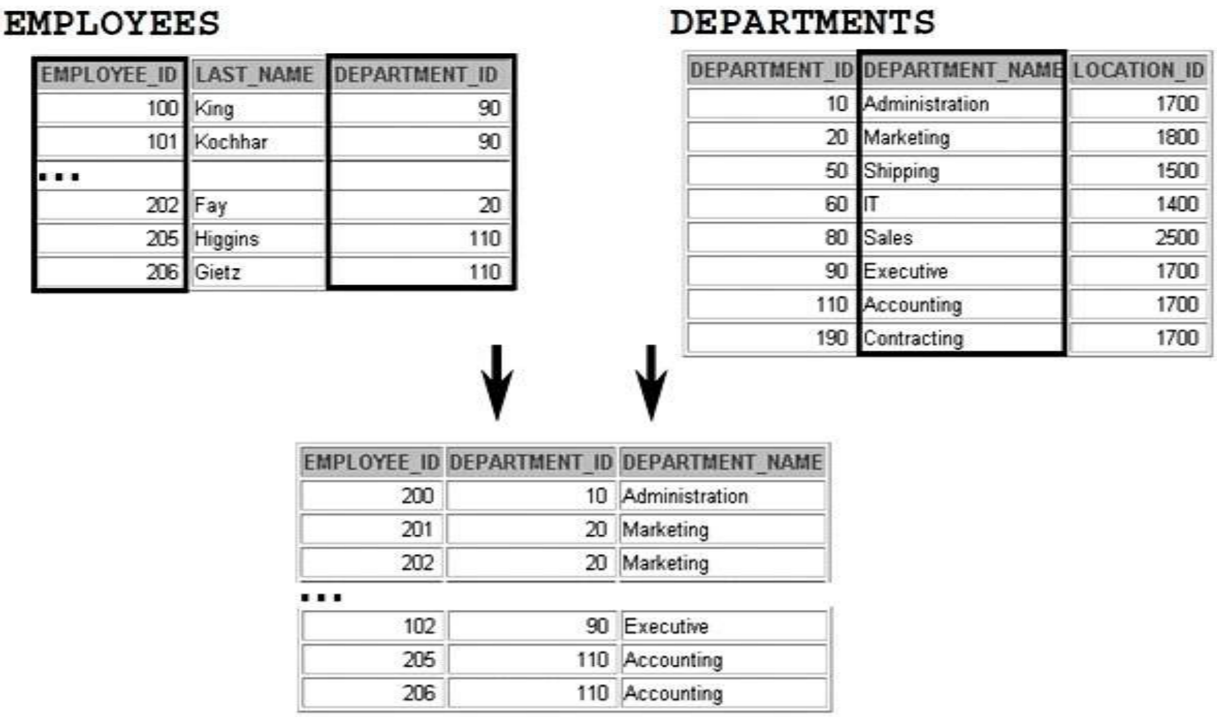
CSE 311L(Database Management System)

LAB-Week 04 (Lecture 1)

Displaying Data from Multiple Tables

Topics:

- Obtaining Data from Multiple Tables
- Generating a Cartesian Product
- Retrieving Records with Equijoins
- Joining a Table to Itself Creating Joins with the ON Clause



EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
...		
202	Fay	20
205	Higgins	110
206	Gietz	110

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
10	Administration	1700
20	Marketing	1800
50	Shipping	1500
60	IT	1400
80	Sales	2500
90	Executive	1700
110	Accounting	1700
190	Contracting	1700

EMPLOYEE_ID	DEPARTMENT_ID	DEPARTMENT_NAME
200	10	Administration
201	20	Marketing
202	20	Marketing
...		
102	90	Executive
205	110	Accounting
206	110	Accounting

