**6G4Z0016 Databases**

**Introduction to MariaDB Databases and SQL**

**Part 5: Functions and Joining**

**Aim**

The aim of this session is to be able to write SQL queries involving:

* Row-functions to alter the data
* Join clauses to combine data from more than 1 table

**Activities**

**Part 1 Functions**

1. In SQL, the CONCAT function is used to join string parameters together. Type in the following query and execute it.

SELECT employee\_id, CONCAT(first\_name, last\_name) AS "Employee Name"

FROM employees

WHERE department\_id = 90;

Observe what is shown on the screen. Notice that the column heading for the concatenated values is “Employee Name”.

1. The ROUND function rounds a column, expression or value to *n* decimal places. Write a query that displays an employee’s first name, last name and their salary divided by 1000 for all employees working in departments 50 and 90. Round the salary off to 0 decimal places. Hint: You will need to use the department\_id column in the WHERE clause to restrict the result returned. Record your answer in the box below.
2. Type in and execute the following query and observe how a date is displayed.

SELECT last\_name, hire\_date

FROM employees

WHERE hire\_date < '1988-02-01';

1. Type in an execute the following query

SELECT CURDATE();

What does this query do? Record your answer in the box below.

1. Which function would you use if you wanted to calculate how many months an employee had been hired?

Write an SQL query that displays the number of months an employee has been hired. Your query should display the employee’s ID and salary and label the number of month’s column appropriately. Order the results in descending order of salary. Record your answer in the box below.

1. Amend the query you have just written in (5) to add an additional column that calculates the number of years an employee has worked. Round up the number of months to whole years and label the column appropriately.
2. The Human Resources Department needs a report to display the employee number, last name, salary and the salary increased by 15.5% (expressed as a whole number) for each employee. Label the column New Salary. Write a query to fulfil these requirements.

**Part 2 Joins**

When writing a query, you will often need to extract data from 2 or more database tables. In order to do this, you will need to JOIN tables together based upon their Primary and Foreign Key Relationships.

For this unit you need to know how to perform a JOIN between tables.

You can join tables automatically based upon the columns in two tables that have matching types and names using the NATURAL JOIN keywords. Type in the following query and execute it.

SELECT department\_id, department\_name, location\_id, city  
FROM departments NATURAL JOIN locations;

In this example, the LOCATIONS table is joined to the DEPARTMENTS table by the column location\_id. This is because location\_id is the only column of the same name in both tables. To confirm this, describe the structure of each table as follows.

DESCRIBE departments;

DESCRIBE locations;

1. Write a NATURAL JOIN to display the employees first name, last name and the name of the department that they work in. Record your answer in the box below.

What is the name of the column that joins the two tables together?

If several columns have the same names but the data types do not match a natural join will produce an error. Moreover, natural joins can produce unpredictable results. For these reasons, INNER JOINs are more frequently used.

INNER JOINs always involve specifying the columns that the tables join on. For INNER JOINs you do not have to write “INNER JOIN”. If you just write “JOIN” it is also an INNER JOIN. But it is better practice to be explicit and write INNER JOIN. An INNER JOIN can be applied by using two methods.

Method 1: INNER JOINs that use the USING clause to specify which columns should be used in the join.

Type in the following query and execute it:

SELECT employee\_id, last\_name, location\_id, department\_id  
FROM employees   
INNER JOIN departments USING (department\_id);

Observe that department\_id is the common column. Without the join we cannot find out the location\_id of an employee, because the location information comes from the departments table.

Method 2: This uses the ON clause to specify columns to join. The ON clause is necessary when the common columns have different names in the two tables.

Type in the following query and execute it:

SELECT e.employee\_id, e.last\_name, e.department\_id, d.department\_id, d.location\_id, d.department\_id

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

In this example, when a department\_id in the EMPLOYEES table equals a department\_id in the DEPARTMENTS table, the row is returned. Note how the tables are given aliases (e.g. e for employees table) in the FROM clause of the SELECT statement. MariaDB needs to know exactly what attribute you are talking about.

1. Try removing the “d.” from the start of “d.department\_id” in the SELECT clause of the query. What error do you get?

Each time an attribute that appears in both tables is referred to the table of origin also has to be specified. The use of table alias is necessary to qualify the matching column names.

1. Using the ON clause, write a query that displays the employee\_id, city and department name. *Hint: You will have to join three tables: employees, departments and locations to obtain all the information.*
2. Amend your answer from the previous question and rewrite the query using the USING clause.
3. The HR department needs a report of employees working in the city of Toronto. Write a query to display the last name, job, department number and department name for all employees who work in Toronto.
4. Write a query to display employee’s last name and employee id, along with their manager’s last name and manager number. Label the columns Employee, Emp#, Manager and Mgr# respectively. This query is challenging and requires a self-join! A self-join is a join between a table and itself. The database system makes a copy of the table to join it with itself.

*Hint: It is possible to give the same table two different aliases that alter depending upon what information you are extracting.*