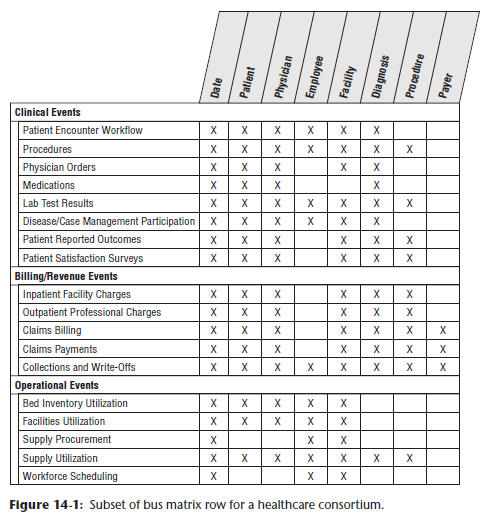
# Kimball Data Warehouse Toolkit

## Ch 14 – Healthcare

* Healthcare industry is undergoing tremendous change as it seeks to **both improve patient outcomes, while simultaneously improving operational efficiencies**
* The **challenges are plentiful** as organizations attempt to integrate their clinical + admin information
* **Healthcare data presents several interesting dimensional design patterns**
* **Concepts:**
* **Bus matrix** snippet for a healthcare organization
* **Accumulating snapshot fact table** to handle the **claims billing and payment pipeline**
* **Dimension role playing** for multiple dates and physicians
* **Multivalued dimensions**, such as patient diagnoses
* **Supertype** and **subtype** handling of healthcare charges
* Treatment of **textual comments**
* Measurement type dimension for **sparse, heterogeneous measurements**
* **Handling of images** with dimensional schemas
* Facility/equipment inventory utilization as **transactions** and **periodic snapshots**

### Healthcare Case Study and Bus Matrix

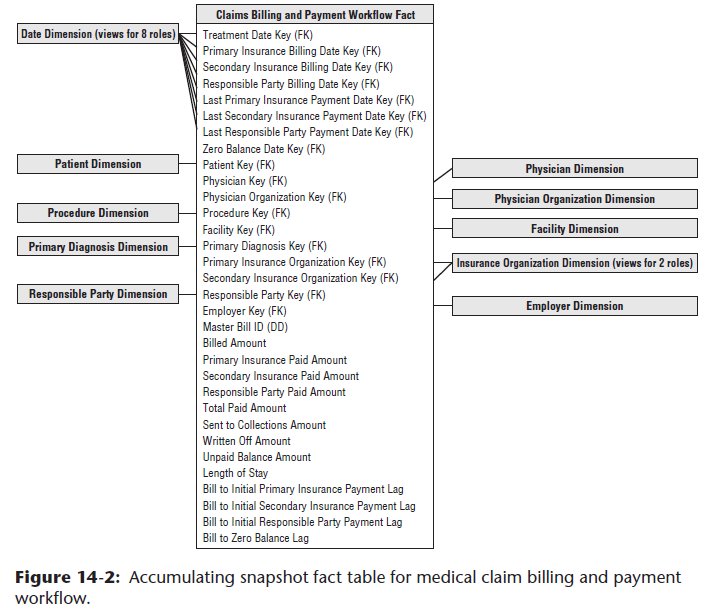
* In the face of unprecedented consumer focus + governmental policy regulations, coupled w/ internal pressures, **healthcare organizations need to leverage information more effectively to impact both patient outcomes + operational efficiencies**
* Healthcare organizations **typically wrestle w/ many disparate systems to collect their clinical, financial, + operational performance metrics**
* This **information needs to be better integrated to deliver more effective patient care, while concurrently managing costs + risks**
* Healthcare **analysts want to better understand which procedures deliver the best outcomes, while identifying opportunities to impact resource utilization, including labor, facilities, + associated equipment + supplies**
* Large healthcare consortiums with networks of physicians, clinics, hospitals, pharmacies, + labs are focused on these requirements, especially as both the federal government + private payers are encouraging providers to assume more responsibility for the quality + cost of their healthcare services
* The figure below illustrates a sample snippet of a healthcare organization’s bus matrix



