

## Table of Contents:

<b>Table of Contents:</b>	<b>1</b>
<b>Abstract Code + SQL</b>	<b>2</b>
Login	2
Register	3
View Profile	4
Check Tool Availability	5
View Tool Details	7
View/ Add/ Calculate Tools for Rent	10
Check and Submit Reservation	12
View/ Add/ Calculate Tools for Purchase	13
Check and Submit Purchase Confirmation	14
View All Pick-UP Reservations	15
View/ Confirm Pick-Up Reservation	15
View/ Update Credit Card Info	16
View Rental Contract	17
View All Drop-Off Reservations	18
View/ Confirm Drop-Off Reservation Summary	18
View Final Receipt	19
Add Tool	20
Repair Tool	23
View/ Filter/ Override Current Status of All Tools in Repair	25
View/ Filter/ Sell Available Tools	25
View/ Filter Sale Status	26
View Clerk Report	27
View Customer Report	28
View/ Filter Tool Inventory Report	29

## Abstract Code + SQL

### Login

#### Abstract Code

- User enters *username* (**\$Username**) and *password* (**\$Password**) into input fields.
- User must select the buttons between "Customer" and "Clerk" to identify oneself.
- If **Customer** button is selected:
  - If data validation is successful for both *username* and *password* input fields, then:
    - When **Login** button is clicked - run **Login** task:

```
SELECT Password FROM Customer WHERE Customer.username = ` $Username `;
```
    - If User record is found but **Customer.password** != **`\$Password`**
      - Go back to **Login** form, with error message "Password is incorrect."
    - Else if User record is not found:
      - Go to **Registration**
    - Else:
      - Go to **Customer Main Menu**.
  - Else *username* and *password* input fields are invalid, display **Login** form, with error message.
- Else if **Clerk** button is selected:
  - If data validation is successful for both *username* and *password* input fields, then:
    - When **Login** button is clicked - run **Login** task:

```
SELECT Password FROM Clerk WHERE Clerk.username = ` $Username `;
```
    - If User record is found but **Clerk.password** != **`\$Password`**
      - Go back to **Login** form, with error message "Password is incorrect."
    - Else if User record is not found:
      - Display error message "Username is not registered as Clerk"
    - Else:
      - Go to **Clerk Main Menu**.
  - Else *username* and *password* input fields are invalid, display **Login** form, with error message.

[Table of Contents:](#)

## Register

### Abstract Code

- User enters *FirstName* (*\$FirstName*), *Middle Name* (*\$MiddleName*), *Last Name* (*\$LastName*), *Home Phone* (*\$HomePhoneAreaCode*, *\$HomePhoneMainNumber*, *\$HomePhoneExtension*), *Work Phone* (*\$WPAreaCode*, *\$WPMainNumber*, *\$WPExtension*), *Cell Phone* (*\$CPAreaCode*, *\$CPMainNumber*, *\$CPEExtension*), One of Home Phone, Work Phone, Cell Phone as Primary Phone (*\$PPAreaCode*, *\$PPMainNumber*, *\$PPEExtension*), *Email* (*\$Email*), *Password* (*\$Password*), *Street Address* (*\$Street*), *City* (*\$City*), *State* (*\$State*), *Zip Code* (*\$ZipCode*), *Name on Credit Card* (*\$NameOnCreditCard*), *Credit Card Number* (*\$CreditCardNumber*), *Expiration Month* (*\$ExpirationMonth*), *Expiration Year* (*\$ExpirationYear*), and *CVC* (*\$CVC*).
- If data validation is successful AND Primary Phone Number is filled, then:
  - When **Register** button is clicked:
    - If *Email* exists in the Customer table
      - Go back to Register Form with error message "The email address entered has been registered"
    - Else:
      - Run **Insert Into Customer Table** task to store *FirstName*, *MiddleName*, *LastName*, *Home Phone*, *Work Phone*, *Cell Phone*, *Email*, *Password*, *Street Address*, *City*, *State*, and *Zip Code* in User Table
      - Run **Insert Into CreditCardInfo Table** to store *Name on Credit Card*, *Credit Card No*, *Expiration Month*, *Expiration Year*, and *CVC*. in Credit Card Information Table.

```
INSERT INTO Customer (FirstName, MiddleName, LastName, Email, Password, Street Address,
City, State, ZipCode, AreaCode, MainNumber, Extension, CreditCardNumber)
VALUES ('$FirstName', '$MiddleName', '$LastName', '$Email', '$Password', '$Street ', '$City',
'$State', '$ZipCode', '$PPAreaCode', '$PPMainNumber', '$PPEExtension', '$CreditCardNumber')
```

```
INSERT INTO Phone (PhoneType, AreaCode, MainNumber, Extension)
VALUES ('Work Phone', '$WPAreaCode', '$WPMainNumber', '$WPExtension'),
      ('Home Phone', '$HPAreaCode', '$HPMainNumber', '$HPEExtension'),
      ('Cell Phone', '$CPAreaCode', '$CPMainNumber', '$CPEExtension')
```

```
INSERT INTO Other (Email, AreaCode, MainNumber, Extension)
VALUES ('$Email', '$WPAreaCode', '$WPMainNumber', '$WPExtension'),
      ('$Email', '$HPAreaCode', '$HPMainNumber', '$HPEExtension'),
      ('$Email', '$CPAreaCode', '$CPMainNumber', '$CPEExtension')
```

```
DELETE FROM Other
```

```
WHERE Email = '$Email' AND MainNumber = '$PPMainNumber'
INSERT INTO CreditCard (NameOnCreditCard, CreditCardNumber, ExpirationMonth,
ExpirationYear, CVC)
VALUES ('$NameOnCreditCard', '$CreditCardNumber', '$ExpirationMonth', '$ExpirationYear',
'$CVC')
```

## View Profile

### Abstract Code

- User clicks on **View Profile** button from **Main Menu**:
- Run the **View Profile** task: query for information from Customer and Customer's Reservations
  - Find the current Customer using the Email;
  - Display Customer Profile information;
    - Display Customer Email, FullName, Address

```
SELECT Email, CONCAT_WS(' ', FirstName, MiddleName, LastName) AS
FullName, CONCAT(Street, ' ', City, ' ', State, ' ', ZipCode ) AS Address
FROM Customer
WHERE Customer.Email = '$Email';
```

- Display Customer HomePhone, WorkPhone, CellPhone

```
SELECT Phone.AreaCode, Phone.MainNumber, Phone.Extension, CONCAT("(",
Phone.AreaCode, ") ", Phone.MainNumber, "-", Phone.Extension) AS
PhoneNumber, Phone.PhoneType
FROM Phone
LEFT JOIN Other
ON CONCAT("(", Phone.AreaCode, ") ", Phone.MainNumber, "-",
Phone.Extension) = CONCAT("(", Other.AreaCode, ") ", Other.MainNumber, "-",
Other.Extension)
WHERE Other.Email = '$Email'
UNION
SELECT Customer.AreaCode, Customer.MainNumber, Customer.Extension,
CONCAT("(", Customer.AreaCode, ") ", Customer.MainNumber, "-",
Customer.Extension) AS PhoneNumber, Phone.PhoneType
FROM Customer
LEFT JOIN Phone
ON CONCAT("(", Phone.AreaCode, ") ", Phone.MainNumber, "-",
Phone.Extension) = CONCAT("(", Customer.AreaCode, ") ",
Customer.MainNumber, "-", Customer.Extension)
WHERE Customer.Email = '$Email';
```

