



## **Oracle Application Express: Developing Database Web Applications**

### **Hands-On-Labs Guide**

#### Unit 2: Using SQL Workshop



This exercise includes two hands-on-labs.

**HOL 2-1: Loading the Tables and Data:** In this lab, you use SQL Workshop to create the underlying database objects and data required for you to build the Demo Projects application.

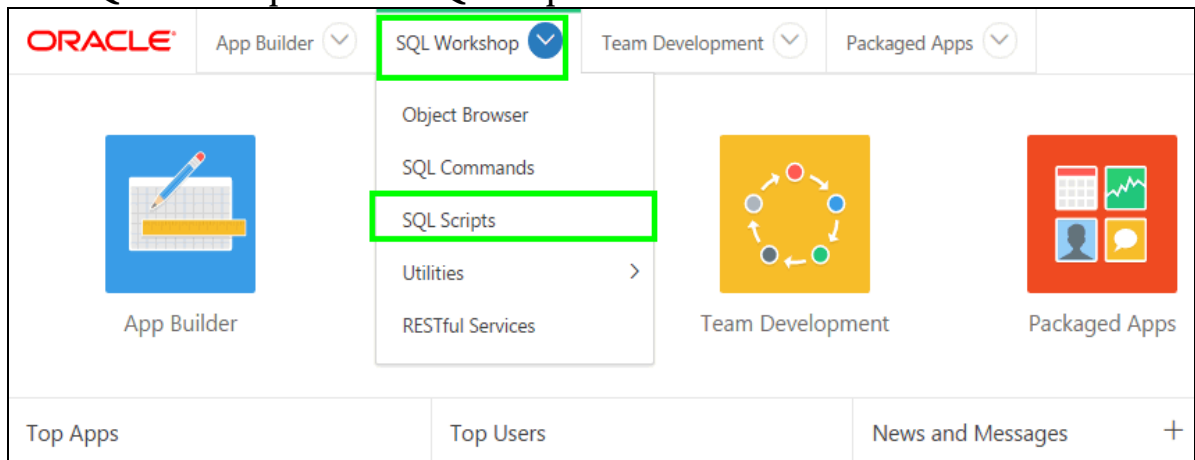
**HOL 2-2: Creating a Lookup Table:** In this lab, you create a table named `HARDWARE` and load data in to the table. Then, you create a lookup table.



## HOL 2-1: Loading the Tables and Data

It is essential to have at least the tables defined in order for the Create Application wizard to generate pages in your application. In this hands-on-lab, you create the required database objects, and populate the tables with sample data.

1. Use SQL Workshop to upload a script that creates the tables for the Demo Projects application. Perform the following steps:
  - a) Click **SQL Workshop** and select **SQL Scripts**.



- b) Click **Upload**.
- c) Click **Choose File**, navigate to the working directory where you extracted `apex-course-labs.zip`. Locate the **Project\_Basic\_Tables.sql** file, and double-click the file or click the file and then click **Open**.

d) Click **Upload**.

