Eamonn’s Hardware Proposal

Cassidy Halpin

June 7th 2024

Eamonn’s hardware, a small local business, has recently seen a lot of growth in their business of renting and selling tools. Up till now all the store’s records have been kept through an excel spreadsheet. However, the store is now in need of a system that can grow with them and increase efficiency. After a meeting with the owner here are the main entities of the new database they would like to have in place.

Goals of Database

1. **Tool Inventory:** Tools are identified by a unique ID for the store and a number from the manufacturer.
2. **Tool Rentals:** Customers can rent up to 3 tools at a time.
3. **Tool Categories:** Tools need to be categorized by manufacturer, supplier, area of use, power system, price, rental fee, insurance, etc.
4. **Customer Management:** List and add customers, supporting business growth.
5. **Manufacturers and Suppliers:** Tools are sourced from specific manufacturers and suppliers.

Business Questions

1. Which tools are currently rented out?

2. How many tools are rented by each customer?

3. What is the total rental income generated per customer?

4. Which tools are rented most frequently?

5. What is the average rental fee for tools in each category?

6. Which customers have rented tools in the last month?

7. What are the total earnings from rentals for the last month?

8. Which suppliers provide the most rented tools?

9. What is the distribution of tools by power system?

10. What is the total number of tools rented by each category?

SQL code for Business Questions

-- 1. Which tools are currently rented out?

SELECT Tools.ToolID, Tools.manuID, Rental\_transaction.CustID, Rental\_transaction.RentalDate, Rental\_transaction.ReturnDate

FROM Tools

JOIN Rental\_details ON Tools.ToolID = Rental\_details.ToolID

JOIN Rental\_transaction ON Rental\_details.transactionID = Rental\_transaction.transactionID

WHERE Rental\_transaction.ReturnDate IS NULL;

-- 2. How many tools are rented by each customer?

SELECT Rental\_transaction.CustID, COUNT(Rental\_details.ToolID) AS ToolsRented

FROM Rental\_transaction

JOIN Rental\_details ON Rental\_transaction.transactionID = Rental\_details.transactionID

GROUP BY Rental\_transaction.CustID;

-- 3. What is the total rental income generated per customer?

SELECT Rental\_transaction.CustID, SUM(Tools.RentFee) AS TotalRentalIncome

FROM Rental\_transaction

JOIN Rental\_details ON Rental\_transaction.transactionID = Rental\_details.transactionID

JOIN Tools ON Rental\_details.ToolID = Tools.ToolID

GROUP BY Rental\_transaction.CustID;

-- 4. Which tools are rented most frequently?

SELECT Tools.ToolID, Tools.manuID, COUNT(Rental\_details.transactionID) AS RentalCount

FROM Tools

JOIN Rental\_details ON Tools.ToolID = Rental\_details.ToolID

GROUP BY Tools.ToolID, Tools.manuID

ORDER BY RentalCount DESC;

-- 5. What is the average rental fee for tools in each category?

SELECT Tools.type, AVG(Tools.RentFee) AS AverageRentalFee

FROM Tools

GROUP BY Tools.type;

-- 6. Which customers have rented tools in the last month?

SELECT DISTINCT Customers.CustID, Customers.first\_name, Customers.last\_name

FROM Customers

JOIN Rental\_transaction ON Customers.CustID = Rental\_transaction.CustID

WHERE Rental\_transaction.RentalDate >= NOW() - INTERVAL '1 month';

-- 7. What are the total earnings from rentals for the last month?

SELECT SUM(Tools.RentFee) AS TotalEarnings

FROM Rental\_transaction

JOIN Rental\_details ON Rental\_transaction.transactionID = Rental\_details.transactionID

JOIN Tools ON Rental\_details.ToolID = Tools.ToolID

WHERE Rental\_transaction.RentalDate >= NOW() - INTERVAL '1 month';

-- 8. Which suppliers provide the most rented tools?