- Display Reservation ID, Tool Description (Power Source + SubOption + SubType), Start Date, End Date, pickup clerk ID, dropoff clerk ID, number of days, total deposit price, total rental price of all reservations made by the Customer
  - Sort the reservations from most recent to the oldest.

```
SELECT Reservation.ReservationID, GROUP_CONCAT(CONCAT_WS(" ", Tools.PowerSource,
Tools.SubOption, Tools.SubType)) AS ToolsDescription, StartDate, EndDate, CONCAT_WS(" ",
CK1.FirstName, CK1.MiddleName, CK1.LastName) AS PickupClerkFullName, CONCAT_WS(" ",
CK2.FirstName, CK2.MiddleName, CK2.LastName) AS DropoffClerkFullName, (EndDate -
StartDate) AS NumberofDays, SUM((.40 * PurchasePrice) )AS TotalDepositPrice, SUM((.15 *
PurchasePrice)) AS TotalRentalPrice
FROM Customer
INNER JOIN Reservation
ON Customer.Email = Reservation.Email
INNER JOIN Been
ON Reservation.ReservationID = Been.ReservationID
INNER JOIN Tools
ON Tools.ToolID = Been.ToolID
LEFT OUTER JOIN Clerk AS CK1
ON Reservation.PickUp_ClerkID = CK1.ClerkID
LEFT OUTER JOIN Clerk AS CK2
ON Reservation.DropOff_ClerkID = CK2.ClerkID
WHERE Customer.Email = '$Email'
GROUP BY ReservationID, StartDate, EndDate, PickupClerkFullName, DropOffClerkFullName,
NumberofDays
ORDER BY StartDate DESC;
```

## Check Tool Availability

### Abstract Code

- User clicks on **Check Tool Availability** button from **Main Menu**:
- User enters *Start Date* (\$StartDate), *End Date* (\$EndDate), select one Type (\$Type) from [All Tools, Hand Tools, Garden Tools, Ladder, and Power Tool], *Power Source* (\$PowerSource), *Sub-Type* (\$SubType), and *Custom Search* (\$CustomSearch as string with syntax %customsearch%).
  - When user select tool types, auto-populate Power Source and Sub Type dropdowns

```
SELECT PowerSource
FROM Tools
WHERE Type = '$ToolType';
```

```
SELECT SubType
FROM Tools
WHERE Type = '$ToolType';
```

- If all data validation is successful:
  - When **Search** button clicked, run the **Check Tool Availability** Task: find Tool ID, Description, Rental Price, and Deposit Price of tools which are not in repair or reserved between the dates entered and are not for sales and have type, power source, and sub type corresponding to user selection

```
SELECT COUNT(DISTINCT(Tools.ToolID))
FROM Tools
LEFT JOIN '$ToolType' AS SelectedTool
ON Tools.ToolID = SelectedTool.ToolID
WHERE Tools.ToolID NOT IN
    (SELECT ServiceOrder.ToolID
     FROM ServiceOrder
     WHERE ('$StartDate' > ServiceOrder.StartDate AND '$StartDate' <
ServiceOrder.EndDate) OR ('$EndDate' > ServiceOrder.StartDate AND '$EndDate' <
ServiceOrder.EndDate)
    UNION
    SELECT SaleOrder.ToolID
    FROM SaleOrder
    UNION
    SELECT Been.ToolID
    FROM Been
    LEFT JOIN Reservation
    ON Been.ReservationID = Reservation.ReservationID
    WHERE ('$StartDate' > Reservation.StartDate AND '$StartDate' <
Reservation.EndDate) OR ('$EndDate' > Reservation.StartDate AND '$EndDate' <
Reservation.EndDate))
AND PowerSource = '$PowerSource' AND SubType = '$SubType' AND SubOption LIKE
'$CustomSearch';
```

- If more than 10 results are returned
  - Prompt User to narrow down the search by entering more criteria

```

SELECT Tools.ToolID, CONCAT(PowerSource, ' ', SubOption, ' ', SubType) AS Description, (0.15
* PurchasePrice) AS RentalPrice, (0.4 * PurchasePrice) AS DepositPrice
FROM Tools
LEFT JOIN '$ToolType' AS SelectedTool
ON Tools.ToolID = SelectedTool.ToolID
WHERE Tools.ToolID NOT IN
    (SELECT ServiceOrder.ToolID
    FROM ServiceOrder
    WHERE ('$StartDate' > ServiceOrder.StartDate AND '$StartDate' <
ServiceOrder.EndDate) OR ('$EndDate' > ServiceOrder.StartDate AND '$EndDate' <
ServiceOrder.EndDate)
    UNION
    SELECT SaleOrder.ToolID
    FROM SaleOrder
    UNION
    SELECT Been.ToolID
    FROM Been
    LEFT JOIN Reservation
    ON Been.ReservationID = Reservation.ReservationID
    WHERE ('$StartDate' > Reservation.StartDate AND '$StartDate' <
Reservation.EndDate) OR ('$EndDate' > Reservation.StartDate AND '$EndDate' <
Reservation.EndDate))
AND PowerSource = '$PowerSource' AND SubType = '$SubType' AND SubOption LIKE
'$CustomSearch';

```

## View Tool Details

### Abstract Code

- User clicks on **Tool Full Description** button/link for the specific Tool ID:
  - Run **View Tool Details** task based on the Tool ID.
    - Display *Tool ID (\$ToolID)*, *Tool Type (\$Type)*, *Short Description (Combination of Power Source, Sub Option, Sub Type from Tools)*, *Full Description (Combination of Dimensions, Power Source, Sub Option, Sub type, other descriptors, and Manufacturer from Tools)*, *Deposit Price (\$DepositPrice)*, and *Rental Price (\$RentalPrice)*

```

SELECT T.ToolID, T.Type, CONCAT_WS(' ', (CASE PowerSource
    WHEN 'manual' THEN ''
    ELSE PowerSource
    END), SubOption, SubType) AS ShortDescription, CONCAT_WS(' ',
Width_Diameter_Int, Width_Diameter_Fraction, Width_Diameter_Unit, 'W x', LengthInt,
LengthFraction, LengthUnit, 'L', Weight, 'lb', (CASE PowerSource

```

[Table of Contents:](#)

```

        WHEN 'manual' THEN ''
        ELSE PowerSource
    END), SubOption, SubType, (CASE Type
        WHEN 'Hand Tool' THEN (CASE SubType
            WHEN 'Screwdriver' THEN CONCAT('#', SCR.ScrewSize)
            WHEN 'Socket' THEN CONCAT_WS(' ', SOC.DriveSize,
SOC.SaeSize, (CASE DeepSocket
                WHEN TRUE THEN 'Deep Socket'
                ELSE 'Non-Deep Socket'
            END))
            WHEN 'Ratchet' THEN RAT.DriveSize
        WHEN 'Plier' THEN (CASE PLI.Adjustable
            WHEN TRUE THEN 'Adjustable'
            ELSE 'Non-Adjustable'
        END)
        WHEN 'Gun' THEN CONCAT_WS(' ', GUN.GaugeRating,
GUN.Capacity, 'nails/staples')
        WHEN 'Hammer' THEN (CASE HAM.AntiVibration
            WHEN TRUE THEN 'Anti-Vibration'
            ELSE 'Non Anti-Vibration'
        END)
    END)
    WHEN 'Garden Tool' THEN CONCAT_WS(' ', HandleMaterial, (CASE
SubType
        WHEN 'Pruning Tool' THEN CONCAT_WS(' ', PRU.BladeMaterial,
PRU.BladeLength)
        WHEN 'Striking Tool' THEN HeadWeight
        WHEN 'Digging Tools' THEN CONCAT_WS(' ', DIG.BladeWidth,
DIG.BladeLength)
        WHEN 'Rake Tools' THEN TineCount
        WHEN 'WheelbarrowTools' THEN CONCAT_WS(' ', BinMaterial,
BinVolume, WheelCount)
    END))
    WHEN 'LadderTools' THEN CONCAT_WS(' ', StepCount, WeightCapacity,
(CASE SubType
        WHEN 'Straight' THEN (CASE RubberFeet
            WHEN TRUE THEN 'Rubber Feet'
            ELSE 'Non-Rubber Feet'
        END)
        WHEN 'Strep' THEN (CASE PailShelf
            WHEN TRUE THEN 'Pail Shelf'
            ELSE 'Non-Pail Shelf'
        END)
    END))
END))