* Traditionally, **healthcare insurance payers have leveraged claims information to better understand their risk, improve underwriting policies, + detect potential fraudulent activity**
* Payers have historically been more sophisticated than healthcare provider organizations in leveraging data analytically, perhaps in part because their **prime data source (claims) was more reliably captured + structured than providers’ data**
* However, **claims data is both a benefit + a curse for payers’ analytic efforts, because it historically *hasn’t* provided the robust, granular clinical picture**
* **Increasingly, healthcare payers are partnering w/ providers to leverage detailed patient information to support more predictive analysis**
* **In many ways, needs + objectives of the providers + payers are converging, especially with the push for shared-risk delivery models**
* **Every patient’s episode of care w/ a healthcare organization generates mounds of information**
* **Patient-centric transactional data falls into 2 prime categories**: **administrative** and **clinical**
* The **claims billing data** provides **detail on a patient bill from a physician’s office/clinic/hospital/lab**
* The **clinical medical record**, on the other hand, is **more comprehensive + includes not only the services resulting in charges, but also lab test results, prescriptions, physician’s notes or orders, + sometimes outcomes**
* **The issues of conforming common dimensions remain exactly the same for healthcare as in other industries**
* **Obviously, the most important conformed dimension is the patient**
* In Chapter 8: CRM, we described the need for a 360-degree view of customers
* It’s easy to argue that a 360-degree view of patients is even *more* critical given the stakes
* Adoption of patient electronic medical record (EMR) + electronic health record (EHR) systems clearly focus on this objective.
* **Other dimensions that must be conformed include:**
* **Date**
* **Responsible party**
* **Employer**
* **Health plan**
* **Payer (primary and secondary)**
* **Physician**
* **Procedure**
* **Equipment**
* **Lab test**
* **Medication**
* **Diagnosis**
* **Facility (office, clinic, outpatient facility, + hospital)**
* **In the healthcare arena, some of these dimensions are hard to conform, whereas others are easier than they look at first glance**
* The **patient dimension has historically been challenging**, at least in the US, due to lack of a reliable national identity number and/or consistent patient identifier across facilities + physicians
* To further complicate matters, HIPAA includes strict privacy + security requirements to protect the confidential nature of patient information
* **Operational process improvements, like EMR’s, are ensuring more consistent master patient identification**
* The **diagnosis and treatment dimensions are considerably more structured + predictable** than you might expect **because the insurance industry + government have mandated their content**
* Ex: Diagnosis + disease classifications follow the International Classification of Diseases (ICD) standard for consistent reporting.
* Similarly, the Healthcare Common Procedure Coding System (HCPCS) is based on the American Medical Association’s Current Procedural Terminology (CPT) to describe medical, surgical, + diagnostic services, along with supplies + devices
* Dentists use the Current Dental Terminology (CDT) code set, which is updated and distributed by the American Dental Association (ADA)
* Finally, **beyond integrated patient-centric clinical + financial information, healthcare organizations also want to analyze *operational* information regarding utilization of their workforce, facilities, + supplies**
* Much of the discussion from earlier chapters about HR, inventory management, + procurement processes is also applicable to healthcare organizations

### Claims Billing and Payment

* Imagine you work in the healthcare consortium’s billing organization 🡪 **receive the primary charges from the physicians + facilities, prepare bills for the responsible payers, + track the progress of the claims payments received**
* **A dimensional model for the claims billing process must address a *number* of business objectives**
* **Want to:**
* **Analyze the billed dollar amounts by *every available* *dimension*, including patient, physician, facility, diagnosis, procedure, + date**
* **See *how* these claims have been paid + what % of the claims have not been collected**
* **See how long it takes to get paid, + the current status of all unpaid claims**
* As discussed in Chapter 4: Inventory, **whenever a source business process is considered for inclusion in the DW/BI system, there are 3 essential grain choices:**
* **Remember: the fact table’s granularity determines what constitutes a fact table row**
* i.e., ***what is the measurement event being recorded?***
* **1) The transaction grain is the most fundamental**
* In the **healthcare billing** example, the **transaction grain would include *every* billing transaction from the physicians + facilities, as well as *every* claim payment transaction** **received**
* **2) The** **periodic snapshot** **is the grain of choice for long-running time series, such as bank accounts + insurance policies**
* However, it **DOESN’T do a good job of capturing the behavior of relatively *short*-lived processes, such as orders or medical claims billing**
* **3) The accumulating snapshot grain** **is chosen to analyze the claims billing + payment workflow**
* A **single fact table row represents a single line on a medical claim**
* Furthermore, the **row represents the accumulated history of the line item from the moment of creation to the current state**
* **When anything about the line changes, the row is *revisited* + *modified* appropriately**
* From the POV of the billing organization, **assume the standard scenario of a claim includes**:
* Treatment date
* Primary + secondary insurance billing dates
* Responsible party billing date
* Last primary + secondary insurance payment dates
* Last responsible party payment date
* Zero balance date
* **These dates describe the typical claim workflow**
* An **accumulating snapshot does NOT attempt to *fully* describe *unusual* situations**
* Business **users** undoubtedly **need to see all details of messy claim payment scenarios** because multiple payments are sometimes received for a single line, or conversely, a single payment sometimes applies to multiple claims
* **Companion transaction schemas inevitably will be needed**
* *In the meantime*, **the purpose of the accumulating snapshot grain is to place every claim into a standard framework so that the analytic objectives described earlier can be satisfied easily**
* With a **clear understanding** that **an individual fact table row represents the accumulated history of a line item on a claim bill**, you can **identify the dimensions by carefully listing everything known to be true in the context of this row**
* In this hypothetical scenario, you know the patient, responsible party, physician, physician organization, procedure, facility, diagnosis, primary insurance organization, secondary insurance organization, + master patient bill ID number, as shown below



