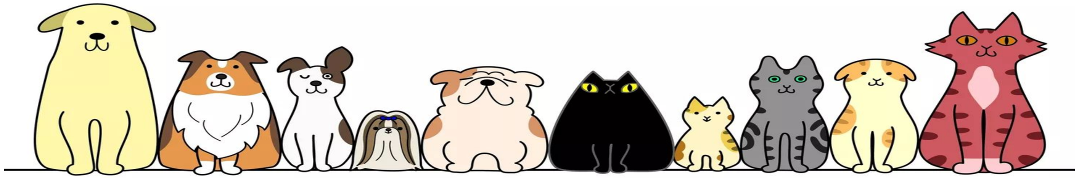
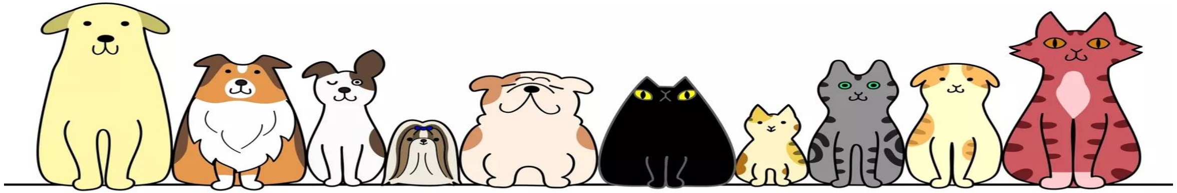
Business Database Design for Pet Grooming Shop

Presented by
Upasana Angara, Graham Cason, Joyce Guan,
Ke Yin, Shibo Zhou



Overview

- Business Scenario
- Project Introduction
- Data Modeling
- Data Normalization
- SQL Queries



Business Scenario

Business Definition: Pet Grooming Store

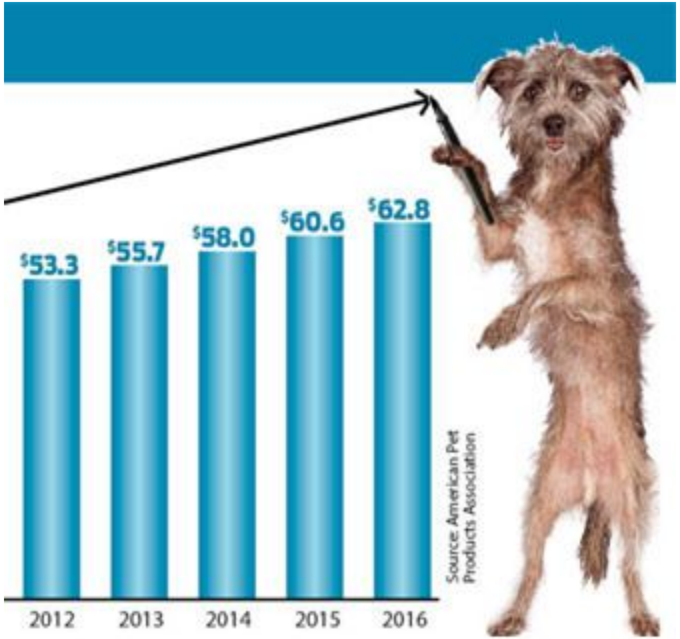
Business Requirements:

> An internal business database

- Recording business operating activities:
  - Sales
  - Purchases/Inventory
  - Appointments
- Storing backend information of company architecture
  - Departments
  - Employee Information

> Primary Users:

- Departments
- Employees



Source: American Pet Products Association

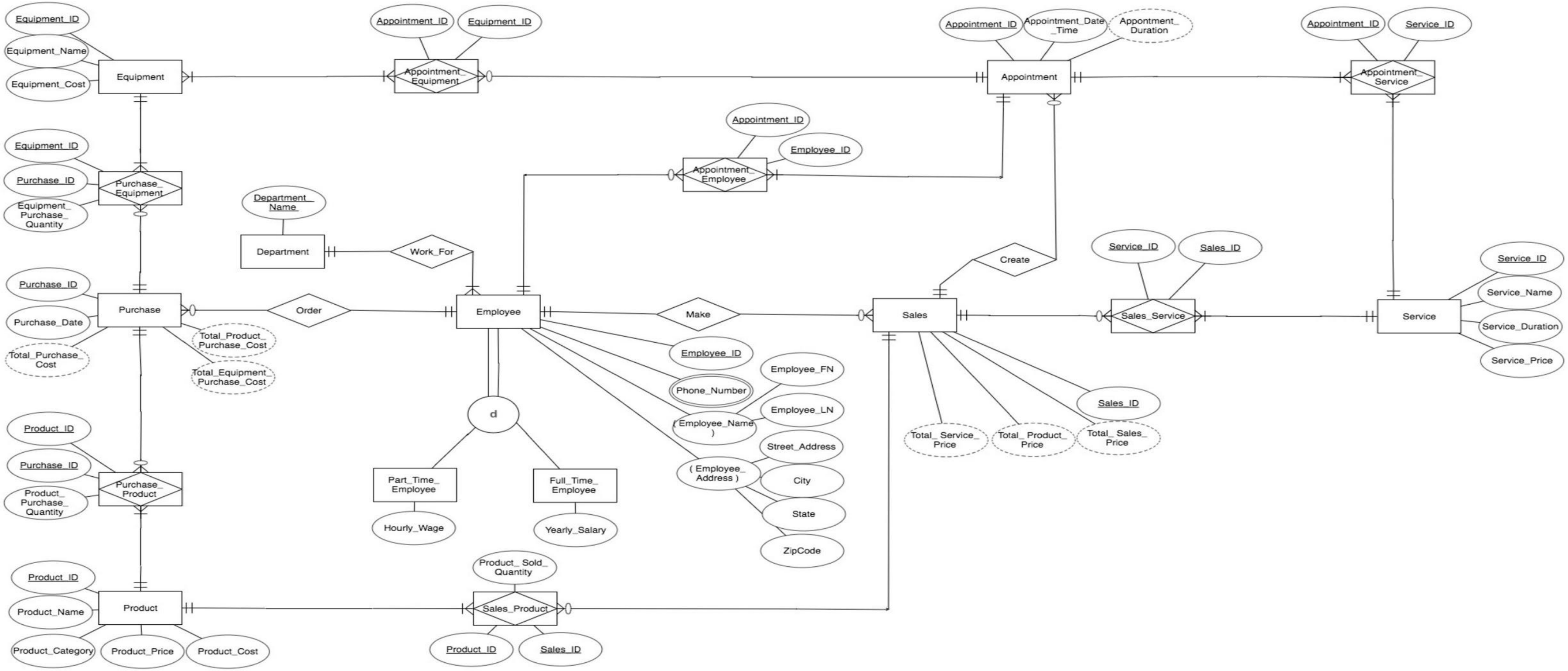
Project Introduction

Goals:

- Store and retrieve information related to internal business operations.
- Create ad-hoc reports for strategic usages.
- Perform calculation and group by functions to solve business related problems.

Entity Relationship Model

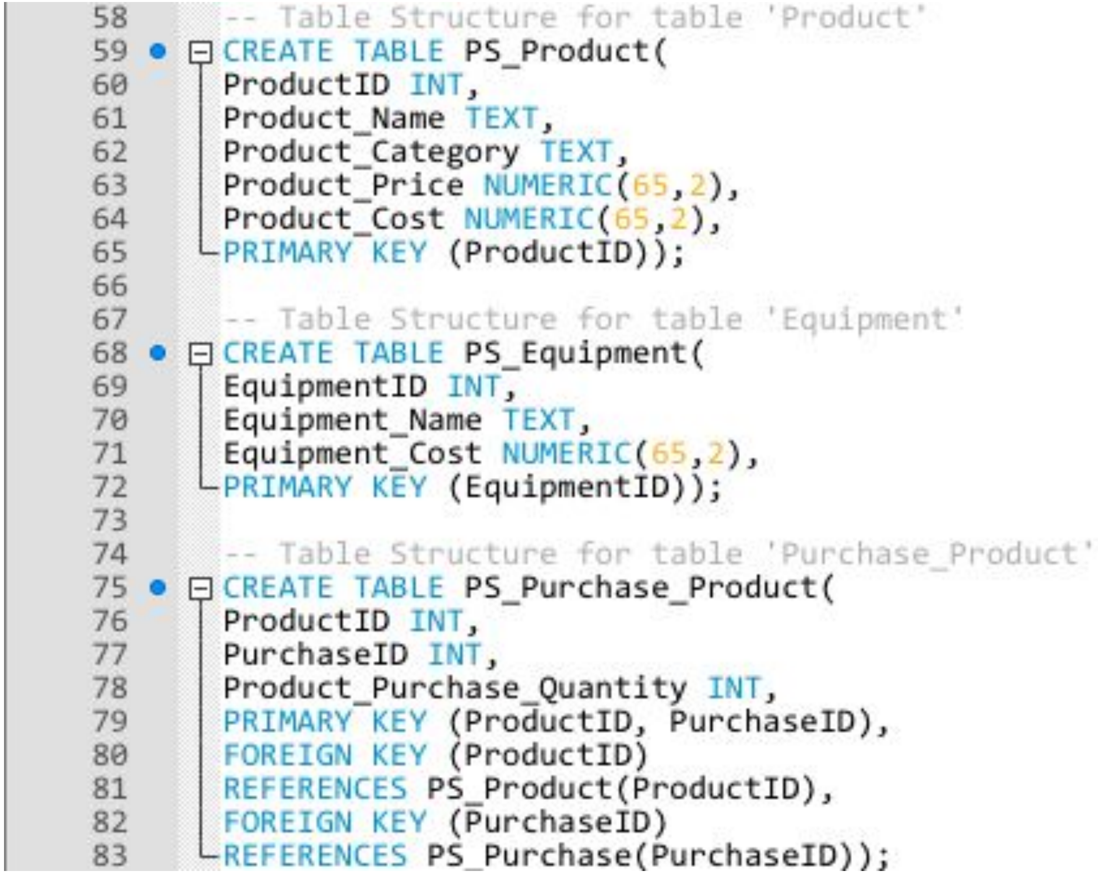
[ER Diagram]


Equipment_ID
Equipment_Name
Equipment_Cost
Equipment
Appointment_ID
Equipment_ID
Appointment_ Equipment
Appointment_ID
Appointment_Date _Time
Appontment_ Duration
Appointment
Appointment_ID
Service_ID
Appointment_ Service
Appointment_ID
Employee_ID
Appointment_ Employee
Equipment_ID
Purchase_ID
Equipment_ Purchase_ Quantity
Purchase_ Equipment
Department_ Name
Department
Work_For
Create
Service_ID
Sales_ID
Service_ID
Service_Name
Service_Duration
Service_Price
Purchase_ID
Purchase_Date
Total_Purchase_ Cost
Purchase
Order
Employee
Make
Sales
Sales_Service
Service
Total_Product_ Purchase_Cost
Total_Equipment_ Purchase_Cost
Employee_ID
Phone_Number
Employee_FN
Employee_LN
(Employee_Name)
Street_Address
(Employee_ Address)
City
State
ZipCode
d
Part_Time_ Employee
Full_Time_ Employee
Hourly_Wage
Yearly_Salary
Sales_ID
Total_ Service_ Price
Total_ Product_ Price
Total_ Sales_ Price
Product_ID
Purchase_ID
Product_ Purchase_ Quantity
Purchase_ Product
Product_ID
Product_Name
Product
Product_Category
Product_Price
Product_Cost
Product_ Sold_ Quantity
Sales_Product
Product_ID
Sales_ID

