

# Contenido

Capítulo 2 .....	2
<b>Chapter 3.....</b>	<b>15</b>
<b>Character, Number and Miscellaneous Functions.....</b>	<b>15</b>
<b>3.1.2. Concatenate Strings.....</b>	<b>19</b>
<b>Lab 3.2 3.2.1. Use Number Functions and Perform Mathematical .....</b>	<b>20</b>
<b>Computations.....</b>	<b>20</b>
<b>3.3.1. Apply Substitution Functions and Other Miscellaneous Functions .....</b>	<b>21</b>
3.3.2. Utilize the Power of the DECODE Function and the CASE Expression .....	22
<b>Chapter 4. Date and Conversion Functions.....</b>	<b>24</b>
Chapter 5 Equijoins .....	27
Lab 6.1 The Two-Table Join.....	27
Lab 6.1 Exercises.....	29
Lab 6.2 Exercises .....	32
6.2.1. Join Three or More Tables.....	32
Chapter 7. Subqueries.....	34
Lab 7.1 Exercises.....	34
7.2 exercises .....	43
Lab 7.4 Exercise .....	45
Chapter 8. Set Operators.....	48
Lab 8.1 Exercises.....	49
Lab 8.2 Exercises.....	52
Chapter 9. Complex Joins.....	54
Lab 9.1 Exercises .....	56

## CAPITULO 2

rem [Rischert, 2004,148]

```
SELECT description
FROM grade_type
WHERE description >= 'Midterm'
AND description <= 'Project';
```

```
SQL> SELECT description
      2 FROM grade_type
      3 WHERE description >= 'Midterm'
      4 AND description <= 'Project';
```

DESCRIPTION

-----  
Midterm  
Participation  
Project

SQL>

e) Execute the following query and determine how many rows the query returns.  
SELECT last\_name, student\_id FROM student WHERE ROWNUM <= 10

LAST_NAME	*	STUDENT_ID
-----	*	-----
Crocitto	*	102
Landry	*	103
Enison	*	104
Moskowitz	*	105
Olvsade	*	106
Mierzwa	*	107
Sethi	*	108
Walter	*	109
Martin	*	110
Noviello	*	111

10 filas seleccionadas.

SQL>

SELECT ROWNUM, last\_name, student\_id FROM student WHERE ROWNUM <= 10

```
SQL> SELECT ROWNUM, last_name, student_id
2   FROM student
3   WHERE ROWNUM <= 10;
```

ROWNUM	*	LAST_NAME	*	STUDENT_ID
1	*	Crocitto	*	102
2	*	Landry	*	103
3	*	Enison	*	104
4	*	Moskowitz	*	105
5	*	Olvsade	*	106
6	*	Mierzwa	*	107
7	*	Sethi	*	108
8	*	Walter	*	109
9	*	Martin	*	110
10	*	Noviello	*	111

10 filas seleccionadas.

SQL>

```
SQL> SELECT course_no, description
2   FROM course
3   WHERE prerequisite IS NULL
4   ORDER BY description;
```

COURSE_NO	*	DESCRIPTION
10	*	DP Overview
20	*	Intro to Computers
146	*	Java for C/C++ Programmers
310	*	Operating Systems

SQL>

```
SQL> SELECT course_no, description
2   FROM course
3   WHERE prerequisite IS NULL
4   ORDER BY description DESC;
```

COURSE_NO	*	DESCRIPTION
310	*	Operating Systems
146	*	Java for C/C++ Programmers
20	*	Intro to Computers
10	*	DP Overview

SQL>

```
SQL> SELECT course_no, description
2   FROM course
3   WHERE prerequisite IS NULL
4   ORDER BY 2 DESC;
```

COURSE_NO	*	DESCRIPTION
310	*	Operating Systems
146	*	Java for C/C++ Programmers
20	*	Intro to Computers
10	*	DP Overview

SQL>

```
SQL> SELECT first_name first,
2   first_name "First Name",
3   first_name AS "First"
4   FROM student
5   WHERE zip = '10025';
```

FIRST	*	First Name	*	First
Nicole	*	Nicole	*	Nicole
Jerry	*	Jerry	*	Jerry
Frank	*	Frank	*	Frank

SQL>

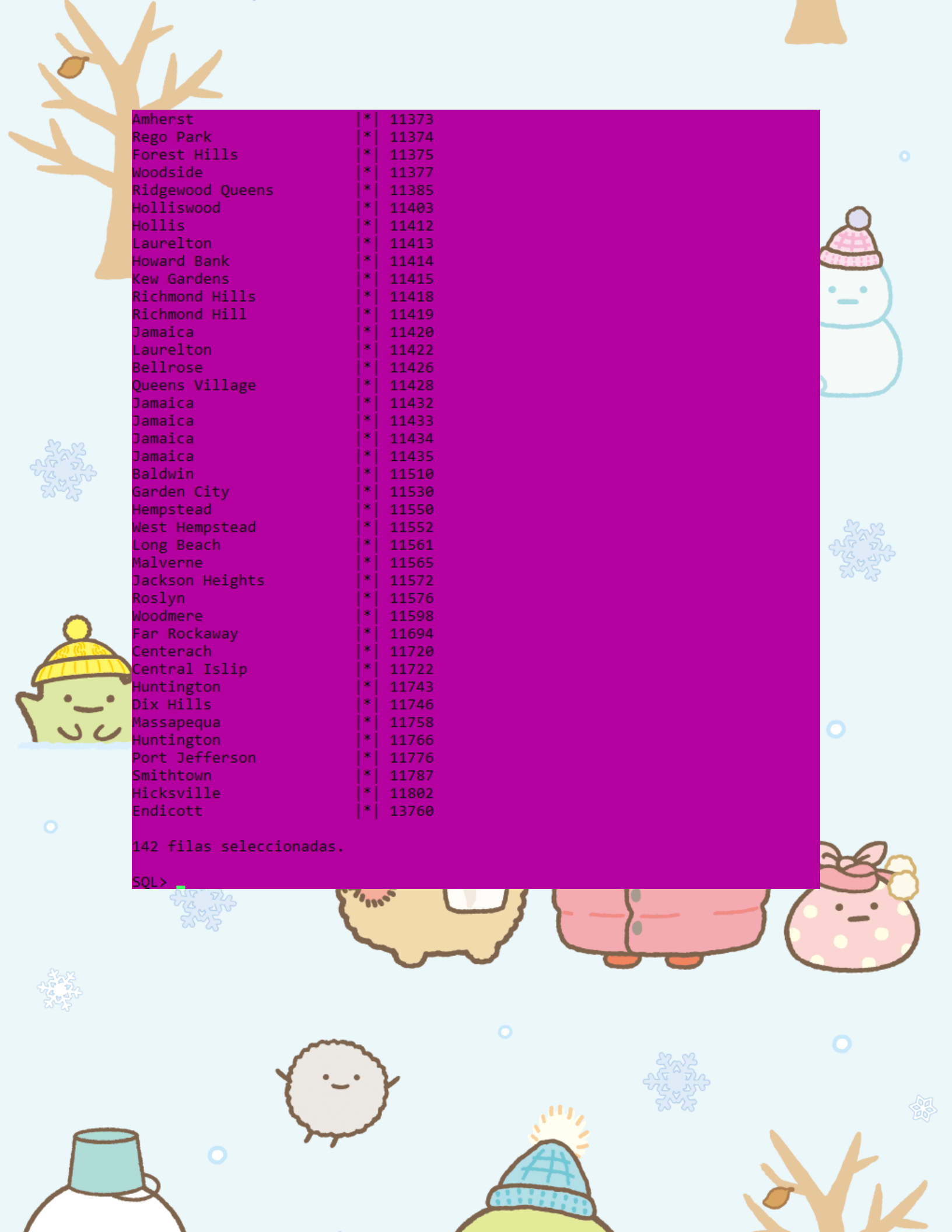
```
SQL> SELECT first_name first, first_name "First Name",
2   first_name AS "First"
3   FROM student
4   WHERE zip = '10025'
5   ORDER BY "First Name";
```

FIRST	*	First Name	*	First
Frank	*	Frank	*	Frank
Jerry	*	Jerry	*	Jerry
Nicole	*	Nicole	*	Nicole

SQL>

```
SQL> 5 /;
SQL>
SQL> SQL> SELECT DISTINCT first_name, last_name
SP2-0734: inicio "SQL> SELEC..." de comando desconocido - resto de la l nea ignorado.
SQL> 2 FROM student
SQL> 3 WHERE zip = '10025'
SQL> 4 ORDER BY student_id;
SQL> SELECT student_id, last_name
2 FROM student
3 ORDER BY last_name;
```

STUDENT_ID	LAST_NAME
119	Abdou
399	Abdou
257	Abid
190	Affinito
309	Airall
120	Alexander
385	Allende
198	Allende
312	Allende
356	Ancean
162	Andrew
371	Anglin
334	Annina
206	annunziato
322	Archer
302	Anchor
172	Arias
127	Aung
317	Austin
358	Avia
202	Axch
140	Baltazar
170	Balterzar
252	Barogh
388	Bathmanapan
160	Beitler
271	Benitez
143	Biers
253	Boremman
195	Bose
175	Boyce



Amherst	*	11373
Rego Park	*	11374
Forest Hills	*	11375
Woodside	*	11377
Ridgewood Queens	*	11385
Holliswood	*	11403
Hollis	*	11412
Laurelton	*	11413
Howard Bank	*	11414
Kew Gardens	*	11415
Richmond Hills	*	11418
Richmond Hill	*	11419
Jamaica	*	11420
Laurelton	*	11422
Bellrose	*	11426
Queens Village	*	11428
Jamaica	*	11432
Jamaica	*	11433
Jamaica	*	11434
Jamaica	*	11435
Baldwin	*	11510
Garden City	*	11530
Hempstead	*	11550
West Hempstead	*	11552
Long Beach	*	11561
Malverne	*	11565
Jackson Heights	*	11572
Roslyn	*	11576
Woodmere	*	11598
Far Rockaway	*	11694
Centerach	*	11720
Central Islip	*	11722
Huntington	*	11743
Dix Hills	*	11746
Massapequa	*	11758
Huntington	*	11766
Port Jefferson	*	11776
Smithtown	*	11787
Hicksville	*	11802
Endicott	*	13760

142 filas seleccionadas.

