

SQL Database Programming: Section 6-1: Cross Joins and Natural Joins

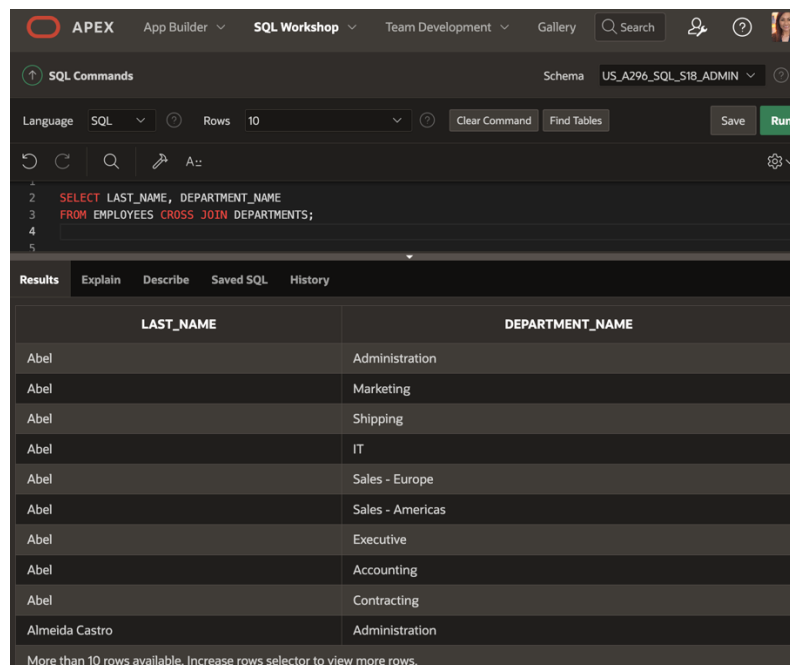
Vocabulary

Cross Join – Returns the Cartesian product from two tables.

Natural Join – Joins two tables based on the same column name.

Try It/Solve It

1. Create a cross-join that displays the last name and department name from the employees and departments tables.



The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' panel is active, showing a query: `SELECT LAST_NAME, DEPARTMENT_NAME
FROM EMPLOYEES CROSS JOIN DEPARTMENTS;`. The 'Schema' is set to 'US_A296_SQL_S18_ADMIN'. The 'Language' is 'SQL' and 'Rows' are set to 10. The 'Run' button is highlighted in green. Below the query editor, the 'Results' tab is selected, displaying a table with two columns: 'LAST_NAME' and 'DEPARTMENT_NAME'. The table contains 11 rows of data, showing the Cartesian product of the 'EMPLOYEES' and 'DEPARTMENTS' tables. The first 10 rows show 'Abel' with various departments, and the 11th row shows 'Almeida Castro' with 'Administration'. A message at the bottom states: 'More than 10 rows available. Increase rows selector to view more rows.'

LAST_NAME	DEPARTMENT_NAME
Abel	Administration
Abel	Marketing
Abel	Shipping
Abel	IT
Abel	Sales - Europe
Abel	Sales - Americas
Abel	Executive
Abel	Accounting
Abel	Contracting
Almeida Castro	Administration

2. Create a query that uses a natural join to join the departments table and the locations table. Display the department id, department name, location id, and city.

The screenshot shows the APEX SQL Workshop interface. The SQL Commands panel contains the following query:

```
1  
2 SELECT DEPARTMENT_ID, DEPARTMENT_NAME, LOCATION_ID, CITY  
3 FROM DEPARTMENTS NATURAL JOIN LOCATIONS;  
4  
5
```

The Results panel displays the following table:

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
60	IT	1400	Southlake
50	Shipping	1500	South San Francisco
10	Administration	1700	Seattle
90	Executive	1700	Seattle
110	Accounting	1700	Seattle
190	Contracting	1700	Seattle
20	Marketing	1800	Toronto
85	Sales - Americas	2100	Rio de Janeiro
80	Sales - Europe	2500	Oxford

9 rows returned in 0.02 seconds

3. Create a query that uses a natural join to join the departments table and the locations table. Restrict the output to only department IDs of 20 and 50. Display the department id, department name, location id, and city.

The screenshot shows the APEX SQL Workshop interface. The SQL Commands panel contains the following query:

```
1  
2 SELECT DEPARTMENT_ID, DEPARTMENT_NAME, LOCATION_ID, CITY  
3 FROM DEPARTMENTS NATURAL JOIN LOCATIONS  
4 WHERE DEPARTMENT_ID IN (20, 50);
```

The Results panel displays the following table:

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
50	Shipping	1500	South San Francisco
20	Marketing	1800	Toronto

2 rows returned in 0.01 seconds

SQL Database Programming: Section 6-2: Join Clauses

Vocabulary

- **Join with the ANSI-99 ON Clause** – Allows a natural join based on an arbitrary condition or two columns with different names.
- **Join with the ANSI-99 USING Clause** – Performs an equijoin based on one specified column name

Try It/Solve It

1. Join the Oracle database locations and departments table using the location_id column. Limit the results to location 1400 only.