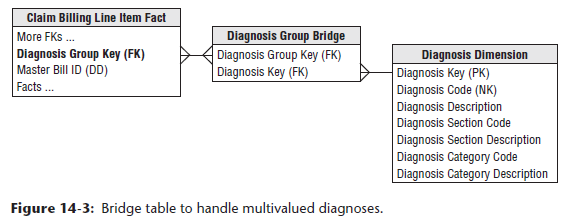
* **Interesting facts accumulated over the claim line’s history include: billed amount, primary + secondary insurance paid amounts, responsible party paid amount, total paid amount (*calculated*), amount sent to collections,** **amount written off, amount remaining to be paid (*calculated*), length of stay, number of days from billing to initial primary insurance, secondary insurance, + responsible party payments**, and, finally, **number of days to zero balance**
* A **row is initially created in this fact table when the charge transactions are received from the physicians or facilities and the initial bills are generated**
* On a given bill, perhaps the primary insurance company is billed, but the secondary insurance + responsible party are *not* billed, pending a response from the primary insurance company
* **For a period of time after the row is first entered into the fact table, the last 7 dates are *not* applicable**
* **Because the surrogate date keys in the fact table must *NOT* be NULL, they will point to a date dimension row reserved for a To Be Determined (TBD) date**
* **In the weeks *after* creation of the row, some payments are received**
* Bills are then sent to the secondary insurance company + responsible party
* **Each time these events take place, the same fact table row is revisited, + the appropriate keys + facts are destructively updated**
* This **destructive updating poses some challenges for the DBA**
* **If most of the accumulating rows stabilize + stop changing within a given timeframe, a physical reorganization of the database at that time can recover disk storage + improve performance**
* **If the fact table is partitioned on the treatment date key, the physical clustering or partitioning probably will be well preserved throughout these changes because the treatment date is *not* revisited and changed**

#### Date Dimension Role Playing

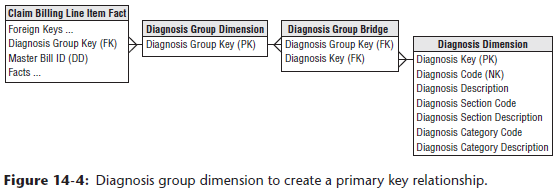
* **Accumulating snapshot fact tables *always* involve multiple date stamps**, like the 8 FKs pointing to the date dimension in our current schema
* These 8 **date FKs should *not* JOIN to a *single* instance of the date dimension table**
* **Instead, create Views on the *single* underlying date dimension table, + JOIN the fact table separately** to these 8 views, as if they were 8 independent date dimension tables
* The 8 **View definitions should cosmetically relabel column names to be distinguishable, so BI tools accessing the Views will present understandable column names to business users**
* Although the **role-playing behavior of the date dimension is a common characteristic of accumulating snapshot fact tables, other dimensions in our schema play roles in similar ways,** such as the payer dimension
* We will see the physician dimension play multiple roles, depending on whether the physician is the referring physician, attending physician, or working in a consulting/assisting capacity

