



­

Hands-on Exercise:

ANSI SQL Functions

Version: ANSI SQL/Hands-on Exercise/05/1.0

Table of Contents

[ANSI SQL 05: ANSI SQL Functions 3](#_Toc356564953)

[Exercise.1 3](#_Toc356564954)

ANSI SQL 05: ANSI SQL Functions

**Pre-requisites:** Tables should be created and data should be loaded into the tables.

Exercise.1

|  |  |
| --- | --- |
| Estimated Completion Time: 60 Minutes | Marks: xx |
|  |  |
| Hands-on Exercise Objective |
| After completing the hands-on exercises, you will be able to:  Write queries using ANSI SQL Functions. |
|  |  |

Problem Statement:

**Problem 1:** Develop a query which will display the module name and module Infra fees of the entire module. The infra fee should be rounded to 2 decimal point.

=>SELECT Module\_Name, ROUND(infra\_fee, 2) from module\_info;

**Problem 2:** Develop a query which will list all the module id and module names in Module\_Info table where in the first letter should be capital letter.

=>SELECT ename, ascii(substr(ename,1,1)) FROM emp

WHERE ascii(substr(ename,1,1)) between 66 and 98

**Problem 3:** Develop a query which will display the module id and the number of days between the current date and module start date in associate\_status table.

=>SELECT Module\_id, datediff(current\_date(),start\_date) as numberofdays from associate\_status;

**Problem 4:** Develop a query which will concatenate the Module Name and Module id in the following format and display all the modules in the module\_info table.

“< Module Name><Module id>”

=>SELECT concat('<',Module\_name,'>','<',Module\_id,'>')as new from module\_info;

**Problem 5:** Develop a query which will display all the Module Name in upper case.

=>Select upper(modulename) from Module\_info;

**Problem 6:** Develop a query which will display all the characters between 1 and 3 of the Module name column for all the modules in the Module\_Info table.

=>SELECT Substr(Module\_Name,1,3) FROM Module\_Info;

**Problem 7**: Develop a query calculate average of all the module base fees, any records whose base fee is null needs to be considered as zero.

=>Select avg(IFNULL(basefee,0) from Module\_info;

**Problem 8:**

Write a query which will convert Trainer\_Info’s Trainer\_Id to Number and add 100000 and display it for all the trainers in the Trainer\_Info table.

=>SELECT convert((trim(leading ‘F’ FROM Trainer\_id)),decimal) + 100000

FROM Trainer\_info;

**Problem 9:**

Write a query which will convert Base\_Fees into Varchar from the Module\_info table.

And display in the following format

**‘The Base** Fees Amount for the module name’ <Module Name>’ is ’<Base Fees>

=>ALTER TABLE Module\_info MODIFY base\_fees Varchar(20);

SELECT concat(Module\_name, ‘is’, base\_fees)

FROM Module\_info;

**Problem 10:** Write a query which will display the total number of records in Module\_Info table.

=> Select count(\*) from Module\_info;

**Problem 11**: Develop a query which will give the sum of all base fees of all modules in the Module\_Fees table.

=>Select sum(base\_fee) from Module\_fees;

**Problem 12:**  Display the minimum and maximum base fees of the modules.

=> Select min(base\_fee),max(base\_fee) from module\_fees;

Deliverables Expected:

**Solution Queries**

Tips:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Funtion** | **Description** | **Example** | **Output** | **Reference** |
| DATE\_FORMAT(date,format) | Formats the ***date*** value according to the ***format*** string. | **SELECT DATE\_FORMAT('2009-10-04', '%W %M %Y');** | Sunday October 2009 | [https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html#function\_date-format](https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html" \l "function_date-format) |
| **SELECT DATE\_FORMAT('2009-10-04', '%d-%b-%Y');** | 4-Oct-2009 |
| **SELECT DATE\_FORMAT('2009-10-04', '%d-%m-%Y');** | 4-10-2009 |
| STR\_TO\_DATE(str,format) | Formats a String to currosponding date | SELECT STR\_TO\_DATE('01-5-2013','%d-%m-%Y'); | 5/1/2013 | [https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html#function\_date-format](https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html" \l "function_date-format) |
| EXTRACT(unit FROM date) | extracts parts from the date | **SELECT EXTRACT(YEAR\_MONTH FROM '2009-07-02 01:02:03');** | 200907 | [https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html#function\_extract](https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html" \l "function_extract) |
| PERIOD\_DIFF(period1,period2) | Return the number of months between periods | **SELECT PERIOD\_DIFF(200802,200703);** | 11 | [https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html#function\_period-diff](https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html" \l "function_period-diff) |
| FORMAT (N, D) | converts a number to a format like ‘#,###,###.##’ which is rounded upto the number of decimal places specified (in the second argument) and returns the result as a string. | SELECT FORMAT(12324.2573,3); | 12,324.26 | [http://dev.mysql.com/doc/refman/5.0/en/string-functions.html#function\_format](http://dev.mysql.com/doc/refman/5.0/en/string-functions.html" \l "function_format) |
| CONCAT(str1,str2) | Return concatenated string | SELECT CONCAT('My', 'SQL'); | MySQL | <http://dev.mysql.com/doc/refman/5.0/en/string-functions.html> |
| SUBSTR() | Return the substring as specified | SELECT SUBSTRING ('Quadratically',5,6); | 'ratica' | [http://dev.mysql.com/doc/refman/5.0/en/string-functions.html#function\_substr](http://dev.mysql.com/doc/refman/5.0/en/string-functions.html" \l "function_substr) |
| ROUND() | Round the argument | SELECT ROUND(1.298, 1); | 1.3 | [http://dev.mysql.com/doc/refman/5.0/en/mathematical-functions.html#function\_round](http://dev.mysql.com/doc/refman/5.0/en/mathematical-functions.html" \l "function_round) |
| TRUNCATE() | Truncate to specified number of decimal places | SELECT TRUNCATE(1.223,1); | 1.2 | [http://dev.mysql.com/doc/refman/5.0/en/mathematical-functions.html#function\_truncate](http://dev.mysql.com/doc/refman/5.0/en/mathematical-functions.html" \l "function_truncate) |
| CASE | CASE value WHEN [compare\_value] THEN result [WHEN [compare\_value] THEN result ...] [ELSE result] END | SELECT CASE WHEN 1>0 THEN 'true' ELSE 'false' END; | TRUE |  |
| IF | IF(expr1,expr2,expr3) | **SELECT IF(1<2,'yes','no');** | yes |  |
| IFNULL | IFNULL(expr1,expr2) | **SELECT IFNULL(NULL,5);** | 5 |  |