Generating a Cartesian Product

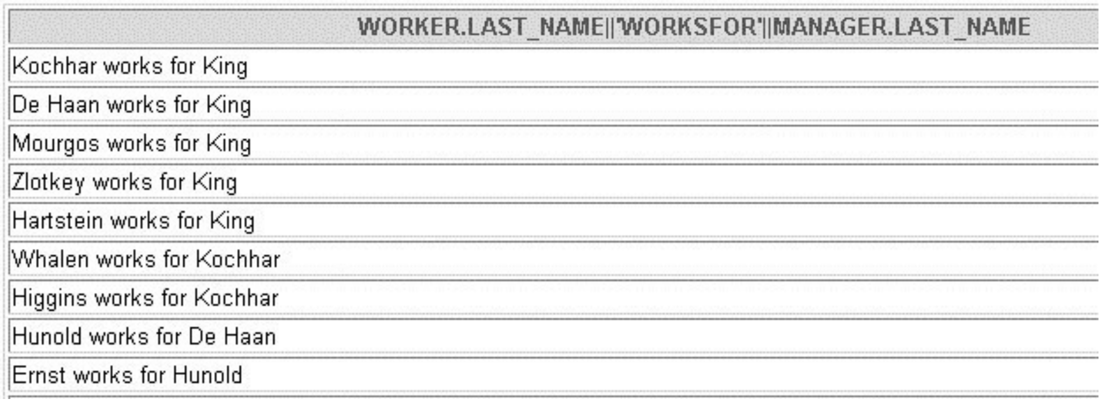
```
SELECT last_name, department_name dept_name
FROM employees, departments;
```

Retrieving Records with Equijoins

```
SELECT e.employee_id, e.last_name, e.department_id, d.department_id, d.location_id
FROM employees e , departments d
WHERE e.department_id = d.department_id;
```

Joining a Table to Itself

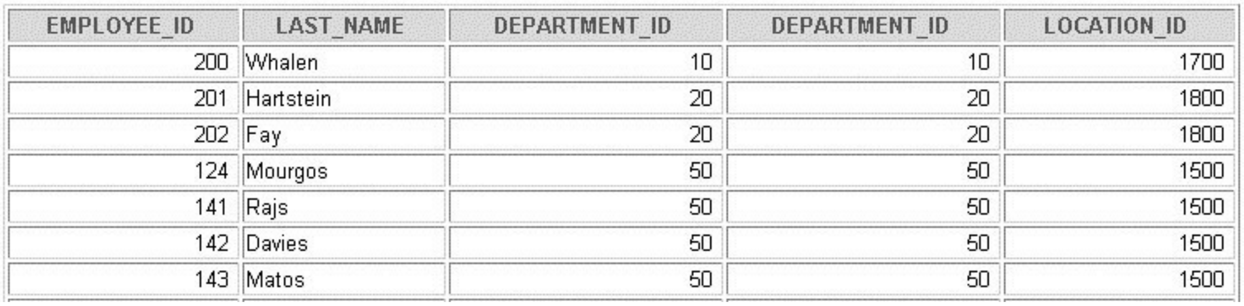
```
SELECT CONCAT(worker.last_name, ’ works for ’, manager.last_name )
FROM employees worker, employees manager
WHERE worker.manager_id = manager.employee_id ;
```



WORKER.LAST_NAME||'WORKSFOR'||MANAGER.LAST_NAME
Kochhar works for King
De Haan works for King
Mourgos works for King
Zlotkey works for King
Hartstein works for King
Whalen works for Kochhar
Higgins works for Kochhar
Hunold works for De Haan
Ernst works for Hunold

Creating Joins with the ON Clause

```
SELECT e.employee_id,e.last_name,e.department_id,d.department_id,
d.location_id
FROM employees e JOIN departments d
ON (e.department_id = d.department_id);
```



EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
200	Whalen	10	10	1700
201	Hartstein	20	20	1800
202	Fay	20	20	1800
124	Mourgos	50	50	1500
141	Rajs	50	50	1500
142	Davies	50	50	1500
143	Matos	50	50	1500

Activity 01:

Write a query to display the last name, department number, and department name for employees.

Activity 02:

Write a query to display the employee last name, department name, location ID, and city of employees who earn a commission.

