**Login**

//read $Login, $Password, $UserType

//if $UserType is Customer

SELECT Email

FROM Customer

WHERE Email = $Login and Password = $Password

//if customer found, login successful

//if $UserType is Clerk

SELECT Login

FROM Clerk

WHERE Login = $Login and Password = $Password

//if clerk found, login successful

**Create Profile**

//if “Submit” is clicked, insert a new customer

//read $Email, $Password, $FirstName, $LastName, $HomePhoneAreaCode, $HomePhoneLocalNumber, $WorkPhoneAreaCode, $WorkPhoneLocalNumber, $Address

INSERT INTO Customer

VALUES ($Email, $Password, $FirstName, $LastName, $HomePhoneAreaCode, $HomePhoneLocalNumber, $WorkPhoneAreaCode, $WorkPhoneLocalNumber, $Address);

**View Profile**

//assume $Email of current user is managed by application

//display personal information

SELECT Email, FirstName, LastName, HomePhoneAreaCode, HomePhoneLocalNumber, WorkPhoneAreaCode, WorkPhoneLocalNumber, Address

FROM Customer

WHERE Email = $Email;

//display reservations

SELECT ReservationNumber, GROUP\_CONCAT(AbbrDescription SEPARATOR ', ') AS Tools, StartDate, EndDate, SUM(DailyRentalPrice\*DATEDIFF(EndDate, StartDate)) AS RentalPrice, SUM(Deposit) AS Deposit, P.FirstName AS PickupClerk, D.FirstName AS DropoffClerk

FROM Reservation NATURAL JOIN ReservationReservesTool NATURAL JOIN Tool, Clerk AS P, Clerk AS D

WHERE CustomerLogin = $Email and P.Login = PickupClerkLogin and D.Login = DropoffClerkLogin

GROUP BY ReservationNumber

ORDER BY StartDate DESC;

**Check Tool Availability**

//if user clicks “Submit”

//read $ToolType, $StartDate, $EndDate

//Assume $StartDate, $EndDate validity is checked by application

//display Tool Availability

SELECT ToolID, AbbrDescription, Deposit, DailyRentalPrice

FROM Tool

WHERE ToolType = $ToolType and SaleDate is NULL and ToolID NOT IN

   (SELECT ToolID

    FROM Reservation NATURAL JOIN ReservationReservesTool

    WHERE EndDate > $StartDate AND StartDate < $Enddate

   UNION

    SELECT ToolID

    FROM ServiceRequest

    Where EndDate > $StartDate AND StartDate < $EndDate);

//if user enters a ToolID and clicks “View Details”

//read $ToolID

//display Tool Details

SELECT ToolID, AbbrDescription, FullDescription, PurchasePrice, DailyRentalPrice, Deposit

FROM Tool

WHERE ToolID = $ToolID;

**Make a Reservation**

//populate Type of Tool dropdown

SELECT DISTINCT(ToolType)

FROM Tool;

//read $StartDate, $EndDate

//$StartDate, $EndDate validity is checked by application

//if a tool type is selected, populate Tool dropdown

//read $ToolType

SELECT ToolID, AbbrDescription, DailyRentalPrice

FROM Tool

WHERE ToolType = $ToolType and SaleDate is NULL and ToolID NOT IN

   (SELECT ToolID

    FROM Reservation NATURAL JOIN ReservationReservesTool

    WHERE EndDate > $StartDate AND StartDate < $EndDate

   UNION

    SELECT ToolID

    FROM ServiceRequest

    Where EndDate > $StartDate AND StartDate < $EndDate);

//if “Add More Tools” is clicked, display the form with room for a new tool

//if “Remove Last Tool” is clicked,  remove last tool

//if "Calculate Total" is clicked, display Reservation Summary

//read $ToolID

//display Tools Desired

for each $ToolID

SELECT ToolID, AbbrDescription

FROM Tool

WHERE ToolID = $ToolID;

end for

//display Total Rental Price and Total Deposit Required

SET @TotalRentalPrice:= 0.00;

