

CIS 9340
(CIS Department Laptop Loaning System)
Final Project Report

Submitted by:-
Chani Thakkar
Pooja Vivek Vichare
Vansha Dower

Project Description:-

CIS department facilitates a laptop loan program for professors and students of their department. Every time a professor needs laptop for the students in his/her class, they have to raise a request. Presently, the entire process of loaning laptop is handled in an adhoc manner by using google drive (for laptop reservations), emails/ verbal discussions for communicating system issues to BCTC and Google sheets to manually assign laptop requests to graduate assistants. Separate sheet is maintained to track hours worked by graduate assistants. The whole process requires lot of paperwork and manual intervention without any integrated loaning and communication system

Existing System Description

The CIS department loans the laptops with all the required software installed, in order to help the students, get an easy access to the laptops for the class duration. At present, these programs are managed in an entirely ad hoc fashion consisting a mixture of shared Google Drive folders, miscellaneous spreadsheets, various incompatible web forms, e-mail chains, and even pen & paper records. If a student requires a laptop for a CIS course, he/she request for the same to their respective professors who then inputs the request in a google form. This form is then accessed by the graduate assistants who manually enter the requests into an excel sheet to be able to track the loaning requests. If a professor wishes to cancel the request made earlier then, he will have to intimate either the department Professors who manages the process i.e. Professor Kannan Mohan or intimate the same to the Graduate Assistants via email. All loan scheduling, laptop incident reports, mailing lists, etc. is carried out manually by both faculty and graduate students without any provision for tracking data related to laptop usage, laptop inventory, graduate assistant hours, or seminar participation. Being a manual process, it is prone to errors and miscommunications leading to unnecessary chaos and time consumption. Moreover, there was no way to accurately collect data from any of the programs, as doing so was entirely dependent on the assistants' manual records.

Existing System Problems

1) Lack of standardization:

As mentioned previously, all tasks are carried out ad hoc using a wide variety of electronic and paper forms. The lack of any unified structure or information system to manage this data makes the process needlessly time consuming and prone to error.

2) Inability to accurately track usage:

As a consequence of the lack of standardization, data regarding program activity cannot be accurately recorded or tracked. One cannot easily access data visualizations via a dashboard or similar interface.

3) Difficulty tracking budget / hours:

The program runs on a limited budget, and all scheduling and hour tracking must be performed manually by the administrators and assistants. If someone cancels or changes hours, everything must be updated by hand, often in multiple places. This makes it difficult to plan projects, as the amount of budgeted hours being consumed is difficult to track.

4) Limited communication:

There currently exists a limited, and again entirely manual, chain of communication between the assistants, program administrators, and Baruch IT. For example, when laptops have problems, this must be communicated manually to both the administrators and Baruch IT. For seminars, mass mailing of notifications must be conducted by the person running the seminar, as each person utilizes a different method of advertisement (Google page, blog page, etc.). Similarly, our ability to inform students of the availability of the department computer lab was limited due to lack of a notification system.

Group's Experience (Easiest/Hardest part of the project):

- From the entire project experience the easiest step was to select the topic of the project. Initially we thought that it would be easy to work with this project topic as all the entities were clearly defined.
- However, once we started working on the project, things started getting more complicated. It got even more complex when we created tables, inserted values in it. At this stage, we uncovered various conflicts between the tables and had to sit and resolve them. Once this was done, it was much easy to create queries.

What you learned that you did not expect to

- We didn't expect to have so many issues while inserting Primary key and Foreign Key in all the tables. But while working on the project we came across various issues and we resolved it successfully.

What you would have done differently

- In this project we divided the work and when we created tables we had issue while inserting data. We should have assigned that to 1 person and other person should have inserted data in the schema and the third person should have created queries. This would have made the task easier for all of us.

Benefits of the proposed system

The goals of the new system are to replace all the manual methods described above with a completely automated and integrated web application system. The web application will offer the following features:

- A **fully automated scheduling system** for laptop loans and graduate assistants consisting of web forms connected to a database, supported by application logic for accurately constructing and updating a schedule. Cancellations, changes, etc. will be handled internally by the

application logic, and assistant hours will be accurately tracked.

- **User-based authentication** for administrators and graduate assistants backed by profiles. Administrators will have privileges to update schedules, view hours, usage, etc. Assistants will be able to see at a glance their work responsibilities.
- An **incident reporting system** for the laptop loan program. Users will be able to report faults and have those reports automatically mailed to BCTC support for rectification.

Final comments

- It was great learning experience for all of us both as individually and as a group.
- We created a whole database from scratch and learned throughout the database development life cycle about the various issues that are faced by person who actually works in the kind of job profile.
- Starting from the ER models, relational model, identifying tables and attributes, normalizing the tables till 3NF, creating tables in access, creating a relationship diagram and applying queries to retrieve data from the database helped us gain practical knowledge in the field of Database Management Systems.

SQL Queries

For *each normalized relation*:

- we have created tables using **SQL CREATE TABLE** statement.
- written separate **ALTER TABLE** statements to add PRIMARY KEY and FOREIGN KEY constraints to the tables.
- Supplied Data for each table by writing **SQL INSERT** statements.
- We have also created various SQL queries using **SQL DDL statements** (JOIN, SUB QUERY, etc.).

List of Queries:

1. List all the items in ComplaintDetails table.
 - This is a simple query to list down all the attributes of the table
 - Here, we have used ‘*’ to list all the details.
2. List ComplaintID, GAEmpID and LaptopBarcodeNo for all the items where status is “open” and description is “hardware issue”.
 - Here, we have used Join to combine two tables (ComplaintDetails and ComplaintStatus) to get the ComplaintID, GAEmpID and LaptopBarcodeNo for all those laptop complaints having hardware issue.
 - LaptopBarcodeNo is common in both the tables and hence we have joined the table based on this attribute.
3. List full details of Graduate Assistants.
 - This is a simple query to list down all the attributes of the table GraduateAssistant.
4. How many graduate assistants are there in the department.
 - Used aggregate function, we have counted the total graduate assistants there in the CIS Department working for the laptop loaning system.

