

Database systems final



What are the issues in the normalization vs.. Performance trade-off? Discuss. Your essay should (1) define and describe the characteristics of normalization, (2) define and describe the characteristics of performance, and (3) define and describe what is meant by " performance trade-off, (4) summarize your essay with recommendations and/or guidelines regarding use of normalization In database design.

NAS: When we take a close look in designing a database one may not be thinking of normalization and performance, we may just be thinking, " I have this major project u in several months, I do not have time to be considering about formal techniques and performance, I will worry about this later. " Though this may occur In industry, database designers need to keep In mind that utilization of scarce resources such as processor time, network utilization, memory, etc. , are very critical and can have a significant impact on database access and retrieval.

As I had previously mentioned in our forum, one of the main concerns to keep in mind is that in other fields being 2 or 3 times slower may not be noticeable and certainly may not have a long term affect UT in the realm of computers, having a multiplier can be detrimental In our resultant. What does this mean? It would simply mean that 1) Your database Is not consistent OR 2) Your database Is lacking the performance. Finding the equilibrium in this case can be a task like working with two equations setting them equal to each other for all else to be equal.

What is Normalization? Normalization as stated in our text is, " A technique for producing a set of relations with desirable properties, given the data

requirements of an enterprise. " What does this mean? It means that normalization is a process and it is often to perform a sequence of tests to see whether it satisfies or violates the requirements of a given normal form. The five normal forms that we are mainly concerned about would be 1NF - 5NF. These normal forms in database design are based on practical/functional dependencies among the attributes of a relation.

These are the stronger forms and are known as the Boyce-Codd Normal form. As you see that we have 1NF - 5NF, note that the 1NF and 5NF are often used in very rare conditions. With all of this said, we can conclude that normalization's main objectives are to identify relations based on primary and candidate keys and their functional dependency. It also aids database designers by presenting a sequence of steps to help reduce data redundancy and to avoid update anomalies. The anomalies that we consider are insertion, deletion and modification.

In these categories a unique situation can arise where data was not inserted, deleted and/or removed or not modified. In any case, these situations can have an adverse effect on our database and/or remove any of these consistencies that arise when database designers do not take the time to properly design their database due to lack of time. Performance in the database perspective can be defined as the effectiveness and speed in a database performs a transaction. Of course this is going to be dependent as to how this database system will make use of its scarce resources.

As mentioned in our text some of the variables that we should be concerned with is 1) the access time (I/O) cost involved in accessing the physical data

on the disk, 2) the CPU time cost incurred when performing operation on data in main memory and 3) the immunization cost associated with the transmission of data across the network. If you do not have your hardware setup properly either it be in a centralized system or distributed environment, it may not perform to your necessary standards in spite of your design. You need to have both equations of normalization and performance equal to each other in order to find the optimal local Max.

The normalization vs.. Performance tradeoff can be quite of a debate. If we were to look at this methodically, without normalization, the bulk of our database will and can come convoluted to the point that no matter how well your centralized or distributed computing setup performs, you will have see a detrimental hit in performance. In either case, database designers need to carefully take into account all variables and weigh the advantages as well as the disadvantages. Taking in account as to how the database is going to be used is also a major consideration.

Another point to take into account is the foreseeable future and the growth. Taking into account these variables can alleviate the pain that you can encounter later. The interpretation of this matter is that normalization and performance work hand in hand. If you are a database designer, you need to weigh the advantages of using INFO -INFO or not. As we go through our endeavors of database design we should always consider the whole picture, the final project. Having a clear perspective at the end resultant can help our decision making process of to use normalization or not.

