Database Management System
Department of Computer Science and Engineering
Indian Institute of Technology Madras

Dr.S.Srinath

Lecture – 30
Introduction to Data Warehousing & OLAP

Hello and welcome. In today's session, we are going to be looking at a slightly different topic from the conventional idea of databases and such a change in topic occurs simply because the kind of users that the database is going to be is meant for is going to change. Until now, we have a kind we have had an implicit assumptions that the users of who are going to be using the database are in some sense if I can put it this way or in some sense clerical users. In the sense that, they are most interested in adding and retrieving data elements into the database as efficiently as possible that means, the users who are using this database are involved in the operational aspects of a larger system.

For example: if you are thinking of, let us say railway reservation we are talking about how best we can design a database system for reserving a ticket. Therefore, what it means is whenever a new ticket entry is made it has to be efficiently entered into the database and suppose if there are any modifications for a given ticket entry, it has to be efficiently modified that means, it should be able to efficiently search the given entry and make modifications at just one place if possible and not many more places and keep that overall consistency of the database in time and that is what is meant by operational aspects of a database system or operational aspects of a system is a day to day operation. Somebody comes you gives a request for a reservation, you enter the request, you reserve a seat for him and give him the ticket and or cancel it something or ask for concessions or whatever and so on.

But, there are other kinds of users who use the database system as well and specifically we are talking about users who take strategic decisions in addition to or in contrast to the tactical decisions that are taken by the operational people that is the folk sitting in front of a let us say a ticket counter take very tactical decisions. How efficiently can you perform your operations? But then, there are a variety of strategic decisions which is the best way which is the best location for me to place my next railway reservation counter. Which part of the city has the most people traveling by trains or which part of the city has the most people traveling by first class ac or which part of which time of year is the best for me to offer concessions on second class sleeper, something like that.

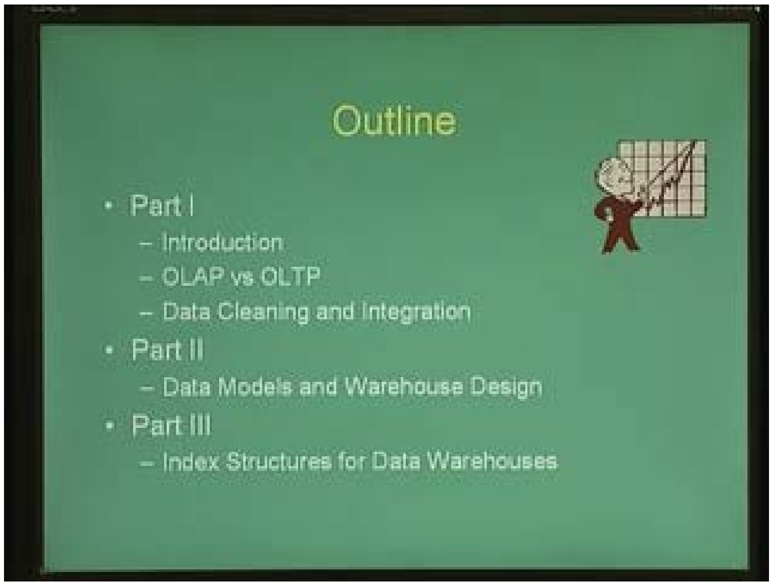
So these kinds of strategic decisions are of a qualitatively different nature than the tactical decisions that are taken for operational issues of any system but if you watch closely it is the same ……(Refer Slide Time: 4:20) or fed and retrieved in operational situations is the same data that is required for making strategic decisions as well. If you have to make strategic decisions about let us say which is the best location for me to open my next open the next reservation ………(Refer Slide Time: 4:40)

For that we need a lot of information about what is the, from what addresses are people coming and booking tickets were right. If somebody comes from area ‘a’ and books tickets in area ‘b’ and assuming that area ‘b’ is the nearest reservation counter and the address information will in turn show me that because there are a lot of such people coming from here to there, it probably makes more sense for me to open a next counter here and so on. So we will see in the next few sessions how the whole aspect of data base design changes when the usage scenario changes from an operational data usage to a strategic decision making usage. So that brings us to the topic of data warehousing.

Data warehousing as you might understand the term warehousing, a warehouse is where you keep your inventory stocks right that is where you have stocks from several different sources are going to several different sources and you are essentially talking about a large number of stocks that are maintained in the warehouse and a warehouse is typically of strategic importance. If you take up, let us say some kind of civil engineering project, the location of your warehouse is of prime importance depending upon because there are which depends upon several factors like say what is the cost of transportation, logistics coordination and so on.

Suppose you are having you are handling a large infrastructure project somewhere. Let’s say building a flyover something like that, having the location of your warehouse in a in a way that is easily accessible with possible cost constraints is probably one of the most crucial strategic decisions that can be taken. In an analogous sense, a data warehouse is a warehouse of data elements that have been captured from different operational data sources. So that this whole set of data elements becomes is of strategic importance. You can take strategic decisions based on the data elements that you have gathered and which leads us to the next term in the slide here which is called the OLAP or OnLine Analytical Processing that means we are looking for requests or queries that are of an analytical nature rather than an operational nature or what you called as a transactional nature. We have talked about transactions quite a bit. Now we are looking into analytical queries where queries in turn help in strategic decision…….(Refer Slide Time:07:44)

[Refer Slide Time 7:49]



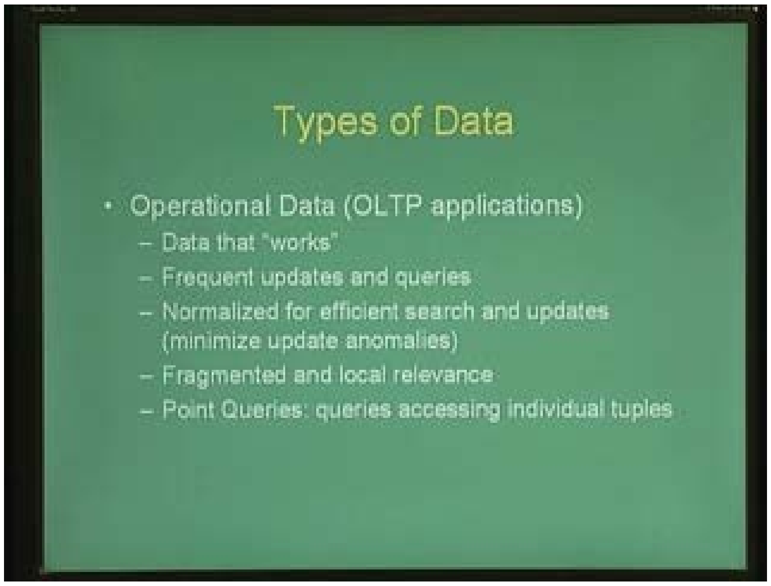
Outline

- Part I
  - Introduction
  - OLAP vs OLTP
  - Data Cleaning and Integration
- Part II
  - Data Models and Warehouse Design
- Part III
  - Index Structures for Data Warehouses

………….. by itself is a vast subject and that several research papers that have been written in several commercial implementations that have been implemented and the huge amount of interest in data warehouses. However, in this course we shall probably be touching just a small part of this vast ocean of data warehouses and the way we have partitioned this, our exploration into data warehouses are in three parts as shown in the slide here. In the first part, we shall be looking into an important difference between OLAP that is Online Analytical Processing versus the Online Transaction Processing that traditional data bases are geared towards ….. in data warehousing namely that of data cleaning and data integration.

The next two parts deal with the deal with the warehouse core itself where we are talking about the data models that that go inside the data warehouse and what are some kinds of the thumb rules which go towards data warehouse design. We will also look at some kind of index structures for data warehouse based on the data models that we studied and see that, how they differ from (Refer Slide Time :9:13) data bases like say b plus trees or b trees or hash based indexes or something like that.

[Refer Slide Time 9:26]



Types of Data

- Operational Data (OLTP applications)
  - Data that "works"
  - Frequent updates and queries
  - Normalized for efficient search and updates (minimize update anomalies)
  - Fragmented and local relevance
  - Point Queries: queries accessing individual tuples

When we are talking about data, we can essentially divide data into two kinds of data. Essentially I have called them as operational data and the next slide calls it as historical data, that is operational data can also be considered as data that works within quotes that is data that is involved in the operation of a particular system. For example: if you want to withdraw money from your account, you need data about your account, your account number, your pin number if you are using an atm your account balance and so on.

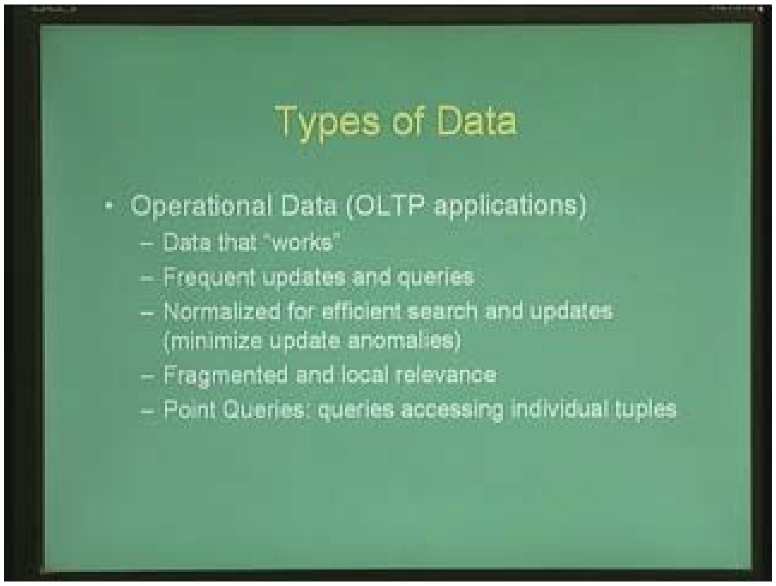
All these data elements are all operational data elements because they are crucial, they are necessary for performing the operation of withdrawal of money right and what kinds or what are the characteristics of operational data? Operational data are subjected to frequent updates and queries. You could be withdrawing money almost every day or some kind of operations would be happening on your account almost on a daily basis so there should be, there would be some kind of queries or updates happening quite frequently to the database that is maintaining your account information.