The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
1 SELECT LOCATION_ID
2 FROM LOCATIONS JOIN DEPARTMENTS USING (LOCATION_ID)
3 WHERE LOCATION_ID = '1400';
```

The results pane shows a single row with the value 1400 under the column header LOCATION_ID.

LOCATION_ID
1400

1 rows returned in 0.01 seconds

2. Join DJs on Demand d_play_list_items, d_track_listings, and d_cds tables with the JOIN USING syntax. Include the song ID, CD number, title, and comments in the output.

The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
1 SELECT SONG_ID, CD_NUMBER, TITLE, COMMENTS
2 FROM D_PLAY_LIST_ITEMS JOIN D_TRACK_LISTINGS USING (SONG_ID)
3 JOIN D_CDS USING (CD_NUMBER);
```

The results pane shows 6 rows of data with columns SONG_ID, CD_NUMBER, TITLE, and COMMENTS.

SONG_ID	CD_NUMBER	TITLE	COMMENTS
47	91	Party Music for All Occasions	Play early
49	91	Party Music for All Occasions	Play first
47	91	Party Music for All Occasions	Play for the father
45	92	Back to the Shire	Play late
46	93	Songs from My Childhood	-
48	95	Here Comes the Bride	Play after cake cutting

6 rows returned in 0.02 seconds

3. Display the city, department name, location ID, and department ID for departments 10, 20, and 30 for the city of Seattle.

The screenshot shows the APEX SQL Workshop interface. The SQL Commands tab is active, displaying a query that selects city, department name, location ID, and department ID for departments 10, 20, and 30 where the city is Seattle. The Results tab shows a single row of data.

```
2 SELECT CITY, DEPARTMENT_NAME, LOCATION_ID, DEPARTMENT_ID
3 FROM DEPARTMENTS JOIN LOCATIONS USING (LOCATION_ID)
4 WHERE DEPARTMENT_ID IN (10, 20, 30) AND CITY = 'Seattle';
5
6
```

CITY	DEPARTMENT_NAME	LOCATION_ID	DEPARTMENT_ID
Seattle	Administration	1700	10

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

4. Display country name, region ID, and region name for Americas.

The screenshot shows the APEX SQL Workshop interface. The SQL Commands tab is active, displaying a query that selects country name, region ID, and region name for regions with names like 'Americas'. The Results tab shows 'no data found'.

```
1
2 SELECT COUNTRY_NAME, REGION_ID, REGION_NAME
3 FROM COUNTRIES JOIN REGIONS USING (REGION_ID)
4 WHERE REGION_NAME LIKE 'Americas';
5
```

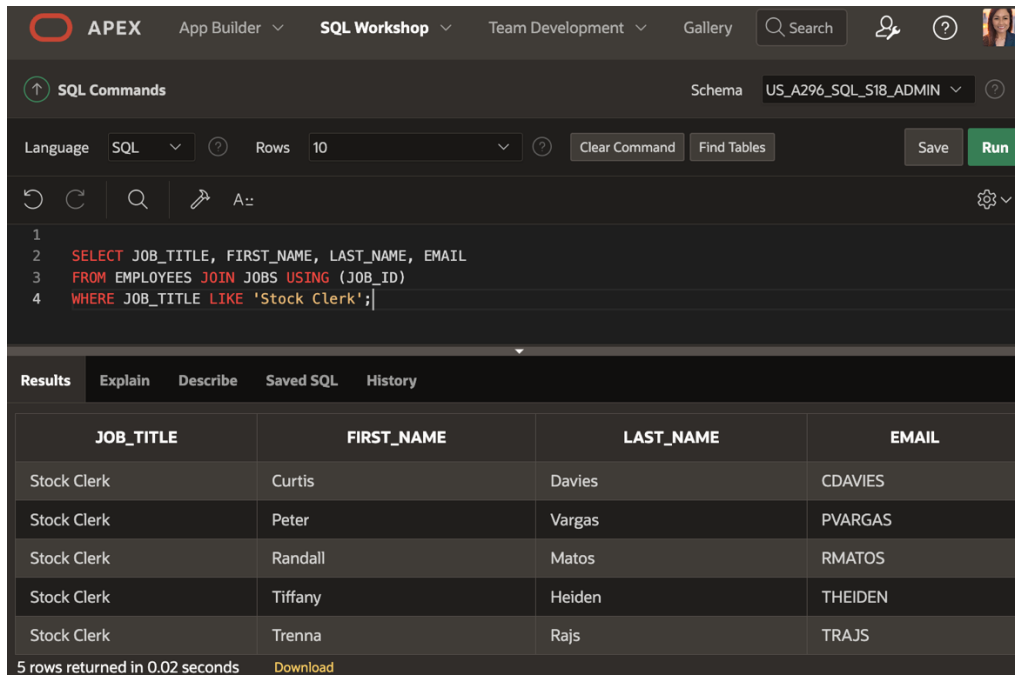
Results Explain Describe Saved SQL History

no data found

5. Write a statement joining the employees and jobs tables. Display the first and last names, hire date, job id, job title, and maximum salary. Limit the query to those employees who are in jobs that can earn more than \$12,000.

