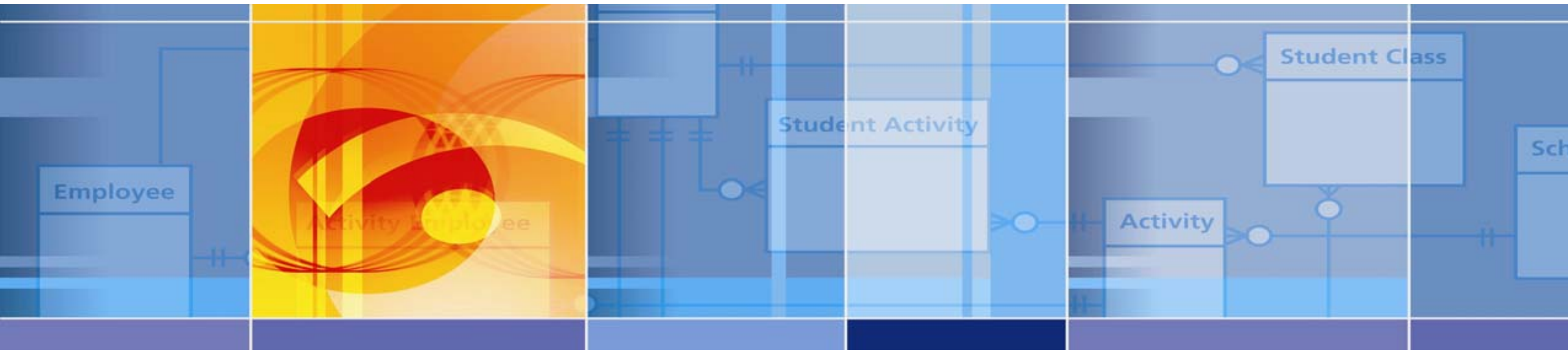


The Optimal Reference Guide:

What's Behind Your Data Warehouse?

Data Warehouse Series – Part II

Extraordinary insight into today's education topics

Steven King, ESP Solutions Group
Alexander Jackl, ESP Solutions Group

ESP Solutions Group

Table of Contents

Foreword

Data warehouse and **data quality**, these are two constructs that go together well. In fact, the motivation for implementing a data warehouse is often the expectation of a boost in data quality for better informed decision making. The idea is that if an organization can get all of its important data together in one place, organized in a common way, and following standard rules, then the data will reach the required heights of quality to drive better analysis and better decision making. I think that is a reasonable expectation.

Getting there is the challenge.

Our focus today builds on the context created in a previous paper, entitled "*What's Really 'In Store' for Your Data Warehouse?*" That paper described the history, tradition, and goals for a data warehouse. This whitepaper details more of the design, structure, and configuration parameters of a data warehouse.

A data warehouse does not create quality data, but both the discipline that is required for integrity within the data warehouse and the cross functional dependence upon a valued data source to guide decision making throughout an organization does drive quality data. Forcing an organization to create a data model that pictures the relationships among its data areas is the classic first step. Establishing common standards (a metadata dictionary) for the contents of the data warehouse is another significant accomplishment. Specifying the periodicity of the activities that consolidate the data into the data warehouse imposes the traffic cop function that is necessary for control. Managing access to the data within the data warehouse ensures compliance with FERPA and freedom of information mandates.

This whitepaper recognizes the importance of structure and design for a data warehouse to fulfill its promise of improved data quality.

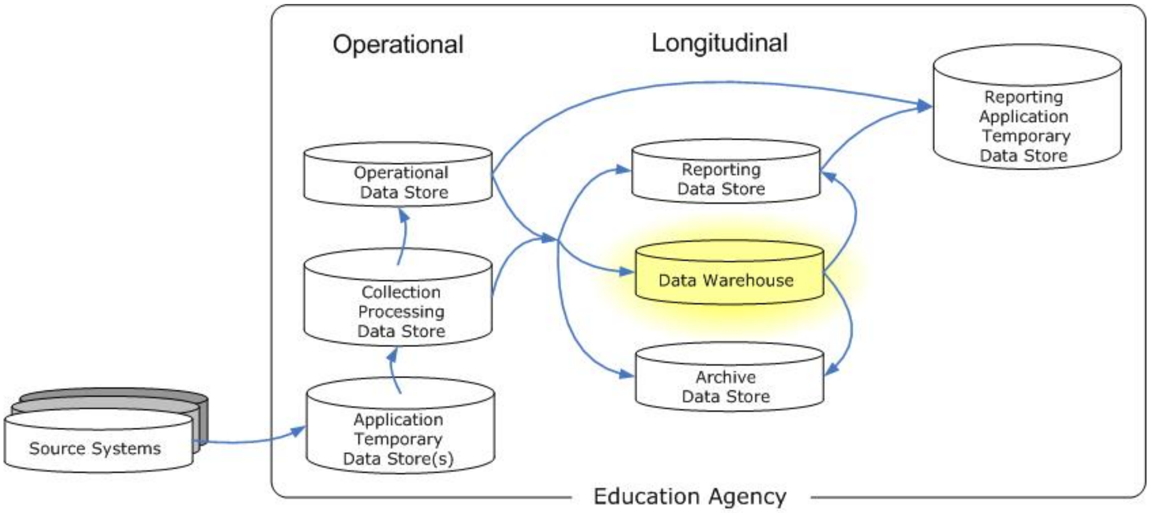
There is a lot more to a data warehouse than just building some files to consolidate disparate data systems.

- Scope: figure out your information needs and design to meet them.
- Reality: initially be practical about the questions to be answered and the data to be consolidated.
- Scalability: ensure that the selected data warehouse can grow to meet not only increased demand throughout the organization, but also additional data sets for even better analysis and answers to questions that were not originally planned for.

One concept you will see downplayed in our whitepapers is data mining. Education agencies do not have the time or resources to build the ultimate data warehouse. We suggest selecting the one that meets your needs today and that can grow as your requirements for improved analysis and data driven decision making grows. After all, the goal is to provide quality and timely data for improved decision making.

Introduction

This is the second in a series of ESP Solutions Group whitepapers on the topic of data warehousing and longitudinal data systems. In the first paper, ESP identified the various functional data stores within a local school district or state education agency and also tried to put the data warehouse in its proper context. In that paper we identified two longitudinal data stores: the reporting data store and the data warehouse (see below). In this whitepaper we will discuss the design considerations for these two data stores.



The reporting data store and the data warehouse are very similar. In most cases the questions to be answered by the reporting data store are known prior to its design. The data warehouse, on the other hand is built to support analysts doing work where the questions are not known. In both cases the data tend to be loaded in large batches and the systems need to be tuned for analysis and reporting.

These distinctions are functional distinctions. In many cases, there is a single database supporting both functions. Since many of the design characteristics are the same, we discuss them together in this document and highlight specific differences where they occur.

Longitudinal Data Store Design

This section is designed to be short and sweet — covering the entire arena of longitudinal data store design, so as to set the context for examples which dive into the details.

Strategic Design

There are two ways to begin building and implementing longitudinal data stores. The first is a bottom-up approach. In this approach, a particular policy area is selected and a complete solution is built and implemented. Then another area is independently selected and developed. Each of these "mini-warehouses" or "data marts" has its own set of stakeholders and analysis needs.

An alternative is a comprehensive top-down approach. All of the procedures of interest are modeled and the common dimensions are defined in a coordinated, enterprise-wide solution.

ESP Insight

Our experience is that a pure application of either approach will fail. A more blended approach is required.

Our experience is that a pure application of either approach will fail. A more blended approach is required. In the blended approach, the crucial processes are identified, and facts and dimensions are developed both locally for the given area and with the larger picture in mind.

In our examples, we designed the attendance set first, but revised the structure after exploring the assessment structure. In education, the problems or processes are seldom isolated and distinct but often overlap with other processes.

We want to encourage and facilitate as much "drill across" as we can. That requires common dimensions and granularity. Clearly, it is best to design with an enterprise focus.

ESP Insight

Successful implementation requires progress and "quick wins."

Picking an area with a moderate profile can gain support for the resources needed to continue. Initial pilot implementations have enough pressures without having a spotlight on every action. On the other hand, successful implementation requires progress and "quick wins." If you select a process for which nobody really cares, obtaining resources to continue the effort may be difficult.

As we will see in the examples below, we made changes to our first attendance implementation based on the needs of the assessment system. High profile systems are less tolerant of these adjustments. You don't need to lock yourself into a solution before a fuller enterprise look can be developed.

