


Section 8 Lesson 1: Creating Tables

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				FK	
Nulls/Unique					
FK Column				1	
Datatype	NUMBER	VARCHAR2	VARCHAR2	NUMBER	DATE
Length	6			3	

2. Write the syntax to create the grad_candidates table.

Rows  Save Run

```
create table grad_candidates
(student_id number(6,0),
last_name varchar2(35),
first_name varchar2(35),
credits number(3,0),
graduation_date date);
```

Results Explain Describe Saved SQL History

Table created.

0.01 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

3. Confirm creation of the table using DESCRIBE.

Rows
1000
Save
Run

```
describe grad_candidates;
```

Results Explain Describe Saved SQL History

Object Type TABLEObject GRAD_CANDIDATES

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
GRAD_CANDIDATES	STUDENT_ID	NUMBER	-	6	0	-	✓	-	-
	LAST_NAME	VARCHAR2	35	-	-	-	✓	-	-
	FIRST_NAME	VARCHAR2	35	-	-	-	✓	-	-
	CREDITS	NUMBER	-	3	0	-	✓	-	-
	GRADUATION_DATE	DATE	7	-	-	-	✓	-	-
1 - 5									

Application Express 4.2.5.00.08

Workspace: US_1350 User: US_1350_SQL01_S25

Language: en | Copyright © 1999, 2014, Oracle. All rights reserved.

- 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 wall/s_table.

Rows1000

SaveRun

```
create table walls_table
as(select * from grad_candidates);
```

Results

Explain

Describe

Saved SQL

History

Table created.

0.02 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

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

The screenshot shows the Oracle SQL Developer interface. At the top, there is a toolbar with a 'Rows' dropdown set to '1000', a 'Save' button, and a 'Run' button. Below the toolbar is a text area containing the following SQL statement:

```
insert into walls_table  
values (696214, 'Walls', 'Omie', 119,  
to_date('05-22-2016', 'mm-dd-yyyy'));
```

Below the text area is a tabbed interface with the following tabs: 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing the following output:

```
1 row(s) inserted.  
  
0.01 seconds
```

At the bottom of the interface, the workspace information is displayed: 'Workspace: US_1350 User: US_1350_SQL01_S25'.

6. 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.

Querying `USER_TABLES` will return data from user-created tables. `USER_OBJECTS` returns data objects (which appear to be essentially entities). `USER_CATALOG` queries return the table name and type of each table in the oracle database, user and not user created.

Section 8 Lesson 2: Using Datatypes

Try It / Solve It

- Using the examples provided in Tell Me/Show Me, create each of the three time-zone tables.
 - TIMESTAMP WITH LOCAL TIME ZONE
 - INTERVAL YEAR TO MONTH
 - INTERVAL DAY TO SECOND
- Execute a `SELECT *` from each table to verify your input.



LONDON



NEW YORK



Rows: 1000 Save Run

```
select * from sec_8_les_2_table_1;
```

Results Explain Describe Saved SQL History

LOCAL_TZ_TIMESTAMP
15-NOV-07 08.00.00.000000 AM -08:00

1 rows returned in 0.02 seconds [Download](#)

Workspace: US_1350 User: US_1350_SQL01_S25

Rows: 1000 Save Run



```
select * from sec_8_les_2_table_2;
```

Results Explain Describe Saved SQL History

YTM_INTERVAL
+10-00

1 rows returned in 0.01 seconds [Download](#)

Workspace: US_1350 User: US_1350_SQL01_S25

Rows   Save Run

```
select * from sec_8_les_2_table_3;
```

Results Explain Describe Saved SQL History

DTS_INTERVAL
+25 00:00:00.000000

1 rows returned in 0.01 seconds [Download](#)

Workspace: US_1350 User: US_1350_SQL01_S25

3. Give 3 examples of organizations and personal situations where it is important to know to which time zone a date-time value refers.
1. To provide correct logistics information for a non-local business transaction.
 2. To provide accurate destination arrival on a scheduled flight.
 3. To know the time of a phone-call referential to the location in which it was placed or received.