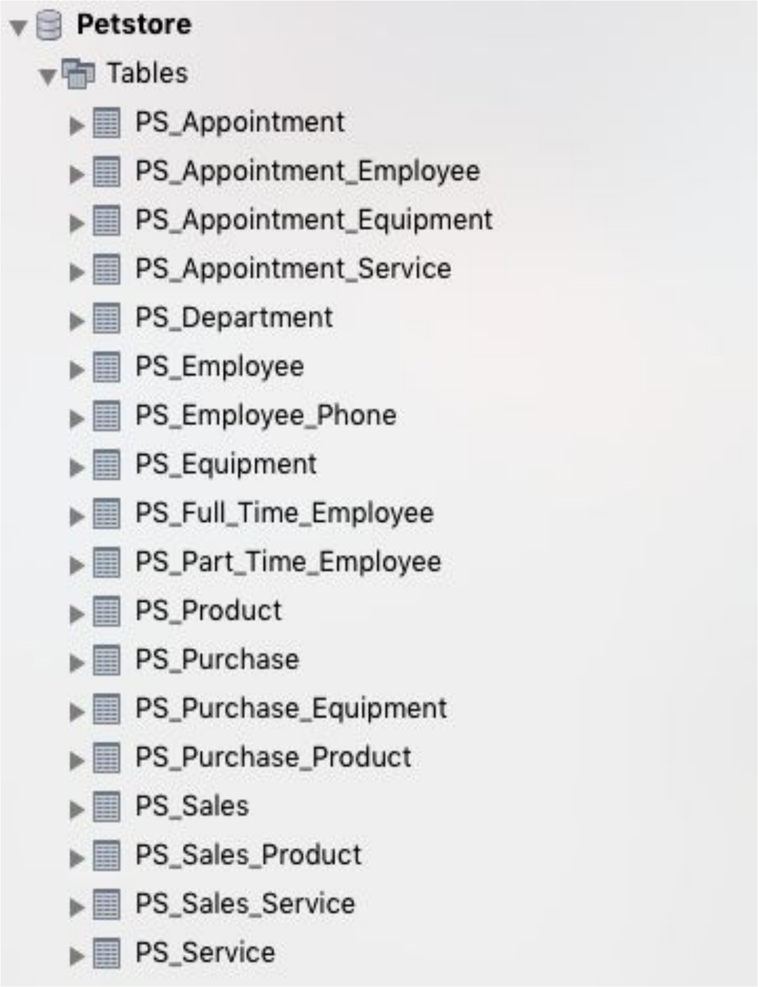
Data Normalization

1) Department (Department_Name(PK))
2) Employee (EmployeeID(PK), Employee_FN, Employee_LN, Street_Address, City, State, ZipCode, Department_Name(FK))
3) Employee_Phone(EmployeeID(FK/PK), Phone_Number)
4) Part_Time_Employee(EmployeeID(FK/PK), Hourly_Wage)
5) Full_Time_Employee(EmployeeID(FKPK), Yearly_Salary)
6) Purchase (PurchaseID(PK), PurchaseDate, EmployeeID(FK))
7) Product (ProductID, Product_Name, Product_Category, Product_Price, Product_Cost)
8) Equipment (EquipmentID, Equipment_Name, Equipment_Cost,)
9) Purchase_Product(ProductID(FK/PK), PurchaseID(FK/PK), Product_Purchase_Quantity)
10) Purchase_Equipment(EquipmentID(FK/PK), PurchaseID(FK/PK), Equipment_Purchase_Quantity)
11) Appointment (AppointmentID(PK), Appointment_Date_Time, SalesID(FK))
12) Sales (SalesID(PK/FK), EmployeeID(FK))
13) Service (ServiceID,(PK) Service_Name, Service_Duration, Service_Price)
14) Sales_Product(ProductID(PK/FK), SalesID(PK/FK))
15) Sales_Service(ServiceID(PK/FK), SalesID(PK/FK))
16) Appointment_Employee(AppointmentID(PK/FK), EmployeeID(PK/FK))
17) Appointment_Equipment(AppointmentID(PK/FK), EquimentID(PK/FK))
18) Appointment_Service(AppointmentID(PK/FK), ServiceID(PK/FK))

Database



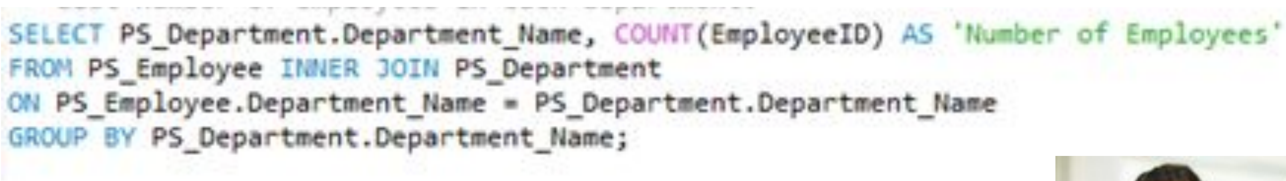
```
-- Table Structure for table 'Product'
CREATE TABLE PS_Product(
ProductID INT,
Product_Name TEXT,
Product_Category TEXT,
Product_Price NUMERIC(65,2),
Product_Cost NUMERIC(65,2),
PRIMARY KEY (ProductID));

-- Table Structure for table 'Equipment'
CREATE TABLE PS_Equipment(
EquipmentID INT,
Equipment_Name TEXT,
Equipment_Cost NUMERIC(65,2),
PRIMARY KEY (EquipmentID));

-- Table Structure for table 'Purchase_Product'
CREATE TABLE PS_Purchase_Product(
ProductID INT,
PurchaseID INT,
Product_Purchase_Quantity INT,
PRIMARY KEY (ProductID, PurchaseID),
FOREIGN KEY (ProductID)
REFERENCES PS_Product(ProductID),
FOREIGN KEY (PurchaseID)
REFERENCES PS_Purchase(PurchaseID));
```



