1. **Equipment Insight**

* **Quantity currently in store and quantity currently rented out of store**

We’ve decided to answer both in one query

Below we’ve found that ***Store 1,*** they have **38 total items** in their inventory that they rent out. **Five of those items** are currently being rented out (with the rental end date being March 18), with **33 of those items** which are still in store.

We’ve also decided to retrieve data regarding one specific table saw based on its serial number. We found that the specific table saw is **currently being rented out from Store 1**. We’ve also found that Store 1 carries three of the same table saws, **and that one is currently being rented out.** **This means two table saws remain in store.**

Query:

select\*from ProductInventory where StoreID='1'

/\* number of products the store has, in stock or rented \*/

declare @NUM int

select @NUM = count(ProductSerialNumber) from ProductInventory where StoreID= '1'

select @NUM

/\* March 18, 2021 will be the end date to show active orders \*/

Select ProductInventory.ProductSerialNumber, Order\_ProductRel.Order\_ProductRelID, Orders.OrderID

from ProductInventory

full outer join Order\_ProductRel on ProductInventory.ProductSerialNumber = Order\_ProductRel.ProductSerialNumber

full join Orders on Order\_ProductRel.OrderID = Orders.OrderID

where ProductInventory.StoreID = '1' and Orders.EndDate > '2021-03-18';

/\* to determine how mant items are currently being rented \*/

declare @renting int

select @renting = count(ProductSerialNumber) from Order\_ProductRel where Order\_ProductRelID = '7376' or Order\_ProductRelID ='7393' or Order\_ProductRelID ='7400' or Order\_ProductRelID ='7410' or Order\_ProductRelID ='7420'

select @renting

/\*To determine total items in store in store \*/

select (@NUM - @renting)

/\* Now we repeat a similar process, but for one single item: A Table saw from StoreID=1 \*/

Select ProductInventory.ProductID, Product.Name

from ProductInventory

inner join Product on ProductInventory.ProductID = Product.ProductID

where ProductInventory.ProductSerialNumber = '229QE7JL' and ProductInventory.StoreID = '1';

select count(ProductSerialNumber) from ProductInventory where StoreID='1' and ProductSerialNumber='229QE7JL';

Select ProductInventory.ProductSerialNumber, Order\_ProductRel.Order\_ProductRelID, Orders.OrderID

from ProductInventory

full outer join Order\_ProductRel on ProductInventory.ProductSerialNumber = Order\_ProductRel.ProductSerialNumber

full join Orders on Order\_ProductRel.OrderID = Orders.OrderID

where ProductInventory.StoreID = '1' and Orders.EndDate > '2021-03-18' and ProductInventory.ProductSerialNumber='229QE7JL';

/\* As you can see, that specific Table Saw from StoreID=1 is currently being rented out \*/

/\* Now lets how many Tables saws the StoreID=1 has, and lets see how many are being rented out currently \*/

declare @tb int;

select @tb = count(ProductSerialNumber) from ProductInventory where ProductID='A225489' and StoreID = '1';

select @tb;

/\* there are 3 table saws belonging to this store \*/

Select Orders.OrderID, Order\_ProductRel.ProductSerialNumber, ProductInventory.ProductID

from Orders

full outer join Order\_ProductRel on Order\_ProductRel.OrderId = Orders.OrderID

full join ProductInventory on Order\_ProductRel.ProductSerialNumber = ProductInventory.ProductSerialNumber

where ProductInventory.StoreID ='1' and ProductID='A225489' and Orders.EndDate > '2021-03-18';

declare @out int

select @out = count(ProductSerialNumber) from ProductInventory where ProductSerialNumber= '229QE7JL'

