Database Management Systems

ICT1222

MySQL

Department of ICT
Faculty of Technology
University of Ruhuna

Practical 04

- Create the following tables and details are given in the document “Student_Marks”.
- Do all the exercises provided in this document.
- Make a word document including cmd query screenshot with the output screenshot for each exercise in the document.
- Save it as “TGXXX.docs” and upload it to the moodle.

MySQL Aggregate Functions

- Aggregate functions allow you to perform a calculation on a set of records and return a single value.

- MIN()
  - Return the lowest value (minimum) in a set of non-NULL values.
- MAX()
  - Return the highest value (maximum) in a set of non-NULL values.
- SUM()
  - Return the summation of all non-NULL values a set.
- COUNT()
  - Return the number of rows in a group, including rows with NULL values.
- AVG()
  - Return the average of non-NULL values.
- There are many other functions available in MySQL. Please refer the Reference manual of respective version

https://dev.mysql.com/doc/refman/8.0/en/sql-function-reference.html

SQL MIN() Function

- The MIN() function returns the smallest value of the selected column

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

- Exercise 1
  - Get minimum ict marks from ict_marks table

SQL MAX() Function

- The MAX() function returns the largest value of the selected column

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```

- Exercise 2
  - Get maximum ict marks from ict_marks table

SQL SUM() Function

- The SUM() function returns the total sum of a numeric column.

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

- Exercise 3
  - Find the total of ict marks of all students

SQL COUNT() Function

- The COUNT() function returns the number of rows that matches a specified criteria

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

- Exercise 4
  - Count how many records are in student_data table
  - How many students got marks more than 50

SQL AVERAGE() Function

- The AVG() function returns the average value of a numeric column.

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

- Exercise 5
  - Find average marks of students

Exercise 5

- Get the minimum age of students from StudentDetails table.
- Get the maximum age of students from StudentDetails table.
- Get the sum of the ages of students from StudentDetails table.
- Get the avetage age of the students from StudentDetails table.
- Get the count of the total students from StudentDetails table.

MySQL Comparison Functions

- Comparison functions allow you to compare values in MySQL.

- ISNULL()
  - return 1 if the argument is NULL, otherwise, return zero.
- GREATEST()
  - Take n arguments and return the greatest value of the n arguments respectively.
- LEAST()
  - Take n arguments and return the least values of the n arguments respectively.

- There are many other functions available in MySQL. Please refer the Reference manual of respective version

https://dev.mysql.com/doc/refman/8.0/en/sql-function-reference.html

SQL ISNULL() Function

- The ISNULL() function returns 1 if the argument is NULL, otherwise, return zero.

SELECT ISNULL(*value*);

- Exercise 6
  - SELECT ISNULL(NULL);
  - SELECT ISNULL(1);

SQL GREATEST() & LEAST() Functions

- GREATEST
- Take n arguments and return the greatest value of the n arguments respectively.
- LEAST
- Take n arguments and return the least values of the n arguments respectively.

```
SELECT GREATEST(values);
SELECT LEAST(values);
```

- Exercise 7
  - SELECT GREATEST(81,50,15);
  - SELECT LEAST(81,50,15);
  - Try with value set with a NULL value

MySQL Math Functions

- These functions are the representations of most commonly used math functions in MySQL.

- ROUND()
  - Rounds a number to a specified number of decimal places
- MOD()
  - Returns the remainder of a number divided by another number
- CEIL()
  - Returns the smallest integer value greater than or equal to the input number
- FLOOR()
  - Returns the largest integer value not greater than the argument
- ABS()
  - Returns the absolute value of a number
- There are many other functions available in MySQL. Please refer the Reference manual of respective version

https://dev.mysql.com/doc/refman/8.0/en/sql-function-reference.html

SQL ROUND() Function

- Rounds a number to a specified number of decimal places.

SELECT ROUND(*number,decimal places*);

- Exercise 8
  - SELECT ROUND(10.56);
  - SELECT ROUND(10.56, 1);
  - Try minus decimal places
    - ex: SELECT ROUND(11.56789, -3);

SQL MOD() Function

- Returns the remainder of a number divided by another number

SELECT ROUND(dividend, divisor);

  - dividend is a literal number or a numeric expression to divide.
  - divisor is a literal number or a numeric expression by which to divide the dividend.
- Exercise 9
  - SELECT MOD(21/7);
  - SELECT MOD(56/ (3*2));
  - Try different expressions/functions for divisor

SQL CEIL() Function

- Returns the smallest integer value greater than or equal to the input number

```
SELECT  CEIL(number);
```