SQL>

```
SQL> SELECT description "Descr", prerequisite
2   FROM course
3   WHERE prerequisite IS NOT NULL
4   ORDER BY "Descr";
```

Descr	*	PREREQUISITE
Advanced Java Programming	*	122
Advanced Unix Admin	*	132
Basics of Unix Admin	*	130
Database Design	*	420
Database System Principles	*	25
DB Programming in Java	*	350
GUI Programming	*	20
Hands-On Windows	*	20
Intermediate Java Programming	*	120
Internet Protocols	*	310
Intro to Internet	*	10
Intro to Java Programming	*	80
Intro to Programming	*	140
Intro to SQL	*	20
Intro to the Basic Language	*	25
Intro to Unix	*	310
JDeveloper	*	122
JDeveloper Lab	*	125
JDeveloper Techniques	*	350
Network Administration	*	130
Oracle Tools	*	220
PL/SQL Programming	*	80
Project Management	*	20
Structured Analysis	*	20
Structured Programming Techniques	*	204
Unix Tips and Techniques	*	134

26 filas seleccionadas.

```
SQL> SELECT salutation "Sal", first_name "First Name",
2   last_name "Last Name"
3   FROM student
4   WHERE last_name = 'Grant'
5   ORDER BY "Sal" DESC, "First Name" ASC;
```

Sal	*	First Name	*	Last Name
Ms.	*	Eilene	*	Grant
Ms.	*	Verona	*	Grant
Mr.	*	Omaira	*	Grant
Mr.	*	Scott	*	Grant

SQL>



SQL&gt;



```
SQL> DESCR dual;
Nombre                               Nulo? Tipo
-----
DUMMY                                VARCHA2(1)

SQL>
```

```
SQL> SELECT LTRIM('0001234500', '0') left,
2      RTRIM('0001234500', '0') right,
3      LTRIM(RTRIM('0001234500', '0'), '0') both
4      FROM dual;
```

```
LEFT      |*| RIGHT   |*| BOTH
-----
1234500   |*| 00012345 |*| 12345

SQL>
```

```
SQL> SELECT last_name,
2      SUBSTR(last_name, 1, 5),
3      SUBSTR(last_name, 6)
4      FROM student;
```

LAST_NAME	* SUBSTR(LAST_NAME,1,5)	* SUBSTR(LAST_NAME,6)
Crocitto	* Croci	* tto
Landry	* Landr	* y
Enison	* Eniso	* n
Moskowitz	* Mosko	* witz
Olvsa	* Olvsa	* de
Mierzwa	* Mierz	* wa
Sethi	* Sethi	* s/Datos
Walter	* Walte	* r
Martin	* Marti	* n
Noviello	* Novie	* llo
Thomas	* Thoma	* s
Kulina	* Kulin	* a
Laporte	* Lapor	* te
Kuehn	* Kuehn	* s/Datos
Lopez	* Lopez	* s/Datos
Abdou	* Abdou	* s/Datos
Alexander	* Alexa	* nder
Pineda	* Pined	* a
Lippen	* Lippe	* n
Radicola	* Radic	* ola
Wicelinski	* Wicel	* inski
Aung	* Aung	* s/Datos
Runyan	* Runya	* n
Grant	* Grant	* s/Datos
Oates	* Oates	* s/Datos
Reed	* Reed	* s/Datos
Torres	* Torre	* s
Masser	* Masse	* r
Lasseter	* Lasse	* ter
Miller	* Mille	* r
Smith	* Smith	* s/Datos
Garshman	* Garsh	* man
Baltazar	* Balta	* zar
Boyd	* Boyd	* s/Datos
Waldman	* Waldm	* an

```
SQL> SELECT LENGTH('Hello there')
2      FROM dual;
```

```
LENGTH('HELLOTHERE')
-----
11

SQL>
```

```
SQL> SELECT first_name, last_name
2   FROM student
3   WHERE INSTR(first_name, '.') > 0
4   ORDER BY LENGTH(last_name);
```

FIRST_NAME	LAST_NAME
Suzanne M.	Abid
D.	Orent
Austin V.	Cadet
J.	Dalvi
M.	Diokno
D.	Dewitt
James E.	Norman
J.	Landry
A.	Tucker
O.	Garnes
Rafael A.	Torres
J.	Segall
E.A.	Torres
Lynwood A.	Thorton
John T.	Beitler
R.	Sprouse
Thomas E.	La Blank
V.	Greenberg
P.	Balterzar
V.	Saliternan
Z.A.	Scrittonale

21 filas seleccionadas.

```
SQL>
```

```
SQL> SELECT first_name,
2   SUBSTR(first_name, INSTR(first_name, '.')-1) mi,
3   SUBSTR(first_name, 1, INSTR(first_name, '.')-2) first
4   FROM student
5   WHERE INSTR(first_name, '.') >= 3;
```

FIRST_NAME	MI
John T.	T.
John	
Rafael A.	A.
Rafael	
James E.	E.
James	
Suzanne M.	M.
Suzanne	
Thomas E.	E.
Thomas	
Austin V.	V.
Austin	
Lynwood A.	A.
Lynwood	

7 filas seleccionadas.

```
SQL>
```

```
SQL> SELECT city||', '||state||' '||zip
2 FROM zipcode;
```

```
CITY||', '||STATE||' '||ZIP
```

```
-----
Santurce, PR 00914
North Adams, MA 01247
Dorchester, MA 02124
Tufts Univ. Bedford, MA 02155
Weymouth, MA 02189
Sandwich, MA 02563
Ansonia, CT 06401
Middlefield, CT 06455
Oxford, CT 06483
New Haven, CT 06520
Bridgeport, CT 06605
Woodbury, CT 06798
Georgetown, WV 06820
Greenwich, CT 06830
Norwalk, CT 06850
Norwalk, CT 06851
Rowayton, CT 06853
Old Greenwich, CT 06870
Ridgefield, CT 06877
Westport, CT 06880
Weston, CT 06883
Wilton, CT 06897
Stamford, CT 06902
Stamford, CT 06903
Stamford, CT 06905
Stamford, CT 06907
Bayonne, NJ 07002
Bloomfiel, NJ 07003
North Caldwell, NJ 07006
Carteret, NJ 07008
Cedar Grove, NJ 07009
Cliffside Park, NJ 07010
Clifton, NJ 07011
```

```
SQL> SELECT student_id, last_name
2 FROM student
3 WHERE SOUNDEX(last_name) = SOUNDEX('MARTIN');
```

```
STUDENT_ID|*| LAST_NAME
-----|*|-----
110|*| Martin
393|*| Martin
324|*| Marten
```

```
SQL>
```

```
SQL> SELECT description "Description",
2 INITCAP(description) "Initcap Description"
3 FROM course
4 WHERE description LIKE '%SQL%';
```

```
Description|*| Initcap Description
-----|*|-----
Intro to SQL|*| Intro To Sql
PL/SQL Programming|*| Pl/Sql Programming
```

```
SQL>
```

```
SQL> SELECT last_name
       2   FROM instructor
       3  WHERE LENGTH(last_name) >= 6;
```

LAST\_NAME

-----  
Wojick  
Schorin  
Pertez  
Morris  
Smythe  
Frantzen  
Willig

7 filas seleccionadas.

SQL>

```
SQL> SELECT SUBSTR('12345', 3),
       2   SUBSTR('12345', 3, 2),
       3   SUBSTR('12345', -4, 3)
       4   FROM dual;
```

SUB		SU		SUB
---		--		---
345		34		234

SQL>

ZIP		LTRIM		RTRIM
-----		-----		-----
11418		11418		11418
11419		11419		11419
11420		11420		11420
11422		11422		11422
11426		11426		11426
11428		11428		11428
11432		11432		11432
11433		11433		11433
11434		11434		1143
11435		11435		11435
11510		11510		11510
11530		11530		11530
11550		11550		11550
11552		11552		11552
11561		11561		11561
11565		11565		11565
11572		11572		11572
11576		11576		11576
11598		11598		11598
11694		11694		1169
11717		11717		11717
11720		11720		11720
11722		11722		11722
11743		11743		11743
11746		11746		11746
11758		11758		11758
11766		11766		11766
11776		11776		11776
11787		11787		11787
11802		11802		11802
13760		13760		13760
30342		30342		30342
33431		33431		33431
43224		43224		4322
48104		48104		4810

227 filas seleccionadas.

SQL>

```
SQL> SELECT TRIM('01' FROM '01230145601')
       2   FROM dual;
SELECT TRIM('01' FROM '01230145601')
       *
```

ERROR en línea 1:

ORA-30001: el recorte definido sólo debe contener un carácter

SQL>

```
SQL> SELECT TRANSLATE('555-1212', '0123456789',
2  '#####')
3  FROM dual;
```

```
TRANSLAT
```

```
-----
###-####
```

```
SQL> 
```

```
SQL> SELECT (
2  LENGTH('Fred fed Ted bread, and Ted fed Fred bread.') -
3  LENGTH(REPLACE(
4  'Fred fed Ted bread, and Ted fed Fred bread.',
5  'ed', NULL))
6  ) /2 AS occurr
7  FROM dual;
```

```
OCCURR
```

```
-----
6
```

```
SQL> 
```

```
SQL> SELECT 10.245, ROUND(10.245, 1), ROUND(10.245, -1)
2  FROM dual;
```

```
10.245|*| ROUND(10.245,1)|*| ROUND(10.245,-1)
-----*|-----*|-----
10.245|*|          10.2|*|          10
```

```
SQL> 
```

```
SQL> SELECT ROUND(120.09, -2), ROUND(1444.44, -3)
2  FROM dual;
```

```
ROUND(120.09,-2)|*| ROUND(1444.44,-3)
-----*|-----
100|*|          1000
```

```
SQL> 
```

```
SQL> SELECT DISTINCT cost, cost*1.75, ROUND(cost*1.75)
2  FROM course;
```

```
COST|*| COST*1.75|*| ROUND(COST*1.75)
-----*|-----*|-----
$1,195.00|*| 2091.25|*| 2091
s/Datos|*| s/Datos|*| s/Datos
$1,095.00|*| 1916.25|*| 1916
$1,595.00|*| 2791.25|*| 2791
```

```
SQL> 
```

```
SQL> SELECT DISTINCT numeric_grade, ROUND(numeric_grade / 2)
2 FROM grade;
```

NUMERIC_GRADE	ROUND(NUMERIC_GRADE/2)
90	45
91	46
96	48
80	40
81	41
84	42
92	46
74	37
72	36
93	47
98	49
99	50
97	49
95	48
78	39
71	36
70	35
75	38
73	37
87	44
86	43
88	44
85	43
76	38
83	42
82	41
77	39
89	45
79	40
94	47

30 filas seleccionadas.