- Failed to successfully run a query, I keep getting error messages.

6. Display job title, employee first name, last name, and email for all employees who are stock clerks.



The screenshot shows the APEX SQL Workshop interface. At the top, there's a navigation bar with 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below this, the 'SQL Commands' section is active, showing a schema of 'US_A296_SQL_S18_ADMIN'. The SQL editor contains the following query:

```
1  
2 SELECT JOB_TITLE, FIRST_NAME, LAST_NAME, EMAIL  
3 FROM EMPLOYEES JOIN JOBS USING (JOB_ID)  
4 WHERE JOB_TITLE LIKE 'Stock Clerk';
```

Below the editor, the 'Results' tab is selected, displaying a table with 5 rows. The table has columns: JOB_TITLE, FIRST_NAME, LAST_NAME, and EMAIL. The data is as follows:

JOB_TITLE	FIRST_NAME	LAST_NAME	EMAIL
Stock Clerk	Curtis	Davies	CDAVIES
Stock Clerk	Peter	Vargas	PVARGAS
Stock Clerk	Randall	Matos	RMATOS
Stock Clerk	Tiffany	Heiden	THEIDEN
Stock Clerk	Trenna	Rajs	TRAJS

At the bottom of the results section, it states '5 rows returned in 0.02 seconds' and provides a 'Download' link.

SQL Database Programming: Section 6-3: Inner versus Outer Joins

Vocabulary

FULL OUTER Join – Performs a join on two tables, retrieves all the rows in the Left table, even if there is no match in the Right table. It also retrieves all the rows in the Right table, even if there is no match in the Left table.

OUTER Join – A join that returns the unmatched rows as well as matched rows

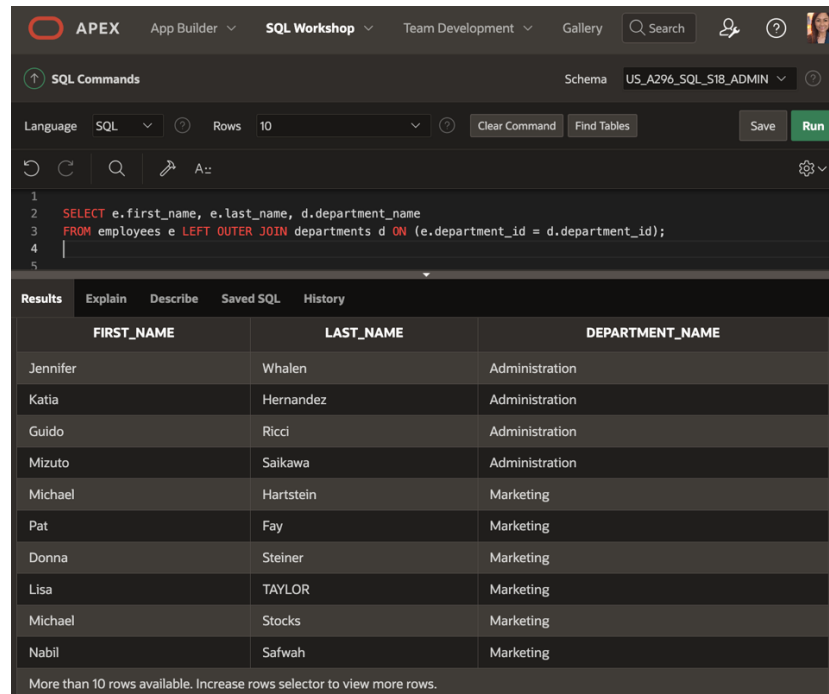
LEFT Outer Join – Performs a join on two tables, retrieves all the rows in the Left table even if there is no match in the Right table.

RIGHT Outer Join – Performs a join on two tables, retrieves all the rows in the Right table even if there is no match in the Left table.

INNER Join – A join of two or more tables that returns only matched rows

Try It/Solve It

1. Return the first name, last name, and department name for all employees including those employees not assigned to a department.