Policy issues of interest are most likely local. That is, you may need to go through this exercise to identify your data needs. An outsider cannot tell you what data you have or what your analysis needs are. There are some common design patterns however, and solutions in one context can be useful in others. Experts can help you with the system design and set-up. They can also help you around many of the obstacles others have run into.

Aggregated vs. Record-Level Data

The term "disaggregation" is a misnomer. Aggregated data contains less detail than its source. There is no way to take aggregated data and split it into more detailed aggregates without starting from the most detailed information and building the aggregate view back up. This is an important principle to understand when building a longitudinal data store. It is useful to have the most detailed record-level data possible when building your data store. Once you have this record-level data, you can build aggregates freely based on any dimensions available to you.

Source Level of Data

Where does data come from? It is essential that all data clearly be tagged with the source level they originated from. Attendance data captured directly from the classroom is far more reliable than attendance data received from regional service center. The usual rule is that every step removed the source of data is from the actual occurrence of the data (student's actually attending the classroom) the less reliable the data.

Periodic Categorization

In a longitudinal data store you are tracking data over time rather than data at a moment in time. However, individual data elements often change at different frequencies. How data changes are recorded in the longitudinal data store has a great impact on the accuracy of the data store and what kinds of analysis can be performed. For example, a data store record for a student may contain information about the student's name, birth date, gender, ethnicity, grade level, program participation, etc. The student's name might change, but very infrequently. Gender, ethnicity and birth date will not change for a student. However, program participation and grade level could change several times throughout a school year. There are several strategies for dealing with this issue.

ESP Insight

How data changes are recorded in the longitudinal data store has a great impact on the accuracy of the data store and what kinds of analysis can be performed.

1. One popular strategy is to store a complete snapshot of all of the data at the lowest level of periodicity. This is the simplest method and is often a good enough solution. However, if you have data that could change at a frequent rate (such as daily), then you are likely to waste space storing many copies of the same data.

2. Another strategy is to separate the data elements into groups of similar periodicity. Your design may accommodate this by having a group of data elements that almost never change, a group that changes regularly (once or twice a year), and another group that changes frequently (daily).

This is an issue to consider carefully when designing your store. Most solutions will end up being a careful mix of these strategies.

Directory Snapshots

We frequently hear the question, how often should I store a snapshot of my student/teacher/class directory information? The answer is simple. Since this directory information is usually in support of analysis of other metrics such as test scores, accountability measures, attendance rates and discipline rates you need to take snapshots as frequently as you need in order to properly analyze those metrics. If you are primarily analyzing assessment data, then your directory information should be at least as up to date as the test score data.

ESP Insight
How often should I store a snapshot of my student/teacher/class directory information? You need to take snapshots as frequently as you need in order to properly analyze those metrics.

Data Connectivity Over Time

For true longitudinal data analysis data must be able to be connected over time. For instance, you may have a series of test scores from year to year, but do you know for sure that the John Smith who scored well on the test this year is the same John Smith who scored poorly last year? With only names to relate data, the link over time will be tenuous. The best way to link data over time is to use a unique identifier such as a state student identifier. Most states have adopted student identifiers or are looking to adopt them soon. Incorporating a state student identifier into the longitudinal store is crucial.

ESP Insight
Incorporating a state student identifier into the longitudinal store is crucial.

Auditability Requirements

This is a function of policy and common sense. You need to come to some sense of which transactions should be tracked and also what data will need to be rolled back to a former state should an error occur. Audit tables, transaction tracking tables, and archives are crucial components of an enterprise data system. They are not glorious or exciting but they are essential — and a core part of a disaster recovery plan.

Data Retrieval

These sections are about designing the physical model ahead of time to optimize how it will be used. This is often overlooked and then bitterly regretted.

Retrieval Speed

Your table design needs to be optimized by how often the table will get hit and how fast you need the data to be retrieved. It would make sense to layout what your ten most common queries will be and walk through the chain of data calls that pulls that result.

Indexing Schemas and Requirements

You need to design how you are going to use the database before you design the physical model. You will want to design indices that optimize your most common queries. The good news is that this can be adjusted post design.

FERPA issues

Each state and sometimes each district have different policies and regulations on how to handle privacy and accountability for data. No matter how "loose" your organization is with this you must account for FERPA issues. This issue comes to the forefront when we discuss Reporting and Analytics in one of our next papers. Our recommendation is that you have the most granular, unconstrained data possible in your core data stores. Then you design display tables, data marts, view cubes, etc. to show data that has been properly parsed for "n" value constraints (show no category of people with less than "5" students) or role- and organization-based constraints on who gets to see that view of the data.

Data Warehouse and Reporting Data Store Differences

In this paper, the discussion applies to both the reporting data store and the data warehouse data store. It would be logical for a reader to ask, "What's the difference?"

The distinction is functional and it is conceivable that a single store could serve both purposes.

The reporting data store is designed to support the needs of known reports, analysis, or query needs. In most cases, the questions to be answered are known in advance. Users of these systems will be receiving reports in standard report formats or viewing data on web sites where the analysis is predictable.

The data warehouse store is designed to support the needs of data analyst who will "exploring" new arenas. Sometimes called "data mining," these analysts will be digging through the data looking for new patterns and relationships in unpredictable ways. They will be doing the types of drill across analysis described above.

Because the questions may not be known in advance, or analysts will be making new comparisons, there is a heightened need for rich and complete metadata in the data warehouse. Analysts linking data together in new ways need to know that these connections are valid. They also need to know that two sets of facts are for the same time period, student population, school, etc. They need very descriptive information about each of the dimensions available for a given set of facts. Without the rich metadata required to support the data warehouse data store, it is too easy for users or analysts to make invalid comparisons.

ESP Insight

Because the questions may not be known in advance, or analysts will be making new comparisons, there is a heightened need for rich and complete metadata in the data warehouse.

The time dimensions in education data can be problematic if not correctly matched. Education facts can be snapshots taken on a particular representative sample date. October 1 enrollment or December 1 special education child counts are obvious examples. Alternatively, education facts are accumulated counts summarized over a time period. Total school year dropout counts or average daily attendance are examples of period summary data. In many cases, the snapshots are intended to be representative of a particular time period.

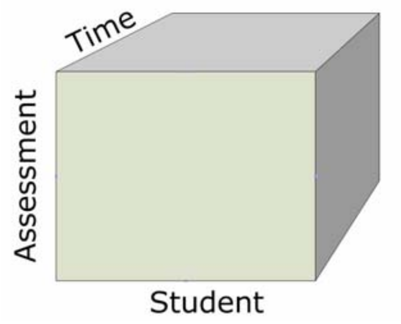
Data from different time dimensions can be combined, but the differences and the potential shortcomings should be explicit. For example, we often compare special education percents calculated using December 1st special education counts divided by October 1st enrollment counts. Per pupil expenditures are calculated as the ratio of October 1st enrollments to accumulated expenditures over the school year. The metadata must make it quite obvious, what time period is represented by the data or a specific calculation

Tactical Design

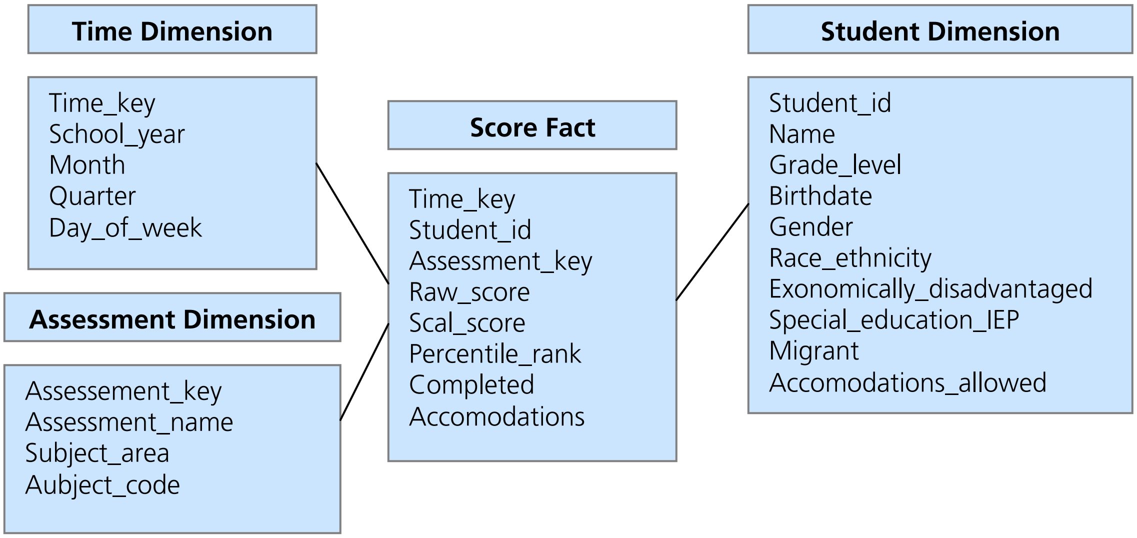
Imagine a state superintendent who describes one of the core functions of the SEA as:

We administer assessments to students and measure student and school performance over time.

