**Computers by Ralph  
Forrest Fallon**

**Summary** The world of personal computers is one with thousands of variables; this is often the case when a functioning product requires dozens of parts with dozens of options pertaining to what a person needs out of their computer. My imaginary friend, Ralph, has been operating a pc-building business for about a year now, and his customer base grows weekly.

Ralph consults with his customers about what they need their computer to be capable of and what their budget is. After the initial meeting, Ralph will use the parts available in his inventory to construct the computer his customer has agreed upon.

Ralph has a few different means of purchasing computer parts for a customer as well as maintaining a base inventory. Ralph considers the following to be necessary for a functioning office PC: case, motherboard, RAM, storage, and the processor. For a gaming PC, Ralph simply adds on a graphics card and a better processor cooler. Ralph purchases parts from Cosmo’s Parts Store in his city and will purchase parts in bulk, so he has plenty of options on hand. Cosmo knows Ralph well, and often calls him with updates on stock. In a way, Cosmo relies on Ralph for advertising, as does Ralph rely on Cosmo for the same. Their relationship is surprisingly mutually beneficial; Ralph provides the one-on-one customer service that Cosmo’s store is just not capable of doing, but at the same time Ralph recommends customers to Cosmo’s for specialty repair services or parts that deal with “after-market” customizations.

When Ralph reaches a low count of any given part, he will go out and buy that part in bulk. The low counts of each part are listed in the business rules. If he does not have a part on hand, he must go out and buy this part in bulk which always results in the customer delivery date being delayed.

What Ralph is struggling with: Ralph feels as though his re-order points are not weighted correctly to correlate with the popularity of each product. He finds himself going to the store to purchase specific products more often than others and feels as though his reorder points should reflect that better. Without a real-time inventory tracking system, Ralph merely hopes that he has the inventory on hand to fulfil his current orders. Not only does he want a better grasp on his re-order points, but he wants to in-turn decrease the average wait time his customers have. He does not have the time to figure this out himself, so he needs a database created to show the popularity of each package, the total parts used within given timeframes, total parts on hand, number of re-orders per part, and average time to deliver.

**Post Summary**

Ralph now has a database! Though every database is a work in progress, Ralph now has a digital ledger that he can keep better track of his order histories from. For now, Ralph has to manually enter his orders and inventory additions/subtractions, however, with time we can build more advanced inputs to speed up that process.

I was able to answer data questions from Part 1 via advanced SELECT queries that presented sums, counts, top 5’s, and detailed inventory counts given specific occurrences. These SELECTS will grant insight for Ralph that he would have to work hours for on pen and paper; some of them also show short comings in his product offerings!

**Data Questions**

1. **What is Ralph’s most popular product? Least?**
2. **What is the percentage breakdown of each product purchased from Ralph?**
3. **What parts are re-ordered the most? Least?**
4. **How many orders can Ralph handle at his current re-order points before he runs out of stock?**
5. **Are there any products that Ralph should only buy parts for at a per order basis?**

**Stakeholders**

1. Ralph: business owner
2. Customers: computer purchasers
3. Cosmo’s Part Store: Computer parts supplier
4. Computer parts manufacturers
5. Shipping Companies

**Business Rules**

1. An office PC must have a case, motherboard, ram, storage, and processor.
2. A Gaming PC must have a case, motherboard, ram, storage, processor, graphics card, and processor cooling system.
3. A customer must present their full name, phone number, mailing address, and email address for an order.
4. A customer must choose at least one of 6 available options under Ralph’s offerings.
5. A computer part must have a name and Part Tier attached.
6. Cosmo’s parts store orders from Ralph must have transaction dates and amounts purchased.
7. A delivery date must have a date ordered, a date delivered, and a summation of days between the order and delivery date.
8. Ralph’s current average days-to-delivery is 4 business days.
9. The PC packages currently available through Ralph include low-tier office PC, mid-tier office PC, high-tier office PC, low-tier gaming PC, mid-tier gaming PC, high-tier gaming PC.
   1. Office PC
      1. Low Tier: Tier 1 case, Tier 1 motherboard, Tier 1 RAM, Tier 1 storage, Tier 1 processor
      2. Mid-Tier: Tier 1 case, Tier 1 motherboard, Tier 1 RAM, Tier 2 storage, Tier 2 processor
      3. High Tier: Tier 2 case, Tier 2 motherboard, Tier 2 RAM, Tier 3 storage, Tier 3 processor
   2. Gaming PC
      1. Low Tier: Tier 1 case, Tier 1 motherboard, Tier 1 RAM, Tier 1 Storage, Tier 1 processor, Tier 1 graphics, Tier 1 cooling system
      2. Mid-Tier: Tier 2 case, Tier 2 motherboard, Tier 2 RAM, Tier 2 Storage, Tier 2 processor, Tier 2 graphics, Tier 2 cooling system
      3. High Tier: Tier 3 case, Tier 3 motherboard, Tier 3 RAM, Tier 3 Storage, Tier 3 processor, Tier 3 graphics, Tier 3 cooling system
10. Low counts:
    1. Cases: 9 cases (3 of each different flow tier)
    2. Motherboards: 9 motherboards (3 of each different tier)
    3. RAM: 12 units (4 of each tier)
    4. Processor: 12 chips (4 of each different tier)
    5. Graphics cards: 6 cards (2 of each different tier)
    6. Cooling systems: 6 systems (2 of each different tier)
11. Customer name, phone number, email, mailing address, and payment method are stored in Ralph’s books. He does not need any more info on the customer than that, but Ralph also stores the “build” and parts used on each customer profile once complete.
12. Customers are able to order more than one PC from Ralph at a time.

**Glossary**

A **Computer Part (often referred to throughout this project as a “part”)** is a modular, individual piece required to make a functioning computer. Each part in Ralph’s offerings is needed to complete a PC build.A **PC** is a personal computer, the summation of the following:

1. **Case:** where all the computer parts are mounted and safely encased. Higher end cases feature better air flow for cooling off the computer parts, more options for mounting of parts, and cosmetic features as well.
2. **Motherboard**: the hub for all the components. Higher quality motherboards contain features such as more ports for parts, Wi-Fi on board, and cosmetic RGB lighting.
3. **RAM:** Also known as random access memory, this is what the computer uses to multitask efficiently. Every software program uses a small part of this memory, and more intensive programs (video games, video/sound editing software) use a large majority of available RAM in most systems. The more RAM, the more the computer can handle at once while maintaining operating speed.
4. **Hard Drive:** Not to be confused with RAM, hard drives are what determine the number of files a computer can store. Ralph currently offers two types of hard drive, as detailed below. It is worth noting that Ralph’s offerings deal in **gb’s (gigabytes, 1024 megabytes) and tb’s (terabytes, 1024 gigabytes).**
5. **Processor:** Also known as the “computer chip.” A processor is the conductor of the orchestra of parts on board. Processors are often graded on their GHz, or gigahertz, which are explained below. The higher the GHz a processor has, the faster and more capable that processor is for completing tasks.
   * **GHz/Gigahertz:** A unit used to measure the amount of “clock ticks” a processor can provide in a single second. If a processor is rated at 3.0-GHz, its clock is ticking 3.1 billion times per second.
6. **Graphics Card:** Also known as a video card or display card, these are what allows for real-time rendering of graphics created by video games or video editing software and display them to a monitor. For the sake of this project, graphics card grading has been simplified to tier 1, 2, or 3 series cards, with Tier 1 being the lowest quality and Tier 3 being the highest quality. Higher quality graphics cards are capable of rendering higher resolution images at higher speeds than their less capable counterparts.
7. **Cooling System:** When using a computer for gaming or editing of some kind, the processor chip is often working as hard as it possibly can. Because of this, an immense amount of heat is produced by the chip, and without proper cooling systems in place, this decreases the lifespan of the processor chip. Ralph offers a tier 1, tier 2, and tier 3 cooling system which cool the processor to cooler temperatures as the tier goes up.
8. **Part Tier:** As described in each of the above, computer parts in Ralph’s business have a Tier rating 1-3. 1 being the lowest, cheapest option, and 3 being the most fully featured product.
9. **Part Type:** As shown in the conceptual model, parts are labeled as Part Type. This includes motherboard, case, storage, processor, RAM, graphics card, and cooling system.
10. **Kit Type:** Each of Ralph’s offerings either lie in the OfficeKit or GamingKit. These “kits” are a summation of parts needed to complete each kit.

**Conceptual Model**

**Diagram

Description automatically generated**

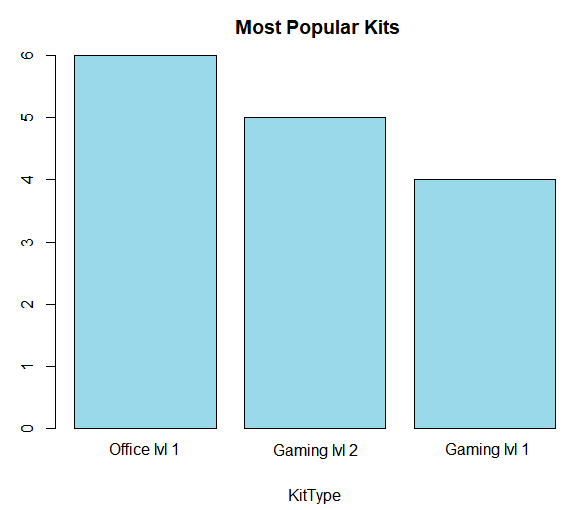
**Logical Model  
Diagram

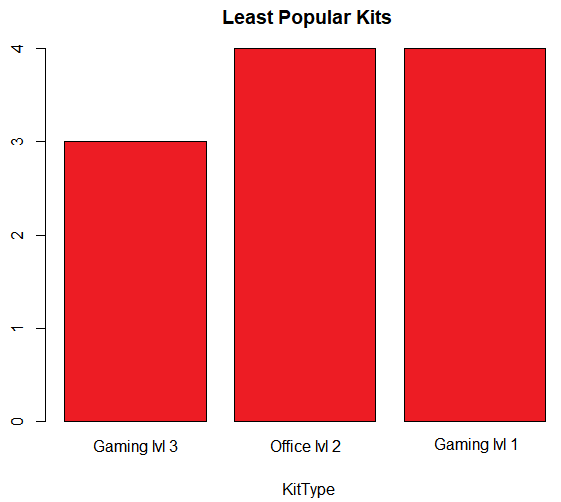
Description automatically generated**

**Data Questions**

**1. What is Ralph’s most popular product? Least?**

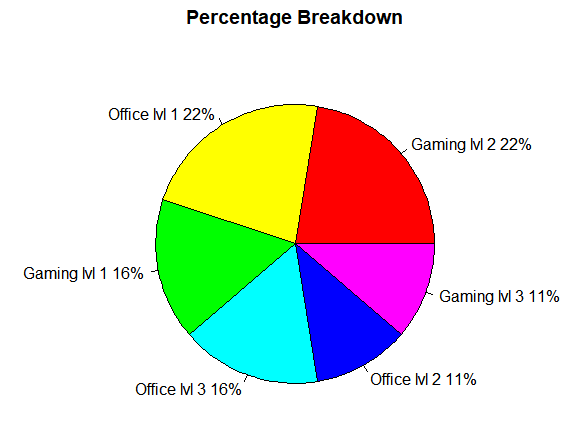
Via illustration tools with Rstudio, I was able to create the following charts for Ralph’s most and least popular orders:



****

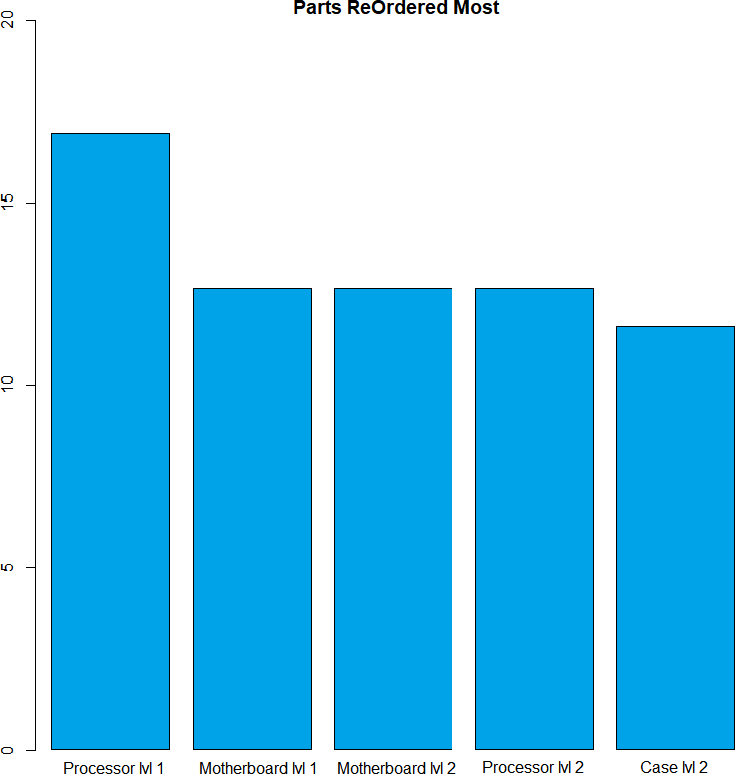
**2. What is the percentage breakdown of each product purchased from Ralph?**

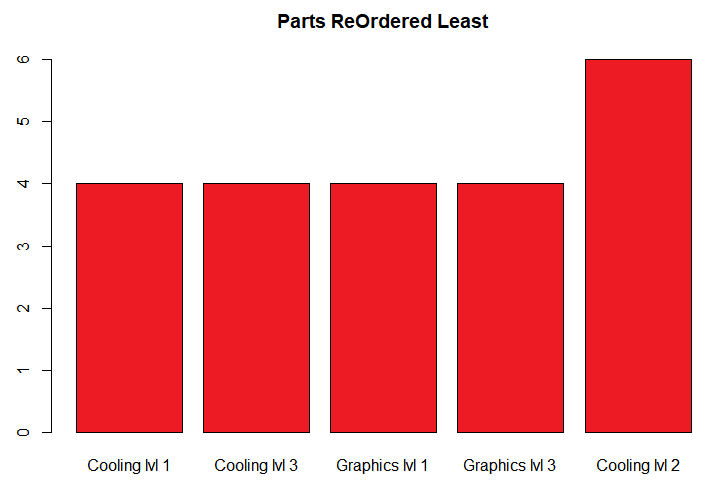
As shown below, Office lvl 1 and Gaming lv l 2 are currently tied! Using this pie chart, Ralph can easily see where his products are ranked, and maybe even plan inventory counts accordingly.

