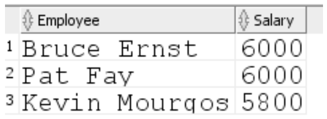
Chapter 4 Problem Set **ITEC 4200 Advanced Database**

Include the following comments at the top:
-- Your Name
-- Date
Save the file as ***LASTNAME_FIRSTNAME_Ch4_ProblemSet**.sql*

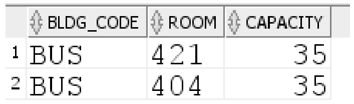
Make sure your answers are easy to read and in order. Put SELECT, FROM, and WHERE clauses on separate lines. Capitalize all keywords, use lowercase for tables and columns. Label all column headings as shown in the results.

Ch4p1 - Display the full name and salary for all employees whose salary is in the range of $5000 - $6000. Sort in descending order by salary.



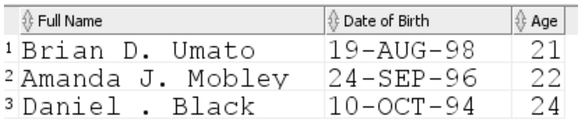
	Employee	Salary
1	Bruce Ernst	6000
2	Pat Fay	6000
3	Kevin Mourgos	5800

Ch4p2 - Using the LOCATION table, list the building code, room number and capacity for all locations that have capacity of 35 or more, and are on the 4th floor (the room number begins with '4'). Order by capacity.



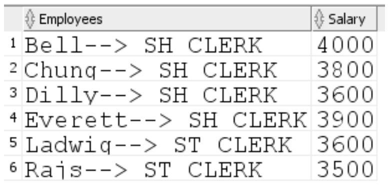
	BLDG_CODE	ROOM	CAPACITY
1	BUS	421	35
2	BUS	404	35

Ch4p3 - Display the name (concatenate the first name, middle initial, and last name), date of birth, and age for all students. Show the age with no decimal places, and only include those students who are 21 or older. Order by age, as shown below: (Hint: Use the TRUNC function. The ages may be different, depending on the date that the query is run.)



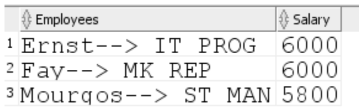
	Full Name	Date of Birth	Age
1	Brian D. Umato	19-AUG-98	21
2	Amanda J. Mobley	24-SEP-96	22
3	Daniel . Black	10-OCT-94	24

Ch4p4 - Write a script file to display the employee name and salary for all employees whose salary is between a given low and high salary. Prompt the user for the two salaries. Concatenate the name and job together, separated by a space and a comma, and order by the employee's last name. Here are two sample runs. The first run uses the range 3500 - 4000:



	Employees	Salary
1	Bell--> SH CLERK	4000
2	Chung--> SH CLERK	3800
3	Dilly--> SH CLERK	3600
4	Everett--> SH CLERK	3900
5	Ladwig--> ST CLERK	3600
6	Rajs--> ST CLERK	3500

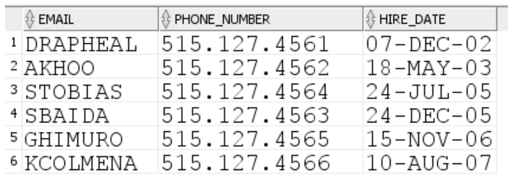
This run uses the range 5000 – 6000:



	Employees	Salary
1	Ernst--> IT PROG	6000
2	Fay--> MK REP	6000
3	Mourgos--> ST MAN	5800

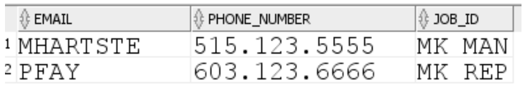
Ch4p5 - Using the EMPLOYEES table, display the email, phone number, and then one other column (the name of the column is entered by the user). This same column is then also used in the ORDER BY clause to order the resulting rows. Restrict the rows to those that are equal to a certain department id (this number is also entered in by the user.) Use the double ampersand for the column substitution variable to insure that the user is prompted to enter the column name ONLY ONCE, and that it is "cleared out" after the run (UNDEFINE).