Upload Script

\* File  Project\_Tables.sql

Script Name

File Character Set

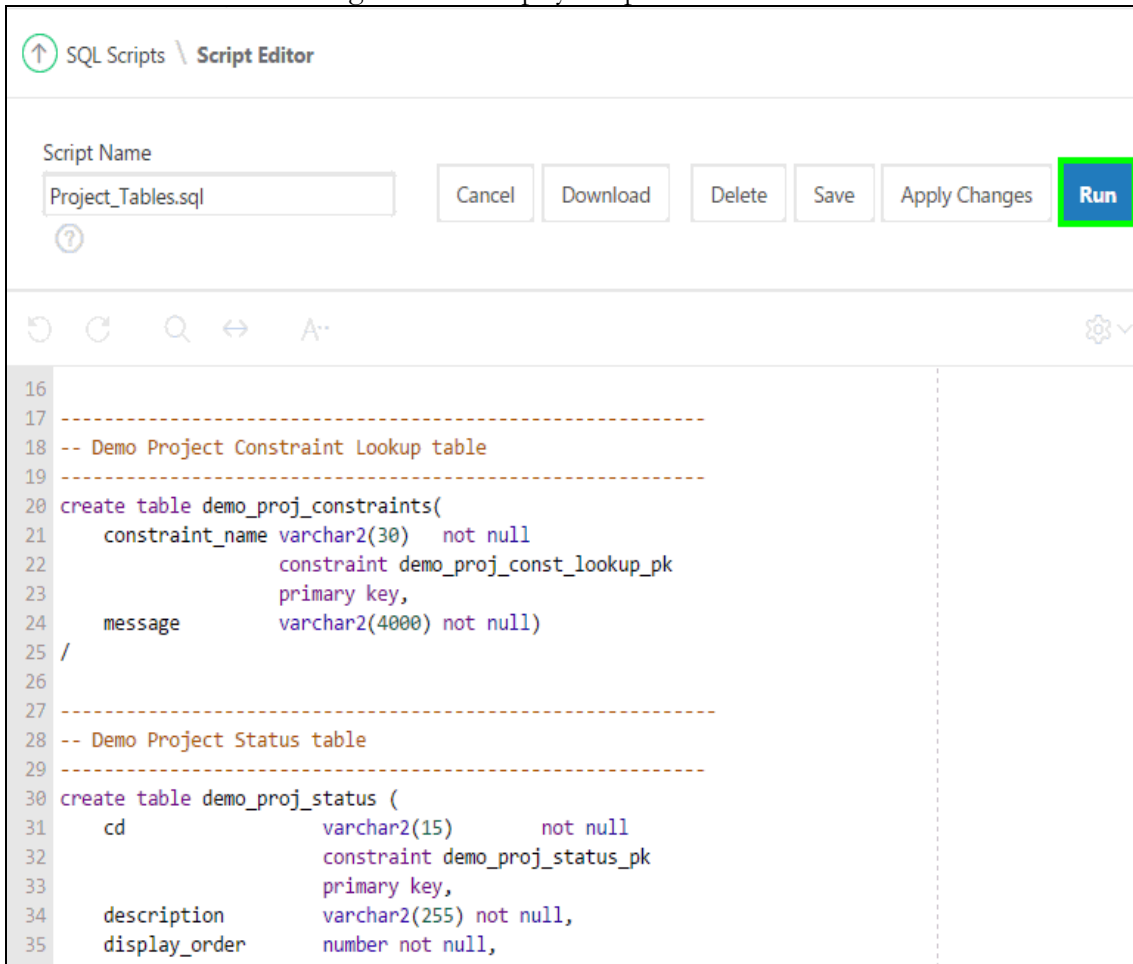
e) Review the uploaded script to see what tables will be created.  
In the SQL Scripts list, click the **Edit** icon (pencil), to the left of the script you just uploaded.

SQL Scripts

Go Actions Delete Checked Upload > Create >

<input type="checkbox"/>	Edit	Owner	Name	Created	Updated By	Updated	Bytes	Results	Run
<input type="checkbox"/>		CHAITANYA	Project_Tables.sql	Now	CHAITANYA	Now	15,164	0	

- f) Click the **Run** icon to the right of the script you uploaded.



SQL Scripts \ Script Editor

Script Name  
Project\_Tables.sql

Cancel Download Delete Save Apply Changes **Run**

16  
17  
18 -----  
19 -- Demo Project Constraint Lookup table  
20 -----  
21 create table demo\_proj\_constraints(  
22 constraint\_name varchar2(30) not null  
23 constraint demo\_proj\_const\_lookup\_pk  
24 primary key,  
25 message varchar2(4000) not null)  
26 /  
27  
28 -----  
29 -- Demo Project Status table  
30 -----  
31 create table demo\_proj\_status (  
32 cd varchar2(15) not null  
33 constraint demo\_proj\_status\_pk  
34 primary key,  
35 description varchar2(255) not null,  
36 display\_order number not null,

Click **Run Now**.

g) Click the **View Results** icon for the script you just ran.