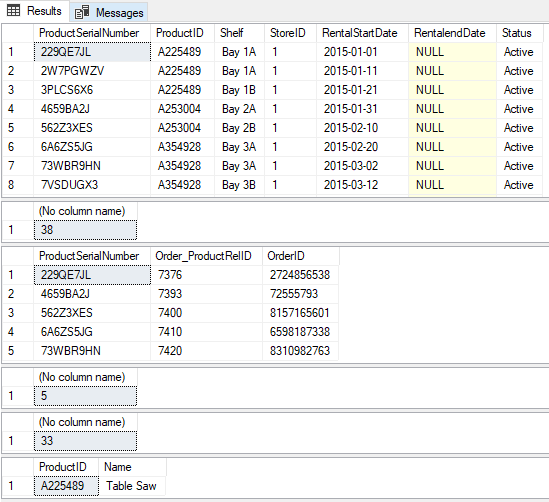
select @out

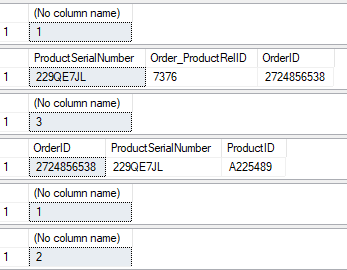
/\* 1 table saw is currently being rented out \*/

select (@tb - @out)

/\* 2 table saws are currently in store, and 1 is being rented out \*/

Results:





* **The quantity of table saws returning to StoreID=1 is just one**; it will be returning **April 12th, 2021.**

Query:

Select Orders.EndDate, Order\_ProductRel.ProductSerialNumber, ProductInventory.StoreID

from Orders

inner join Order\_ProductRel on Orders.OrderID = Order\_ProductRel.OrderID

inner join ProductInventory on Order\_ProductRel.ProductSerialNumber = ProductInventory.ProductSerialNumber

where ProductInventory.StoreID ='1' and Orders.EndDate > '2021-03-18' and ProductInventory.ProductID='A225489'

/\* As you can see from the join results, there is one Table Saw being rented out, and it will be returning to StoreID=1 on April 12, 2021 \*/

Results:



* **The average days in rental for defective rentals is three**. We found three total table saws that were defective and were returned. Two table saws were on the same order and were returned on the same day for both being defective. They were retrieved after 5 days. The third table saw was returned after 1 day

Query:

/\* average lifetime queries will be done for Table saws that are not from StoreID=1 because none from that store are defective,

so we will be choosing another table saw \*/

select ProductInventory.ProductSerialNumber,Order\_ProductRel.Order\_ProductRelID, Defects.RetrivalID

from ProductInventory

inner join Order\_ProductRel on ProductInventory.ProductSerialNumber = Order\_ProductRel.ProductSerialNumber

inner join Defects on Order\_ProductRel.Order\_ProductRelID = Defects.Order\_ProductRelID

where ProductInventory.ProductID = 'A225489' and ProductInventory.StoreID = '1'

select ProductInventory.ProductSerialNumber,Order\_ProductRel.Order\_ProductRelID, Defects.RetrivalID

from ProductInventory

inner join Order\_ProductRel on ProductInventory.ProductSerialNumber = Order\_ProductRel.ProductSerialNumber

inner join Defects on Order\_ProductRel.Order\_ProductRelID = Defects.Order\_ProductRelID

where ProductInventory.ProductID = 'A225489'

/\*there have been three defective table saws \*/

Select Product.Name, ProductInventory.ProductSerialNumber, ProductInventory.StoreID

from Product

inner join ProductInventory on Product.ProductID = ProductInventory.ProductID

where ProductInventory.ProductSerialNumber = '263Q9BK3'

/\*the table saw we will use now is from StoreID=4 \*/

/\*Now looking for total days rented before being returned for defect \*/

declare @date date

Select @date = DateRetrived from Retrieval where RetrivalID ='9271257677'

Select @date

Select Orders.StartDate, Order\_ProductRel.ProductSerialNumber, Orders.OrderID

from Orders

Inner join Order\_ProductRel on Orders.OrderID = Order\_ProductRel.OrderId

where Order\_ProductRel.ProductSerialNumber = '263Q9BK3'

declare @bdate date

select @bdate = StartDate from Orders where OrderId = '6581789623';

select @bdate

declare @x int

SELECT @x = DATEDIFF(day, @bdate, @date) ;

select @x

/\* 5 day fdifference difference for this table saw \*/

/\* Note that two different table saws, including the one queried above, were on the same order and were both defective, thus have the same retrivalID and the same 5 day difference.