SET @TotalDeposit:= 0.00;

for each $ToolID

SELECT @TotalRentalPrice:= @TotalRentalPrice + DailyRentalPrice\*DATEDIFF(EndDate, StartDate), @TotalDeposit:= @TotalDeposit + Deposit

FROM Tool

WHERE ToolID = $ToolID;

end for

//if 'Submit' is clicked, insert a new reservation

//assume $ReservationNumber and $CustomerLogin is managed by application

INSERT INTO Reservation

   VALUES($ReservationNumber, $StartDate, $EndDate, $CustomerLogin, NULL, NULL, NULL, NULL, NULL, NULL);

for each $ToolID

INSERT INTO ReservationReservesTool(ReservationNumber, ToolID)

   VALUES($ReservationNumber, $ToolID);

end for

//display Reservation Final

//display Tools Rented

SELECT AbbrDescription

FROM ReservationReservesTool Natural Join Tool

WHERE ReservationNumber = $ReservationNumber;

//display Reservation Details

SELECT ReservationNumber, StartDate, EndDate, SUM(DailyRentalPrice)\*DATEDIFF(EndDate, StartDate) + SUM(Deposit) AS TotalRentalPrice, SUM(Deposit) AS TotalDepositRequired

FROM Reservation Natural Join ReservationReservesTool Natural Join Tool

WHERE ReservationNumber = $ReservationNumber

GROUP BY ReservationNumber;

**Pick-up**

//read $ReservationNumber

//display summary of the reservation

SELECT ToolID, AbbrDescription

FROM ReservationReservesTool Natural Join Tool

WHERE ReservationNumber = $ReservationNumber;

SELECT ReservationNumber, SUM(Deposit) AS DepositRequired, SUM(DailyRentalPrice)\*DATEDIFF(EndDate, StartDate) + SUM(Deposit) AS EstimateCost

FROM Reservation NATURAL JOIN ReservationReservesTool NATURAL JOIN Tool

WHERE ReservationNumber = $ReservationNumber;

//if a tool ID is entered and 'View Details' is clicked

//read $ToolID

SELECT ToolID, AbbrDescription, FullDescription, PurchasePrice, DailyRentalPrice, Deposit

FROM Tool

WHERE ToolID = $ToolID;

//if 'Complete Pick-Up' is clicked

//read $CreditCardNumber, $CreditCardExpirationDate

//assume $PickupClerkLogin is managed by application

//update Reservation

UPDATE Reservation

SET PickupClerkLogin = $PickupClerkLogin, PickupDate = CURRDATE(), CreditCardNumber = $CreditCardNumber, CreditCardExpirationDate = $CreditCardExpirationDate

WHERE ReservationNumber = $ReservationNumber;

//display Rental Contract

//display Tools Rented

SELECT ToolID, AbbrDescription

FROM ReservationReservesTool Natural Join Tool

WHERE ReservationNumber = $ReservationNumber;

//display Clerk on duty

SELECT FirstName, LastName

FROM Clerk

WHERE Login = $PickupClerkLogin;

//display Customer Name

SELECT FirstName, LastName

FROM Customer AS C, Reservation AS R

WHERE C.Email = R.CustomerLogin and R.ReservationNumber = $ReservationNumber;

//display Reservation

SELECT ReservationNumber, CreditCardNumber, StartDate, EndDate, SUM(Deposit) AS DepositHeld, SUM(DailyRentalPrice)\*DATEDIFF(EndDate, StartDate) + SUM(Deposit) AS EstimateRental

FROM Reservation Natural Join ReservationReservesTool Natural Join Tool

WHERE ReservationNumber = $ReservationNumber

GROUP BY ReservationNumber;

**Drop-off**

//read $ReservationNumber

//display summary of the reservation

SELECT ToolID, AbbrDescription

FROM ReservationReservesTool Natural Join Tool

WHERE ReservationNumber = $ReservationNumber;

SELECT ReservationNumber, SUM(Deposit) AS DepositRequired, SUM(DailyRentalPrice)\*DATEDIFF(EndDate, StartDate) + SUM(Deposit) AS EstimateCost