Therefore we saw in the sessions on normalizations and functional dependencies. We saw that in order to make the process of updates and queries efficient, we need to normalize the set of data elements that is we need to normalize what is to prevent any kind of redundancies occurring in the data and also to prevent any kind of update anomalies. Suppose update records of in some place and there are redundant copies of the same record of some fields, then I will have to update all occurrences of this field.

So in order to minimize update anomalies we essentially normalize the set of data elements and a set of tables that are normalized are essentially fragmented, because given data elements. Let us say your account information may require several information something like your account number, name, address, branch location, balance, transaction history and type of account so on and so forth and because they are normalized, the set of all tables are fragmented in different phrases and the operational data is usually of local relevance which is kind of emphasized in the should kind of emphasized that means it is very unlikely to expect. Let us say that you have opened an account in Chennai and you go to Delhi and you want to access your account. It is unlikely to expect the account information that you have opened in your bank in Chennai to also be present in Delhi. It has to be queried in Chennai and your request has to be routed through Chennai and it has to be queried here and the result sent back to Delhi.

So, operational data is usually of local relevance, wherever the data is it is relevant there in in the geographic location and the kind of queries over operational data is also what are called as point queries. What is a point query? A point query is something like asking something about asking query about some individual tuple. For example: what is the balance in your account with account number so and so or what was the last set of ten transactions that your account had. Therefore, what is happening here is that, there is there is a particular point or a specific tuple is the key where your account number in this case where which is being used to access all relevant information for the operationalization.

[Refer Slide Time: 13:38]



Types of Data

- Operational Data (OLTP applications)
  - Data that "works"
  - Frequent updates and queries
  - Normalized for efficient search and updates (minimize update anomalies)
  - Fragmented and local relevance
  - Point Queries: queries accessing individual tuples

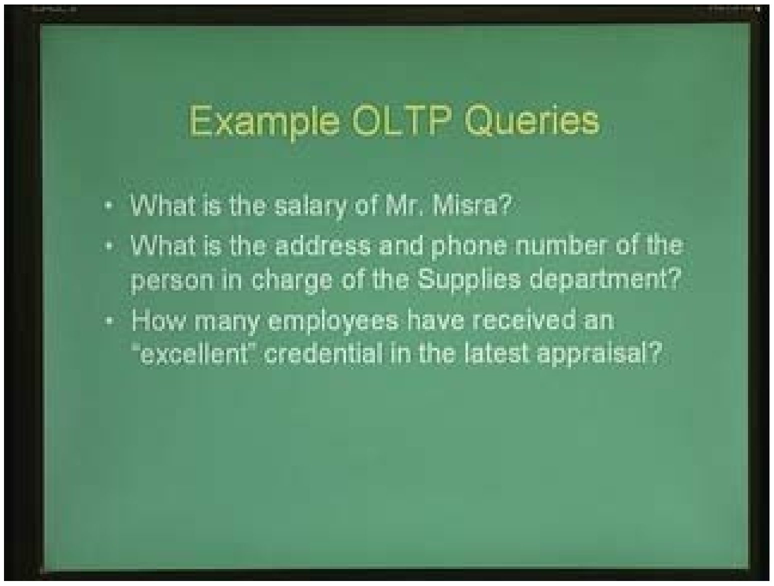
On the other hand, there are you can think of another kind of data called as historical data or also called as archival data that is data which are archived over a long period of time over a long period of different operational data sets. It is a long set of operational data sets and it is these kinds of data which tell us something that is it is also called as data that tell us something about the overall trends of operationalization that is happening, that is suppose I have a data collected over from all railway reservation centers for the past ten years it can tell us about trends as to which are the peak times in which people travel what kind of people travel? In what kind of or make what kinds of railway reservations and so on. I mean people having what kind of salary range travel in what kind of classes? Sleeper classes AC or whatever and so on.

So it is that data that tells us something about the larger system which is using the data set and as you can see here, the kind of updates that, this data sets undergoes is quite infrequent in nature. It is not like every day that you are going to get historical data. You have to get historical data over a period of time. Therefore, you collect historical data let us say once in six months or once in a year or something of that order.

So it is a very infrequent updates but what is more important here is, that updates are not all that important that is they happen very infrequently. So, we can take care of them whenever they happen, but it is the queries that are more important, that is analytical queries require huge amounts of aggregation. What is the average age of the person traveling in second class sleeper? If you want to calculate that, it needs huge amount of aggregation over a large data set based on all different second class sleeper tickets that have been sold and this is an integrated data set with a global relevance.

We are not it usually it does not make sense to look at trends for trends in a given operational data source. It does not make sense to say what is the trend in this particular reservation center? I mean it is it usually would be very small compared to the larger set of all different operational data sources and the performance issues in managing historical data occurs mainly in query times and not in update times because of course, updates are far infrequent or far more infrequent than queries and queries need to access a large amount of data and you have to perform a number of aggregate operations before being able to return the query and they have to return it in an online fashion, that is why it is called online analytical processing. You have to return it in a return query results in an interactive response time that is the user should be sitting in front of a terminal and the interactive response time is usually of the order of maximum a few seconds and you cannot expect the user to be waiting for a large period of time after giving the query.

[Refer Slide Time 17:22]



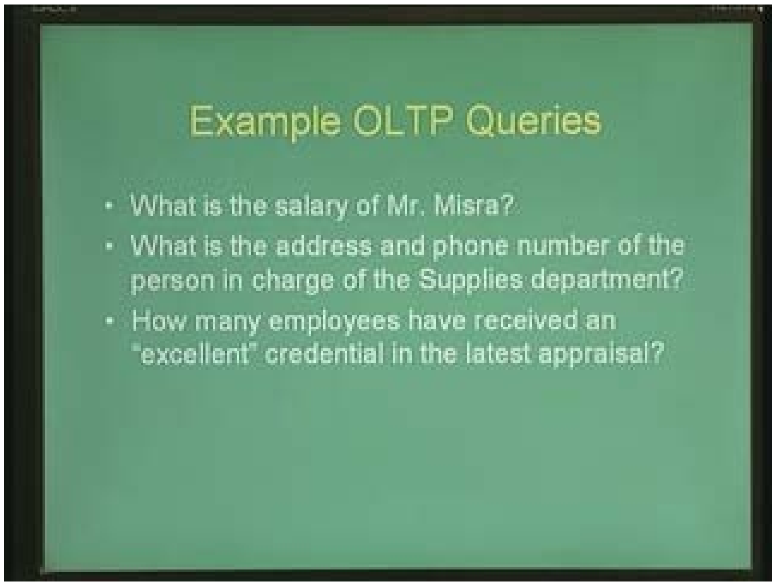
Example OLTP Queries

- What is the salary of Mr. Misra?
- What is the address and phone number of the person in charge of the Supplies department?
- How many employees have received an "excellent" credential in the latest appraisal?

Let us look at some examples of a transactional data and historical data. A transactional data or operational data are those kinds of data limits that are handled by systems that are called as OLTP systems or Online Transactional Processing Systems that is these systems which maintain transactions and handle different transactional activities in the system. Now what are some examples of operational queries, something like say what is the salary of Mr.Misra? It is some it is a point query. As you can see here that is, you find out the employee number of Mr.Misra and then get the salary field and that is it or who is the what is the address and phone number of the person in charge of the supplies department?

Again it is a point query that is just a question of following different references find the supplies department and look up the set of managers given the suppliers department id look up who is the manager managers id and then given the managers id, find the address and phone number of the manager and so on. So these are some kind of typical, analytical queries, typical transactional queries which we have seen quite often. Now that is we have seen how to specify these queries in SQL, how to optimize these queries, how to create transactions around them and so and so queries shown in this slide.

[Refer Slide Time: 18:58]



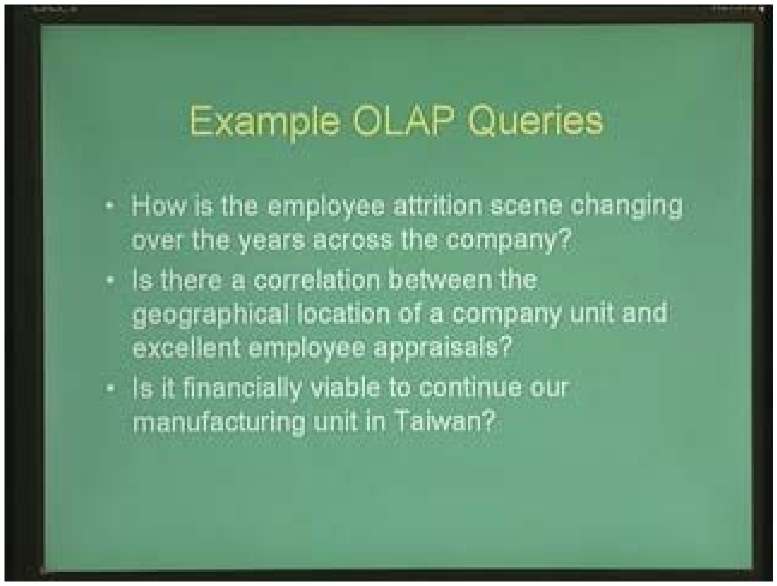
Example OLTP Queries

- What is the salary of Mr. Misra?
- What is the address and phone number of the person in charge of the Supplies department?
- How many employees have received an "excellent" credential in the latest appraisal?