#### Multivalued Diagnoses

* **Typically, dimensions surrounding a fact table take on a single value in the context of a fact event**
* **However, there ARE situations where multi-valued-ness is natural and unavoidable**
* The **diagnosis dimension** in healthcare fact tables is a good example
* At the moment of a procedure/lab test, a patient has 1 or more diagnoses
* EMR applications facilitate the physician’s selection of multiple diagnoses well beyond the historical practice of providing the minimal coding needed for reimbursement, + the result is a richer, more complete picture of the severity of the patient’s medical condition
* **There is strong analytic incentive to retain the multivalued diagnoses, along w/ other financial performance data, especially as organizations do more comparative utilization + cost benchmarking**
* If there were always a maximum of 3 diagnoses, for instance, you might be tempted to create 3 diagnosis FKs in the fact table with corresponding dimensions, almost as if they were **roles**.
* **However, diagnoses don’t behave like independent roles**
* And unfortunately, there are often more than 3 diagnoses, especially for hospitalized elderly patients who may present 20 simultaneous diagnoses!
* **Diagnoses don’t fit into well-defined roles other than potentially the primary admitting and discharging diagnoses**
* **Finally, a design w/ multiple diagnosis FKs would make for very inefficient BI applications, because the query doesn’t know which dimensional slot to constrain for a particular diagnosis**
* The **design shown below handles the open-ended nature of multiple diagnoses**



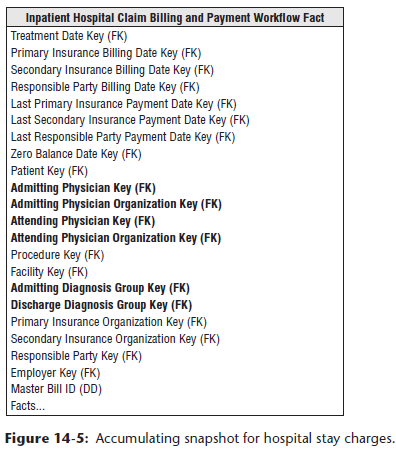
* The **diagnosis FK in the fact table is replaced with a diagnosis *group* key**, which is **connected by a many-to-many JOIN to a diagnosis group bridge table, which contains a separate row for each individual diagnosis in a particular group**
* If a patient has 3 diagnoses, they are assigned a diagnosis group with 3 corresponding rows in the bridge table
* In Chapter 10: Financial Services, we described the use of a **weighting factor on each bridge table row to allocate the fact table’s metrics accordingly**
* However, **in the case of multiple patient diagnoses, it’s virtually impossible to weight their impact on a patient’s treatment or bill**, beyond the potential determination of a primary diagnosis
* **Without a realistic way of assigning weighting factors, the analysis of diagnosis codes must largely focus on *impact questions*** like “What is the total billed amount for procedures involving the diagnosis of congestive heart failure?”
* Most healthcare analysts understand **impact analysis may result in over counting as the same metrics are associated with multiple diagnoses**
* **NOTE:** **Weighting factors in multivalued bridge tables provide an elegant way to prorate numeric facts to produce correctly weighted reports**
* However, weighting factors are **by no means *required* in a dimensional design**
* **If there is no agreement or enthusiasm within the business community for the weighting factors, they should be left out**
* Also, **in a schema with more than 1 multivalued dimension, it is not worth trying to decide how multiple weighting factors would interact**
* **If the many-to-many JOIN in the bridge schema above causes problems for a modeling tool that insists on proper FK-to-PK relationships, the equivalent design below can be used**



* **An *extra* table whose PK is a diagnosis *group* is inserted between the fact + bridge tables**
* There is likely no new information in this extra table, unless there were labels for a cluster of diagnoses, such as “Kimball Syndrome”, but **now both the fact table + bridge table have conventional many-to-one joins in all directions**
* **If a unique diagnosis group is created for every patient encounter, the number of rows could become astronomical and many of the groups would be identical**
* **Better approach = have a portfolio of diagnosis groups that are repeatedly used**
* **Each set of diagnoses would be looked up in the master diagnosis group table during the ETL 🡪 If the existing group is found, it is used, + if not found, a new diagnosis group is created**
* Chapter 19: ETL Subsystems and Techniques provides guidance for creating and administering bridge tables
* In an **inpatient** hospital stay scenario, the **diagnosis group may be unique to each patient if it evolves over time during the patient’s stay**
* In this case, you’d **supplement the bridge table with 2 date stamps to capture begin + end dates**
* **Although the twin date stamps complicate updates to the diagnosis group bridge table, they are useful for change tracking**, as described more fully in Chapter 7: Accounting