SELECT Supplier.Name, COUNT(Rental\_details.ToolID) AS ToolsRented

FROM Supplier

JOIN Tools ON Supplier.SupplierID = Tools.SupplierID

JOIN Rental\_details ON Tools.ToolID = Rental\_details.ToolID

GROUP BY Supplier.Name

ORDER BY ToolsRented DESC;

-- 9. What is the distribution of tools by power system?

SELECT Power, COUNT(\*) AS ToolCount

FROM Tools

GROUP BY Power;

-- 10. What is the total number of tools rented by each category?

SELECT Tools.type, COUNT(Rental\_details.ToolID) AS TotalRentedTools

FROM Tools

JOIN Rental\_details ON Tools.ToolID = Rental\_details.ToolID

GROUP BY Tools.type;

ERD

A diagram of a data flow

Description automatically generated

**Tables**

**Customers**:

**Attributes**: CustID, First\_Name, Last\_Name, Street, City, Zip, Phone

**Rationale**: This table stores information about the customers who rent or purchase tools. It is fundamental for tracking transactions and managing customer data, such as contact information and address.

**Tools**:

**Attributes**: ToolID, ManuID, Description, Power, Type, Price, RentFee, Insurance, SupplierID

**Rationale**: This table contains details about the tools available for rental or purchase. Each tool is described by its specifications, pricing, and rental details. Linking to manufacturers and suppliers provides additional context for each tool.

**Rental\_Transaction**:

**Attributes**: TransactionID, CustID, RentalDate, ReturnDate

**Rationale**: This table tracks each rental transaction, associating it with the customer who made the rental and the dates involved. It helps in managing the rental lifecycle from the initiation to the return of the tools.

**Rental\_Details**:

**Attributes**: TransactionID, ToolID

**Rationale**: This table represents the details of which tools are included in each rental transaction. It handles the many-to-many relationship between rentals and tools, as a single transaction can involve multiple tools, and a single tool can be part of multiple transactions over time.

**Supplier**:

**Attributes**: SupplierID, Name

**Rationale**: This table stores information about suppliers who provide the tools. Maintaining supplier data is essential for managing inventory, ordering new tools, and understanding tool provenance.

**Manufactures**:

**Attributes**: ManuID, Name

**Rationale**: This table contains information about the manufacturers of the tools. It's crucial for identifying the source of the tools and managing relationships with manufacturers, which can affect warranties and support.

**Purchases**:

**Attributes**: PurchaseID, SupplierID, ToolID, CustID, PurchaseDate, PurchaseTotal

**Rationale**: This table records purchase transactions, linking customers with the tools they buy and the suppliers who provided them. It’s vital for tracking sales and understanding customer buying behavior.

Cardinalities

**One to Many Relationships:**

**Customers** and **Rental\_Transaction**: A customer can have multiple rental transactions.

**Rental\_Transaction** and **Rental\_Details**: A rental transaction can include multiple tools.

**Tools** and **Rental\_Details**: A tool can be included in multiple rental details.

**Purchases** and **Customers**: A customer can have multiple purchases.

**Purchases** and **Supplier**: A supplier can be involved in multiple purchases.

**Purchases** and **Tools**: A tool can be part of multiple purchases.

**Many to One Relationships:**

**Tools** and **Supplier**: Many tools can be supplied by one supplier.

**Tools** and **Manufactures**: Many tools can be manufactured by one manufacturer.

SQL Code for Table Creation

CREATE TABLE Customers (

CustID INT ,

First\_Name VARCHAR (255),

Last\_Name VARCHAR (255),

Street VARCHAR (255),

City VARCHAR (255),

Zip VARCHAR (255),

Phone VARCHAR (255),

PRIMARY KEY (CustID)

);

CREATE TABLE Tools (

ToolID INT,

ManuID INT,

Description VARCHAR (255),

Power VARCHAR (255),

Type VARCHAR (255),

Price DECIMAL (10, 2),

RentFee DECIMAL (10, 2),

Insurance DECIMAL (10,2),

SupplierID INT,

PRIMARY KEY (ToolID)