```



```

        WHEN 'Power Tools' THEN CONCAT_WS(' ', PT.VoltRating,
PT.AmpRating, PT.AmpRatingUnit, PT.RPMRatingMin, PT.RPMRatingMax, (CASE SubType
        WHEN 'Drill' THEN CONCAT_WS(' ', (CASE AdjustableClutch
        WHEN TRUE THEN 'Adjustable Clutch'
        ELSE 'Non-Adjustable Clutch'
        END), TorqueRatingMin, TorqueRatingMax)
        WHEN 'Saw' THEN BladeSize
        WHEN 'Sander' THEN (CASE DustBag
        WHEN TRUE THEN 'Dust Bag'
        ELSE 'Non-Dust Bag'
        END)
        WHEN 'AirCompressor' THEN CONCAT_WS(' ', TankSize, 'gallon' ,
PressureRatingMin, 'psi', PressureRatingMax, 'psi')
        WHEN 'Mixer' THEN CONCAT_WS(' ', MotorRating, 'HP',
DrumSize, 'cu ft')
        WHEN 'Generator' THEN CONCAT_WS(' ', GEN.PowerGenerated,
GEN.PowerFraction, GEN.PowerRatingUnit)
        END))
        END), Manufacturer, Material, ACC.Quantity,
ACC.AccessoryDescription, ACC.BatteryQuantity, ACC.BatteryType, ACC.VoltRating) AS
FullDescriptions, ROUND(.15* PurchasePrice,2) AS RentalPrice, ROUND(.4 * PurchasePrice, 2)
AS DepositPrice
FROM Tools AS T
    LEFT OUTER JOIN HandTools AS HT ON T.ToolID = HT.ToolID
    LEFT OUTER JOIN Screwdriver AS SCR ON T.ToolID = SCR.ToolID
    LEFT OUTER JOIN Socket AS SOC ON T.ToolID=SOC.ToolID
    LEFT OUTER JOIN Ratchet AS RAT ON T.ToolID = RAT.ToolID
    LEFT OUTER JOIN Plier AS PLI ON T.ToolID = PLI.ToolID
    LEFT OUTER JOIN Gun AS GUN ON T.ToolID = GUN.ToolID
    LEFT OUTER JOIN Hammer AS HAM ON T.ToolID = HAM.ToolID
    LEFT OUTER JOIN GardenTools AS GT ON T.ToolID = GT.ToolID
    LEFT OUTER JOIN PruningTools AS PRU ON T.ToolID = PRU.ToolID
    LEFT OUTER JOIN StrikingTools AS STR ON T.ToolID = STR.ToolID
    LEFT OUTER JOIN DiggingTools AS DIG ON T.ToolID = DIG.ToolID
    LEFT OUTER JOIN RakeTools AS RAK ON T.ToolID = RAK.ToolID
    LEFT OUTER JOIN WheelbarrowTools AS WHE ON T.ToolID = WHE.ToolID
    LEFT OUTER JOIN PowerTools AS PT ON T.ToolID = PT.ToolID
    LEFT OUTER JOIN Drill AS DRI ON T.ToolID = DRI.ToolID
    LEFT OUTER JOIN Saw AS SAW ON T.ToolID = SAW.ToolID
    LEFT OUTER JOIN Sander AS SAN ON T.ToolID = SAN.ToolID
    LEFT OUTER JOIN AirCompressor AS AIR ON T.ToolID = AIR.ToolID
    LEFT OUTER JOIN Mixer AS MIX ON T.ToolID = MIX.ToolID
    LEFT OUTER JOIN Generator AS GEN ON T.ToolID = GEN.ToolID
    LEFT OUTER JOIN LadderTools AS LD ON T.ToolID = LD.ToolID

```

```
LEFT OUTER JOIN Straight AS STL ON T.ToolID = STL.ToolID
LEFT OUTER JOIN Strep AS STE ON T.ToolID = STE.ToolID
LEFT OUTER JOIN Contain AS CONT ON T.ToolID = CONT.ToolID
LEFT OUTER JOIN Accessories AS ACC ON CONT.AccessoryID = ACC.AccessoryID
WHERE T.ToolID = '$ToolID'
```

### View/ Add/ Calculate Tools for Rent

Abstract code

- Customer select *StartDate* (\$StartDate), *EndDate* (\$EndDate), *Type* (\$Type) [one of All Tools, Hand Tool, Garden Tool, Ladder, Power Tool] and *Power Source* (\$PowerSource), *SubType* (\$SubType), and *CustomSearch* (\$CustomSearch as string with syntax %customsearch%).
  - When *PowerSource* dropdown is selected, *SubType* dropdown would change accordingly based on business constraints.

```
SELECT PowerSource
FROM Tools
WHERE Type = '$ToolType';

SELECT SubType
FROM Tools
WHERE Type = '$ToolType';
```

- When **Search** button clicked: find Tool ID, Description, Rental Price, and Deposit Price of tools which are not in repair or reserved between the dates entered and are not for sales and have type, power source, and sub type corresponding to user selection
- When **Add** tickbox is ticked, move the tool ticked from Available Tools for Rent to Tools Added to Reservation and count the number of tools in Tools Added to Reservation section.

```

SELECT COUNT(DISTINCT(Tools.ToolID))
FROM Tools
LEFT JOIN '$ToolType' AS SelectedTool
ON Tools.ToolID = SelectedTool.ToolID
WHERE Tools.ToolID NOT IN
    (SELECT ServiceOrder.ToolID
     FROM ServiceOrder
     WHERE ('$StartDate' > ServiceOrder.StartDate AND '$StartDate' <
ServiceOrder.EndDate) OR ('$EndDate' > ServiceOrder.StartDate AND '$EndDate' <
ServiceOrder.EndDate)
    UNION
    SELECT SaleOrder.ToolID
    FROM SaleOrder
    UNION
    SELECT Been.ToolID
    FROM Been
    LEFT JOIN Reservation
    ON Been.ReservationID = Reservation.ReservationID
    WHERE ('$StartDate' > Reservation.StartDate AND '$StartDate' <
Reservation.EndDate) OR ('$EndDate' > Reservation.StartDate AND '$EndDate' <
Reservation.EndDate))
AND PowerSource = '$PowerSource' AND SubType = '$SubType';

```

- If the number of tools exceeds 10 in the Tools Added to Reservation section
  - Prompt the user to reduce the number of tools in Tools Added to Reservation section by adding more search criteria.
- When **Remove** tickbox is ticked, move the tool ticked from Tools Added to Reservation to Available Tools for Rent.
- If **Calculate Total** button is pushed
  - Store the information in Tools Added to Reservation and run **Check and Submit Reservation** Task.

```

SELECT Tools.ToolID, CONCAT(PowerSource, ' ', SubOption, ' ', SubType) AS Description, (0.15
* PurchasePrice) AS RentalPrice, (0.4 * PurchasePrice) AS DepositPrice
FROM Tools
LEFT JOIN '$ToolType' AS SelectedTool
ON Tools.ToolID = SelectedTool.ToolID
WHERE Tools.ToolID NOT IN
    (SELECT ServiceOrder.ToolID
    FROM ServiceOrder
    WHERE ('$StartDate' > ServiceOrder.StartDate AND '$StartDate' <
ServiceOrder.EndDate) OR ('$EndDate' > ServiceOrder.StartDate AND '$EndDate' <
ServiceOrder.EndDate)
    UNION
    SELECT SaleOrder.ToolID
    FROM SaleOrder
    UNION
    SELECT Been.ToolID
    FROM Been
    LEFT JOIN Reservation
    ON Been.ReservationID = Reservation.ReservationID
    WHERE ('$StartDate' > Reservation.StartDate AND '$StartDate' <
Reservation.EndDate) OR ('$EndDate' > Reservation.StartDate AND '$EndDate' <
Reservation.EndDate))
AND PowerSource = '$PowerSource' AND SubType = '$SubType';

