Instructions

For each answer, please include your answer as text, and any screenshot(s) which demonstrate your answer was executed. Most importantly, make sure to include evidence your answer is correct. This will most likely be a screenshot. If you had issues, problems, or had to make assumptions include them in your answer.

Your Answers:

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
   a. for a product named 'Leather Jacket' the product category would be 'Jacket'
   b. 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.

```
use fudgemart_v3
go

-- multiple sub queries to get to the end product category
select
	product_id,
	product_name,
	substring(product_category, charindex(' ', product_category) + 1, 50) as
product_category,
	product_department
from (
	-- start sub query
	select
		product_id,
		product_name,
		substring(product_category, charindex(' ', product_category) + 1, 50) as
product_category,
		product_department
	from (
		-- start subquery
		select
			product_id,
			product_name,
			substring(product_name, charindex(' ', product_name) + 1, 50) as
product_category,
			product_department
		from
			fudgemart_products
		-- end subquery
		) sub
	-- end subquery
	) sub
Go
```

RESULT

Results | Messages

	product_id	product_name	product_category	product_department
1	1	Straight Claw Hammer	Hammer	Hardware
2	2	Sledge Hammer	Hammer	Hardware
3	3	Rip Claw Hammer	Hammer	Hardware
4	4	Dri-Fit Tee	Tee	Clothing
5	5	Running Pants	Pants	Clothing
6	6	Wool Socks	Socks	Clothing
7	7	Squeaky Sneaks	Sneaks	Clothing
8	8	Cool Jeans	Jeans	Clothing
9	9	Denim Jacket	Jacket	Clothing
10	10	Leather Jacket	Jacket	Clothing
11	11	Courdory Pants	Pants	Clothing
12	12	Work Pants	Pants	Clothing
13	13	Work Gloves	Gloves	Clothing
14	14	Comfor-fit Tee	Tee	Clothing

2. 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.

```
use fudgemart_v3
go

drop function if exists dbo.f_total_vendor_sales
go

create function dbo.f_total_vendor_sales(
	@vendor_id as int -- user input
	) returns int as
begin
	declare @revenue int -- revenue = qty * price
	set @revenue = (select sum(product_wholesale_price * order_qty)
				from fudgemart_order_details
				join fudgemart_products on
fudgemart_products.product_id = fudgemart_order_details.product_id
				join fudgemart_vendors on fudgemart_vendors.vendor_id =
fudgemart_products.product_vendor_id
				where vendor_id = @vendor_id)
	return @revenue
end
go

select
	vendor_id,
	vendor_name,
	cast(isnull(dbo.f_total_vendor_sales(vendor_id), 0) as money) as revenue -- format
```

```
from
        fudgemart_vendors
go
```

RESULT

	vendor_id	vendor_name	revenue
1	10	Blackened-Deckhand	302535.00
2	6	Fudgeman	0.00
3	11	fudgemaster	0.00
4	9	Fudgeoco	0.00
5	7	Leaveeyes	30750.00
6	5	Mikerosoft	117533.00
7	2	Mikey	57603.00
8	1	Soney	1532050.00
9	3	Stanlee	48340.00
10	8	Weebock	59943.00

3. 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.

BEFORE

	vendor_id	vendor_name	vendor_phone	vendor_website
1	1	Soney	555-2939	http://www.soney.com
2	2	Mikey	555-2870	http://mikee.com
3	3	Stanlee	555-9920	NULL
4	5	Mikerosoft	555-2220	http://www.mikerosoft.com
5	6	Fudgeman	555-1239	http://www.fudgeman.com
6	7	Leaveeyes	555-2931	NULL
7	8	Weebock	555-0002	http://www.weebock.com
8	9	Fudgeoco	555-0232	http://www.fudgeoco.com
9	10	Blackened-Deckhand	555-9922	NULL
10	11	fudgemaster	555-999	www.fudgemaster.com

```
use fudgemart_v3
go

drop procedure if exists dbo.p_write_vendor
go

create procedure dbo.p_write_vendor(
	@vendor_name varchar(50),
	@vendor_phone varchar(20),
	@vendor_website varchar(1000) = NULL
)as
begin
	if exists (select vendor_name from fudgemart_vendors where vendor_name =
@vendor_name)
	begin
		update fudgemart_vendors set vendor_phone = @vendor_phone where vendor_name
= @vendor_name
		update fudgemart_vendors set vendor_website = @vendor_website where
vendor_name = @vendor_name
	end
	else
	begin
		insert into fudgemart_vendors (vendor_name, vendor_phone, vendor_website)
		values (@vendor_name, @vendor_phone, @vendor_website)
	end
end
go

exec
	dbo.p_write_vendor @vendor_name = 'goodvendor', @vendor_phone = '999-5552'
go

exec
	dbo.p_write_vendor @vendor_name = 'goodvendor', @vendor_phone = '966-5812',
@vendor_website = 'www.goodvendor.com'
go
```