Now lets see the average for all table saws that were returned (2 tables saws on the same order were both defective) \*/

Select Orders.StartDate, Orders.OrderID, Retrieval.RetrivalID

from Orders

Inner join Retrieval on Orders.OrderID = Retrieval.OrderID

where Retrieval.RetrivalID = '928064832'

declare @cdate date

select @cdate = StartDate from Orders where OrderID='4679537191'

declare @ddate date

select @ddate = DateRetrived from Retrieval where RetrivalID = '928064832'

select @ddate

declare @y int

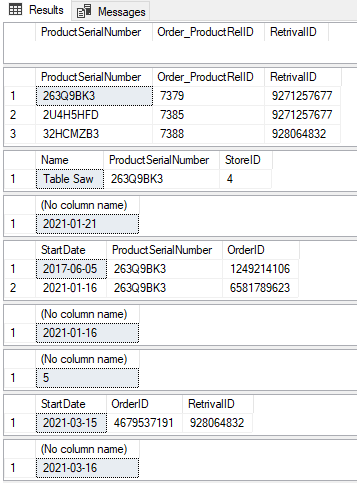
select @y = DATEDIFF(day, @cdate, @ddate)

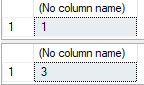
select @y

Select (@y + @x) /2

/\* Average days before being recalled as a defect is 3 days for the three table saws)

Results:





1. **Business Customer Insight Queries**

* Acceptance Rate for PM Jobi Bamlett is: 80%

Query:

/\* For this query, we will be picking one Project Manager (Jobi Bamlett) to see his acceptance rate \*/

select AccountID from Person where RoleID = 1; /\* this is to show all the project managers (PM is AccountID=1, and Support Staff is AccountID=2) \*/

Select AccountID from Person where FirstName = 'Jobi' and LastName = 'Bamlett';

Select Status from Orders where AccountID = '1159769643'

Select Status, AccountID from Orders where AccountID='1159769643';

Declare @Total decimal

Select @Total = count(AccountID) from Orders where AccountID='1159769643' ;

Select @Total

/\*PM was associated with a total of five order \*/

DECLARE @Approved decimal

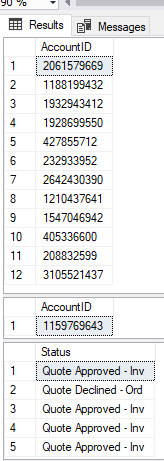
SELECT @Approved = count(Status) from Orders where AccountID='1159769643' and Status = 'Quote Approved - Inv';

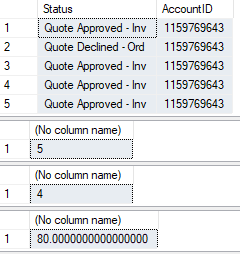
select @Approved

/\* four out of the five orders were approveed \*/

select (@Approved/@Total\*100);

Results:

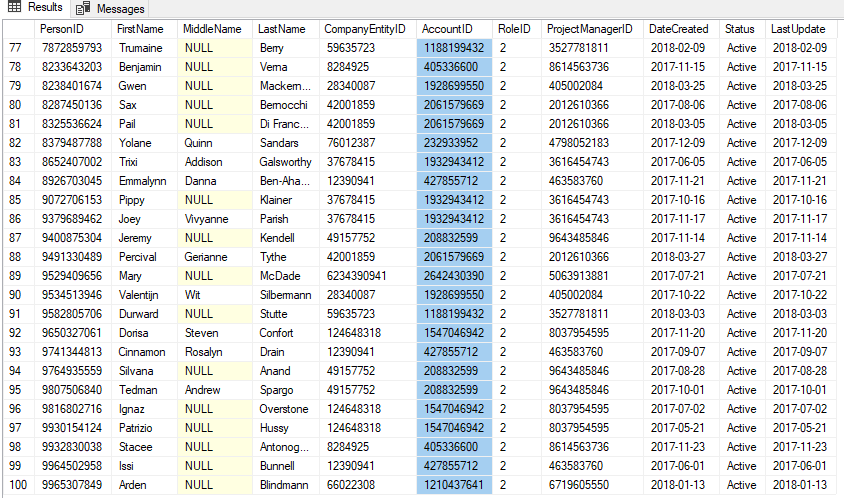


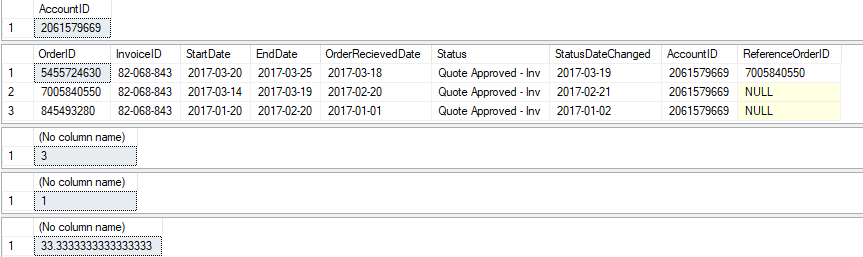


* The Support Staff **extension rate for Rossie Amy is 33.3%**

Query:

Results:





* Defective Return Rate for the orders belonging to **Customer Cybercash is 100%,** but the defective item rate for the same **customer is 20%**

Query:

SELECT distinct Account.AccountID, Person.CompanyEntityID, CompanyEntity.Name

FROM Account

INNER JOIN Person ON Person.AccountID = Account.AccountID

INNER JOIN CompanyEntity ON Person.CompanyEntityID = CompanyEntity.CompanyEntityID

where BusinessType = 'B2B' order by AccountID;

Select Account.AccountID, Orders.OrderID, Retrieval.RetrivalID

from Account

Inner join Orders on Orders.AccountID = Account.AccountID

Inner join Retrieval on Retrieval.OrderID = Orders.OrderID

where Orders.AccountID = '3105521437' and BusinessType = 'B2B' order by AccountID ;

/\* Looking at the joined tables, we will look at the defective orders of the company Cybercash (AccountID = '3105521437') \*/

/\* We selected all the orders from the Retreival table belonging to Cybercassh, and queried the retreival table with those orders to see if there were any defective orders \*/

Select\*from Retrieval where OrderID = '72555793' or OrderID = '8157165601' or OrderID = '8310982763' or OrderID = '7620342230' or OrderID = '6598187338' or OrderID = '4895202534';

declare @total decimal

select @total = count(RetrivalType) from Retrieval where OrderID = '72555793' or OrderID = '8157165601' or OrderID = '8310982763' or OrderID = '7620342230' or OrderID = '6598187338' or OrderID = '4895202534';

select @total

declare @defective decimal

select @defective = count(RetrivalID) from Retrieval where OrderID = '72555793' or OrderID = '8157165601' or OrderID = '8310982763' or OrderID = '7620342230' or OrderID = '6598187338' or OrderID = '4895202534' and RetrivalType='Bad Item Return';

select @defective

select(@defective/@total\*100)

/\* now to query the database to see the defective EQUIPMENT RATE \*/

select\*from Order\_ProductRel where OrderID = '72555793' or OrderID = '8157165601' or OrderID = '8310982763' or OrderID = '7620342230' or OrderID = '6598187338' or OrderID = '4895202534';

declare @totalItem decimal

select @totalItem = count(ProductSerialNumber) from Order\_ProductRel where OrderID = '72555793' or OrderID = '8157165601' or OrderID = '8310982763' or OrderID = '7620342230' or OrderID = '6598187338' or OrderID = '4895202534';

