# IST769 Homework Submission Template

## Basic Information

Your Name: Srihari Busam   
Your SUID: sbusam  
Your Email: sbusam@syr.edu  
Date Due: 10/14/2021  
Homework #: 2

## QUESTIONS:

1. Use built in SQL functions to write an SQL Select statement on **fudgemart\_products** which derives a **product\_category** column by extracting the last word in the product name. For example
   1. for a product named ‘Leather Jacket’ the product category would be ‘Jacket’
   2. for a product named ‘Straight Claw Hammer’ the category would be ‘Hammer’

Your select statement should include product id, product name, product category and product department.

1. Write a user defined function called **f\_total\_vendor\_sales** which calculates the sum of the wholesale price \* quantity of all products sold for that vendor. There should be one number associated with each vendor id, which is the input into the function. Demonstrate the function works by executing an SQL select statement over all vendors calling the function.
2. Write a stored procedure called **p\_write\_vendor** which when given a required vendor name, phone and optional website, will look up the vendor by name first. If the vendor exists, it will update the phone and website. If the vendor does not exist, it will add the info to the table. Write code to demonstrate the procedure works by executing the procedure twice so that it adds a new vendor and then updates that vendor’s information.
3. Create a view based on the logic you completed in question 1 or 2. Your SQL script should be programmed so that the entire script works every time, dropping the view if it exists, and then re-creating it.
4. Write a table valued function **f\_employee\_timesheets** which when provided an employee\_id will output the employee id, name, department, payroll date, hourly rate on the timesheet, hours worked, and gross pay (hourly rate times hours worked).

## **Your Answers:**

1.

Graphical user interface, application

Description automatically generated

|  |
| --- |
| SQL statement |
| SELECT product\_id,  product\_name,  CASE  WHEN CHARINDEX(' ', product\_name) < 1 THEN product\_name  ELSE RIGHT(product\_name, CHARINDEX(' ', REVERSE(product\_name))-1)  END AS product\_category  FROM fudgemart\_products; |

2.

Graphical user interface, text, application

Description automatically generated

|  |
| --- |
| CODE |
| DROP FUNCTION IF EXISTS dbo.f\_total\_vendor\_sales;  go  CREATE OR ALTER FUNCTION dbo.f\_total\_vendor\_sales ( @vendor\_id AS INT )  RETURNS money  AS  BEGIN  RETURN  (  SELECT Sum(PRD.product\_wholesale\_price \* ORDD.order\_qty)  FROM fudgemart\_products PRD  INNER JOIN fudgemart\_order\_details ORDD  ON PRD.product\_id = ORDD.product\_id  WHERE PRD.product\_vendor\_id = @vendor\_id  );  END  go  PRINT 'Execute the function dbo.f\_total\_vendor\_sale for all vendors'  SELECT vendor\_name,  dbo.F\_total\_vendor\_sales(vendor\_id) AS TotalSales  FROM dbo.fudgemart\_vendors  ORDER BY vendor\_name;  go |

3.

Graphical user interface, text, application

Description automatically generated

|  |
| --- |
| CODE |
| DROP PROCEDURE IF EXISTS dbo.p\_write\_vendor;  go  CREATE OR ALTER PROCEDURE dbo.p\_write\_vendor  @vendor\_name nvarchar(50),  @vendor\_phone varchar(15),  @vendor\_website varchar(256)  AS  IF EXISTS ( SELECT \* FROM fudgemart\_vendors WHERE vendor\_name = @vendor\_name)  UPDATE dbo.fudgemart\_vendors  SET vendor\_phone = @vendor\_phone, vendor\_website = @vendor\_website  WHERE vendor\_name = @vendor\_name;  ELSE  INSERT INTO dbo.fudgemart\_vendors ( vendor\_name, vendor\_phone, vendor\_website)  VALUES ( @vendor\_name, @vendor\_phone, @vendor\_website);  go  EXEC dbo.p\_write\_vendor 'test vendor', '000-000-000', 'http://test-website.com';  go  SELECT \* FROM fudgemart\_vendors WHERE vendor\_name = 'test vendor'  go  EXEC dbo.p\_write\_vendor 'test vendor', '999-999-999', 'http://test-website.com';  go  SELECT \* FROM fudgemart\_vendors WHERE vendor\_name = 'test vendor'  go |

4.

Graphical user interface, application

Description automatically generated

|  |
| --- |
| CODE |
| DROP VIEW IF EXISTS dbo.vw\_total\_vendor\_sales;  go  CREATE OR ALTER VIEW dbo.vw\_total\_vendor\_sales AS  SELECT PRD.product\_id, PRD.product\_vendor\_id,  PRD.product\_wholesale\_price \* ORDD.order\_qty as sales\_value  FROM fudgemart\_products PRD  INNER JOIN fudgemart\_order\_details ORDD  ON PRD.product\_id = ORDD.product\_id  go  SELECT \* from dbo.vw\_total\_vendor\_sales  go |

5.

Graphical user interface, text, application

Description automatically generated

|  |
| --- |
| CODE |
| DROP FUNCTION IF EXISTS dbo.f\_employee\_timesheets;  go  CREATE OR ALTER FUNCTION dbo.f\_employee\_timesheets ( @employee\_id AS INT )  RETURNS TABLE  AS  RETURN  (  SELECT EMP.employee\_id,  EMP.employee\_lastname,  EMP.employee\_firstname,  EMP.employee\_department,  ETS.timesheet\_payrolldate,  ETS.timesheet\_hourlyrate,  ETS.timesheet\_hours \* ETS.timesheet\_hourlyrate as gross\_pay  FROM fudgemart\_employee\_timesheets AS ETS  INNER JOIN fudgemart\_employees AS EMP  ON EMP.employee\_id = ETS.timesheet\_employee\_id  WHERE EMP.employee\_id = @employee\_id  );  go  SELECT \* FROM dbo.f\_employee\_timesheets(2);  go |