# Kimball Data Warehouse Toolkit

## Ch 15 – E-Commerce

* A web-intensive business’s **clickstream****data records the gestures of every web visitor**
* In its most elemental form, the **clickstream is every page event recorded by each of the company’s web servers**
* The **clickstream contains a number of new dimensions, such as page, session, and referrer, which are not found in other data sources**
* The **clickstream is a torrent of data that can be difficult and exasperating for DW/BI professionals**
* Does it connect to the rest of the DW/BI system? Can its dimensions + facts be conformed in the EDW bus architecture?
* We start this chapter by describing the raw clickstream data source + designing its relevant dimensional models, then discuss the impact of Google Analytics, which can be thought of as an external DW delivering information about your website
* We then integrate clickstream data into a larger matrix of more conventional processes for a web retailer, + argue that the profitability of the web sales channel can be measured if you allocate the right costs back to the individual sales
* **Concepts:**
* **Clickstream** data and its **unique dimensionality**
* Role of **external services** such as Google Analytics
* **Integrating clickstream data with the other business processes** on the bus matrix
* Assembling a **complete view of profitability** **for a web enterprise**

### Clickstream Source Data

* **The clickstream is NOT just another data source that is extracted, cleaned, + dumped into the DW/BI environment**, it is an **evolving *collection* of data sources**
* There are a number of **server log file formats for capturing clickstream data**, which have ***optional* data components that, if used, can be very helpful in identifying visitors, sessions, + the true meaning of behavior**
* **Because of the *distributed* nature of the web, clickstream data often is collected simultaneously by *different* physical servers, even when a visitor thinks they are interacting with a single website**
* Even if the log files collected by these separate servers are compatible, a very interesting **problem arises in synchronizing the log files after the fact**
* Remember that a busy web server may be processing hundreds of page events per second
* **It is unlikely the clocks on separate servers will be in synchrony to one-hundredth of a second**
* You **also obtain clickstream data from different parties**
* Besides your **own log files**, you may get clickstream data **from referring partners or from ISPs**
* Another important form of clickstream data is the **search specification** **given to a search engine** that then directs the visitor to the website
* Finally, if you are an ISP providing web access to directly connected customers, you have a unique perspective because you see every click of your captive customers that may allow more powerful and invasive analyses of the customer’s sessions.
* The **most basic form of clickstream data from a normal website is stateless**.
* That is, the **log shows an isolated page retrieval event but does not provide a clear tie to other page events elsewhere in the log**
* Without some kind of contextual help, it is difficult/impossible to reliably identify a complete visitor session
* The other big frustration with basic clickstream data is the **anonymity** **of the session**
* Unless visitors agree to reveal their identity in some way, you often cannot be sure who they are, or if you have ever seen them before
* In certain situations, you may not distinguish the clicks of 2 visitors who are simultaneously browsing the website

#### Clickstream Data Challenges

* **Clickstream data contains many ambiguities**
* Identifying visitor origins, visitor sessions, + visitor identities is something of an interpretive art.
* Browser caches + proxy servers make these identifications more challenging.

##### Identifying the Visitor Origin

* If very lucky, your site is the default home page for the visitor’s browser
* Every time they open their browser, your home page is the 1st thing they see
* This is **pretty unlikely unless you are the webmaster for a portal site or an intranet home page,** but many sites have buttons that prompt visitors to set their URL as the browser’s home page
* **Unfortunately, there is no easy way to determine from a log whether your site is set as a browser’s home page**
* A visitor may be **directed to your site from a search at a portal** such as Yahoo or Google
* Such **referrals can come either from the portal’s index**, for which you may have paid a placement fee, **or from a word or content search**
* For some websites, the most common source of visitors is from a **browser bookmark**
* For this to happen, the visitor must have previously bookmarked your site, + this can occur only after the site’s interest + trust levels cross the visitor’s bookmark threshold
* Finally, your site may be reached as a result of a **clickthrough** (a deliberate click on a text or graphical link from another site)
* This may be a paid-for referral via a banner ad, or a free referral from an individual or cooperating site
* **In the case of clickthroughs, the referring site will almost *always* be identifiable as a field in the web event record**
* **Capturing this crucial clickstream data is important to verify the efficacy of marketing programs**
* It **also provides crucial data for auditing invoices you may receive** from clickthrough advertising charges