Section 8 Lesson 3: Modifying a Table

Try It / Solve It

Before beginning the practice exercises, execute a DESCRIBE for each of the following tables: o_employees, o_departments and o_jobs. These tables will be used in the exercises. If they do not exist in your account, create them as follows:

- CREATE TABLE o_jobs AS (SELECT * FROM jobs);
- CREATE TABLE o_employees AS (SELECT * FROM employees);
- CREATE TABLE o_departments AS (SELECT * FROM departments);

You will need to know which columns do not allow null values.



1. Why is it important to be able to modify a table? In a company, factors change in day-to-day business. Without the ability to modify tables, the table's form might become outdated if it cannot be changed to fit a growing business. Having the ability to modify tables also helps when making mistakes and errors when creating or updating a table.
2. CREATE a table called Artists.
 - a. Add the following to the table:
 - artist ID
 - first name
 - last name
 - band name
 - email
 - hourly rate
 - song ID from d_songs table

Rows  

Save Run

```
alter table artists
add foreign key (song_id)
references d_songs(id);
```

Results Explain Describe Saved SQL History

Rows  

Save Run

```
create table artists (
artist_id number(5,0),
first_name varchar2(35),
last_name varchar2(35),
band_name varchar2(50),
email varchar2(35),
hourly_rate number(6,2),
song_id number(3,0));
```


Results Explain Describe Saved SQL History

Table created.

0.02 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

b. INSERT one artist from the d_songs table.

Rows 

Save

Run

```
insert into artists (song_id, band_name)
select id, artist
from d_songs
where type_code = 1;
```

Results

Explain

Describe

Saved SQL

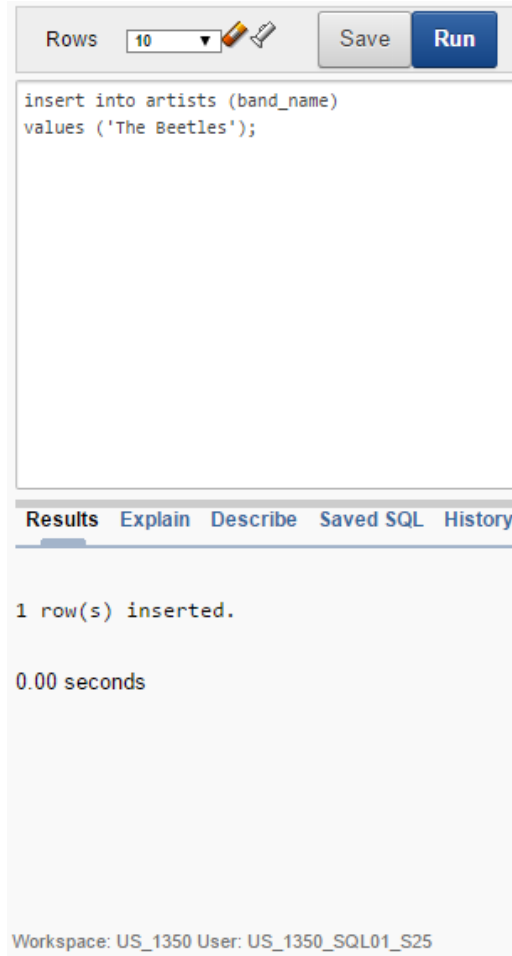
History

1 row(s) inserted.

0.01 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

c. INSERT one artist of your own choosing; leave song_id blank.



The screenshot shows a SQL IDE interface. At the top, there is a toolbar with a 'Rows' dropdown set to '10', a 'Save' button, and a 'Run' button. Below the toolbar is a text area containing the SQL statement: `insert into artists (band_name)`
`values ('The Beatles');`. Below the text area is a tabbed interface with tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing the output: `1 row(s) inserted.` and `0.00 seconds`. At the bottom of the window, the workspace information is displayed: `Workspace: US_1350 User: US_1350_SQL01_S25`.

d. Give an example how each of the following may be used on the table that you have created:

- ALTER TABLE
`ALTER TABLE Artists`
`MODIFY band_name VARCHAR2(50);`
- DROP TABLE
`DROP TABLE Artists`
- RENAME TABLE
`ALTER TABLE`
`RENAME TABLE Artists TO Artist`
- TRUNCATE
`TRUNCATE TABLE Artists`
- COMMENT ON TABLE
`COMMENT ON TABLE artists IS 'This is a table for Artists.';`

3. In your o_employees table, enter a new column called "Termination." The datatype for the new column should be VARCHAR2. Set the DEFAULT for this column as SYSDATE to appear as character data in the format: February 20th, 2003.