SQL>

```
SQL> SELECT first_name||' '|| last_name name,
2 phone oldphone,
3 NVL(phone, '212-555-1212') newphone
4 FROM student
5 WHERE phone IS NULL;
```

NAME	OLDPHONE	NEWPHONE
Peggy Noviello	s/Datos	212-555-1212

SQL>

```
SQL> SELECT first_name||' '|| last_name name,
2 phone oldphone,
3 NVL(phone, '212-555-1212') newphone
4 FROM student
5 WHERE NVL(phone, 'NONE') = 'NONE';
```

NAME	OLDPHONE	NEWPHONE
Peggy Noviello	s/Datos	212-555-1212

SQL>

```
SQL> SELECT course_no, cost,
2 NVL(cost,1000)*0.9 new
3 FROM course
4 WHERE course_no >= 430;
```

COURSE_NO	COST	NEW
430	\$1,195.00	1075.5
450	s/Datos	900

SQL>

## Chapter 3.

### Character, Number and Miscellaneous Functions

#### Lab 3.1 Exercises

##### 3.1.1. Use a Character Function in a SQL Statement

a) Execute the following SQL statement. Based on the result, what is the purpose of the INITCAP function?

```
SELECT description "Description",  
       INITCAP(description) "Initcap Description"  
FROM course  
WHERE description LIKE '%SQL%'
```

```
SQL> SELECT description "Description",  
2  INITCAP(description) "Initcap Description"  
3  FROM course  
4  WHERE description LIKE '%SQL%';  
  
ninguna fila seleccionada
```

Con la primer sentencia podemos observar que al momento de ingresarla no funciona.

b) Write the question answered by the following SQL statement.

```
SELECT last_name  
FROM instructor  
WHERE LENGTH(last_name) >= 6
```

```
SQL> SELECT last_name  
2  FROM instructor  
3  WHERE LENGTH(last_name) >= 6;  
  
LAST_NAME  
-----  
Willig
```

La sentencia nos devuelve los apellidos que tengan seis o mas letras.



c) Describe the result of the following SQL statement. Pay particular attention to the egative number parameter.

```
SELECT SUBSTR('12345', 3),  
SUBSTR('12345', 3, 2),  
SUBSTR('12345', -4, 3)  
FROM dual
```

```
SQL> SELECT SUBSTR('12345', 3),  
2 SUBSTR('12345', 3, 2),  
3 SUBSTR('12345', -4, 3)  
4 FROM dual;
```

```
SUB SU SUB  
----  
345 34 234
```

En la Columba SUBSTR toma los números a partir de la tercera posición, en la segunda Columba, toma los primeros dos valores a partir del tercero y en la tercer columna como es un número negativo empieza desde el final de la cadena, por lo tanto comienza a contar a partir del dos, tomando solamente tres caracteres.

d) Based on the result of the following SQL statement, describe the purpose of the LTRIM and RTRIM functions.

```
SELECT zip, LTRIM(zip, '0'), RTRIM(ZIP, '4')  
FROM zipcode  
ORDER BY zip
```

```
SQL> SELECT zip, LTRIM(zip, '0'), RTRIM(ZIP, '4')  
2 FROM zipcode  
3 ORDER BY zip;  
  
ninguna fila seleccionada
```

e) What do you observe when you execute the next statement? How would you change the statement to achieve the desired result?

```
SELECT TRIM('01' FROM '01230145601')  
  
FROM dual
```

```
SQL> SELECT TRIM('01' FROM '01230145601')  
2 FROM dual;  
SELECT TRIM('01' FROM '01230145601')  
*  
ERROR en línea 1:  
ORA-30001: el recorte definido sólo debe contener un carácter
```

```
SQL> SELECT LTRIM('01230145601', '01') left,  
2 RTRIM('01230145601', '01') right,  
3 RTRIM(LTRIM('01230145601', '01'), '01') both,  
4 REPLACE('01230145601', '01') replace  
5 FROM dual;
```

LEFT	RIGHT	BOTH	REPLA
230145601	012301456	2301456	23456

f) What is the result of the following statement?

```
SELECT TRANSLATE('555-1212', '0123456789',  
                '#####')  
  
FROM dual
```

```
SQL> SELECT TRANSLATE('555-1212', '0123456789',  
2 '#####')  
3 FROM dual;
```

```
TRANSLAT  
-----  
###-####
```

g) Write the SQL statement to retrieve those students that have a last name with the lowercase letter 'o' occurring three or more times.

```
SQL> SELECT student_id, last_name  
2 FROM student  
3 WHERE INSTR(last_name, 'o', 1, 3) > 0;  
  
ninguna fila seleccionada
```

h) The following statement determines how many times the string 'ed' occurs in the phrase 'Fred fed Ted bread, and Ted fed Fred bread.' Explain how this is accomplished.

```
SELECT (  
    LENGTH('Fred fed Ted bread, and Ted fed Fred bread.') -  
    LENGTH(REPLACE('Fred fed Ted bread, and Ted fed Fred bread.',  
        'ed', NULL))  
    ) /2 AS occurr  
  
FROM dual  
  
OCCURR  
-----  
6  
1 row selected.
```

```
SQL> SELECT (  
2  LENGTH('Fred fed Ted bread, and Ted fed Fred bread.') -  
3  LENGTH(REPLACE(  
4  'Fred fed Ted bread, and Ted fed Fred bread.',  
5  'ed', NULL))  
6  ) /2 AS occurr  
7  FROM dual;  
  
OCCURR  
-----  
6
```

### 3.1.2. Concatenate Strings

- a) Write a SELECT statement that returns each instructor's last name, followed by a comma and a space, followed by the instructor's first name, all in a single column in the result set.

```
SQL> SELECT last_name||', '||first_name  
2 FROM instructor;
```

```
LAST_NAME||', '||FIRST_NAME
```

```
-----  
Willig, Irene
```

- b) Using functions in the SELECT list, WHERE, and ORDER BY clauses, write the SELECT statement that returns course numbers and course descriptions from the COURSE table and looks like the following result set:

```
Description
```

```
-----
```

```
204.....Intro to SQL
```

```
130.....Intro to Unix
```

```
230.....Intro to Internet
```

```
20.....Intro to Computers
```

```
25.....Intro to Programming
```

```
120.....Intro to Java Programming
```

```
240.....Intro to the Basic Language
```

```
7 rows selected.
```

```
SQL> SELECT RPAD(course_no, 10, '.')||description  
2 AS "Description"  
3 FROM course  
4 WHERE INSTR(description, 'Intro') = 1  
5 ORDER BY LENGTH(description);
```

```
ninguna fila seleccionada
```

## Lab 3.2 3.2.1. Use Number Functions and Perform Mathematical Computations.

a) Describe the effect of the negative precision as a parameter of the ROUND function in

the following SQL statement.

```
SELECT 10.245, ROUND(10.245, 1), ROUND(10.245, -1)
FROM dual
```

```
SQL> SELECT 10.245, ROUND(10.245, 1), ROUND(10.245, -1)
2 FROM dual;

10.245 ROUND(10.245,1) ROUND(10.245,-1)
-----
10.245          10.2          10
```

b) Write a SELECT statement that displays distinct course costs. In a separate column, show the COST increased by 75% and round the decimals to the nearest dollar.

```
SQL> SELECT DISTINCT cost, cost*1.75, ROUND(cost*1.75)
2 FROM course;

ninguna fila seleccionada
```

c) Write a SELECT statement that displays distinct numeric grades from the GRADE table and half those values expressed as a whole number in a separate column.

```
SQL> SELECT DISTINCT numeric_grade, ROUND(numeric_grade / 2)
2 FROM grade;

ninguna fila seleccionada
```

### 3.3.1. Apply Substitution Functions and Other Miscellaneous Functions

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.

```
SQL> SELECT first_name||' '|| last_name name,  
2 phone oldphone,  
3 NVL(phone, '212-555-1212') newphone  
4 FROM student  
5 WHERE phone IS NULL;
```

ninguna fila seleccionada

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.

```
SQL> SELECT course_no, cost,  
2 NVL(cost,1000)*0.9 new  
3 FROM course  
4 WHERE course_no >= 430;
```

ninguna fila seleccionada

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.

```
SQL> SELECT student_id id, first_name||' '|| last_name name,  
2 phone,  
3 NVL2(phone, 'Phone# exists.', 'No phone# exists.')  
4 "Get this result"  
5 FROM student  
6 WHERE student_id IN (111, 112);
```

ninguna fila seleccionada



### 3.3.2. Utilize the Power of the DECODE Function and the CASE Expression

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

```
SQL> SELECT student_id, first_name||' '|| last_name name,  
2 phone,  
3 DECODE(phone, NULL, 'No phone# exists.', 'Phone# exists.')  
4 "Get this result"  
5 FROM student  
6 WHERE student_id IN (111, 112);
```

ninguna fila seleccionada

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.

```
SQL> SELECT course_no, description, prerequisite "ORIGINAL",  
2 CASE WHEN prerequisite = 120 THEN '200'  
3 WHEN prerequisite = 130 THEN 'N/A'  
4 WHEN prerequisite IS NULL THEN 'None'  
5 ELSE TO_CHAR(prerequisite)  
6 END "NEW"  
7 FROM course  
8 WHERE course_no IN (20, 120, 122, 132)  
9 ORDER BY course_no DESC;
```

ninguna fila seleccionada



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

```
SQL> SELECT student_id, zip, phone,  
2 DECODE(SUBSTR(phone, 1, 3), '212',  
3 DECODE(zip, '10048', 'North Campus',  
4 'West Campus'),  
5 'Off Campus') loc  
6 FROM student  
7 WHERE student_id IN (150, 145, 325);
```

ninguna fila seleccionada

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.

```
SQL> SELECT DISTINCT salutation  
2 FROM instructor  
3 ORDER BY DECODE(salutation, 'Ms', 1,  
4 'Mrs', 1,  
5 'Miss', 1);
```

```
SALUT  
-----  
Ms
```

## Chapter 4. Date and Conversion Functions

```
SQL> SELECT last_name, registration_date
       2 FROM student
       3 WHERE student_id IN (123, 161, 190);
```

LAST_NAME		REGISTRAT
Radicola	*	27-JAN-03
Grant	*	02-FEB-03
Affinito	*	03-FEB-03

```
SQL>
```

```
140|*|117|*|02-JUN-2003 09:30
140|*|118|*|09-MAY-2003 09:30
142|*|119|*|14-JUL-2003 09:30
142|*|120|*|10-JUN-2003 09:30
142|*|121|*|09-APR-2003 09:30
144|*|122|*|15-APR-2003 10:30
145|*|123|*|14-JUL-2003 09:30
145|*|124|*|09-MAY-2003 09:30
146|*|125|*|29-APR-2003 09:30
124|*|126|*|14-JUL-2003 09:30
124|*|127|*|24-JUL-2003 09:30
124|*|128|*|09-APR-2003 09:30
124|*|129|*|07-MAY-2003 09:30
125|*|130|*|22-MAY-2003 09:30
125|*|131|*|24-JUL-2003 09:30
125|*|132|*|09-APR-2003 09:30
125|*|133|*|03-MAY-2003 09:30
125|*|134|*|11-JUN-2003 09:30
130|*|135|*|15-APR-2003 09:30
130|*|136|*|24-APR-2003 09:30
130|*|137|*|03-MAY-2003 09:30
132|*|138|*|21-MAY-2003 09:30
132|*|139|*|09-JUN-2003 09:30
134|*|140|*|16-APR-2003 09:30
100|*|141|*|14-APR-2003 09:30
100|*|142|*|24-JUL-2003 09:30
100|*|143|*|03-JUN-2003 09:30
100|*|144|*|04-MAY-2003 09:30
100|*|145|*|15-MAY-2003 09:30
120|*|146|*|16-MAY-2003 09:30
120|*|147|*|24-JUL-2003 09:30
120|*|148|*|24-MAY-2003 09:30
120|*|149|*|04-MAY-2003 09:30
120|*|150|*|15-MAY-2003 09:30
120|*|151|*|12-JUN-2003 09:30
122|*|152|*|29-APR-2003 09:30
122|*|153|*|24-JUL-2003 09:30
122|*|154|*|21-MAY-2003 09:30
122|*|155|*|04-MAY-2003 09:30
122|*|156|*|15-MAY-2003 09:30
```