);

CREATE TABLE Rental\_Transaction (

TransactionID INT,

CustID INT,

RentalDate DATE,

ReturnDate DATE,

PRIMARY KEY (TransactionID)

);

CREATE TABLE Rental\_Details (

TransactionID INT,

ToolID INT

);

CREATE TABLE Supplier (

SupplierID INT,

Name VARCHAR (255),

PRIMARY KEY (SupplierID)

);

CREATE TABLE Manufactures (

ManuID INT,

Name VARCHAR (255),

PRIMARY KEY (ManuID)

);

CREATE TABLE Purchases (

PurchaseID INT,

SupplierID INT,

ToolID INT,

CustID INT,

PurchaseDate DATE,

PurchaseTotal DECIMAL (10, 2),

PRIMARY KEY (PurchaseID)

);

To implement this database successfully staffing levels are essential. System administrators for maintaining the database, handling system performance. Additionally, Database administrators to ensure data integrity and security. Adequate staffing is necessary to quickly respond to any issue that may arise as well as ongoing maintenance of the system.

Training is needed for both IT staff and employees to understand the new system architecture and be able to increase efficiency and user acceptancy.

Ensuring data accuracy when migrating the system from the old structure to the new database is essential so a thorough plan must be created and thorough testing to ensure accuracy.

SQL Code for Database Seeding

INSERT INTO public.customers(

custid, first\_name, last\_name, street, city, zip, phone)

VALUES

(1, 'Brad', 'Smith', 'Nike rd', 'Dallas', '75225', '5551234567'),

(2, 'Joseph', 'Stevens', 'Cherrystone rd', 'Dallas', '75225', '5552345678'),

(3, 'Roger', 'Enright', 'Ridge rd', 'Dallas', '75229', '5553456789'),

(4, 'Jacob', 'Harper', 'Serenity ln', 'Dallas', '75225', '555456890'),

(5, 'Patrick', 'Myers', 'Oak dr', 'Dallas', '75225', '5555678901'),

(6, 'Jane', 'Fuller', 'Presley ln', 'Dallas', '75229', '5556789012'),

(7, 'Sam', 'Calvert', 'Mystic rd', 'Dallas', '75229', '5557890123'),

(8, 'John', 'Nelson', 'Orleans ln', 'Dallas', '75229', '5558901234'),

(9, 'Jerry', 'Jackson', 'Hampton dr', 'Dallas', '75229', '5559012345'),

(10, 'Andrew', 'Henson', 'Lance ct', 'Dallas', '75229', '5550123456'),

(11, 'Carmen', 'Yonn', 'Bluebird dr', 'Dallas', '75229', '5552234567'),

(12, 'Matthew', 'Whalen', 'Olympic way', 'Dallas', '75229', '5553345678'),

(13, 'Finn', 'Ruble', 'Websters st', 'Dallas', '75225', '5554456789'),

(14, 'Simon', 'Liska', 'Newcastle st', 'Dallas', '75229', '5555567890'),

(15, 'Paul', 'Young', 'Queensway', 'Dallas', '75229', '5556678901'),

(16, 'Frank', 'Cobar', 'Kensington rd', 'Dallas', '75229', '5557789012'),

(17, 'Anna', 'Mastin', 'Sheperds ln', 'Dallas', '75225', '5558890123'),

(18, 'Calvin', 'Gresham', 'Wimbledon rd', 'Dallas', '75229', '5559901234'),

(19, 'Ricky', 'Miner', 'Austin ct', 'Dallas', '75229', '5551122345'),

(20, 'James', 'Howie', 'Machester ct', 'Dallas', '75225', '5552213456')

;

INSERT INTO public.manufactures(

manuid, name)