select @totalItem

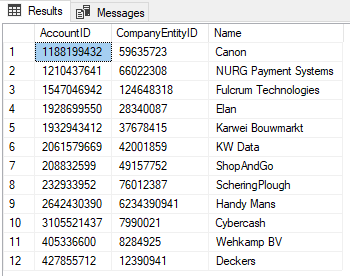
declare @defectiveItems decimal

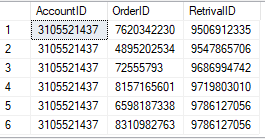
select @defectiveItems = count(Order\_ProductRelID) from Defects where Order\_ProductRelID ='7391' or Order\_ProductRelID = '7392' or Order\_ProductRelID ='7393' or Order\_ProductRelID = '7394' or Order\_ProductRelID ='7395' or Order\_ProductRelID = '7396' or Order\_ProductRelID ='7397' or Order\_ProductRelID = '7398' or Order\_ProductRelID ='7399' or Order\_ProductRelID = '7400' or Order\_ProductRelID = '7401' or Order\_ProductRelID = '7402' or Order\_ProductRelID = '7403' or Order\_ProductRelID = '7404' or Order\_ProductRelID ='7405' or Order\_ProductRelID ='7406' or Order\_ProductRelID ='7407' or Order\_ProductRelID = '7408' or Order\_ProductRelID = '7409' or Order\_ProductRelID = '7410' or Order\_ProductRelID='7411' or Order\_ProductRelID ='7412' or Order\_ProductRelID = '7413' or Order\_ProductRelID = '7414' or Order\_ProductRelID = '7415' or Order\_ProductRelID = '7416' or Order\_ProductRelID = '7417' or Order\_ProductRelID = '7418' or Order\_ProductRelID = '7419' or Order\_ProductRelID = '7420'

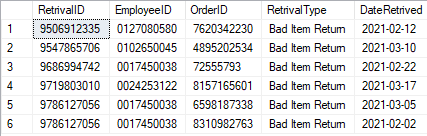
select @defectiveItems

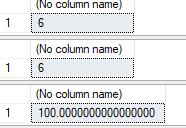
Select(@defectiveItems/@totalItem\*100)

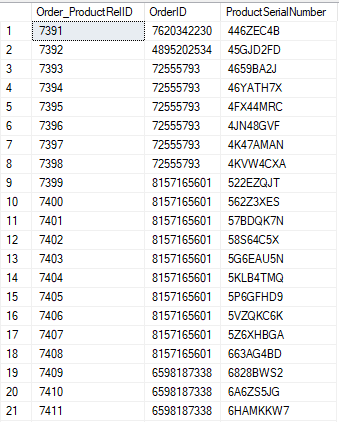
Results:

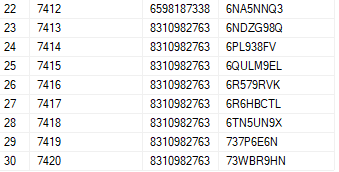


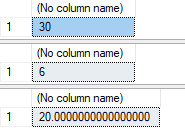












1. **Financial Insight**

* Total Invoices past due is  **$255,954**

/\*This first Query is here to determine the number of days that the contract is; this is important as we need to determine how many days we need to charge each item for\*/

SELECT DATEDIFF(DD, StartDate, EndDate) AS 'Contract Length in Days'

FROM "Order" where InvoiceID is not NULL

/\* Now that we have the totals of the invoices, we can determine which invoices still need to be paid and which ones have already been paid; When we determine the invoices that have been paid, we can determine to total amount\*/

We need to preform inner joins as the daily cost for each item rental is found in the Product table.

The follow displays the Total amount that the company has been paid\*/

Select sum(Product.RentalCost \* DATEDIFF(DD, StartDate, EndDate)) AS 'Invoice Total Paid in Dollars' from Order\_ProductRel

Inner join "Orders" on Order\_ProductRel.OrderID = "Orders".OrderID

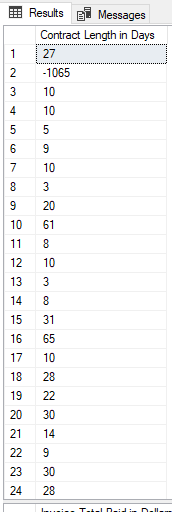
Inner join Payment on "Orders".InvoiceID = Payment.InvoiceID

Inner join ProductInventory on Order\_ProductRel.ProductSerialNumber = ProductInventory.ProductSerialNumber