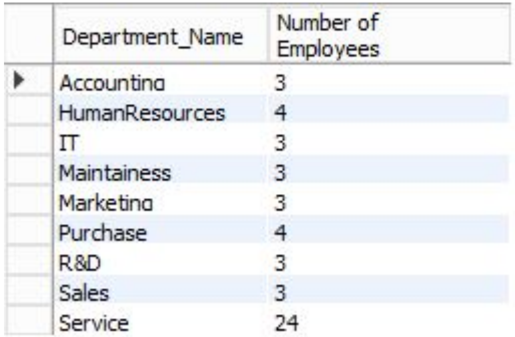
- Petstore
  - Tables
    - PS_Appointment
    - PS_Appointment_Employee
    - PS_Appointment_Equipment
    - PS_Appointment_Service
    - PS_Department
    - PS_Employee
    - PS_Employee_Phone
    - PS_Equipment
    - PS_Full_Time_Employee
    - PS_Part_Time_Employee
    - PS_Product
    - PS_Purchase
    - PS_Purchase_Equipment
    - PS_Purchase_Product
    - PS_Sales
    - PS_Sales_Product
    - PS_Sales_Service
    - PS_Service

SQL Query #1

The number of employees in each department.



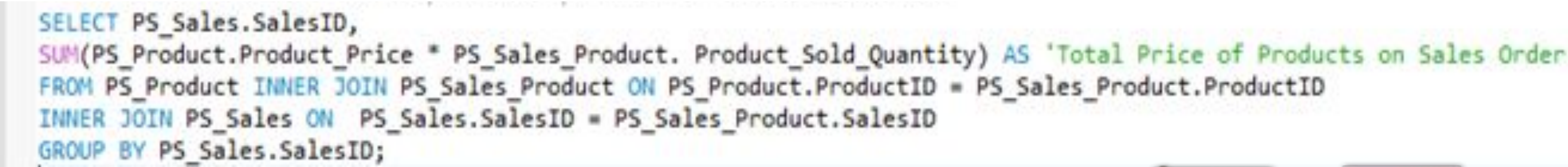
```
SELECT PS_Department.Department_Name, COUNT(EmployeeID) AS 'Number of Employees'
FROM PS_Employee INNER JOIN PS_Department
ON PS_Employee.Department_Name = PS_Department.Department_Name
GROUP BY PS_Department.Department_Name;
```



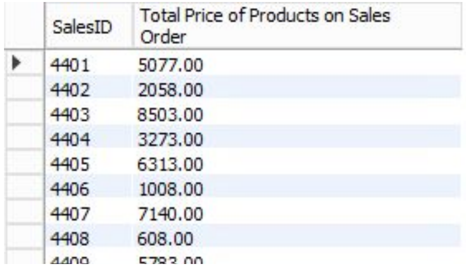
	Department_Name	Number of Employees
▶	Accounting	3
	HumanResources	4
	IT	3
	Maintainess	3
	Marketing	3
	Purchase	4
	R&D	3
	Sales	3
	Service	24

SQL Query #2

Total Revenue or sales from “Products” purchased by customers



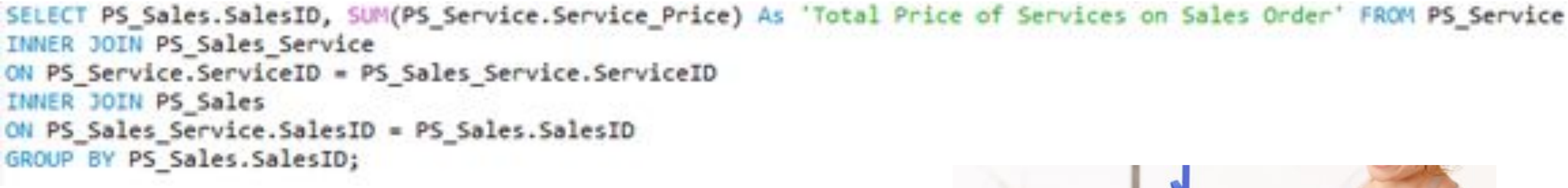
```
SELECT PS_Sales.SalesID,
SUM(PS_Product.Product_Price * PS_Sales_Product. Product_Sold_Quantity) AS 'Total Price of Products on Sales Order
FROM PS_Product INNER JOIN PS_Sales_Product ON PS_Product.ProductID = PS_Sales_Product.ProductID
INNER JOIN PS_Sales ON  PS_Sales.SalesID = PS_Sales_Product.SalesID
GROUP BY PS_Sales.SalesID;
```



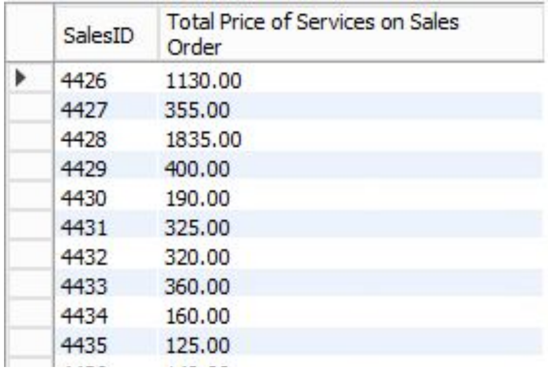
SalesID	Total Price of Products on Sales Order
4401	5077.00
4402	2058.00
4403	8503.00
4404	3273.00
4405	6313.00
4406	1008.00
4407	7140.00
4408	608.00
4409	5783.00

SQL Query #3

Total Revenue or Sales from “Services” availed by customers.



