Gavin Erezuma

CS 180

Final Report

**General Meats**

General Meats is a small family owned butcher shop that sells only land animals from local organic farms, no seafood. They have a total of 8 employees: 2 managers ( one front house, one backhouse), 4 butchers ( three full time and one part time)- one of the full time butchers is a junior butcher, and finally 2 front house employees (these are the ones that handle final sales and help to keep the small merchandise items stocked (beer, wine, marinades, dry goods, small grocery section).

They receive deliveries two days a week from multiple local organic farms. All items that are delivered are checked in by one of the managers. The meat that is delivered is bought by the pound for the entire carcass and then sold by the pound; dependent on the cut of meat sold. All the drygoods and grocery items are sold at a marked up price.

The butcher house also has a cash bank in house. The daily minimum in the house bank is $400 ( 225 dollars - $1, 100 dollars - $5, 50 dollars $.25, 10 dollars $.10, 10 dollars $.05, 5 dollars $.01). All excess cash received throughout daily sales will be recorded and deposited on a daily basis to the local bank.

The carcasses are cut into forequarter and hind quarters. Then from there they are broken down dependent on the cuts that are processed by the type of animal being processed. Birds can be sold as whole birds, half birds, breast with bones, boneless breast, leg quarters, thigh bone in, thigh boneless, legs, wings, and bones.

The butcher shop will also have marinated meats, VAP [value added products]- (sausages, pates, meatloaf’s, ect) and other products that are made or processed in the shop. The shop also sells stocks (chicken, beef), soups (TBD), dog food, and whole cooked chickens.

General Meats is open 7 days a week. Sunday through Thursday from 10:30am to 7:00pm and Friday and Saturday from 10:30am to 8:00pm. The butchers arrive in one of three shifts - (6:00am-2:30pm) (10:00am-6:30pm) (12:00pm-8:30pm) with half an hour break. The front house staff has two shifts (10:00am- 6:30pm) and (12:30pm-9pm) with half hour break.

Sanitary logs are kept throughout the day by the front house staff. These logs include temperature of all refrigeration, freezers and heating units. The butcher department is responsible for logging all waste from there cuts. Anything animal product that is not sold and placed in a waste bin must be weighed (due to spoilage or unsellable items). This weight is then tallied and given a fixed price as loss in revenue. VAP items will follow specified recipes that will be weighed using the metric system.

Inventory will be kept on a monthly basis. All meat products will be weighed and tallied by animal. This total will have the total waste added to it and compared to the theoretical inventory. All other inventory items will be tallied by each/case or weight dependent on how the item is entered into the system initially. The linens (hand towels, aprons) will be a line item as they are provided by a vendor.

**General Meats report requirements as per client**

**List of Employee with name, address, and hourly rate--**

select concat(last\_name,", ", first\_name) as Employee,

street , city, state,

pay\_rate\_per\_hour as "Hourly Rate",

department as "Department"

from

Employee join Human\_Resources using(hr\_id)

join Address using(address\_id);

**Output**



**Printed Schedule for Each Employees--**

create procedure printSchedule

(

employee\_first\_name\_param varChar(50),

employee\_last\_name\_param varChar(50)

)

begin

declare employee\_id\_var int;

select employee\_id into employee\_id\_var from Employee where employee\_id = findEmployee(employee\_first\_name\_param, employee\_last\_name\_param);

if findEmployee(employee\_first\_name\_param, employee\_last\_name\_param) = 0 then

select "No Employee Found" as Message;

else

select concat(first\_name, " " , last\_name) as "Employee",

week\_day as "Day" , start\_shift as "Start Time", end\_shift as "End Time"

from Employee join Employee\_Schedule\_Link using(employee\_id)

join Emp\_Schedule using(emp\_schedule\_id)