How is the employee attrition scene changing over the years across the company? As you can see, this kind of query is qualitatively different. It is qualitatively in a different caliber than the kinds of queries which we saw in the previous slide. If you want to answer this query, let us say how the employee attrition scene is changing over the years and across the company. It is not a question of accessing one particular record or one particular tuple. You need to look at employee attrition information across the company and across the years and then look for trends saying it is increasing, it is decreasing, and so on. That is you have to find out some aggregate employee attrition information across the company and then plot it against the years and see how it is changing or something like, is it financially viable to continue our manufacturing unit in Taiwan.

It is a pretty vague query right. What do we mean by financial viability? What do we how do we calculate financial viability profits against cost and so on? So, how do we know that we are incurring more profits than costs or we are gaining more profits than the cost that we are incurring. It is not quite easy to find that out, that is you need to be able to aggregate a number of different information sources and then say profits are generally more than the cost and so on right. So, these are the kinds of queries which are of analytical nature and typically for interest for strategic decision, making managers who perform strategic decision making.

[Refer Slide Time: 20:54]

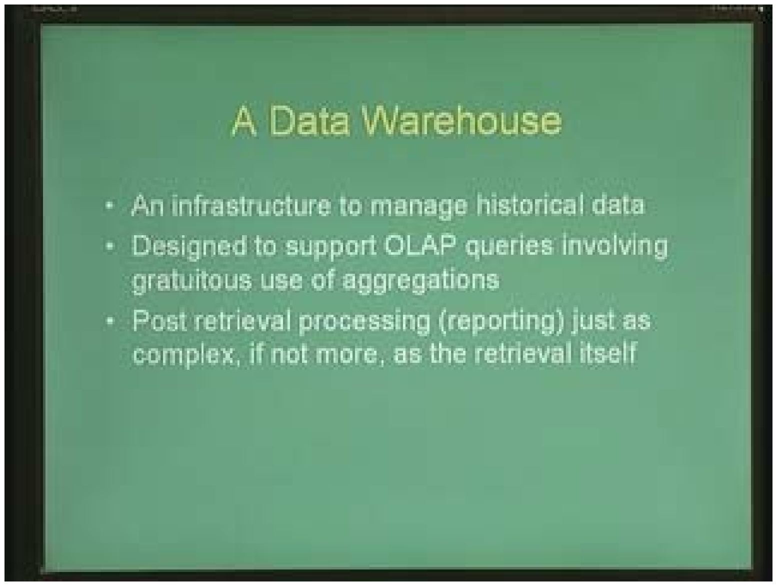


Example OLAP Queries

- How is the employee attrition scene changing over the years across the company?
- Is there a correlation between the geographical location of a company unit and excellent employee appraisals?
- Is it financially viable to continue our manufacturing unit in Taiwan?

A data warehouse is an infrastructure to manage such kinds of historical data and it is designed to support OLAP queries involving very gratuitous use aggregations aggregated queries and so on and it is not just a queries, there are also a number of post retrieval processing also called as reporting which is just as complex or possibly more complex than the query retrieval itself, that is I have gathered huge amount of information about all possible profits and costs from all possible centers and so on. How do I project this information? How do I give out all these information to the decision maker which is again a quite involved in itself?

[Refer Slide Time 21:44]



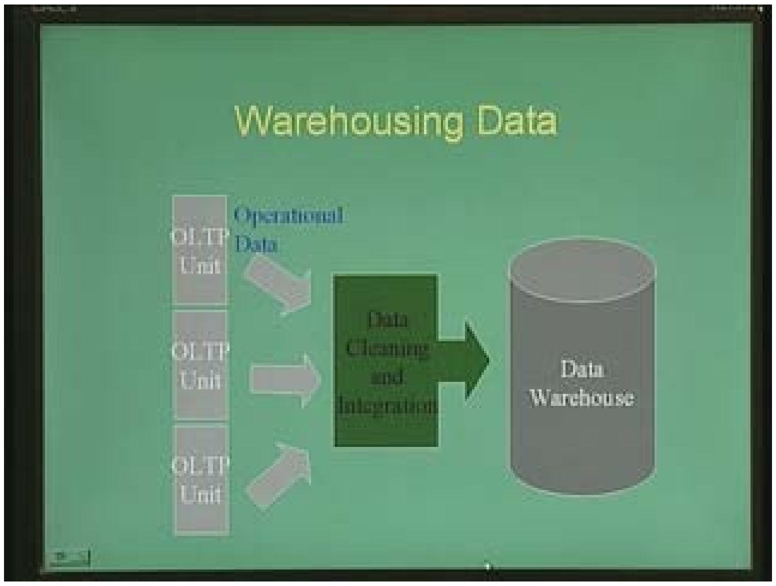
A Data Warehouse

- An infrastructure to manage historical data
- Designed to support OLAP queries involving gratuitous use of aggregations
- Post retrieval processing (reporting) just as complex, if not more, as the retrieval itself

So these slide shows the schematic diagram of the oral architecture of a data warehouse. A data warehouse as we saw in the previous slide is meant for managing historical data and where do you get the historical data from or how do you first of all come out with historical data? Historical data you essentially obtain from all the operational data sources or the OLTP units. So you have several OLTP units for your organization. Whenever you are thinking about an organization here, think of a large organization something like say the life insurance corporation of India or the Indian railways or something like that where they have a number of different units each of them having their own databases.

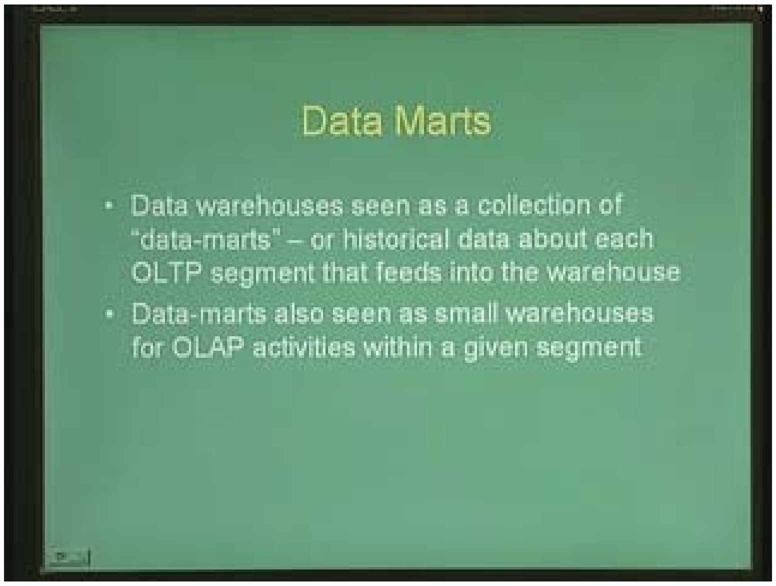
Each of them handling their own accounts, each of them handling their own reservations and transactions and so on and from each of these OLTP units, we obtain huge number of operational data elements everyday right everyday there are number of people traveling in trains and from number of different and they would have booked their tickets from number of different sources. Now all of these operational data are then subjected to a data cleaning and integration process. We shall see shortly what is meant by data cleaning and integration? There are huge number of possible inconsistencies or possible sources of what we called as dirty data that can exist in the OLTP sources and we will see what these sources of dirty data that can exist and how we can clean, what is also called as data scrubbing and so on before we are able to present it as historical data. Now once operational data is subjected to data cleaning and are integrated across all of these OLTP sources, we present them as historical data into a data warehouse that is, a data warehouse would already exist and you just integrate this set of data into the data warehouse. You just update the data warehouse with these set of data.

[Refer Slide Time: 24:04]


Warehousing Data

There is also a notion of data marts which is probably of interest here at least look at the definition. Whenever we talk about data warehouses, we look at then as a collection of data marts that is a data mart is a historical data about one specific kind of one specific segment or one specific OLTP segment. Suppose you are considering let us say again take the example of Indian railways. Let us say, we have one segment called express train reservations that is one data mart. Suppose, we are interested in strategic decision making only about express trains or may be let us say Rajdhani express or something like that all the super-fast trains. So all of the historical data that we gathered about Rajdhani express would go into that data mart called Rajdhani express that is of all Rajdhani expresses across the railways and over a period of time and so on. So and the data warehouse is usually seen as the collection of this different data marts that feeds into the data warehouse and data warehouse data marts are also seen as small warehouses where you can in in some way you can support all activities that you support on a data warehouse also on a data mart and but these are within a given segment.

[Refer Slide Time 25:45]



Data Marts

- Data warehouses seen as a collection of "data-marts" – or historical data about each OLTP segment that feeds into the warehouse
- Data-marts also seen as small warehouses for OLAP activities within a given segment

Now let us expand this system or this box and data cleaning and integration and see what happens inside that? As we mentioned earlier, we obtain data from different OLTP data sources, operational data sources that feed into the data warehouse. Now can we or what kind of (Refer Slide Time: 26:12)………… Before we are able to integrate operational data sources or before we are able to populate the data warehouse from the operational data sources. Operational data sources as we have seen are mainly meant for the data that works that is as long as the data is sufficient for the operation to be performed. It is good enough. It is(Refer Slide Time:26:49) ……… characteristic of operational data is that operational data is of local relevance. It is usually relevant, that is data in an operational data source is relevant only locally.