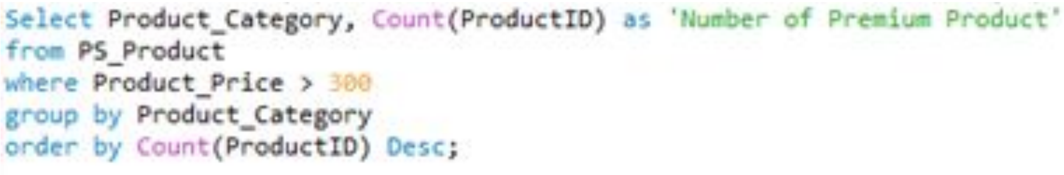
```
SELECT PS_Sales.SalesID, SUM(PS_Service.Service_Price) As 'Total Price of Services on Sales Order' FROM PS_Service
INNER JOIN PS_Sales_Service
ON PS_Service.ServiceID = PS_Sales_Service.ServiceID
INNER JOIN PS_Sales
ON PS_Sales_Service.SalesID = PS_Sales.SalesID
GROUP BY PS_Sales.SalesID;
```



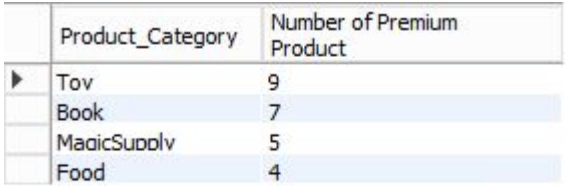
	SalesID	Total Price of Services on Sales Order
▶	4426	1130.00
	4427	355.00
	4428	1835.00
	4429	400.00
	4430	190.00
	4431	325.00
	4432	320.00
	4433	360.00
	4434	160.00
	4435	125.00

SQL Query #4

All the “Premium products” (products > $300) we offer from each Product category



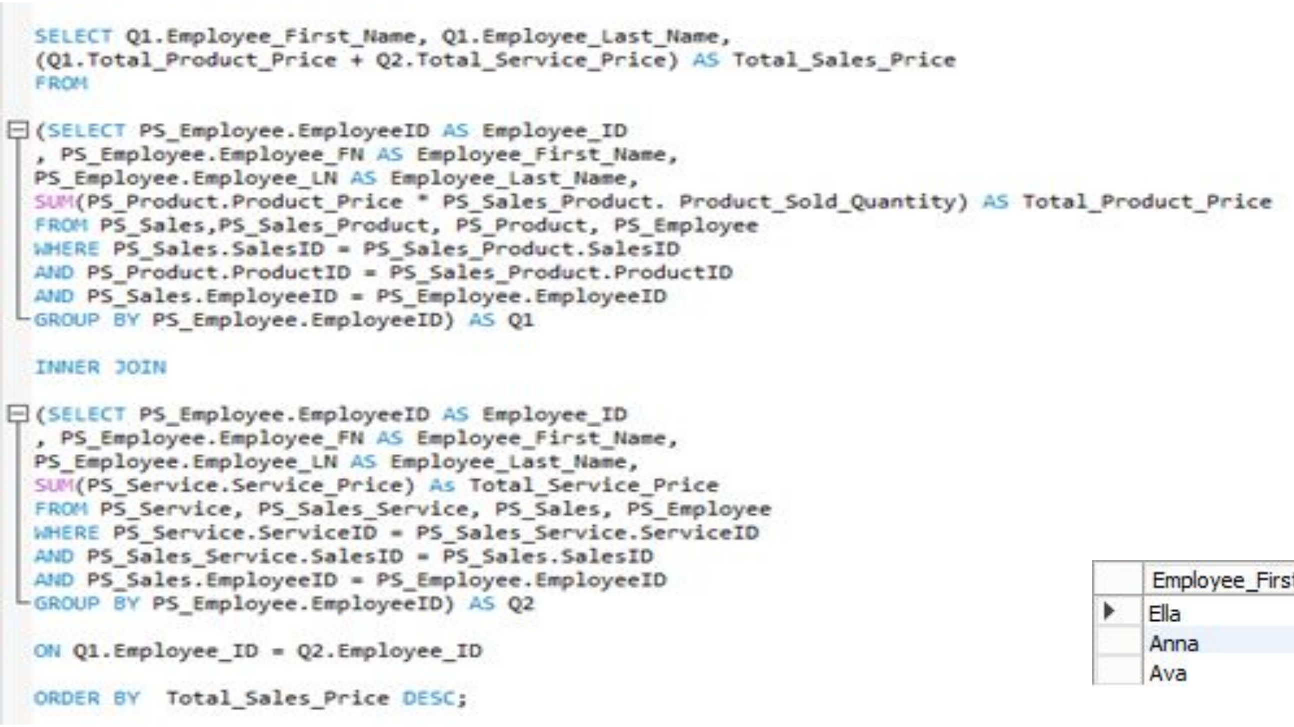
```
Select Product_Category, Count(ProductID) as 'Number of Premium Product'
from PS_Product
where Product_Price > 300
group by Product_Category
order by Count(ProductID) Desc;
```



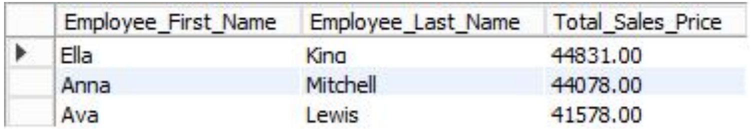
	Product_Category	Number of Premium Product
▶	Toy	9
	Book	7
	MagicSupply	5
	Food	4