SQL Scripts \ Manage Script Results

Search:  Go Actions Delete Checked

Script	Run By	Started	Finished	Elapsed	Status	Security Group Id	Statements	Bytes	View Results
Project_Tables.sql	CHAITANYA	1 seconds ago	02/07/2017	0.46	Completed	1841547862027812	45 of 45	0	

Number	Elapsed	Statement	Feedback	Rows
1	0.01	create table demo_proj_constraints( constraint_name varc	Table created.	0
2	0.01	create table demo_proj_status ( cd varc	Table created.	0
3	0.00	create or replace trigger biu_demo_proj_status before in	Trigger created.	0
4	0.02	create table demo_proj_team_members ( id	Table created.	0
5	0.01	alter table demo_proj_team_members add constraint demo_proj_	Table altered.	0
6	0.02	create or replace trigger biu_demo_proj_team_members bef	Trigger created.	0
7	0.02	create table demo_projects ( id number	Table created.	0
8	0.00	alter table demo_projects add constraint demo_projects_uk	Table altered.	0
9	0.01	alter table demo_projects add constraint demo_proj_team_memb	Table altered.	0
10	0.00	create index demo_proj_team_member_idx on demo_projects (pro	Index created.	0
11	0.02	alter table demo_projects add constraint demo_proj_status_fk	Table altered.	0
12	0.00	create index demo_proj_status_idx on demo_projects (status_c	Index created.	0
13	0.01	create or replace trigger biu_demo_projects before inser	Trigger created.	0
14	0.02	create table demo_proj_milestones ( id	Table created.	0
15	0.01	alter table demo_proj_milestones add constraint demo_proj_ms	Table altered.	0

Download row(s) 1 - 15 of 45 [Next](#)

45

Statements Processed

45

Successful

0

With Errors

2. Currently the tables you created do not have any data. A script has been provided that creates an Oracle database package which can be run at any time to insert or reset the data in the tables. Use SQL Workshop to upload a script that you can use to populate table data. Perform the following steps:
  - a) Click **SQL Scripts**. Click **Upload**.
  - b) Click **Choose File**, where you extracted apex-course-labs.zip.
  - c) Locate the **Project\_Data.sql** file, and double-click the file or click the file and then click **Open**.

d) Click **Upload**.

### Upload Script

\* File  Project\_Data.sql ?

Script Name  ?

File Character Set  ?

e) Click the **Run** icon to the right of the script you uploaded (top row).

Edit	Owner	Name	Created	Updated By	Updated	Bytes	Results	Run
<input type="checkbox"/>	CHAITANYA	Project_Data.sql	78 seconds ago	CHAITANYA	78 seconds ago	133,939	0	<span style="border: 2px solid green; border-radius: 50%; padding: 2px;">▶</span>
<input type="checkbox"/>	CHAITANYA	Project_Tables.sql	22 minutes ago	CHAITANYA	20 minutes ago	15,164	1	<span style="border: 1px solid gray; border-radius: 50%; padding: 2px;">▶</span>

Click **Run Now**.

f) Click the View Results icon for the script you just ran (top row).

Script	Run By	Started	Finished	Elapsed	Status	Security Group Id	Statements	Bytes	View Results
Project_Data.sql	CHAITANYA	Now	02/07/2017	0.37	Completed	1841547862027812	3 of 3	0	<span style="border: 2px solid green; border-radius: 50%; padding: 2px;">🔍</span>
Project_Tables.sql	CHAITANYA	20 minutes ago	02/07/2017	0.46	Completed	1841547862027812	45 of 45	0	<span style="border: 1px solid gray; border-radius: 50%; padding: 2px;">🔍</span>

SQL Scripts \ Results

Script: **Project\_Data.sql** Status: **Complete**

View:  Detail  **Summary** Rows: 15

Number ↑	Elapsed	Statement	Feedback	Rows
1	0.03	create or replace package demo_projects_data_pkg as functi	Package created.	0
2	0.12	create or replace package body demo_projects_data_pkg as	Package Body created.	0
3	0.22	begin demo_projects_data_pkg.load_sample_data; end;	Statement processed.	1

Download row(s) 1 - 3 of 3

3  
Statements Processed

3  
Successful

0  
With Errors

3. In step 1, you uploaded a package called DEMO\_PROJECTS\_DATA\_PKG. However, this package has not yet been run so the tables you created still do not have any data. The SQL Commands facility, within SQL Workshop, allows a developer to run any valid SQL commands. You will run a SQL command to execute the data package and populate the tables. Use SQL Commands to execute n Oracle Database package. Perform the following steps:
- a) Click the Up arrow, before SQL Scripts.