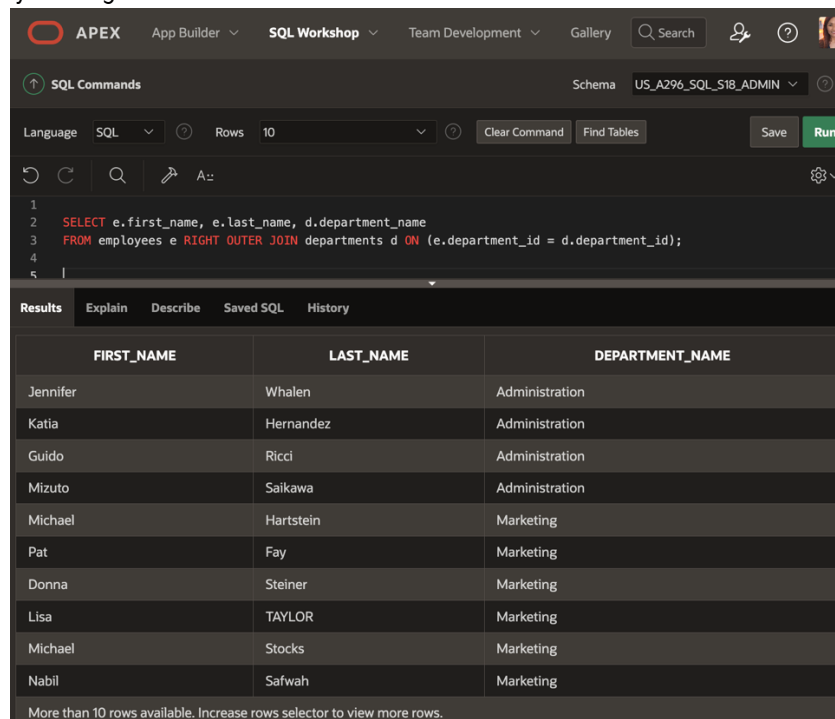
The screenshot shows the APEX SQL Workshop interface. The SQL command is:

```
1  
2 SELECT e.first_name, e.last_name, d.department_name  
3 FROM employees e LEFT OUTER JOIN departments d ON (e.department_id = d.department_id);  
4  
5
```

The results are displayed in a table with columns FIRST_NAME, LAST_NAME, and DEPARTMENT_NAME. The table contains 10 rows of data, showing employees and their corresponding departments. The first 10 rows are displayed, and a message indicates that more than 10 rows are available.

FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
Jennifer	Whalen	Administration
Katia	Hernandez	Administration
Guido	Ricci	Administration
Mizuto	Saikawa	Administration
Michael	Hartstein	Marketing
Pat	Fay	Marketing
Donna	Steiner	Marketing
Lisa	TAYLOR	Marketing
Michael	Stocks	Marketing
Nabil	Safwah	Marketing

2. Return the first name, last name, and department name for all employees including those departments that do not have an employee assigned to them.



The screenshot shows the APEX SQL Workshop interface. The SQL command is:

```
1  
2 SELECT e.first_name, e.last_name, d.department_name  
3 FROM employees e RIGHT OUTER JOIN departments d ON (e.department_id = d.department_id);  
4  
5
```

The results are displayed in a table with columns FIRST_NAME, LAST_NAME, and DEPARTMENT_NAME. The table contains 10 rows of data, showing employees and their corresponding departments. The first 10 rows are displayed, and a message indicates that more than 10 rows are available.

FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
Jennifer	Whalen	Administration
Katia	Hernandez	Administration
Guido	Ricci	Administration
Mizuto	Saikawa	Administration
Michael	Hartstein	Marketing
Pat	Fay	Marketing
Donna	Steiner	Marketing
Lisa	TAYLOR	Marketing
Michael	Stocks	Marketing
Nabil	Safwah	Marketing

3. Return the first name, last name, and department name for all employees including those departments that do not have an employee assigned to them and those employees not assigned to a department.

The screenshot shows the APEX SQL Workshop interface. The 'SQL Commands' tab is active, displaying a query that performs a full outer join between the 'employees' and 'departments' tables. The query is as follows:

```

1
2 SELECT e.first_name, e.last_name, d.department_name
3 FROM employees e FULL OUTER JOIN departments d ON (e.department_id = d.department_id);
4
5

```

The 'Results' tab is selected, showing a table with 10 rows and 3 columns: FIRST_NAME, LAST_NAME, and DEPARTMENT_NAME. The data is as follows:

FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
Jennifer	Whalen	Administration
Katia	Hernandez	Administration
Guido	Ricci	Administration
Mizuto	Saikawa	Administration
Michael	Hartstein	Marketing
Pat	Fay	Marketing
Donna	Steiner	Marketing
Lisa	TAYLOR	Marketing
Michael	Stocks	Marketing
Nabil	Safwah	Marketing

More than 10 rows available. Increase rows selector to view more rows.

4. Create a query of the DJs on Demand database to return the first name, last name, event date, and description of the event the client held. Include all the clients even if they have not had an event scheduled.