VALUES

(1, 'Bosch'),

(2, 'Craftsman'),

(3, 'DeWalt'),

(4, 'Dremel'),

(5, 'Kobalt'),

(6, 'Makita'),

(7, 'Milwaukee'),

(8, 'PorterCable'),

(9, 'Rigid'),

(10, 'Ryobi'),

(11, 'Stanley'),

(12, 'Hilt'),

(13, 'Hitachi'),

(14, 'Husky'),

(15, 'Bostitch'),

(16, 'Jet'),

(17, 'Fein'),

(18, 'Wen'),

(19, 'Triton'),

(20, 'Metabo')

;

INSERT INTO public.purchases(

purchaseid, supplierid, toolid, custid, purchasedate, purchasetotal)

VALUES

(1, 1, 1, 1, '05-17-2024', '$125.45'),

(2, 2, 2, 2, '05-23-2024', '$154.56'),

(3, 3, 3, 3, '04-15-2024', '$132.33'),

(4, 4, 4, 4, '05-10-2024', '$122.34'),

(5, 5, 5, 5, '05-12-2024', '$175.35'),

(6, 6, 6, 6, '05-12-2024', '$123.45'),

(7, 7, 7, 7, '04-23-2024', '$176.54'),

(8, 8, 8, 8, '05-03-2024', '$134.45'),

(9, 9, 9, 9, '05-03-2024', '$167.98'),

(10, 10, 10, 10, '04-25-2024', '$163.23'),

(11, 11, 11, 11, '04-26-2024', '$154.32'),

(12, 12, 12, 12, '05-07-2024', '$121.34'),

(13, 13, 13, 13, '05-20-2024', '$141.56'),

(14, 14, 14, 14, '05-11-2024', '$198.76'),

(15, 15, 15, 15, '05-21-2024', '$187.55'),

(16, 16, 16, 16, '05-22-2024', '$178.67'),

(17, 17, 17, 17, '05-22-2024', '$143.56'),

(18, 18, 18, 18, '04-17-2024', '$122.43'),

(19, 19, 19, 19, '05-23-2024', '$177.89'),

(20, 20, 20, 20, '05-13-2024', '$193.76')

;

INSERT INTO public.rental\_transaction(

transactionid, custid, rentaldate, returndate)

VALUES

(1, 1, '03-04-2024', '04-01-2024'),

(2, 2, '03-12-2024', '03-28-2024'),

(3, 3, '04-20-2024', '05-23-2024'),

(4, 4, '04-03-2024', '04-28-2024'),

(5, 5, '04-10-2024', '04-30-2024'),

(6, 6, '04-15-2024', '05-03-2024'),

(7, 7, '04-22-2024', '05-27-2024'),

(8, 8, '04-23-2024', '05-26-2024'),

(9, 9, '04-29-2024', '05-16-2024'),

(10, 10, '04-29-2024', '05-20-2024'),

(11, 11, '04-30-2024', '05-15-2024'),

(12, 12, '05-03-2024', '05-23-2024'),

(13, 13, '05-04-2024', '05-25-2024'),

(14, 14, '05-04-2024', '05-30-2024'),

(15, 15, '05-07-2024', '05-27-2024'),

(16, 16, '05-08-2024', '05-16-2024'),

(17, 17, '05-08-2024', '05-11-2024'),

(18, 18, '05-10-2024', '05-14-2024'),

(19, 19, '05-16-2024', '05-30-2024'),

(20, 20, '05-18-2024', '05-24-2024')

;

INSERT INTO public.rental\_details(

transactionid, toolid)

VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(2, 10),

(4, 11),

(7, 12),

(11, 13),

(15, 14),

(17, 15),

(1, 16),

(3, 17),

(5, 18),

(6, 19),

(8, 20)

;

INSERT INTO public.supplier(

supplierid, name)