CSE 311L(Database Management System)

LAB-Week 04 (Lecture 2)

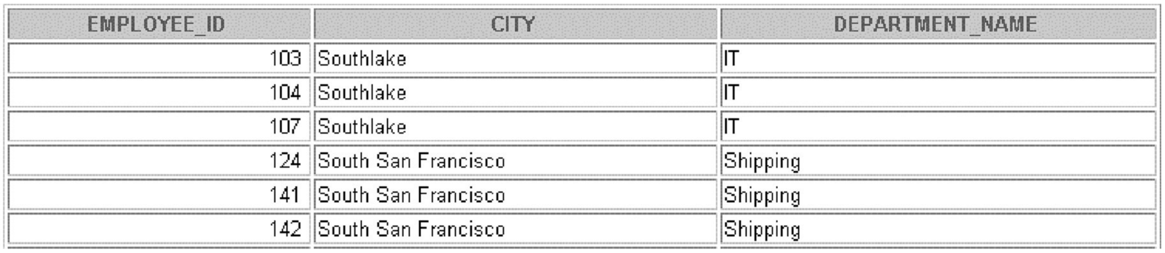
Topics:

After completing this lesson, you should be able to do:

- Creating Three-Way Joins with the ON Clause
- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN
- Additional Conditions

Creating Three-Way Joins with the ON Clause

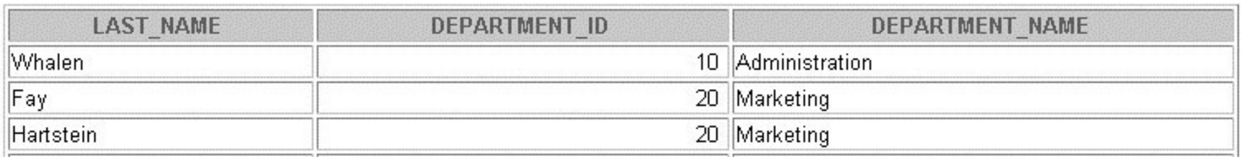
```
SELECT employee_id, city, department_name
FROM employees e
JOIN departments d
ON (d.department_id = e.department_id)
JOIN locations l
ON (d.location_id = l.location_id);
```



EMPLOYEE_ID	CITY	DEPARTMENT_NAME
103	Southlake	IT
104	Southlake	IT
107	Southlake	IT
124	South San Francisco	Shipping
141	South San Francisco	Shipping
142	South San Francisco	Shipping

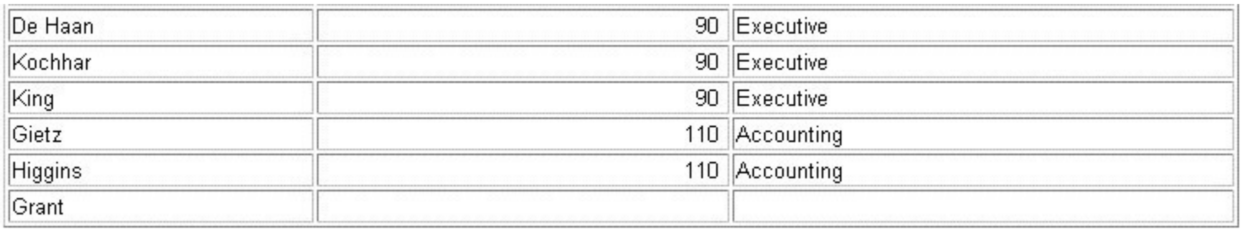
LEFT OUTER JOIN

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
LEFT JOIN departments d
ON (e.department_id = d.department_id) ;
```



LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing

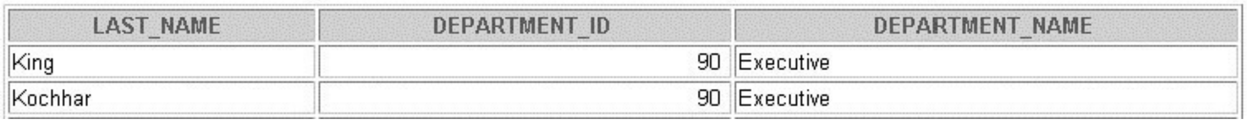
....



De Haan	90	Executive
Kochhar	90	Executive
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		

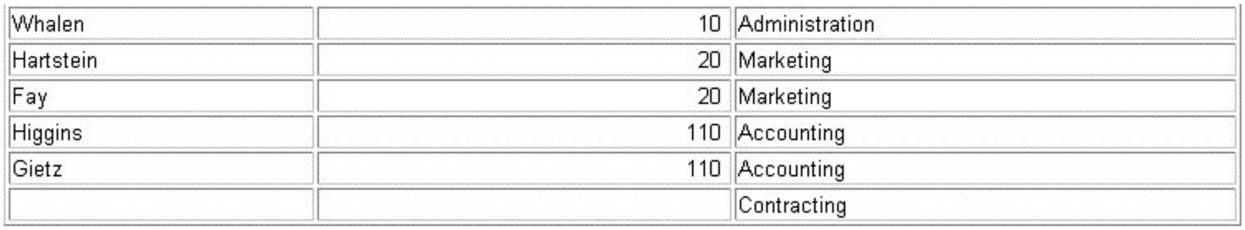
RIGHT OUTER JOIN

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
RIGHT JOIN departments d
ON (e.department_id = d.department_id) ;
```



LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
King	90	Executive
Kochhar	90	Executive

....



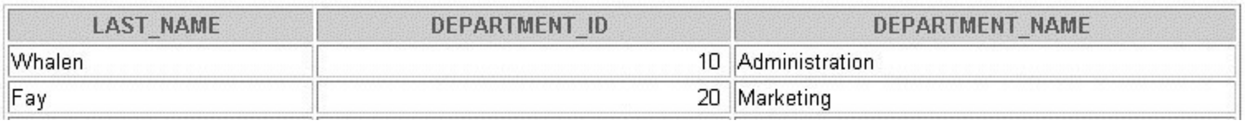
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Higgins	110	Accounting
Gietz	110	Accounting
		Contracting

FULL OUTER JOIN

- UNION/UNION ALL command combines the result set of two or more SELECT statements
- UNION includes only distinct values
- UNION ALL allows duplicate values

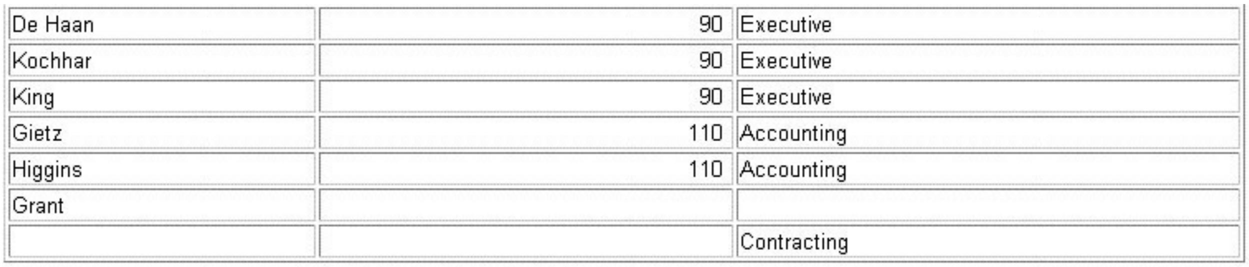
```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
LEFT JOIN departments d
ON (e.department_id = d.department_id)
UNION
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
RIGHT JOIN departments d
ON (e.department_id = d.department_id)
```

