a) List the last name, first name, and phone number of students who do not have a phone

number. Display '212-555-1212' for the phone number.

**SELECT first\_name||' '|| last\_name name, phone oldphone, NVL(phone, '212-555-1212') newphone FROM student WHERE phone IS NULL**

b) For course numbers 430 and greater, show the course cost. Add another column

reflecting a discount of 10% off the cost and substitute any NULL values in the COST

column with the number 1000. The result should look similar to the following output.

COURSE\_NO COST NEW

-------------- ------- ------

430 1195 1075.5

450 900

2 rows selected.

discount calculation. Otherwise, the calculation yields a NULL..

**SELECT course\_no, cost,**

**NVL(cost,1000)\*0.9 new**

**FROM course**

**WHERE course\_no >= 430**

c) Write the query to accomplish the following output using the NVL2 function in the

column 'Get this result'.

ID NAME PHONE Get this result

------------------ ------------ -----------------

112 Thomas Thomas 201-555-5555 Phone# exists.

111 Peggy Noviello No phone# exists.

2 rows selected

**SELECT student\_id id, first\_name||' '|| last\_name name,**

**phone,**

**NVL2(phone, 'Phone# exists.', 'No phone# exists.')**

**"Get this result"**

**FROM student**

**WHERE student\_id IN (111, 112)**

a) Rewrite the query from Exercise 3.3.1 c) using the DECODE function instead.

**SELECT student\_id, first\_name||' '|| last\_name name,**

**phone,**

**DECODE(phone, NULL, 'No phone# exists.', 'Phone# exists.')**

**"Get this result"**

**FROM student**

**WHERE student\_id IN (111, 112)**

b) For course numbers 20, 120, 122, and 132, display the description, course number, and

prerequisite course number. If the prerequisite is course number 120, display 200; if the

prerequisite is 130, display 'N/A'. For courses with no prerequisites, display 'None'.

Otherwise, list the current prerequisite. The result should look like the one listed below.

COURSE\_NO DESCRIPTION ORIGINAL NEW

--------- ------------------------------ -------- ----

132 Basics of Unix Admin 130 N/A

122 Intermediate Java Programming 120 200

120 Intro to Java Programming 80 80

20 Intro to Computers None

4 rows selected.

**SELECT course\_no, description, prerequisite "ORIGINAL",**

**CASE WHEN prerequisite = 120 THEN '200'**

**WHEN prerequisite = 130 THEN 'N/A'**

**WHEN prerequisite IS NULL THEN 'None'**

**ELSE TO\_CHAR(prerequisite)**

**END "NEW"**

**FROM course**

**WHERE course\_no IN (20, 120, 122, 132)**

**ORDER BY course\_no DESC**

c) Display the student ID, zip code, and phone number for students with student IDs 145,

150, or 325. For those students living in the 212 area code and in zip code 10048,

display 'North Campus'. List students living in the 212 area code but in a different zip

code as 'West Campus'. Display students outside the 212 area code as 'Off Campus'.

The result should look like the following output. Hint: The solution to this query requires

nested DECODE functions or nested CASE expressions.

STUDENT\_ID ZIP PHONE LOC

---------- ----- --------------- ------------

145 10048 212-555-5555 North Campus

150 11787 718-555-5555 Off Campus

325 10954 212-555-5555 West Campus

3 rows selected.

**SELECT student\_id, zip, phone,**

**CASE WHEN SUBSTR(phone, 1, 3) = '212' THEN**

**CASE WHEN zip = '10048' THEN 'North Campus'**

**ELSE 'West Campus'**

**END**

**ELSE 'Off Campus'**

**END loc**

**FROM student**

**WHERE student\_id IN (150, 145, 325)**

d) Display all the distinct salutations used in the INSTRUCTOR table. Order them alphabetically except for female salutations, which should be listed first. Hint: Use the DECODE function or CASE expression in the ORDER BY clause.

**SELECT DISTINCT salutation**

**FROM instructor**

**ORDER BY CASE salutation WHEN 'Ms' THEN '1'**

**WHEN 'Mrs' THEN '1'**

**WHEN 'Miss' THEN '1'**

**ELSE salutation**

**END**