Here is a run with hire_date and department 30:



	EMAIL	PHONE_NUMBER	HIRE_DATE
1	DRAPHEAL	515.127.4561	07-DEC-02
2	AKHOO	515.127.4562	18-MAY-03
3	STOBIAS	515.127.4564	24-JUL-05
4	SBAIDA	515.127.4563	24-DEC-05
5	GHIMURO	515.127.4565	15-NOV-06
6	KCOLMENA	515.127.4566	10-AUG-07

Here is another run with job_id and department 20:



	EMAIL	PHONE_NUMBER	JOB_ID
1	MHARTSTE	515.123.5555	MK MAN
2	PFAY	603.123.6666	MK REP

Using your **Semester project schema**, create the follow new problems using the features introduced in this chapter. Create a new SQL file for saving all of your semester project queries, as we go from one chapter to the next. Upload this file to D2L Assignment | Semester Project SQL Queries

Here is a list of requirements for each query that you create for your semester project:

- All queries must be saved in <u>one</u> SQL script
- Include a comment that identifies the query as Query 1, Query 2, etc.
- Include a comment as the next line that explains in plain English what the query does
- All queries must have ***an ORDER BY***
- All columns ***must have an alias***
- The results of the query should be nicely formatted and one row fits on one line of output.

QUERY 1 must contain:

- AND or OR for a compound condition (not BETWEEN)

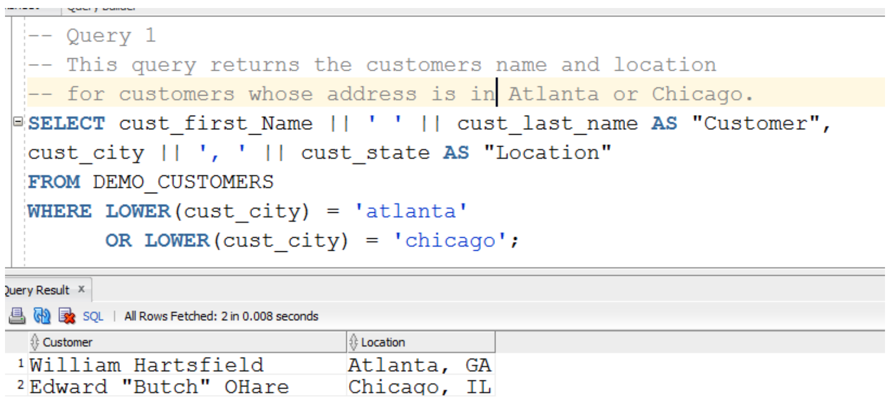
QUERY 2 must contain:

- A value entered by a substitution variable
- LIKE operator

QUERY 3 must contain:

- An expression that contains IN or BETWEEN
- The FETCH option to limit rows

Here is how you would do Query1, for example:



```
-- Query 1
-- This query returns the customers name and location
-- for customers whose address is in Atlanta or Chicago.
SELECT cust_first_Name || ' ' || cust_last_name AS "Customer",
cust_city || ', ' || cust_state AS "Location"
FROM DEMO_CUSTOMERS
WHERE LOWER(cust_city) = 'atlanta'
      OR LOWER(cust_city) = 'chicago';
```

```
  Customer                  Location
1 William Hartsfield        Atlanta, GA
2 Edward "Butch" OHare      Chicago, IL
```

Chapter 4 checklist:

(1) Upload SQL script for solutions to ch4p1 – ch4p5 to D2L | Assignments | Chapter 4 Problem Set
(2) Upload SQL script for Queries 1 – 3 to D2L | Assignments | Semester Project SQL Queries
(3) Printout of Semester Project Queries 1 – 3 (each printed as shown above)