**CS3431-A22: Project Description**

**Building a Database Application**

**Phase 2: Views, Triggers, and Procedures**

**Due Date:** Sa 10/1 at 11:59pm (same time as Phase 3). No late submissions because solutions will be posted

immediately so students can study for exam 2.

**Teams:** The project is to be done in the same teams of three as previously.

**Submission:** The SQL code should be in a file named p2.sql and NOT include the SQL commands from p2start.sql. Make sure to include all of your names in a comment line at the top of the project submission. Only one student is to submit the assignment so be certain to determine who that will be in advance and each team member is responsible for confirming that it was submitted properly and on time.

**Description:**

For this assignment, the solutions to Project Phase 1 have been provided as a starting point. Run the p2start.sql file to create and populate the tables for this project phase.

1. You will create a view and then use it in a query.
2. Create a view named **TransportAndLabRequests** that lists the existing patient transport and lab requests but not the medical equipment requests. The view displays the service requestID, the role, the status, the requester employee ID, the handler employeeID, the destination floor, the destination short name, the location type, the transportation patient ID with the heading of TransportPatient, the transportation itemID as TransportVehicle, the lab patient as LabPatient, and the patient lab test. Sort by request ID. Hint: Using a regular join will result in an empty table. The resulting table should look like the Employee table where there are a lot of nulls.
3. Create a query that uses the TransportAndLabRequests to show the count of patient lab requests, and the count of patient transport requests grouped by status. Sort the results by status.
4. Create a procedure named **EmployeeLocation** that takes an employee’s firstName and lastName as input parameters and using dbms\_output.put\_line(), displays the employee’s name, the short name of the location where they are, the employeeID, and their username. First make sure to turn on the server output using the following command:  
   **set serveroutput on;**

As an example, your procedure will display the following text in the specified format for the employee Joselyn Klein:

**Joselyn Klein located in Sharf Center. EmployeeID: 170. Username: JKlein.**

Create the following triggers (do not use views in your solutions):

1. Create a trigger named **TransportationEquipment** for new internal patient transportation requests that requires the medical equipment to be either a wheel chair or a recliner. If it is not, then the following error message is displayed.

ERROR: The equipment type for the transportation request is not a recliner or wheelchair!

1. Create a trigger named **UniqueNPI** that prevents inserting a new employee with an existing NPI number. If the NPI number already is in the system, then display the following error message:  
   ERROR: New doctor's NPI number already exists in the system
2. Create a trigger named **DefaultXRayLocation**. When a new lab request is created (assume the corresponding service request has already been entered), if it is an XRay test and the location type is not a lab, then change the service request destination to be the outpatient fluoroscopy lab, LABS002L1.

In a separate SQL file, insert, update, and delete records to test the triggers above. You will NOT submit this SQL file – it is strictly for your testing purposes.