**Homework Assignment # 3**

**7.3E Write an SQL query that retrieves all pairs of suppliers who supply the**

**same product, along with their product purchase price if applicable.**

Ans:

SELECT a.supplier\_id AS supplier1,

b.supplier\_id AS supplier2,

a.product\_id,

a.purchase\_price AS price1,

b.purchase\_price AS price2

FROM supplies a, supplies b

WHERE a.product\_id = b.product\_id

AND a.supplier\_id < b.supplier\_id

AND EXISTS (SELECT 1

FROM supplies c

WHERE c.product\_id = a.product\_id

AND c.supplier\_id = b.supplier\_id);

**7.9E Write an SQL query using EXISTS to retrieve the supplier numbers and**

**names of all suppliers that do not have any outstanding orders.**

Ans:

**SELECT**

**supplier\_id,**

**supplier\_name**

**FROM**

**suppliers s**

**WHERE**

**NOT EXISTS (**

**SELECT 1**

**FROM orders o**

**WHERE o.supplier\_id = s.supplier\_id**

**AND o.status = 'outstanding'**

**);**

**7.16E Write a correlated SQL query to retrieve the three lowest product**

**numbers.**

**ANS: SELECT product\_id**

**FROM products p1**

**WHERE 3 > (**

**SELECT COUNT(\*)**

**FROM products p2**

**WHERE p2.product\_id < p1.product\_id**

**)**

**ORDER BY product\_id ASC**

**LIMIT 3;**

**2.1 Write the following queries in SQL, using the university schema.**

**a. Find the titles of courses in the Comp. Sci. department that have 3 credits.**

ANS: SELECT title

FROM course

WHERE dept\_name = 'Comp. Sci.' AND credits = 3;

**b. Find the IDs of all students who were taught by an instructor named Einstein; make sure**

**there are no duplicates in the result.**

ANS: SELECT DISTINCT s.ID

FROM student s

JOIN takes t ON s.ID = t.ID -- "takes" would be the relationship table

JOIN teaches tc ON t.sec\_id = tc.sec\_id AND

t.semester = tc.semester AND

t.year = tc.year AND

t.course\_id = tc.course\_id

JOIN instructor i ON tc.ID = i.ID

WHERE i.name = 'Einstein';

**c. Find all instructors earning the highest salary (there may be more than one with the same**

**salary).**

ANS:

SELECT ID, name, dept\_name, salary

FROM instructor

WHERE salary = (SELECT MAX(salary) FROM instructor);

**d. Find the maximum enrollment, across all sections, in Autumn 2009.**

ANS:

SELECT MAX(enrollment) AS max\_enrollment

FROM (

SELECT COUNT(ID) AS enrollment

FROM takes

WHERE semester = 'Autumn' AND year = 2009

GROUP BY course\_id, sec\_id, semester, year

) AS section\_enrollments;

**e. Find the sections that had the maximum enrollment in Autumn 2009.**

**ANS:**

**SELECT course\_id, sec\_id, COUNT(ID) AS enrollment**

**FROM takes**

**WHERE semester = 'Autumn' AND year = 2009**

**GROUP BY course\_id, sec\_id, semester, year**

**HAVING COUNT(ID) = (**

**SELECT MAX(section\_count)**

**FROM (**

**SELECT COUNT(ID) AS section\_count**

**FROM takes**

**WHERE semester = 'Autumn' AND year = 2009**

**GROUP BY course\_id, sec\_id, semester, year**

**) AS counts**

**);**

**3 a. Find all customers of the bank who have an account but not a loan. (use except statement)**

**ANS:**

SELECT customer\_name

FROM depositor

EXCEPT

SELECT customer\_name

FROM borrower;

**b. Find the names of all customers who live on the same street and in the same city**

**as “Smith”.**

**ANS:**

SELECT customer\_name

FROM customer

WHERE customer\_street = (SELECT customer\_street FROM customer WHERE customer\_name = 'Smith')

AND customer\_city = (SELECT customer\_city FROM customer WHERE customer\_name = 'Smith')

AND customer\_name <> 'Smith';

**4a. Find the names, street addresses, and cities of residence of all employees who work**

**for First Bank Corporation and earn more than $10,000.**

ANS :

SELECT e.employee\_name, e.street, e.city

FROM employee e, works w

WHERE e.employee\_name = w.employee\_name

AND w.company\_name = 'First Bank Corporation'

AND w.salary > 10000;

**b. Find all employees in the database who earn more than each employee of Small**

**Bank Corporation.**

ANS:

SELECT DISTINCT e.employee\_name

FROM employee e, works w1

WHERE e.employee\_name = w1.employee\_name

AND w1.salary > ALL (

SELECT w2.salary

FROM works w2

WHERE w2.company\_name = 'Small Bank Corporation'

);

**c. Assume that the companies may be located in several cities. Find all companies located in**

**every city in which Small Bank Corporation is located.**

ANS:

SELECT c1.company\_name

FROM company c1

JOIN company c2 ON c1.city = c2.city

WHERE c2.company\_name = 'Small Bank Corporation'

GROUP BY c1.company\_name

HAVING COUNT(DISTINCT c1.city) = (

SELECT COUNT(DISTINCT city)

FROM company

WHERE company\_name = 'Small Bank Corporation'

);

d. Find the company that has the most employees.

ANS: SELECT company\_name

FROM works

GROUP BY company\_name

HAVING COUNT(employee\_name) >= ALL (

SELECT COUNT(employee\_name)

FROM works

GROUP BY company\_name

);

e**. Find those companies whose employees earn a higher salary, on average, than the average**

salary at First Bank Corporation.

ANS:

SELECT w.company\_name

FROM works w

GROUP BY w.company\_name

HAVING AVG(w.salary) > (

SELECT AVG(salary)

FROM works

WHERE company\_name = 'First Bank Corporation'

);

**4.2 Consider the relational database of Figure 6. Give an expression in SQL for each of**

**the following queries.**

1. **Modify the database so that Jones now lives in Newtown.**

ANS: UPDATE employee

SET city = 'Newtown'

WHERE employee\_name = 'Jones';

**b. Give all managers of First Bank Corporation a 10 percent raise unless the salary becomes**

**greater than $100,000; in such cases, give only a 3 percent raise. (use case statement)**

UPDATE works

SET salary = CASE

WHEN salary \* 1.10 > 100000 THEN salary \* 1.03

ELSE salary \* 1.10

END

WHERE company\_name = 'First Bank Corporation'

AND employee\_name IN (

SELECT manager\_name

FROM manages

);