The data staff should identify key concepts in this statement: assessments, students, and time. We can think of this as a cube of data with labels on each axis of Time, Assessment, and Student. Any point inside the cube represents the intersection of the three axes, i.e., the results for a particular student on a particular assessment at a particular point in time. The points represent the measures of interest to the education enterprise.



This is basis for the dimensional data model. Another common name for this model is a star schema. The diagram for these models tends to be a large central table with a collection of attendant tables connected radially around it.



The central table is referred to as the **fact table** and the others as the **dimension tables**.

The fact tables are where the measures of interest to the enterprise are stored. The fact tables tend to be quite long, that is, have lots of rows. In the above example there would be one row for every assessment taken by every student. Assuming a half a million students taking assessments in four subjects, then 5 years of data would need 10 million rows of storage.

ESP Insight

The fact tables are where the measures of interest to the enterprise are stored. The dimension tables are where the descriptions of the dimensions are stored.

In virtually all of the queries, we will be selecting thousands or millions of rows to be summarized into a few dozen records for the answer set. Overwhelmingly, the most useful way to do this is by adding them (or averaging which requires adding and counting). Thus, the best facts are numeric and additive. For non-additive facts, the best we can do is to summarize via counts.

The dimension tables are where the descriptions of the dimensions are stored. The best dimension tables tend to be wide with many attributes (i.e., lots of columns). The attributes are discrete and used as sources for pick lists, filters, row and column headers, etc. Since they are used to describe the dimension, they are best if they are text. Any inherent hierarchies would also be defined in the dimensional tables, i.e., a school dimension would define the districts and optionally regions to which the school belongs.

Attributes are the source of all the interesting constraints. Attributes provide row and column headers in reports. Consequently, the value and strength of the repository is largely determined by the quality of the dimensional attributes. Time should be spent identifying all the attributes, ensuring values in the attribute fields are complete, finding good descriptive text, and quality assuring these values.

ESP Insight

Time should be spent identifying all the attributes, ensuring values in the attribute fields are complete, finding good descriptive text, and quality assuring these values.

Education Nuances

As with most things in education, we cannot draw a hard and fast line between dimensions and facts. The AYP status of a school may be treated as a fact about the school in one analysis, and treated as a dimensional characteristic (i.e., grouping or filtering condition) in another.

It is not uncommon to summarize a set of facts or generate a calculated fact that will be used as a dimension in later queries. For example, we can calculate a poverty level for schools based on enrollment and poverty level student facts. This value may be stored and used later as a dimensional characteristic for the school. Or we can further summarize the value into a set of "buckets" storing the bucket value (0-5% poverty, 5-15% poverty, 15-25% poverty, etc) as a dimensional characteristic.

Sometimes we do analysis only on dimensions and counts. We join teacher dimension tables to student dimension tables through courses. For these analyses, the course acts only as a joining mechanism – a kind of fact table in the traditional star schema, but without any facts. It is a "factless" fact table.

ESP Insight

Longitudinal data analysis systems came from the business environment where most analyses of interest can be tied to well-defined fiscal measures. The fact-dimension concepts are useful for discussion purposes, but longitudinal data system implementers need to be flexible when applying these concepts in education.

Longitudinal data analysis systems came from the business environment where most analyses of interest can be tied to well-defined fiscal measures. The fact-dimension concepts are useful for discussion purposes, but longitudinal data system implementers need to be flexible when applying these concepts in education.

Typical Query Structure

The typical longitudinal analysis begins by picking a fact and selecting one or more constraints and then one or more attributes to summarize. For example, we could select average scale score; filtered by school year=2004-05, subject=mathematics, and district=North Fabulous; and then reported by grade_level.

It is not necessary that readers know Structured Query Language (SQL), but the SQL from the example below is typical and illustrative. In most cases, the user's analytical tool can build this SQL statement behind the scenes. Knowing the components of the query, however, can be helpful in understanding the reason behind the star schema design.

```
1   SELECT p.grade_level,  AVG(s.scale_score)          ← SELECT field list
2   FROM  score s, student p, assessment a,            ← list of tables with aliases
3            time t,  school d                         s, p, a, t, and d
4   WHERE s.student_id = p.student_id                  ← join condition
5      AND s.assessment_key = p.assessment_key         ← join condition
6      AND s.time_key = t.time_key                     ← join condition
7      AND s.school_id = d.school_id                   ← join condition
8      AND t.school_year = ‘2004-05’                   ← filter constraint
9      AND a.subject = “Mathematics’                   ← filter constraint
10     AND s.district_name = ‘North Fabulous’          ← filter constraint
11  GROUP BY p.grade_level                             ← group by clause
12  ORDER BY p.grade_level                             ← sort order of result set
```

The query starts on line 1 with the list of fields we want included in our result set — in this case, the grade level and the average scale score at that grade. In virtually every query, the result set consists of fields that become row headers and aggregate facts. In most cases, the row headers will come from the dimension tables and the aggregate facts are summarized from the fact table.

The FROM clause on lines 2 and 3 lists the tables that will be used in the query. In this example, we have included an alias for each: the letters s, p, a, t or d. The alias is used to distinguish the student_id in the student table from the student_id in the score fact table.

Lines 4-7 define how records in the score fact table are matched to the correct records in the dimension tables. In the first case, we are saying the student demographic information for a particular test score result can be found by taking the student_id from the test score (s.student_id) and finding the matching student_id in the student table (p.student_id), likewise for the next three lines.

There will be many scores in the score table for a particular student in the student table. Most relational databases are very fast at this type of look-up. The advantage of the star schema approach is that **all** the query actions are single level look-ups from the central table.

ESP Insight

The advantage of the star schema approach is that ***all*** *the query actions are single level look-ups from the central table.*

Lines 8-10 define the constraints on the records to be returned. Only records that have a school year of “2004-05” in the time dimension will be summarized. Likewise, only score records that match assessment records with a subject of “Mathematics” will be used.

Now that we have all the records from the tables, we will group scores by the student’s grade level and average them. The final step is to sort the results by grade level.

To see why grade 4 is below the others, we may wish to dig deeper to see if there are performance differences by gender (commonly referred to as **drilling down**). In our example, this would be a simple addition of a.gender to the SELECT field list and the group by and order by clauses. Drilling down is simply adding another column to the query results.

Implementation Procedure

Here are some quick and dirty tips to implementing. For some of us they may seem obvious, but we have seen many attempts to implement data warehouses that ignored each and sometimes all of these tips!

Design Tools: Don't skimp.

You don't want to skimp on design tools. Even though it is imperative that the tools not drive design or architecture, the tools you choose will influence how quickly and how elegantly you can implement your requirements.

Database Tools: Yes, the database makes a difference

Likewise the database makes a huge difference. The key: do your homework. Many of the enterprise class databases can handle the loads and the transactions needed but make sure that the database software you will use can handle the scope of your project — not just now but the vision five years from now. If you intend to have daily inserts from the teacher/classroom level you need to vet that that works.

ESP Insight

There are so many pitfalls and dead ends that you will certainly save money by bringing in experience early.

Making the Right Choices early: Use experts, it will save money

Experience is the key to success in this kind of endeavor.   Although there is a short term temptation to design it yourself, only do that if you have personnel with deep experience in building k12 data warehouses. There are so many pitfalls and dead ends that you will certainly save money by bringing in experience early.

Scope and Sequence

How you scope and sequence your design is critical. To be most effective you want to think about the data needs of the entire enterprise. Start implementation at a very small scope and then grow from success.

The Organic Model: One Data Mart at a time.

Often selecting a tight domain to start with is very smart. Accountability, assessment, reporting, or a particular program are all candidates for an initial build.   This should be driven by the program and business needs currently on the plate.

The Waterfall Method: The Big Dig of Database projects

Don't do this – don't design the entire database for the lowest level. It will take forever and there will be little reward for a long time to whoever is paying the bill. Design the whole thing at the 100,000 foot level but then start to implement one small chunk of it at a time.

Table Design

When designing the tables for a longitudinal data system, there are four steps that one typically needs to go through. The order of these steps is important for an efficient system design. It is common to pick a policy area of interest, build the star schema for that topic, and then repeat the cycle for another area, thus growing your longitudinal system in an organic fashion.