The screenshot shows the APEX SQL Workshop interface. The 'SQL Commands' tab is active, displaying a query that performs a left outer join between the 'D_CLIENTS' and 'D_EVENTS' tables. The query is as follows:

```

1
2 SELECT c.FIRST_NAME, c.LAST_NAME, e.EVENT_DATE, e.DESRIPTION
3 FROM D_CLIENTS c LEFT OUTER JOIN D_EVENTS e ON (e.CLIENT_NUMBER = c.CLIENT_NUMBER);
4
5

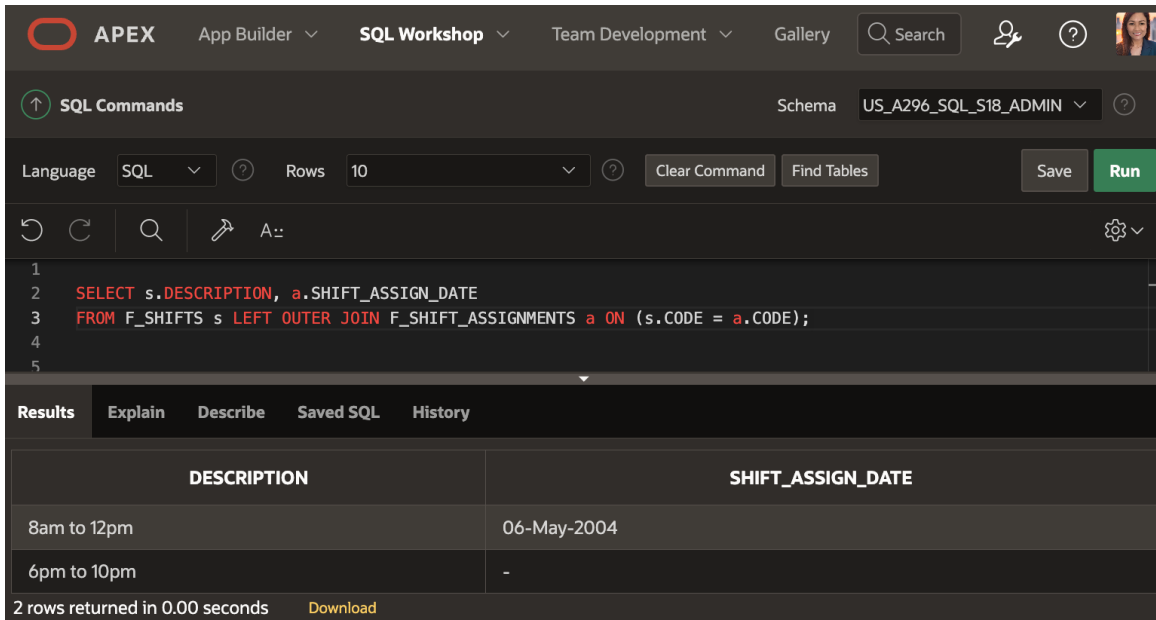
```

The 'Results' tab is selected, showing a table with 3 rows and 4 columns: FIRST_NAME, LAST_NAME, EVENT_DATE, and DESCRIPTION. The data is as follows:

FIRST_NAME	LAST_NAME	EVENT_DATE	DESCRIPTION
Hiram	Peters	14-May-2004	Party for 200, red, white, blue motif
Lauren	Vigil	28-Apr-2004	Black tie at Four Season hotel
Serena	Jones	-	-

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

5. Using the Global Fast Foods database, show the shift description and shift assignment date even if there is no date assigned for each shift description.



The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' section is active, showing a schema of 'US_A296_SQL_S18_ADMIN'. The SQL editor contains the following query:

```
1  
2 SELECT s.DESCRPTION, a.SHIFT_ASSIGN_DATE  
3 FROM F_SHIFTS s LEFT OUTER JOIN F_SHIFT_ASSIGNMENTS a ON (s.CODE = a.CODE);  
4  
5
```

The 'Results' tab is selected, displaying a table with two columns: 'DESCRIPTION' and 'SHIFT_ASSIGN_DATE'. The table contains two rows of data:

DESCRIPTION	SHIFT_ASSIGN_DATE
8am to 12pm	06-May-2004
6pm to 10pm	-

At the bottom, it states '2 rows returned in 0.00 seconds' and provides a 'Download' link.

SQL Database Programming: Section 6-4: Self Joins and Hierarchical Queries

Vocabulary

SELF-JOIN – Joins a table to itself

HIERARCHICAL QUERY – Retrieves data based on a natural hierarchical relationship between rows in a table