Rows

Save Run

```
alter table o_employees
add Termination varchar2(25) default
to_char(sysdate, 'Month ddth, yyyy');
```

Results Explain Describe Saved SQL History

Table altered.

0.03 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

4. Create a new column in the o_employees table called start_date. Use the TIMESTAMP WITH LOCAL TIME ZONE as the datatype.

Rows

Save Run

```
alter table o_employees
add start_date timestamp with local time zone;
```



Results Explain Describe Saved SQL History

Table altered.

0.04 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

5. Truncate the o_job_description table. Then do a SELECT * statement. Are the columns still there? Is the data still there? **There is no data found.**

Rows   Save Run

truncate table o_job_description;

Results Explain Describe Saved SQL History

Table truncated.

0.06 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

Rows   Save Run

select * from o_job_description;


Results Explain Describe Saved SQL History

no data found

Workspace: US_1350 User: US_1350_SQL01_S25

6. What is the distinction between TRUNCATE, DELETE, and DROP for tables? **TRUNCATE removes all rows of data from a table. DELETE removes all rows and data structure. DROP removes the table from the database.**
7. List the changes that can and cannot be made to a column. **You can add, modify or drop a column in a table, but you cannot specify where the column appears.**

8. Add the following comment to the o_jobs table:
"New job description added"
View the data dictionary to view your comments.

Rows  Save Run


```
comment on table o_jobs  
is 'New job description added';
```

Results Explain Describe Saved SQL History

Statement processed.

0.01 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

Rows  Save Run

```
select table_name, comments  
from user_tab_comments  
where table_name = upper('o_jobs');
```

Results Explain Describe Saved SQL History

TABLE_NAME	COMMENTS
O_JOBS	New job description added

1 rows returned in 0.01 seconds [Download](#)

Workspace: US_1350 User: US_1350_SQL01_S25

9. Rename the o_jobs table to o_job_description.

Rows

Save

Run

```
alter table
o_jobs rename
to o_job_description;
```

Results

Explain

Describe

Saved SQL

History

Statement processed.

0.03 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

10.F_staffs table exercises:

- a. Create a copy of the f_staffs table called copy_f_staffs and use this copy table for the remaining labs in this lesson.

Rows

Save

Run

```
create table copy_f_staffs
as (select * from f_staffs);
```

Results

Explain

Describe

Saved SQL


History

Table created.

0.08 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

b. Describe the new table to make sure it exists.

Rows500

SaveRun

describe copy_f_staffs;

ResultsExplainDescribeSaved SQLHistory

Object Type TABLEObject COPY_F_STAFFS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COPY_F_STAFFS	ID	NUMBER	-	5	0	-	✓	-	-
	FIRST_NAME	VARCHAR2	25	-	-	-	-	-	-
	LAST_NAME	VARCHAR2	35	-	-	-	-	-	-
	BIRTHDATE	DATE	7	-	-	-	-	-	-
	SALARY	NUMBER	-	8	2	-	-	-	-
	OVERTIME_RATE	NUMBER	-	5	2	-	✓	-	-
	TRAINING	VARCHAR2	50	-	-	-	✓	-	-
	STAFF_TYPE	VARCHAR2	20	-	-	-	-	-	-
	MANAGER_ID	NUMBER	-	5	0	-	✓	-	-
	MANAGER_BUDGET	NUMBER	-	8	2	-	✓	-	-
	MANAGER_TARGET	NUMBER	-	8	2	-	✓	-	-
1 - 11									

Application Express 4.2.5.00.08


Workspace: US_1350 User: US_1350_SQL01_S25

Language: en | Copyright © 1999, 2014, Oracle. All rights reserved.

c. Drop the table.

Rows

500



Save

Run

drop table copy_f_staffs;

Results

Explain

Describe

Saved SQL


History

Table dropped.

0.06 seconds

Workspace: US_1350 User: US_1350_SQL01_S25


d. Try to select from the table.