The four steps are:

1) **Pick a process to model**: The process should be a major operation of the enterprise for which there is support and data to feed the data store. Examples include student performance, school finance, attendance, dropouts and graduation, etc.
2) **Select the level of detail for the facts** (the *grain* of the table): Decide the atomic unit of data that will be stored in the fact table. Typical grains could be October 1 snapshot; annual, quarterly, or monthly summaries; individual events (i.e., assessments or dropouts); student, school, or district summary; etc. The next step cannot be done until this step is completed.
3) **Choose the dimensions**: Time and school/district are almost always included. Student level data would have a student demographics dimension – likewise for staff data. Finance data would have dimensions (and hierarchies) for the expenditure function and object categories.
4) **Choose the facts for the fact table**: Select the measured facts that will go into the fact table. Again, shoot for additive, numeric values. For example, percents are numeric but you can't add or average them. Try to get the raw data that generated the percents.

Try to be consistent about the granularity of the fact tables. When the fact tables have the same granularity, then the dimension tables of the two processes can be shared. The effort that goes into cleaning the attributes generates maximum benefit. That is, we want one school/district dimension table in the warehouse, one expenditure function/object dimension, etc.

ESP Insight

Try to be consistent about the granularity of the fact tables. When the fact tables have the same granularity, then the dimension tables of the two processes can be shared.

It may be easiest if we pick some example processes and work through them in more detail. We will start with the relatively simple student attendance and then move to student performance on state assessments. In each case, we will go through the four implementation steps outlined above.

Example 1: Attendance

We will start with a relatively simple process: student attendance. We will show how the dimensions and facts are selected, tables designed and the data populated. We will also give example extensions while warning about potential pitfalls.

In this example we will:

- highlight the development of the time dimension,
- talk about why and how one would want to break the normal relational database normalization rules,
- wrestle with the issue of tracking school characteristics over time, and discuss the trade-offs that designers need to make when defining the grain of the data warehouse fact tables.

The key concepts are highlighted in the "ESP Insights" boxes in the side margins.

Process to Model

Let's look at student attendance and absence patterns. To keep the example simple, we will assume we are modeling a single district and that the district has a single calendar used at all schools in the district.

Teachers at the elementary schools take attendance twice a day. The middle schools and high schools take attendance in every period. The middle schools run on 7 period days. Seventh graders have lunch during period 4, while the eighth graders have lunch during period 5. The two high schools run on a block schedule – 4 blocks on each of two alternating days. On 'A' days student attend periods 1-4 and on 'B' days they attend periods 5-8.

All teachers enter attendance on the computer in their classroom attached to the school's student information system. Only absences are recorded in the SIS, not tardies. All schools in the district use the same student information system and all of the school servers update the district server nightly.

Daily attendance is not used to drive district funding in this state. Rather, this state funds schools and districts on monthly snapshots of membership. The implication of this is that the actual attendance numbers in the SIS are not audited by the state. The local school board has, however, set improving attendance as one of their goals for the district.

The district tracks five kinds of absences: excused (i.e., sick with note), participating in a school sponsored activity, excused by parent for family activity (family vacation or trip), unexcused, and suspended. The third category accounts for those students who are pulled from school with the approval of their parents. For policy purposes, this is different than students who are sick or student that skip classes without parent permission.

All absences are initially recorded as unexcused. Students have two days after their return to school to clear up any unexcused absences. After that the absence stays recorded as unexcused. Unexcused absences are what the state counts for the truancy rate calculation.

Select Level of Detail for the Facts

Aggregating data can be likened to sorting gravel into various piles — maybe size, or color. To disaggregate, we have to resort the gravel into more or different categories — size **and** color. At some point we are dealing with the individual pieces of gravel and not collections. That is the grain of the table — the level of detail below which we cannot summarize.

In most cases, the fact tables are some level of summary of the source data. They could be simple year-end summaries, monthly summaries, single day snapshots, or as detailed as daily or hourly totals, but still a summary. As we will see later, each fact gets tied to a point or period in time and represents the summary for that time period.

This decision about what level of detail to collect in the fact tables will affect the type of analysis that can be done with those data.

The Magic of Disaggregation

Disaggregation is a misnomer – like magic, it is based on illusion and misdirection. Disaggregation implies that a person can take the numbers or statistics on a report and somehow break that down into more detail — i.e., undo the aggregation that created them. You don't actually split an existing summary into pieces; you have to go back to the original data and re-summarize the data at a more detailed level. You must have the detail data to do this.

ESP Insight

This decision about what level of detail to collect in the fact tables will affect the type of analysis that can be done with those data.

In our attendance example we have various choices for the granularity of our table. We could collect:

- monthly total by grade and school of the aggregate days in attendance of students, or
- the total weekly attendance by school and student demographic group, or
- we could sample and get total daily attendance at each school and grade on Monday of each week, or
- for each student absent, a percentage of the day absent, or
- a reporting of absences period-by-period, or
- we could get for each student, for each period an accounting of absent, tardy, or present, or
- a report by student of present, tardy, or absent for each class period with the class subject and teacher.

These types of breakdowns can continue almost without end. For example, we could look at a course location dimension. Do portables have the same attendance rate as in the main building? Does weather affect attendance? How about in combination — class rooms with southern exposures, on warm spring days, after lunch?

Clearly, each of the more detailed levels will support more detailed analysis, but they also come with an increased data collection and storage burden. Systems can be automated to collect and feed the data warehouse from online attendance systems.

In most cases, we want to populate the fact tables with the most detailed data available. Not because we want to ever look at these low-level individual records, but because we want our queries to be able to slice through the data in very precise ways.

Since students have two days after they return to clear up their absence (excused, school activity, etc.), there will be a three-day delay after students return for the data to be loaded and stable. Students can occasionally be sick for week or more with the clean-up occurring for a few days after. The data loading procedures should not load the data store until the absence status is resolved. In any case, analysts should not attempt to work with absences occurring in the last two weeks—the data are not stable within that time frame.

While it would be interesting to see if math classes have a higher or lower attendance rate than social studies classes, the district suspects the rate is more likely affected by the teacher than class subject. Also, they don't have a course-to-subject categorization that works for all classes consistently.

ESP Insight

Any data warehouse designer has to be aware of the potential misuses of the data they are making available. Sometimes politics and other non-technical concerns will influence design decisions.

The North Fabulous school board decided, for political reasons, not to track attendance rates by individual teacher. Any data warehouse designer has to be aware of the potential misuses of the data they are making available. Sometimes politics and other non-technical concerns will influence design decisions. These can shift as people learn to trust that the data "are what they are" and administrators learn to use the findings appropriately.

To simplify our example, we are going to set the grain of the table as:

For each school, for each student, for each day absent, the fraction of the day absent.

We decided not to track absences on a period-by-period basis. That is, if a student misses one period during the day, we won't be able to tell if it was 1st period, 4th period, or 7th period. We won't be able to tell if half-day elementary students were absent in the morning or afternoon. These are good analyses to do, but for our first example the differing schedules at the different schools adds a level of complexity that gets in the way of the lesson.

At the elementary schools, where attendance is taken only twice a day, an entry in the fact table will be either 1.0 or 0.5. Students at the middle schools attend 6 periods per day, so their absence entries could be any of 0.17, 0.33, 0.5, 0.67, 0.83 or 1.0. Each class at the high school is 0.25 of the day.

These numbers are readily available from the district attendance systems. These facts are numeric and summable.

We have decided to only record whole period absences. Therefore, when a student is not absent from any class, their "portion of the day absent" is zero. There is little point in cluttering our fact table with a bunch of zero records indicating nothing happened. Consequently, we do not record anything when a student is present all day.

Choose the Dimensions

The next step is to decide what characteristics of the absences we wish to track. As these are being decided we will flesh out all the attributes of those characteristics. These will become the dimensional tables in our warehouse.

So far we have identified four dimensions: absence type, time, student, and school. Each of these dimensions will be fleshed out below. As we look into our analysis further, we may identify additional dimensions.

As we are defining the dimensional tables for our Attendance example, we want to be thinking about other uses for these dimensional tables. We will spend considerable time and energy on these tables so we want to maximize their use.

ESP Insight

As we are defining the dimensional tables for our Attendance example, we want to be thinking about other uses for these dimensional tables. We will spend considerable time and energy on these tables so we want to maximize their use.

Absence Type Dimension

The absence type dimension in our model is a fairly straightforward look-up table with additional information about each of the absence types. This table will contain codes to be used in the fact table, it contains the descriptive text that will be used for row headers and column headers in any reports, and it contains additional flags for how the types may be grouped.