```

- If the tools selected is not available, the system will check if an identical tool requested is due to return within the next 24 hours from the time of request.

```

SELECT
    T.ToolID,
    R.EndDate
FROM
    Tool AS T
INNER JOIN
    Been AS B ON B.ToolID = T.ToolID
INNER JOIN
    Reservation AS R ON R.ReservationID = B.ReservationID
WHERE
    R.EndDate = CURDATE() + INTERVAL 1 DAY
    AND T.Type = '$type'
    AND T.PowerSource LIKE '$PowerSource'
    AND T.SubType LIKE '$SubType'
    AND T.SubOption LIKE '$CustomSearch'

```

## Check and Submit Reservation

### Abstract Code

- When **Calculate Total** button from **Make Reservation** Form is clicked
  - Bring in information from Tools Added to Reservation section
  - If **Submit** button is clicked:
    - Check the availability for the Tools selected.
      - If not available for any tool selected
        - return an error message prompt the user to re-select and return to **Make Reservation** Form
      - Else
        - Insert Record for **Reservation** Table

```
INSERT INTO Reservation (Email, StartDate, EndDate)
VALUES ('$Email', '$StartDate', '$EndDate');
```

- Insert Record for **Been** Table

```
INSERT INTO Been (ToolID, ReservationID)
SELECT T.ToolID, R.ReservationID
FROM `Tools T`, `Reservation R`
WHERE R.Email = '$Email' AND T.ToolID = '$ToolID';
```

- Display **Reservation Confirmation** Form which populates information from Reservation Summary table
  - Display Reservation ID, Reservation Dates, and Number of Days Rented

```
SELECT R.ReservationID, CONCAT(R.StartDate, " - ", R.EndDate) AS
ReservationDates, DATEDIFF(R.EndDate, R.StartDate) AS
NumberOfDaysRented
FROM `Reservation R`
WHERE R.Email = '$Email';
```

- Display ToolID, Total Deposit Price, Total Rental Price, and Tool Description

```
Select B.ToolID, CONCAT(T.PowerSource, " ", T.SubOption, " ", T.SubType) AS
ToolDescription, ROUND(T.PurchasePrice * 0.15 * DATEDIFF(R.EndDate, R.StartDate), 2)
AS RentalPrice, ROUND(T.PurchasePrice * 0.4, 2) AS DepositPrice
From `Been B`, `Tools T`, `Reservation R`
WHERE (R.Email = '$Email' AND R.PickUp_ClerkID is NULL) AND T.ToolID = B.ToolID ;
```

- If **Reset** button is clicked
  - Return to **Make Reservation** Form

### View/ Add/ Calculate Tools for Purchase

#### Abstract Task

- Customer selects one of Type (**\$Type**) All Tools, Hand Tool, Garden Tool, Ladder, Power Tool and Power Source (**\$PowerSource**) and SubType (**\$SubType**), and Custom Search (**\$CustomSearch** as string with syntax %customsearch%)
  - When *PowerSource* dropdown is selected, *SubType* dropdown would change accordingly based on business constraints.

```
SELECT DISTINCT PowerSource
FROM Tools
WHERE Type = '$ToolType';
```

```
SELECT DISTINCT SubType
FROM Tools
WHERE Type = '$ToolType';
```

- When **Search** button is clicked, search tools available for purchase based on criteria selected and return results satisfying *Tool ID, Description, and Purchase Price*.

```
SELECT Tools.ToolID, CONCAT(PowerSource, ' ', SubOption, ' ', SubType) AS Description, (0.15 * PurchasePrice) AS RentalPrice, (0.4 * PurchasePrice) AS DepositPrice
FROM Tools
INNER JOIN SaleOrder
ON SaleOrder.ToolID = Tools.ToolID
WHERE SaleOrder.SoldDate IS NULL AND PowerSource = '$PowerSource' AND SubType = '$SubType' AND SubOption LIKE '$CustomSearch'
```

- When **Calculate Total** button is pushed, close **Purchase Tool** Form and open **Purchase Summary** Form and populate the form with Tool information from Tools Added to Purchase section.

### Check and Submit Purchase Confirmation

#### Abstract Code

- When **Purchase Tool** button from **Purchase Tool** Form is clicked
  - Bring in the corresponding tool information
  - Check the availability for the Tool selected.
    - If not available for tool selected
      - return an error message prompt the user to re-select and return to **Purchase Tool** Form

[Table of Contents:](#)

- Else
  - Update Record for **SaleOrder** Table

```
UPDATE SaleOrder S
SET S.SoldDate = CURDATE(), S.Email = (SELECT C.Email FROM
`Customer C` WHERE C.Email = '$Email')
WHERE S.ToolID = '$ToolID';
```

- Display **Purchase Confirmation** Form which populates information from Purchase Summary table
  - Display SaleOrder ID, SoldDate, and Price

```
SELECT S.SaleOrderID, S.SoldDate, ROUND(T.PurchasePrice * 0.5, 2) AS
Price
FROM `SaleOrder S`, `Tools T`
WHERE S.Email = '$Email' AND S.ToolID = T.ToolID ;
```

- Display ToolID, Tool Description, and Sold Price

```
SELECT T.ToolID, CONCAT(T.PowerSource, " ", T.SubOption, " ",
T.SubType) AS ToolDescription, ROUND(T.PurchasePrice * 0.5, 2) AS
Price
FROM `SaleOrder S`, `Tools T`
WHERE S.Email = '$Email' AND S.ToolID = T.ToolID ;
```

- If **Reset** button is clicked
  - Return to **Make Reservation** Form

## View All Pick-UP Reservations

### Abstract Code

- User selects **Pickup** button from **Main Menu**
  - Display all Reservations that has not been picked up by customers, (reservations where Pickup Clerk is Null.)
  - If Data validation succeeds for Reservation ID entered
    - If the **Reservation ID** entered can be found in the Reservation ID list in the form
      - Open **Summary of Pickup Reservation** Form

```
SELECT R.ReservationID, C.Username, C.CustomerID, R.StartDate, R.EndDate
FROM Customer AS C
INNER JOIN Reservation AS R
ON C.Email = R.Email
WHERE R.PickUp_ClerkID IS NULL;
```

- Else return error message indicating Reservation ID not found

### View/ Confirm Pick-Up Reservation

#### Abstract Code

- When **Summary of Pickup Reservation** Form is opened/ loaded
- Display reservation selected to be picked-up, including Reservation ID, user name, total deposit price, and total rental price. Reservation ID entered must be from the Reservation IDs in the pick-up reservation table

```
# View Reservation ID, customer name
SELECT R.ReservationID, CONCAT(C.FirstName, ' ', LastName) AS FullName,
FROM Customer AS C
INNER JOIN Reservation AS R
ON C.Email = R.Email
WHERE R.ReservationID = '$ReservationID';

#View total deposit and total rental price
SELECT SUM(0.15 * PurchasePrice) AS TotalRentalPrice, SUM(0.4 * PurchasePrice) AS
TotalDepositPrice
FROM Tools
WHERE ToolID IN (SELECT ToolID
                  FROM Been
                  WHERE ReservationID = '$ReservationID');
```

