William Alspaugh

Celine Del Mundo

Daniel Kissner

Dylan Rule

Module 12

**Case Study: Outland Adventures**

A diagram of a company

Description automatically generated

**Business Rules**

One employee can make many sales.

Many inventory items can be sold many times.

One Customer can make many purchases.

One customer can go on many trips.

Many Trips can be taken to one location.

One employee plans many trips.

Employees

EmployeeID – Primary - INT

FirstName – VARCHAR

LastName – VARCHAR

JobTitle – VARCHAR

Customers

CustomerID – Primary -INT

FirstName – VARCHAR

LastName – VARCHAR

Innoculations – BOOL

TravelVisas – BOOL

Inventory

ItemID – Primary – INT

Description – VARCHAR

UnitPrice – DECIMAL

RentalPrice – DECIMAL

Quantity – INT

IntakeDate – DATE

EquipmentSales

TransactionID – Primary

Category – (“Sale”, “Rental”) – Primary – VARCHAR

ItemID – Foreign/Primary – INT

Quantity – INT

Price – DECIMAL

EmployeeID – Foreign – INT

CustomerID – Foreign – INT

TransactionDate – DATE

Locations

LocationID – Primary – INT

LocationName – VARCHAR

Trip

TripID – Primary

CustomerID – Primary/Foreign – INT

EmployeedID – Foreign – INT

LocationID – Foreign – INT

Price – DECIMAL

TripDate – DATE

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

**Do enough customers buy equipment to keep equipment sales?A screenshot of a computer screen

Description automatically generated**

To answer this question, we wanted to compare Sales data against Rental Data for each item available in inventory for any specific year. We displayed the item ID and description, the Year, Sales quantity and revenue, and finally Rental quantity and revenue. From here, the business may decide if they would like to keep sales going.

**Query:**

SELECT

i.ItemID,

i.Description,

COALESCE(YEAR(s.TransactionDate), {year}) AS SalesYear,

SUM(CASE WHEN s.Category = 'sale' THEN s.Quantity ELSE 0 END) AS TotalSalesQuantity,

SUM(CASE WHEN s.Category = 'sale' THEN s.Price ELSE 0 END) AS TotalSalesAmount,

SUM(CASE WHEN s.Category = 'rental' THEN s.Quantity ELSE 0 END) AS TotalRentalsQuantity,

SUM(CASE WHEN s.Category = 'rental' THEN s.Price ELSE 0 END) AS TotalRentalsAmount

FROM Inventory i

LEFT JOIN EquipmentSales s

ON i.ItemID = s.ItemID AND YEAR(s.TransactionDate) = {year}

GROUP BY i.ItemID, i.Description, SalesYear;

**Is there any one of those locations that has a downward trend in bookings?**

**A screen shot of a black screen

Description automatically generated**

To answer this question, we wanted to compare the data for trips that were taken to each available location in a specific year against the previous year. We displayed the location ID and Location, the number of former year’s trips and trending year’s trips, and finally a confirmation of whether the year’s trend has risen, fell, or not changed from the previous year.

**Query:**

SELECT

l.LocationID,

l.Location,

COUNT(CASE WHEN YEAR(t.TripDate) = {year - 1} THEN t.TripID END),

COUNT(CASE WHEN YEAR(t.TripDate) = {year} THEN t.TripID END),

CASE

WHEN COUNT(CASE WHEN YEAR(t.TripDate) = {year} THEN t.TripID END) > COUNT(CASE WHEN YEAR(t.TripDate) = {year - 1} THEN t.TripID END) THEN 'Rise'

WHEN COUNT(CASE WHEN YEAR(t.TripDate) = {year} THEN t.TripID END) < COUNT(CASE WHEN YEAR(t.TripDate) = {year - 1} THEN t.TripID END) THEN 'Fall'

ELSE 'No Change'

END

FROM Locations l

LEFT JOIN Trips t

ON l.LocationID = t.LocationID

GROUP BY l.LocationID, l.Location;

**Are there inventory items that are over five years old?**

A black screen with white text

Description automatically generated

To Answer this question, we wanted to know how old an item in inventory is by using the Intake date of that item. We displayed the item ID, Description, and the age of the item. To get the age of the item, we took the difference between the current date and the intake date, then divided it by 365 days.

**Query:**

SELECT ItemID, Description, TRUNCATE(DATEDIFF(CURDATE(), IntakeDate)/365, 0)

FROM Inventory

WHERE DATEDIFF(CURDATE(), IntakeDate)/365 >= 5;