#### Supertypes and Subtypes for Charges

* We’ve **described a design for billed healthcare treatments to cover both inpatient (IP) + outpatient (OP) claims**
* **In reality, healthcare charges resemble the supertype and subtype pattern** described in Chapter 10
* **Facility charges for IP hospital stays differ from professional charges for OP treatments in clinics + doctor offices**
* If focused exclusively on hospital stays, it would be reasonable to tweak the current dimensional structure to incorporate more hospital-specific information
* Below shows a **revised set of dimensions specialized for hospital stays,** w/ new dimensions bolded



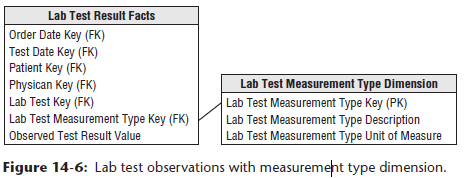
* You can see 2 roles for the physician: **admitting** + **attending** physician
* The figure shows physician organizations for both roles because physicians may represent different organizations in a hospital setting
* With more complex surgical events, such as a heart transplant operation, **whole teams** of specialists + assistants are assembled
* In this case, you **could include a key in the fact table for the primary responsible physician, +** **the other physicians + medical staff would be linked to the fact row via a group key to a multivalued bridge table**
* You also have **2 multi-valued diagnosis dimensions on each fact table row**
* The **admitting diagnosis group is determined at the beginning of the hospital stay + should be the same for every treatment row that is part of the same hospital stay**
* The **discharge diagnosis group is not known until a patient is discharged**

### Electronic Medical Records (EMR)

* Many healthcare organizations are moving from paper-based processes to EMRs
* In the US, federally mandated quality goals to support improved population health management may be achievable only with their adoption
* Healthcare providers are aggressively implementing **EHR systems**, + the movement is **significantly impacting healthcare DW/BI initiatives**
* **EMRs** can present **challenges for DW environments** because of their **extreme variability + potentially extreme volumes**
* Patients’ medical record data comes in *many* different forms, ranging from numeric data, to freeform text comments entered by a healthcare professional, to images + photographs
* One thing is certain 🡪 The **amount + variability of electronic data in the healthcare industry will continue to grow**

#### Measure Type Dimension for Sparse Facts

* As designers, it is **tempting to strive for a more standardized framework that could be extended to handle data variability**
* Ex: Could potentially handle the variability of lab test results with a “measurement type” dimension, describing what the fact row means, or, in other words, what the generic fact represents
* The unit of measure for a given numeric entry is found in the associated “measurement type” dimension row, along with any additivity restrictions, as shown below



* This approach is **superbly flexible** 🡪 can add new measurement types simply by adding new rows in the measurement type dimension, *not by altering the structure of the fact table*
* This approach **also eliminates the NULLs in the classic positional fact table design because a row exists only if the measurement exists**
* ***However*, there are trade-off’s**
* **Using a “measurement type” dimension may generate lots of new fact table rows, because the grain is “one row per measurement per event” rather than the more typical “one row per event.”**
* If a lab test results in 10 numeric measurements, there are now 10 rows in the fact table rather than a single row in the classic design
* **For *extremely sparse* situations, such as clinical lab or manufacturing test environments, this is a reasonable compromise**
* ***However*, as the density of the facts grows, you end up spewing out too many fact rows** + at this point you **no longer have sparse facts and should return to the classic fact table design with fixed columns**
* Moreover, **this measurement type approach may complicate BI data access applications**
* In the **relational star schema, combining 2 numbers that were captured as part of a single event is more difficult with this approach because now you must fetch 2 rows from the fact table**
* **SQL likes to perform arithmetic functions *within* a row, not *across* rows**
* In addition, **must be careful not to mix incompatible amounts in a calculation because all the numeric measures reside in a single amount column**
* **Worth noting that multidimensional OLAP cubes are more tolerant of performing calculations across measurement types**