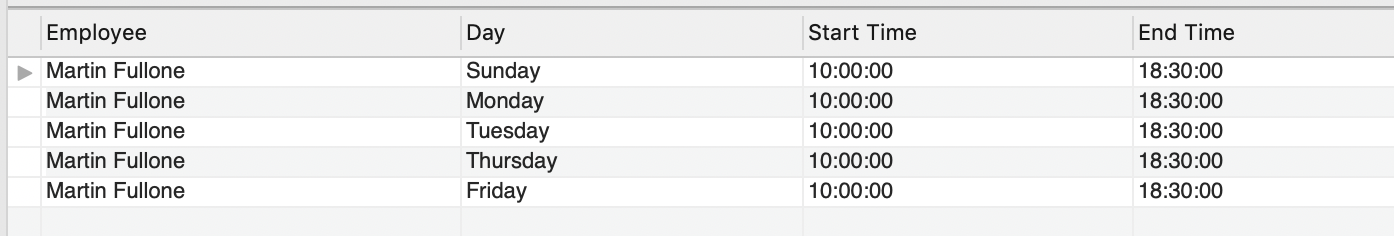
join Day\_Of\_Week using(day\_id)

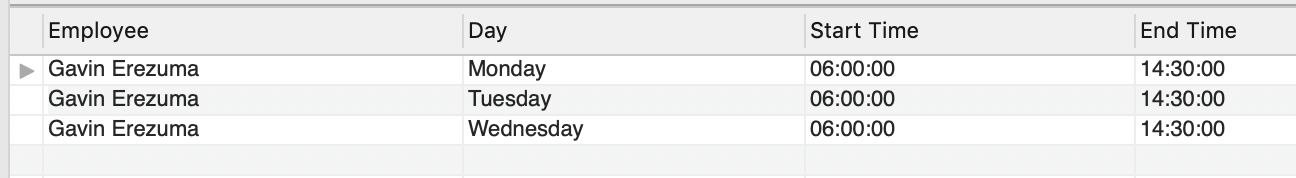
where Employee\_Schedule\_Link.employee\_id = employee\_id\_var;

end if;

end//

**Output**





**Breakdown of Species by Vendor--**

select vendor\_name as "Vendor",

species\_type as "Species",

cut\_name as "Cut",

vendors\_price as "Our Cost"

from Vendor

join Vendor\_Species\_Link using(vendor\_id)

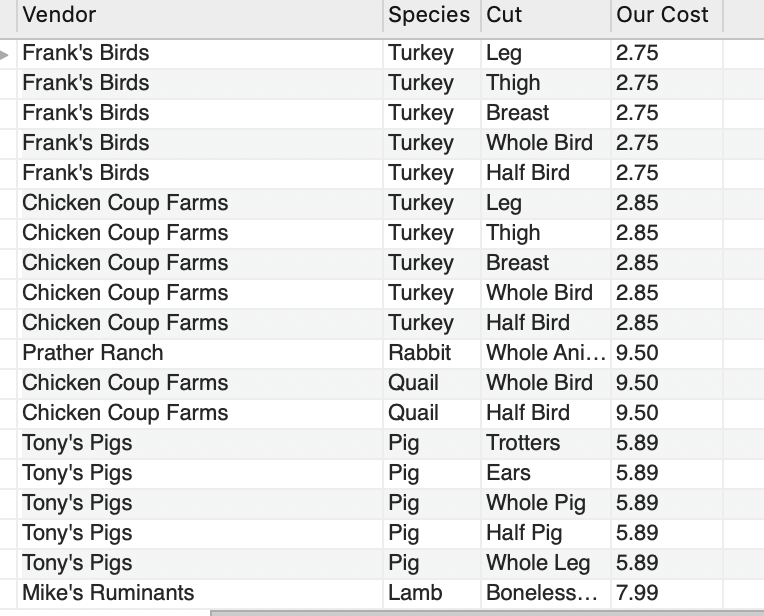
join Species using(species\_id)

join Species\_Cuts\_Link using(species\_id)

join Cuts using(cuts\_id)

order by species\_type desc;

**Output**



**Breakdown of Grocery and Dry Goods by Vendor--**

**The functions and procedures to add and find items as well as link items to vendors.**

**This was done because an item may be sold by multiple vendors.**

**Below is the dry goods. Grocery and Species are very similar.**

**Find:**

create function findDryGoods

(

item\_name\_param varChar(50)

)

returns int

deterministic reads sql data

begin

declare dryGoods\_id\_var int;

declare continue handler for not found

set dryGoods\_id\_var = 0;

select dry\_goods\_id into dryGoods\_id\_var

from Dry\_Goods

where item\_name = item\_name\_param;

return dryGoods\_id\_var;

end//

**Add:**

create procedure addDryGoods

(

item\_name\_param varChar(50),

case\_size\_param int,

unit\_price\_param decimal(9,2)