SQL Scripts \ Results

Script: **Project\_Data.sql** Status: **Complete**


View:  Detail  **Summary** Rows: 15

Number ↑	Elapsed	Statement	Feedback
1	0.03	create or replace package demo_projects_data_pkg as functi	Package created.
2	0.12	create or replace package body demo_projects_data_pkg as	Package Body created.
3	0.22	begin demo_projects_data_pkg.load_sample_data; end;	Statement processed.


- b) Click **SQL Commands**.

ORACLE Application Express


App Builder | SQL Workshop | Team Development | Packaged Apps



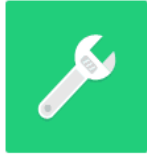
Object Browser




SQL Commands



SQL Scripts



Utilities

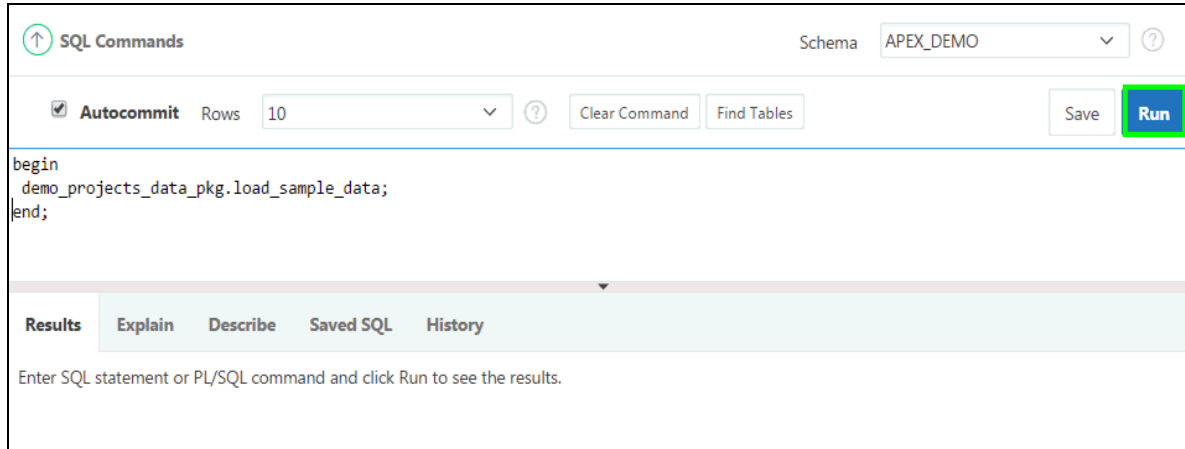


RESTful Services

- c) Enter the following code:

```
begin
demo_projects_data_pkg.load_sample_data;
end;
```

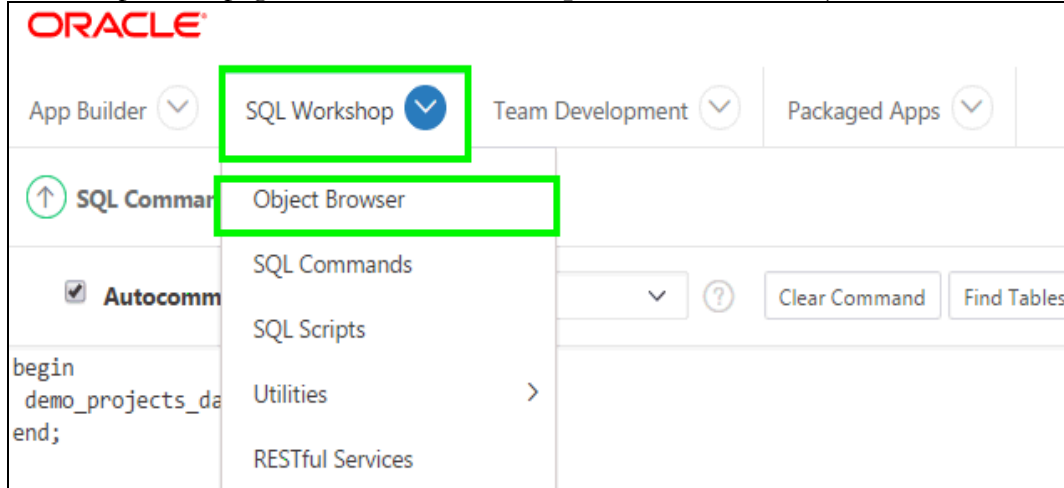
d) Click **Run**.