****

1. **What parts are re-ordered the most? Least?**

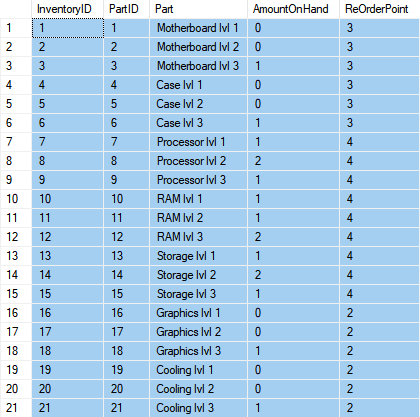
Thought it can usually be assumed that the cheapest offerings of any business are ordered the most, visualizing that data never hurts. From the graphs below, Ralph can assume without a doubt that the Processor lvl 1 should be prioritized, and parts for Gaming PC’s could perhaps be ordered less.

****

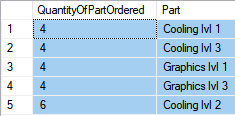
****

1. **How many orders can Ralph handle at his current re-order points before he runs out of stock?**

As shown in the SQL script, the below inventory count resulted from just 8 orders for Ralph. There are enough parts here to create another PC, however, the parts listed as available here do not pertain to one of Ralph’s products. Perhaps he should create another!

****

1. **Are there any products that Ralph should only buy parts for at a per order basis?**

****

By using a similar select statement for data question number 3, we can see that Ralph has only ordered 4 Cooling lvl 1, Cooling lvl 3, Graphics lvl 1, and Graphics lvl 3 parts since the new year. It seems as though Gaming PC’s are not Ralph’s top product, and he should consider adjusting his re-order points for these parts lest he end up with too many on hand.

**Reflection** I had a feeling my project idea would be a bit complicated seeing as though the number of tables I had was right on the line of what was acceptable. The number of relationships and dependencies present here made for quite the challenge both logically and semantically. At times I certainly considered taking the easy way out via simplifying the data, however, I wanted to see exactly what I was capable of.

It quickly became apparent that an inventory system in SQL requires a bit more coding prowess than I was able to accomplish in the timeframe of this class. Most of the inputs I have in my code are manual and would be much more efficient with more advanced SQL commands. For example, the coding process would have been much faster (and a better procedure for Ralph) if the Orders inputs automatically subtracted from the Inventory upon entry. That is not the fault of the curriculum though! Again, that goes back to the complicated nature of my project.

The class curriculum helped the most in way of creating functions and views that allowed for easier inputs for mass data inputs, my InventoryAddition/InventorySubtraction commands immediately come to mind. I had some entry level SQL experience coming into this class, so learning new functions and views was a welcomed addition to the arsenal!

Overall, this project was a great learning experience. I now know the scope of my abilities via SQL and am more confident in my ability to use it. I am also keenly aware of my limitations and know where to focus my learning efforts to improve my SQL knowledge.

**SQL Code**

--Drop procedures, views, formulas

drop function if exists MonthlyOrders

go

drop view if exists OutstandingOrders

go

drop procedure if exists InventoryAddition

go

drop procedure if exists InventorySubtraction

go

drop function if exists OrderReport

go

--drop tables

--Drop Orders Table

drop table if exists Orders

go

--Drop Customer Table

drop table if exists Customer

go

--Drop ComputerKitParts Table

drop table if exists ComputerKitParts

go

--Drop ComputerKit Table

drop table if exists ComputerKit

go

--Drop KitType Table

drop table if exists KitType

go

--Drop Inventory Table

drop table if exists Inventory

go

--Drop CosmoOrders Table

drop table if exists CosmoOrders

go

--Drop Parts Table

drop table if exists Parts

go

--Drop PartsTier Table

drop table if exists PartTier

go

--Drop PartType Table

drop table if exists PartType

go

--Create PartType table

create table PartType (

PartTypeID int identity,

PartName varchar(30) not null,

constraint PK\_PartType Primary Key (PartTypeID)

)

go

--Create PartTier table

create table PartTier (

PartTierID int identity,

Tier varchar(1) not null,

constraint PK\_PartTier primary key (PartTierID)

)

go

--Create Parts table

create table Parts (

PartID int identity,

Part varchar(20),

PartTypeID int not null,

PartTierID int not null,

constraint PK\_Part primary key (PartID),

constraint FK1\_Part foreign key (PartTypeID) references PartType(PartTypeID),

constraint FK2\_Part foreign key (PartTierID) references PartTier(PartTierID)

)

go

--Create CosmoOrders table

create table CosmoOrders (

CosmoID int identity,

DatePurchased date not null default GetDate(),

QuantityPurchased numeric not null,

PartID int,

constraint PK\_CosmoID primary key (CosmoID),

constraint FK1\_CosmoID foreign key (PartID) references Parts(PartID)

)

go

--Create Inventory table

create table Inventory (

InventoryID int identity,

PartID int,

Part varchar(20),

AmountOnHand numeric not null,

ReOrderPoint numeric not null,

constraint PK\_Inventory primary key (InventoryID),

constraint FK1\_Inventory foreign key (PartID) references Parts(PartID),

)

go

--Create KitType table

create table KitType (

KitTypeID int identity,

KitType varchar(15) not null,

constraint PK\_KitType primary key (KitTypeID)

)

go

--Create ComputerKit table

create table ComputerKit (

ComputerKitID int identity,

KitTypeID int not null,

constraint PK\_ComputerKit primary key (ComputerKitID),

constraint FK\_ComputerKit foreign key (KitTypeID) references KitType(KitTypeID)

)

go

--Create ComputerKitParts table

create table ComputerKitParts (

ComputerKitPartsID int identity,

ComputerKitID int not null,

PartID int not null,

constraint PK\_ComputerKitParts primary key (ComputerKitPartsID),

constraint FK1\_ComputerKitParts foreign key (ComputerKitID) references ComputerKit(ComputerKitID),

constraint FK2\_ComputerKitParts foreign key (PartID) references Parts(PartID)

)

go

--Create Orders table

create table Orders (

OrdersID int identity,

ComputerKitID int not null,

CustomerID int not null,

OrderDate datetime default GetDate(),

DeliveryDate datetime default GetDate(),

TimeToDelivery numeric,

constraint PK\_Orders primary key (OrdersID)

