DBS301 Lab 5 Solutions - Simple Joins

1)      Display the department name, city, street address and postal code for all departments sorted by city and department name.

SELECT department\_name, city, street\_address, postal\_code

FROM departments JOIN locations USING (location\_id)

ORDER BY city, department\_name;

2)      Display full name of the employees using format of Last, First, their hire date and salary together with their department name and city, but only for departments which names start with an A or S sorted by department name and employee name.

SELECT last\_name || ', ' || first\_name as "name", hire\_date, salary, department\_name, city

FROM employees e JOIN departments d USING (department\_id)

JOIN locations l USING (location\_id)

WHERE department\_name like 'A%' OR department\_name like 'S%'

ORDER BY department\_name, name;

3)      Display the full name of the manager of each department in states/provinces of Ontario, California and Washington along with the department name, city, postal code and province name. Sort the output by city and then by department name.

SELECT first\_name || ' ' || last\_name as "Manager", department\_name, city, postal\_code,

state\_province

FROM employees e JOIN departments d USING (manager\_id)

JOIN locations l USING (location\_id)

WHERE LOWER(state\_province) IN ('ontario', 'california', 'washington')

ORDER BY city, department\_name;

4)      Display employee’s last name and employee number along with their manager’s last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr# respectively.

SELECT e.last\_name as "Employee", e.employee\_id "Emp#", m.last\_name as "Manager", e.manager\_id as "Mgr#"

FROM employees e LEFT OUTER JOIN employees m ON (e.manager\_id = m.employee\_id);

NOTE: left join required to include employees without value for manager\_id since manager\_id in employees table permits nulls

**DBS301 Lab 6 Solutions – Group Functions**

0.      Make sure you exist as an employee with a NULL salary in department 90.

INSERT INTO employees VALUES (999, 'My', 'Name', 'MyEmail', NULL, SYSDATE, 'AC\_ACCOUNT', NULL, 0.2, NULL, 90);

1.      display the highest, lowest, total and average salary (of all salaried employees) and average salary of all employees

SELECT MAX(salary), MIN(salary), SUM(salary), ROUND(AVG(salary),2) AS "Average of Salaried", ROUND( AVG(NVL(salary, 0)),2) AS "Average of All"

FROM employees;

2.      display the highest, lowest, total and average projected compensation of all employees in a department to be provided at run-time where compensation is sum of salary and commission (where commission = commission\_pct \* projected sales) and projected sales is also to be provided at runtime

SELECT MAX((NVL(salary,0) + (NVL(commission\_pct,0) \* '&&projected\_sales'))) AS "MAX", MIN((NVL(salary,0) + (NVL(commission\_pct,0) \* '&&projected\_sales'))) AS "MIN",

SUM((NVL(salary,0) + (NVL(commission\_pct,0) \* '&&projected\_sales'))) AS "Total", ROUND(AVG((NVL(salary,0) + (NVL(commission\_pct,0) \* '&&projected\_sales'))),2) AS "Average"

FROM employees

WHERE department\_id = '&dept';

3.      display the highest, lowest, total and average salary (of all salaried employees) and average salary of all employees in a city to be provided at run-time

SELECT MAX(salary), MIN(salary), SUM(salary), ROUND(AVG(salary),2) AS “Avg of Salaried Employees”, ROUND(AVG(NVL(salary,0)),2) AS “Avg of All”

FROM employees

JOIN departments USING (department\_id)

JOIN locations USING (location\_id)

WHERE UPPER(city) = UPPER('&city');

4.       Show the number of employees with a manager and the number of employees without a manager.

SELECT COUNT(manager\_id) AS "# EMPs with a MGR", COUNT(\*)-COUNT(manager\_id) AS "# EMPs without a MGR"

FROM employees;

## Lab 7 Solutions - Subqueries

1.SET AUTOCOMMIT ON (do this each time you log on) so any updates, deletes and inserts are automatically committed before you exit from Oracle.

2.Make sure you exist as an employee with a NULL salary and 0.2 commission\_pct in department 90.

3.Change the salary of the employees with a last name of Matos and Whalen to be 2500.

  UPDATE employees  
    SET salary = 2500  
    WHERE last\_name = 'Matos' OR last\_name = 'Whalen';

You must use subqueries for these questions (must minimize the number of tables being used in the main query)

4.Display the last names of all employees who are in the same department as the employee named Abel.

