates**

**( student\_id NUMBER(6,0),**

**last\_name VARCHAR2(75) CONSTRAINT gcs\_last\_name\_nn NOT NULL ENABLE,**

**first\_name VARCHAR2(75) CONSTRAINT gcs\_first\_name\_nn NOT NULL ENABLE,**

**credits NUMBER(5,2) CONSTRAINT gcs\_credits\_nn NOT NULL ENABLE,**

**graduation\_date DATE,**

**CONSTRAINT gcs\_std\_id\_pk PRIMARY KEY (student\_id),**

**CONSTRAINT gcs\_req\_fk FOREIGN KEY(credits) REFERENCES requirements(credits) ENABLE**

**);**

Note: Since there is no requirements table already existing for foreign key above stamen will fail. I may create this table if choose not to include this constraint.

1. Confirm creation of the table using DESCRIBE.

**DESCRIBE graduate\_candidates;**

1. Create a new table using a subquery. Name the new table your last name -- e.g., smith\_table. Using a subquery, copy grad\_candidates into smith\_table.

Even if the foreign key world have been created in problem 2, it would have been lost here in copy.

**CREATE TABLE kumar\_table**

**AS ( SELECT \* FROM graduate\_candidates);**

see if copy worked well:

DESCRIBE kumar\_table;

student\_id is nullable and not a primary key here in kumar\_table.

1. Insert your personal data into the table created in question 4.

**INSERT INTO kumar\_table (student\_id, last\_name, first\_name, credits, graduation\_date)**

**Values(10,'kumar','he',999.99,NULL);**

SELECT LENGTH(credits) FROM kumar\_table WHERE student\_id = 10;

1. Query the data dictionary for each of the following:
   * USER\_TABLES
   * USER\_OBJECTS
   * USER\_CATALOG or USER\_CAT

In separate sentences, summarize what each query will return.

--USER\_TABLES describes the relational tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL\_TABLES.

**SELECT \* FROM user\_tables;**

48 rows returned in 0.64 seconds

--USER\_CATALOG lists indexes, tables, views, clusters, synonyms, and sequences owned by the current user. Its columns are the same as those in "ALL\_CATALOG".

**SELECT \* FROM user\_catalog;**

407 rows returned in 0.05 seconds

**SELECT DISTINCT(table\_type) FROM user\_catalog;**

3 rows returned in 0.02 seconds

--USER\_OBJECTS describes all objects owned by the current user. Its columns are the same as those in "ALL\_OBJECTS".

**SELECT \* FROM user\_objects;**

117 rows returned in 0.19 seconds

**SELECT DISTINCT(object\_type) FROM user\_objects;**

6 rows returned in 0.17 seconds

SELECT \* FROM user\_cat;

ORA-00942: table or view does not exist

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.