)

go

--Create Customer table

create table Customer (

CustomerID int identity,

CustName varchar(30) not null,

Phone varchar(10) not null,

Email varchar(30) not null,

MailingAddress varchar(50) not null,

PaymentMethod varchar(20) not null,

constraint PK\_Customer primary key (CustomerID),

)

go

alter table Orders add constraint

FK1\_Orders foreign key (CustomerID) references Customer(CustomerID)

go

--Insert PartType values

insert into PartType (PartName) values ('Motherboard lvl 1'), ('Motherboard lvl 2'), ('Motherboard lvl 3'),

('Case lvl 1'), ('Case lvl 2'), ('Case lvl 3'),

('Processor lvl 1'), ('Processor lvl 2'), ('Processor lvl 3'),

('Ram lvl 1'), ('Ram lvl 2'), ('Ram lvl 3'),

('Storage lvl 1'), ('Storage lvl 2'), ('Storage lvl 3'),

('Graphics lvl 1'), ('Graphics lvl 2'), ('Graphics lvl 3'),

('Cooling lvl 1'), ('Cooling lvl 2'), ('Cooling lvl 3')

go

--Insert PartTier values

insert into PartTier (Tier) values ('1'), ('2'), ('3')

go

--Insert Parts values

insert into Parts (PartTypeID, PartTierID) values ('1', '1'), ('2', '2'), ('3', '3'),

('4', '1'), ('5', '2'), ('6', '3'),

('7', '1'), ('8', '2'), ('9', '3'),

('10', '1'), ('11', '2'), ('12', '3'),

('13', '1'), ('14', '2'), ('15', '3'),

('16', '1'), ('17', '2'), ('18', '3'),

('19', '1'), ('20', '2'), ('21', '3')

go

--Insert CosmoOrders values

insert into CosmoOrders (DatePurchased, QuantityPurchased, PartID) values ('01/01/2020', '3', '1'), ('01/01/2020', '3', '2'), ('01/01/2020', '3', '3'),

('01/01/2020', '3', '4'), ('01/01/2020', '3', '5'), ('01/01/2020', '3', '6'),

('01/01/2020', '4', '7'), ('01/01/2020', '4', '8'), ('01/01/2020', '4', '9'),

('01/01/2020', '4', '10'), ('01/01/2020', '4', '11'), ('01/01/2020', '4', '12'),

('01/01/2020', '3', '13'), ('01/01/2020', '3', '14'), ('01/01/2020', '3', '15'),

('01/01/2020', '2', '16'), ('01/01/2020', '2', '17'), ('01/01/2020', '2', '18'),

('01/01/2020', '2', '19'), ('01/01/2020', '2', '20'), ('01/01/2020', '2', '21')

go

--Update Parts table with Part names

update Parts set Part = 'Motherboard lvl 1' where PartID = 1

update Parts set Part = 'Motherboard lvl 2' where PartID = 2

update Parts set Part = 'Motherboard lvl 3' where PartID = 3

update Parts set Part = 'Case lvl 1' where PartID = 4

update Parts set Part = 'Case lvl 2' where PartID = 5

update Parts set Part = 'Case lvl 3' where PartID = 6

update Parts set Part = 'Processor lvl 1' where PartID = 7

update Parts set Part = 'Processor lvl 2' where PartID = 8

update Parts set Part = 'Processor lvl 3' where PartID = 9

update Parts set Part = 'RAM lvl 1' where PartID = 10

update Parts set Part = 'RAM lvl 2' where PartID = 11

update Parts set Part = 'RAM lvl 3' where PartID = 12

update Parts set Part = 'Storage lvl 1' where PartID = 13

update Parts set Part = 'Storage lvl 2' where PartID = 14

update Parts set Part = 'Storage lvl 3' where PartID = 15

update Parts set Part = 'Graphics lvl 1' where PartID = 16

update Parts set Part = 'Graphics lvl 2' where PartID = 17

update Parts set Part = 'Graphics lvl 3' where PartID = 18

update Parts set Part = 'Cooling lvl 1' where PartID = 19

update Parts set Part = 'Cooling lvl 2' where PartID = 20

update Parts set Part = 'Cooling lvl 3' where PartID = 21

go

--Insert KitType values

insert into KitType(KitType) values ('Office lvl 1'), ('Office lvl 2'),

('Office lvl 3'), ('Gaming lvl 1'), ('Gaming lvl 2'), ('Gaming lvl 3')

go

--Insert ComputerKit values

insert into ComputerKit values (1), (2), (3), (4), (5), (6)

go

--Insert ComputerKitParts values

insert into ComputerKitParts(ComputerKitID, PartID) values

((select ComputerKitID from ComputerKit where KitTypeID = 1), (select PartID from Parts where Part = 'Motherboard lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 1), (select PartID from Parts where Part = 'Case lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 1), (select PartID from Parts where Part = 'Processor lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 1), (select PartID from Parts where Part = 'RAM lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 1), (select PartID from Parts where Part = 'Storage lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 2), (select PartID from Parts where Part = 'Motherboard lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 2), (select PartID from Parts where Part = 'Case lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 2), (select PartID from Parts where Part = 'Processor lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 2), (select PartID from Parts where Part = 'RAM lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 2), (select PartID from Parts where Part = 'Storage lvl 2' )),

((select ComputerKitID from ComputerKit where KitTypeID = 3), (select PartID from Parts where Part = 'Motherboard lvl 2' )),

((select ComputerKitID from ComputerKit where KitTypeID = 3), (select PartID from Parts where Part = 'Case lvl 2' )),

((select ComputerKitID from ComputerKit where KitTypeID = 3), (select PartID from Parts where Part = 'Processor lvl 2' )),

((select ComputerKitID from ComputerKit where KitTypeID = 3), (select PartID from Parts where Part = 'RAM lvl 3' )),

((select ComputerKitID from ComputerKit where KitTypeID = 3), (select PartID from Parts where Part = 'Storage lvl 3' )),

((select ComputerKitID from ComputerKit where KitTypeID = 4), (select PartID from Parts where Part = 'Motherboard lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 4), (select PartID from Parts where Part = 'Case lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 4), (select PartID from Parts where Part = 'Processor lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 4), (select PartID from Parts where Part = 'RAM lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 4), (select PartID from Parts where Part = 'Storage lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 4), (select PartID from Parts where Part = 'Graphics lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 4), (select PartID from Parts where Part = 'Cooling lvl 1' )),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (select PartID from Parts where Part = 'Motherboard lvl 2' )),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (select PartID from Parts where Part = 'Case lvl 2' )),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (select PartID from Parts where Part = 'Processor lvl 2' )),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (select PartID from Parts where Part = 'RAM lvl 2' )),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (select PartID from Parts where Part = 'Storage lvl 2' )),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (select PartID from Parts where Part = 'Graphics lvl 2' )),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (select PartID from Parts where Part = 'Cooling lvl 2' )),

((select ComputerKitID from ComputerKit where KitTypeID = 6), (select PartID from Parts where Part = 'Motherboard lvl 3' )),

((select ComputerKitID from ComputerKit where KitTypeID = 6), (select PartID from Parts where Part = 'Case lvl 3' )),