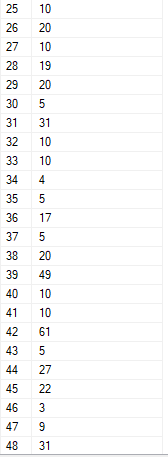
Inner join Product on ProductInventory.ProductID = Product.ProductID

where Payment.PaymentDate is not null

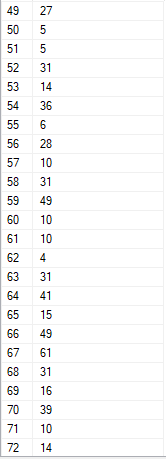
Results:

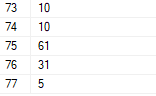


25-48 continued below



49-72 continued below







* **Total due today is $59642**

/\* There are some invoices however that have not been paid out yet. This query will display this amount that is still with standing/the company is entitled. \*/

Select sum(Product.RentalCost \* DATEDIFF(DD, StartDate, EndDate)) AS 'Invoice Total Pending in Dollars' from Order\_ProductRel

Inner join "Orders" on Order\_ProductRel.OrderID = "Orders".OrderID

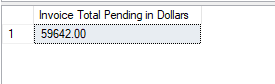
Inner join Payment on "Orders".InvoiceID = Payment.InvoiceID

Inner join ProductInventory on Order\_ProductRel.ProductSerialNumber = ProductInventory.ProductSerialNumber

Inner join Product on ProductInventory.ProductID = Product.ProductID

where Payment.PaymentDate is null

Results:



/\* We Can also look at the individual Invoice totals that have been paid out to the company; we can also check the amount that the company is waiting to receive b ychaning the ‘where Payment.PaymentDate is not null’ part of the code to is null \*/

Payment.InvoiceID, sum(Product.RentalCost \* DATEDIFF(DD, StartDate, EndDate)) AS 'Invoice Total Paid in Dollars' from Order\_ProductRel

Inner join "Orders" on Order\_ProductRel.OrderID = "Orders".OrderID

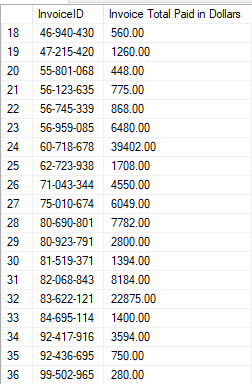
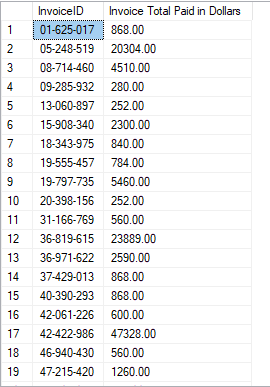
Inner join Payment on "Orders".InvoiceID = Payment.InvoiceID

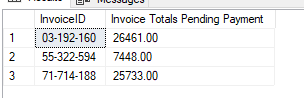
Inner join ProductInventory on Order\_ProductRel.ProductSerialNumber = ProductInventory.ProductSerialNumber

Inner join Product on ProductInventory.ProductID = Product.ProductID

where Payment.PaymentDate is not null

group by Payment.InvoiceID





1. **Staff Insight**

* The **wrong delvery rate for Drew Loenescu (shipper for ATSR) is 12.5%,** as 2 out of the 16 orders he delivered were wrong

Query:

Select\*from Retrieval where RetrivalType='Wrong Order'

Select\*from Shipment where OrderID='3661181459' or OrderID='9111771593'

Select FirstName, LastName from Employee where EmployeeID='0006426851'

/\* Now we know out of all the orders, only two were incorrect, and they were both incorrectly delivered by Drew Lonescu

Lets see how many orders he delviered in total, and find out the rate at which orders he delivers wrong\*/

Declare @total decimal

Select @total = count(ShipmentID) from Shipment where EmployeeID='0006426851'

Select @total

/\*note that the person who delivers the order does not have to be the person who retreives the order \*/

Declare @wrong decimal

select @wrong = count(RetrivalID) from Retrieval where RetrivalType='Wrong Order'

select @wrong

select (@wrong/@total \*100)

/\* Drew has a wrong delivery rate of 12.5% \*/

Results:

