Why analyse data?

Business

Records may be fixed length (padded with spaces) or variable length (e. g.

comma delimited). As we move to more complex applications, the type of data we work with expands. Eventually the flat file or single table becomes unsuitable. There may be many users that need to get access to the data for view or edit. Having more that one user updating the file at once causes problems.

Different areas of the program may require access to different parts of the data. If the data is organised, smaller amounts need to be open for update. This improves accessibility and therefore performance. Speed of access reduces with increasing record count and record length. Often there are a number of similar items on a record e.

g. exam results against a pupil on a student record (how many subjects, how many papers, how many slots do you need?). Organising you data in to different files or tables allows greater flexibility. Organising your data can be a difficult process, particularly with large systems with hundreds of tables. There are methods of analysis, which help us to avoid these problems. ConsistencyEach data item should be stored only once to ensure consistency of data.

If an item of data is stored in several locations, it may be that not all of them are updated when the data changes. This causes inconsistency. Redundancy of dataRedundancy is the repetition of the same pieces of data on many records. An example of this can be seen on a basic order form. The customer’s name and address is often requested. If the customer places orders regularly, the name and address would be entered on each form.

If the forms are keyed in to a database, the same address is entered time and time again. Problems with redundancy: -1. Causes excess work for the data entry clerk2. Takes up excess space in the data store3. Causes problems when updating the data e. g.

when the customer moves need to update all of their records. IndependenceCompanies often have many different computer applications each with its own data store. Analysing the business and its data across the board results in a data model, which is machine and program independent. Altering the data structure by adding a field should not effect the program. This independence allows for growth both of the type of data included and the range applications that can be used on it. MaintainabilityPrograms, once written, seldom remain the same.

They are always being upgrade to include more functionality. If the data structure is correct, it is easier to make changes to the program to include the new processes. In some cases, the data structure limits this. SecurityIt may be that some data within a database is available for only a few to see and fewer to edit. During data analysis, this would be taken in to consideration. The resulting data structure may separate data in to different files or tables to allow for this security problem.

Data Analysis is a very important part of any software life cycle. The general rule of thumb is that the further down the software life cycle you go before finding errors, the more those errors cost. Data analysis helps to iron out errors at an early stage. Exercise 1Library exampleBelow is an example of

data that could be used by a library. In a library we have books, customers and loans.

NB there are errors in this to demonstrate the techniques. Open up MS Access. Create a new database called library. Create each of these tables. The field in Bold should be declared as the Key for each table. ObjectFieldCustomerBorrower IDNameAddressContact NumberBook LimitBook StatusRecord LimitRecord StatusVideo LimitVideo StatusCD LimitCD StatusDVD LimitDVD StatusAccount StatusBookISBN CodeTitleAuthorPublisherLoan PeriodLoanLoan IDISBN CodeBorrower IDBorrower nameDate outDate due backFine DueTimes RenewedTop Down Data AnalysisAs in all areas of IT, the processes of analysis are quite simple but are wrapped in jargon.

Below is a brief explanation to introduce you to the key words. In ‘ Top Down’ analysis we look at Entity Relationship Modelling followed by Data Normalisation. Entity Relationship ModellingThe data can be organised into a series of objects called ENTITIES. Entities are generally ‘ THINGS’ such as ‘ Customer’; ‘ Book’ that can be uniquely identified. Entities are implemented as tables. The entities have ATTRIBUTES (fields).

Attributes are the data types that are specific to the entity. In the library example, the entity CUSTOMER has ATTRIBUTES of ID, Name, address, contact number, limit and status. Attributes have DOMAINS. The domain of the attribute (field) is the set of values that are contained in the attribute across all the records. Looking at the entity ‘ Customer’ in the library example, the attribute ‘ Name’ would have a domain containing the names

of all the customers and Status would have a domain of containing the different states of a customer account e. g.

blocked, closed, normalThe Entities also have RELATIONSHIPS with each other. The relationship between the keys of two entities operates in both directions. The relationship between Customer and loans would be described as ‘ A Customer may have one or more Loans’ and ‘ A Loan must belong to one customer’. The relationship contains information telling us if a record in one table can exist with out a partner record in another table. Here, a loan can not exist with out a customer but clearly a customer may exist who never uses the library and never has a loan.