After reviewing the policies of the school district and the issue of concern for the district administration, five categories of absences were identified. These have a coding structure that is meaningful to teachers and that is in the district student information systems. The five types and their codes are:

E – Excused: medical reason or family emergency
A – School sponsored activity or field trip
P – Parent removal for family activity
S – Suspended
U – Unexcused

The code value is what is stored in the fact table for each absence and will be the primary key for our dimensional table.

The district treats the parent removal and the unexcused absences similarly. These both count as truancies in the state calculation, make-up credit is at the discretion of the teacher, and course credit can be denied if too many absences of these types accumulate. We will want to facilitate this type of analysis.

Let's think about how our dimensional fields will be used. They will provide headers for table rows and columns in summaries. It is nice to keep column headers short so tables don't get too wide; row headers can be more descriptive. The contents of a dimensional column can be queried to populate controls on a filtering screen. For example, it is easier to choose a checkbox labeled "Excused" or "Activity" rather than one labeled "E" or "A". Descriptive text from the dimensional table can be used as the sources for tool tips on report screens. We may even want a complete definition for each of the options.

Most consumers of the products of these systems are lay staff. The system will be more useful if the choices and reports are descriptive and clear. There are only five records in the AbsenceType table so several, even dozens, of fields in this table are not going to be a size constraint on the system.

Both long and short description fields will be created. The short description will mostly be used as a column header while the longer description can be used as row headers in reports.

Two columns are added to the AbsenceType table to handle the tracking of the absences that count as truancies. A descriptive field is added to be used for headers on summary reports and a Boolean field is added for use with a Truant check box on a filtering dialog.

Our AbsenceType table now looks like:

Code	Long_Description	Short_Description	is_truant	truant_flag
E	Excused: Medical Reason or Emergency	Excused	Not Truant	false
A	School Sponsored Activity or Field Trip	Activity	Not Truant	false
P	Parental Removal for Family Activity	Parent	Truant	true
S	Disciplinary Suspension	Suspension	Not Truant	false
U	Unexcused Absence	Unexcused	Truant	true

The Time Dimension

ESP Insight

Every fact record in every longitudinal data store has time as a dimensional attribute.

Every fact record in every longitudinal data store has time as a dimensional attribute. The facts may represent a single point in time, as in the case of October 1 enrollment snapshots. Or the facts may be a year-end summary representing an entire school year. Or the facts may be events that occur at specific points in time like dropouts, or, for our example, student absences.

Since a lot of the analysis the school district has been interested in relates to time, we'll spend considerable effort making this a rich dimensional table. In the time table we'll have a record for every day of the year; ten years of data is only 3,650 records. We'll have clarifying information about each day: day of the week; holiday flag; A-day or B-day; in-service day—half day or full; parent-teacher conference day, etc. We will have fields indicating to which school year the day belongs, to which quarter or semester, and whether the state assessment is occurring. We can have long and short version for each. Additional fields can be added to account for any special type of day a school district may have.

Analysts will be able to study absentee and attendance rates by day of the week (Monday vs. Wednesday vs. Friday). They will be able to look at the impact of holidays and three-day weekends on absentee rates. They will be able to look at the impact of in-service days on attendance (for example, are students more likely to be absent on half-day in-service days?).

#	Field	Description	Example Data
1	Date	The date to be matched	30-Sep-2006
2	School_year	The school year to which the date belongs, Summer belongs to the following school year	2006-07
3	Day_of_week	The day of the week for this date	Monday Thursday
4	Holiday	Is the date a legal or school holiday where no students or teachers are expected to attend. Y for yes, N for no	Holiday Non-Holiday
5	Block_sched_day	Is this an 'A' day or a 'B' day at the high schools that are on the block schedule	A-day B-day
6	Instructional_day_short	Short descriptive name for the type of instructional day where students attend classes at least part of the day.	Full-day Part-day Non-Instruction Early-Release Cancelled Make-up
7	Instructional_day	Full descriptive name for the type of instructional day where students attend classes at least part of the day	Full Instructional Day Non-Instructional Day (i.e.,, holiday, week-end, teacher work day)
8	In_service_day	Is this a teacher in-service day, i.e., students don't attend.	Full-day In-service Half-day In-service Non-In-service
9	Semester	The school semester to which this day belongs	1st 2006-07 2nd 2006-07
10	Quarter	The quarter to which this day belongs	1st 2006-07 3rd 2006-07
11	State_assessment	days that are in the state assessment testing window	Testing Week Not Testing Week

Many more fields are possible that could be added to this table. Again, think about where the contents of the field will appear – row and column headers, filter controls, etc. With that in mind, we want the options to be descriptive and useful for lay users. Cryptic codes may be known to analysts and program staff, but administrators, board members, and the public will find plain text more useful (analysts and program staff will too when they actually have it.)

ESP Insight

Cryptic codes may be known to analysts and program staff, but administrators, board members, and the public will find plain text more useful (analysts and program staff will too when they actually have it.)

The School Dimension

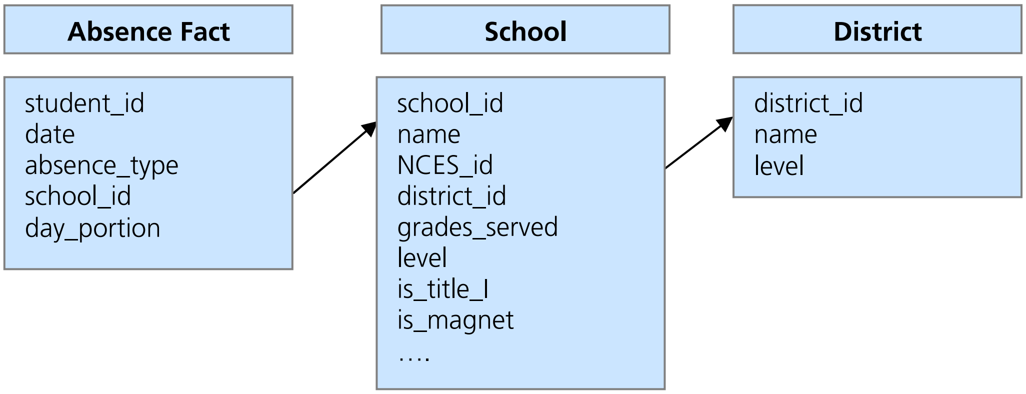
As in the above dimensions, we want to populate our school dimension with many descriptive text fields that will be useful analysis groups and filters. Those criteria should be kept in mind as fields are identified and populated.

The school_id will be our primary key. We will need some descriptive fields such as school name and possibly NCES_id. The crucial fields in the school dimension are the characteristics of the school and its community. Things like grades served, level (elementary, middle, or high school), magnet school, charter school, Title I school (school wide or targeted assistance), made AYP, and school improvement status.

You may want some demographics, but again think about reporting groups. For example, rather than a simple percent low income, also have poverty quartiles. These are the categories within which schools, and their attendance rates will be grouped.

Resist the Urge to Normalize Dimensions

State analysis will want to look at attendance rates by district as well as school. In the traditional relational database world, we would have a district table with records related to each school in the district. This is would look like:



Absence Fact

student_id
date
absence_type
school_id
day_portion

School

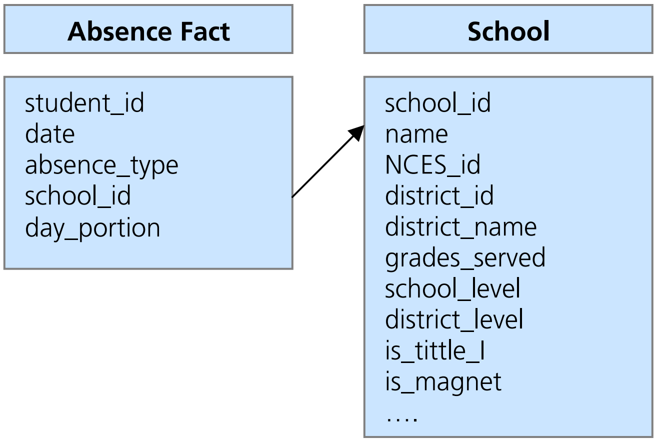
school_id
name
NCES_id
district_id
grades_served
level
is_title_I
is_magnet
….

District

district_id
name
level

This adds another layer of complexity to our structure and another join to the query needed to retrieve the data. Each extra level of joining adds time to processing of the query. For dimensional tables it is better to "flatten" the design and store the district information in the school table. Yes, it means there is duplicate data in the school table, but it also means all of our queries by district are easier to generate and run faster.

The revised structure is:



Absence Fact

student_id
date
absence_type
school_id
day_portion

School

school_id
name
NCES_id
district_id
district_name
grades_served
school_level
district_level
is_tittle_I
is_magnet
….