SELECT last\_name  
    FROM employees  
    WHERE department\_id = (SELECT department\_id

FROM employees

WHERE LOWER(last\_name) = 'abel');  
NOTE: an = should be used if only 1 employee is to be matched; otherwise IN should be used

5.Display the last name of the lowest paid employee(s)

SELECT last\_name  
    FROM employees  
    WHERE salary = (SELECT MIN(salary)

FROM employees);

6.Display the city that the lowest paid employee(s) are located in.

SELECT city FROM locations

WHERE location\_id IN (SELECT location\_id

FROM departments

where department\_id IN (SELECT department\_id

FROM employees

where salary=( SELECT MIN(salary)

FROM employees)));

7.Display the last name of the lowest paid employee(s) in each department

SELECT department\_id, last\_name  
    FROM employees  
    WHERE (department\_id, salary) IN (SELECT department\_id, MIN(salary) FROM employees

GROUP BY department\_id);

8.Display the last name of the lowest paid employee(s) in each city

SELECT city, last\_name  
    FROM employees  
    JOIN departments USING (department\_id)  
    JOIN locations USING (location\_id)  
    WHERE (city, salary) IN (SELECT city, MIN(salary) FROM employees

JOIN departments USING (department\_id)   
          JOIN locations USING (location\_id)

GROUP BY city);

**DBS301 Lab 3 Solutions – Basic SELECT statements**

1. Display all unique combinations of department codes and job ids of all employees.

SELECT DISTINCT department\_id, job\_id  
FROM employees;

1. Display the employee\_id, last name and salary of employees earning from $8000 to $15,000 whose job id is IT\_PROG or SA\_REP. Sort the output by top salaries first and then alphabetically by last name.

SELECT employee\_id, last\_name, salary  
FROM employees  
WHERE salary BETWEEN 8000 AND 15000 AND job\_id IN ('IT\_PROG', 'SA\_REP')   
ORDER BY salary, last\_name;

1. Display the last name, job\_id and salary of employees earning more than $10,000 hired before 1998. List the most recently hired employees first.

SELECT last\_name, job\_id, salary  
FROM employees  
WHERE salary > 10000 AND hire\_date < '01-JAN-98'  
ORDER BY hire\_date DESC;

1. Display the job id and full names (as 1 column with a heading of Managers) of employees whose job\_id contains the phrase MAN (in upper or lower case).

SELECT job\_id || first\_name||' '||last\_name as "Managers"  
FROM employees  
WHERE UPPER(job\_id) LIKE '%MAN%);

1. For each employee in departments 20, 50 and 80 who is not paid a commission display their last name, first name, salary and salary increased by 5% and expressed as a whole number. Label the last column Proposed Salary. Also add a column that subtracts the old salary from the new salary and multiplies by 12. Label this column Proposed Annual Increase.

SELECT last\_name, first\_name, salary, ROUND(salary\*1.05) AS "Proposed Salary", 12\*(salary\*0.05) AS "Proposed Annual Increase"  
FROM employees  
WHERE department\_id IN (20, 50, 80) AND commission\_pct IS NULL;

1. For each employee in department 80 display their last name, first name, salary, commission and commission multiplied by salary. Label the last column Estimated Commission.

SELECT last\_name, first\_name, salary, commission\_pct,  
commission\_pct\*salary AS "Estimated Commission"  
FROM employees  
WHERE department\_id = 80;

1. List all departments that have a manager.

SELECT department\_id  
FROM departments  
WHERE manager\_id IS NOT NULL;

1. Create a query that prompts for department id and displays the last\_name and job\_id of employees in that department except for employees whose job id contains the phrases PRES or MAN

SELECT last\_name, job\_id   
FROM employees  
WHERE department\_id = &department\_id  
AND (job\_id NOT LIKE '%PRES%' AND job\_id NOT LIKE '%MAN%';

1. Display the employee’s Full Name and Job Title for all employees whose last name ends with an sand first name starts with a *C* or *K* sorted by last name under a heading of *Employee Job Listing* in the following format: *lname, fname* is *job*

SELECT first\_name||' '||last\_name||' is '||job\_id AS "Employee Job Listing"  
FROM employees  
WHERE last\_name LIKE '%s' AND (first\_name LIKE 'C%' OR first\_name LIKE 'K%')  
ORDER BY last\_name;

1. Display the city names, country codes and state province names, but only for those cities that start with an *S* and have at least 8 characters in their name. If a city does not have a province name assigned, then display *Unknown Province.*

SELECT city, country\_id, NVL(state\_province, 'Unknown Province') as "State/Province"  
FROM locations  
WHERE city LIKE 'S%' AND LENGTH(city) > 7;