##### Identifying the Session

* Most **web-centric analyses require** every visitor **session** **(visit)** **to have its own unique identity tag,** similar to a supermarket receipt number 🡪 **the session ID**
* Records for every individual visitor action in a session, whether derived from the clickstream or an application interaction, *must* contain this tag
* But keep in mind **the *operational application*, such as an order entry system, generates this session ID, NOT the web server**
* The **basic protocol for the web, HTTP is stateless** **(lacks the concept of a session)**
* There are no intrinsic login or logout actions built into the HTTP protocol, so **session identity must be established in some other way**
* There are **several ways to do this:**
* **1.** In many cases, the **individual hits comprising a session can be consolidated by collating time-contiguous log entries from the same host (IP address)**
* If a log contains a number of entries w/ the same host ID in a short period of time (ex: 1 hour), you can reasonably assume the entries are for the same session
* **This method breaks down for websites with large numbers of visitors,** because dynamically assigned IP addresses may be reused immediately by different visitors over a brief time period
* Also, **different IP addresses may be used w/in the same session for the same visitor**
* It **also presents problems when dealing with browsers that are behind some firewalls**
* Notwithstanding these problems, **many commercial log analysis products use this method of session tracking, and it requires no cookies or special web server features**
* **2.** Another much more satisfactory method is to **let the web browser place a *session-level* cookie into the visitor’s web browser**
* This **cookie will last as long as the browser is open**, + in general, **won’t be available in subsequent browser sessions**
* The **cookie value can serve as a temporary session ID not only to the browser, but also to *any application that requests the session cookie from the browser***
* BUT using a transient cookie has a **disadvantage = you can’t tell when the visitor returns to the site at a later time in a new session**
* **3. HTTP’s secure sockets layer (SSL)** offers an opportunity to track a visitor session because it **may include a login action by the visitor + the exchange of encryption keys**
* The **downside to using this method is that to track the *session*, the *entire* information exchange needs to be in high-overhead SSL**, + the visitor may be put off by security advisories that can pop up using certain browser
* Also, **each host must have its own unique security certificate.**
* **4.** If **page generation is dynamic**, you **can try to maintain visitor state by placing a session ID in a hidden field of each page returned to the visitor**
* This session ID can be returned to the web server as a query string appended to a subsequent URL
* This method of session tracking **requires a great deal of control over the website’s page generation methods to ensure the thread of a session ID is not broken**
* If the visitor clicks links that don’t support this session ID ping-pong, a single session may appear to be multiple sessions
* This **approach also breaks down if multiple vendors supply content in a single session unless those vendors are closely collaborating.**
* **5.** Finally, the **website may establish a persistent cookie in the visitor’s machine that is not deleted by the browser when the session ends**
* Of course, it’s **possible a visitor will have the browser set to refuse cookies**, or **may manually clean out the cookie file**, so there is **no absolute guarantee that even a persistent cookie will survive**
* Although any given cookie can be read only by the website that caused it to be created, **certain groups of websites can agree to store a common ID tag that would let these sites combine their separate notions of a visitor session into a “super session”**
* In summary, the **most reliable method of session tracking from web server log records is obtained by setting a persistent cookie in the visitor’s browser (option 5 from above)**
* Less reliable, but good results can be obtained by setting a session level + a *non*-persistent cookie + by associating time-contiguous log entries from the same host
* The latter method requires a robust algorithm in the log postprocessor to ensure satisfactory results and to decide when not to take the results seriously

##### Identifying the Visitor

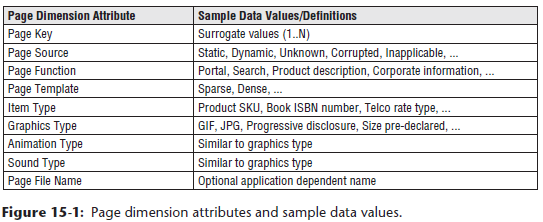
* **Identifying a specific visitor who logs into your site presents some of the most challenging problems facing a site designer, webmaster, or manager of the web analytics group**
* Web visitors ***want* to be anonymous**
* They may have no reason to trust you, the internet, or their computer with personal identification or credit card information
* If you request visitors’ identity, they **may not provide accurate information**
* Can’t be sure which family member is visiting your site
* If you obtain an identity by association, for instance from a persistent cookie left during a previous visit, the **identification is only for the computer, not for the specific visitor**
* Any family member or company employee may have been using that particular computer at that moment in time
* **Can’t assume an individual is always at the same computer**
* Server-provided cookies identify a *computer*, NOT an individual
* If someone accesses the same website from an office computer, home computer, *and* mobile device, a **different website cookie is probably put into each machine**