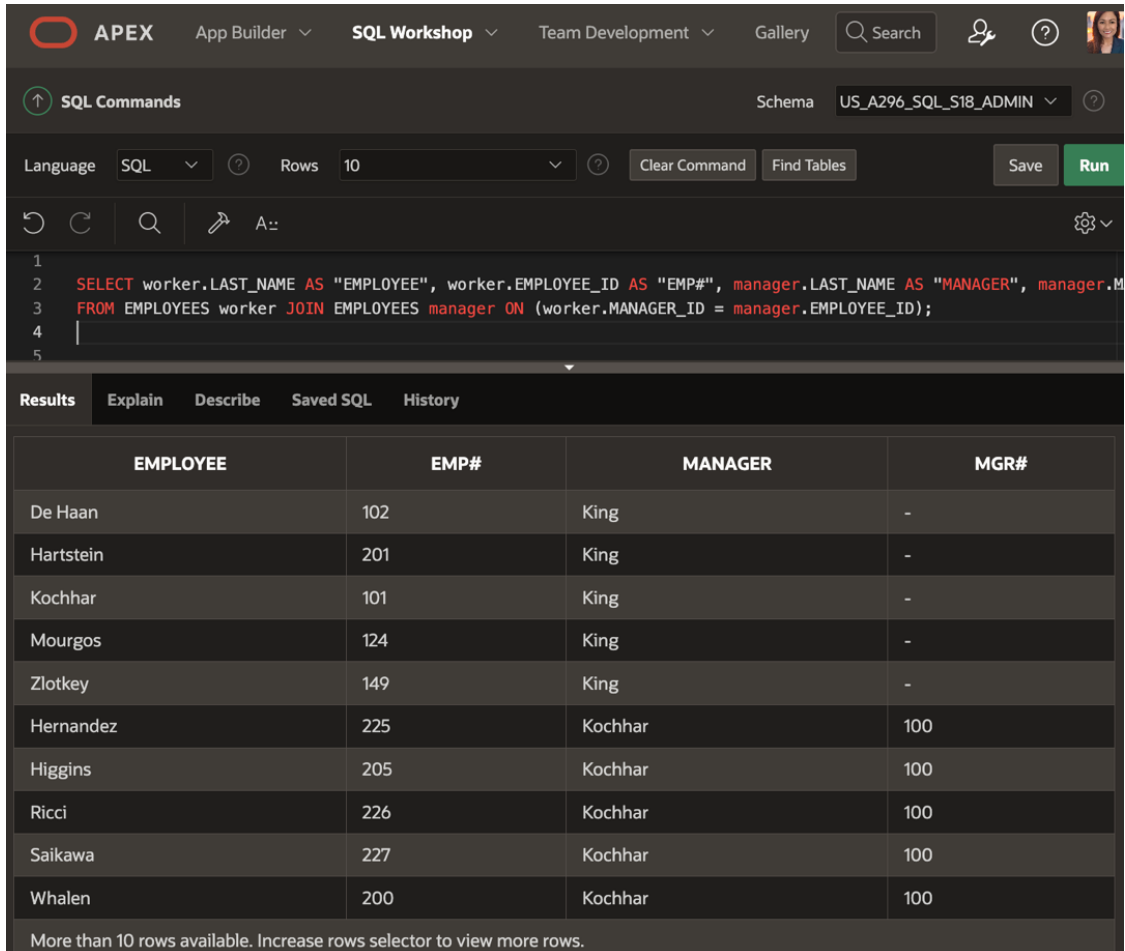
LEVEL – Determines the number of steps down from the beginning row that should be returned by a hierarchical query

START WITH – Identifies the beginning row for a hierarchical query

CONNECT BY PRIOR - Specifies the relationship between parent rows and child rows of a hierarchical query

Try It/Solve It

1. Display the employee's last name and employee number along with the manager's last name and manager number. Label the columns: Employee, Emp#, Manager, and Mgr#, respectively.



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes the APEX logo, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and user profile are also present. The main area is titled "SQL Commands" and shows the schema "US_A296_SQL_S18_ADMIN". The language is set to "SQL" and the number of rows to display is "10". The SQL command entered is:

```
1  
2 SELECT worker.LAST_NAME AS "EMPLOYEE", worker.EMPLOYEE_ID AS "EMP#", manager.LAST_NAME AS "MANAGER", manager.MANAGER_ID AS "MGR#"   
3 FROM EMPLOYEES worker JOIN EMPLOYEES manager ON (worker.MANAGER_ID = manager.EMPLOYEE_ID);  
4  
5
```

The results are displayed in a table with the following columns: EMPLOYEE, EMP#, MANAGER, and MGR#. The table contains 10 rows of data, showing the employee's last name, employee ID, manager's last name, and manager ID. The results are as follows:

EMPLOYEE	EMP#	MANAGER	MGR#
De Haan	102	King	-
Hartstein	201	King	-
Kochhar	101	King	-
Mourgos	124	King	-
Zlotkey	149	King	-
Hernandez	225	Kochhar	100
Higgins	205	Kochhar	100
Ricci	226	Kochhar	100
Saikawa	227	Kochhar	100
Whalen	200	Kochhar	100

More than 10 rows available. Increase rows selector to view more rows.

2. Modify question 1 to display all employees and their managers, even if the employee does not have a manager. Order the list alphabetically by the last name of the employee.

The screenshot shows the APEX SQL Workshop interface. At the top, there's a navigation bar with 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and user profile are also present. Below this, the 'SQL Commands' section is active, showing a schema of 'US_A296_SQL_S18_ADMIN'. The SQL editor contains the following query:

```
1  
2 SELECT worker.EMPLOYEE_ID AS "EMP#", worker.LAST_NAME AS "EMPLOYEE LAST NAME", worker.FIRST_NAME, manager.MANAGER_ID AS "MGR#",  
3 FROM EMPLOYEES worker JOIN EMPLOYEES manager ON (worker.MANAGER_ID = manager.EMPLOYEE_ID)  
4 ORDER BY worker.LAST_NAME;  
5
```

Below the editor, the 'Results' tab is selected, displaying a table with 6 columns: EMP#, EMPLOYEE LAST NAME, FIRST_NAME, MGR#, MANAGER LAST NAME, and MANAGER FIRST NAME. The table contains 12 rows of data.