The Results show: Statement Processed.

4. Use the Object Browser within SQL Workshop to review all of the database objects, such as the tables and packages you created, available in the underlying Oracle database schema which is associated with the Application Express workspace you logged into. Perform the following steps:

a) At the top of the page, select **SQL Workshop** and then select **Object Browser**.



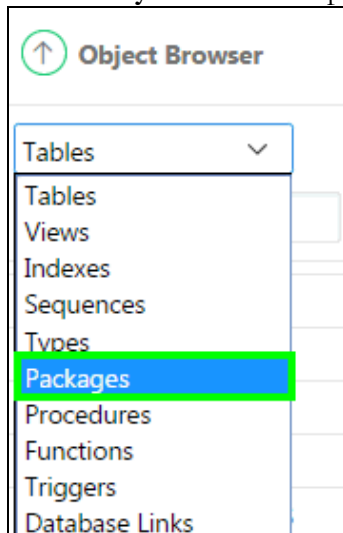
b) In Object Browser, select the **DEMO\_PROJ\_TEAM\_MEMBERS** table, and then click on the **Data** tab.

The screenshot shows the Oracle APEX Object Browser interface. On the left, a list of tables is displayed, with 'DEMO\_PROJ\_TEAM\_MEMBERS' highlighted in green. The main area shows the 'Data' tab for this table, which is also highlighted in green. The table contains three rows of data:

EDIT	ID	USERNAME	FULL_NAME	EMAIL	PROFILE	PHOTO_FILENAME	PHOTO_BLOB
	6	MADY	Madison Smith	madison.smith@email.com	Mady to my friends - I love being creative and coming up with beautiful solutions.	team_member_06.jpg	[datatype]
	7	TKING	Tyson King	alonso.king@email.com	No problem too big, no problem too small!	team_member_07.jpg	[datatype]
	8	DJ	Daniel James	daniel.lee@email.com	I am a DJ, I am what I say. If you ask me a question and I don't answer, it is not because I'm rude, it is	team_member_08.jpg	[datatype]

**Note:** There are a number of other tables listed, outside of those you created using the script file above. The APEX\$ tables are created by Application Express to store internal data specific to your workspace. Tables such as DEMO\_CUSTOMERS were created when the Sample Database Application was installed. The Sample Database Application is installed by default when an Application Express Workspace is created.

- c) To review the package you created, select Packages and select DEMO\_PROJECTS\_DATA\_PKG. Click **Body** to review the primary PL/SQL rather than the specification.



**Note:** This package includes complex PL/SQL code to insert images and replicate users entering in records. It is not important that you understand the PL/SQL code in this package, as you will not normally have to populate data in this matter. Generally, you would

create the tables with no data and then use the application you build to insert the records.