### Clickstream Dimensional Models

* Before designing clickstream dimensional models, consider *all* the dimensions that may have relevance in a clickstream environment
* *Any single dimensional model will not use all the dimensions at once*, but it is nice to have a portfolio of dimensions waiting to be used
* The **list of dimensions for a web retailer could include:**
* Date, Time of day, Part, Vendor, Status, Carrier, Facilities location, Product, Customer, Media, Promotion, Internal organization, Employee, **Page, Event, Session, Referral**
* All the dimensions listed above, except for the last four bolded, are familiar dimensions, most of which used in earlier chapters of this book.
* But the **last 4 are the unique dimensions of the clickstream and warrant some careful attention**

#### Page Dimension

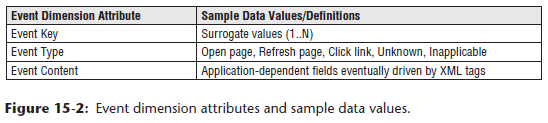
* The **page dimension**describes the **page context for a web page event**, as illustrated below



* The **grain of this dimension is the individual page**
* The **definition of “page” must be flexible enough to handle the evolution of web pages from static page delivery to highly dynamic page delivery in which the exact page the customer sees is unique at that instant in time**
* Assume even in the case of the dynamic page that there is a well-defined function that characterizes the page, + we will use that to describe the page
* **Will NOT create a page row for every instance of a dynamic page, because that would yield a dimension with an astronomical number of rows**
* These rows also would not differ in interesting ways, + **you want a row in this dimension for each interesting distinguishable type of page**
* **Static pages probably get their own row, but dynamic pages would be grouped by similar function + type**
* **When the definition of a static page changes because it is altered by the webmaster, the page dimension row can either be type 1 SCD overwritten or treated w/ an alternative SCD**
* This decision = a matter of policy for the DW, depending on whether old + new descriptions of a page differ materially, + whether an old definition should be kept for historical analysis purposes
* **Website designers, data governance reps from the business, + the DW/BI architects need to collaborate to assign descriptive codes + attributes to each page served by the web server, *whether the page is dynamic or static***
* ***Ideally*, web page developers supply descriptive codes + attributes with each page they create + embed these codes + attributes into the optional fields of the web log files**
* **This crucial step is at the foundation of the implementation of this page dimension**
* Before leaving the page dimension, note that **some internet companies track the more granular individual elements on each page of their web sites, including graphical elements + links**
* Each element generates its own row for each visitor for each page request
* A single complex web page **can generate** **100s of rows each time the page is served to a visitor**
* This extreme granularity generates astronomical amounts of data, often exceeding 10TB per day
* Similarly, gaming companies may generate a row for every gesture made by every online game player, which again can result in 100s of millions of rows per day.
* In both of the above cases, the **most atomic fact table will have extra dimensions** describing the graphical element, link, or game situation

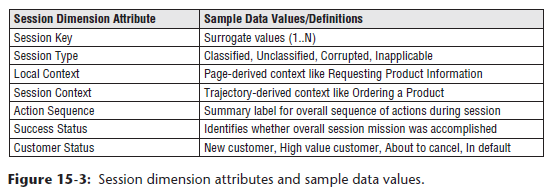
#### Event Dimension

* The event dimension **describes what happened on a particular page at a particular point in time**
* The **main interesting events are: Open Page, Refresh Page, Click Link, + Enter Data**
* You want to capture that information in this **small event dimension**, as illustrated below



#### Session Dimension

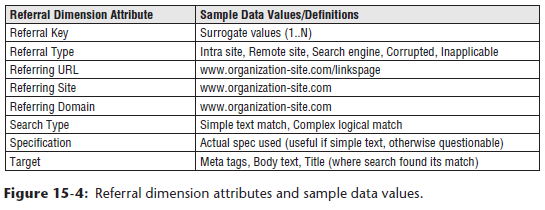
* The **session dimension** provides **1 or more levels of diagnosis for the visitor’s session as a whole**, as shown below



* Ex: The *local* context of the session might be Requesting Product Information, but the *overall* session context might be Ordering a Product
* Success status would diagnose whether the mission was completed
* The local context may be decidable from just the identity of the current page, but the overall session context probably can be judged only by processing the visitor’s *complete* session at data extract time
* **Customer status** attribute = a convenient place to **label the customer for periods of time**, with labels that are **not clear either from the page or immediate session**
* These statuses may be derived from auxiliary business processes in the DW/BI system, but by placing these labels deep within the clickstream, you can directly study the behavior of certain types of customers
* **Do NOT put these labels in the customer dimension because they may change over very short periods of time**
* If there are a **large number of these statuses, consider creating a separate customer status mini-dimension** rather than embedding this information in the session dimension
* This dimension groups sessions for analysis, such as:
* How many customers consulted your product information before ordering?
* How many customers looked at your product information and never ordered?
* How many customers did not finish ordering? Where did they stop?