FROM Reservation Natural Join ReservationReservesTool Natural Join Tool

WHERE ReservationNumber = $ReservationNumber

GROUP BY ReservationNumber;

//if a tool ID is entered and 'View Details' is clicked

//read $ToolID

SELECT ToolID, AbbrDescription, FullDescription, PurchasePrice, DailyRentalPrice, Deposit

FROM Tool

WHERE ToolID = $ToolID;

//if 'Complete Drop-off' is clicked

//assume $DropoffClerkLogin is managed by application

//update Reservation

UPDATE Reservation

SET DropoffClerkLogin = $DropoffClerkLogin, DropoffDate = CURRDATE()

WHERE ReservationNumber = $ReservationNumber;

//display Rental Receipt

//display Clerk on duty

SELECT FirstName, LastName

FROM Clerk

WHERE Login = $DropoffClerkLogin;

//display Customer Name

SELECT FirstName, LastName

FROM Customer AS C, Reservation AS R

WHERE C.Email = R.CustomerLogin and R.ReservationNumber = $ReservationNumber;

//display Reservation

SELECT ReservationNumber, CreditCardNumber, StartDate, EndDate, SUM(DailyRentalPrice)\*DATEDIFF(EndDate, StartDate) + SUM(Deposit) AS RentalPrice, -SUM(Deposit) AS DepositHeld, SUM(DailyRentalPrice)\*DATEDIFF(EndDate, StartDate) As Total

FROM Reservation Natural Join ReservationReservesTool Natural Join Tool

WHERE ReservationNumber = $ReservationNumber

GROUP BY ReservationNumber;

**Add Tool**

//if “Add New Tool” is clicked

//populate Tool Type dropdowns

SELECT  \*

FROM ToolType;

//if “Submit New Tool” is clicked, insert new tool

//assume $ToolID and $AddClerkLogin is managed by application

//read $ToolType, $Abbr.Description, $FullDescription, $PurchasePrice, $DailyRentalPrice, $Deposit

INSERT INTO Tool

   VALUES ($ToolID, $ToolType, $Abbr.Description, $FullDescription, $PurchasePrice, $DailyRentalPrice, $Deposit, , $AddClerkLogin, NULL, NULL, NULL);

//if the tool type is "Power Tools"

//read $Accessory

for each $Accessory

INSERT INTO PowerToolAccessories

   VALUES($ToolID, $Accessory);

end for

**Sell Tool**

//if "Sell Tool" is clicked and a ToolID is entered

//read $ToolID

//assume $SellClerkLogin is managed by application

//check if the tool is reserved, held or sold

SELECT ToolID

FROM Reservation NATURAL JOIN ReservationReservesTool

WHERE ToolID = $ToolID AND EndDate > CURRDATE()

UNION

SELECT ToolID

FROM ServiceRequest

WHERE ToolID = $ToolID AND EndDate > CURRDATE()

UNION

SELECT ToolID

FROM Tool

WHERE ToolID = $ToolID AND SaleDate IS NOT NULL;

//if return empty set, the tool is not reserved, held or sold

//if the tool is not reserved, held or sold, update tool

UPDATE Tool

SET SellClerkLogin = $SellClerkLogin, SaleDate = CURRDATE(), SalePrice = PurchasePrice/2

WHERE ToolID = $ToolID;

//Return the sale price of the tool

SELECT ToolID, AbbrDescription, SalePrice

FROM Tool

WHERE ToolID = $ToolID;

**Hold Tool for Repair**

//if “Submit” is clicked

//read $ToolID, $StartDate, $EndDate, $EstimateRepairCost

//$StartDate, $EndDate validity is checked by application

//assume $HoldClerkLogin is managed by application

//check if the tool is reserved, held or sold

SELECT ToolID

FROM Reservation NATURAL JOIN ReservationReservesTool

WHERE ToolID = $ToolID AND EndDate > $StartDate AND StartDate < $EndDate

UNION

SELECT ToolID

FROM ServiceRequest

WHERE ToolID = $ToolID AND EndDate > $StartDate AND StartDate < $EndDate

UNION

SELECT ToolID

FROM Tool

WHERE ToolID = $ToolID AND SaleDate IS NOT NULL;