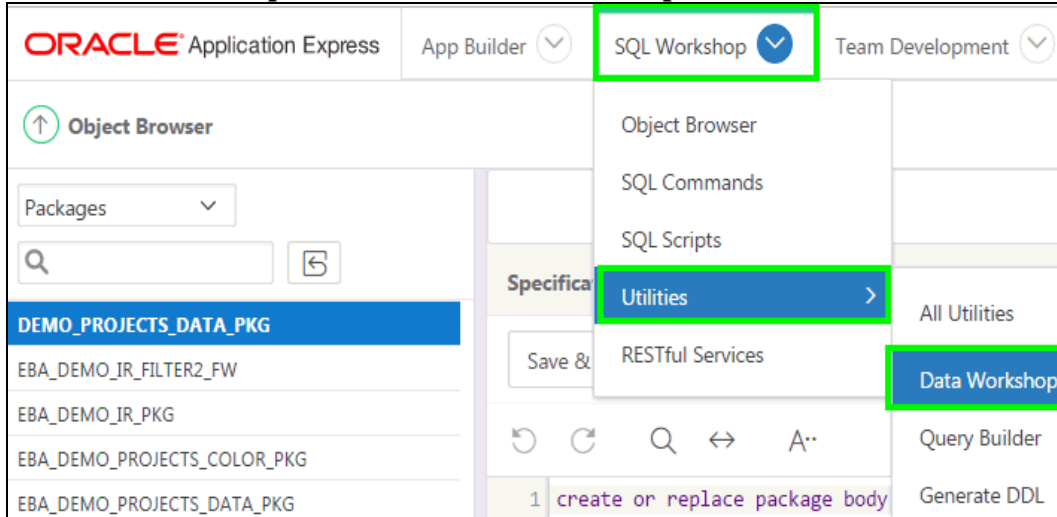
The screenshot displays the Oracle APEX Object Browser interface. On the left, the 'Object Browser' pane shows a list of packages under the 'APEX\_DEMO' schema. The package 'DEMO\_PROJECTS\_DATA\_PKG' is selected and highlighted in blue. The main editor pane shows the 'Body' tab of this package, which is highlighted with a green box. The code in the editor defines a function 'varchar2\_to\_blob' and a procedure 'load\_sample\_data'. The function takes a table of VARCHAR2 values and returns a BLOB. The procedure is designed to load sample data into a table.

```
1 create or replace package body demo_projects_data_pkg as
2   function varchar2_to_blob(p_varchar2_tab in dbms_sql.varchar2_table)
3     return blob
4   is
5     l_blob blob;
6     l_raw raw(500);
7     l_size number;
8   begin
9     dbms_lob.createtemporary(l_blob, true, dbms_lob.session);
10    for i in 1 .. p_varchar2_tab.count loop
11      l_size := length(p_varchar2_tab(i)) / 2;
12      dbms_lob.writeappend(l_blob, l_size, hextoraw(p_varchar2_tab(i)));
13    end loop;
14    return l_blob;
15  exception
16    when others then
17      dbms_lob.close(l_blob);
18  end varchar2_to_blob;
19
20 procedure load_sample_data is
21   i          dbms_sql.varchar2_table;
22   j          dbms_sql.varchar2_table default wwv_flow_api.empty_varchar2_table;
23   l_blob     blob;
24   l_full_name varchar2(255);
25   l_email    varchar2(255);
26   l_add_days number;
27 begin
```

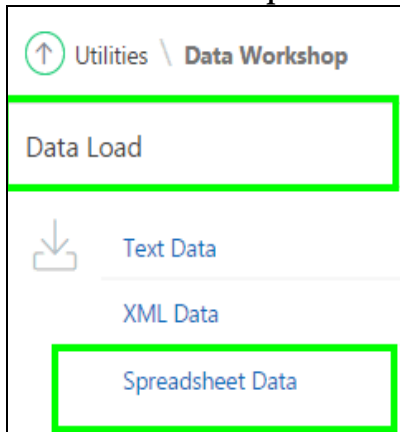
## HOL 2-2: Creating a Lookup Table

In this hands-on-lab, you use the Data Workshop utility to create a table and populate the table with data. Once this table is created, you also create a lookup table.

1. Click **SQL Workshop > Utilities > Data Workshop**.



2. Click **Data Load > Spreadsheet Data**.



3. The Load Data Wizard appears.  
Under Load To, select **New Table**, and under Load From, select **Upload file**.

Click **Next**.

**Load Data - Target and Method**

Target and Method

Load To:  Existing table  **New table**

Load From:  **Upload file (comma separated or tab delimited)**  Copy and paste

Cancel Next >

- Click **Choose File**, open the working directory where you extracted **apex-course-labs.zip**. Locate the **hardware.csv** file, and double-click the file or click the file and then click **Open**. Ensure that First row contains column names is selected, accept the remaining defaults, and click **Next**.

**Load Data - File Details**

File Details

The file to be uploaded must be text-based. To upload a .XLS file, first save it as CSV.

\* Text File  hardware.csv

Preview

serial	cpu type	cpu speed	purchase date	brand	model	form factor	purc
0C8765T	Pentium I	200	09.10.1998	Dell	Optiplex	D	2343
1U09CM2423	Pentium II	667	01.06.1999	Dell	Optiplex	D	3256
0KBU4	Pentium III	1000	12.05.2001	Dell	Optiplex	L	4532
51214246AB	Pentium III	450	14.06.2001	Dell	Optiplex	L	5401

\* Separator: (\t for tab)

Optionally Enclosed By

**First row contains column names.**

< Cancel Next >

5. Enter **Hardware** for Table Name, verify the table properties, and click **Next**.

### Load Data - Table Properties

**Table Properties**

\* Schema:  ?

