7.1.1. Write Subqueries in the WHERE and HAVING Clauses

a) Write a SQL statement that displays the first and last names of students who registered

first.

**SELECT first\_name, last\_name**

**FROM student**

**WHERE registration\_date =**

**(SELECT MIN(registration\_date)**

**FROM student)**

b) Show the sections with the lowest course cost and a capacity equal to or lower than the

average capacity. Also display the course description, section number, capacity, and

cost.

**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)**

c) Select the course number and total capacity for each course. Show only the courses

with a total capacity less than the average capacity of all the sections.

**SELECT course\_no, SUM(capacity)**

**FROM section**

**GROUP BY course\_no**

**HAVING SUM(capacity) <**

**(SELECT AVG(capacity)**

**FROM section)**

d) Choose the most ambitious students: Display the STUDENT\_ID for students enrolled in

the most sections.

**SELECT student\_id, COUNT(\*)**

**FROM enrollment**

**GROUP BY student\_id**

**HAVING COUNT(\*) =**

**(SELECT MAX(COUNT(\*))**

**FROM enrollment**

**GROUP BY student\_id)**

7.1.2. Write Subqueries Returning Multiple Rows

a) Select the STUDENT\_ID and SECTION\_ID of enrolled students living in zip code 06820.

**SELECT student\_id, section\_id**

**FROM enrollment**

**WHERE student\_id IN**

**(SELECT student\_id**

**FROM student**

**WHERE zip = '06820')**

b) Display the course number and course description of the courses taught by instructor

Fernand Hanks.

**SELECT course\_no, description**

**FROM course**

**WHERE course\_no IN**

**(SELECT course\_no**

**FROM section**

**WHERE instructor\_id IN**

**(SELECT instructor\_id**

**FROM instructor**

**WHERE last\_name = 'Hanks'**

**AND first\_name = 'Fernand'))**

c) Select the last name and first name of students not enrolled in any class.

**SELECT last\_name, first\_name**

**FROM student**

**WHERE student\_id NOT IN**

**(SELECT student\_id**

**FROM enrollment)**

7.1.3. Write Subqueries Returning Multiple Columns

a) Determine the STUDENT\_ID and last name of students with the highest FINAL\_GRADE

for each section. Also include the SECTION\_ID and the FINAL\_GRADE columns in the

result.

**columns in the outer query.**

**SELECT s.student\_id, s.last\_name, e.final\_grade,**

**e.section\_id**

**FROM enrollment e, student s**

**WHERE e.student\_id = s.student\_id**

**AND (e.final\_grade, e.section\_id) IN**

**(SELECT MAX(final\_grade), section\_id**

**FROM enrollment**

**GROUP BY section\_id)**

b) Select the sections and their capacity where the capacity equals the number of students

enrolled.

**SELECT section\_id, capacity**

**FROM section**

**WHERE (section\_id, capacity) IN**

**(SELECT section\_id, COUNT(\*)**

**FROM enrollment**

**GROUP BY section\_id)**