VALUES

(1, 'SupplierA'),

(2, 'SupplierB'),

(3, 'SupplierC'),

(4, 'SupplierA'),

(5, 'SupplierB'),

(6, 'SupplierC'),

(7, 'SupplierA'),

(8, 'SupplierB'),

(9, 'SupplierC'),

(10, 'SupplierA'),

(11, 'SupplierB'),

(12, 'SupplierC'),

(13, 'SupplierA'),

(14, 'SupplierB'),

(15, 'SupplierC'),

(16, 'SupplierA'),

(17, 'SupplierB'),

(18, 'SupplierC'),

(19, 'SupplierA'),

(20, 'SupplierB')

;

INSERT INTO public.tools(

toolid, manuid, description, power, type, price, rentfee, insurance, supplierid)

VALUES

(1, 1, 'hammer', 'manual', 'multipurpose', '$15', '$5', '$1', 1),

(2, 2, 'screwdriver set', 'manual', 'multipurpose', '$40', '$10', '$2', 2),

(3, 3, 'lawn mower', 'fuel', 'outdoor', '$175', '$25', '$5', 3),

(4, 4, 'wrench', 'manual', 'multipurpose', '$15', '$5', '$1', 4),

(5, 5, 'pliers', 'manual', 'multipurpose', '$15', '$5', '$1', 5),

(6, 6, 'utility knife', 'manual', 'multipurpose', '$20', '$5', '$1', 6),

(7, 7, 'level', 'manual', 'multipurpose', '$15', '$5', '$1', 7),

(8, 8, 'cordless drill', 'battery', 'multipurpose', '$60', '$10', '$2', 8),

(9, 9, 'circular saw', 'AC', 'multipurpose', '$100', '$25', '$5', 9),

(10, 10, 'jigsaw', 'AC', 'multipurpose', '$100', '$25', '$5', 10),

(11, 11, 'angle grinder', 'AC', 'multipurpose', '$100', '$25', '$5', 11),

(12, 12, 'chisel', 'manual', 'multipurpose', '$25', '$5', '$1', 12),

(13, 13, 'sander', 'battery', 'multipurpose', '$100', '$25', '$5', 13),

(14, 14, 'workbench', 'manual', 'multipurpose', '$100', '$25', '$5', 14),

(15, 15, 'clamp set', 'manual', 'multipurpose', '$40', '$10', '$2', 15),

(16, 16, 'stud finder', 'battery', 'indoor', '$20', '$5', '$1', 16),

(17, 17, 'miter saw', 'AC', 'multipurpose', '$200', '$35', '$5', 17),

(18, 18, 'socket set', 'manual', 'multipurpose', '$40', '$10', '$2', 18),

(19, 19, 'heat gun', 'battery', 'multipurpose', '$100', '$25', '$5', 19),

(20, 20, 'shop vacuum', 'AC', 'indoor', '$100', '$25', '$5', 20)

;

Lessons Learned

My first problem I ran into with my initial ERD was I only included one table for the Rental of tools. However, because of this setup it only allowed for one item per rental instead of many items. This is important as specified in the goals of the database that up to 3 tools can be rented at a time. I remedied this by creating a Rental Details table that could include many rentals of many different tools.

When I first began to run script for the proposed business questions I ran into errors since I could not use math statements on character varying attributes. Therefore, I had to go back and change all the price quantities to decimal values. I could not make the prices integer values as they needed to be able to represent as fractional amounts instead of whole number values. So, while ID numbers could remain integers, prices needed to be decimal values which I define as (10,2) which allows for 8 digits before and 2 digits after the decimal.

Example of error I received trying to calculate with character varying attributes

A screenshot of a computer

Description automatically generated

Overall, the project aided in understanding the process of creating a database beginning from client consultation to considering how the system will perform for common business tasks.

References

IBM. (2021). Data migration: Best practices, process & tools. Retrieved June 6, 2024, from https://www.ibm.com/cloud/learn/data-migration

Mannino, M.V. (2023) *Database design, Query, formulation, and administration using Oracle and PostgreSQL*. Los Angeles: SAGE.

Techopedia. (n.d.). System administrator. Retrieved June 6, 2024, from https://www.techopedia.com/definition/2983/system-administrator