78 filas seleccionadas.

```
SQL>
```

```
140|*|117|*|02-JUN-2003 09:30
140|*|118|*|09-MAY-2003 09:30
142|*|119|*|14-JUL-2003 09:30
142|*|120|*|10-JUN-2003 09:30
142|*|121|*|09-APR-2003 09:30
144|*|122|*|15-APR-2003 10:30
145|*|123|*|14-JUL-2003 09:30
145|*|124|*|09-MAY-2003 09:30
146|*|125|*|29-APR-2003 09:30
124|*|126|*|14-JUL-2003 09:30
124|*|127|*|24-JUL-2003 09:30
124|*|128|*|09-APR-2003 09:30
124|*|129|*|07-MAY-2003 09:30
125|*|130|*|22-MAY-2003 09:30
125|*|131|*|24-JUL-2003 09:30
125|*|132|*|09-APR-2003 09:30
125|*|133|*|03-MAY-2003 09:30
125|*|134|*|11-JUN-2003 09:30
130|*|135|*|15-APR-2003 09:30
130|*|136|*|24-APR-2003 09:30
130|*|137|*|03-MAY-2003 09:30
132|*|138|*|21-MAY-2003 09:30
132|*|139|*|09-JUN-2003 09:30
134|*|140|*|16-APR-2003 09:30
100|*|141|*|14-APR-2003 09:30
100|*|142|*|24-JUL-2003 09:30
100|*|143|*|03-JUN-2003 09:30
100|*|144|*|04-MAY-2003 09:30
100|*|145|*|15-MAY-2003 09:30
120|*|146|*|16-MAY-2003 09:30
120|*|147|*|24-JUL-2003 09:30
120|*|148|*|24-MAY-2003 09:30
120|*|149|*|04-MAY-2003 09:30
120|*|150|*|15-MAY-2003 09:30
120|*|151|*|12-JUN-2003 09:30
122|*|152|*|29-APR-2003 09:30
122|*|153|*|24-JUL-2003 09:30
122|*|154|*|21-MAY-2003 09:30
122|*|155|*|04-MAY-2003 09:30
122|*|156|*|15-MAY-2003 09:30
```

78 filas seleccionadas.

```
SQL>
```

```
SQL> SELECT course_no, section_id,
2 TO_CHAR(start_date_time, 'DD-MON-YYYY HH24:MI')
3 FROM section
4 WHERE TRUNC(start_date_time) = TO_DATE('04-MAY-2003', 'DD-MON-YYYY');
```

COURSE_NO	SECTION_ID	TO_CHAR(START_DATE_TIME, 'D')
25	88	04-MAY-2003 09:30
100	144	04-MAY-2003 09:30
120	149	04-MAY-2003 09:30
122	155	04-MAY-2003 09:30

SQL>

```
SQL> SELECT course_no, section_id,
2 TO_CHAR(start_date_time, 'DD-MON-YYYY HH24:MI')
3 FROM section
4 WHERE TRUNC(start_date_time) = DATE '2003-05-04';
```

COURSE_NO	SECTION_ID	TO_CHAR(START_DATE_TIME, 'D')
25	88	04-MAY-2003 09:30
100	144	04-MAY-2003 09:30
120	149	04-MAY-2003 09:30
122	155	04-MAY-2003 09:30

SQL>

```
SQL> SELECT course_no, section_id,
2 TO_CHAR(start_date_time, 'DY DD-MON-YYYY')
3 FROM section
4 WHERE TO_CHAR(start_date_time, 'DY') = 'TUE';
```

COURSE_NO	SECTION_ID	TO_CHAR(START_DATE_TIME, 'DYDD-MON')
25	86	TUE 10-JUN-2003
220	98	TUE 15-APR-2003
310	103	TUE 29-APR-2003
350	106	TUE 03-JUN-2003
134	110	TUE 10-JUN-2003
135	114	TUE 15-APR-2003
142	120	TUE 10-JUN-2003
144	122	TUE 15-APR-2003
146	125	TUE 29-APR-2003
130	135	TUE 15-APR-2003
100	143	TUE 03-JUN-2003
122	152	TUE 29-APR-2003

12 filas seleccionadas.

SQL>

```
SQL> SELECT course_no, section_id,
2 TO_CHAR(start_date_time, 'Day DD-Mon-YYYY')
3 FROM section
4 WHERE TO_CHAR(start_date_time, 'fmDay') = 'Tuesday';
```

COURSE_NO	SECTION_ID	TO_CHAR(START_DATE_TIME, 'DAYDD-MON-YYYY')
25	86	Tuesday 10-Jun-2003
220	98	Tuesday 15-Apr-2003
310	103	Tuesday 29-Apr-2003
350	106	Tuesday 03-Jun-2003
134	110	Tuesday 10-Jun-2003
135	114	Tuesday 15-Apr-2003
142	120	Tuesday 10-Jun-2003
144	122	Tuesday 15-Apr-2003
146	125	Tuesday 29-Apr-2003
130	135	Tuesday 15-Apr-2003
100	143	Tuesday 03-Jun-2003
122	152	Tuesday 29-Apr-2003

12 filas seleccionadas.

SQL>

```
SQL Plus
117 * 02-JUN-2003 09:30:00
118 * 09-MAY-2003 09:30:00
119 * 14-JUL-2003 09:30:00
120 * 10-JUN-2003 09:30:00
121 * 09-APR-2003 09:30:00
122 * 15-APR-2003 10:30:00
123 * 14-JUL-2003 09:30:00
124 * 09-MAY-2003 09:30:00
125 * 29-APR-2003 09:30:00
126 * 14-JUL-2003 09:30:00
127 * 24-JUL-2003 09:30:00
128 * 09-APR-2003 09:30:00
129 * 07-MAY-2003 09:30:00
130 * 22-MAY-2003 09:30:00
131 * 24-JUL-2003 09:30:00
132 * 09-APR-2003 09:30:00
133 * 03-MAY-2003 09:30:00
134 * 11-JUN-2003 09:30:00
135 * 15-APR-2003 09:30:00
136 * 24-APR-2003 09:30:00
137 * 03-MAY-2003 09:30:00
138 * 21-MAY-2003 09:30:00
139 * 09-JUN-2003 09:30:00
140 * 16-APR-2003 09:30:00
141 * 14-APR-2003 09:30:00
142 * 24-JUL-2003 09:30:00
143 * 03-JUN-2003 09:30:00
144 * 04-MAY-2003 09:30:00
145 * 15-MAY-2003 09:30:00
146 * 16-MAY-2003 09:30:00
147 * 24-JUL-2003 09:30:00
148 * 24-MAY-2003 09:30:00
149 * 04-MAY-2003 09:30:00
150 * 15-MAY-2003 09:30:00
151 * 12-JUN-2003 09:30:00
152 * 29-APR-2003 09:30:00
153 * 24-JUL-2003 09:30:00
154 * 21-MAY-2003 09:30:00
155 * 04-MAY-2003 09:30:00
156 * 15-MAY-2003 09:30:00

78 filas seleccionadas.

SQL>
```

```
SQL> SELECT LAST_DAY(TO_DATE('13-FEB-1964','DD-MON-YYYY')) lastday,
2 LAST_DAY(TO_DATE('13-FEB-1964','DD-MON-YYYY'))
3 - TO_DATE('13-FEB-1964','DD-MON-YYYY') days
4 FROM dual;
```

LASTDAY	DAYS
29-FEB-64	16

SQL>

```
SQL> SELECT MONTHS_BETWEEN(TO_DATE('17-AUG-2003','DD-MON-YYYY'),
2 TO_DATE('29-SEP-1999','DD-MON-YYYY')) months
3 FROM dual;
```

MONTHS
46.6129032

SQL>

```
SQL> SELECT TO_CHAR(SYSDATE, 'DD-MON-YYYY HH24:MI:SS') "Current",
2 TO_CHAR(SYSDATE+3, 'DD-MON-YYYY HH24:MI:SS') "Answer"
3 FROM dual;
```

Current	Answer
21-SEP-2021 16:23:32	24-SEP-2021 16:23:32

SQL>

```
SQL> SELECT col_date, col_timestamp, col_timestamp_w_tz
2 FROM date_example;
FROM date_example
*
```

ERROR en línea 2:  
ORA-00942: la tabla o vista no existe

SQL>

```
SQL> SELECT TZ_OFFSET('Europe/London') "London",
2  TZ_OFFSET('America/New_York') "NY",
3  TZ_OFFSET('America/Chicago') "Chicago",
4  TZ_OFFSET('America/Denver') "Denver",
5  TZ_OFFSET('America/Los_Angeles') "LA"
6  FROM dual
7  ;
```

London	NY	Chicago	Denver	LA
+01:00	-04:00	-05:00	-06:00	-07:00

```
SQL>
```

## Chapter 5 Equijoins

### Lab 6.1 The Two-Table Join

rem [Rischert, 2004,330]

```
SELECT c.course_no, s.section_no, c.description,
s.location, s.instructor_id
FROM course c, section s
WHERE c.course_no = s.course_no;
```

Seleccin SQL Plus

COURSE_NO	SECTION_NO	DESCRIPTION	LOCATION	INSTRUCTOR_ID
240	1	Intro to the Basic Language	L509	101
240	2	Intro to the Basic Language	L214	102
310	1	Operating Systems	L507	103
330	1	Network Administration	L511	104
350	3	JDeveloper Lab	L509	107
350	1	JDeveloper Lab	L509	105
350	2	JDeveloper Lab	L214	106
420	1	Database System Principles	M311	108
450	1	DB Programming in Java	L507	101