AFTER

	vendor_id	vendor_name	vendor_phone	vendor_website
1	1	Soney	555-2939	http://www.soney.com
2	2	Mikey	555-2870	http://mikee.com
3	3	Stanlee	555-9920	NULL
4	5	Mikerosoft	555-2220	http://www.mikerosoft.com
5	6	Fudgeman	555-1239	http://www.fudgeman.com
6	7	Leaveeyes	555-2931	NULL
7	8	Weebock	555-0002	http://www.weebock.com
8	9	Fudgeoco	555-0232	http://www.fudgeoco.com
9	10	Blackened-Deckhand	555-9922	NULL
10	11	fudgemaster	555-999	www.fudgemaster.com
11	12	goodvendor	966-5812	www.goodvendor.com

4. 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.

```
use fudgemart_v3
go

drop view if exists dbo.myview
go

create view dbo.myview as

select
	t1.product_id,
	t1.product_category,
	t1.vendor_id,
	t2.revenue as vendor_revenue
from
(select
	product_id,
	vendor_id,
	substring(product_category, charindex(' ', product_category) + 1, 50) as
product_category
from (
	select
		product_id,
		vendor_id,
		substring(product_category, charindex(' ', product_category) + 1, 50) as
product_category
	from (
		select
			product_id,
			vendor_id,
			substring(product_name, charindex(' ', product_name) + 1, 50) as
product_category
		from
			fudgemart_products
```

```
                    join fudgemart_vendors on fudgemart_vendors.vendor_id =
fudgemart_products.product_vendor_id
                ) sub
        ) sub
) t1

left join

(select
        vendor_id,
        cast(isnull(dbo.f_total_vendor_sales(vendor_id), 0) as money) as revenue
from
        fudgemart_vendors
) t2

on t1.vendor_id = t2.vendor_id

go

select * from dbo.myview
go
```

RESULT

	product_id	product_category	vendor_id	vendor_revenue
1	1	Hammer	3	48340.00
2	2	Hammer	3	48340.00
3	3	Hammer	3	48340.00
4	4	Tee	2	57603.00
5	5	Pants	2	57603.00
6	6	Socks	2	57603.00
7	7	Sneaks	2	57603.00
8	8	Jeans	7	30750.00
9	9	Jacket	7	30750.00
10	10	Jacket	7	30750.00
11	11	Pants	7	30750.00
12	12	Pants	3	48340.00
13	13	Gloves	3	48340.00
14	14	Tee	8	59943.00
15	15	Shorts	8	59943.00

5. 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).

```
use fudgemart_v3
go

drop function if exists dbo.f_employee_timesheets
go
```

```
create function dbo.f_employee_timesheets(
	@employee_id as int
) returns table as
return(
	select
		fudgemart_employees.employee_id,
		fudgemart_employees.employee_firstname,
		fudgemart_employees.employee_lastname,
		fudgemart_employees.employee_department,
		fudgemart_employee_timesheets.timesheet_payrolldate,
		fudgemart_employee_timesheets.timesheet_hourlyrate,
		fudgemart_employee_timesheets.timesheet_hours,
		fudgemart_employee_timesheets.timesheet_hourlyrate *
fudgemart_employee_timesheets.timesheet_hours as grosspay
	from
		fudgemart_employees
		join fudgemart_employee_timesheets on
fudgemart_employee_timesheets.timesheet_employee_id = fudgemart_employees.employee_id
	where
		fudgemart_employees.employee_id = @employee_id
)
go

select * from dbo.f_employee_timesheets(10)
go
```

RESULT

	employee_id	employee_firstname	employee_lastname	employee_department	timesheet_payrolldate	timesheet_hourlyrate	timesheet_hours	grosspay
1	10	Artie	Choke	Hardware	2010-01-04 00:00:00.000	11.95	40.0	478.00000
2	10	Artie	Choke	Hardware	2010-01-11 00:00:00.000	11.95	40.0	478.00000
3	10	Artie	Choke	Hardware	2010-01-18 00:00:00.000	11.95	40.0	478.00000
4	10	Artie	Choke	Hardware	2010-01-25 00:00:00.000	11.95	40.0	478.00000
5	10	Artie	Choke	Hardware	2010-02-01 00:00:00.000	11.95	40.0	478.00000
6	10	Artie	Choke	Hardware	2010-02-08 00:00:00.000	11.95	40.0	478.00000
7	10	Artie	Choke	Hardware	2010-02-15 00:00:00.000	11.95	40.0	478.00000
8	10	Artie	Choke	Hardware	2010-02-22 00:00:00.000	11.95	40.0	478.00000
9	10	Artie	Choke	Hardware	2010-03-01 00:00:00.000	11.95	40.0	478.00000
10	10	Artie	Choke	Hardware	2010-03-08 00:00:00.000	11.95	40.0	478.00000
11	10	Artie	Choke	Hardware	2010-03-15 00:00:00.000	11.95	40.0	478.00000
12	10	Artie	Choke	Hardware	2010-03-22 00:00:00.000	11.95	40.0	478.00000
13	10	Artie	Choke	Hardware	2010-03-29 00:00:00.000	11.95	40.0	478.00000
14	10	Artie	Choke	Hardware	2010-04-05 00:00:00.000	11.95	40.0	478.00000
15	10	Artie	Choke	Hardware	2010-04-12 00:00:00.000	11.95	40.0	478.00000