However, when we are talking about only historical data, in a data warehouse there are two different things that you have to (Refer Slide Time: 27:11)……… one is that because we are using different kinds of data sources, we should have a uniform standard of data representation and semantics across all of these different data sources and secondly we should remove any kind of duplicate information that are present in these data sources.

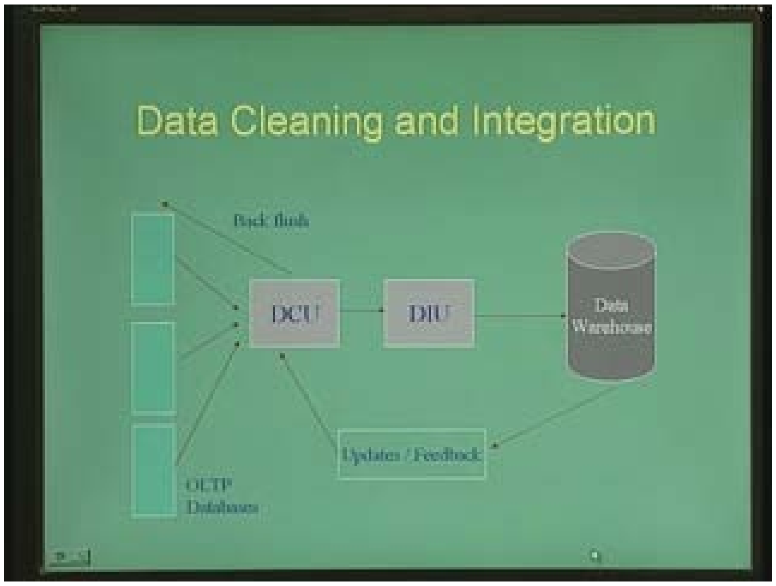
Suppose you have opened an account. Let us say, let us again take the running example of the Indian railways. Now suppose you have opened an account for booking tickets over the web. Now you have booked a few tickets over the web and then you have not booked tickets for quite significant period of time in which time you forgot your user name and password and you forgot that you have ever opened an account here and then you go back and open another account for buying tickets over the web. Now as you can see if these two accounts the behavior in these two accounts are considered different are considered to be from different users, then it will give us erroneous information or erroneous aggregate information for strategic decision making.

It is important that for historical data that these two accounts should be clubbed or we should recognize that these two accounts belong to the same user and then club them together whenever we are performing any kind of aggregate queries. So, let us see what does it entails to do all these or what are the kinds of complexities that we encounter in performing these kinds of data integration and cleaning and so on and so forth.

The general model of data integration and cleaning is shown in the slide here. As we saw in the previous slide, these are the different OLTP data sources which feed into the data cleaning and integration unit and usually the (Refer Slide Time: 29:26)………… we have opened the box which was a black box here. Now we have opened the box and you see that there are usually two different units here. The data cleaning unit and the data integration unit and the process by which data flows through this unit is not a unidirectional process that is you garner a data from this OLTP data sources and pass them through data cleaning, pass them through data integration and also perform some kind of a back flushing that is (Refer Slide Time: 30: 02)…………… that you have back into the OLTP database itself.

We will look at some examples which tell us what it means actually and once the data is integrated it is fed into the data warehouse and based on queries in the data warehouse or updates in the data warehouse you get a feedback from the warehouse itself which tells how we said that you should perform your cleaning? What is important for the data warehouse? Cleaning essentially means that we have to change whatever is important for OLTP? We have to change representations from whatever is important for OLTP to representations where whatever is important for the data warehouse.

[Refer Slide Time: 30:51]



Data Cleaning and Integration

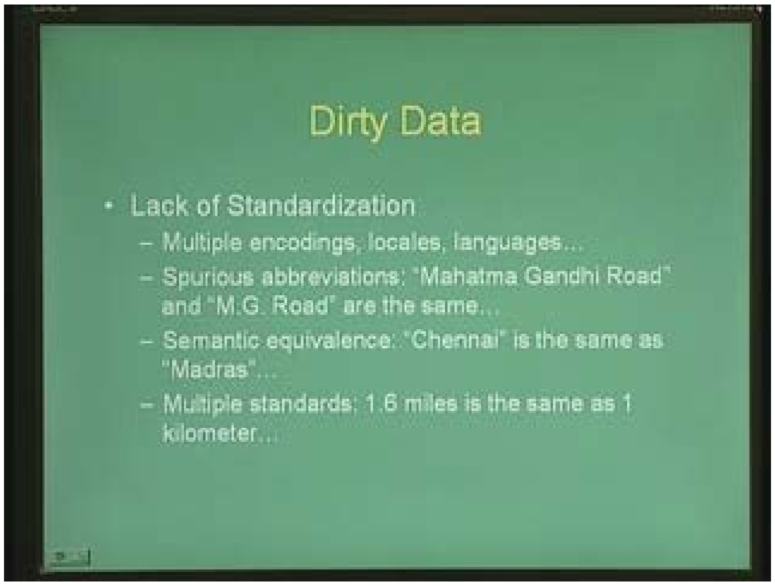
So, let us look at data cleaning or the dcu in a little bit more detail. As the name suggests, data cleaning performs cleaning operation on data sets that is data sets that contain dirty data. What is meant by dirty data that is, what are the different sources of dirt? I am talking about dirt within quotes. So that one can encounter for given operational data source. There are several different possible sources of dirt that requires cleaning to be performed. The first kind of dirt is what called as lack of standardization. For example: because you have different branches, different reservation centers or different cities. There could be multiple encodings. One of them could be using ASCII base systems, one of them could be using Unicode or one of them could be using head side or something or the other.

So there could be multiple encodings, multiple locales multiple languages in which data is represented and you should be able on to standard encoding or language or whatever. There could be spurious abbreviations. Somewhere somebody writes Mahatma Gandhi road in part of an address and somebody else writes M.G road. Now we need to be able to recognize that M.G road is same as the Mahatma Gandhi road. So when we are talking about, when we are answering a query like how many people comes from in and around Mahatma Gandhi road or live in and around Mahatma Gandhi road who buy tickets.

We should be able to say that, we should be able to search both M.G road and Mahatma Gandhi road further because they are the same and similarly there could be semantic equivalence, M.G road and Mahatma Gandhi road probably you can write an intelligent algorithm that will see whether mahatma Gandhi road is an expansion of M.G road and so on and you could kind a well fairly find out this is a spurious abbreviations and so on. However what if there are semantic equivalence? Chennai and madras. Some people may use Madras for the city name and some people may use Chennai and because we are talking about historical data, note that which is important here that is at one point in time officially Chennai was called as madras. So even in official documents, you would be using the name madras but then again it would have become Chennai.

Now you should be able to query later or say that these two are the same and knowing though intelligent algorithm can do that. You need extra knowledge in addition to the data sources. You need knowledge about the overall environment, the governmental policies, the standards and so on in order to be able to identify these duplications. Similarly there could be multiple standards. There could be one data source which could be using the metric systems, while somebody else could be using miles and feet and so on and so forth. So somebody might in someone data source might have said 1.6 well 1.6 kilometers is the same as one mile. So 1.6 kilometers and somebody else might say one mile and so on.

[Refer Slide Time: 34:30]



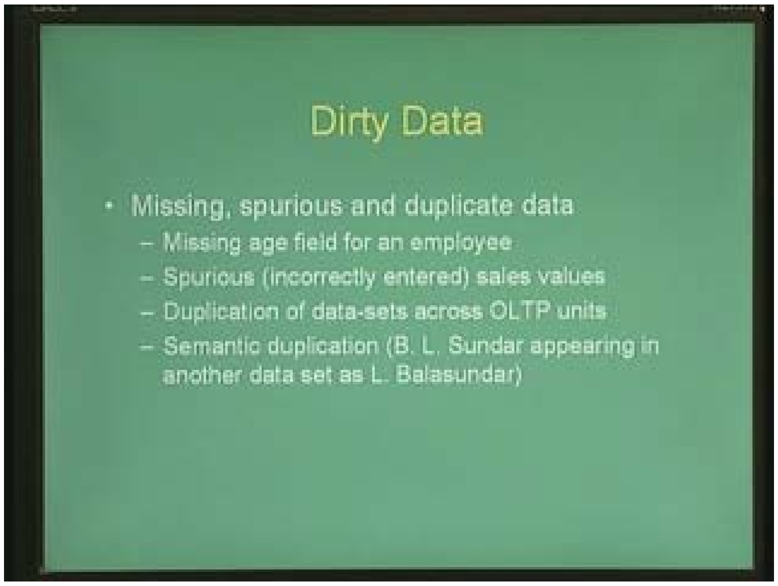
Dirty Data

- Lack of Standardization
  - Multiple encodings, locales, languages…
  - Spurious abbreviations: "Mahatma Gandhi Road" and "M.G. Road" are the same…
  - Semantic equivalence: "Chennai" is the same as "Madras"…
  - Multiple standards: 1.6 miles is the same as 1 kilometer…

There could be other sources of dirty data like missing spurious or duplicate data. For example: the age field of an employee could be missing which will hamper my query of what is the average age of an employee or something like that, what is the average or what is the correlation between the age of an employee and the attrition and the probability of attrition and so on and so forth or there could be incorrectly entered sales values or incorrectly or some typographical errors and so on or there could be duplication of data sets and duplication appearing in different forms.