- When Confirm button is pushed, pickup clerk ID in **Reservation** form will be updated and then **View Rental Contract** form would open.

```
UPDATE Reservation
SET PickUp_ClerkID = ('$PickUp_ClerkID')
WHERE ReservationID = '$ReservationID';
```

### View/ Update Credit Card Info

#### Abstract Code

- User chooses either enter "Existing" or "New" credit card information
  - If "Existing" is selected,
    - Enter Updated Credit Card Information section would be hidden
  - If "New" is selected,

[Table of Contents:](#)



- User enter new credit card information
  - Display current CreditCard Info and the new CreditCardNumber information will replace the existing CreditCardNumber on [Customer](#) table.

```
SELECT *
FROM `CreditCard C`
WHERE C.CreditCardNumber = (SELECT Customer.CreditCardNumber
                           FROM `Customer`
                           WHERE Customer.Email = '$Email');

SET foreign_key_checks = 0;
UPDATE `Customer`
SET CreditCardNumber = '$CreditCardNumber'
WHERE Customer.Email = '$Email';
SET foreign_key_checks = 1;
```

- Insert creditcard Info in [CreditCard](#) table, if the new credit card info does not exist in current [CreditCard](#) table

```
INSERT INTO CreditCard (CreditCardNumber, NameOnCreditCard, CVC, ExpirationMonth,
ExpirationYear)
SELECT '$CreditCardNumber', '$NameOnCreditCard', '$CVC', '$ExpirationMonth',
'ExpirationYear'
FROM `DUAL`
WHERE NOT EXISTS (
    SELECT CreditCardNumber
    FROM `CreditCard`
    WHERE CreditCardNumber = '$CreditCardNumber'
);
```

## View Rental Contract

### Abstract Code

- Display **Rental Contract** form with clerk full name, customer full name, customer credit card number, start date, end data, and tool information including tool ID, tool name, rental price, deposit price in the reservation selected.

```

# Display clerk full name
SELECT CONCAT(FirstName, ' ', LastName) AS ClerkFullName
FROM Clerk
WHERE ClerkID = (SELECT Pickup_ClerkID
                  FROM Reservation
                  WHERE ReservationID = 'ReservationID');

# Display customer full name
SELECT CONCAT(FirstName, ' ', LastName) AS FullName
FROM Customer
WHERE Email = (SELECT Email
                FROM Reservation
                WHERE ReservationID = 'ReservationID');

# Display customer credit card number
SELECT CreditCardNumber
FROM Customer
WHERE Customer.Email = (SELECT Email
                        FROM Reservation
                        WHERE ReservationID = '$ReservationID')

# Display start date and end date
SELECT StartDate, EndDate
FROM Reservation
WHERE ReservationID = '$ReservationID'

# Display tool ID, tool name, rental price and deposit price
SELECT Tools.ToolID, CONCAT(PowerSource, ' ', SubOption, ' ', SubType) AS ToolName, (0.15 *
PurchasePrice) AS RentalPrice, (0.4 * PurchasePrice) AS DepositPrice
FROM Tools
WHERE ToolID IN (SELECT ToolID
                  FROM Been) AND ReservationID = '$ReservationID'

```

### View All Drop-Off Reservations

Abstract Code

- User selects **Dropoff** button from **Main Menu**
  - Display all reservations available for drop off from Tools, Reservation, and Customer Tables

```

SELECT R.ReservationID, C.Username, C.CustomerID, R.StartDate, R.EndDate
FROM Customer AS C
INNER JOIN Reservation AS R
ON C.Email = R.Email
WHERE R.PickUp_ClerkID IS NOT NULL AND R.DropOff_ClerkID IS NULL;

```

[Table of Contents:](#)

- If Data validation succeeds for Reservation ID entered
  - If the **Reservation ID** entered can be found in the Reservation ID list in the form
    - Open **Summary of Drop Off Reservation** Form
  - Else return error message indicating Reservation ID not found

### View/ Confirm Drop-Off Reservation Summary

Abstract Code

- When **Summary of Drop Off Reservation** Form is opened/ loaded
- Display reservation selected to be dropped up, including Reservation ID, user name, total deposit price, and total rental price. Reservation ID previously entered must be from the Reservation IDs in the drop-off reservation table

```
# View Reservation ID, customer name
SELECT R.ReservationID, CONCAT(C.FirstName, ' ', LastName) AS FullName,
FROM Customer AS C
INNER JOIN Reservation AS R
ON C.Email = R.Email
WHERE R.ReservationID = '$ReservationID';

#View total deposit and total rental price
SELECT SUM(0.15 * PurchasePrice) AS TotalRentalPrice, SUM(0.4 * PurchasePrice) AS
TotalDepositPrice
FROM Tools
WHERE ToolID IN (SELECT ToolID
                  FROM Been
                  WHERE ReservationID = '$ReservationID');
```

- When Confirm button is pushed, dropoff clerk ID in **Reservation** form will be updated and then **View Final Receipt** form would open.

```
UPDATE Reservation
SET DropOff_ClerkID = ('$DropOff_ClerkID')
WHERE ReservationID = '$ReservationID';
```

### View Final Receipt

Abstract Code

- Display **Final Receipt** form with reservation ID, customer full name, total rental price, and total deposit price in the reservation selected.

[Table of Contents:](#)

```
# View Reservation ID, customer name
SELECT R.ReservationID, CONCAT(C.FirstName, ' ', LastName) AS FullName,
FROM Customer AS C
INNER JOIN Reservation AS R
ON C.Email = R.Email
WHERE R.ReservationID = '$ReservationID';

#View total deposit and total rental price
SELECT SUM(0.15 * PurchasePrice) AS TotalRentalPrice, SUM(0.4 * PurchasePrice) AS
TotalDepositPrice
FROM Tools
WHERE ToolID IN (SELECT ToolID
                  FROM Been
                  WHERE ReservationID = '$ReservationID');

# Display tool ID, tool name, rental price and deposit price
SELECT Tools.ToolID, CONCAT(PowerSource, ' ', SubOption, ' ', SubType) AS ToolName, (0.15 *
PurchasePrice) AS RentalPrice, (0.4 * PurchasePrice) AS DepositPrice
FROM Tools
WHERE ToolID IN (SELECT ToolID
                  FROM Been) AND ReservationID = '$ReservationID'
```

## Add Tool

### Abstract Code

- Clerk click **Add Tool** button from Main Menu:
  - Clerk then enter Type (\$Type), SubType (\$SubType), SubOption, (\$SubOption), Width\_Diameter\_Int (\$Width\_Diameter\_Int), Width\_Diameter\_Fraction (\$Width\_Diameter\_Fraction), Width\_Diameter\_Unit (\$Width\_Diameter\_Unit), LengthInt (\$LengthInt), LengthFraction (\$LengthFraction), LengthUnit (\$LengthUnit), Weight (\$Weight), Manufacturer (\$Manufacturer), Material (\$Material), PowerSource (\$PowerSource), PurchasePrice (\$PurchasePrice).
    - If Hand Tool is selected:
 

Clerk will have the ability to see the following fields: ScrewSize (\$ScrewSize), DriveSize (\$DriveSize), SaeSize (\$SaeSize), DeepSocket (\$DeepSocket), Adjustable (\$Adjustable), GaugeRating (\$GaugeRating), Capacity (\$Capacity), AntiVibration (\$AntiVibration). These fields will be dynamically shown/hidden based on the Clerk's inputs. Some of the fields are required.
    - If Garden Tool is selected:
 

Clerk will have the ability to see the following fields: HandleMaterial (\$HandleMaterial), BladeMaterial (\$BladeMaterial), BladeLength (\$BladeLength), HeadWeight (\$HeadWeight), BladeWidth (\$BladeWidth), TineCount (\$TineCount), BinMaterial (\$BinMaterial), BinVolume

(\$BinVolume), WheelCount (\$WheelCount). These fields will be dynamically shown/hidden based on the Clerk's inputs. Some of the fields are required.