Rows 

Save Run

```
select * from copy_f_staffs;
```

Results Explain Describe Saved SQL History

 **ORA-00942: table or view does not exist**

Workspace: US_1350 User: US_1350_SQL01_S25

- e. Investigate your recyclebin to see where the table went.

Rows Save Run

```
select object_name, original_name
from recyclebin
where original_name = upper('copy_f_staffs');
```

Results Explain Describe Saved SQL History

OBJECT_NAME	ORIGINAL_NAME
BIN\$JqbBjxsWT2DgU8xjFJD+Pg==\$0	COPY_F_STAFFS
BIN\$JqbZGMdzU9TgU8xjFJCX0Q==\$0	COPY_F_STAFFS

2 rows returned in 0.90 seconds [Download](#)

Workspace: US_1350 User: US_1350_SQL01_S25

- f. Try to select from the dropped table by using the value stored in the OBJECT_NAME column. You will need to copy and paste the name as it is exactly, and enclose the new name in “ ” (double quotes). So if the dropped name returned to you is BIN\$Q+x1nJdcUnngQESYELVldQ==\$0, you need to write a query that refers to “BIN\$Q+x1nJdcUnngQESYELVldQ==\$0”.

Rows Save Run

```
select *
from recyclebin
where object_name = 'BIN$JqbZGMdzU9TgU8xjFJCX0Q==$0'
```

Results Explain Describe Saved SQL History

OBJECT_NAME	ORIGINAL_NAME	OPERATION	TYPE	TS_NAME	CREATETIME	DROPTIME	DROPSCN	PARTITION_NAME	CAN_UNDROP	CAN_PURGE	RELATED
BIN\$JqbZGMdzU9TgU8xjFJCX0Q==\$0	COPY_F_STAFFS	DROP	TABLE	FLOW_17000002145309107	2015-12-11:13:11:06	2015-12-11:13:17:17	829476345	-	YES	YES	6494417

1 rows returned in 6.52 seconds [Download](#)

Application Express 4.2.5.00.08

Workspace: US_1350 User: US_1350_SQL01_S25

Language: en | Copyright © 1999, 2014, Oracle. All rights reserved.

g. Undrop the table.

Rows

Save Run

flashback table copy_f_staffs to before drop;

Results Explain Describe Saved SQL History

Statement processed.

0.08 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

h. Describe the table.

Rows
500
Save
Run

```
describe copy_f_staffs;
```

Results
Explain
Describe
Saved SQL
History

Object Type TABLEObject COPY_F_STAFFS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COPY_F_STAFFS	ID	NUMBER	-	5	0	-	✓	-	-
	FIRST_NAME	VARCHAR2	25	-	-	-	-	-	-
	LAST_NAME	VARCHAR2	35	-	-	-	-	-	-
	BIRTHDATE	DATE	7	-	-	-	-	-	-
	SALARY	NUMBER	-	8	2	-	-	-	-
	OVERTIME_RATE	NUMBER	-	5	2	-	✓	-	-
	TRAINING	VARCHAR2	50	-	-	-	✓	-	-
	STAFF_TYPE	VARCHAR2	20	-	-	-	-	-	-
	MANAGER_ID	NUMBER	-	5	0	-	✓	-	-
	MANAGER_BUDGET	NUMBER	-	8	2	-	✓	-	-
	MANAGER_TARGET	NUMBER	-	8	2	-	✓	-	-
									1 - 11

Workspace: US_1350 User: US_1350_SQL01_S25

11. Still working with the copy_f_staffs table, perform an update on the table.

- Issue a select statement to see all rows and all columns from the copy_f_staffs table;

Rows
500
Save
Run

```
select * from copy_f_staffs;
```

Results
Explain
Describe
Saved SQL
History

ID	FIRST_NAME	LAST_NAME	BIRTHDATE	SALARY	OVERTIME_RATE	TRAINING	STAFF_TYPE	MANAGER_ID	MANAGER_BUDGET	MANAGER_TARGET
12	Sue	Doe	01/Jul/1980	10	11.95	-	Order Taker	19	-	-
9	Bob	Miller	19/Mar/1979	10	.75	Grill	Cook	19	-	-
19	Monique	Tuttle	30/Mar/1969	60	-	-	Manager	-	50000	70000

3 rows returned in 0.01 seconds [Download](#)


Workspace: US_1350 User: US_1350_SQL01_S25

Application Express 4.2.5.00.08
Language: en | Copyright © 1999, 2014, Oracle. All rights reserved.

b. Change the salary for Sue Doe to 12 and commit the change.

