

Ryan Paw
Week 1, Lab
ANA 660

Complete the following Lab Exercises in the textbook

1. Lab 3.2 Exercises (all the questions)
2. Lab 6.1 Exercises (all the questions)
3. Lab 10.1 Exercises (questions a, b, d, e, and f). Note: ignore alternative solutions presented on pages 414-416.
4. Lab 17.2 Exercises (questions a, b, and c)

I, Ryan Paw, finished and completed this week's reading requirements and lab activities. I created and ran the code in SQL Plus myself.

Lab 3.2 Exercises

- a)** Write a SELECT statement that lists each city and zip code in New York or Connecticut. Sort the results in ascending order by zip code.
- b)** Write a SELECT statement that lists course descriptions and their prerequisite course numbers, sorted in ascending order by description. Do not list courses that do not have a prerequisite.
- c)** Show the salutation, first name, and last name of students with the last name Grant. Order the results by salutation in descending order and by first name in ascending order.
- d)** Execute the following query. What do you observe about the last row returned by the query?

```
SELECT student_id, last_name  
FROM student  
ORDER BY last_name
```

3.2, a)

Code:

```
SQL> select city, zip  
2   from zipcode  
3   where state='NY' or state='CT'  
4   order by zip;
```

Output:

Garden City	11530
Hempstead	11550
West Hempstead	11552

CITY	ZIP
-----	-----
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 rows selected.

3.2, b)

Code:

```
SQL> select description, prerequisite
  2   from course
  3  where prerequisite is not null
  4  order by description;
```

Output:

DESCRIPTION	PREREQUISITE
Advanced Java Programming	122
Advanced Unix Admin	132
Basics of Unix Admin	130
DB Programming with Java	350
Database Design	420
Database System Principles	25
GUI Design Lab	20
Hands-On Windows	20
Intermediate Java Programming	120
Internet Protocols	310
Intro to Java Programming	80
Intro to Programming	140
Intro to SQL	20
Intro to Unix	310
Intro to the BASIC Language	25
Intro to the Internet	10
Java Developer I	122
Java Developer II	125
Java Developer III	350
Network Administration	130
Oracle Tools	220
DESCRIPTION	PREREQUISITE
PL/SQL Programming	80
Programming Techniques	204
Project Management	20
Systems Analysis	20
Unix Tips and Techniques	134
26 rows selected.	

3.2, c)

Code:

```
SQL> select salutation, first_name, last_name
  2  from student
  3  where last_name='Grant'
  4  order by salutation DESC, first_name ASC;
```

Output:

SALUT	FIRST_NAME	LAST_NAME
Ms.	Eilene	Grant
Ms.	Verona	Grant
Mr.	Omaira	Grant
Mr.	Scott	Grant

3.2, d)

```
241 TOURISH
248 Zapulla
331 Zopf
184 Zuckerberg
206 annunziato
268 rows selected.
```

The last_name in the last row is lowercase. Since it is lowercase, SQL organizes it at the bottom because it prioritizes the uppercase letters first.

Lab 6.1 Exercises

- Write a SELECT statement that determines how many courses do not have a prerequisite.
- Write a SELECT statement that determines the total number of students enrolled. Count each student only once, no matter how many courses the student is enrolled in.
- Determine the average cost for all courses. If the course cost contains a null value, substitute the value 0.
- Write a SELECT statement that determines the date of the most recent enrollment.

6.1, a)

Code:

```
SQL> select count(*)  
 2  from course  
 3  where prerequisite is null;
```

Output:

```
  COUNT(*)  
-----  
          4
```

6.1, b)

Code:

```
SQL> SELECT COUNT(DISTINCT student_id)  
 2  FROM enrollment;
```

Output:

```
  COUNT(DISTINCTSTUDENT_ID)  
-----  
                          165
```

6.1, c)

Code:

```
SQL> SELECT AVG(NVL(cost,0))  
 2  FROM course;
```

Output:

```
  AVG(NVL(COST,0))  
-----  
          1158.5
```

6.1, d)

Code:

```
SQL> select max(enroll_date)  
 2  from enrollment;
```

Output:

```

MAX(ENROL
-----
21-FEB-07

```

Lab 10.1 Exercises

a) Explain why Oracle returns an error message when you execute the following SELECT statement.

```

SELECT c.course_no, s.course_no, s.section_id,
       c.description, s.start_date_time
FROM course c, section s
WHERE c.course_no(+) = s.course_no(+)

```

b) Show the description of all courses with the prerequisite course number 350. Include in the result the location where the sections meet. Return course rows even if no corresponding row in the SECTION table is found.

d) Show all the city, state, and zip code values for Connecticut. Display a count of how many students live in each zip code. Order the result alphabetically by city. The result should look similar to the following output. Note that the column STUDENT_COUNT displays a zero when no student lives in a particular zip code.

CITY	ST	ZIP	STUDENT_COUNT
-----	--	----	-----
Ansonia	CT	06401	0
Bridgeport	CT	06605	1
...			
Wilton	CT	06897	0
Woodbury	CT	06798	1

19 rows selected.

e) Display the course number, description, cost, class location, and instructor's last name for all the courses. Also include courses where no sections or instructors have been assigned.

f) For students with the student ID 102 and 301, determine the sections they are enrolled in. Also show the numeric grades and grade types they received, regardless of whether they are enrolled or received any grades.