\* Table Name:  ?  Preserve Case

Column Names	<input type="text" value="serial"/>	<input type="text" value="cpu_type"/>	<input type="text" value="cpu_speed"/>	<input type="text" value="purchase_date"/>	<input type="text" value="brand"/>	<input type="text" value="model"/>
Data Type	<input type="text" value="VARCHAR2"/>	<input type="text" value="VARCHAR2"/>	<input type="text" value="NUMBER"/>	<input type="text" value="DATE"/>	<input type="text" value="VARCHAR2"/>	<input type="text" value="VARCH"/>
Format	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Column Length	<input type="text" value="30"/>	<input type="text" value="30"/>	<input type="text" value="30"/>	<input type="text" value="7"/>	<input type="text" value="30"/>	<input type="text" value="30"/>
Upload	<input type="text" value="Yes"/>	<input type="text" value="Yes"/>	<input type="text" value="Yes"/>	<input type="text" value="Yes"/>	<input type="text" value="Yes"/>	<input type="text" value="Yes"/>
Row 1	0C8765T	Pentium I	200	09.10.1998	Dell	Optiplex
Row 2	1U09CM2423	Pentium II	667	01.06.1999	Dell	Optiplex
Row 3	0KBU4	Pentium III	1000	12.05.2001	Dell	Optiplex
Row 4	51214246AB	Pentium III	450	14.06.2001	Dell	Optiplex

6. Accept the defaults on the Load Data - Primary Key page, and click **Load Data**.

### Load Data - Primary Key

**Primary Key**

Use this page to identify the primary key of the table. If no single column in your data set is appropriate as a primary key then identify a new column as the primary key. A primary key is used to uniquely identify a row in a table.

Schema: **APEX\_DEMO** ?

Table Name: **HARDWARE** ?

Primary Key From:  Use an existing column ?  **Create new column**

\* New Primary Key Column:  ?

\* PK Constraint Name:  ?

7. The new table is now created and is populated with the data.

Click **Object Browser**.

The screenshot shows the Oracle APEX SQL Workshop interface. The 'SQL Workshop' menu is open, and the 'Object Browser' option is highlighted with a green box. Below the menu, a table displays the results of a file import. The table has columns for Details, File, Imported By, Imported On, Type, Schema, Table, Bytes, Succeeded, Failed, and To Be Deleted. The data row shows a file named 'hardware.csv' imported by 'CHAITANYA' 1 second ago, with a type of 'Text Import', schema 'APEX\_DEMO', and table 'HARDWARE'.

Details	File	Imported By	Imported On	Type	Schema	Table	Bytes	Succeeded	Failed	To Be Deleted
	hardware.csv	CHAITANYA	1 seconds ago	Text Import	APEX_DEMO	HARDWARE	2,734	37	0	02/21/2017

8. In the Object Selection pane, click **Hardware**.

The Detail pane now shows details about Hardware table. For the Hardware table, review the column names and data types.

Click **Data**.

Now you see a report of the data contained in the Hardware table.

The screenshot shows the Oracle APEX Object Browser interface. The 'Object Browser' pane on the left lists various tables, with 'HARDWARE' selected and highlighted in green. The main pane displays the 'HARDWARE' table details, including columns and data. The 'Data' tab is selected, showing a table with columns: EDIT, ID, SERIAL, CPU\_TYPE, CPU\_SPEED, PURCHASE\_DATE, BRAND, MODEL, FORM\_FACTOR, and PURCHASE\_PRICE. The data rows are as follows:

EDIT	ID	SERIAL	CPU_TYPE	CPU_SPEED	PURCHASE_DATE	BRAND	MODEL	FORM_FACTOR	PURCHASE_PRICE
	1	0C8765T	Pentium I	200	10/09/1998	Dell	Optiplex	D	2343
	2	1U09CM2423	Pentium II	667	06/01/1999	Dell	Optiplex	D	3256
	3	0KBU4	Pentium III	1000	05/12/2001	Dell	Optiplex	L	4532
	4	51214246AB	Pentium III	450	06/14/2001	Dell	Optiplex	L	5401
	5	H0MH12	Pentium	1700	06/15/2002	Dell	Optiplex	D	6588

9. You want to create a lookup table now. Perform the following steps:

- a) Click **Table**. Click **Create Lookup Table**.