Normal relational designs are fine when the data are changing frequently and you want to keep data in sync and keep duplicate data entry at a minimum. In longitudinal data stores, however, we are storing the data once and not changing it again. We can afford to have some duplicate data to gain in query speed and system understanding.

The Student Dimension

The student dimension is developed and fleshed out as the other dimensions. We need the key field (student_id) and the demographic data of interest. The common demographic information is gender, race/ethnicity, and birth date of the student.

We will want any special programs the student is eligible for or participates in. These include gifted and talented programs, title I, special education, or International Baccalaureate. We want to do analysis by the NCLB common subgroups: homeless and migrant students. We will want to know if students are eligible for free or reduced price lunch.

To compare absentee rates among new comers to the school or long time enrollees, we will want the date the student entered and exited both the school and the district. If the student has exited the school, we will want the reason for exiting.

It would be useful to know if students are attending their neighborhood school or not. We will want to know the reason if not; magnet program, open enrollment, NCLB choice, persistently dangerous school choice, etc.

It is useful to know if the student needs district transportation to get to school. It is useful to compare the average daily duration among bus riders and non-bus riders. That is, is a non-bus rider more likely to attend part of a day than a bus rider.

This is not an operational system, but rather a system to support the analysis of attendance patterns to inform school calendaring and schedule development. As a consequence many of the fields that are required in the attendance system, but do not contribute to analysis are not included here. The student's phone number, address, parent/guardian contact information, etc. are all needed in the operational system but are not useful for analysis. We will not be grouping or filtering data by any of these fields. One could even argue, the student's name is not needed for this purpose.

Choose the facts for the fact table

The final step for this analysis is to select the facts for the facts table. When we were deciding the level of granularity in step one, we decided we would track the portion of a day absent for each absence by a student at a school.

For the fact table in this example, we have four fields that link to the four dimensional tables. Those four fields are absence_type, date, student_id, and school_id.

You'll note we made school a characteristic of the absence and not the student. That is, we link the school-id to the absence directly and not through the student. The reasoning is the same as when we flattened the school and district information. We want all of our joins to be directly to the fact table not nested in another.

The fact or measure of interest in this case is a simple decimal measure of the portion of the day absent. A full day absent is recorded as 1.00 and a half day absent as 0.50. These values can be added and averaged in any way that we slice the data.

We are not tracking whether a student is present or tardy. If a student has perfect attendance they will not have any entries in our fact table. The typical student may have 5-15 entries in the table in any given school year.

For a medium size district of 10,000 students, this table could have 100,000 records for each school year. Clearly a finely grained fact table, even a sparse one like this, will be much larger than any of the dimensional tables. This is why we don't worry about the size of the dimensional tables and spend effort making all the contents robust and fully descriptive.

Analysis Examples

It might be useful to look at the type of policy questions of interest and how the query would be set-up. We are not going to show the final code of the query, but rather the main selections that would be made in an analysis system's screens. *(NOTE: We will discuss analysis system characteristics in the next data warehouse whitepaper.)*

In each of these examples, we will pose the question and then:

- identify the fields that will be returned in the result,
- identify the filter conditions if any,
- identify the tables that need to be queried and how they will be joined, and finally,
- how the detailed records should be grouped for analysis.

Each of these steps will identify a set of clauses that is needed to create the SQL code for the query.

The first step will almost always include a mixture of dimensional fields and aggregate facts. Most databases and analytical tools have aggregate functions that can be used on sets of records. We will restrict our examples to SUM, AVERAGE, and COUNT.

The more powerful analytical systems have richer sets of aggregate functions. Be aware that most general analysis packages focus on general business and financial capabilities. You will need to look to educational analysis tools for sophisticated educational analysis.

1. Absentees by day of the week

For this we will select the day of the week from the time dimension table and the sum of the day_portions from the absence fact table. This will give us a frequency distribution by day of the week.

There are no filter conditions for this query.

We are selecting data just from the fact table and the time dimension table. They are the only two tables we need to include and we join records based on the date of the absence.

We will group records by the day of the week field. That means the database will select all absences whose date falls on a Monday and add the day_portions. It will repeat this for each day of the week.

The results might look like:

Day_of_Week	SUM(day_portion)
Monday	1,857.50
Tuesday	1,843.33
Wednesday	1,799.14
Thursday	1,822.80
Friday	2,150.75

2. Absentees by gender by day of the week

This is the same as in the first example, but with an additional column.

In this example, we select the day of the week from the time dimension, the student's gender from the student dimension, and again sum the portion of a day from the absence fact table.

Again, we have no filter conditions.

We now have three tables in our query. The time dimension is still joined based on the absence date. We are adding the student dimension based on the student_id.

We group the absence records by both day of the week and gender. In this case the database sums the Monday absences of males and then Monday absences of females. It again repeats for each day of the week.

Results might look like:

Day_of_Week	Gender	SUM(day_portion)
Monday	Female	737.17
Monday	Male	1,120.33
Tuesday	Female	752.83
Tuesday	Male	1,090.50
Wednesday	Female	740.74
Wednesday	Male	1,058.40
Thursday	Female	735.13
Thursday	Male	1,087.67
Friday	Female	730.50
Friday	Male	1,420.25

The addition of another column is sometimes referred to as drilling down. We are getting more detail in our analysis. Some might say "we dis-aggregated the data by gender" but I hope this example shows we did not dis-aggregate anything. In both cases, we aggregated the data, just at different levels of detail. We could not do either without the detail records to begin with.

Many more examples could be described. Analysis such as the attendance rates of Fridays prior to a 3-day holiday weekend compared to a "normal" Friday. Attendance rates in the morning of a half-day in-service day compared to the normal morning rate. Does this vary by day of the week, i.e., in-service half-days should be scheduled on which day of the week to minimize the impact on attendance? Is the attendance of students who ride a school bus impacted more or less than non-bused students on half-day in-service days?

Logical Extensions

There are several logical extensions and next steps.

The first is the move to period-by-period analysis. With this level of detail, the effect of different times of the day can be analyzed, i.e., classes after lunch vs. right before lunch. For this analysis, there will need to be the addition of a period dimension.

This extension is partially complicated by the fact that the schedules are different at the different schools leading to a "multi-grain" issue. The grain of the fact table at the elementary schools is ½ day while the grain at the middle schools is 1/6 day. The high school grain is ¼ every other day. Analysis can be done across schools with similar grain, that is, schools at the same level.

The addition of a course dimension to the period-by-period analysis at the middle and high schools can be quite revealing. With the course dimension, we can analyze of the effects of different subjects. That is, is the attendance rate for math classes different that science, language arts, or P.E.? If teachers are tied to the classes, then analysis across teachers, though controversial, can be accomplished.

The course dimension relies on a consistent course coding structure across the district — or at least a way to tie a course to a consistent set of subjects. We would also want to identify the difficultly level of the course so we can look at attendance rates in honors classes, "regular" classes, or remedial/basic classes. We will want to identify required courses for graduation, courses that are recommended for college bound students, courses that are part of a vocational sequence, and courses that are just electives.

Clearly, this level of detail can be a rich source for helping school or district administrators develop schedule and calendaring policies.

A second extension is to build this type of data store at an intermediate or state level. The largest complication is tracking the different calendars in use. In essence, the time dimension contains records for every day and every different school calendar. This also occurs in districts where each of the schools have their own calendar.

The time dimension table structure is similar with just the addition of a field to track the organization to which the calendar belongs. Maintaining the data on multiple district or school calendars is best managed by delegating calendar maintenance to the appropriate school or district or by building a mechanism to automate calendar updates and changes.

Example 2: Student Assessment Results

Our next example will be built around a fact table tracking individual student performance on state assessments. This is a very common need at state education agencies after the passing of the No Child Left Behind Act.

We will assume this data store resides in a state education agency. That will allow us to describe how to manage the additional complexity of tracking the school to district hierarchy. Barring that addition, this model works just as well at a district or intermediate education agency.

Process Being Modeled

In this example, we will look at a system that will track individual student performance on state administered assessments. This state assesses reading, math, science, and writing in grades 3 through 8 and grade 11 in the spring of each school year.

Each student in the state is assigned a permanent identifier to be used on all of their education records. Local school districts ensure that student demographic and program participation information is current in the local student information systems as of the start of the state assessment testing window. The SIF infrastructure in the state propagates any updates into the state's student demographics operational data store. (A traditional student level demographics collection just prior to test administration could serve the same purpose.)