10.1, a)

```

SQL> select c.course_no, s.course_no, s.section_id,
2      c.description, s.start_date_time
3  from course c, section s
4  where c.course_no(+) = s.course_no(+);
where c.course_no(+) = s.course_no(+)
*
ERROR at line 4:
ORA-01468: a predicate may reference only one outer-joined table

```

Oracle is returning an error message because the outer join symbol (+) should only be used on 1 side of the equation. In the example, the outer join symbol (+) is used on both sides.

10.1, b)

Code:

```
SQL> select c.course_no, s.course_no, c.description, c.prerequisite, s.location, s.section_id
  2   from course c, section s
  3   where c.course_no = s.course_no(+)
  4   and c.prerequisite = 350;
```

Output:

CNO	SNO	DESCRIP	SEC_ID
PREREQ	LOC		
430		Java Developer III	
350			
450	450	DB Programming with Java	
350	L507		109

10.1, d)

Code:

```
SQL> select city, state, z.zip, count(s.zip) as student_count
  2   from zipcode z, student s
  3   where z.zip=s.zip(+) and state='CT'
  4   group by city, state, z.zip;
```

Output:

CITY	ST	ZIP	STUDENT_COUNT
Middlefield	CT	06455	0
New Haven	CT	06520	0
Norwalk	CT	06850	1
Westport	CT	06880	2
Woodbury	CT	06798	1
Ridgefield	CT	06877	1
Greenwich	CT	06830	3
Stamford	CT	06905	1
Wilton	CT	06897	0
Bridgeport	CT	06605	1
Rowayton	CT	06853	1
Old Greenwich	CT	06870	1
Ansonia	CT	06401	0
Oxford	CT	06483	1
Norwalk	CT	06851	1
Weston	CT	06883	0
Stamford	CT	06903	2
Stamford	CT	06902	1
Stamford	CT	06907	1

19 rows selected.

10.1, e)

Code:

```
SQL> select c.course_no, c.description, c.cost, s.location, i.last_name
  2  from course c, section s, instructor i
  3  where c.course_no=s.course_no(+) and s.instructor_id=i.instructor_id(+)
  4  order by c.course_no;
```

Output:

```
L214                                WOJICK
      240 Intro to the BASIC Language          1095
L509                                Hanks
      310 Operating Systems                    1195
L507                                Schorin
      330 Network Administration              1195
L511                                Pertez
      350 Java Developer II                   1195
L509                                Frantzen
      350 Java Developer II                   1195
L509                                Morris

COURSE_NO DESCRIPTION                                COST
-----
LOCATION                                LAST_NAME
-----
L214      350 Java Developer II                   1195
                                Smythe
M311      420 Database System Principles          1195
                                Lowry
      430 Java Developer III                     1195
      450 DB Programming with Java
L507                                Hanks

80 rows selected.
```

10.1, f)

Code:


```
SQL> select student_id, section_id, grade_type_code, numeric_grade
  2  from student
  3  left outer join enrollment using (student_id)
  4  left outer join grade using (student_id, section_id)
  5  where student_id in (102,301);
```

Output:

STUDENT_ID	SECTION_ID	GR	NUMERIC_GRADE
102	86	FI	85
102	86	HM	90
102	86	HM	99
102	86	HM	82
102	86	HM	82
102	86	MT	90
102	86	PA	85
102	86	QZ	90
102	86	QZ	84
102	86	QZ	97
102	86	QZ	97
102	89	FI	92
102	89	MT	91
301			

14 rows selected.

Lab 17.2 Exercises

a) Describe the effect of the following SQL statement and its resulting output.

```
SELECT salutation AS SALUTATION, SUBSTR(phone, 1,3)
      AS "Area Code",
      TO_CHAR(registration_date, 'MON') AS "Reg.Month",
      COUNT(*)
FROM student
WHERE SUBSTR(phone, 1,3) IN ('201','212','718')
  AND salutation IN ('Mr.', 'Ms.')
GROUP BY ROLLUP (salutation, SUBSTR(phone, 1,3),
                TO_CHAR(registration_date, 'MON'))
```

SALUT	Area Code	Reg.Month	COUNT(*)
Mr.	201	FEB	34
Mr.	201	JAN	9
Mr.	201		43
Mr.	212	FEB	1
Mr.	212	JAN	1
Mr.	212		2
Mr.	718	FEB	72
Mr.	718	JAN	17
Mr.	718		89
Mr.			134
Ms.	201	FEB	27
Ms.	201	JAN	5
Ms.	201		32
Ms.	212	FEB	2
Ms.	212	JAN	1
Ms.	212		3
Ms.	718	FEB	52
Ms.	718	JAN	13
Ms.	718		65
Ms.			100
			234

21 rows selected.

b) Answer the following questions about the result set.

How many female students are there in total?

How many male students live in area code 212?

What is the total number of students?

How many female students live in the area code 718 and registered in January?

c) If the CUBE operator is used on the query in exercise a instead of ROLLUP, how many different combinations of groups do you get? List the groups.

17.2, a)

The ROLLUP statement is similar to GROUP BY, and it creates a hierarchy with the input columns. The first group is by salutation, area code, registration date. The second group is by salutation and area code. The third group is by salutation. The last row shows that total count of all the rows.

17.2. b)

There are 100 female students in total. This is found on the 2nd to last row in the provided output.

There are 2 male students that live in area code 212. This is found on the 6th row in the provided output.

There are 13 female students who live in area code 718 and registered in January. This is found on the 18th row in the provided output.

Code:

Output:

```
36 rows selected.
```

Similar to ROLLUP, CUBE generates subtotals and creates subtotals for all combinations of grouping columns specified in the GROUP BY clause. There are 3 columns. Therefore, $2^3 = 8$ combinations of groups.