//if return empty set, the tool is not reserved, held or sold

//if the tool is not reserved, held or sold, insert service request

INSERT INTO ServiceRequest

   VALUES ($HoldClerkLogin, $ToolID, $StartDate, $EndDate, $EstimateRepairCost);

**Generate Reports**

//Report 1

//assume $ReportDate is managed by application

SELECT T.ToolID, AbbrDescription, COALESCE(RentalProfit, 0) AS RentalProfit, COALESCE(RepairCost, 0) + PurchasePrice AS CostOfTool, COALESCE(RentalProfit, 0) - COALESCE(RepairCost, 0) - PurchasePrice As TotalProfit

FROM

Tool AS T

LEFT JOIN

   (SELECT ToolID, SUM(DailyRentalPrice \* DATEDIFF(EndDate, StartDate)) AS RentalProfit

    From Reservation NATURAL JOIN ReservationReservesTool NATURAL JOIN Tool

    WHERE EndDate <= $ReportDate

    GROUP BY ToolID) AS R

ON T.ToolID = R.ToolID

LEFT JOIN

   (SELECT ToolID, SUM(EstimateRepairCost) AS RepairCost

    From ServiceRequest

    WHERE EndDate <= $ReportDate

    GROUP BY ToolID) AS S

On T.ToolID = S.ToolID

ORDER BY TotalProfit DESC;

//Report 2

//number of rentals is the sum of days of each tool rented in last month

//assume $MonthStart and $MonthEnd is managed by application

SELECT CONCAT(FirstName,' ', LastName) AS Name, C.Email AS EmailAddress, SUM(Rentals) AS Rentals

FROM Customer AS C,

(SELECT CustomerLogin, DATEDIFF(EndDate, StartDate) AS Rentals

FROM Reservation NATURAL JOIN ReservationReservesTool

WHERE StartDate >= $MonthStart AND EndDate <= $MonthEnd

UNION ALL

SELECT CustomerLogin, DATEDIFF(EndDate, $MonthStart) AS Rentals

FROM Reservation NATURAL JOIN ReservationReservesTool

WHERE StartDate < $MonthStart AND EndDate <= $MonthEnd AND EndDate > $MonthStart

UNION ALL

SELECT CustomerLogin, DATEDIFF($MonthEnd, StartDate) AS Rentals

FROM Reservation NATURAL JOIN ReservationReservesTool

WHERE StartDate >= $MonthStart AND StartDate < $MonthEnd AND EndDate > $MonthEnd

UNION ALL

SELECT CustomerLogin, DATEDIFF($MonthEnd, $MonthStart) AS Rentals

FROM Reservation NATURAL JOIN ReservationReservesTool

WHERE StartDate < $MonthStart AND EndDate > $MonthEnd) AS R

WHERE C.Email = R.CustomerLogin

GROUP BY C.Email

ORDER BY Rentals DESC, C.LastName;

//Report 3

//assume $MonthStart and $MonthEnd is managed by application

SELECT CONCAT(FirstName,' ', LastName) AS Name, COALESCE(Pickups, 0) AS Pickups, COALESCE(Dropoffs, 0) AS Dropoffs, COALESCE(Pickups, 0) + COALESCE(Dropoffs, 0) AS Total

FROM Clerk AS C

LEFT JOIN

    (SELECT PickupClerkLogin AS Login, COUNT(PickupClerkLogin) AS Pickups

     FROM Reservation

     WHERE PickupDate >= $MonthStart AND PickupDate <= $MonthEnd

     GROUP BY PickupClerkLogin) AS P

ON C.Login = P.Login

LEFT JOIN

   (SELECT DropoffClerkLogin AS Login, COUNT(DropoffClerkLogin) AS Dropoffs

     FROM Reservation

     WHERE DropoffDate >= $MonthStart AND DropoffDate <= $MonthEnd

     GROUP BY DropoffClerkLogin) AS D

ON C.Login = D.Login

ORDER BY Total DESC;