The student demographics file at the state contains all the characteristics and program eligibility or participation data needed for No Child Left Behind. The student demographics data store is an operational system at the state. There is a data extract process that cleanses, validates, transforms, and loads the student dimension table for our longitudinal repository on a regular basis.

This state student record ID is coded on the test booklet (or used when the student logs on for the online portion of the assessment). This allows the assessment results to be associated with the student's demographic information.

Records in the student demographics file can be matched with student assessment results from the testing contractor. Any student for whom there is not an assessment result is assumed to not have been tested. Local school districts confirm this and supply background information about why the student was not tested.

The state needs to track historical student performance to meet local requirements (a growth model) and grade level performance by school for No Child Left Behind's Adequate Yearly Progress (AYP) calculations.

Questions on the assessments have been developed against state subject area and strand benchmark standards. The subject areas are high level like mathematics or writing; the strands are lower level concepts such as number concepts, geometry, data analysis, probability; or grammar, organization of writing, use of proper voice, etc.

Each subject is assessed separately. The assessment vendor supplies the state with a file with the assessment raw score, scale score, scale score for each strand for the subject, and percentile rank by student and school. Cut scores for the state's 4 performance levels in each subject are also supplied.

(Scale scores are equivalent across test forms and school years within a subject. Each subject has its own scale, so a 350 in mathematics may not mean the same thing as a 350 in writing.)

The supplied assessment file also notes any special testing conditions that may affect the results such as any accommodations that were used or whether the assessment was incomplete.

Level of Detail for the Facts

For the growth model requirement we know we will need at least student subject level performance by school year and school. Scale scores are the only measure that can be validly compared over test administrations and test forms so we will use that.

How we want to deal with the multiple subjects is an interesting question. Typically, we might define the fact table to have fields for subject and score for each student. There would exist 4 records from each student each year; one each for mathematics, science, reading and writing. Alternatively, we could have a fact table with student level records that has four sets of facts, all in the same record. The two table structures are partially shown below.

student_id
subject_area
scale_score
….

student_id
math_scale_score
reading_scale_score
science_scale_score
writing_scale_score

To use the left hand structure, the user would always have to filter on the particular subject being studied since the scale scores are not equivalent across subjects. This structure is a more flexible design if additional subjects are likely to be added in the future. To handle additional subjects we only have to allow additional values in the subject_area field.

The right hand structure makes it more difficult for a user to accidentally compare score inappropriately. This structure is less flexible if additional subject assessments are likely. Normally, we want to design systems that minimize possible changes to table structures. Since analysis systems are built on top of the data store table structures, changes in these structures can ripple throughout.

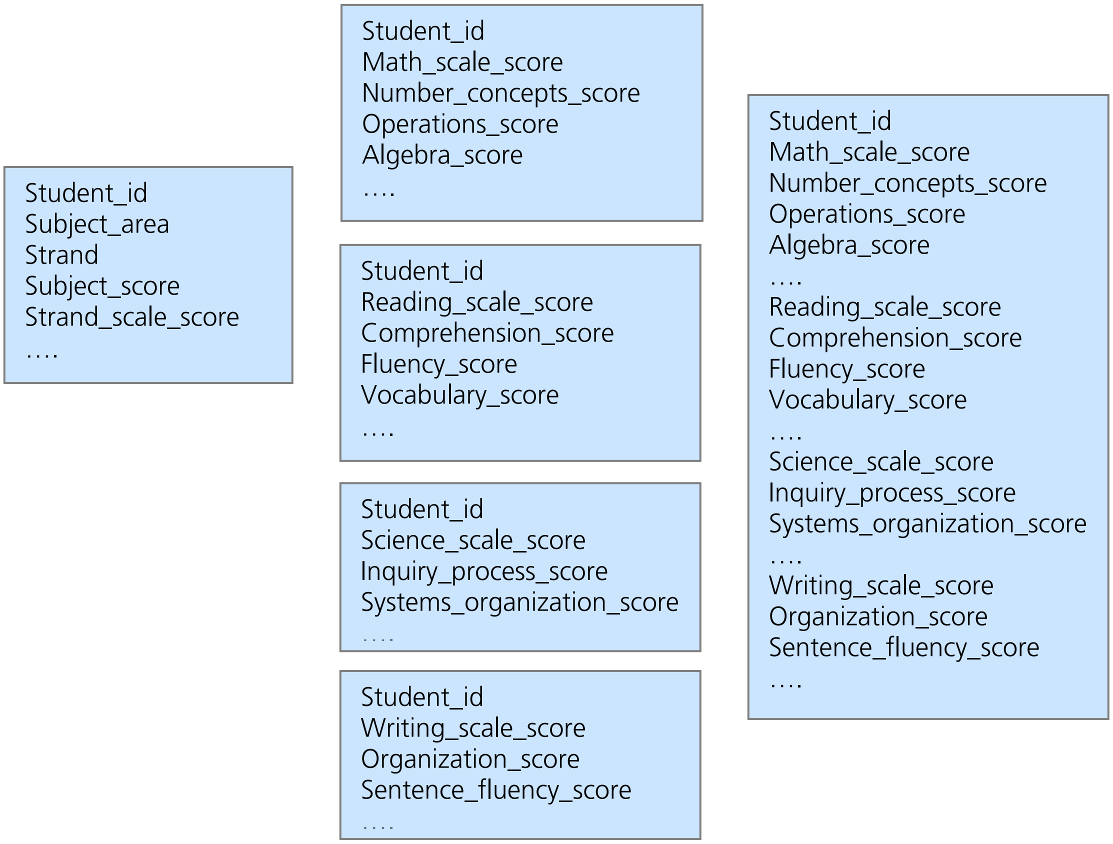
A middle approach might have a fact table for each subject, i.e., a table for math scores and a separate table for science scores. This has the advantage of discouraging users from making inappropriate comparisons across subjects. The impact of adding subjects is more, however, than the modification of the right hand structure as whole new systems will need to be built for each subject added.

ESP Insight
Flexibility requirements may have more of an influence on system design decisions at the district level.

A decision depends partially on our analysis of the likelihood additional subjects will be added. At the state level, we can be fairly confident the set of subjects for the foreseeable future is set. A district, on the other hand, may need to assess social studies, foreign language, the arts, health, and PE, or vocational education. Consequently, flexibility requirements may have more of an influence on system design decisions at the district level.

How we need to handle strand-level results also affects our design decisions. Remember, the strands are the main subject constituent concepts. For mathematics, they are number concepts, arithmetic operations, algebraic concepts, geometry and space relations, and statistics and probability. For writing, the strands cover the six traits of ideas, organization, voice, sentence fluency, word choice, and conventions. The state assessment gives a score for each of the strands.

Again, the scale score for student performance on the strands is comparable across student and time, but not between strands. The design considerations are similar to our subject level decisions. We could set the grain such that we have a table with a record for each strand score (column 1 below); we could have the subject specific tables with fields for the overall subject score and all the strand scores (middle column); or we could extend the right-hand table above to have all the subject and strand scores for a particular student and test cycle.



Student_id
Subject_area
Strand
Subject_score
Strand_scale_score
....

Student_id
Math_scale_score
Number_concepts_score
Operations_score
Algebra_score
....

Student_id
Reading_scale_score
Comprehension_score
Fluency_score
Vocabulary_score
....

Student_id
Science_scale_score
Inquiry_process_score
Systems_organization_score
....

Student_id
Writing_scale_score
Organization_score
Sentence_fluency_score
....

Student_id
Math_scale_score
Number_concepts_score
Operations_score
Algebra_score
....
Reading_scale_score
Comprehension_score
Fluency_score
Vocabulary_score
....
Science_scale_score
Inquiry_process_score
Systems_organization_score
....
Writing_scale_score
Organization_score
Sentence_fluency_score
....

For the purposes of this paper, we will select the right hand solution. We will set the grain as the results from the set of state assessments given to a particular student in the spring of a particular school year. Unlike our attendance example where the fact table only contained one fact in each record, this fact table has several, one each for each subject and one each for each strand.

This structure aligns well with the contents of the file from the assessment contractor. It also is easier for a school or district administrator or teacher to understand.

Again, a district with more subjects, a less stable set of subjects, or a more frequent cycling of testing will probably want to make a different grain decision.

Choose the Dimensions

Now that the grain has been decided, we need to identify and define the dimensions. As with all longitudinal data stores, we will have a time dimension. We will have student and school dimensions. We will also have a dimension for the assessment administration.

The Time Dimension

In the attendance example, the facts were daily events and therefore the time dimension had a record for each day. In this example, the assessment represents student learning for a particular school year. Therefore, there will be a record in the time dimension for each school year assessed.