((select ComputerKitID from ComputerKit where KitTypeID = 6), (select PartID from Parts where Part = 'Processor lvl 3' )),

((select ComputerKitID from ComputerKit where KitTypeID = 6), (select PartID from Parts where Part = 'RAM lvl 3' )),

((select ComputerKitID from ComputerKit where KitTypeID = 6), (select PartID from Parts where Part = 'Storage lvl 3' )),

((select ComputerKitID from ComputerKit where KitTypeID = 6), (select PartID from Parts where Part = 'Graphics lvl 3' )),

((select ComputerKitID from ComputerKit where KitTypeID = 6), (select PartID from Parts where Part = 'Cooling lvl 3' ))

go

--Insert Inventory values

insert into Inventory(PartID, Part, AmountOnHand, ReOrderPoint) values

((select PartID from Parts where Part = 'Motherboard lvl 1'), (select Part from Parts where PartID = 1), 6, 3),

((select PartID from Parts where Part = 'Motherboard lvl 2'), (select Part from Parts where PartID = 2), 4, 3),

((select PartID from Parts where Part = 'Motherboard lvl 3'), (select Part from Parts where PartID = 3), 3, 3),

((select PartID from Parts where Part = 'Case lvl 1'), (select Part from Parts where PartID = 4), 7, 3),

((select PartID from Parts where Part = 'Case lvl 2'), (select Part from Parts where PartID = 5), 4, 3),

((select PartID from Parts where Part = 'Case lvl 3'), (select Part from Parts where PartID = 6), 3, 3),

((select PartID from Parts where Part = 'Processor lvl 1'), (select Part from Parts where PartID = 7), 7, 4),

((select PartID from Parts where Part = 'Processor lvl 2'), (select Part from Parts where PartID = 8), 5, 4),

((select PartID from Parts where Part = 'Processor lvl 3'), (select Part from Parts where PartID = 9), 4, 4),

((select PartID from Parts where Part = 'RAM lvl 1'), (select Part from Parts where PartID = 10), 8, 4),

((select PartID from Parts where Part = 'RAM lvl 2'), (select Part from Parts where PartID = 11), 5, 4),

((select PartID from Parts where Part = 'RAM lvl 3'), (select Part from Parts where PartID = 12), 4, 4),

((select PartID from Parts where Part = 'Storage lvl 1'), (select Part from Parts where PartID = 13), 6, 4),

((select PartID from Parts where Part = 'Storage lvl 2'), (select Part from Parts where PartID = 14), 5, 4),

((select PartID from Parts where Part = 'Storage lvl 3'), (select Part from Parts where PartID = 15), 4, 4),

((select PartID from Parts where Part = 'Graphics lvl 1'), (select Part from Parts where PartID = 16), 5, 2),

((select PartID from Parts where Part = 'Graphics lvl 2'), (select Part from Parts where PartID = 17), 4, 2),

((select PartID from Parts where Part = 'Graphics lvl 3'), (select Part from Parts where PartID = 18), 3, 2),

((select PartID from Parts where Part = 'Cooling lvl 1'), (select Part from Parts where PartID = 19), 5, 2),

((select PartID from Parts where Part = 'Cooling lvl 2'), (select Part from Parts where PartID = 20), 4, 2),

((select PartID from Parts where Part = 'Cooling lvl 3'), (select Part from Parts where PartID = 21), 3, 2)

go

-- existing customers before the new year

insert into Customer(CustName, Phone, Email, MailingAddress, PaymentMethod) values

('John Johnson', '5555555', 'johnj@email.com', '123 Smith St.', 'Credit Card'),

('Silvia Green', '6660000', 'sgreen@email.com', '246 Ace Rd.', 'Credit Card'),

('Boris Smith', '1006611', 'Bsmith@email.com', '157 Green Dr.', 'Cash'),

('Yancy Day', '1113399', 'YDay@email.com', '357 Day Dr.', 'Debit Card'),

('Steve Bruce', '3339999', 'Brucey@email.com', '468 Spruce Ct.', 'Cash'),

('Yennifer North', '5555533', 'Yenn@email.com', '3456 Spring Ct.', 'Credit Card'),

('Lenny Boy', '2222222', 'Lenny6@email.com', '5555 Spring Ct.', 'Check'),

('Victoria Smek', '4444444', 'vicsmek@email.com', '235 Right Way', 'Cash'),

('Truit Russ', '4444455', 'truitbusiness@email.com', '356 Industry Dr.', 'Credit Card'),

('Holly Flax', '5559988', 'Hflax@email.com', '4367 Forest Dr.', 'Credit Card')

go

--first orders of the new year

insert into Orders(ComputerKitID, CustomerID, OrderDate, DeliveryDate, TimeToDelivery) values

((select ComputerKitID from ComputerKit where KitTypeID = 4), (select CustomerID from Customer where CustName = 'Steve Bruce'), '01/05/2020', '01/09/2020', 4),

((select ComputerKitID from ComputerKit where KitTypeID = 3), (select CustomerID from Customer where CustName = 'Yennifer North'), '01/06/2020', '01/10/2020', 4),

((select ComputerKitID from ComputerKit where KitTypeID = 1), (select CustomerID from Customer where CustName = 'Lenny Boy'), '01/03/2020', '01/06/2020', 3),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (select CustomerID from Customer where CustName = 'Victoria Smek'), '01/07/2020', '01/11/2020', 4),

((select ComputerKitID from ComputerKit where KitTypeID = 6), (select CustomerID from Customer where CustName = 'Silvia Green'), '02/01/2020', '02/04/2020', 3),

((select ComputerKitID from ComputerKit where KitTypeID = 2), (select CustomerID from Customer where CustName = 'Boris Smith'), '02/05/2020', '02/09/2020', 3)

go

--function to return previous orders for specific parts ordered by date

create function OrderReport

(

@Part varchar(20)

)

returns table

as

return

select ComputerKitParts.PartID, Orders.ComputerKitID, Orders.OrdersID, Orders.OrderDate, Parts.Part from ComputerKitParts

inner join Orders on ComputerKitParts.ComputerKitID = Orders.ComputerKitID

join Parts on Parts.PartID = ComputerKitParts.PartID

where Parts.Part = @Part

go

--procedure to subtract from inventory

create or alter procedure InventorySubtraction(@PartName varchar(20), @AmountOrdered int, @AmountOnHand int) as

begin

update Inventory set AmountOnHand = (@AmountOnHand - @AmountOrdered)

where AmountOnHand = @AmountOnHand and Part = @PartName

return scope\_identity()

end

go

--procedure to add to inventory

create or alter procedure InventoryAddition(@PartName varchar(20), @AmountOrdered int, @AmountOnHand int) as

begin

update Inventory set AmountOnHand = (@AmountOnHand + @AmountOrdered)

where AmountOnHand = @AmountOnHand and Part = @PartName

return scope\_identity()

end

go

--A view for Ralph to use to see his currently outstanding orders

create view OutstandingOrders as

(

select \* from Orders where DeliveryDate is null and TimeToDelivery is null

)

go

--Function to gather parts ordered by month

create or alter function MonthlyOrders

(

@Month datetime

)

returns table

as

return

select Orders.ComputerKitID, ComputerKitParts.PartID, Parts.Part, Orders.DeliveryDate,