We should always keep in mind the value of fully normalized model. It is a valuable technique which: 1) Minimizes redundancy in a database. 2) Virtually no inconsistencies in database. 3) Removes redundancies from relational data models. 4) Normalization also eliminates the dependency of one fact on another. 5) It assists in the discovery and clarification of business rules. As you see here, this technique is a tool to be used to your advantage and before undertaking a project, when the overall picture is understood, you can definitely rely on any given level of normalization to assist in your design. An essay should (1) define and describe the characteristics of a database management system, (2) define and describe the characteristics of a file management system, (3) present the similarities between a DB'S and a file management system, (4) present the differences between a DB'S and a file management system, (5) present a problem scenario that is best solved using a DB'S, (6) present a problem scenario that is best solved using a file management system. A database management system (DB'S) is a collection of interrelated data and a set of programs to access information.

It allows the storage, retrieval, and manipulation of information in a prescribed format. In the purest formal definition, a DB'S does not allow for unformatted data. By having this restriction, this allows for quick indexing, sorting and other data processing procedures. Some of the most common DB'S applications that are commonly used today include Access, Oracle, DUB and SQL. These popular applications offer the same features as would a file based system. They both are used to retrieve and store information.

The fundamental difference between the two would be that a DB'S provides the user data integrity and security. Another difference that is obvious is that

a DB'S can be used in a variety of settings such as a centralized environment or a distributed one. In either case, the database can have concurrent data access by several users which our counter part does not. The uses of DB'S has grown tremendously to the point where even ASS at one point was known for statistical analysis also has a functional use of a database management system.

With this one example, you can see that the popularity of these applications have increased due to their flexibility and power and supply is creating its own demand. In the long-run situation, If you are considering a DB'S or a file based system, the initial costs of your DB'S are relatively high, but again, your benefit will be greater. Below are some examples as to where a database management system should be implemented. As you will see, in these examples, you will have data being accessed simultaneously and require security and data integrity. 1) Airline - reservations, flights, etc..) Universities - student enrollment information, grades, etc.,. 3) Newspaper agencies - customer payment, subscriptions A file based management system is the predecessor of database systems. Fact of the matter is that a file system is a collection of programs, each entity manages and defines their own data. A file based management system was an early attempt to imputer the manual filing system that existed and in certain businesses still exists. Of course you have to keep in mind that a file based management system works best for small amounts of data. When we examine a DB'S vs.. File based management system, we do find some similarities: 1) Both of them store and retrieve data 2) Both of them have fast streaming I/O, of course let's hope the DB'S is normalized. 3) It is easy to add or delete data.

4) Serves its purpose commensurate with its need. So the question remains, if these two have the same basic purpose, why so different? The key point mentioned above is, " purpose. " A file based system would be best when there is not much data. Having a vast amount of data while using this type of file system can cause problems that can lead to data loss.

So, having this in mind, where would you be able to implement this type. That is simple, to answer based on personal experience. A friend of mine, several years ago opened up his insurance office. In his business he had one employee and not many customers. After several years in business, he began to realize how quickly his computer system became slow specially when he was trying to locate customer accounts. He had the old filing system, in cabinets, in several locations within his office. That was just one issue, now he had several employees and all with different filing systems.

Before he knew it was chaos just to retrieve a single bit of information here you see a classic example of a file based system served its purpose for a while but as needs become increasingly detrimental for the business, you cannot be adamant to change. Create Has *.. 1 Member of Stay Home Videos ERE Diagram Assigned Becomes * Rents Rent 0.. * 1. 1 Provides . Formulate the SQL queries below using the following: Staff Skill(skill_code, description, charge_out_rate) Booking(staff_no, project_no, date_worked, hours_worked) Where: Staff contains details and staff_no is the key.

Skill contains descriptions of skill codes (programmer, analyst, manager, etc) and the charge_out_rate per hour for that skill; the key is skill_code Project contains project details and project_no is the key Booking contains details of

the date and the number of hours that a member of staff worked on a project and the key is Formulate the following queries using SQL: a) For all projects that were active in July, 2003, list the staff name, project number, and the date and number of hours worked on the project, ordered by staff name, within staff name by project number and within project number by date. B) List all staff with a charge out rate greater than the average charge out rate.