SQL Query #5

Sales made by each employee(Sales department) along with their total sales



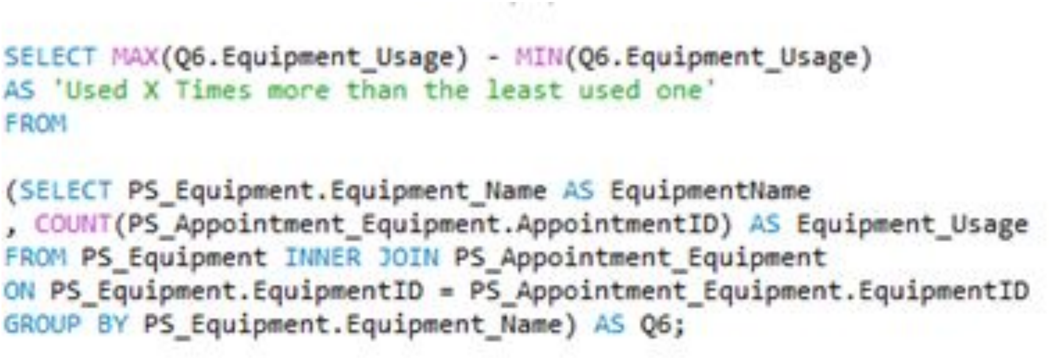
```
SELECT Q1.Employee_First_Name, Q1.Employee_Last_Name,
(Q1.Total_Product_Price + Q2.Total_Service_Price) AS Total_Sales_Price
FROM

(SELECT PS_Employee.EmployeeID AS Employee_ID
, PS_Employee.Employee_FN AS Employee_First_Name,
PS_Employee.Employee_LN AS Employee_Last_Name,
SUM(PS_Product.Product_Price * PS_Sales_Product. Product_Sold_Quantity) AS Total_Product_Price
FROM PS_Sales,PS_Sales_Product, PS_Product, PS_Employee
WHERE PS_Sales.SalesID = PS_Sales_Product.SalesID
AND PS_Product.ProductID = PS_Sales_Product.ProductID
AND PS_Sales.EmployeeID = PS_Employee.EmployeeID
GROUP BY PS_Employee.EmployeeID) AS Q1

INNER JOIN

(SELECT PS_Employee.EmployeeID AS Employee_ID
, PS_Employee.Employee_FN AS Employee_First_Name,
PS_Employee.Employee_LN AS Employee_Last_Name,
SUM(PS_Service.Service_Price) As Total_Service_Price
FROM PS_Service, PS_Sales_Service, PS_Sales, PS_Employee
WHERE PS_Service.ServiceID = PS_Sales_Service.ServiceID
AND PS_Sales_Service.SalesID = PS_Sales.SalesID
AND PS_Sales.EmployeeID = PS_Employee.EmployeeID
GROUP BY PS_Employee.EmployeeID) AS Q2

ON Q1.Employee_ID = Q2.Employee_ID

ORDER BY  Total_Sales_Price DESC;
```



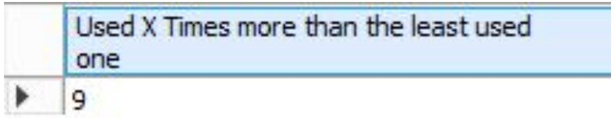
	Employee_First_Name	Employee_Last_Name	Total_Sales_Price
▶	Ella	King	44831.00
	Anna	Mitchell	44078.00
	Ava	Lewis	41578.00

SQL Query #6

Finding out how many times the most used equipment is used compared to the least used equipment



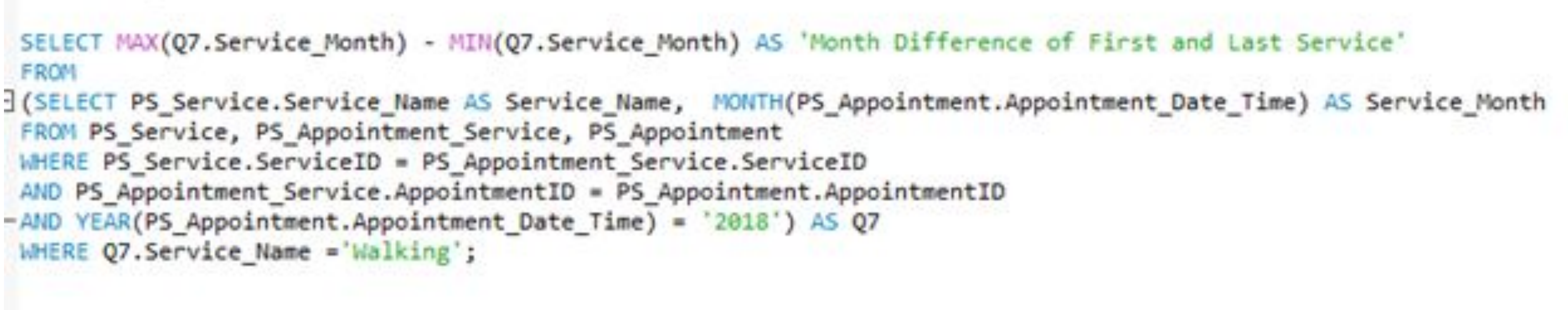
```
SELECT MAX(Q6.Equipment_Usage) - MIN(Q6.Equipment_Usage)
AS 'Used X Times more than the least used one'
FROM

(SELECT PS_Equipment.Equipment_Name AS EquipmentName
, COUNT(PS_Appointment_Equipment.AppointmentID) AS Equipment_Usage
FROM PS_Equipment INNER JOIN PS_Appointment_Equipment
ON PS_Equipment.EquipmentID = PS_Appointment_Equipment.EquipmentID
GROUP BY PS_Equipment.Equipment_Name) AS Q6;
```



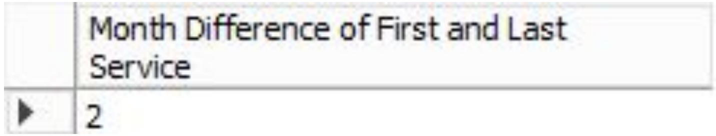
```
Used X Times more than the least used
one
9
```

SQL Query #7

In financial year 2018, how many months has it been since “ Walking service” has been offered to a customer between their first and last visit?