ROW\_NUMBER() OVER (order by Part) AS RowNum

from Orders

join ComputerKitParts on Orders.ComputerKitID = ComputerKitParts.ComputerKitID

join Parts on Parts.PartID = ComputerKitParts.PartID

where DeliveryDate > @Month

go

--Subtracting first orders from Inventory

exec InventorySubtraction 'Case lvl 1', 3, 7

go

exec InventorySubtraction 'Case lvl 2', 2, 4

go

exec InventorySubtraction 'Case lvl 3', 1, 3

go

exec InventorySubtraction 'Cooling lvl 1', 1, 5

go

exec InventorySubtraction 'Cooling lvl 2', 1, 4

go

exec InventorySubtraction 'Cooling lvl 3', 1, 3

go

exec InventorySubtraction 'Graphics lvl 1', 1, 5

go

exec InventorySubtraction 'Graphics lvl 2', 1, 4

go

exec InventorySubtraction 'Graphics lvl 3', 1, 3

go

exec InventorySubtraction 'Motherboard lvl 1', 3, 6

go

exec InventorySubtraction 'Motherboard lvl 2', 2, 4

go

exec InventorySubtraction 'Motherboard lvl 3', 1, 3

go

exec InventorySubtraction 'Processor lvl 1', 3, 7

go

exec InventorySubtraction 'Processor lvl 2', 2, 5

go

exec InventorySubtraction 'Processor lvl 3', 1, 4

go

exec InventorySubtraction 'RAM lvl 1', 2, 8

go

exec InventorySubtraction 'RAM lvl 2', 2, 5

go

exec InventorySubtraction 'RAM lvl 3', 2, 4

go

exec InventorySubtraction 'Storage lvl 1', 2, 6

go

exec InventorySubtraction 'Storage lvl 2', 2, 5

go

exec InventorySubtraction 'Storage lvl 3', 2, 4

go

--Ralph has hit some re-order points, so he puts in a CosmoOrder

--and uses the InventoryAddition function

insert into CosmoOrders (DatePurchased, QuantityPurchased, PartID) values

('02/10/2020', 3, (select PartID from Parts where Part = 'Motherboard lvl 1')), ('02/10/2020', 4, (select PartID from Parts where Part = 'Motherboard lvl 2')),

('02/10/2020', 3, (select PartID from Parts where Part = 'Motherboard lvl 3')), ('02/10/2020', 4, (select PartID from Parts where Part = 'Case lvl 2')),

('02/10/2020', 2, (select PartID from Parts where Part = 'Case lvl 3')), ('02/10/2020', 3, (select PartID from Parts where Part = 'Processor lvl 1')),

('02/10/2020', 3, (select PartID from Parts where Part = 'Processor lvl 2')), ('02/10/2020', 2, (select PartID from Parts where Part = 'Processor lvl 3')),

('02/10/2020', 3, (select PartID from Parts where Part = 'RAM lvl 2')), ('02/10/2020', 4, (select PartID from Parts where Part = 'RAM lvl 3')),

('02/10/2020', 2, (select PartID from Parts where Part = 'Storage lvl 1')), ('02/10/2020', 3, (select PartID from Parts where Part = 'Storage lvl 2')),

('02/10/2020', 4, (select PartID from Parts where Part = 'Storage lvl 3')), ('02/10/2020', 2, (select PartID from Parts where Part = 'Graphics lvl 3')),

('02/10/2020', 2, (select PartID from Parts where Part = 'Cooling lvl 3'))

go

exec InventoryAddition 'Motherboard lvl 1', 3, 3

go

exec InventoryAddition 'Motherboard lvl 2', 4, 2

go

exec InventoryAddition 'Motherboard lvl 3', 3, 2

go

exec InventoryAddition 'Case lvl 2', 4, 2

go

exec InventoryAddition 'Case lvl 3', 2, 2

go

exec InventoryAddition 'Processor lvl 2', 3, 3

go

exec InventoryAddition 'Processor lvl 3', 2, 3

go

exec InventoryAddition 'RAM lvl 2', 3, 3

go

exec InventoryAddition 'RAM lvl 3', 4, 2

go

exec InventoryAddition 'Storage lvl 1', 2, 4

go

exec InventoryAddition 'Storage lvl 2', 3, 3

go

exec InventoryAddition 'Storage lvl 3', 4, 2

go

exec InventoryAddition 'Graphics lvl 3', 2, 2

go

exec InventoryAddition 'Cooling lvl 3', 2, 2

go

--A couple customers changed some info!

update Customer set Phone = 55565555, MailingAddress = '34567 John Drive'

where CustomerID = 1

update Customer set CustName = 'Silvia Johnson', MailingAddress = '34567 John Drive'

where CustomerID = 1

update Customer set PaymentMethod = 'Cash'

where CustomerID = 6

go

--inserting more orders

insert into Orders(ComputerKitID, CustomerID, OrderDate, DeliveryDate, TimeToDelivery) values

((select ComputerKitID from ComputerKit where KitTypeID = 1), (select CustomerID from Customer where CustName = 'Silvia Johnson'), '02/12/2020', '02/16/2020', 4),

((select ComputerKitID from ComputerKit where KitTypeID = 3), (select CustomerID from Customer where CustName = 'Boris Smith'), '02/12/2020', '02/16/2020', 4),

((select ComputerKitID from ComputerKit where KitTypeID = 1), (select CustomerID from Customer where CustName = 'Truit Russ'), '02/14/2020', '02/16/2020', 2),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (select CustomerID from Customer where CustName = 'Holly Flax'), '02/18/2020', '02/21/2020', 3),

((select ComputerKitID from ComputerKit where KitTypeID = 6), (select CustomerID from Customer where CustName = 'Yennifer North'), '02/19/2020', '02/24/2020', 5),

((select ComputerKitID from ComputerKit where KitTypeID = 4), (select CustomerID from Customer where CustName = 'Yancy Day'), '02/20/2020', '02/24/2020', 4)

go

--Subtracting second orders from Inventory

exec InventorySubtraction 'Case lvl 1', 3, 4

go

exec InventorySubtraction 'Case lvl 2', 2, 6

go

exec InventorySubtraction 'Case lvl 3', 1, 4

go

exec InventorySubtraction 'Cooling lvl 1', 1, 4

go

exec InventorySubtraction 'Cooling lvl 2', 1, 3

go

exec InventorySubtraction 'Cooling lvl 3', 1, 4

go

exec InventorySubtraction 'Graphics lvl 1', 1, 4

go

exec InventorySubtraction 'Graphics lvl 2', 1, 3

go

exec InventorySubtraction 'Graphics lvl 3', 1, 4

go

exec InventorySubtraction 'Motherboard lvl 1', 3, 6

go

exec InventorySubtraction 'Motherboard lvl 2', 2, 6

go

exec InventorySubtraction 'Motherboard lvl 3', 1, 5

go

exec InventorySubtraction 'Processor lvl 1', 3, 4

go

exec InventorySubtraction 'Processor lvl 2', 2, 6

go

exec InventorySubtraction 'Processor lvl 3', 1, 5

go

exec InventorySubtraction 'RAM lvl 1', 3, 6

go

exec InventorySubtraction 'RAM lvl 2', 1, 6

go

exec InventorySubtraction 'RAM lvl 3', 2, 6

go

exec InventorySubtraction 'Storage lvl 1', 3, 6

go

exec InventorySubtraction 'Storage lvl 2', 1, 6

go

exec InventorySubtraction 'Storage lvl 3', 2, 4

go

--2nd CosmoOrder of the New Year

insert into CosmoOrders (DatePurchased, QuantityPurchased, PartID) values

('02/25/2020', 6, (select PartID from Parts where Part = 'Motherboard lvl 1')), ('02/25/2020', 6, (select PartID from Parts where Part = 'Case lvl 1')),