#### Freeform Text Comments

* Freeform text comments, such as clinical notes, are sometimes associated with fact table events
* **Although text comments are not very analytically potent unless parsed into well-behaved dimension attributes, business users are often unwilling to part with them given the embedded nuggets of information**
* **Textual comments should NOT be stored in a fact table *directly*, because they waste space + rarely participate in queries**
* Some designers think it’s permissible to store textual fields in the fact table, as long as they’re referred to as degenerate dimensions
* **Degenerate dimensions are most typically used for operational transaction control numbers + identifiers, and it’s *not* an acceptable approach/pattern for contending with bulky text fields**
* **Storing freeform comments in the fact table adds clutter that may negatively impact performance of analysts’ more typical quantitative queries**
* **Unbounded text comments should either be:**
* **1) Stored in a separate “comments” dimension or**
* **2) Treated as attributes in a transaction event dimension**
* A **key consideration when evaluating these 2 approaches = the text field’s cardinality**
* **If there’s nearly a unique comment for every fact table event, storing the textual field in a transaction dimension makes the most sense**
* **However, in many cases, “No Comment” is associated with numerous fact rows**
* Because the **number of unique text comments in this scenario is much smaller than the number of unique transactions, it’d make more sense to store the textual data in a comments dimension with an associated FK in the fact table**
* **In either case, queries involving both the text comments + fact metrics will perform relatively poorly, given the need to resolve JOINs between two voluminous tables**
* **Often business users want to drill into text comments for further investigation after highly selective fact table query filters have been applied**

#### Images

* Sometimes the data captured in a patient’s EMR is an **image**, in addition to either quantitative numbers or qualitative notes
* There are **trade-off s between capturing a JPEG filename in the fact table to refer to an associated image vs. embedding the image as a blob directly in the database**
* The **advantage of using a JPEG filename = other image creation, viewing, + editing programs can freely access the image**
* The **disadvantage = a separate database of graphic files must be maintained in synchrony with the fact table**

### Facility/Equipment Inventory Utilization

* **In addition to financial + clinical data, healthcare organizations are also keenly interested in more operationally oriented metrics**, such as utilization + availability of their assets, whether referring to patient beds or surgical operating theatres
* In Chapter 4, we discussed **product inventory data as transaction events, as well as periodic snapshots, + facility/equipment inventories in a healthcare organization can be handled similarly**
* Ex: Envision a **bed utilization periodic snapshot** with every bed’s status at regularly recurring points in time, perhaps at midnight, the start of every shift, or even more frequently throughout the day
* In addition to a snapshot date and potentially time-of-day, this fact-less fact table would include FKs to identify the patient, attending physician, + perhaps an assigned nurse on duty
* Conversely, imagine treating the **bed inventory data as a transaction fact table** with **1 row per movement into and out of a hospital bed**
* This may be a simplistic transaction fact table w/ transaction date + time dimension FKs, along with dimensions to describe the type of movement, such as filled or vacated
* In the case of OR utilization + availability, can envision a lengthier list of statuses, such as pre-op, post-op, or downtime, along with time durations.
* **If inventory changes are not terribly volatile**, such as the beds in a rehab or eldercare IP environment, **consider a timespan fact table**, as discussed in Chapter 8, **with row effective and expiration dates and times to represent the various states** of a bed **over a period of time**

### Dealing With Retroactive Changes

* As DW/BI practitioners, we have well-developed techniques for accurately capturing the historical flow of data from our enterprise’s source applications: **Numeric measurements go into fact tables, which are surrounded with contemporary descriptions of what you know is true at the time of the measurements, packaged as dimension tables**
* The **descriptions of patient, physician, facility, + payer evolve as SCD’s whenever these entities change their descriptions**
* **However, in the healthcare industry, especially w/ legacy operational systems, you often need to contend with late-arriving data that should have been loaded into the DW weeks or months ago**
* Ex: Might receive data regarding patient procedures that occurred several weeks ago, or updates to patient profiles that were back-dated as effective several months ago
* **The more delayed the incoming records are, the more challenging the DW/BI system’s ETL processing becomes**
* We discuss these late arriving fact and dimension scenarios in Chapter 19
* **Unfortunately, these patterns are common in healthcare DW/BI environments, + in fact, they may be the dominant modes of processing rather than specialized techniques for outlier cases**
* **Eventually, more effective source data capture systems should reduce the frequency of these late arriving data anomalies**