■ If Ladder is selected:

Clerk will have the ability to see the following fields: StepCount (\$StepCount), WeightCapacity (\$WeightCapacity), RubberFeet (\$RubberFeet), PailShelf (\$PailShelf). These fields will be dynamically shown/hidden based on the Clerk's inputs. Some of the fields are required.

■ If Power Tool is selected:

Clerk will have the ability to see the following fields: VoltRating (\$VoltRating), AmpRating (\$AmpRating), RPMRatingMin (\$RPMRatingMin), RPMRatingMax (\$RPMRatingMax), AdjustableClutch (\$AdjustableClutch), TorqueRatingMin (\$TorqueRatingMin), TorqueRatingMax (\$TorqueRatingMax), BladeSize (\$BladeSize), DustBag (\$DustBag), TankSize (\$TankSize), PressureRatingMin (\$PressureRatingMin), PressureRatingMax (\$PressureRatingMax), MotorRating (\$MotorRating), DrumSize (\$DrumSize), PowerRating (\$PowerRating). These fields will be dynamically shown/hidden based on the Clerk's inputs. Some of the fields are required.

- If power tool accessory is available:
  - Insert Quantity (\$Quantity), AccessoryDescription (\$AccessoryDescription)
- If cordless tool is selected:
  - Insert BatteryType (\$BatteryType), BatteryQuantity (\$BatteryQuantity), VoltRating (\$VoltRating)
- When **Confirm** button is clicked
  - The system will check whether all the required fields are filled.
    - Insert record to the Tool table

```
INSERT INTO Tools (Type, SubType, SubOption, Width_Diameter_Int,
Width_Diameter_Fraction, Width_Diameter_Unit, LengthInt, LengthFraction, LengthUnit,
Weight, Manufacturer, Material, PowerSource, PurchasePrice) VALUES ($Type, $SubType,
$SubOption, $Width_Diameter_Int, $Width_Diameter_Fraction, $Width_Diameter_Unit,
$LengthInt, $LengthFraction, $LengthUnit, $Weight, $Manufacturer, $Material,
$PowerSource, $PurchasePrice)
```

- Get the ToolID (\$ToolID) for the Tool that was just inserted

```
SELECT LAST_INSERT_ID()
```

- If Hand Tools is selected

```
INSERT INTO HandTool (ToolID) VALUES ($ToolID)
```

- If Screwdriver is selected

```
INSERT INTO Screwdriver (ToolID, ScrewSize) VALUES ($ToolID, $ScrewSize)
```
- If Socket is selected

```
INSERT INTO Socket (ToolID, DriveSize, SaeSize, DeepSocket) VALUES ($ToolID, $DriveSize, $SaeSize, $DeepSocket)
```
- If Ratchet is selected

```
INSERT INTO Ratchet (ToolID, DriveSize) VALUES ($ToolID, $DriveSize)
```
- If Plier is selected

```
INSERT INTO Plier (ToolID, Adjustable) VALUES ($ToolID, $Adjustable)
```
- If Gun is selected

```
INSERT INTO Gun (ToolID, GaugeRating, Capacity) VALUES ($ToolID, $GaugeRating, $Capacity)
```
- If Hammer is selected

```
INSERT INTO Hammer (ToolID, AntiVibration) VALUES ($ToolID, $AntiVibration)
```
- If Gardern Tools is selected

```
INSERT INTO GardenTool (ToolID, HandleMaterial) VALUES ($ToolID, $HandleMaterial)
```
- If Pruning is selected

```
INSERT INTO PruningTools (ToolID, BladeMaterial, BladeLength) VALUES ($ToolID, $BladeMaterial, $BladeLength)
```
- If Striking is selected

```
INSERT INTO StrikingTools (ToolID, HeadWeight) VALUES ($ToolID, $HeadWeight)
```
- If Digging is selected

```
INSERT INTO DiggingTools (ToolID, BladeWidth, BladeLength) VALUES ($ToolID, $BladeWidth, $BladeLength)
```
- If RakeTools is selected

```
INSERT INTO RakeTools (ToolID, TineCount) VALUES ($ToolID, $TineCount)
```
- If Wheelbarrow is selected

```
INSERT INTO WheelbarrowTools (ToolID, BinMaterial, BinVolume,  
WheelCount) VALUES ($ToolID, $BinMaterial, $BinVolume,  
$WheelCount)
```

- If Ladder is selected

```
INSERT INTO Ladder (ToolID, StepCount, WeightCapacity) VALUES ($ToolID,  
$StepCount, $WeightCapacity)
```

- If Straight is selected

```
INSERT INTO Straight (ToolID, RubberFeet) VALUES ($ToolID,  
$RubberFeet)
```

- If Step is selected

```
INSERT INTO Step (ToolID, PailShelf) VALUES ($ToolID, $PailShelf)
```

- If Power Tool is selected

```
INSERT INTO PowerTool (ToolID, VoltRating, AmpRating, RPMRatingMin,  
RPMRatingMax, Quantity, AccessoryDescription, BatteryType, BatteryQuantity,  
VoltRating) VALUES ($ToolID, $VoltRating, $AmpRating, $RPMRatingMin,  
$RPMRatingMax, $Quantity, $AccessoryDescription, $BatteryType, $BatteryQuantity,  
$VoltRating)
```

- If Power Drill is selected

```
INSERT INTO Drill (ToolID, AdjustableClutch, TorqueRatingMin,  
TorqueRatingMax) VALUES ($ToolID, $AdjustableClutch,  
$TorqueRatingMin, $TorqueRatingMax)
```

- If Power Saw is selected

```
INSERT INTO Saw (ToolID, BladeSize) VALUES ($ToolID, $BladeSize)
```

- If Power Sander is selected

```
INSERT INTO Sander (ToolID, DustBag) VALUES ($ToolID, $DustBag)
```

- If Power Air-Compressor is selected

```
INSERT INTO AirCompressor (ToolID, TankSize, PressureRatingMin,  
PressureRatingMax) VALUES ($ToolID, $TankSize, $PressureRatingMin,  
$PressurRatingMax)
```

- If Power Mixer is selected

```
INSERT INTO Mixer (ToolID, MotorRating, DrumSize) VALUES ($ToolID,  
$MotorRating, $DrumSize)
```

- If Power Generator is selected

```
INSERT INTO Generator (ToolID, PowerGenerated, PowerFraction, Unit)
VALUES ($ToolID, $PowerGenerated, $PowerFraction, $Unit)
```

## Repair Tool

### Abstract Code

- User selects one of Type (\$Type) [All Tools, Hand Tool, Garden Tool, Ladder, Power Tool], Power Source (\$PowerSource) and SubType (\$SubType), and Custom Search (\$CustomSearch as string with syntax %customsearch%).
  - When PowerSource dropdown and SubType dropdown would be auto-populated.

```
SELECT PowerSource
FROM Tools
WHERE Type = '$ToolType';
SELECT SubType
FROM Tools
WHERE Type = '$ToolType';
```