)

begin

if findDryGoods(item\_name\_param) = 0 then

insert into Dry\_Goods(item\_name, case\_size, unit\_price) values

(item\_name\_param, case\_size\_param, unit\_price\_param);

end if;

end//

**Link:**

create procedure linkVendorDryGoods

(

item\_name\_param varChar (50),

vendor\_name\_param varChar (50),

vendors\_price\_param decimal(9,2)

)

begin

declare duplicate\_entry tinyint default false;

declare continue handler for 1062

set duplicate\_entry = true;

if duplicate\_entry = false then

if findDryGoods(item\_name\_param) and findVendor(vendor\_name\_param) then

insert into Vendor\_Dry\_Goods\_Link (dry\_goods\_id, vendor\_id, vendors\_price)

values (findDryGoods(item\_name\_param),findVendor(vendor\_name\_param),vendors\_price\_param);

end if;

end if;

if findDryGoods(item\_name\_param) = 0 then

select "No item found" as Message;

end if;

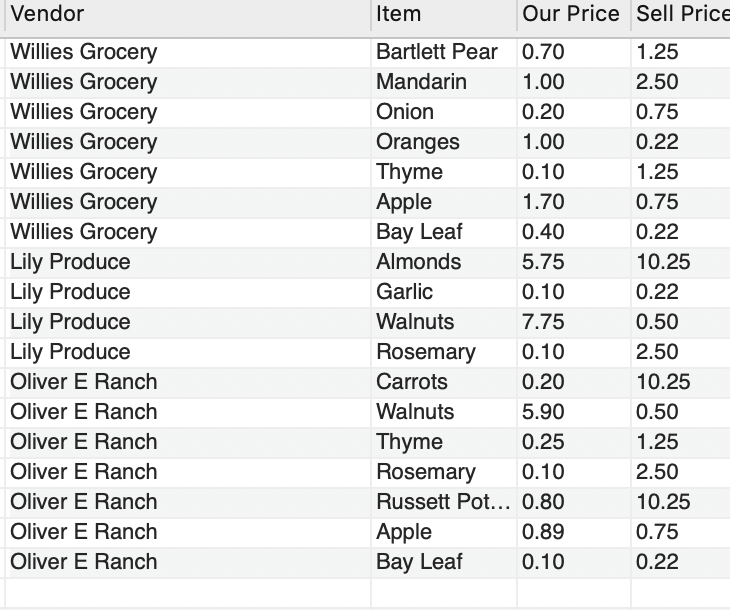
if findVendor(vendor\_name\_param) = 0 then

select "No vendor found" as Message;

end if;

end//

**Output**



**Sales report with cut name, species name,**

**time of sale , quantity, price, and a total of each line item--**

**The functions and procedures:**

**Find:**

create function findCustomer()

returns int

deterministic reads sql data

begin

declare customer\_id\_var int;

select max(customer\_id) into customer\_id\_var

from Customer;

return customer\_id\_var;

end//

**Find:**

create function findSpecies

(

species\_type\_param varChar(20)

)

returns int

deterministic reads sql data

begin

declare species\_id\_var int;

declare continue handler for not found

set species\_id\_var = 0;

select species\_id into species\_id\_var

from Species

where species\_type = species\_type\_param;

return species\_id\_var;

end//

**Sell:**

create procedure sell\_Meat\_Item

(

species\_param varChar(20),

cut\_param varChar(50),

quantity\_param int,

price\_param decimal(9,2)

)

begin

declare cus\_id\_var int;

insert into Purchase\_Order values (default,findCustomer(),findSpecies(species\_param),species\_param,

findCuts(cut\_param),cut\_param,quantity\_param,current\_timestamp(),price\_param);

end//

**Query:**

select distinct last\_name as "Customer",

Purchase\_Order.cut\_name as "Cut",

Purchase\_Order.species\_type as "Species",

purchase\_time as "Time of Purchase",

quantity as "Quantity",

purchase\_price as "Price",

(quantity \* purchase\_price) as "Total Line Item"

from

Customer join Purchase\_Order using(customer\_id)

join Species using(species\_id)

join Species\_Cuts\_Link using(species\_id)

join Cuts using(cuts\_id)

order by last\_name;

**Output**