78 filas seleccionadas.

```
SQL>
```

rem [Rischert, 2004,331]

```
SELECT course_no, s.section_no, c.description,  
       s.location, s.instructor_id  
FROM course c JOIN section s  
USING (course_no);
```

```

SQL Plus
-----
LOCATION                                INSTRUCTOR_ID
-----
L509      240      1 Intro to the Basic Language      101
L214      240      2 Intro to the Basic Language      102
L507      310      1 Operating Systems                103

COURSE_NO SECTION_NO DESCRIPTION
-----
LOCATION                                INSTRUCTOR_ID
-----
L511      330      1 Network Administration           104
L509      350      3 JDeveloper Lab                   107
L509      350      1 JDeveloper Lab                   105

COURSE_NO SECTION_NO DESCRIPTION
-----
LOCATION                                INSTRUCTOR_ID
-----
L214      350      2 JDeveloper Lab                   106
M311      420      1 Database System Principles       108
L507      450      1 DB Programming in Java           101

78 filas seleccionadas.
SQL>

```

rem [Rischert, 2004,334]

```
SELECT COUNT(*)
FROM section, instructor;
```

```

SQL Plus
350      1 Developer Lab
L509                                     105

COURSE_NO SECTION_NO DESCRIPTION
-----
LOCATION                                INSTRUCTOR_ID
-----
350      2 Developer Lab
L214                                     106
420      1 Database System Principles
B311                                     108
450      1 DB Programming in Java
L507                                     101

78 filas seleccionadas.

SQL> SELECT course_no, s.section_no, c.description,
2   s.location, s.instructor_id
3   FROM course c NATURAL JOIN section s;

ninguna fila seleccionada

SQL> SELECT course_no, s.section_no, c.description,
2   s.location, s.instructor_id
3   FROM course c NATURAL JOIN section s
4   no rows selected;
no rows selected
*
ERROR en línea 4:
ORA-00933: comando SQL no terminado correctamente

SQL> SELECT COUNT(*)
2   FROM section, instructor;

COUNT(*)
-----
780

SQL>

```

```
rem [Rischert, 2004,334]
SELECT s.instructor_id s_instructor_id,
i.instructor_id i_instructor_id
FROM section s, instructor i;
```

```
SELECT s.instructor_id s_instructor_id,  
i.instructor_id i_instructor_id  
FROM section s, instructor i;
```

```
i.instructor_id i_instructor_id
FROM section s, instructor i;
```

FROM section s, instructor i;

```
SQL Plus
```

S_INSTRUCTOR_ID	I_INSTRUCTOR_ID
105	110
105	110
105	110
106	110
106	110
106	110
106	110
106	110
106	110
106	110
106	110
106	110
106	110
S_INSTRUCTOR_ID	I_INSTRUCTOR_ID
106	110
106	110
107	110
107	110
107	110
107	110
107	110
107	110
107	110
107	110
107	110
S_INSTRUCTOR_ID	I_INSTRUCTOR_ID
107	110
108	110
108	110
108	110
108	110
108	110
108	110
108	110
108	110
108	110
108	110

780 filas seleccionadas.

```
SQL>
```

## Lab 6.1 Exercises

### 6.1.1. Write Simple Join Constructs

a) For all students, display last name, city, state, and zip code. Show the result ordered by zip code.

```
SELECT s.last_name, s.zip, z.state, z.city
FROM student s, zipcode z
```

FROM student s, zipcode z

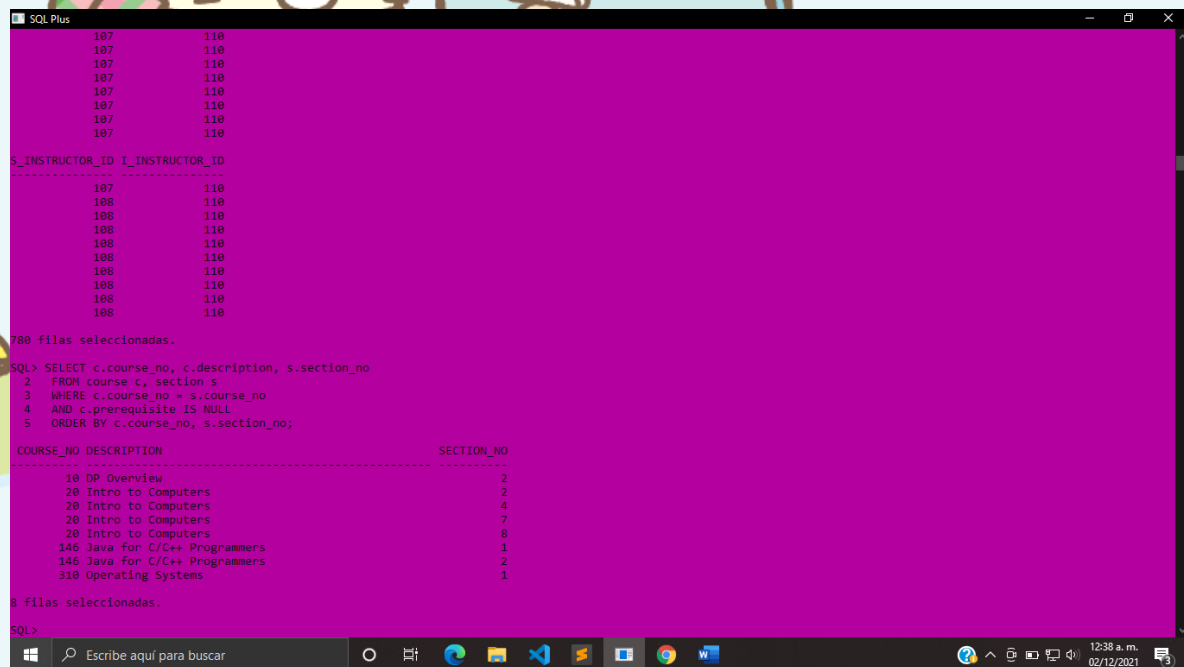
WHERE s.zip = z.zip ORDER BY s.zip



### 6.1.2. Narrow Down Your Result Set

a) Execute the following SQL statement. Explain your observations about the WHERE clause and the resulting output.

```
SELECT c.course_no, c.description, s.section_no  
FROM course c, section s  
WHERE c.course_no = s.course_no AND c.prerequisite  
IS NULL ORDER BY c.course_no, s.section_no
```



The screenshot shows a SQL Plus window with a dark background. The top part displays the results of a query, showing two columns: S\_INSTRUCTOR\_ID and I\_INSTRUCTOR\_ID. The results are as follows:

S_INSTRUCTOR_ID	I_INSTRUCTOR_ID
107	110
107	110
107	110
107	110
107	110
107	110
107	110
107	110
107	110
108	110
108	110
108	110
108	110
108	110
108	110
108	110
108	110
108	110
108	110

Below the results, it says "780 filas seleccionadas." (780 rows selected).

The SQL code entered is:

```
SQL> SELECT c.course_no, c.description, s.section_no  
2 FROM course c, section s  
3 WHERE c.course_no = s.course_no  
4 AND c.prerequisite IS NULL  
5 ORDER BY c.course_no, s.section_no;
```

The output shows a table with three columns: COURSE\_NO, DESCRIPTION, and SECTION\_NO. The results are as follows:

COURSE_NO	DESCRIPTION	SECTION_NO
10	DP Overview	2
20	Intro to Computers	2
20	Intro to Computers	4
20	Intro to Computers	7
20	Intro to Computers	8
146	Java for C/C++ Programmers	1
146	Java for C/C++ Programmers	2
310	Operating Systems	1

Below the results, it says "8 filas seleccionadas." (8 rows selected).

The bottom of the window shows the Windows taskbar with the search bar and various application icons. The system clock shows 12:38 a.m. on 02/12/2021.

rem [Rischert, 2004,339]

```
SELECT c.course_no, c.description, s.section_no  
FROM course c, section s  
WHERE c.course_no = s.course_no  
AND c.prerequisite IS NULL  
ORDER BY c.course_no, s.section_no;
```

```
SQL Plus

LAST_NAME          ZIP    ST CITY
-----
Abid               11565  NY Malverne
Ross               11565  NY Malverne
Balterzar          11576  NY Roslyn
McHowell           11717  NJ Brentwood
Carey              11743  NY Huntington
Brendler           11746  NY Dix Hills
Krot               11746  NY Dix Hills
Korka              11746  NY Dix Hills
Galik              11758  NY Massapequa
Waldman            11766  NY Huntington
Gates              11787  NY Smithtown

LAST_NAME          ZIP    ST CITY
-----
Bose               11787  NY Smithtown
Hwangi             13760  NY Endicott
Gilloon            43224  OH Columbus
Snow               48104  MI Ann Arbor

268 filas seleccionadas.

SQL> SELECT c.course_no, c.description, s.section_no
2 FROM course c, section s
3 WHERE c.course_no = s.course_no
4 AND c.prerequisite IS NULL
5 ORDER BY c.course_no, s.section_no;

COURSE_NO DESCRIPTION          SECTION_NO
-----
10 DP Overview                2
20 Intro to Computers         2
20 Intro to Computers         4
20 Intro to Computers         7
20 Intro to Computers         8
146 Java for C/C++ Programmers 1
146 Java for C/C++ Programmers 2
310 Operating Systems         1

8 filas seleccionadas.

SQL>
```