If this were a district level repository, and the assessment were given twice a year, quarterly, or some other cycle, then the time dimension table would need to reflect that time frame.

The Assessment Dimension

The assessment dimension fleshes out the information about a particular administration of the assessment. The characteristics for an assessment vary by school year, grade level, and subject.

The decision to flatten the fact table to include all four subjects means we need to similarly flatten the assessment dimensional table. That is, we will include the math performance level cut scores, the science cut scores, the reading cut scores, and the write cut scores in a single record.

Assessment Dimension
School_year
Grade_level
Math_basic_cut_score
Math_proficient_cut_score
Math_advanced_cut_score
Number_concepts_cut_score
….
Reading_basic_cut_score
Reading_proficient_cut_score
Reading_advanced_cut_score
Fluency_cut_score
….
Science_basic_cut_score
Science_proficient_cut_score
Science_advanced_cut_score
Inquiry_process_cut_score
….
Writing_basic_cut_score
Writing_proficient_cut_score
Writing_advanced_cut_score
Organization_cut_score
….

We need the cut scores for the four proficiency levels: below basic, basic, proficient, and advanced. The proficiency level cut scores set the minimum for the proficiency level. Below basic is any score that doesn't reach the basic_cut_score.

The strands don't have individual proficiency levels, just a pass or fail so there is a single cut score for each.

The School Dimension

The school dimension discussion from the attendance example applies here as well. Each student's test results will be associated with the school that the student attends. We will associate each assessment result with a school directly, rather than to a student and then the student to a school. The decision to flatten the school-district relation into a single table still applies.

Tracking Historical School Characteristics

We will be doing analysis of results by characteristics of the school. For example, comparing Title I schools to non-title schools, charter schools to "regular" schools, alternative schools to traditional high schools, etc. Many of these characteristics change over time. We need to know, not what a school looks like today, but what were its characteristics at the time the assessment was given.

ESP Insight
We need to know, not what a school looks like today, but what were its characteristics at the time the assessment was given.

We have three options for tracking these historical changes. (In data warehouse terminology, this is referred to as tracking "slowly changing dimensions").

1. We can treat each change as if we had a brand new school. That is, create a new school id and add a new school record whenever one of these characteristics changes. Assessment results are tied to the correct set of school characteristics in place at the time of the assessment. But, we cannot track a school across time whenever one of these changes. In some cases, a change that would generate a new school record doesn't affect our analysis and sometimes it does.

2. We could add an effective_date and expiration_date to the school records. Whenever, the characteristics changes, we expire the old record and create a new record with the same school_id but a new effective_date. To match an assessment set of facts to the right school requires matching the school_id and then finding the record where the effective_date is before the assessment window, and the expiration_date is either after the window or empty (i.e., has not been replaced).

   To simplify matching, each new set of characteristics for a particular school can get a "version number." We can then set up the fact table so each set of assessment results is tied to a school and the correct version of its characteristics. With this structure we can still follow a school over time yet see what it looked like at any point in time. By matching each set of assessment facts to a school_id and version, we speed up any queries – we are not doing all the date comparison calculations described above.

3. The third option is to create a new set of records for each school each year. From the fact table, we would match on both the school_id and school_year. Whether we use the second, versioning option, or the third, add a new school record every year, approach depends on how volatile the characteristics are for the schools in your state.

   For example, you may want a percent_poverty field in the school dimension table. If so, this value is likely to change every year, so the versioning approach is likely to generate nearly a new record every year anyway.

   For this example, we will use option two: create a new school dimension record each time the characteristics change. The assessment facts will match to the combination of school_id and school_version.

The Student Dimension

The student dimension for assessment results can be the same as from the attendance example. We have the slowly changing dimension issue here as well (eligible for free/reduced lunch, participates in Title I or Special Education, for example). Since most students change grade levels every year, we will use the third approach for tracking the correct student characteristics. The assessment facts will match on the combination of student_id and school_year.

Choose the Facts for the Fact Table

In this example we will use the scale score in each of the subjects and the strands as our facts. Raw scores are supplied by the testing contractor. These are not comparable across different forms of the test nor across school years. Percentile ranks can not be averaged.

A student's performance (level, below basic, basic, proficient, or advanced) needs to be counted by school and district to calculate AYP. We could calculate these as needed, but it is probably more useful to calculate them once and store the calculated value in the assessment score fact table. Likewise, it is useful to store the pass/fail flags for each of the strands as well.

We will also want any information about the conditions of a particular assessment. These conditions include whether accommodations were used, and if so, which ones. We will need to know, for students that were not tested, the reason they were not tested. If a test was only partially complete, or there is some other reason the results are invalid for AYP purposes should be noted as well.

The fact table will have the fields necessary to connect to the dimensional tables: school_year, school_id, student_id, student_version, and grade_level.

Supporting "Drill Across"

The changes we made to both the student and school dimensions for this example apply to the attendance example as well. We don't want to have two sets of student and school dimensions. Rather we will build these tables once and can use them in both sets of analysis.

If we wish to use the revised dimensions for tracking attendance, then we need to modify the attendance fact table to add the new dimensional key fields. We need to add the school version so we can match to the school dimension on school_id and version. We need to add school_year so we can match to the correct student characteristics record.

When facts share common granularity and dimensions, then they can be compared at that level of detail. This is called "drill across." Tables with different granularity can support drill across if they are first summarized to a common granularity. If we summarize student's attendance by school year (easier now that we have the new school year field in the attendance fact record), then we can look at the relation of attendance rates to student performance.

ESP Insight

When facts share common granularity and dimensions, then they can be compared at that level of detail. This is called "drill across." Tables with different granularity can support drill across if they are first summarized to a common granularity.

In this example, a student's school year attendance rate can be joined to their assessment results through the common dimensions of school_year, school, school_version, and student_id. At this point it is as if the attendance rate were an additional fact in the assessment results fact table. We can do a rich set of analysis across these two fact sets – school characteristics, across school years, differing student characteristics, etc.

Just a reminder note: a relationship between performance and attendance does not imply one causes the other. Students may perform poorly because they don't attend and so have not been taught; or they may not attend because they are not performing well and feel unsuccessful.

Topics Not Discussed

Several more advanced data warehouse and reporting topics have not been covered in this paper.

Different database management systems have different capabilities for tuning longitudinal data. We did not discuss these features or what capabilities consumers should seek when purchasing software.

We also did not discuss the differing indexing strategies that can be used to speed these queries.

Some of the databases will identify common queries and help pre-generate summaries to speed up future requests. Some of these systems will access summary data when appropriate and transparently shift to accessing the detail records as users drill deeper into the data.

Most database management systems today will handle the longitudinal data needs of most districts and states. Only when educational systems get quite large, do these features significantly impact performance. The typical education query may take 20-30 seconds to execute. Spending large dollars to tune a query to get 5-10 second results may not be necessary. As the data volume grows so tuning queries to 30 seconds from several minutes may be more justifiable.

Many data warehouse and reporting systems generate reports that drive school funding or program accountability systems. Due to the high stakes associated with the products of these systems, the data stores need to include change logging and other audit trail information. In the introduction, we identified an archive data store in the longitudinal data stack that fulfills that function. We have not discussed that in detail in this paper.

Coming Attractions

The next whitepaper in this series will explore the analytic and reporting tools. The real power of longitudinal data stores is not realized until users can get results. Easy to use tools are critical to the success of any longitudinal data system project.

We will also cover FERPA reporting concerns. These topics include cell size restrictions, security roles and access concerns, and protecting individually identifiable data.

About ESP Solutions Group

ESP Solutions Group provides its clients with *Extraordinary Insight*™ into K-12 education data systems and psychometrics. Our team is comprised of industry experts who pioneered the concept of "data driven decision making" and now help optimize the management of our clients' state and local education agencies.

ESP personnel have advised school districts, all 52 state education agencies, and the U.S. Department of Education on the practice of K-12 school data management. We are regarded as leading experts in understanding the data and technology implications of the **No Child Left Behind Act (NCLB)**, **Education Data Exchange Network (EDEN)**, and the Schools **Interoperability Framework (SIF).**

Dozens of education agencies have hired ESP to design and build their student record collection systems, federal reporting systems, student identifier systems, data dictionaries, evaluation/assessment programs and data management/analysis systems.

To learn how ESP can give your agency *Extraordinary Insight*™ into your K-12 education data, contact Greg Nadeau at (781) 370-1017 or gnadeau@espsg.com.

This document is part of *The Optimal Reference Guide* Series, designed to help education data decision makers analyze, manage, and share data in the 21st Century.

ESP Solutions Group

(512) 458-8364
www.espsolutionsgroup.com
Austin • Boston • Washington DC