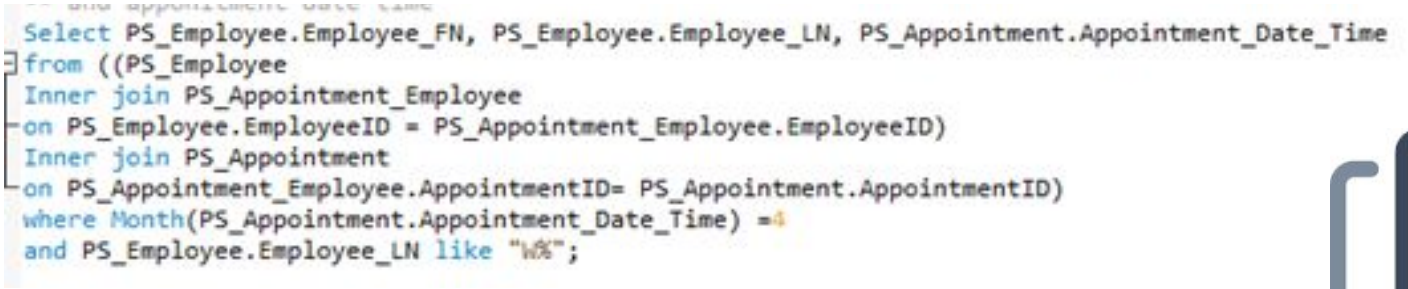
```sql
SELECT MAX(Q7.Service_Month) - MIN(Q7.Service_Month) AS 'Month Difference of First and Last Service'
FROM
(SELECT PS_Service.Service_Name AS Service_Name,  MONTH(PS_Appointment.Appointment_Date_Time) AS Service_Month
FROM PS_Service, PS_Appointment_Service, PS_Appointment
WHERE PS_Service.ServiceID = PS_Appointment_Service.ServiceID
AND PS_Appointment_Service.AppointmentID = PS_Appointment.AppointmentID
AND YEAR(PS_Appointment.Appointment_Date_Time) = '2018') AS Q7
WHERE Q7.Service_Name ='Walking';
```



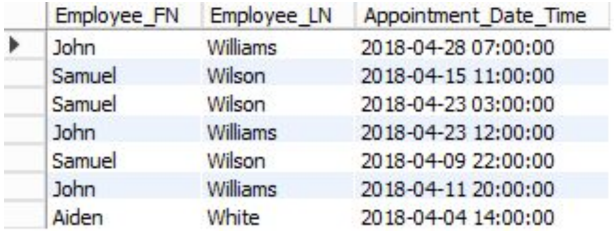
Month Difference of First and Last Service
2

SQL Query #8

#8 The appointment details in the month of April for an employee whose last name starts with “W”



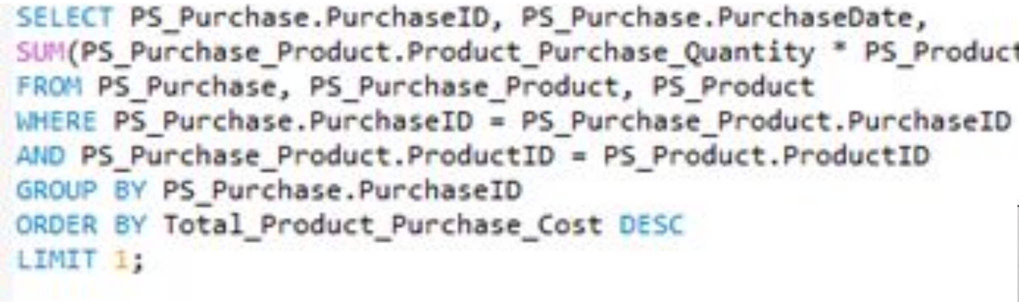
```
-- and appointment date time
Select PS_Employee.Employee_FN, PS_Employee.Employee_LN, PS_Appointment.Appointment_Date_Time
from ((PS_Employee
Inner join PS_Appointment_Employee
on PS_Employee.EmployeeID = PS_Appointment_Employee.EmployeeID)
Inner join PS_Appointment
on PS_Appointment_Employee.AppointmentID= PS_Appointment.AppointmentID)
where Month(PS_Appointment.Appointment_Date_Time) =4
and PS_Employee.Employee_LN like "W%";
```



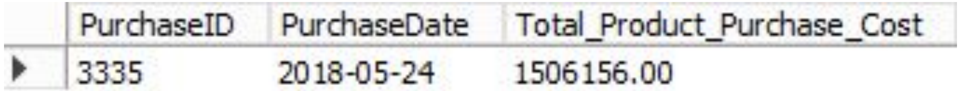
Employee_FN	Employee_LN	Appointment_Date_Time
John	Williams	2018-04-28 07:00:00
Samuel	Wilson	2018-04-15 11:00:00
Samuel	Wilson	2018-04-23 03:00:00
John	Williams	2018-04-23 12:00:00
Samuel	Wilson	2018-04-09 22:00:00
John	Williams	2018-04-11 20:00:00
Aiden	White	2018-04-04 14:00:00

SQL Query #9

The Purchase order with the highest cost to the store, date of purchase order and the total cost of purchase