rem [Rischert, 2004,340]

```
SELECT e.student_id, s.course_no,
TO_CHAR(e.enroll_date,'MM/DD/YYYY HH:MI PM'),
e.section_id
FROM enrollment e JOIN section s
ON (e.section_id = s.section_id)
WHERE s.course_no = 20
AND e.enroll_date >= TO_DATE('01/30/2003','MM/DD/YYYY')
AND e.enroll_date < TO_DATE('01/31/2003','MM/DD/YYYY');
```

```
SQL Plus

Bose               11787  NY Smithtown
Hwangi             13760  NY Endicott
Gilloon            43224  OH Columbus
Snow               48104  MI Ann Arbor

268 filas seleccionadas.

SQL> SELECT c.course_no, c.description, s.section_no
2 FROM course c, section s
3 WHERE c.course_no = s.course_no
4 AND c.prerequisite IS NULL
5 ORDER BY c.course_no, s.section_no;

COURSE_NO DESCRIPTION          SECTION_NO
-----
10 DP Overview                2
20 Intro to Computers         2
20 Intro to Computers         4
20 Intro to Computers         7
20 Intro to Computers         8
146 Java for C/C++ Programmers 1
146 Java for C/C++ Programmers 2
310 Operating Systems         1

8 filas seleccionadas.

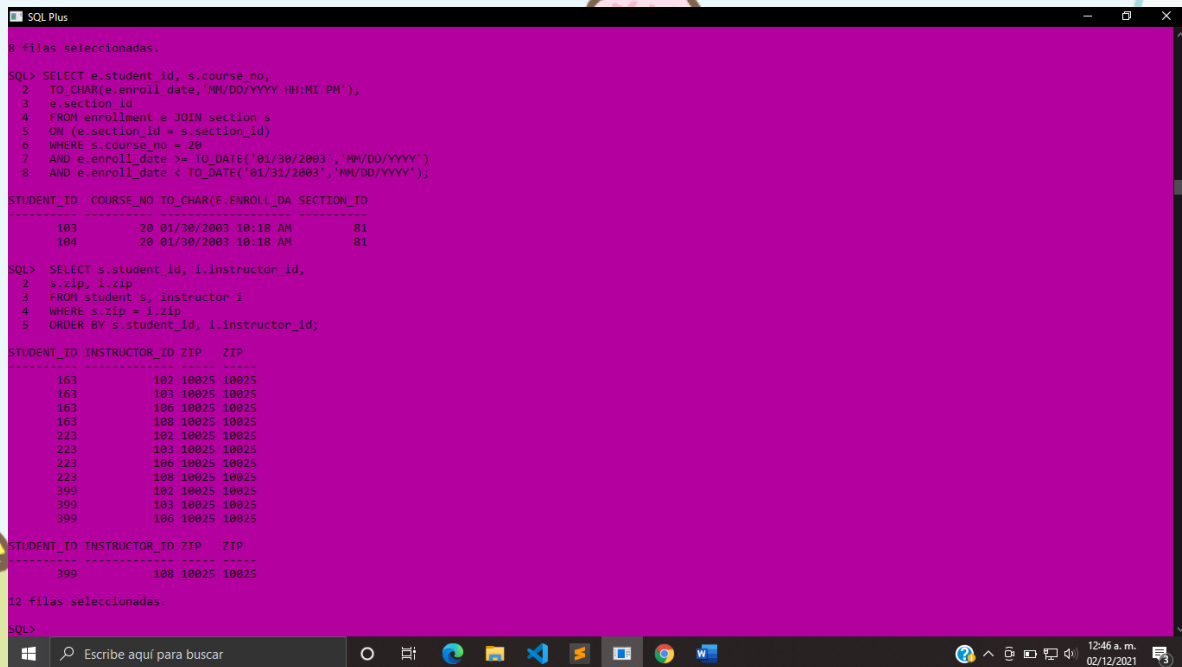
SQL> SELECT e.student_id, s.course_no,
2 TO_CHAR(e.enroll_date,'MM/DD/YYYY HH:MI PM'),
3 e.section_id
4 FROM enrollment e JOIN section s
5 ON (e.section_id = s.section_id)
6 WHERE s.course_no = 20
7 AND e.enroll_date >= TO_DATE('01/30/2003','MM/DD/YYYY')
8 AND e.enroll_date < TO_DATE('01/31/2003','MM/DD/YYYY');

STUDENT_ID  COURSE_NO  TO_CHAR(E.ENROLL_DA  SECTION_ID
-----
103         20 01/30/2003 10:18 AM      81
104         20 01/30/2003 10:18 AM      81

SQL>
```

rem [Rischert, 2004,341]

```
SELECT s.student_id, i.instructor_id,  
  
s.zip, i.zip  
  
FROM student s, instructor i  
  
WHERE s.zip = i.zip  
  
ORDER BY s.student_id, i.instructor_id;
```



8 filas seleccionadas.

```
SQL> SELECT e.student_id, s.course_no,  
2 TO_CHAR(e.enroll_date,'MM/DD/YYYY HH:MI PM'),  
3 e.section_id  
4 FROM enrollment e JOIN section s  
5 ON (e.section_id = s.section_id)  
6 WHERE s.course_no = 20  
7 AND e.enroll_date >= TO_DATE('01/30/2003','MM/DD/YYYY')  
8 AND e.enroll_date < TO_DATE('01/31/2003','MM/DD/YYYY');
```

STUDENT_ID	COURSE_NO	TO_CHAR(E.ENROLL_DA	SECTION_ID
103	20	01/30/2003 10:18 AM	81
104	20	01/30/2003 10:18 AM	81

```
SQL> SELECT s.student_id, i.instructor_id,  
2 s.zip, i.zip  
3 FROM student s, instructor i  
4 WHERE s.zip = i.zip  
5 ORDER BY s.student_id, i.instructor_id;
```

STUDENT_ID	INSTRUCTOR_ID	ZIP	ZIP
103	102	10025	10025
103	103	10025	10025
103	106	10025	10025
103	108	10025	10025
223	102	10025	10025
223	103	10025	10025
223	106	10025	10025
223	108	10025	10025
399	102	10025	10025
399	103	10025	10025
399	106	10025	10025

12 filas seleccionadas.

```
SQL>
```

## Lab 6.2 Exercises

### 6.2.1. Join Three or More Tables

Display the student ID, course number, and section number of enrolled students where the instructor of the section lives in zip code 10025. Additionally, the course should not have any prerequisites.

```
SQL> SELECT c.course_no, s.section_no, e.student_id
2   FROM course c, section s, instructor i, enrollment e
3   WHERE c.prerequisite IS NULL
4   AND c.course_no = s.course_no
5   AND s.instructor_id = i.instructor_id
6   AND i.zip = '10025'
7   AND s.section_id = e.section_id;
```

COURSE_NO	SECTION_NO	STUDENT_ID
-----------	------------	------------

20	2	103
20	2	104
146	2	117
310	1	119
310	1	120
10	2	128
146	2	140
20	8	158
20	8	199
310	1	238
20	2	240

COURSE_NO	SECTION_NO	STUDENT_ID
-----------	------------	------------

310	1	243
-----	---	-----

12 filas seleccionadas.

rem [Rischert, 2004,358]

```
SELECT last_name, first_name
FROM student
WHERE last_name = 'Crocitto'
AND first_name = 'Fred';
```

```
SQL> SELECT last_name, first_name
2   FROM student
3   WHERE last_name = 'Crocitto'
4   AND first_name = 'Fred';
```

LAST_NAME	FIRST_NAME
Crocitto	Fred

## Chapter 7. Subqueries

### Lab 7.1 Exercises

rem [Rischert, 2004,376]

```
SELECT first_name, last_name
FROM student
WHERE registration_date =
(SELECT MIN(registration_date)
FROM student);
```

```
SQL> SELECT first_name, last_name
2   FROM student
3   WHERE registration_date =
4   (SELECT MIN(registration_date)
5   FROM student);
```

FIRST_NAME	LAST_NAME
Fred	Crocitto
J.	Landry
Laetia	Enison
Angel	Moskowitz
Judith	Olvsade
Catherine	Mierzwa
Judy	Sethi
Larry	Walter

8 filas seleccionadas.

```

SQL> SELECT c.description, s.section_no, c.cost, s.capacity
2   FROM course c, section s
3   WHERE c.course_no = s.course_no
4   AND s.capacity <=
5   (SELECT AVG(capacity)
6   FROM section)
7   AND c.cost =
8   (SELECT MIN(cost)
9   FROM course);

```

DESCRIPTION	SECTION_NO	COST
-----	-----	-----
CAPACITY		
-----		
Unix Tips and Techniques 15	4	1095

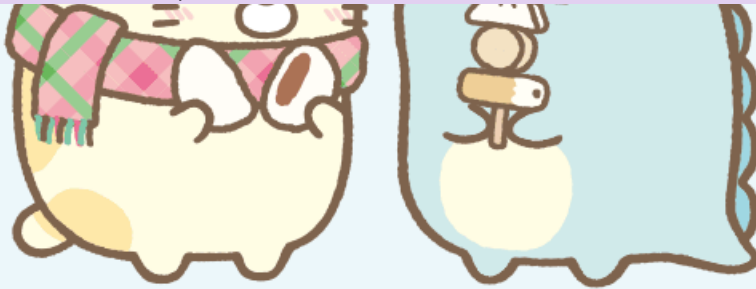
Unix Tips and Techniques 15	2	1095
Intro to Internet 15	2	1095

DESCRIPTION	SECTION_NO	COST
-----	-----	-----
CAPACITY		
-----		
Intro to Internet 12	1	1095
Intro to the Basic Language 15	2	1095
Intro to the Basic Language 10	1	1095

6 filas seleccionadas.

rem [Rischert, 2004,376]

```
SELECT c.description, s.section_no, c.cost, s.capacity
FROM course c, section s
WHERE c.course_no = s.course_no
AND s.capacity <=
(SELECT AVG(capacity)
FROM section)
AND c.cost =
(SELECT MIN(cost)
FROM course);
```





rem [Rischert, 2004,381]

```
SELECT instructor_id, last_name, first_name, zip
FROM instructor i
WHERE EXISTS
(SELECT 'X'
FROM section
WHERE i.instructor_id = instructor_id);
```

```
SQL> SELECT instructor_id, last_name, first_name, zip
2   FROM instructor i
3   WHERE EXISTS
4   (SELECT 'X'
5   FROM section
6   WHERE i.instructor_id = instructor_id);
```

INSTRUCTOR_ID	LAST_NAME	FIRST_NAME	ZIP
101	Hanks	Fernand	10015
102	Wojick	Tom	10025
103	Schorin	Nina	10025
104	Pertez	Gary	10035
105	Morris	Anita	10015
106	Smythe	Todd	10025
107	Frantzen	Marilyn	10005
108	Lowry	Charles	10025

8 filas seleccionadas.

rem [Rischert, 2004,388]

```
SELECT section_id, capacity
FROM section
WHERE course_no = 20;
```

```
SQL> SELECT section_id, capacity
2   FROM section
3   WHERE course_no = 20;
```

SECTION_ID	CAPACITY
81	15
82	15
83	25
84	25

NAME		
-----		
STREET_ADDRESS		
-----		
City State Zip	START_DT	SECT
-----		
518 West 120th New York, NY 10025	06/09/03	139
Gary Pertez 34 Sixth Ave New York, NY 10035	06/12/03	90
NAME		
-----		
STREET_ADDRESS		
-----		
City State Zip	START_DT	SECT
-----		
Gary Pertez 34 Sixth Ave New York, NY 10035	06/10/03	120
Gary Pertez 34 Sixth Ave New York, NY 10035	06/03/03	143
NAME		
-----		
STREET_ADDRESS		
-----		
City State Zip	START_DT	SECT
-----		
Gary Pertez 34 Sixth Ave New York, NY 10035	06/12/03	151
17 filas seleccionadas.		


rem [Rischert, 2004,377]

```
SELECT course_no, SUM(capacity)
FROM section
GROUP BY course_no
HAVING SUM(capacity) <
(SELECT AVG(capacity)
FROM section);
```

6 filas seleccionadas.

```
SQL> SELECT course_no, SUM(capacity)
2   FROM section
3   GROUP BY course_no
4   HAVING SUM(capacity) <
5   (SELECT AVG(capacity)
6   FROM section);
```