For example: take look at the last example. A person called L. Bala Sundar would have registered in one, would have bought a ticket in one reservation center sighting his name as L. Bala Sundar the same person could have gone to a different reservation center and bought another ticket at some other point in time, sighting his name as B L Sunder and with the same address and the same phone numbers and so on and so forth and we should be able to detect such kind of semantic duplication. That is, thee is duplicate data that is occurring at different data sources, but not in the same form that is the name has changed or something has changed, but semantically they are still duplicate.

[Refer Slide Time: 35:56]



Dirty Data

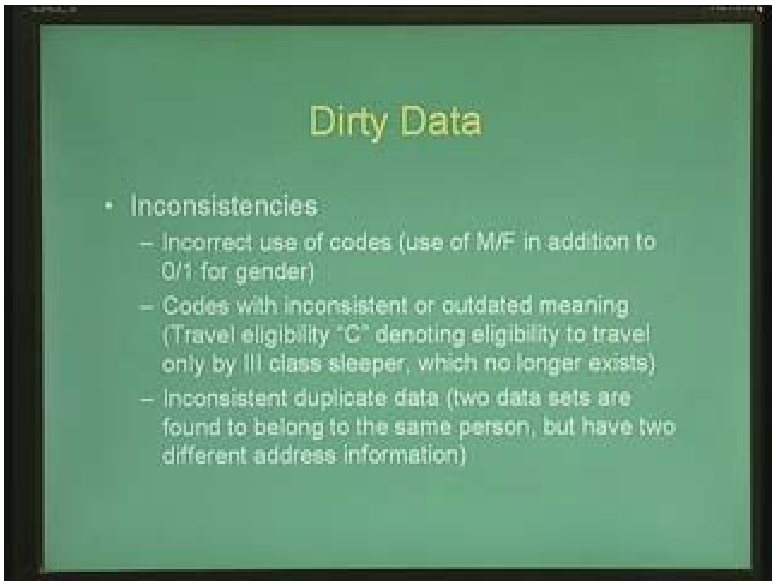
- Missing, spurious and duplicate data
  - Missing age field for an employee
  - Spurious (incorrectly entered) sales values
  - Duplication of data-sets across OLTP units
  - Semantic duplication (B. L. Sundar appearing in another data set as L. Balasundar)

There could be other sources of dirty data like inconsistencies or incorrect or inconsistent use of codes. Let us say, in one of reservation center the gender of the person who is reserving a ticket is maintained as a character either m or f or and in other reservation centre, it is maintained as 0 or 1. It is just the different standards that each data base designer has designed for each center and this is especially true if this has evolved over time. Let us say that reservation center, number one would be using historical data base that was designed which is not changed because of cost considerations and the new reservation centers using a new data base system that has been redesigned which is termed to be more efficient and so on and so forth but which uses different codes for the same data element and there could be codes with some outdated meaning.

Let us say third class sleeper because we are talking about historical data. It is quite possible that we encounter some data about third class sleepers which no longer exists in railways today and so on and there could be inconsistent duplicate data that is, there could be two data sets that are found to belong to the same persons. We have reasons to believe that it is the same person. But, he has given two different addresses in these two sources and so on how do we detect such things and other kinds of inconsistencies like inconsistent associations that is, let us say one department provides particular kinds of says that the sales have been so on and so on. But, it does not add up to the total sales figure that is also been provided. So, associations across different data sources may be inconsistent and there could be semantic inconsistencies. Somebody might have typed February 31st in a date.

So what is that referred to? Is it February 31st or March 1st or March 31st or what or integrative violations in referential integrity like referential inconsistency. There is, let us say 10 lakh rupees sales reported from a unit that has actually closed down which is not even there and so on. So several kinds of these sources of dirty data crop up in practice and perhaps cleaning or data cleaning is perhaps the single biggest research problem that is still kind of open in the field of data warehouses, because there is no single thumb rule for data cleaning. We cannot just say all data is pass through one kind of data cleaner and it will be cleaned, because there is several sources of dirty data that can occur and there is no single way of cleaning them.

[Refer Slide Time: 39:10]



Dirty Data

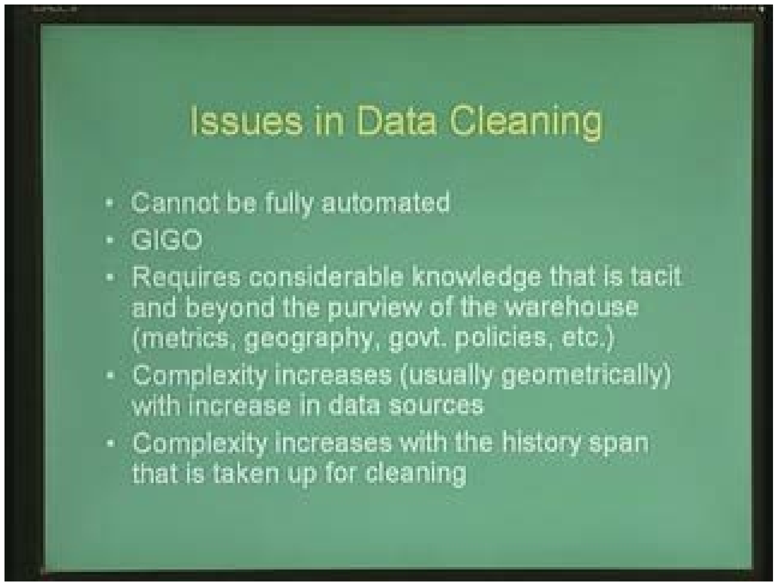
- Inconsistencies
  - Incorrect use of codes (use of M/F in addition to 0/1 for gender)
  - Codes with inconsistent or outdated meaning (Travel eligibility "C" denoting eligibility to travel only by III class sleeper, which no longer exists)
  - Inconsistent duplicate data (two data sets are found to belong to the same person, but have two different address information)

So what are the issues in data cleaning? It (Refer Slide Time:39:45) ………..is automated. It is quite difficult to automate for the computer to learn by itself that let us say Chennai was called Madras or Burma was called Myanmar and so on and so forth and the thing we don't even know whether the data cleaner is performing correctly which is what is called as GIGO or garbage in or garbage out. Suppose you give some garbage and we don't have correct rules. We just give some junk data and it just gives some junk outputs we don't know whether it has actually cleaned or whether the output is correct or not and it is its quite hard to verify the cleanliness of the output data.

And data cleaning requires considerable knowledge that is tacit For example, one kilometer or one point six kilometer is equal to one mile which we know but it is kind of a tacit knowledge it is not exquisite knowledge and which goes beyond the purview of the database like Chennai was called madras. So you should know data about governmental policies you should know data about geography should know some rather some knowledge about geography, governmental policies, metrics and so on and so forth which has to go into the data cleaning unit.

So, it is not possible to design a data cleaning unit in a way which kind of paraphrase does one size fits all or which says that this is the data cleaning unit and this will work for any kind of data in any context what so ever and so on and the data cleaning complexity increases as we increase the number of data sources or they increase the history span that we have taken up for cleaning.

[Refer Slide Time 41:10]



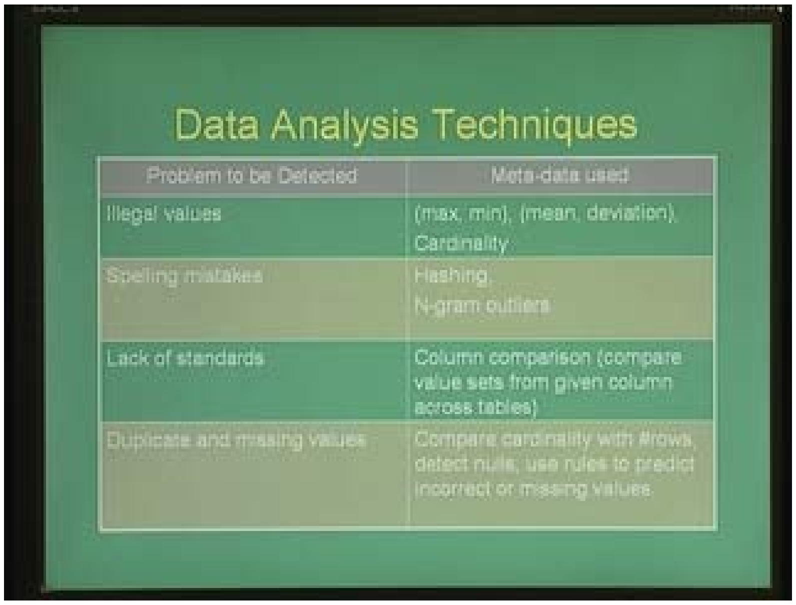
Issues in Data Cleaning

- Cannot be fully automated
- GIGO
- Requires considerable knowledge that is tacit and beyond the purview of the warehouse (metrics, geography, govt. policies, etc.)
- Complexity increases (usually geometrically) with increase in data sources
- Complexity increases with the history span that is taken up for cleaning

What are some of the steps in data cleaning? How does typical data cleaning process looks like? Essentially, there are five different steps that go into a data cleaning process. Firstly, we will start with data analysis. Given the set of OLTP data sources; we start by analyzing them and look for certain kind of metadata that is what we can learn about the data that we have and then give all those metadata back to the user or the data cleaning.