5. Provide the phone number and GAEmpID of GA named “Pooja”
 - In this query, we have connected two tables **PhoneDetails** and **GraduateAssistant**
 - The common attribute which joins both tables is GAEmpID
 - The query returns the phone number of the GA with FirstName “Pooja”
6. Display GAEmpID,LaptopBarcodeNo,GFirstName of all the GA’s whose laptop complaint status is ‘open’.
 - In this query, we have connected two tables ComplaintDetails and GraduateAssistant
 - We have used SubQuery to solve this query
 - The common attribute here is GAEmpID, which joins both the tables.
7. Subquery: List all the details from GraduateAssistant table where the NumberOfHoursWorked of the Graduate Assistants is greater than 20.
 - Used SubQuery to solve this query
 - Two tables used here- GraduateAssistant and PhoneDetails to display all the attributes of those Graduate Assistants whose TotalNoOfHoursWorked is more than 20.
8. Display LaptopBarcodeNo of all the complaints which are filed between 11/20/2017 to 11/25/2017.
 - Used BETWEEN to display LaptopBarcodeNo of all the complaints which are filed between 11/20/2017 to 11/25/2017.
9. Display all the professors that requested for laptops.(subquery)
 - Used SubQuery to solve this query
 - Used RequestDetails table to retrieve all the details and Professor table in the outer query to display details of professors to display only those professor details that has requested for laptops in the requestdetails table

10. Count the number of request generated by professor and the average number of laptop required (use aggregate functions)
- Used aggregate function to count the number of request and average number of laptops required by each request from RequirementsDetails Table.
11. Display Professor details and NoOfLaptopRequired details for requests that requires ToolRequirements as "Tableau".(Join three tables)
- Used Join to combine three tables Professor, RequirementsDetails and RequestDetails
 - All the three tables are joined based on the professor ID and RequestID common in them.
 - We even have filtered requests that are only for "Tableau" as toolrequirements.
12. Display list of all request IDs where GAFirstName='Pooja'.
- Here we have used Subquery
 - Displayed RequestID from RequestDetails table and used an inner query on graduate assistant table where firstname= 'Pooja'
 - GraduateAssistant table is used in the inner query and RequestDetails is used for the outer query computation.
13. List the RequestID,Date, Time and Location of requestID's where the professor ID= 'P1001'
- Tables used: RequestDetails, RequirementsDetail.
- Here we have used Join on two table RequestDetails and RequirementDetails.
 - RequestID is common among the two tables.
 - Both the tables are joined and we have use AND condition where only those request that are generated by professorID= 'P1001' will be retrieved from the database.
14. List all the request ids, request date, location for which the no.of laptop requirements were greater than 25 and toolrequirement='Django'.

- This query used 2 tables joined based on the RequestID field.
 - The query retrieves rows from both the tables where NumberOfLaptopsRequired is greater than 30 and ToolRequirement is 'Django'.
15. List the Professor ID and the count of number of requests generated by each professor.
- This query uses an aggregate function for calculating the number of requestID generated by each professor.
 - We join two tables Professor and RequestDetails based on the ProfessorID and use a GROUP BY clause on professorID
16. List the GAEmpID, FirstName ,LastName and the total number of request assigned to that GA, order by firstname.
- This query uses Join on RequestDetails and GraduateAssistant tables joined on GAEmpID.
 - Uses Count function to display the total requests assigned to each Graduate assistant
 - GROUP BY GaEmpID, GFirstName and GLastName and ORDER BY firstname.
17. Display a table listing all Employees working at BCTC. Include their EmployeeID and Email address.
- Used a simple query to get a list of the Email address for all BCTC Employees.
18. Display a table with phone numbers of Graduate Assistants along with their GAEmpID for those who have worked for more than 30 hours.
- Used Operator 'Greater Than' to extract data of all Graduate Assistants who have worked for more than 30 hours from table PhoneDetails.
19. Display the total and average number of hours of the Graduate Assistant AS TotalPaidHoursGA and AvgPaidHoursGA respectively.
- Used the SUM function and AVERAGE function to get hours data for payment made to Graduate Assistants from table .

20. Display the total number of Employees currently working in BCTC Department.
- Used COUNT Operator to add up all the BCTCEmpID to get the number of Employees working within the BCTC Department.
21. Display all the BCTC Employees whose First Name starts with 'H'.
- Used 'LIKE' Operator to get details of BCTC Employees with a particular description
22. Display a table consisting of complaint, LaptopBarcodeNo. Also, add EmailID of the BCTC Employee to whom the issue is assigned.
- Used INNER JOIN on ComplaintDetails table and BCTCEmployeeDetails table to display EmailID of BCTC Employee corresponding to the laptop & complaint assigned to them
23. Display a table consisting of BCTCEmpID and BEmail for complaints having the letter 'ware' in them
- Used Sub-query to display BEmail and BCTCEmpID of only those BCTC Employees (using table BCTCEmployeeDetails) who have been assigned issues with Description containing the word 'ware' (using table ComplaintDetails)
24. Display a table consisting Graduate Assistant's first and last name along with their phone number for only those Graduate Assistants who have worked less than 35 hours.
- Used JOIN to display Graduate Assistant details from Graduate Assistant table and Number of Hours worked from Phone Details table. Further, used WHERE clause to sort Graduate Assistants

Relationship Diagram:-