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
LEFT JOIN departments d
ON (e.department_id = d.department_id)
UNION ALL
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
RIGHT JOIN departments d
ON (e.department_id = d.department_id)
```



LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing

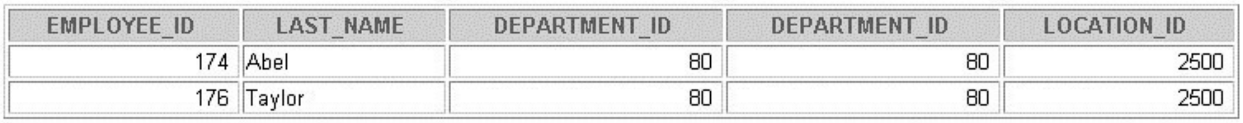
…



De Haan	90	Executive
Kochhar	90	Executive
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		
		Contracting

Additional Conditions

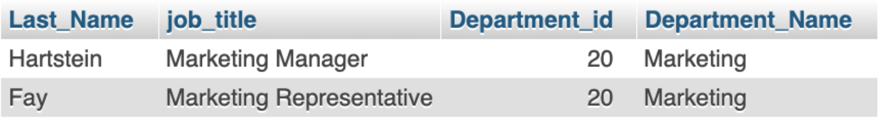
```
SELECT e.employee_id, e.last_name, e.department_id,  d.department_id,
d.location_id
FROM employees e
JOIN departments d
ON (e.department_id = d.department_id)
AND e.manager_id = 149 ;
```



EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
174	Abel	80	80	2500
176	Taylor	80	80	2500

Activity 01:

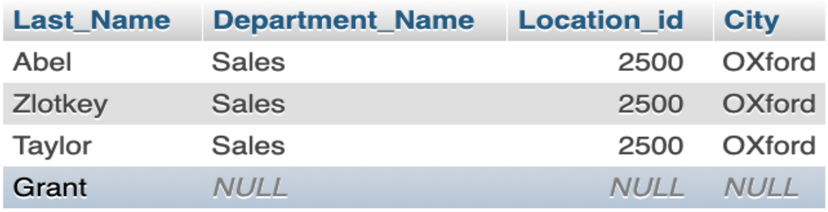
Write a query to display the last name, job title, department number, and department name for all employees who work in Toronto, and return the employees data also who doesn't have a Job ID.



Last_Name	job_title	Department_id	Department_Name
Hartstein	Marketing Manager	20	Marketing
Fay	Marketing Representative	20	Marketing

Activity 02:

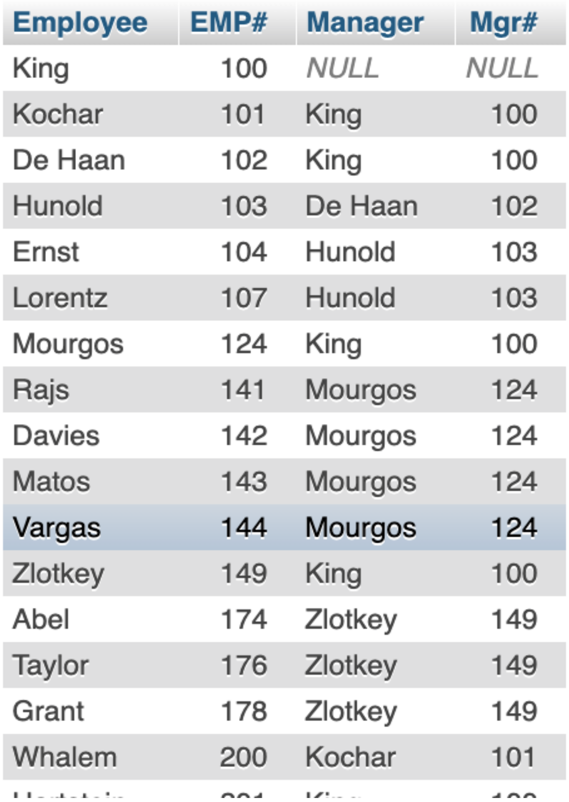
Write a query to display the employee last name, department name, location ID, and city of employees who earn a commission. Return the employees last name also even if there exist no data related to department and location. Sort data in descending order of salary and commissions.



Last_Name	Department_Name	Location_id	City
Abel	Sales	2500	OXford
Zlotkey	Sales	2500	OXford
Taylor	Sales	2500	OXford
Grant	*NULL*	*NULL*	*NULL*

Activity 03:

Display the employee last name and employee number of each and every employee along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively. (Output of first few rows are provided)



Employee	EMP#	Manager	Mgr#
King	100	*NULL*	*NULL*
Kochar	101	King	100
De Haan	102	King	100
Hunold	103	De Haan	102
Ernst	104	Hunold	103
Lorentz	107	Hunold	103
Mourgos	124	King	100
Rajs	141	Mourgos	124
Davies	142	Mourgos	124
Matos	143	Mourgos	124
Vargas	144	Mourgos	124
Zlotkey	149	King	100
Abel	174	Zlotkey	149
Taylor	176	Zlotkey	149
Grant	178	Zlotkey	149
Whalem	200	Kochar	101
Hartstein	201	King	100