('02/25/2020', 3, (select PartID from Parts where Part = 'Case lvl 3')), ('02/25/2020', 6, (select PartID from Parts where Part = 'Processor lvl 1')),

('02/25/2020', 2, (select PartID from Parts where Part = 'Processor lvl 2')), ('02/25/2020', 2, (select PartID from Parts where Part = 'Processor lvl 3')),

('02/25/2020', 6, (select PartID from Parts where Part = 'RAM lvl 1')), ('02/25/2020', 3, (select PartID from Parts where Part = 'RAM lvl 3')),

('02/25/2020', 4, (select PartID from Parts where Part = 'Storage lvl 1')), ('02/25/2020', 2, (select PartID from Parts where Part = 'Graphics lvl 2')),

('02/25/2020', 2, (select PartID from Parts where Part = 'Cooling lvl 2'))

go

--Adding new parts into inventory

exec InventoryAddition 'Motherboard lvl 1', 6, 3

go

exec InventoryAddition 'Case lvl 1', 6, 1

go

exec InventoryAddition 'Case lvl 3', 3, 3

go

exec InventoryAddition 'Processor lvl 1', 6, 1

go

exec InventoryAddition 'Processor lvl 2', 2, 4

go

exec InventoryAddition 'Processor lvl 3', 2, 4

go

exec InventoryAddition 'RAM lvl 1', 6, 3

go

exec InventoryAddition 'RAM lvl 3', 3, 4

go

exec InventoryAddition 'Storage lvl 1', 4, 3

go

exec InventoryAddition 'Storage lvl 2', 4, 3

go

exec InventoryAddition 'Graphics lvl 2', 2, 2

go

exec InventoryAddition 'Cooling lvl 2', 2, 2

go

--March Orders!

insert into Orders(ComputerKitID, CustomerID, OrderDate, DeliveryDate, TimeToDelivery) values

((select ComputerKitID from ComputerKit where KitTypeID = 2), (Select CustomerID from Customer where CustName = 'Lenny Boy'), '03/01/2020', '03/04/2020', 3),

((select ComputerKitID from ComputerKit where KitTypeID = 3), (Select CustomerID from Customer where CustName = 'Victoria Smek'), '03/03/2020', '03/06/2020', 3),

((select ComputerKitID from ComputerKit where KitTypeID = 1), (Select CustomerID from Customer where CustName = 'Truit Russ'), '03/10/2020', '03/12/2020', 2),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (Select CustomerID from Customer where CustName = 'Holly Flax'), '03/15/2020', '03/18/2020', 3),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (Select CustomerID from Customer where CustName = 'Yennifer North'), '03/17/2020', '03/21/2020', 4),

((select ComputerKitID from ComputerKit where KitTypeID = 4), (Select CustomerID from Customer where CustName = 'Steve Bruce'), '03/21/2020', '03/24/2020', 3)

go

--Final inventory subtraction

exec InventorySubtraction 'Case lvl 1', 3, 7

go

exec InventorySubtraction 'Case lvl 2', 3, 4

go

exec InventorySubtraction 'Case lvl 3', 1, 4

go

exec InventorySubtraction 'Cooling lvl 1', 1, 3

go

exec InventorySubtraction 'Cooling lvl 2', 2, 4

go

exec InventorySubtraction 'Graphics lvl 1', 1, 3

go

exec InventorySubtraction 'Graphics lvl 2', 2, 4

go

exec InventorySubtraction 'Motherboard lvl 1', 3, 9

go

exec InventorySubtraction 'Motherboard lvl 2', 3, 4

go

exec InventorySubtraction 'Processor lvl 1', 3, 7

go

exec InventorySubtraction 'Processor lvl 2', 3, 6

go

exec InventorySubtraction 'RAM lvl 1', 2, 9

go

exec InventorySubtraction 'RAM lvl 2', 3, 5

go

exec InventorySubtraction 'RAM lvl 3', 1, 7

go

exec InventorySubtraction 'Storage lvl 1', 2, 7

go

exec InventorySubtraction 'Storage lvl 3', 1, 6

go

--Final CosmoOrder

insert into CosmoOrders (DatePurchased, QuantityPurchased, PartID) values

('04/01/2020', 5, (select PartID from Parts where Part = 'Motherboard lvl 2' )), ('04/01/2020', 4, (select PartID from Parts where Part = 'Case lvl 2' )),

('04/01/2020', 3, (select PartID from Parts where Part = 'Processor lvl 1' )),('04/01/2020', 3, (select PartID from Parts where Part = 'Processor lvl 2' )),

('04/01/2020', 4, (select PartID from Parts where Part = 'RAM lvl 2' )), ('04/01/2020', 2, (select PartID from Parts where Part = 'Graphics lvl 1' )),

('04/01/2020', 2, (select PartID from Parts where Part = 'Graphics lvl 2' )), ('04/01/2020', 2, (select PartID from Parts where Part = 'Cooling lvl 1' )),

('04/01/2020', 2, (select PartID from Parts where Part = 'Cooling lvl 2' ))

go

--Final Inventory Addition

exec InventoryAddition 'Motherboard lvl 2', 5, 1

go

exec InventoryAddition 'Case lvl 2', 4, 1

go

exec InventoryAddition 'Processor lvl 1', 3, 4

go

exec InventoryAddition 'Processor lvl 2', 3, 3

go

exec InventoryAddition 'RAM lvl 2', 4, 2

go

exec InventoryAddition 'Graphics lvl 1', 2, 2

go

exec InventoryAddition 'Graphics lvl 2', 2, 2

go

exec InventoryAddition 'Cooling lvl 1', 2, 2

go

exec InventoryAddition 'Cooling lvl 2', 2, 2

go

--Ralph's most popular order

select top 3

Count(Orders.ComputerKitID) as TotalKitsOrdered, KitType.KitType

from Orders

join ComputerKit on Orders.ComputerKitID = ComputerKit.ComputerKitID

join KitType on KitType.KitTypeID = ComputerKit.KitTypeID

group by KitType.KitType

order by TotalKitsOrdered desc

--Ralph's Least Popular order

select top 3

Count(Orders.ComputerKitID) as TotalKitsOrdered, KitType.KitType

from Orders

join ComputerKit on Orders.ComputerKitID = ComputerKit.ComputerKitID

join KitType on KitType.KitTypeID = ComputerKit.KitTypeID

group by KitType.KitType

order by TotalKitsOrdered

--Percentage breakdown of each product purchased from Ralph

select

(Count(Orders.ComputerKitID)\*100 / (select count (\*) from Orders)) as PercentageTotal, KitType.KitType

