**SQL Query List & Descriptions**

*This document provides a listing of the SQL statements/queries used within the Work Order System application and what each accomplishes. Each heading represents a python script and will be followed by any SQL used within that script.*

## app.py

*No SQL queries found in this file.*

## Assets.py

*No SQL queries found in this file.*

## baseObject.py

*This object was not altered very much, if at all, from the original template so no “new” SQL queries used.*

## ProblemCodes.py

*No SQL queries found in this file.*

## Users.py

The “getByUsername” function uses:

**SELECT \* FROM `{self.tn}` WHERE `Username` = %s**

* Searches the Users table looking for the record(s) that matches the login username entered by the user who is attempting to log into the system.

In “tryLogin”, the following query is used:

**SELECT \* FROM `{self.tn}` WHERE `Username` = %s AND `Password` = %s;**

* Queries the User table for a match with the given username and password.

## WO.py

Within “getByReqID”:

**SELECT \* from `{self.tn}` WHERE `RequesterID` = %s ORDER BY `WorkOrderID` DESC**

* This statement looks within the WorkOrders table to find all records that have the provided RequesterID and lists the results in descending order by WorkOrderID. The results populate the Work Order List for the Requester interface.

“getAllWOs” uses the following:

**SELECT `WorkOrders`. \*, Reqs.`UserFirstName` AS RFName, Reqs.`UserLastName` AS RLName, `Assets`.`AssetTag`, Techs.`UserFirstName` AS TFName, Techs.`UserLastName` AS TLName FROM `WorkOrders`**

**LEFT JOIN `Users` Reqs ON `WorkOrders`.`RequesterID` = Reqs.`UserID`**

**LEFT JOIN `Users` Techs ON `WorkOrders`.`TechnicianID` = Techs.`UserID`**

**LEFT JOIN `Assets` ON `WorkOrders`.`AssetID` = `Assets`.`AssetID`**

**LEFT JOIN `Problems` ON `WorkOrders`.`ProblemID` = `Problems`.`ProblemID`**

**ORDER BY `WorkOrderID` DESC**

* The above SELECT statement returns the values in all of the fields of all records in the WorkOrders table in addition to the fields on the joined tables. This statement joins the WorkOrders table to the Users table two times, once on the RequesterID and once on the Technician ID, as well as the Assets table based on the AssetID, and finally the Problems table based on Problems ID. This allows the Work Order List page to show the actual values of the Requester’s full name, the Technician’s name, the Asset Tag and the Problem Code.

The “getByTechID” function calls the following query:

**SELECT \* from `{self.tn}` where `TechnicianID` = %s AND `Status` = 'Open' ORDER BY `WorkOrderID` DESC**

* This query searches within the WorkOrders table to find all records that have the provided TechnicianID and is Status is “Open”. The results are then provided back in descending order by WorkOrderID. The results populate the Work Order List for the Technician interface.

In “getByWOID”, the below statement is used:

**SELECT \* FROM `{self.tn}` WHERE `WorkOrderID` = %s**

* The above statement searches the WorkOrder table for the provided WorkOrderID.

Within the “getWOFKs” definition, the following queries can be found:

**SELECT `UserID`,`UserFirstName` AS `RFName`, `UserLastName` AS `RLName` FROM `Users` WHERE `UserID` = %s**

* This statement searches the Users table looking for the UserID within the argument passed to the function and returns the UserFirstName and UserLastName, as aliases, for each record found. This is used to get a Work Order’s Requester information.

**SELECT `UserID`,`UserFirstName` AS `TFName`, `UserLastName` AS `TLName` FROM `Users` WHERE `UserID` = %s**

* This query performs the same function as the previous one except for the Technician.

**SELECT `AssetID` AS EqID,`AssetTag` AS `Asset`, `AssetType` FROM `Assets` WHERE `AssetID` = %s**

* The above query connects to the Assets table and returns the AssetTag and AssetType for the AssetID provided.

**SELECT `ProblemID`,`ProblemDesc` AS `ProbDesc`, `ProblemCode` AS `PCode` FROM `Problems` WHERE `ProblemID` = %s**

* This statement steps through the Problems table looking for the value provided within the ProblemID field and returns the ProblemDesc and ProblemCode field as aliases. Aliases are used here to make the reference to the values on the resulting records easier.

## WOComm.py

In “getByWOCommID”, the below statement is used:

**SELECT \* FROM `{self.tn}` WHERE `WOCommID` = %s**

* The above statement searches the WorkOrderComms table for the provided WOCommID.

The “getCommsByWOID” function calls the following query:

**SELECT \* FROM `{self.tn}` WHERE `WkOrdID` = %s**

* This query searches within the WorkOrderComms table to find all records that have the provided Work Order ID and returns all of the fields. It is used to populate the Work Order Communications List page.

“getWOCommsFKs” uses the following:

**SELECT `UserID`,`UserFirstName` AS `FName`, `UserLastName` AS `LName` FROM `Users` WHERE `UserID` = %s**

* The above statement queries the Users table and returns the UsersFirstName and UserLastName, as aliases for coding efficiency, for the record(s) that has the given value in the UserID field.