EMP#	EMPLOYEE LAST NAME	FIRST_NAME	MGR#	MANAGER LAST NAME	MANAGER FIRST NAME
174	Abel	Ellen	100	Zlotkey	Eleni
210	Almeida Castro	Lucas	100	Zlotkey	Eleni
209	Alves Rocha	Sarah	100	Zlotkey	Eleni
207	Barbosa Souza	Sophia	100	Zlotkey	Eleni
216	Bell	George	100	Mourgos	Kevin
142	Davies	Curtis	100	Mourgos	Kevin
102	De Haan	Lex	-	King	Steven
231	Duric	Jelena	101	Higgins	Shelley
104	Ernst	Bruce	102	Hunold	Alexander
202	Fay	Pat	100	Hartstein	Michael

3. Display the names and hire dates for all employees who were hired before their managers, along with their managers' names and hire dates. Label the columns Employee, Emp Hired, Manager and Mgr Hired, respectively.

APEX App Builder ▾ **SQL Workshop** ▾ Team Development ▾ Gallery 🔍 Search 👤 ? 🖼️

SQL Commands Schema US_A296_SQL_S18_ADMIN ?

Language SQL ▾ ? Rows 10 ▾ ? Clear Command Find Tables Save Run

🔄 ↺ 🔍 ✎ A:: ⚙️ ▾

```

1 SELECT worker.LAST_NAME AS "EMPLOYEE", worker.HIRE_DATE as "EMP HIRED", manager.LAST_NAME AS "MANAGER", manage
2 FROM EMPLOYEES worker JOIN EMPLOYEES manager ON (worker.MANAGER_ID = manager.EMPLOYEE_ID)
3 WHERE worker.HIRE_DATE < manager.HIRE_DATE
4 ORDER BY worker.LAST_NAME;
5

```

Results Explain Describe Saved SQL History

EMPLOYEE	EMP HIRED	MANAGER	MGR HIRED
Abel	11-May-2011	Zlotkey	29-Jan-2015
Almeida Castro	16-Aug-2012	Zlotkey	29-Jan-2015
Alves Rocha	06-Feb-2011	Zlotkey	29-Jan-2015
Barbosa Souza	12-Mar-2009	Zlotkey	29-Jan-2015
Bell	01-Apr-2014	Mourgos	16-Nov-2014
Davies	29-Jan-2012	Mourgos	16-Nov-2014
Duric	11-May-2009	Higgins	07-Jun-2009
Grant	24-May-2014	Zlotkey	29-Jan-2015
Hooper	01-Sep-2012	Zlotkey	29-Jan-2015
Hunold	03-Jan-2005	De Haan	13-Jan-2008

4. Write a report that shows the hierarchy for Lex De Haans department. Include last name, salary, and department id in the report.

SQL Commands

Schema: US_A296_SQL_S18_ADMIN

Language: SQL Rows: 10

```

1 SELECT LAST_NAME, SALARY, DEPARTMENT_ID
2 FROM EMPLOYEES
3 START WITH LAST_NAME = 'De Haan'
4 CONNECT BY PRIOR EMPLOYEE_ID = MANAGER_ID
5 ORDER BY SALARY;
6

```

Results Explain Describe Saved SQL History

LAST_NAME	SALARY	DEPARTMENT_ID
Lorentz	4200	60
Ernst	6000	60
Fontaine	7800	60
Li	8000	60
Hunold	9000	60
De Haan	17000	90

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

5. What is wrong in the following statement?

SELECT last_name, department_id, salary

FROM employees

START WITH last_name = 'King'

CONNECT BY PRIOR manager_id = employee_id;

- It should be "employee_id = manager_id": to build the hierarchy correctly

SQL Commands

Schema: US_A296_SQL_S18_ADMIN

Language: SQL Rows: 10

```

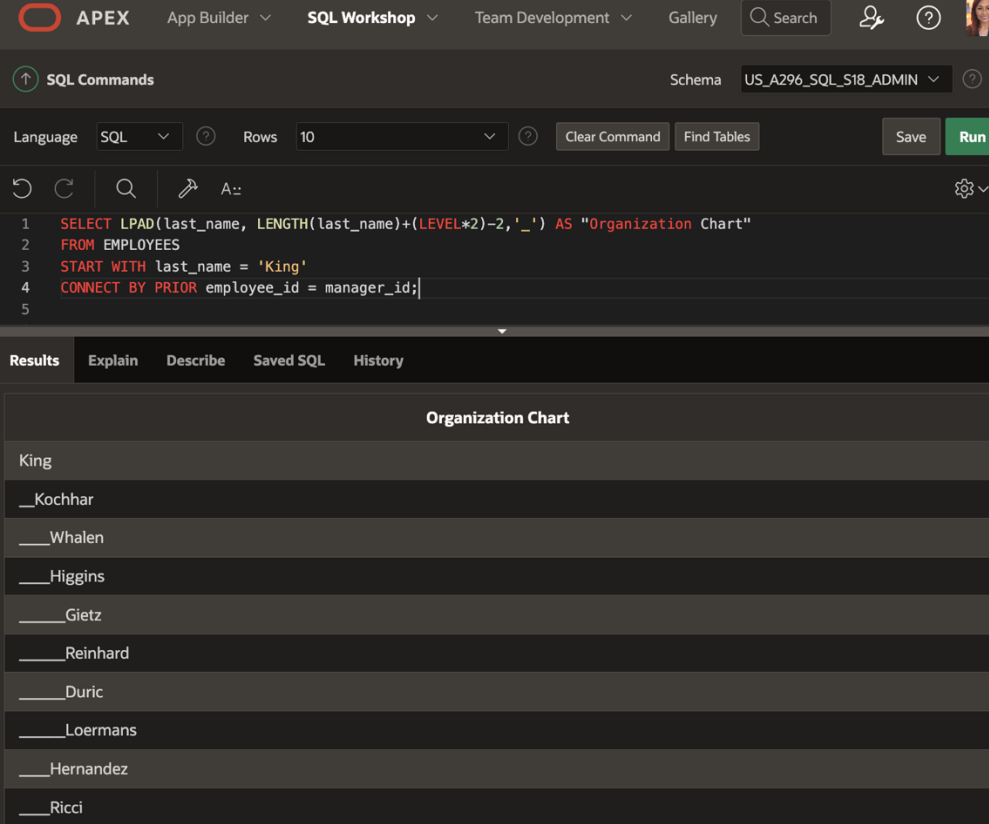
1 SELECT last_name, department_id, salary
2 FROM employees
3 START WITH last_name = 'King'
4 CONNECT BY PRIOR employee_id = manager_id;
5