HARDWARE										
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL
Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	Truncate	Create Lookup Table		
Column Name	Data Type	Nullable	Default	Primary Key						
ID	NUMBER	No	-	1						
SERIAL	VARCHAR2(30)	Yes	-	-						
CPU_TYPE	VARCHAR2(30)	Yes	-	-						
CPU_SPEED	NUMBER	Yes	-	-						
PURCHASE_DATE	DATE	Yes	-	-						
BRAND	VARCHAR2(30)	Yes	-	-						
MODEL	VARCHAR2(30)	Yes	-	-						

- b) Select **DEPARTMENT** for Column. Click **Next**.

HARDWARE	
Create Lookup Table	
Select the column you would like to create a lookup table for. The selected column will become a foreign key to the lookup table.	
Schema:	APEX_DEMO ?
Table Name:	HARDWARE ?
Show:	<input type="radio"/> All Column Types ? <input checked="" type="radio"/> VARCHAR Column Types
* Column:	<input type="radio"/> SERIAL - varchar2 ? <input type="radio"/> CPU_TYPE - varchar2 <input type="radio"/> BRAND - varchar2 <input type="radio"/> MODEL - varchar2 <input type="radio"/> FORM_FACTOR - varchar2 <input checked="" type="radio"/> DEPARTMENT - varchar2
Cancel	Next >

- c) Accept the defaults on this page and click **Next**.

**HARDWARE** + v

---

Create Lookup Table

Specify the new table you wish to create. This new table will store a normalized list of values (that is, a unique list of values in the selected column).

Schema: **APEX\_DEMO** ?

Table Name: **HARDWARE** ?

Column to Normalize: **DEPARTMENT** ?

New Table Name:  ?

New Sequence:  ?

Preserve Case

< Cancel **Next** >

- d) Click **Create Lookup Table**.

**HARDWARE** + v

---

Create Lookup Table

Confirming this request, creates a new table and adjusts your current table structure.

Schema: **APEX\_DEMO** ?

Lookup table: **DEPARTMENT\_LOOKUP** ?

Table: **HARDWARE** ?

Lookup table primary key: **DEPARTMENT\_ID** ?

Lookup based on Column: **DEPARTMENT** ?

Lookup table sequence: **DEPARTMENT\_LOOKUP\_SEQ** ?

< Cancel **Create Lookup Table**

e) Review the table definition of **DEPARTMENT LOOKUP** table.

The screenshot shows the Oracle Object Browser interface for the APEX\_DEMO schema. The 'DEPARTMENT\_LOOKUP' table is selected in the left pane. The main pane displays the table's structure with the following columns:

Column Name	Data Type	Nullable	Default	Primary Key
DEPARTMENT_ID	NUMBER	No	-	1
DEPARTMENT	VARCHAR2(4000)	No	-	-

f) In the Object Selection pane, click **HARDWARE**.

Review the table details. Notice that the DEPARTMENT column has been extracted from the HARDWARE table and is now available in the DEPARTMENT\_LOOKUP table.

The screenshot shows the Oracle Object Browser interface for the APEX\_DEMO schema. The 'HARDWARE' table is selected in the left pane. The main pane displays the table's structure with the following columns:

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
SERIAL	VARCHAR2(30)	Yes	-	-
CPU_TYPE	VARCHAR2(30)	Yes	-	-
CPU_SPEED	NUMBER	Yes	-	-
PURCHASE_DATE	DATE	Yes	-	-
BRAND	VARCHAR2(30)	Yes	-	-
MODEL	VARCHAR2(30)	Yes	-	-
FORM_FACTOR	VARCHAR2(1)	Yes	-	-
PURCHASE_PRICE	NUMBER	Yes	-	-
DEPARTMENT_ID	NUMBER	Yes	-	-



Oracle is committed to developing practices and products that help protect the environment

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
oracle.com

Copyright © 2017, Oracle and/or its affiliates. All rights reserved. Licensed under the Creative Commons Attribution 4.0 International License as shown at <https://creativecommons.org/licenses/by/4.0/legalcode>.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

**Hardware and Software, Engineered to Work Together**