**CIS 422 DBMS**

**SQL Training and Exercises Part3**

**Section 1: Subquery**

1. Plain Subquery

In plain subquery, the outer query makes use of the result returned from the subquery. The outer query depends on the subquery for its value. However, the subquery does not depend on the outer query.

Find the Id, first name and second name for all employees who are working for all departments located at the location whose id is 1700.

SELECT employee\_id, first\_name, last\_name

FROM employees E

WHERE department\_id IN

(SELECT department\_id

FROM departments D

WHERE location\_id = 1700

);

Find the Id, first name and second name for all employees who have the highest salary.

SELECT employee\_id, first\_name, last\_name, salary

FROM employees

WHERE salary =

(SELECT MAX(salary)

FROM employees

);

Find the Id, first name, second name and salary for all employees whose salaries are greater than the lowest salary of every department.

SELECT employee\_id, first\_name, last\_name, salary

FROM employees

WHERE salary >= ALL

(SELECT MIN(salary)

FROM employees

GROUP BY department\_id

);

Calculate the average of average salary of all departments.

SELECT AVG(avgSal)

FROM (

SELECT AVG(salary) avgSal

FROM employees

GROUP BY department\_id) deptSal;

Find the Id, first name, second name, salary, average salary, and the difference between the salary of each employee and the average salary for all employees

SELECT employee\_id, first\_name, last\_name, salary,

(SELECT AVG(salary) FROM employees) avgSal,

salary - (SELECT AVG(salary) FROM employees) difference

FROM

employees;

1. Correlated Subquery

A correlated subquery is a subquery that uses the values from the outer query. Also, a correlated subquery may be evaluated once for each row selected by the outer query. Because of this, a query that uses a correlated subquery may be slow.

Find the Id, first name, second name, salary and department id of all employees whose salary is higher than the average salary of the employees in their departments.

SELECT employee\_id, first\_name, last\_name, salary, department\_id

FROM employees E

WHERE salary >

(SELECT AVG(salary)

FROM employees

WHERE department\_id = e.department\_id

);

Find the Id, first name, second name, salary, department name and the average salary of all employees in their departments.

SELECT employee\_id, first\_name, last\_name, department\_name, salary,

(SELECT AVG(salary)

FROM employees

WHERE department\_id = e.department\_id) avgSalDepts

FROM employees e

INNER JOIN departments d ON d.department\_id = e.department\_id;

Find the Id, first name and second name of all employees who have no dependents.

SELECT employee\_id, first\_name, last\_name

FROM employees e

WHERE NOT EXISTS(

SELECT \*

FROM dependents d

WHERE d.employee\_id = e.employee\_id

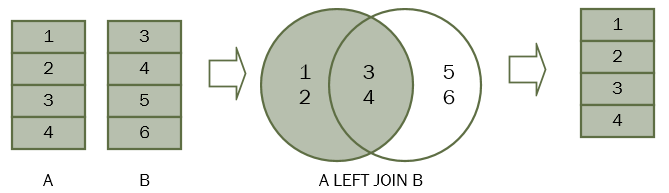
);

**Section 2: Joining Multiple Tables**

1. LEFT OUTER JOIN

The left join, however, returns all rows from the left table whether or not there is a matching row in the right table.

Example: When we join table A with table B, all the rows in table A (the left table) are included in the result set whether there is a matching row in the table B or not.



In SQL, we use the following syntax to join table A with table B.

SELECT

A.n

FROM

A

LEFT JOIN B ON B.n = A.n;

Find the name of countries that do not have any locations in the locations table.

SELECT country\_name

FROM countries c

LEFT JOIN locations l ON l.country\_id = c.country\_id

WHERE l.location\_id IS NULL;

Find the name of every employee along with his/her manager and also find those employees who have no manager.

SELECT

e.first\_name, e.last\_name AS employee,

m.first\_name, m.last\_name AS manager

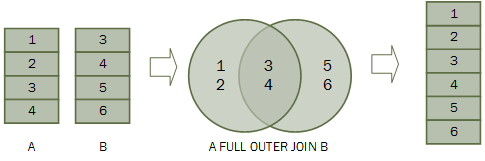
FROM employees e

LEFT JOIN employees m ON m.employee\_id = e.manager\_id;

1. FULL OUTER JOIN

In theory, a full outer join is the combination of a left join and a right join. The full outer join includes all rows from the joined tables whether or not the other table has the matching row.

The following Venn diagram illustrates the full outer join of two tables.



The following statement illustrates the syntax of the full outer join of two tables:

SELECT column\_list

FROM A

FULL OUTER JOIN B ON B.n = A.n;

Return the name of each employee with his/her department, also return each department that has no employees and every employee who does not belong for any department.

SELECT department\_name, first\_name, last\_name

FROM employees

FULL JOIN departments ON departments.department\_id = employees.department\_id;

Find all department, which has no employees.

SELECT department\_name, first\_name, last\_name

FROM employees

FULL OUTER JOIN departments ON departments.department\_id = employees.department\_id

WHERE first\_name IS NULL AND last\_name IS NULL;