Rows

500



Save

Run

```
update copy_f_staffs
set salary = 12
where first_name = initcap('sue')
and last_name = initcap('doe');
```

Results

Explain

Describe

Saved SQL



History

1 row(s) updated.

0.01 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

- c. Issue a select statement to see all rows and all columns from the copy_f_staffs table;

Rows: 500   Save Run

```
select * from copy_f_staffs;
```

Results Explain Describe Saved SQL History



ID	FIRST_NAME	LAST_NAME	BIRTHDATE	SALARY	OVERTIME_RATE	TRAINING	STAFF_TYPE	MANAGER_ID	MANAGER_BUDGET	MANAGER_TARGET
12	Sue	Doe	01/Jul/1980	12	11.95	-	Order Taker	19	-	-
9	Bob	Miller	19/Mar/1979	10	.75	Grill	Cook	19	-	-
19	Monique	Tuttle	30/Mar/1969	60	-	-	Manager	-	50000	70000

3 rows returned in 0.00 seconds [Download](#)

Workspace: US_1350 User: US_1350_SQL01_S25

Application Express 4.2.5.00.08
Language: en | Copyright © 1999, 2014, Oracle. All rights reserved.

- d. For Sue Doe, update the salary to 2 and commit the change.

Rows: 500   Save Run

```
update copy_f_staffs  
set salary = 2  
where first_name = initcap('sue')  
and last_name = initcap('doe');
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.01 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

- e. Issue a select statement to see all rows and all columns from the copy_f_staffs table;

Rows Save Run

select * from copy_f_staffs;

Results Explain Describe Saved SQL History

ID	FIRST_NAME	LAST_NAME	BIRTHDATE	SALARY	OVERTIME_RATE	TRAINING	STAFF_TYPE	MANAGER_ID	MANAGER_BUDGET	MANAGER_TARGET
12	Sue	Doe	01/Jul/1980	2	11.95	-	Order Taker	19	-	-
9	Bob	Miller	19/Mar/1979	10	.75	Grill	Cook	19	-	-
19	Monique	Tuttle	30/Mar/1969	60	-	-	Manager	-	50000	70000

3 rows returned in 0.00 seconds [Download](#)

Application Express 4.2.5.00.08

Workspace: US_1350 User: US_1350_SQL01_S25 Language: en | Copyright © 1999, 2014, Oracle. All rights reserved.

- f. Now, issue a FLASHBACK QUERY statement against the copy_f_staffs table, so you can see all the changes made.

Rows Save Run

select first_name,
last_name,
versions_operation,
versions_starttime,
versions_endtime,
salary
from copy_f_staffs
versions between scn minvalue and maxvalue
where first_name = initcap('sue')
and last_name = initcap('doe');

Results Explain Describe Saved SQL History


FIRST_NAME	LAST_NAME	VERSIONS_OPERATION	VERSIONS_STARTTIME	VERSIONS_ENDTIME	SALARY
Sue	Doe	U	11-DEC-15 02.26.40 PM	11-DEC-15 02.42.04 PM	2
Sue	Doe	U	11-DEC-15 02.23.22 PM	11-DEC-15 02.26.40 PM	12
Sue	Doe	-	-	11-DEC-15 02.23.22 PM	10

3 rows returned in 0.01 seconds [Download](#)

Application Express 4.2.5.00.08

Workspace: US_1350 User: US_1350_SQL01_S25 Language: en | Copyright © 1999, 2014, Oracle. All rights reserved.

- g. Investigate the result of f), and find the original salary and update the copy_f_staffs table salary column for Sue Doe back to her original salary.

Rows 

Save Run

```
update copy_f_staffs
set salary =

(select salary
from copy_f_staffs
versions between scn minvalue and maxvalue
where versions_starttime is null
and first_name = initcap('sue')
and last_name = initcap('doe'))

where first_name = initcap('sue')
and last_name = initcap('doe');
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.01 seconds

Workspace: US_1350 User: US_1350_SQL01_S25