COURSE_NO	SUM(CAPACITY)
10	15
144	15



COURSE_NO	SUM(CAPACITY)
-----------	---------------

240	25
310	25
330	25
350	65
420	25
450	25

28 filas seleccionadas.

```
SQL> SELECT AVG(capacity)
2 FROM section;
```

AVG(CAPACITY)
---------------

21.1794872
------------

```
SQL> SELECT student_id, COUNT(*)
2 FROM enrollment
3 GROUP BY student_id
4 HAVING COUNT(*) =
5 (SELECT MAX(COUNT(*))
6 FROM enrollment
7 GROUP BY student_id);
```

STUDENT_ID	COUNT(*)
------------	----------

124	4
214	4

COUNT(\*)

1  
1  
1  
2  
2  
2  
2  
2  
2  
2  
2  
2


165 filas seleccionadas.

```
SQL> SELECT course_no, description
2 FROM course
3 WHERE course_no IN
4 (SELECT course_no
5 FROM section
6 WHERE instructor_id IN
7 (SELECT instructor_id
8 FROM instructor
9 WHERE last_name = 'Hanks'
10 AND first_name = 'Fernand'));
```

COURSE\_NO DESCRIPTION

-----  
25 Intro to Programming  
120 Intro to Java Programming  
122 Intermediate Java Programming  
125 JDeveloper  
134 Advanced Unix Admin  
140 Structured Analysis  
146 Java for C/C++ Programmers  
240 Intro to the Basic Language  
450 DB Programming in Java

9 filas seleccionadas.



```
SQL> SELECT section_id, capacity
2   FROM section
3   WHERE (section_id, capacity) IN
4   (SELECT section_id, COUNT(*)
5   FROM enrollment
6   GROUP BY section_id);
```

SECTION_ID	CAPACITY
99	12

rem [Rischert, 2004,397]

```
SELECT course_no, section_id
FROM section s
WHERE NOT EXISTS
(SELECT NULL
FROM enrollment
WHERE s.section_id = section_id)
ORDER BY course_no;
```

## 7.2 exercises

Explain what the following correlated subquery accomplishes.

```
3 WHERE 2 >  
4 (SELECT COUNT(*)  
5 FROM enrollment  
6 WHERE section_id = s.section_id);
```

SECTION_ID	COURSE_NO
79	350
80	10
93	25
96	204
97	210
98	220
102	240
109	450
110	134
111	134
113	135

SECTION_ID	COURSE_NO
114	135
115	135
118	140
121	142
122	144
124	145
125	146
127	124
129	124
131	125
134	125

SECTION_ID	COURSE_NO
136	130
139	132
140	134
145	100
149	120

27 filas seleccionadas.



Then write a second query to show the sections without any enrollments (i.e., the SECTION\_ID does not exist in the ENROLLMENT table). To determine these sections, you can use the NOT IN operator because the SECTION\_ID in the ENROLLMENT table is defined as NOT NULL.

```
SQL> SELECT section_id  
2   FROM section  
3   WHERE section_id NOT IN  
4   (SELECT section_id  
5   FROM enrollment);
```

SECTION\_ID

-----  
79  
93  
97  
98  
110  
114  
118  
121  
122  
124  
129

SECTION\_ID

-----  
134  
136  
139

14 filas seleccionadas.

rem [Rischert, 2004,393]

```
SELECT section_id  
FROM section  
WHERE section_id NOT IN  
(SELECT section_id  
FROM enrollment);
```

```
SQL> SELECT course_no, section_id
2   FROM section s
3   WHERE NOT EXISTS
4     (SELECT NULL
5     FROM enrollment
6     WHERE s.section_id = section_id)
7   ORDER BY course_no;
```

COURSE_NO	SECTION_ID
25	93
124	129
125	134
130	136
132	139
134	110
135	114
140	118
142	121
144	122
145	124

COURSE_NO	SECTION_ID
210	97
220	98
350	79

14 filas seleccionadas.

## Lab 7.4 Exercise

rem [Rischert, 2004,417]

```
SELECT student_id, section_id, numeric_grade
FROM grade g
WHERE grade_type_code = 'FI'
AND numeric_grade > ALL
(SELECT numeric_grade
FROM grade
WHERE grade_type_code = 'HM'
AND g.section_id = section_id
AND g.student_id = student_id);
```

STUDENT_ID	SECTION_ID	NUMERIC_GRADE
189	116	84
202	105	99
237	85	84
102	89	92
170	156	76
280	101	85
194	116	85
238	141	92

96 filas seleccionadas.

rem [Rischert, 2004,418]

```
SELECT student_id, section_id, grade_type_code,
MAX(numeric_grade) max, MIN(numeric_grade) min
FROM grade
WHERE student_id = 179
AND section_id = 116
AND grade_type_code IN ('HM', 'FI')
GROUP BY student_id, section_id, grade_type_code;
```

```
SQL> SELECT student_id, section_id, grade_type_code,
2 MAX(numeric_grade) max, MIN(numeric_grade) min
3 FROM grade
4 WHERE student_id = 179
5 AND section_id = 116
6 AND grade_type_code IN ('HM', 'FI')
7 GROUP BY student_id, section_id, grade_type_code;
```

STUDENT_ID	SECTION_ID	GR	MAX	MIN
179	116	FI	90	90
179	116	HM	99	99

rem [Rischert, 2004,420]

```
SELECT student_id, section_id, grade_type_code,  
numeric_grade  
FROM grade  
WHERE student_id = 102  
AND section_id = 89  
AND grade_type_code IN ('HM', 'FI')
```

```
SQL> SELECT student_id, section_id, grade_type_code,  
2 numeric_grade  
3 FROM grade  
4 WHERE student_id = 102  
5 AND section_id = 89  
6 AND grade_type_code IN ('HM', 'FI');
```

STUDENT_ID	SECTION_ID	GR	NUMERIC_GRADE
102	89	FI	92

## Chapter 8. Set Operators

rem [Rischert, 2004,425]

```
SELECT first_name, last_name, phone, COUNT(*)  
FROM student  
GROUP BY first_name, last_name, phone  
HAVING COUNT(*) > 1;
```

```
SQL> SELECT first_name, last_name, phone, COUNT(*)  
2 FROM student  
3 GROUP BY first_name, last_name, phone  
4 HAVING COUNT(*) > 1;
```

FIRST_NAME	LAST_NAME	PHONE	COUNT(*)
Thomas	Edwards	201-555-5555	2
Kevin	Porch	201-555-5555	2

rem [Rischert, 2004,426]

```
SELECT instructor_id id, first_name, last_name, phone  
FROM instructor  
UNION  
SELECT student_id, first_name, last_name, phone  
FROM student  
ORDER BY 3;
```

ID	FIRST_NAME	LAST_NAME	PHONE
248	Tamara	Zapulla	201-555-5555
331	Mei-Wah	Zopf	718-555-5555
184	Salewa	Zuckerberg	718-555-5555

278 filas seleccionadas.

## Lab 8.1 Exercises

```
rem [Rischert, 2004,427]
SELECT first_name, last_name,
       'Instructor' "Type"
FROM instructor
UNION
SELECT first_name, last_name,
       'Student'
FROM student;
```

FIRST_NAME	LAST_NAME	Type
Zalman	Draquez	Student

276 filas seleccionadas.

```
rem [Rischert, 2004,429]
SELECT first_name, last_name,
       'Instructor' "Type"
FROM instructor
UNION
SELECT first_name, last_name,
       'Student'
FROM student;
```

FIRST_NAME	LAST_NAME	Type
Zalman	Draquez	Student

276 filas seleccionadas.

rem [Rischert, 2004,430]

```
SELECT created_by
FROM enrollment
UNION
SELECT created_by
FROM grade
UNION
SELECT created_by
FROM grade_type
UNION
SELECT created_by
FROM grade_conversion;
```

```
SQL> SELECT created_by
2   FROM enrollment
3   UNION
4   SELECT created_by
5   FROM grade
6   UNION
7   SELECT created_by
8   FROM grade_type
9   UNION
10  SELECT created_by
11  FROM grade_conversion;
```

CREATED\_BY

-----  
ARISCHER  
BMOTIVAL  
BROSENZW  
CBRENNAN  
DSCHERER  
JAYCAF  
MCAFFREY

7 filas seleccionadas.



rem [Rischert, 2004,432]

```
SELECT DISTINCT salutation, CAST(NULL AS NUMBER),  
state, z.created_date  
FROM instructor i, zipcode z  
WHERE i.zip = z.zip  
UNION ALL  
SELECT salutation, COUNT(*),  
state, TO_DATE(NULL)  
FROM student s, zipcode z  
WHERE s.zip = z.zip  
GROUP BY salutation, state;
```

SALUT	CAST(NULLASNUMBER)	ST	CREATED_
Dr.	1	NJ	
Ms.	1	MA	
Mr.	41	NJ	
Mr.	1	OH	
Mr.	2	WV	
Rev	1	NJ	
Mr.	13	CT	
Mr.	1	MI	

19 filas seleccionadas.

## Lab 8.2 Exercises

rem [Rischert, 2004,440]

```
SELECT course_no, description
FROM course
MINUS
SELECT s.course_no, c.description
FROM section s, course c
WHERE s.course_no = c.course_no;
```