- When **Search** button is clicked, find Tool ID, Description, Rental Price, and Deposit Price of tools which are not in repair or reserved between the dates entered and are not for sales and have type, power source, and sub type corresponding to user selection

```
SELECT Tools.ToolID, CONCAT(PowerSource, ' ', SubOption, ' ', SubType) AS Description, (0.15
* PurchasePrice) AS RentalPrice, (0.4 * PurchasePrice) AS DepositPrice
FROM Tools
LEFT JOIN '$ToolType' AS SelectedTool
ON Tools.ToolID = SelectedTool.ToolID
WHERE Tools.ToolID NOT IN
    (SELECT ServiceOrder.ToolID
     FROM ServiceOrder
     WHERE ('$StartDate' > ServiceOrder.StartDate AND '$StartDate' <
ServiceOrder.EndDate) OR ('$EndDate' > ServiceOrder.StartDate AND '$EndDate' <
ServiceOrder.EndDate))
    UNION
    SELECT SaleOrder.ToolID
    FROM SaleOrder
    UNION
    SELECT Been.ToolID
    FROM Been
    LEFT JOIN Reservation
    ON Been.ReservationID = Reservation.ReservationID
    WHERE ('$StartDate' > Reservation.StartDate AND '$StartDate' <
Reservation.EndDate) OR ('$EndDate' > Reservation.StartDate AND '$EndDate' <
Reservation.EndDate))
AND PowerSource = '$PowerSource' AND SubType = '$SubType' AND SubOption LIKE
```



```
'$CustomSearch'
```

- User then select the corresponding *ToolID* and Enter the *Service Cost*.
- When **Confirm** button is clicked
  - If data validation is successful for Tool ID and Service Cost
    - If *Tool ID* is found in the list of Available Tools
      - Insert record to ServiceOrder table
    - Else
      - Prompt warning to re-enter tool ID
  - Else
    - Prompt warnings about data validation rules

```
INSERT INTO ServiceOrder (ClerkID, ToolID, StartDate, EndDate, ServiceCost)
VALUES ('$ClerkID', '$ToolID', '$StartDate', '$EndDate', '$ServiceCost');
```

### View/ Filter/ Override Current Status of All Tools in Repair

Abstract Code

- User selects one of Type (*\$Type*) [All Tools, Hand Tool, Garden Tool, Ladder, Power Tool] and *Custom Search* (*\$CustomSearch* as string with syntax %customsearch%)
- When **Search** button is clicked, search tools table based on type selected

```
SELECT ServiceOrderID, S.ToolID, CONCAT(T.PowerSource, ' ', T.SubOption, ' ', T.SubType) AS
Description, StartDate, EndDate, ServiceCost, CONCAT(C.FirstName, ' ', C.LastName) AS Clerk
FROM ServiceOrder AS S
LEFT JOIN Clerk AS C
ON S.ClerkID = C.ClerkID
LEFT JOIN Tools AS T
ON T.ToolID = S.ToolID
WHERE T.Type = '$Type' AND T.SubOption = '$CustomSearch'
```

- When **Fix-Now** button is clicked,
  - Remove the tool from ServiceOrder table

```
DELETE FROM ServiceOrder
WHERE ServiceOrderID = '$ServiceOrderID';
```

### View/ Filter/ Sell Available Tools

Abstract Code

[Table of Contents:](#)

- User selects one of Type (**\$Type**) [All Tools, Hand Tool, Garden Tool, Ladder, Power Tool], Power Source (**\$PowerSource**), SubType (**\$SubType**), and Custom Search (**\$CustomSearch** as string with syntax %customsearch%)
  - When *PowerSource* dropdown and *SubType* dropdown would be auto-populated

```
SELECT PowerSource
FROM Tools
WHERE Type = '$ToolType';

SELECT SubType
FROM Tools
WHERE Type = '$ToolType';
```

- When **Search** button is clicked, find Tool ID, Description, Rental Price, and Deposit Price of tools which are not in repair or reserved between the dates entered and are not for sales and have type, power source, and sub type corresponding to user selection

```
SELECT Tools.ToolID, CONCAT(PowerSource, ' ', SubOption, ' ', SubType) AS Description, (0.15
* PurchasePrice) AS RentalPrice, (0.4 * PurchasePrice) AS DepositPrice
FROM Tools
LEFT JOIN '$ToolType' AS SelectedTool
ON Tools.ToolID = SelectedTool.ToolID
WHERE Tools.ToolID NOT IN
    (SELECT ServiceOrder.ToolID
    FROM ServiceOrder
    WHERE ('$StartDate' > ServiceOrder.StartDate AND '$StartDate' <
ServiceOrder.EndDate) OR ('$EndDate' > ServiceOrder.StartDate AND '$EndDate' <
ServiceOrder.EndDate))
    UNION
    SELECT SaleOrder.ToolID
    FROM SaleOrder
    UNION
    SELECT Been.ToolID
    FROM Been
    LEFT JOIN Reservation
    ON Been.ReservationID = Reservation.ReservationID
    WHERE ('$StartDate' > Reservation.StartDate AND '$StartDate' <
Reservation.EndDate) OR ('$EndDate' > Reservation.StartDate AND '$EndDate' <
Reservation.EndDate))
AND PowerSource = '$PowerSource' AND SubType = '$SubType' AND SubOption LIKE
'$CustomSearch'
```

- When **Sell Tool** button is pushed

Insert ClerkID, ToolID, Email, ForSaleDate into Sale Order table  
Re-query the form

```
INSERT INTO SaleOrder (ClerkID, ToolID, Email, ForSaleDate)
VALUES ('$ClerkID', '$ToolID', '$Email', '$ForSaleDate');
```

### View/ Filter Sale Status

Abstract Code

- User selects one of Type (**\$Type**) [All Tools, Hand Tool, Garden Tool, Ladder, Power Tool] and *Custom Search* (**\$CustomSearch** as string with syntax %customsearch%) under Sale Order Table.
- When **Search** button is clicked, search tools available for sale and sold tools based on criteria selected and return results satisfying Tool ID, Description, Customer Name, SalePrice, and SaleDate.

```
SELECT SaleOrderID, S.ToolID, CONCAT(T.PowerSource, ' ', T.SubOption, ' ', T.SubType) AS
Description, CONCAT(Customer.FirstName, ' ', Customer.LastName) AS Customer,
ROUND((0.5 * PurchasePrice),0) AS SalePrice, ForSaleDate, C.ClerkID
FROM SaleOrder AS S
LEFT JOIN Clerk AS C
ON S.ClerkID = C.ClerkID
LEFT JOIN Tools AS T
ON T.ToolID = S.ToolID
LEFT JOIN Customer
ON S.Email = Customer.Email
WHERE T.Type = '$Type' AND T.SubOption LIKE '$CustomSearch'
```