Now the user in turn will now specify a set of transformation rules that is if this is the kind of dirty data this how we have to transform it into clean data and so on. So the user specifies the set of transformation rules that that are performed either the scheme or data level which transforms dirty data into clean data. We then verify the rule by running them on test data sets that is some kind of sample data sets and then we incorporate the transformation rules into the data cleaner and then perform the data cleaning process So once the data cleaning process is performed we usually also perform what is called as a back flow that is we repopulate the data sources with cleaned data. For example: Suppose we have seen that Chennai and madras are used interchangeably in in the in the OLTP data source, we see to it that or we perform a back flow, so that it is all either Chennai or madras in the OLTP sources that is, there is some kind of standard in the OLTP sources itself which is called as back flow of data.

[Refer Slide Time 42:55]



Data Analysis Techniques

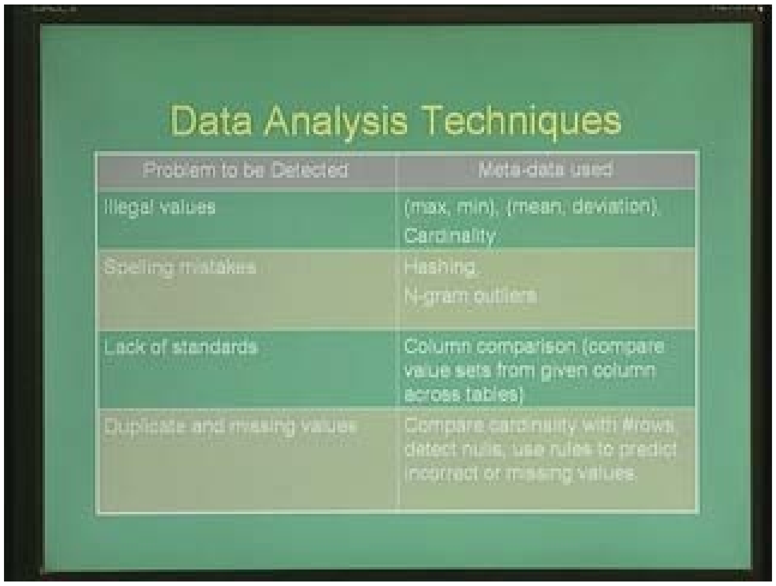
Problem to be Detected	Meta-data used
Illegal values	(max, min), (mean, deviation), Cardinality
Spelling mistakes	Hashing, N-gram outliers
Lack of standards	Column comparison (compare value sets from given column across tables)
Duplicate and missing values	Compare cardinality with #rows, detect nulls, use rules to predict incorrect or missing values

We shall not be looking into great detail into how each of these transformation rules is performed. But let us have a look at some specific or some typical examples of how dirty data can be cleaned. For example: how do we search for illegal values within what is the strategy for illegal values. We could use some kind of a max min or a mean deviation or cardinality criterion that is slide a window through your data source that contains a max min limit for the particular data value.

Whenever a given data value is lesser than the mean or greater than the max, you know that it is an illegal value. So, similarly mean and deviation or cardinality and so on and so forth and spelling mistakes. How would you look for spelling mistakes? There are some techniques what are called as n gram outliers an n gram is essentially a collection or a sequence of n letters that form a different words. For example: suppose I take a three gram, let us say hashing I perform a three gram transformation on hashing, then I get several different three grams. ‘has’ is a three gram, ‘ash’ is a three gram and ‘she’ is a three gram and ‘him’ is a three gram and so on.

So I get different three grams, then what we do is we cluster all these three grams based on their occurrences and we see if (Refer Slide Time: 44:36)…………… layers in the clusters that is, are there any n grams which standalone without belonging to any cluster. First slightly, these outliers are going to be spelling mistakes and that is the strategy that is generally used for checking spelling mistakes. Similarly, lack of standards that is compare values sets from a given column across different tables and see whether they are using the same standard or so on and duplicate and missing values that is compare the number of rows with the cardinality of this particular column. So, you find out if the number rows do not match the cardinality you see that there could be some kind of missing values or null values say in this.

[Refer Slide Time 45:31]



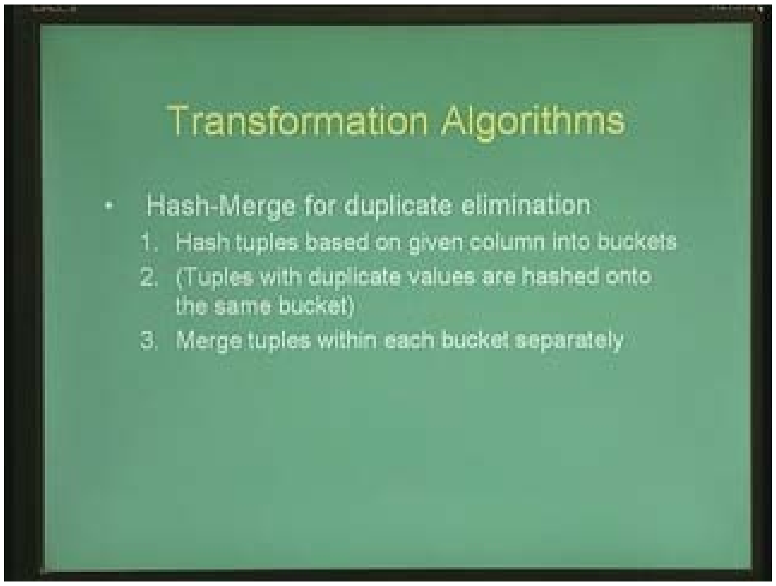
Data Analysis Techniques

Problem to be Detected	Meta-data used
Illegal values	(max, min), (mean, deviation), Cardinality
Spelling mistakes	Hashing, N-gram outliers
Lack of standards	Column comparison (compare value sets from given column across tables)
Duplicate and missing values	Compare cardinality with #rows, detect nulls, use rules to predict incorrect or missing values

So, let us look at one or two algorithms in slightly more detail which can give you an appreciation of the kind of the complexity that it takes or the kind of techniques that are used for data training. Simple algorithm for duplicate elimination what is called as the hash merge algorithm. If you remember the storage structures session that we that we covered you know what is meant by hashing right that is given a particular tuple. Run it through a hashing function so that it is mapped on to a given bucket.

Now what can we say about hashing across different tuples? Duplicate tuples that is tuples with duplicate (Refer Slide Time:46: 21)………….. therefore elimination of duplicates now reduces to searching within a bucket for looking at any duplicate values and then eliminating them.

[Refer Slide Time: 46:36]

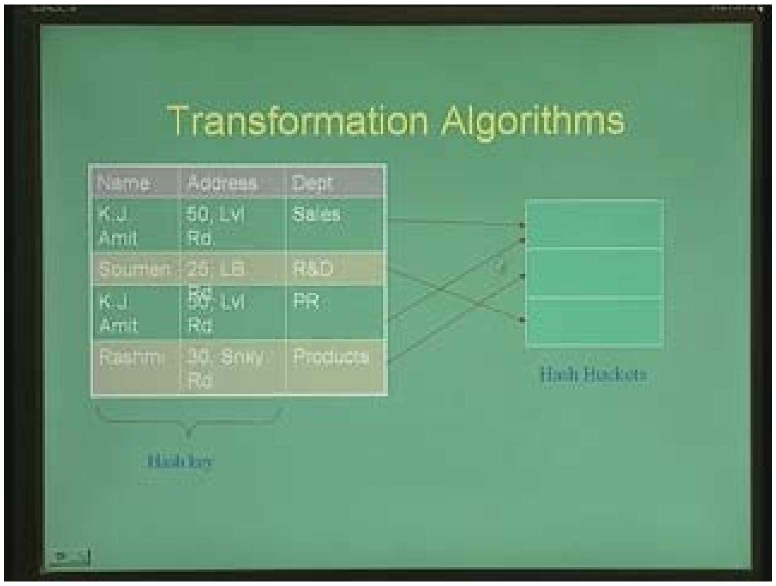


Transformation Algorithms

- Hash-Merge for duplicate elimination
  1. Hash tuples based on given column into buckets
  2. (Tuples with duplicate values are hashed onto the same bucket)
  3. Merge tuples within each bucket separately

So here is an illustration of the hash merge algorithm. Let us say, there are four different records like this and this is the hash key that is the name and address of the person that is we want to eliminate duplicates as for as person information goes. So as you can see, K J amit and 50 Lvl road, both of them map onto the same bucket and these map onto different buckets and so on. So you just compare within a bucket and then eliminate duplicates.

