***University of Pittsburgh***

***School of Information Sciences***

***Graduate Information Science & Technology Program***

**INFSCI 2710: Database Management**

**Fall 2019**

**PROJECT: Database System for Customer Support**

**Purpose of the project**

For this project, you will analyze the requirements, design, implement, document and demonstrate a database system that could be used by a help desk staff to ensure customer issues are resolved.

**Application Requirements**

You will create a catalog of products (5-10) that your company provides. Your system must maintain a record of customers and the products that each customer owns. As each product is purchased by a customer, the database should be updated to reflect what they own. You must keep track of customers, including their personal information, and purchase history.  You should keep track of employees that work at your company. These employees will be the users of your software. They will open cases to support your products and see them through to resolutions

All projects are required to demonstrate *at least* the following functionality:

***Data***

* **Customers:**  name, address (street, city, state, zip code), company gross annual income, etc.
* **Products:** name, description
* **Salespersons:** name, address, e-mail, job title.
* **Employee:** name, address phone, email
* **Case:** summary, description, status (open or closed), employee the case is assigned to, product the case is about.
* **Case comments:** Over time, employees must be able to comment on cases. Time and comment should be recorded.
* **Resolution:** common resolutions, resolution name, steps to resolve, product(s) the resolution applies to.

***Operations***

* **Employee Browsing**Employees must be able to search the database for particular items based on various attributes and must also be able to do browsing (i.e., less focused searching). Examples:
  + Search for cases by status and timeframe
  + Search for a customer and view products the customer has purchased
  + Search for common resolutions by product
* **Update Transactions**The system must be able to handle creation of cases, updates to cases, closing of cases, adding sales and customers, updating customers, adding resolutions
* **Error Checking**The system must be robust and support various application-dependent integrity constraints. For example, a customer should not be able to request support on a product that they have not purchased
* **Data Aggregation**The systemmust provide data aggregation queries:
  + Support cases by product (ordered by largest)
  + Support cases closed by employee (ordered by largest)
  + Support cases by customer (ordered by largest)
  + Support cases by customer’s company (ordered by largest)
  + Other interesting aggregate queries that you will come up with.

**Rules of the game**

* **Implementation Tools:** All projects are expected to be runnable from a web browser at the request of the instructor. SQL is a requirement. A group may choose to use any database systems and front-end implementation tools after discussing it with instructor.
* **Additional Requirements:** The project must represent a fairly sophisticated database application. In particular, the database must contain multiple (e.g., at least seven) relations, and the database design must include indexes, primary keys, etc. (Follow the best practices that we have discussed in class)
* **Groups:** The project is to be done in groups of 3 students. A roster for each group must be submitted to the instructor by the date specified in the ``Due Dates'' section of this assignment. The groups are ``self-policing'' (e.g., each group is responsible for its own division of labor, scheduling, etc.). *Note: If an unreconcilable problem arises in your group, it is your responsibility to contact the instructor as soon as possible. After the project is due, it will be too late.*
* **Assumptions:** In cases where the above description of the application is incomplete, it is acceptable to make assumptions about the application providing that: 1) they are explicitly stated in the final report, 2) they don't conflict with any of the requirements specified above, and 3) they are "reasonable". If you have a question about the acceptability of any of your assumptions, check with the instructor. Interesting questions should be raised in class. The instructor will post interesting questions and answers on Blackboard.
* **Report:** A final report should be handed in for grading at the end of the term. The report must be formatted in a reasonable manner (i.e., using a text processor and a decent printer). The final report is due during class on the "Project Due" date specified in the class schedule.
* **Implementation:** The project requires a working implementation of the system to be built, tested, and demonstrated. A large part of the project grade depends on the quality of this implementation.

**Report Requirements**

*The final report must contain:*

1. A short overview of the system including identification of the various types of users, administrators, etc. who will be accessing the system in various ways.
2. A list of assumptions that you have made about the system.
3. A graphical schema of the database using the E-R diagram with a short description of each entity set, relationship set and their corresponding attributes.
4. A set of relational schema resulting from the E-R diagram with identification of primary and foreign keys.
5. The DDL statements to create the relational schema in some appropriate Normal Form, with identification and justification of the Normal Form.
6. A description of your front-end design as well as the front-end to back-end connection.
7. A brief overview of the system implementation with example screen shots.
8. A description of your testing efforts and erroneous cases that your system can detect and handle.
9. A description of the system's limitations and the possibilities for improvements.

**In addition, a demo of the working system will be required. All members of the group must attend this demo, and must be prepared to explain and demonstrate those aspects of the project for which they were responsible. The source code for the project should be available on-line during the demonstration.**

**Demos must be scheduled prior to the final exam. Options are prior to class or after class. Expect 15-30 minutes. Schedule early…you may need to do more than one presentation.**