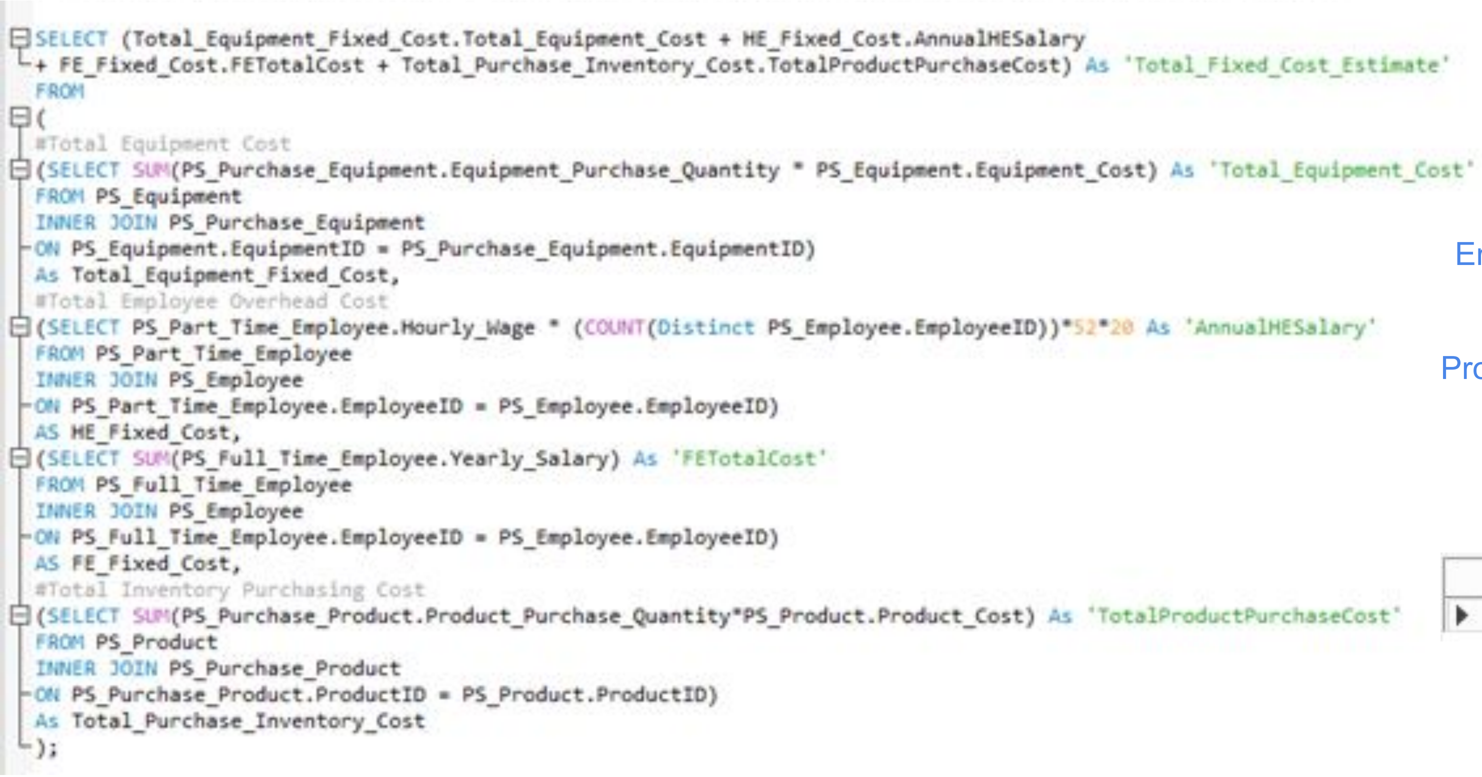
```
SELECT PS_Purchase.PurchaseID, PS_Purchase.PurchaseDate,
SUM(PS_Purchase_Product.Product_Purchase_Quantity * PS_Product.Product_Cost) AS Total_Product_Purchase_Cost
FROM PS_Purchase, PS_Purchase_Product, PS_Product
WHERE PS_Purchase.PurchaseID = PS_Purchase_Product.PurchaseID
AND PS_Purchase_Product.ProductID = PS_Product.ProductID
GROUP BY PS_Purchase.PurchaseID
ORDER BY Total_Product_Purchase_Cost DESC
LIMIT 1;
```



PurchaseID	PurchaseDate	Total_Product_Purchase_Cost
3335	2018-05-24	1506156.00

SQL Query #10

Estimate of ‘Total Costs’ to the Pet Store for the year 2018



```sql
SELECT (Total_Equipment_Fixed_Cost.Total_Equipment_Cost + HE_Fixed_Cost.AnnualHESalary
+ FE_Fixed_Cost.FETotalCost + Total_Purchase_Inventory_Cost.TotalProductPurchaseCost) As 'Total_Fixed_Cost_Estimate'
FROM
(
#Total Equipment Cost
(SELECT SUM(PS_Purchase_Equipment.Equipment_Purchase_Quantity * PS_Equipment.Equipment_Cost) As 'Total_Equipment_Cost'
FROM PS_Equipment
INNER JOIN PS_Purchase_Equipment
ON PS_Equipment.EquipmentID = PS_Purchase_Equipment.EquipmentID)
As Total_Equipment_Fixed_Cost,
#Total Employee Overhead Cost
(SELECT PS_Part_Time_Employee.Hourly_Wage * (COUNT(Distinct PS_Employee.EmployeeID))*52*20 As 'AnnualHESalary'
FROM PS_Part_Time_Employee
INNER JOIN PS_Employee
ON PS_Part_Time_Employee.EmployeeID = PS_Employee.EmployeeID)
AS HE_Fixed_Cost,
(SELECT SUM(PS_Full_Time_Employee.Yearly_Salary) As 'FETotalCost'
FROM PS_Full_Time_Employee
INNER JOIN PS_Employee
ON PS_Full_Time_Employee.EmployeeID = PS_Employee.EmployeeID)
AS FE_Fixed_Cost,
#Total Inventory Purchasing Cost
(SELECT SUM(PS_Purchase_Product.Product_Purchase_Quantity*PS_Product.Product_Cost) As 'TotalProductPurchaseCost'
FROM PS_Product
INNER JOIN PS_Purchase_Product
ON PS_Purchase_Product.ProductID = PS_Product.ProductID)
As Total_Purchase_Inventory_Cost
);
```

Equipment Cost

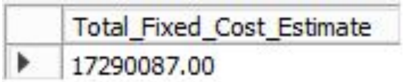
+

Employee Overhead Cost

+

Product Inventory Purchase Cost

=



	Total_Fixed_Cost_Estimate
▶	17290087.00

Thank You!