- Exercise 10
  - SELECT CEIL(65.3);
  - SELECT CEIL(65.7);
  - Try different expressions/functions and their combinations as number

SQL FLOOR() Function

- Returns the largest integer value not greater than the argument

```
SELECT  FLOOR(number);
```

- Exercise 11
  - SELECT FLOOR(65.3);
  - SELECT FLOOR(65.7);
  - Try different expressions/functions and their combinations as number

SQL ABS() Function

- Returns the absolute value of a number

```
SELECT  ABS(number);
```

- Exercise 12
  - SELECT ABS(-10);
  - SELECT ABS(10);
  - Try different expressions/functions and their combinations as number

MySQL String Functions

- These are the most used MySQL string functions that allow you to manipulate character string data effectively.

- CONCAT()
  - Concatenate two or more strings into a single string.
- LENGTH()
  - Get the length of a string in bytes and in characters.
- LOWER()
  - Convert a string to lowercase.
- UPPER()
  - Convert a string to uppercase.
- LEFT(), RIGHT(), LTRIM(), RTRIM(), SUBSTRING(), FORMAT() etc.
- There are many other aggregate functions available in MySQL. Please refer the Reference manual of respective version

https://dev.mysql.com/doc/refman/8.0/en/sql-function-reference.html

SQL CONCAT() Function

- Concatenate two or more strings into a single string

SELECT CONCAT(string1, string2,….);

- Exercise 13
  - SELECT CONCAT(“Hello ” , ”World….!!!”);
  - SELECT CONCAT (“Hello ” , NULL, ”World….!!!”);

MySQL String Functions

Exercise 14

- Try out the rest of the functions by yourself.
- Give example queries for the rest of the functions by yourself.

MySQL Date Functions

- MySQL Date Functions that allow you to manipulate date and time data effectively.
- CURDATE()
  - Returns the current date
- YEAR
  - Return the year for a specified date
- MONTH
  - Returns an integer that represents a month of a specified date
- DAY
  - Gets the day of the month of a specified date
- DATEDIFF(), DATE_FORMAT(), DAYOFWEEK(), DAYNAME()
- There are many other aggregate functions available in MySQL. Please refer the Reference manual of respective version

https://dev.mysql.com/doc/refman/8.0/en/sql-function-reference.html

SQL CURDATE() Function

- Returns the current date

```
SELECT CURDATE();
```

- Exercise 15
  - SELECT CURDATE();

MySQL Date Functions

Exercise 16

- Try the other date functions by yourself

Ex:

```
SELECT YEAR('1997-12-25');
SELECT MONTH('1997-12-25');
SELECT DAY('1997-12-25');
SELECT YEAR(CURDATE());
```

SQL SELECT DISTINCT Statement

- The SELECT DISTINCT statement is used to return only distinct (different) values.
- Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

SQL SELECT DISTINCT Statement

SELECT DISTINCT *column1*, *column2*, ...
FROM *table_name*;

- Exercise 17
  - Select distinct values from gender column
- TRY
  - Retrieve How many distinct values are there in gender column
    - Hint : use COUNT AND DISTINCT TOGETHER

SQL Aliases

- SQL aliases are used to give a table, or a column in a table, a temporary name.
- Aliases are often used to make column names more readable.
- An alias only exists for the duration of the query
- Aliases are useful when,
  - There are more than one table involved in a query
  - Functions are used in the query
  - Column names are big or not very readable
  - Two or more columns are combined together

SQL Aliases

Alias for Columns

SELECT *column_name* AS *alias_name*
FROM *table_name;*

- Exercise 18
  - Display dob as "Date of Birth" in your select query
  - Display reg_no as "Registration No" and student name and gender(together seperated by a ',' as "Basic Data"

SQL Aliases

Exercise 19

- Try to display Registration no and Name from student_data table and Final marks and Grade from ict_marks

SQL Aliases

Alias for Tables

SELECT ***a.column_name/s,b.column_name/s***
FROM ***table1_name AS a, table2_name AS b;***

- Exercise 20
  - Display Registration no and name from student_data table and final marks and grade from ict_marks

- Try
  - Try to display above given fields for ICT001

FIND IT

- Exercise 21
  - Find "Age" of all the students in student data table and display it in following format.

Registration No	Student Name	Date of Birth	Gender	Age

Homework

- Find about SQL HAVING
- Find about Foreign key constrain restrictions
  (RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT)
- Try following things
  - ORDER BY Aliases
  - ORDER BY 1

Questions ???

Thank You