### View Clerk Report

Abstract Code

- User click **Clerk Report** Button under **Report** in **Main Menu**,
  - Display ClerkID, Clerk.Firstname, Clerk.MiddleName, Clerk.LastName, Clerk.Email, Clerk.HireDate, NumberofPickups, NumberofDropoffs, CombinedPickupsAndDropoffs from Clerk Info and Reservation table.
- When **Reload Result** Button is clicked, re-run the query for **View Clerk Report** task.
- When **Back to Report Menu** is clicked, return to **Main Menu** Form.

```
CREATE TEMPORARY TABLE CLERK_R11
AS
(SELECT C.ClerkID, C.Firstname, C.MiddleName, C.LastName, C.Email, C.DateOfHire,
COUNT(R1.Pickup_ClerkID) AS NumberofPickup
FROM Clerk AS C
```

```

LEFT OUTER JOIN Reservation AS R1 ON C.ClerkID = R1.Pickup_ClerkID
GROUP BY C.ClerkID, C.Firstname, C.MiddleName, C.LastName, C.Email, C.DateOfHire);
CREATE TEMPORARY TABLE CLERK_R12
AS
(SELECT C.ClerkID, C.Firstname, C.MiddleName, C.LastName, C.Email, C.DateOfHire,
COUNT(R1.Pickup_ClerkID) AS NumberofPickup
FROM Clerk AS C
LEFT OUTER JOIN Reservation AS R1 ON C.ClerkID = R1.Pickup_ClerkID
GROUP BY C.ClerkID, C.Firstname, C.MiddleName, C.LastName, C.Email, C.DateOfHire);

CREATE TEMPORARY TABLE CLERK_R21
AS
(SELECT C.ClerkID, C.Firstname, C.MiddleName, C.LastName, C.Email, C.DateOfHire,
COUNT(R2.DropOff_ClerkID) AS NumberofDropoff
FROM Clerk AS C
LEFT OUTER JOIN Reservation AS R2 ON C.ClerkID = R2.Dropoff_ClerkID
GROUP BY C.ClerkID, C.Firstname, C.MiddleName, C.LastName, C.Email, C.DateOfHire);

CREATE TEMPORARY TABLE CLERK_R22
AS
(SELECT C.ClerkID, C.Firstname, C.MiddleName, C.LastName, C.Email, C.DateOfHire,
COUNT(R2.DropOff_ClerkID) AS NumberofDropoff
FROM Clerk AS C
LEFT OUTER JOIN Reservation AS R2 ON C.ClerkID = R2.Dropoff_ClerkID
GROUP BY C.ClerkID, C.Firstname, C.MiddleName, C.LastName, C.Email, C.DateOfHire);

SELECT COALESCE(a.ClerkID, CLERK_R21.ClerkID) As ClerkID, a.FirstName,
a.MiddleName, a.LastName, a.Email, a.DateofHire, a.NumberofPickup As
NumberofPickup, CLERK_R21.NumberofDropoff As NumberofDropoff,
(NumberofPickup + NumberofDropoff) AS CombinedTotal
FROM CLERK_R11 AS a
LEFT JOIN CLERK_R21 ON a.ClerkID = CLERK_R21.ClerkID
UNION
SELECT COALESCE(b.ClerkID, CLERK_R22.ClerkID) As ClerkID, b.FirstName,
b.MiddleName, b.LastName, b.Email, b.DateofHire, b.NumberofPickup As
NumberofPickup, CLERK_R22.NumberofDropoff As NumberofDropoff,
(NumberofPickup + NumberofDropoff) AS CombinedTotal
FROM CLERK_R12 AS b
RIGHT JOIN CLERK_R22 ON b.ClerkID = CLERK_R22.ClerkID
order by CombinedTotal DESC, NumberofPickup DESC, NumberofDropoff DESC;

```

## View Customer Report

### Abstract Code

- User click **Customer Report** Button under **Report** in **Main Menu**,
  - Display CustomerID, Customer.Firstname, Customer.MiddleName, Customer.LastName, Customer.Email, Customer.PrimaryPhone, Total#ofReservations, Total#ofToolsRented from Customer Info and Reservation table.
- When **Reload Result** Button is clicked, re-run the query for **View Clerk Report** task.
- When **Back to Report Menu** is clicked, return to **Main Menu** Form.

```
CREATE TEMPORARY TABLE TempCustTable
AS
(SELECT C.CustomerID, C.FirstName, C.MiddleName, C.LastName, C.Email,
CONCAT(C.AreaCode, '-', C.MainNumber, 'x', C.Extension) AS PhoneNumb,
Reservation.ReservationID, Been.ToolID
FROM Customer AS C
    INNER JOIN Reservation ON C.Email = Reservation.Email
    INNER JOIN Been ON Reservation.ReservationID = Been.ReservationID
    INNER JOIN Tools ON Been.ToolID = Tools.ToolID
WHERE Reservation.StartDate >= (CURRENT_DATE() - INTERVAL 30 DAY));

SELECT CustomerID, FirstName, MiddleName, LastName, Email, PhoneNumb,
COUNT(DISTINCT(ReservationID)) AS TotalofReservation , COUNT(ToolID) AS
TotalofToolsRented
FROM TempCustTable AS Temp
GROUP BY CustomerId, FirstName, MiddleName, LastName, Email, PhoneNumb
ORDER BY TotalofToolsRented DESC, LastName DESC
```

## View/ Filter Tool Inventory Report

### Abstract Code

- User click **Tool Inventory Report** Button under **Report** in **Main Menu**,
- User selects one of Type (**\$Type**) [All Tools, Hand Tool, Garden Tool, Ladder, Power Tool] and **Custom Search** (**\$CustomSearch** as string with syntax %customsearch%).
- When **Search** button is clicked, search tools table based on criteria selected and return results satisfying *Tool ID, Current Status, Date, Description, RentalProfit, TotalCost, and TotalProfit* from Tools, Reservations, SaleOrders, ServiceOrder Tables.

```

CREATE TEMPORARY TABLE RentalSummary
SELECT Tools.ToolID, Max(EndDate) as EndDate, Sum(.15 * PurchasePrice) AS
TotalRentalProfit
FROM Reservation
LEFT JOIN BEEN ON Reservation.ReservationID= Been.ReservationID
LEFT JOIN Tools ON Been.ToolID = Tools.ToolID
GROUP BY ToolID;

CREATE TEMPORARY TABLE ServiceSummary
SELECT Tools.ToolID, Max(EndDate) as EndDate, Sum(ServiceCost) AS TotalServiceCost
FROM ServiceOrder
LEFT JOIN Tools ON ServiceOrder.ToolID = Tools.ToolID
GROUP BY ToolID;

SELECT T.ToolID, (CASE
    WHEN SO.ToolID IS NOT NULL AND SO.SoldDate IS NOT NULL THEN 'Sold'
    WHEN SO.ToolID IS NOT NULL AND SO.SoldDate IS NULL THEN 'For Sale'
    WHEN SO.ToolID IS NULL AND NOW() > SS.EndDate AND NOW() > RS.EndDate THEN
'Available'
    WHEN SO.ToolID IS NULL AND NOW() > RS.EndDate AND NOW() < SS.EndDate THEN
'In-Repair'
    WHEN SO.ToolID IS NULL AND NOW() > SS.EndDate AND NOW() < RS.EndDate THEN
'Rented'
    ELSE 'Available'
END) AS CurrentStatus, (CASE
    WHEN SO.ToolID IS NOT NULL AND SO.SoldDate IS NOT NULL THEN SO.SoldDate
    WHEN SO.ToolID IS NOT NULL AND SO.SoldDate IS NULL THEN SO.ForSaleDate
    WHEN SO.ToolID IS NULL AND NOW() > SS.EndDate AND NOW() > RS.EndDate THEN "
    WHEN SO.ToolID IS NULL AND NOW() > RS.EndDate AND NOW() < SS.EndDate THEN
SS.EndDate
    WHEN SO.ToolID IS NULL AND NOW() > SS.EndDate AND NOW() < RS.EndDate THEN
RS.EndDate
    END) AS Date, CONCAT_WS(' ',CASE T.PowerSource WHEN 'manual' THEN '' ELSE
T.PowerSource END, T.SubOption, T.SubType) AS Description, IFNULL(RS.TotalRentalProfit,0)
AS TotalRentalProfit, (IFNULL(SS.TotalServiceCost,0) + T.PurchasePrice) AS TotalCost,
(IFNULL(RS.TotalRentalProfit,0) - IFNULL(SS.TotalServiceCost,0) - T.PurchasePrice) AS
TotalProfit
FROM Tools AS T
    LEFT JOIN RentalSummary AS RS ON T.ToolID = RS.ToolID
    LEFT JOIN ServiceSummary AS SS ON T.ToolID = SS.ToolID
    LEFT JOIN SaleOrder AS SO ON T.ToolID = SO.ToolID
    WHERE T.Type LIKE '$Type' AND T.SubOption = '$CustomSearch'

```