from Orders

join ComputerKit on Orders.ComputerKitID = ComputerKit.ComputerKitID

join KitType on KitType.KitTypeID = ComputerKit.KitTypeID

group by KitType.KitType

order by PercentageTotal desc

--Parts re-ordred the least

select top 5

sum(CosmoOrders.QuantityPurchased) as TotalReOrdered, Parts.Part

from CosmoOrders

join Parts on CosmoOrders.PartID = Parts.PartID

group by Parts.Part

order by TotalReOrdered

--Part re-ordered the most

select top 5

sum(CosmoOrders.QuantityPurchased) as TotalReOrdered, Parts.Part

from CosmoOrders

join Parts on CosmoOrders.PartID = Parts.PartID

group by Parts.Part

order by TotalReOrdered desc

--How many orders can Ralph handle with his current re-order points?

/\*to do this, I had to subtract down the current inventory

counts to where his reorder points where to provide a more clear

select result at the end of the orders. Without a more advanced

inventory selection schema, I am left with counting the orders

manually that would bring all parts down to 0, and inputting those

InventorySubtracts so visualize this for answering the project question.

\*/

exec InventorySubtraction 'Motherboard lvl 1', 3, 6

go

exec InventorySubtraction 'Motherboard lvl 2', 3, 6

go

exec InventorySubtraction 'Motherboard lvl 3', 1, 4

go

exec InventorySubtraction 'Case lvl 1', 1, 4

go

exec InventorySubtraction 'Case lvl 2', 2, 5

go

exec InventorySubtraction 'Case lvl 3', 3, 6

go

exec InventorySubtraction 'Processor lvl 1', 3, 7

go

exec InventorySubtraction 'Processor lvl 2', 2, 6

go

exec InventorySubtraction 'Processor lvl 3', 2, 6

go

exec InventorySubtraction 'RAM lvl 1', 3, 7

go

exec InventorySubtraction 'RAM lvl 2', 2, 6

go

exec InventorySubtraction 'RAM lvl 3', 2, 6

go

exec InventorySubtraction 'Storage lvl 1', 1, 5

go

exec InventorySubtraction 'Storage lvl 2', 1, 5

go

exec InventorySubtraction 'Storage lvl 3', 1, 5

go

exec InventorySubtraction 'Graphics lvl 1', 2, 4

go

exec InventorySubtraction 'Graphics lvl 2', 2, 4

go

exec InventorySubtraction 'Graphics lvl 3', 1, 3

go

exec InventorySubtraction 'Cooling lvl 1', 2, 4

go

exec InventorySubtraction 'Cooling lvl 2', 2, 4

go

exec InventorySubtraction 'Cooling lvl 3', 1, 3

go

/\*Now for the orders! Using just gaming PC's we can

knock out a lot of the parts on hand. \*/

insert into Orders (ComputerKitID, CustomerID, OrderDate, DeliveryDate, TimeToDelivery)

values

((select ComputerKitID from ComputerKit where KitTypeID = 2), (select CustomerID from Customer where CustName = 'Silvia Johnson'), '06/01/2020', '06/01/2020', 1),

((select ComputerKitID from ComputerKit where KitTypeID = 3), (select CustomerID from Customer where CustName = 'Silvia Johnson'), '06/01/2020', '06/01/2020', 1),

((select ComputerKitID from ComputerKit where KitTypeID = 4), (select CustomerID from Customer where CustName = 'Silvia Johnson'), '06/01/2020', '06/02/2020', 1),

((select ComputerKitID from ComputerKit where KitTypeID = 4), (select CustomerID from Customer where CustName = 'Silvia Johnson'), '06/01/2020', '06/02/2020', 1),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (select CustomerID from Customer where CustName = 'Silvia Johnson'), '06/01/2020', '06/02/2020', 1),

((select ComputerKitID from ComputerKit where KitTypeID = 5), (select CustomerID from Customer where CustName = 'Silvia Johnson'), '06/01/2020', '06/02/2020', 1),

((select ComputerKitID from ComputerKit where KitTypeID = 6), (select CustomerID from Customer where CustName = 'Silvia Johnson'), '06/01/2020', '06/02/2020', 1),

((select ComputerKitID from ComputerKit where KitTypeID = 6), (select CustomerID from Customer where CustName = 'Silvia Johnson'), '06/01/2020', '06/02/2020', 1)

--The above translates into the following inventory subtractions

exec InventorySubtraction 'Motherboard lvl 1', 3, 3

go

exec InventorySubtraction 'Motherboard lvl 2', 3, 3

go

exec InventorySubtraction 'Motherboard lvl 3', 2, 3

go

exec InventorySubtraction 'Case lvl 1', 3, 3

go

exec InventorySubtraction 'Case lvl 2', 3, 3

go

exec InventorySubtraction 'Case lvl 3', 2, 3

go

exec InventorySubtraction 'Processor lvl 1', 3, 4

go

exec InventorySubtraction 'Processor lvl 2', 2, 4

go

exec InventorySubtraction 'Processor lvl 3', 3, 4

go

exec InventorySubtraction 'RAM lvl 1', 3, 4

go

exec InventorySubtraction 'RAM lvl 2', 3, 4

go

exec InventorySubtraction 'RAM lvl 3', 2, 4

go

exec InventorySubtraction 'Storage lvl 1', 3, 4

go

exec InventorySubtraction 'Storage lvl 2', 2, 4

go

exec InventorySubtraction 'Storage lvl 3', 3, 4

go

exec InventorySubtraction 'Graphics lvl 1', 2, 2

go

exec InventorySubtraction 'Graphics lvl 2', 2, 2

go

exec InventorySubtraction 'Graphics lvl 3', 1, 2

go

exec InventorySubtraction 'Cooling lvl 1', 2, 2

go

exec InventorySubtraction 'Cooling lvl 2', 2, 2

go

exec InventorySubtraction 'Cooling lvl 3', 1, 2

go

select \* from Inventory

/\*Based on this table, all it took was 8 orders to

use enough parts to clear out Ralph's options.

Ralph's current offerings leave him

unable to sell one of his 6 products. There are enough

parts to create a higher tier office PC out of the remainder,

however, that would involve him updating his offerings. Something

he may not have learned without using visual data!

\*/

--Are there any products that Ralph should only buy parts for at a per order basis?

select sum(CosmoOrders.QuantityPurchased) as QuantityOfPartOrdered, Parts.Part

from CosmoOrders

join Parts on Parts.PartID = CosmoOrders.PartID

group by Part

Order by QuantityOfPartOrdered