In relational databases such as Oracle, Ingress and Access, these relationships can be enforced so that you can not enter a loan record for a customer that does not appear in the customer database. Below is one representation of this relationship. It can also be represented using ‘ crows feet’ for Many and a single straight line as One. The fact that a customer may not have a loan is represented on the diagram by the vertical line behind the ‘ crows foot’. Entity-Relationship diagrams like these are very important documents that assist the original design of an application.

There can be hundreds of entities and the diagrams can be very complex. They are also used when designing any modifications to an application. Please note: Many to Many relationships can not be implemented. Records can not be correctly mapped. These relationships are resolved by inserting an extra entity in the middle of the relationship. Exercise 2In MS Access, create relationships for the three Entities.

Click on the relationship button. Add all three tables to the relationship pageClick on Borrower ID in Customer and drag it to Borrower ID in Loan. Tick the Enforce referential integrity box. Notice the relationship is 1 to many. Click OKThe relationship will now be represented in the diagram.

Set up a relationship between Book and Loan. Open up table loan for data entry. Enter the following data: FieldDataLoan ID1ISBN Code100Borrower ID1Borrower nameBloggsDate out15-11-2000Date due backFine DueTimes RenewedNormalisationData Normalisation is a technique used in conjunction with entity relationship modelling to refine the model. It helps to ensure that the resulting model is flexible. There are three main stages in the process. First Normal Form (1NF) – An entity is in 1NF if there are no repeating groupsSecond Normal Form (2NF) – An entity is in 2NF if it is already in 1NF and each non-identifying attribute depends fully on the keyThird Normal Form (3NF) – An entity is in 3NF if it is already in 2NF and there is no dependency between the non-identifying attributesIf we look at a library example (below), we can see that the entity ‘ Customer’ is not really in 1NF.

It has repeating groups providing data on the different media that are available for loan. What happens when the library starts to lend out other product types e. g. mini discs? The data structure could not cope. To bring it in to 1NF we would create another entity called something like Customer Media DetailsEntityAttributeCustomerBorrower IDNameAddressContact NumberAccount StatusCustomer Media DetailsMedia TypeBorrower IDMedia DescriptionMedia LimitMedia StatusCustomer and Customer Media Details are now in 1NF.

Customer Media Details is not in 2NF. The key is Media Type + Borrower ID together. Media Limit and Media Status are both dependent on the full key. Media description is dependent only upon the Media Type. EntityAttributeCustomer Media DetailsMedia TypeBorrower IDMedia LimitMedia StatusMediaMedia TypeMedia DescriptionThese are now in 2NF and are more flexible.

You could easily incorporate other control mechanisms for each media type. Since the non-identifying attributes are not dependent on each other, they are also in 3NF. To demonstrate 3NF we need to look at entity ‘ Loan’. The Entity Loan has both Borrower ID and Borrower name. There is a dependency between these two attributes. In order for this entity to be converted to 3NF we need to remove borrower name from the table.

EntityAttributeLoanLoan IDISBN CodeBorrower IDBorrower nameDate outDate due backFine DueTimes RenewedIf Borrower name did not already exist in Customer, we would need to create a new entity for it. Since it does already exit, its appearance in this entity was purely data redundancy so it can be removed entirely. EntityAttributeLoanLoan IDISBN CodeBorrower IDDate outDate due backFine DueTimes RenewedThis is now in 3rd Normal Form. We can see that by normalising the data we have generated extra entities, which will in turn have relationships. The process is iterative and for major systems, teams of analysts have ‘ walk through’ sessions to ensure accuracy of the data structure.

Normalisation becomes almost second nature after a while. The problems are often dealt with in the E-R Modelling stage with out realising it because

much of it is common sense. It is still a useful tool to double check that you have not missed anything. Final StructureEntityAttributeCustomerBorrower IDNameAddressContact NumberAccount StatusCustomer Media DetailsMedia TypeBorrower IDMedia LimitMedia StatusMediaMedia TypeMedia DescriptionLoanLoan IDISBN CodeBorrower IDBorrower nameDate outDate due backFine DueTimes RenewedBookISBN CodeTitleAuthorPublisherLoan PeriodNB This data model does not cater for multiple copies of a book. Also, the other media types may not have ISBN numbers. Systems Analysis