[Refer Slide Time: 47:07]



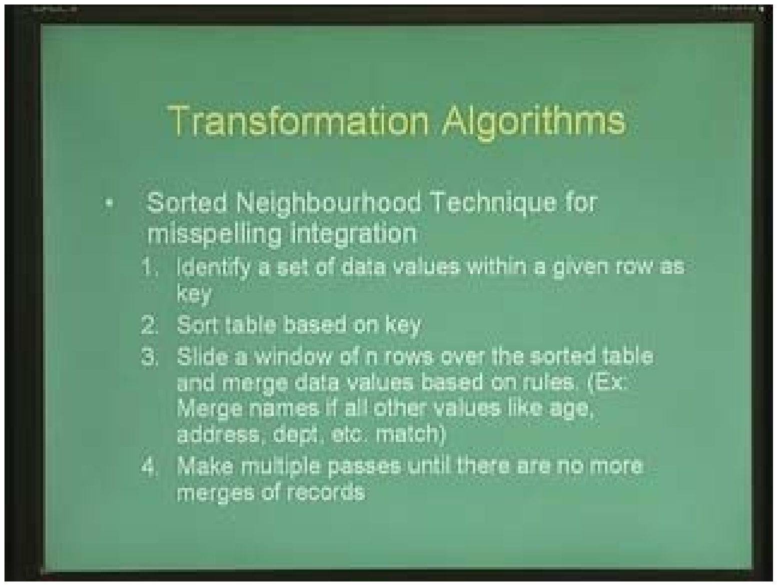
Transformation Algorithms

Name	Address	Dept
K.J. Amit	50, Lvl Rd.	Sales
Soumen	25, LB Rd	R&D
K J Amit	50, Lvl Rd	PR
Rashmi	30, Snwy Rd	Products

There is another technique called sorted neighborhood techniques. This is mainly used for misspelling or detecting misspellings in data sets and sorted neighborhood techniques is given by the following algorithm shown in this slide here. First of all, identify what is the key? As in the previous case, we identified that name and address is the key so identify what is the key within a given tuple. Then sort the table with (Refer Slide Time:47:41)…………… the table based on the key you can see that all duplicates will cluster together right. Similarly all misspellings, there is a high likelihood. It is not that there will be but there is a high likelihood that all misspellings will also be clustered together that is will be quite close to the actual spelling that has to exist that is for example if K.J Amit become let us say K.J Aman or something like that a m a n.

So it is Aman is still quite close to Amit that is assuming that we are sorting tens of thousands of records, misspellings would be quite close to the real spelling. Then, you slide a window that is take a set of n rows and keeps sliding it through the data base to see if there are any misspellings and then merge the misspellings and we have to make multiple passes until there are no more merges of records.

[Refer Slide Time: 48:55]

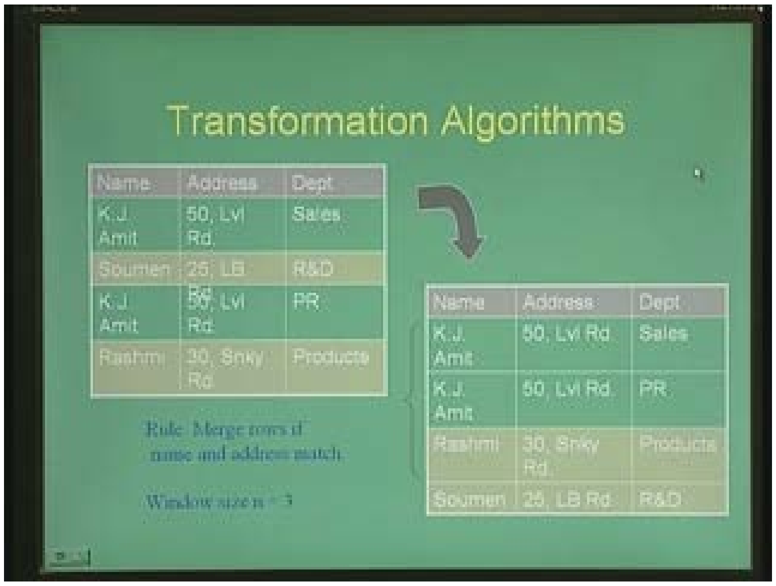


Transformation Algorithms

- Sorted Neighbourhood Technique for misspelling integration
  1. Identify a set of data values within a given row as key
  2. Sort table based on key
  3. Slide a window of n rows over the sorted table and merge data values based on rules. (Ex: Merge names if all other values like age, address, dept, etc. match)
  4. Make multiple passes until there are no more merges of records

So here is an example again where this is used for duplicate elimination in addition to misspelling detection where given this table here it is sorted like this here and given a window of size three we see that we see that there are duplicates within this window and we start eliminating them and the last algorithms that we are going to see for duplicate

[Refer Slide Time 49:15]



Transformation Algorithms

Name	Address	Dept
K.J. Amit	50, Lvl Rd.	Sales
Soumen	25, LB Rd.	R&D
K.J. Amit	50, Lvl Rd.	PR
Rashmi	30, Snky Rd.	Products

Rule: Merge rows if name and address match.

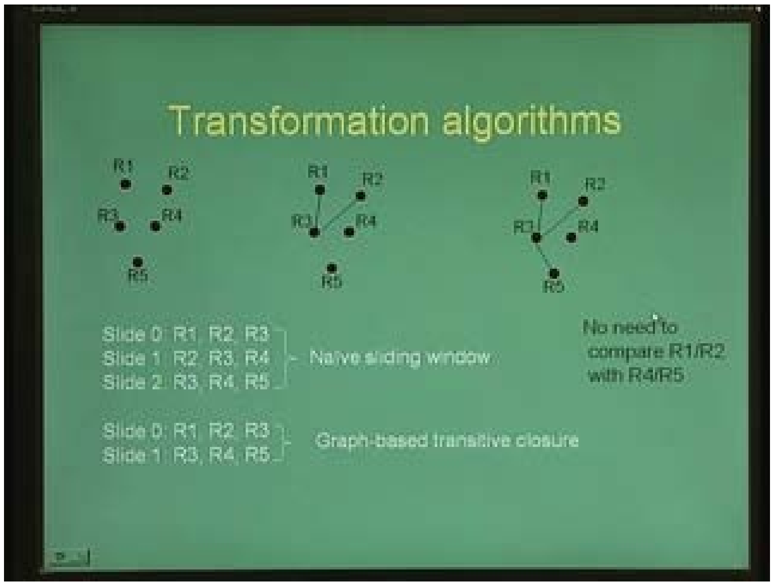
Window size is = 3

Name	Address	Dept
K.J. Amit	50, Lvl Rd	Sales
K.J. Amit	50, Lvl Rd.	PR
Rashmi	30, Snky Rd.	Products
Soumen	25, LB Rd.	R&D

elimination and also kind of and also for misspelling detection and so on what is called as the graph based transitive closure for (Refer Slide Time: 49:32) ………. this algorithm is more of improvement over the sorted neighborhood algorithm where it basically reduces the number of passes. The idea behind the algorithm is based on the notion of transitivity in an equivalence relation. For example: Let us say records R1 and R2. We see that R1 and R2 are duplicates.

So we establish a relationship that, R1 is a duplicate of R2 and so on and then we find that R2 is a duplicate of R3. We can definitely infer that R1 is a duplicate of R3 as well, because the duplicate is duplicate of relationship is a transitive relationship if a is a duplicate of b and b is a duplicate of c the then a should be duplicate of c what this means is when sliding the window once we established that a and b are duplicates, there is no need for us to compare a and c and so on. So that is the essential idea behind graph based transitive closure.

[Refer Slide Time 50:57]



Transformation algorithms

R1
R2
R3
R4
R5

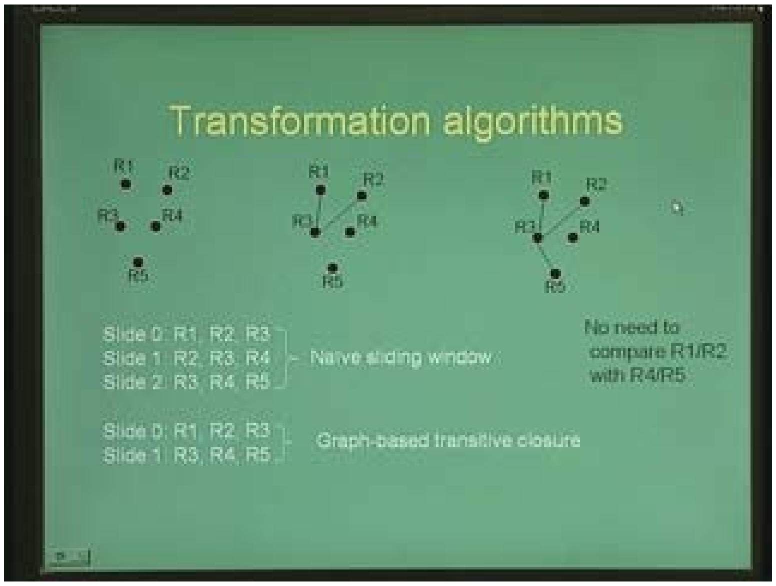
Slide 0: R1, R2, R3
Slide 1: R2, R3, R4 } Naive sliding window
Slide 2: R3, R4, R5

Slide 0: R1, R2, R3 } Graph-based transitive closure
Slide 1: R3, R4, R5

No need to compare R1/R2 with R4/R5

This slide here shows the schematic example where there are five different records and in a naive sliding window sorted neighborhood protocol, we start by comparing R1 R2 and R3, then R2 R3 and R4 and R3 and R4 R5. We are just sliding it with a window size of three. However when we encounter when we slide R1 R2 and R3. Suppose, we see that R1 R2 and R3 are duplicates. So R1 is a duplicate of R3 and R2 is a duplicate of R3 and so on. Now, there is no need to compare R1 and R2 with R4 and R5. It is sufficient if we just compare R3 with R4 and R5. So that is the general idea behind this algorithm which can reduce the number of passes in eliminating duplicates.