```

Results Explain Describe Saved SQL History

LAST_NAME	DEPARTMENT_ID	SALARY
King	90	24000
Kochhar	90	17000
Whalen	10	4400
Higgins	110	12000
Gietz	110	8300
Reinhard	110	8100
Duric	110	5400
Loermans	110	5200
Hernandez	10	4300
Ricci	10	4100

6. Create a report that shows the organization chart for the entire employee table. Write the report so that each level will indent each employee 2 spaces. Since Oracle Application Express cannot display the spaces in front of the column, use - (minus) instead.



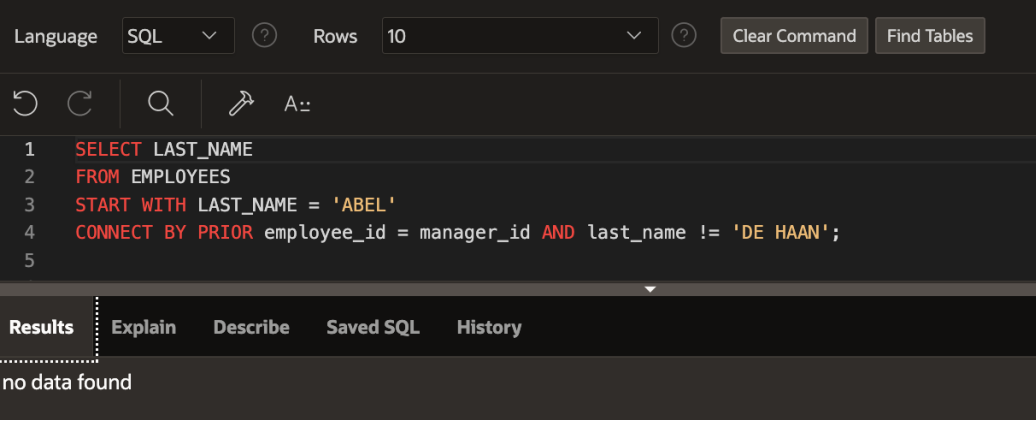
The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a query in the 'US_A296_SQL_S18_ADMIN' schema. The query is as follows:

```
1 SELECT LPAD(last_name, LENGTH(last_name)+(LEVEL*2)-2, '-') AS "Organization Chart"
2 FROM EMPLOYEES
3 START WITH last_name = 'King'
4 CONNECT BY PRIOR employee_id = manager_id;
```

The 'Results' tab is selected, displaying the output of the query. The results are organized into a table with the title 'Organization Chart'. The first row is 'King'. The subsequent rows are indented by two spaces (represented by minus signs) to show the hierarchy: '_Kochhar', '__Whalen', '__Higgins', '__Gietz', '__Reinhard', '__Duric', '__Loermans', '__Hernandez', and '__Ricci'.

Organization Chart
King
_Kochhar
__Whalen
__Higgins
__Gietz
__Reinhard
__Duric
__Loermans
__Hernandez
__Ricci

7. Re-write the report from 6 to exclude De Haan and all the people working for him.



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a query in the 'US_A296_SQL_S18_ADMIN' schema. The query is as follows:

```
1 SELECT LAST_NAME
2 FROM EMPLOYEES
3 START WITH LAST_NAME = 'ABEL'
4 CONNECT BY PRIOR employee_id = manager_id AND last_name != 'DE HAAN';
```

The 'Results' tab is selected, displaying the output of the query. The results are organized into a table with the title 'Organization Chart'. The first row is 'King'. The subsequent rows are indented by two spaces (represented by minus signs) to show the hierarchy: '_Kochhar', '__Whalen', '__Higgins', '__Gietz', '__Reinhard', '__Duric', '__Loermans', '__Hernandez', and '__Ricci'.

Organization Chart
King
_Kochhar
__Whalen
__Higgins
__Gietz
__Reinhard
__Duric
__Loermans
__Hernandez
__Ricci