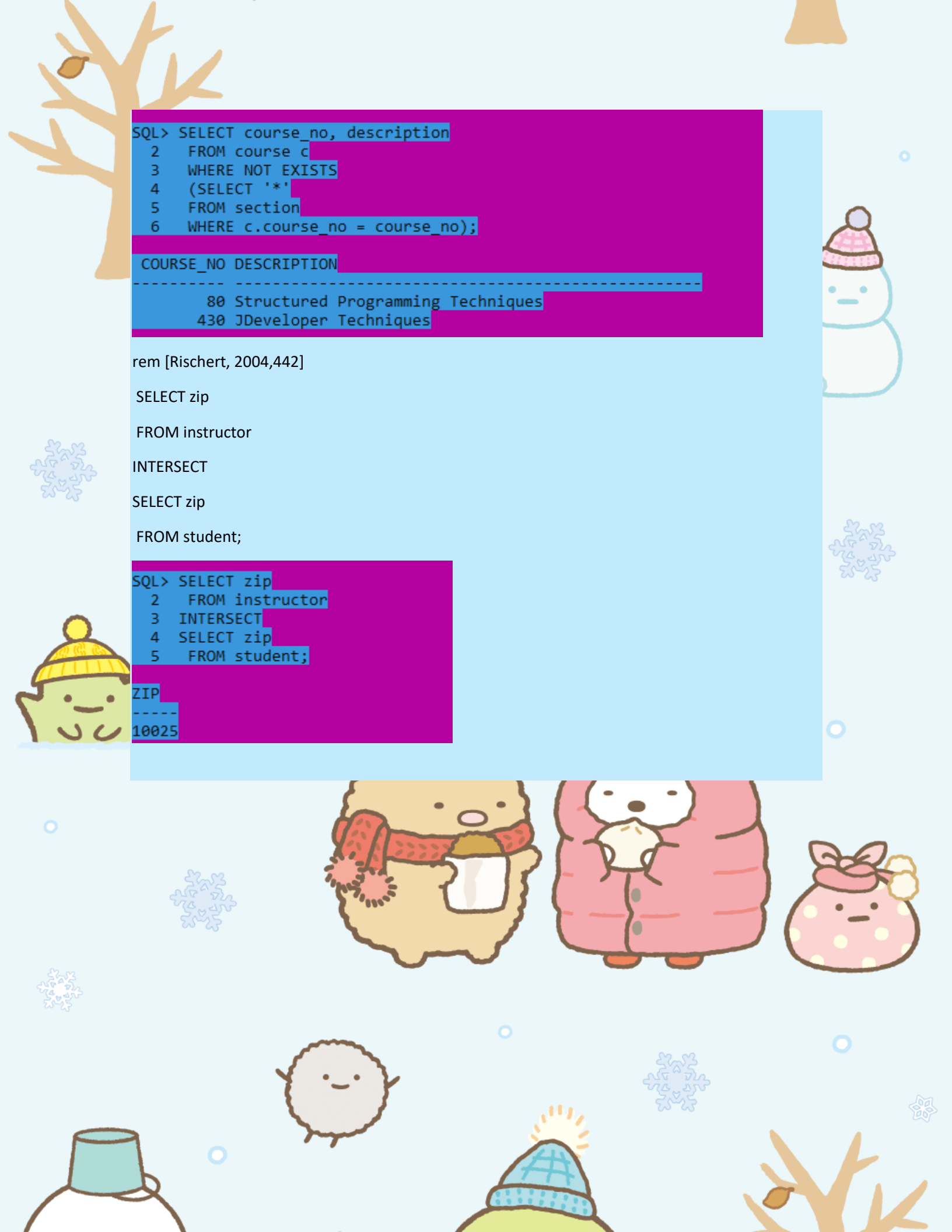
```
SQL> SELECT course_no, description
2   FROM course
3   MINUS
4   SELECT s.course_no, c.description
5   FROM section s, course c
6   WHERE s.course_no = c.course_no;
```

```
COURSE_NO DESCRIPTION
```

```
-----
80 Structured Programming Techniques
430 JDeveloper Techniques
```

rem [Rischert, 2004,441]

```
SELECT course_no, description
FROM course c
WHERE NOT EXISTS
(SELECT '*'
FROM section
WHERE c.course_no = course_no);
```



```
SQL> SELECT course_no, description
2   FROM course c
3   WHERE NOT EXISTS
4     (SELECT '*'
5      FROM section
6      WHERE c.course_no = course_no);
```

```
COURSE_NO DESCRIPTION
```

```
-----
      80 Structured Programming Techniques
      430 JDeveloper Techniques
```

rem [Rischert, 2004,442]

```
SELECT zip
FROM instructor
INTERSECT
SELECT zip
FROM student;
```

```
SQL> SELECT zip
2   FROM instructor
3   INTERSECT
4   SELECT zip
5   FROM student;
```

```
ZIP
----
10025
```

## Chapter 9. Complex Joins

rem [Rischert, 2004,447]

SELECT course\_no, description,

section\_id

FROM course JOIN section

USING (course\_no)

ORDER BY course\_no;

COURSE_NO	DESCRIPTION	SECTION_ID
220	PL/SQL Programming	98
230	Intro to Internet	100
230	Intro to Internet	99
240	Intro to the Basic Language	102
240	Intro to the Basic Language	101
310	Operating Systems	103
330	Network Administration	104
350	JDeveloper Lab	106
350	JDeveloper Lab	105
350	JDeveloper Lab	79
420	Database System Principles	108

COURSE_NO	DESCRIPTION	SECTION_ID
450	DB Programming in Java	109

78 filas seleccionadas.

rem [Rischert, 2004,448]

```
SELECT c.course_no, c.description,  
s.section_id, s.course_no  
FROM course c LEFT OUTER JOIN section s  
ON c.course_no = s.course_no  
ORDER BY c.course_no;
```

COURSE_NO	DESCRIPTION	SECTION_ID
430	JDeveloper Techniques	
450	DB Programming in Java	109
450		

80 filas seleccionadas.

rem [Rischert, 2004,452]

```
SELECT col1, col2  
FROM t1, t2  
WHERE t1.col1 = t2.col2(+)  
UNION  
SELECT col1, col2  
FROM t1, t2  
WHERE t1.col1(+) = t2.col2;
```

## Lab 9.1 Exercises

rem [Rischert, 2004,455]

```
SELECT c.course_no, s.course_no, s.section_id,  
c.description, s.start_date_time  
FROM course c FULL OUTER JOIN section s  
ON c.course_no = s.course_no;
```

```
COURSE_NO  COURSE_NO SECTION_ID  
-----  
DESCRIPTION                                START_DA  
-----  
430  
JDeveloper Techniques  
80  
Structured Programming Techniques  
80 filas seleccionadas.
```

rem [Rischert, 2004,456]

```
SELECT course_no, description  
FROM course  
WHERE prerequisite = 350;
```

```
SQL> SELECT course_no, description  
2 FROM course  
3 WHERE prerequisite = 350;  
  
COURSE_NO DESCRIPTION  
-----  
430 JDeveloper Techniques  
450 DB Programming in Java
```

rem [Rischert, 2004,457]

```
SELECT course_no cno,  
description,  
prerequisite prereq,  
location loc, section_id  
FROM course LEFT OUTER JOIN section  
USING (course_no)  
WHERE prerequisite = 350;
```

```
SQL> SELECT course_no cno,  
2 description,  
3 prerequisite prereq,  
4 location loc, section_id  
5 FROM course LEFT OUTER JOIN section  
6 USING (course_no)  
7 WHERE prerequisite = 350;
```

CNO	DESCRIPTION	PREREQ
L507	SECTION_ID	
430	JDeveloper Techniques	350
450	DB Programming in Java	350
L507		109

rem [Rischert, 2004,460]

```
SELECT c.course_no cno, s.course_no sno,  
c.description,  
c.prerequisite prereq,  
s.location loc, s.section_id  
FROM (SELECT *  
FROM course  
WHERE prerequisite = 350) c LEFT OUTER JOIN  
(SELECT * FROM section  
WHERE location = 'L507') s  
ON (c.course_no = s.course_no);
```



```

SQL> SELECT c.course_no cno, s.course_no sno,
2   c.description,
3   c.prerequisite prereq,
4   s.location loc, s.section_id
5   FROM (SELECT *
6   FROM course
7   WHERE prerequisite = 350) c LEFT OUTER JOIN
8   (SELECT * FROM section
9   WHERE location = 'L507') s
10  ON (c.course_no = s.course_no);

```

CNO	SNO	DESCRIPTION	SECTION_ID
430		JDeveloper Techniques	
350			
450		450 DB Programming in Java	
350	L507		109

rem [Rischert, 2004,461]

```

SELECT c.course_no, c.description
FROM course c, section s
WHERE c.course_no = s.course_no(+)
AND s.course_no IS NULL;

```

```

SQL> SELECT c.course_no, c.description
2   FROM course c, section s
3   WHERE c.course_no = s.course_no(+)
4   AND s.course_no IS NULL;

```

COURSE_NO	DESCRIPTION
80	Structured Programming Techniques
430	JDeveloper Techniques

rem [Rischert, 2004,462]

```
SELECT city, state, zip,  
(SELECT COUNT(*)  
FROM student s  
WHERE s.zip = z.zip) AS student_count  
FROM zipcode z  
WHERE state = 'CT';
```

```
SQL> SELECT city, state, zip,  
2 (SELECT COUNT(*)  
3 FROM student s  
4 WHERE s.zip = z.zip) AS student_count  
5 FROM zipcode z  
6 WHERE state = 'CT';
```

CITY	ST	ZIP	STUDENT_COUNT
Ansonia	CT	06401	0
Middlefield	CT	06455	0
Oxford	CT	06483	1
New Haven	CT	06520	0
Bridgeport	CT	06605	1
Woodbury	CT	06798	1
Greenwich	CT	06830	3
Norwalk	CT	06850	1
Norwalk	CT	06851	1
Rowayton	CT	06853	1
Old Greenwich	CT	06870	1

CITY	ST	ZIP	STUDENT_COUNT
Ridgefield	CT	06877	1
Westport	CT	06880	2
Weston	CT	06883	0
Wilton	CT	06897	0
Stamford	CT	06902	1
Stamford	CT	06903	2
Stamford	CT	06905	1
Stamford	CT	06907	1

19 filas seleccionadas.

rem [Rischert, 2004,471]

```
SELECT student_id, section_id, numeric_grade
FROM grade g
WHERE grade_type_code = 'FI'
AND section_id = 86
AND numeric_grade < ANY
(SELECT numeric_grade
FROM grade
WHERE grade_type_code = 'MT'
AND g.section_id = section_id
AND g.student_id = student_id);
```

```
SQL> SELECT student_id, section_id, numeric_grade
2   FROM grade g
3   WHERE grade_type_code = 'FI'
4   AND section_id = 86
5   AND numeric_grade < ANY
6   (SELECT numeric_grade
7   FROM grade
8   WHERE grade_type_code = 'MT'
9   AND g.section_id = section_id
10  AND g.student_id = student_id);
```

STUDENT_ID	SECTION_ID	NUMERIC_GRADE
102	86	85
108	86	76
211	86	77

rem [Rischert, 2004,472]

```
SELECT DISTINCT a.student_id, a.last_name,  
a.street_address  
FROM student a, student b  
WHERE a.street_address = b.street_address  
AND a.zip = b.zip  
AND a.student_id <> b.student_id  
ORDER BY a.street_address;
```

STUDENT_ID	LAST_NAME
------------	-----------

STREET_ADDRESS
----------------

173 McGill
578 E 40th ST.

322 Archer
578 E 40th ST.

389 Leung
88 Sherman St.

STUDENT_ID	LAST_NAME
------------	-----------

STREET_ADDRESS
----------------

393 Martin
88 Sherman St.

22 filas seleccionadas.

rem [Rischert, 2004,474]

```
SELECT c1.course_no,  
SUBSTR(c1.description, 1,15) course_descr,  
C1.prerequisite,  
SUBSTR(c2.description,1,15) pre_req_descr  
FROM course c1, course c2  
WHERE c1.prerequisite = c2.course_no(+)  
ORDER BY 1;
```

COURSE_NO	COURSE_DESCR
420	Database System
25	Intro to Progra
430	JDeveloper Tech
350	JDeveloper Lab
450	DB Programming
350	JDeveloper Lab

30 filas seleccionadas.