[Refer Slide Time 51:52]



Transformation algorithms

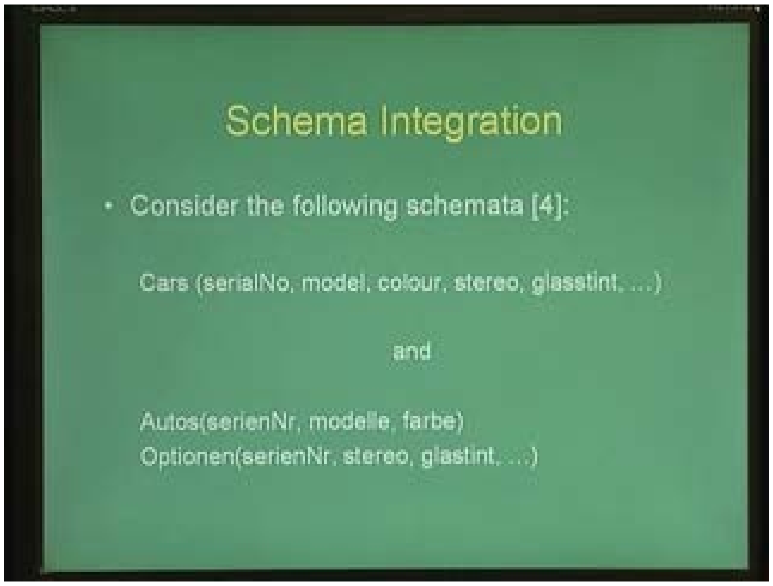
Slide 0: R1, R2, R3
Slide 1: R2, R3, R4 } Naive sliding window
Slide 2: R3, R4, R5

Slide 0: R1, R2, R3 } Graph-based transitive closure
Slide 1: R3, R4, R5

No need to compare R1/R2 with R4/R5

The next topic that we come in in the data cleaning and integration is the issue of integration itself that is, there are different OLTP data sources from which we are getting data. Most of which have different kinds of inconsistencies, sources of dirty data and so on and then we have to clean them and so on. But, after we have clean them we have to integrate them under a under a common banner and this integration can occur at two different levels which are shown here as data integration and schema integration. That is schema integration entails forming an integrated schematic structure based on the set of all desperate data sources that is you have data sources from counter A counter B counter (Refer Slide Time: 52:50)……… under one common schematic structure and data integration essentially entails cleaning and merging data from these different sources that is eliminating duplicates and merging all of them under this schematic structure.

[Refer Slide Time 53:08]



Schema Integration

- Consider the following schemata [4]:

Cars (serialNo, model, colour, stereo, glasstint, …)

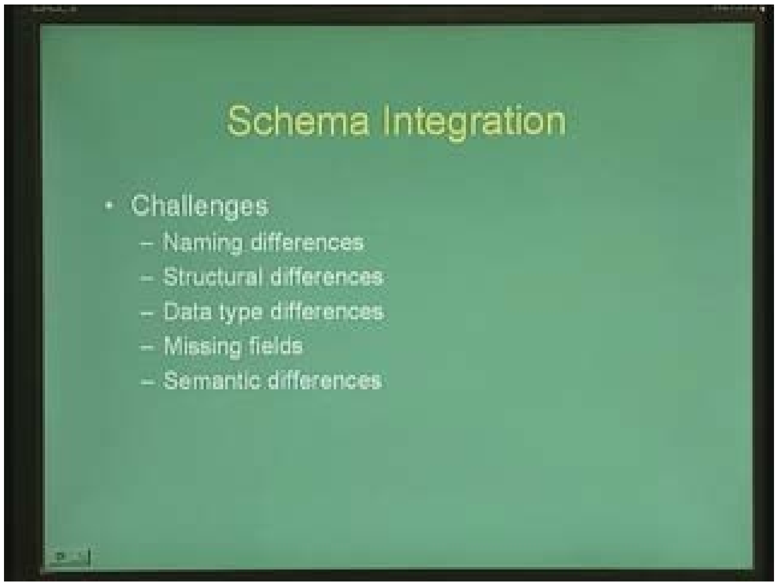
and

Autos(serienNr, modelle, farbe)
Optionen(serienNr, stereo, glastint, …)

Let us briefly look at what are the issues or what is the complexity now in schema integration as well and it is not as simple as it looks like. Consider the following schemata that is shown here or there is a schema that let us say the one of the OLTP sources used which is called which uses a table called cars and the car table has different fields like serial number, model, color, stereo glass tint so on and so forth. Everything is in one table. On the other hand, another retail center let us say uses two different tables for the same information and look at how these tables could be different. First of all, there are two tables rather than just one table then the names are different all these names are in German and these are in English. So cars are called autos and serial numbers are called as serine numeral or something color is called farbe and so on. So, it is not just the structure is different, the name of each of these fields could be different. The data types could be different and so on.

There are several different challenges that we have to face when performing schema integration that is there are naming differences, there are structural differences, data type differences, semantic differences, missing fields and so on and so forth.

[Refer Slide Time 54:42]

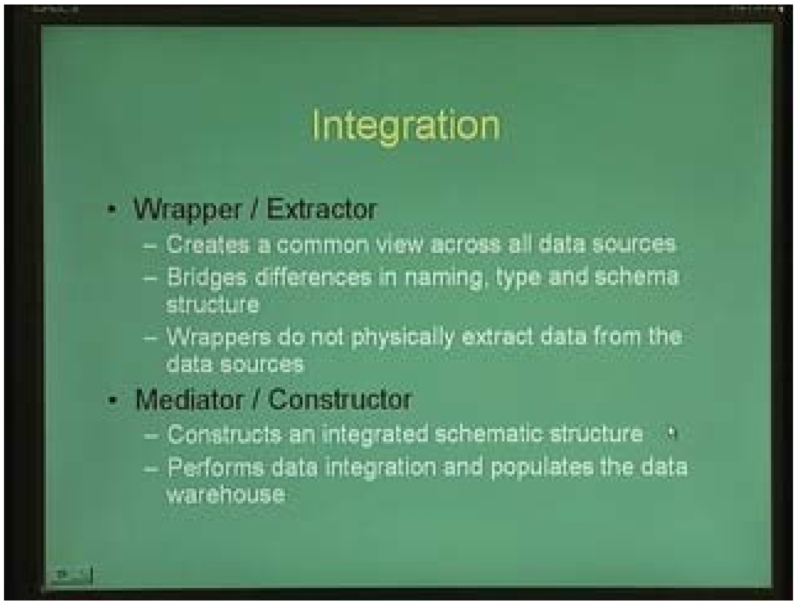


Schema Integration

- Challenges
  - Naming differences
  - Structural differences
  - Data type differences
  - Missing fields
  - Semantic differences

The generic architecture of schema integrator is shown in the slide here where it basically consists of two specific stages. One is the lower most stage is what is called a wrapper or an extractor. That is given a particular schema, the extractor maps all given schema into a standard schema set that is whether it is color or whether it is cars or autos or whatever it maps it onto a common schema set and then the mediator constructs the overall schema set or rather it looks at the federation of different data sources having the same schema and then handles query based on or constructs the data warehouse or handles query or whatever.

[Refer Slide Time 55:35]

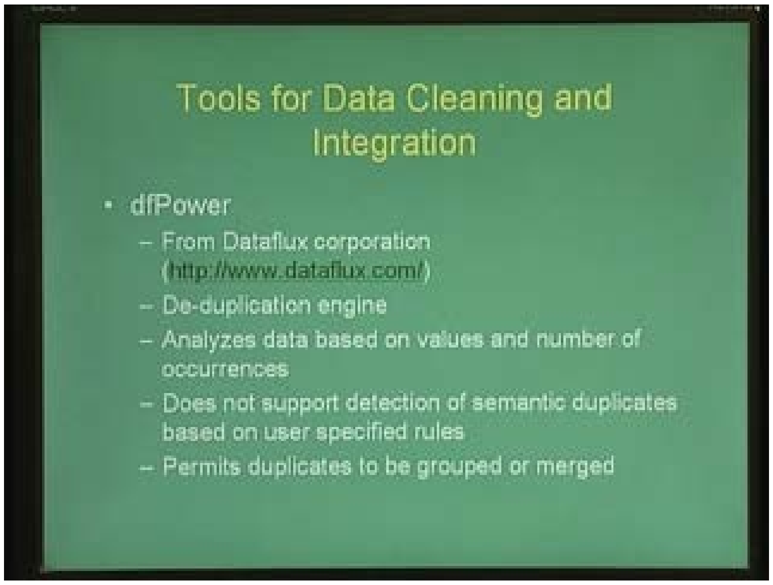


Integration

- Wrapper / Extractor
  - Creates a common view across all data sources
  - Bridges differences in naming, type and schema structure
  - Wrappers do not physically extract data from the data sources
- Mediator / Constructor
  - Constructs an integrated schematic structure
  - Performs data integration and populates the data warehouse

So the difference between a wrapper and an extractor is that an extractor essentially physically extracts data, creates a schema and physically extracts data from the data sources according to the schema, whereas the wrapper is just a logical wrapper. Similarly mediator is a logical mediator while a constructor physically constructs a data warehouses from all the extracted data sources.

[Refer Slide Time 56:04]



Tools for Data Cleaning and Integration

- dfPower
  - From Dataflux corporation (http://www.dataflux.com/)
  - De-duplication engine
  - Analyzes data based on values and number of occurrences
  - Does not support detection of semantic duplicates based on user specified rules
  - Permits duplicates to be grouped or merged

So there are different tools for data cleaning and integration and as I said there is no there is no common tool that can fit all possible requirements, but there are several tools that can make the life easier for sink. So, here are some examples d f power or eti star and SSA name and so on and so forth right. So let us summarize what we studied in this session. We looked at the important differences between OLAP and OLTP queries. What are the difference between analytical queries and transactional queries? What are the characteristics of OLAP (Refer Slide Time:) ………… of a data warehousing system where you take data from different OLTP fields and take them through data cleaning and integration phases before populating the data warehouse and then of course, there is a feed back and back flush and so on and this itself is a major issue with the machine.

So that brings us to the end of the session.