Database Design for Employee Health Check

GROUP 2 MEMBERS:

- *UMAKANTH SAI BALGURI*
- *NYNALI GOPU,*
- *SRI VENKATA VENU GOPAL GUDDATI,*
- *AISHWARYAA MEKALA,*
- *AKSHITHA REDDY MALLIPEDDI,*
- *DHEERAJ SAI CHAVA*

Overview

- Due to the recent emergence of COVID pandemic, there has been a need to immediate changes in the way organizations operate.

- The proposed database design is a copy of "**USF Archivum Return to Campus**" with minor tweaks to ensure all the data required for analysis is gathered.

- Project aim intends to collect campus visit details, checks for various COVID symptoms and finally issues a pass if there is no risk associated with the individual's health condition.

Database ER - Diagram

Design Assumptions

- The design was created using USF Archivum Return to Campus application questionnaire, currently being used to collect COVID symptom check.
- Additionally, few attributes have been added to achieve data integrity.
- Assumptions o Every Student or Faculty member is identified by a unique number (U-number)
- Every Student or Faculty member is identified by a unique number (U-number)
- USF holds 3 campuses, every member is tagged to one of those as a base campus
- Pass will only be issued if the individual is visiting campus and has no Covid related symptoms.
- An individual can visit multiple locations on a given day and he will be issued passes to all the selected locations based on his selections.
- All the visit locations will be tagged to one web form submission (record_id), which individual is expected to fill only once per day.
- A check constraint hasn’t been used on external test location to provide flexibility to input test locations across the world. (In case of incoming International students)
- Pass Issue Y/N is being separately tracked to make the design future proof. (If pass was meant to be issued for mild COVID symptoms.
- Pass QR code will be a unumber based uniquely generated string.

Data Generation and Loading:

- Excel was used to initially create data
- Names, addresses were copied from the csv files downloaded from Kaggle.com
- We used online random timestamp generator for generating date and time
- In excel, Random functions, String concatenation were used to generate derived fields. Eg: Email addresses
- Note that the formulas were finally removed (using paste values) to avoid issues wile loading the data to SQL.
- Row counts-
  - Umember-5k
  - Visit-15k
  - Visit_location-8k
  - Cdeclaration-8k
  - Csdeclaration-2k
  - Cexternaltest-3k
  - Ctestlocation-2k
  - Pass-8k
  - Pass_qrcode-2.5k
  - Campus- 3 records
  - Building-12 records
  - Classroom- 29 records
- Data was uploaded to Oracle SQL using “Data Impot” wizard
- Note that all the records mentioned above have been loaded to the group database account.

How can this Database help?

On a high level, his database can be queried to obtain following information:

- ✓ Students enrolled in each campus
- ✓ Student campus visit information
- ✓ Student health information
- ✓ Faculty campus visit information
- ✓ Faculty health information
- ✓ COVID testing centers

Performance Tuning:

Below are the experiments done

Indexing:

Indexes are associated with tables and are used to retrieve data from the database more quickly than otherwiseand are used to speed up searches/queries. Indexes are one of many means of reducing disk I/O and improve query performance.

We created a duplicate table for “VISIT”. We checked with index and as there was no index found, we executed one query with out index. Later we created index and executed same query again after creating index. From results, we observed that Indexing helps in optimizing query execution time and improves performance.

Additionally, we explored what execution plan is and selectivity of index is as per our database.

Point Queries:

Point queries are highly targeted, returning a very small answer set (or even a single row).

Experiment: In the COVID Symptom Database, we are trying to retrieve a member whose first name is AGNES and who is a FACULTY. In the database, there is only one person with that specific value.

Part 1) All members with Name = Agnes, we got 34 of them
Part 2) We now trying to retrieve member whose Name is ‘Agnes’ and role is ‘Faculty’
From the execution plan, we observed that no index is not accessed while performing this task.
Part 3) Similarly, the single row result set is found by using the primary key (UMEMBER_ID). We can see that this query benefitted from index that is available since this is a primary key column.
From results, we observed that there is an improvement(***cost and performance***) when we use the column UNUMBER_ID (which has an Index because it is a Primary Key).

Performance Tuning(cont..)

Parallelism:

Parallel computing aims at improving performance by applying multiple CPU and I/O resources to the execution of SQL statements. It decomposes tasks into simpler sub tasks thereby increasing throughput for a group of transactions and response time for a single transaction.

- We disabled Parallelism and generated Explain Plan for the Query
- Enabled Parallelism. The degree of parallelism (DOP) is the number of parallel execution servers associated with a single operation. It was set to 3 and generated Explain Plan for the same query.
- Observed that parallelism indeed helped in optimizing the executing time of the query and the cost associated with it.

Performance Tuning(cont..)

Optimizer Modes:

Rule Based Approach – Now obsolete and unsupported but still allowed. Always uses rule-based optimization. Oracle uses heuristics from the data dictionary in order to determine the most effective way to service an Oracle query and translate the declarative SQL command into an actual navigation plan to extract the data. In many pre-Oracle8i systems, rule-based optimization is faster than cost-based.

Rule-based shortcomings – Often chooses the wrong index to retrieve rows

Cost Based Approach – This is a cost-based optimizer mode that ensures that the overall query time is minimized, even if it takes longer to receive the first row. This usually involves choosing a parallel full table scan over a full index scan. Because the ALL_ROWS mode favors full table scans, the ALL_ROWS mode is best suited for batch-oriented queries where intermediate rows are not required for viewing.

Cost-based shortcomings – Often performs unnecessary full tables scans, especially when more than three tables are being joined.

Data Visualization

As Data visualization tools such as Tableau, Microsoft Power BI help us visualize the data available in different graphical representations which can be used to answer key questions. We aim to create following visualizations for our COVID symptom database (will be added during the final submission):

- A line chart with number of cases every day
- A bar graph of number of students who have had symptoms or came into contact with people with virus or number of people who are getting passes with respect to campuses
- Card visualizations to get the numerical data
- Bar graph to represent number of test by testing centers
- A visualization to plot the numbers against each symptom

Count of COVID_Tested by Campus_Visiting

Count of COVID_Tested by Pass Issue

Count of COVID_Tested by mrole

Count of COVID_Tested by Day

Count of COVID_Tested by symptom_headache

Count of Contact_With_Positive by Pass Issue

Count of Pass Issue by Campus_Visiting

Total number of test locations

187

Count of Test_Location

Conclusion:

We leveraged the skills gained in the ADBMS class and applied them to create a database project to store, process and provide information about students and faculty health information that could help in key decision making.

THANK YOU