#### Referral Dimension

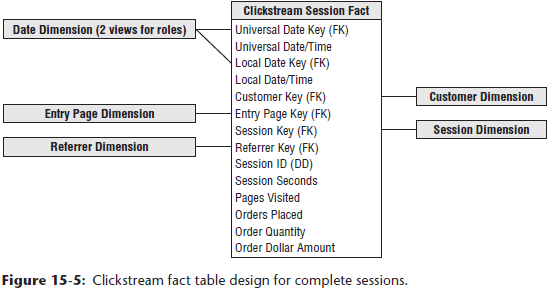
* The **referral dimension**, illustrated in below, describes *how* a customer arrived at the current page



* The **web server logs usually provide this information.**
* The URL of the previous page is identified, + in some cases additional information is present
* If the referrer was a search engine, usually the search string is specified
* May not be worthwhile to put the raw search specification into your database, because the **search specifications are so complicated and idiosyncratic that an analyst may not be able to query them usefully**
* Assume some kind of simplified + cleaned specification is placed in the **specification** attribute

#### Clickstream Session Fact Table

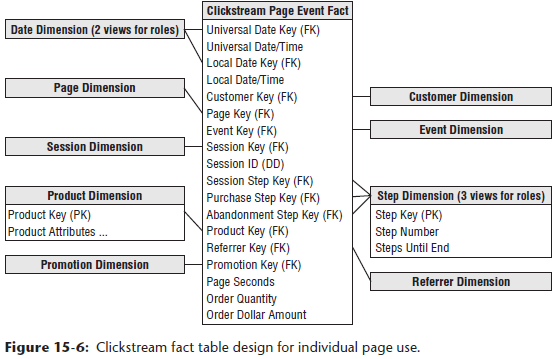
* Now that you have a portfolio of useful clickstream dimensions, you can design the primary clickstream dimensional models based on the web server log data
* This business process can then be integrated into the family of other web retailing subject areas
* **With an eye toward keeping the 1st fact table from growing astronomically, choose the grain to be 1 row for each *completed* customer session.**
* This grain is **significantly higher than the underlying web server logs**, which record each individual page *event*, including individual pages as well as each graphical element on each page
* While we **typically encourage designers to start with the most granular data available in the source system, this is a purposeful deviation from our standard practices**
* Perhaps you have a big site recording > 100M page fetches/day, + 1B micro-page events (graphical elements), but want to start w/ a more manageable number of rows loaded each day
* Assume for the sake of argument that the 100M page fetches boil down to 20M complete visitor sessions, which could arise if an average visitor session touched 5 pages
* The **dimensions appropriate for this 1st fact table are calendar date, time of day, customer, page, session, + referrer**
* Finally, you **can add a set of measured facts for this session including session seconds, pages visited, orders placed, units ordered, + order dollars**
* The completed design is shown below



* There are a number of interesting aspects to this design
* You may wonder why there are *two* connections from the calendar date dimension to the fact table and *two* datetime stamps
* **This is a case in which *both* the calendar date + the time of day must play two *different* roles**
* Because you’re **interested in measuring precise times of sessions, you must meet 2 conflicting requirements:**
* **1)** Want to **make sure you can synchronize all session dates + times internationally across multiple time zones**
* Perhaps you have other date + time stamps from other web servers or non-web systems elsewhere in the DW/BI environment
* **To achieve *true* synchronization of events across multiple servers + processes, you must record *all* session dates + times, *uniformly*, in a *single* time zone,** such as Greenwich Mean Time (GMT) or Coordinated Universal Time (UTC)
* **Interpret the session date + time combinations as the beginning of the session.**
* Because you have the dwell time of the session as a numeric fact, you can tell when the session ended, if that is of interest
* **2) To record the date and time of the session relative to the visitor’s wall clock**
* The best way to represent this information is with a *second* calendar date FK and datetime stamp
* Theoretically, you ***could* represent the time zone of the customer in the customer dimension table, but constraints to determine the correct wall clock time would be horrendously complicated**
* Time differences between 2 cities (such as London and Sydney) can change by as much as 2 hours at different times of the year, depending on when these cities go on and off daylight savings time
* This is **NOT the business of the BI reporting application to work out, but is the business of the database to store this information, so it can be constrained in a simple + direct way**
* The **two role-playing calendar date dimension tables are Views on a *single* underlying table**
* **Column names are massaged in the view definition, so they are slightly different when they show up in the UI pick lists of BI tools**
* **Note that the use of views makes the 2 instances of each table semantically independent**
* We **modeled the exact instant in time with a full datetime stamp** rather than a time-of-day dimension
* Unlike the calendar date dimension, a **time-of-day dimension would contain few if any meaningful attributes**
* You don’t have labels for each hour, minute, or second
* Such a time-of-day dimension could be ridiculously large if its grain were the individual second or millisecond
* Also, the **use of an explicit datetime stamp allows direct arithmetic between different date/time stamps to calculate precise time gaps between sessions, even those crossing days**
* Calculating time gaps using a time-of-day dimension would be awkward
* The inclusion of the entry page dimension in the above schema may seem surprising, given the grain of the design is the customer session
* However, **in a given session, a very interesting page is the entry page**
* The **page dimension in *this* design is the page the session started with**
* In other words, how did the customer hop onto your bus just now?
* **Coupled with the referrer dimension, you now have an interesting ability to analyze how and why the customer accessed your website**
* A **more elaborate** design would also add an **exit page dimension**
* You may be tempted to add the causal dimension to this design, but **if the causal dimension focuses on individual products, it would be inappropriate to add it to this design**
* The symptom that the causal dimension does not mesh with this design is the multivalued nature of the causal factors for a given complete session
* If you run ad campaigns or special deals for several products, how do you represent this multivalued situation if the customer’s session involves several products?
* The **right place for a product-oriented causal dimension will be in the more fine-grained table** described in the next fact table
* Conversely, a more broadly-focused market conditions dimension that describes conditions affecting all products would be appropriate for a session-grained fact table
* **Session seconds fact** = **total number of seconds a customer spent on the site during this session**
* There will be **many cases in which you can’t tell when the customer left**
* Perhaps the customer typed in a new URL, which won’t be detected by conventional web server logs (If the data is collected by an ISP who can see every click across sessions, this particular issue goes away), or the customer got up out of the chair and didn’t return for 1 hour, or the customer just closed the browser without making any more clicks
* In all these cases, your **extract software needs to assign a small + nominal number of seconds to this last session step, so the analysis is not unrealistically distorted**
* We purposely designed this first clickstream fact table to focus on *complete* visitor sessions *while keeping the size under control*
* The next schema drops down to the lowest practical granularity you can support in the DW: the individual page event

#### Clickstream Page Event Fact Table

* **Granularity of the 2nd clickstream fact table is the individual page event in each customer session**
* Underlying micro-events recording graphical elements such as JPGs and GIFs are discarded (unless you are Yahoo or eBay as described previously)
* **Simple static HTML pages can record only 1 interesting event per page view** = **the page view**
* But **as websites employ dynamically created XML-based pages, with the ability to establish an on-going dialogue through the page, the number and type of events will grow**, + this fact table could become astronomical in size
* Resist the urge to aggregate the table up to a coarser granularity because that inevitably involves dropping dimensions
* Actually, the 1st clickstream fact table above represents just such an aggregation
* Although it is a worthwhile fact table, analysts cannot ask questions about visitor behavior or individual pages
* **Having chosen the grain, you can choose the appropriate dimensions**
* The list of dimensions includes: calendar date, time of day, customer, page, event, session, session ID, step (*3 roles*), product, referrer, + promotion
* The completed design:



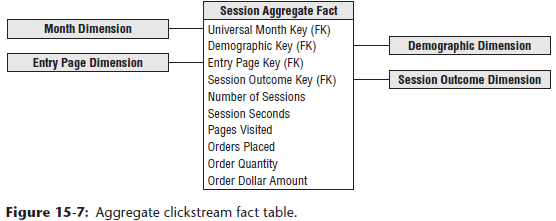
* This looks similar to the 1st design, except for the addition of the **page, event, promotion, + step dimensions**
* This **similarity between fact tables is typical of dimensional models**
* **One of the charms of dimensional modeling = the “boring” similarity of the designs**, which is **where they get their power**
* **When the designs have a predictable structure, all the software up and down the DW/BI chain, from extraction, to database querying, to the BI tools, can exploit this similarity to great advantage**
* The **2 roles played by the calendar date and datetime stamps** have the same interpretation as in the 1st design
* One role is the **universal synchronized time**, the other is the **local wall clock time as measured by the customer**
* In this fact table, these dates and times refer to the individual page event
* The **page dimension** refers to the **individual page**
* This is the **main difference in grain between the 2 clickstream fact tables** 🡪 **In *this* fact table you can see *all* the pages accessed by the customers**
* As described earlier, the **session dimension** describes the **outcome of the session**
* A companion column, **session ID, is a degenerate dimension that does not have a JOIN to a dimension table**, a typical dimensional modeling construct
* The **session ID is simply a unique identifier, with no semantic content, that serves to group together the page events of each customer session in an unambiguous way**
* Did not need a session ID degenerate dimension in the 1st fact table, but it is included as a “parent key” if you want to easily link to the individual page event fact table
* **Recommended that the session dimension be at a higher level of granularity than the session ID**, since the **session dimension is intended to describe classes + categories of sessions, NOT the characteristics of each individual session**
* A **product dimension** in this design is from the assumption this website belongs to a web retailer
* A financial services site probably would have a similar dimension, a consulting services site would have a service dimension, an auction site would have a subject or category dimension describing the nature of the items being auctioned, + a news site would have a subject dimension, although with different content than an auction site.
* Should accompany the product dimension with a **promotion dimension** so you can **attach useful causal interpretations to the changes in demand observed for certain products**
* **For each page event, you should record the number of seconds that elapse before the *next* page event**
* Call this **page seconds** to contrast it with session seconds in the 1st fact table
* *This is a* ***simple example of paying attention to conformed facts***
* If you call both of these measures simply “seconds,” you risk having these seconds inappropriately added or combined
* **Because these seconds are *not* precisely equivalent, name them differently as a warning**
* In this particular case, you would expect the page seconds for a session in this 2nd fact table to add up to the session seconds in the 1st fact table
* The final facts are **units ordered** and **order dollars**
* These columns will be **0 or NULL for many rows in this fact table if the specific page event is not the event that places the order**
* Nevertheless, it is **highly attractive to provide these columns because they tie the all-important web revenue directly to behavior**
* If the units ordered + order dollars were **only available through the production order entry system elsewhere** in the DW/BI environment, it would be **inefficient to perform the revenue-to-behavior analysis across multiple large tables**
* **In many DBMSs, these NULL facts are handled efficiently + may take up literally 0 space in the fact table**

##### Step Dimension

* Because the **fact table grain is the individual page event**, you can add the powerful **step dimension** described in Chapter 8: CRM**, which provides the position of the specific page event within the overall session**
* The step dimension **becomes particularly powerful when attached to the fact table in various roles**
* The schema above shows 3 roles: **overall session**, **purchase sub-session**, **abandonment sub-session**
* A **purchase sub-session**, by definition, **ends in a successful purchase**
* An **abandonment sub-session** **fails to complete a purchase transaction for some reason**
* Using these roles of the step dimension allows some very interesting queries
* Ex: If the purchase step dimension is constrained to step number 1, the query returns nothing but the starting page for successful purchase experiences
* Conversely, if the abandonment step dimension is constrained to 0 steps remaining, the query returns nothing but the last + presumably most unfulfilling pages visited in unsuccessful purchase sessions
* **Although the whole design shown the above schema is aimed at product purchases, the step dimension technique can be used in the analysis of *any* sequential process**

#### Aggregate Clickstream Fact Tables

* **Both clickstream fact tables designed thus far are pretty large**, + there are many **business questions** that would be **forced to summarize millions of rows from these tables**
* Ex: To track the total visits + revenue from major demographic groups of customers accessing your website on a month-by-month basis, you can certainly do that with either fact table
* In the session-grained fact table, you’d constrain the calendar date dimension to the appropriate time span, + then create row headers from the demographics type attribute in the customer dimension + the month attribute in the calendar dimension (to separately label the months in the output)
* Finally, you’d sum the Order Dollars and count the number of sessions
* This all works fine, but it is **likely to be slow without help from an aggregate table**
* **If this kind of query is frequent, the DBA will be encouraged to build an aggregate table**, as shown below



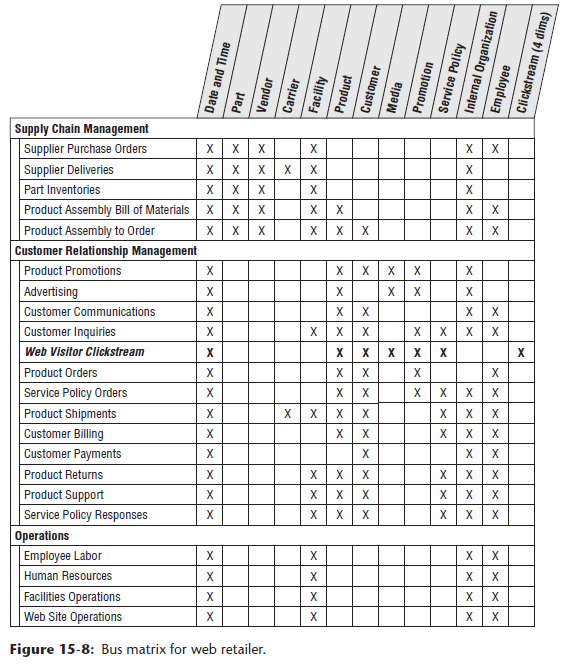
* **Can build this table directly from the 1st fact table, whose grain is the individual session**
* To build it, you GROUP BY month, demographic type, entry page, + session outcome, count the number of sessions, + sum all the other additive facts
* This results in a drastically smaller fact table, almost certainly < 1% of the original session-grained fact table
* This **reduction in size translates directly to a corresponding increase in performance for most queries**
* i.e., you can expect queries directed to this aggregate table to run at least 100X as fast
* May not have been obvious, but we followed a careful discipline in building the aggregate table
* This **aggregate fact table is connected to a set of shrunken rollup dimensions *directly related* to the original dimensions in the more granular fact tables**
* The month dimension is a conformed subset of the calendar day dimension’s attributes
* The demographic dimension is a conformed subset of customer dimension attributes
* Should assume the page and session tables are unchanged, + a careful design of the aggregation logic could suggest a conformed shrinking of these tables as well

#### Google Analytics

* Google Analytics (GA) = a service provided by Google best described as an external DW that provides many insights about how your website is used.
* To use GA, you modify each page of your website to include a GA tracking code (GATC) embedded in a Java code snippet located in the HTML <head> declaration of each page to be tracked
* When a visitor accesses the page, information is sent to the Analytics service at Google, as long as the visitor has JavaScript enabled
* Virtually all the information described in this chapter can be collected through GA, w/ the exception of personally identifiable information (PII), which is forbidden by GA’s terms of service
* GA can be combined with Google’s Adword service to track ad campaigns + conversions (sales)
* Reportedly, GA is used by > 50% of the most popular web sites on the internet
* Data from GA can be viewed in a BI tool dashboard online directly from the underlying GA databases, or data can be delivered to you in a wide variety of standard + custom reports, making it possible to build your own local business process schema surrounding this data
* Interestingly, GA’s detailed technical explanation of the data elements that can be collected through the service are described correctly as either dimensions or measures

### Integrating Clickstreams into Web Retailer’s Bus Matrix

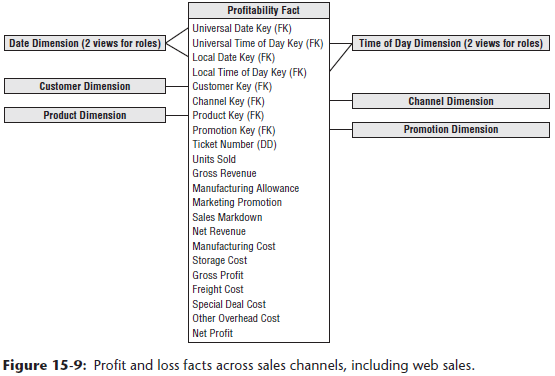
* This section considers the business processes needed by a web-based computer retailer
* The retailer’s EDW bus matrix is illustrated below



* Note **the matrix lists business process subject areas, NOT individual fact tables**
* *Typically*, each matrix row results in a suite of closely associated fact tables and/or OLAP cubes, which all represent a particular business process
* The above matrix has a number of striking characteristics
* There are a *lot* of check marks
* Some dimensions, such as datetime, organization, + employee appear in almost *every* business process
* The product and customer dimensions dominate the middle part of the matrix, where they are attached to business processes that describe customer-oriented activities
* At the top of the matrix, suppliers + parts dominate the processes of acquiring the parts that make up products + building them to order for the customer
* At the bottom of the matrix, you have classic infrastructure + cost driver business processes that are not directly tied to customer behavior
* The web visitor clickstream subject area sits squarely among the customer-oriented processes
* It shares the datetime, product, customer, media, causal, + service policy dimensions with several other business processes nearby
* In this sense it should be obvious that the web visitor clickstream data is well integrated into the fabric of the overall DW/BI system for this retailer
* **Applications tying the web visitor clickstream will be easy to integrate across all the processes sharing these conformed dimensions because separate queries to each fact table can be combined across individual rows of the report**
* The **web visitor clickstream business process contains the 4 special clickstream dimensions not found in the others**
* These dimensions do not pose a problem for applications
* Instead, **the ability of the web visitor clickstream data to bridge between the web world and the brick-and-mortar world is exactly the advantage you are looking for**
* You can constrain + group on attributes from the 4 web dimensions + explore the effect on the other business processes
* Ex: Can see what kinds of web experiences produce customers who purchase certain kinds of service policies and then invoke certain levels of service demands
* Finally, it should be pointed out that **the matrix serves as a kind of communications vehicle for all the business teams + senior management to appreciate the need to conform dimensions + facts**
* **A given column in the matrix is, in effect, an invitation list to the meeting for conforming the dimension**

### Profitability Across Channels Including Web

* After the DW/BI team successfully implements the initial clickstream fact tables + ties them to the sales transaction + customer communication business processes, the team may be ready to tackle **the most challenging subject area of all: web profitability**
* You **can tackle web profitability as an extension of the sales transaction process**
* Fundamentally, you are **allocating all the activity + infrastructure costs down to each sales transaction**
* *Could*, as an alternative, try to build web profitability *on top* of the clickstream
* But this would involve an even more controversial allocation process in which you allocate costs down to each session
* It would be hard to assign activity + infrastructure costs to a session that has no obvious product involvement and leads to no immediate sale.
* A **big benefit of extending the sales transaction fact table is that you get a view of profitability across all sales channels, not just the web**
* In a way, this should be obvious, because you know that you must sort out the costs + assign them to the various channels
* The **grain of the profit + loss (P&L) facts is each individual line item sold on a sales ticket to a customer at a point in time, whether it’s a single sales ticket or single web purchasing session**
* This is the **same as the grain of the sales transaction business process, + includes all channels, assumed to be store sales, tele-sales, + web sales**
* The **dimensions of the P&L facts are also the same as the sales transaction fact table: date, time, customer, channel, product, promotion, and ticket number (degenerate)**
* The **big difference between the profitability and sales transaction fact tables is the breakdown of the costs**, as illustrated below



* Before discussing the allocation of costs, examine the format of the P&L facts
* It is organized as a simple P&L statement
* The 1st fact is the familiar “units sold”, + all other facts are dollar values, beginning with the value of the sale as if it were sold at the list or catalog price, referred to as gross revenue.
* Assuming sales often take place at lower prices, you’d account for any difference with a manufacturer’s allowance, marketing promotion that is a price reduction, or markdown done to move the inventory
* When these effects are taken into account, you can calculate the net revenue, which is the true net price the customer pays times the number of units purchased
* The rest of the P&L consists of a series of subtractions, where you calculate progressively more far-reaching versions of profit
* Begin by subtracting the product manufacturing cost if you manufacture it, or equivalently, the product acquisition cost if it is acquired from a supplier
* Then subtract the product storage cost
* At this point, many enterprises call this partial result the gross profit
* Can divide this gross profit by the gross revenue to get the gross margin ratio
* Obviously, the **net revenue and gross profit columns are calculated directly from the columns immediately preceding them in the fact table**
* **But should you explicitly store these columns in the database?**
* The **answer depends on whether you provide access to this fact table through a View or whether users or BI applications directly access the physical fact table**
* The structure of the P&L is sufficiently complex that, as the DW provider, you **don’t want to risk the important measures like net revenue and gross profit being computed incorrectly**
* If you **provide all access through Views, you can easily provide the computed columns without physically storing them**
* But **if users are allowed to access the underlying physical table, include net revenue, gross profit, and net profit as physical columns**
* Below the gross profit, you can continue subtracting various costs
* Typically, the **DW/BI team must separately source or estimate each of these costs**
* Remember the actual entries in any given fact table row are the fractions of these total costs allocated all the way down to the individual fact row grain
* Often there is significant pressure on the DW/BI team to deliver the profitability business process
* Or to put it another way, there is tremendous pressure to source all these costs
* **But how good are the costs in the various underlying data sets?**
* Sometimes a cost is only available as a national average, computed for an entire year
* Any allocation scheme is going to assign a kind of pro forma value that has no real texture to it
* Other costs will be broken down a little more granularly, perhaps to calendar quarter + by geographic region (if relevant)
* Finally, some costs may be truly activity-based + vary in a highly dynamic, responsive, and realistic way over time
* **Website system costs are an important cost driver in electronic commerce businesses**
* Although website costs are classic infrastructure costs, + are therefore difficult to allocate directly to the product and customer activity, this is a **key step in developing a web-oriented P&L statement**
* Various allocation schemes are possible, including allocating the website costs to various product lines by the number of pages devoted to each product, allocating the costs by pages visited, or allocating the costs by actual web-based purchases
* **The DW/BI team *cannot* be responsible for implementing activity-based costing (ABC) in a large organization**
* When the team is building a profitability dimensional model, the team gets the best cost data available at the moment and publishes the P&L
* Perhaps some of the numbers are simple rule-of-thumb ratios, + others may be highly detailed activity-based costs
* Over time, as the sources of cost improve, the DW/BI team incorporates these new sources and notifies the users that the business rules have improved
* Before leaving this design, it is worthwhile putting it in perspective
* **When a P&L structure is embedded in a rich dimensional framework, you have immense power**
* **Can break down all the components of revenue, cost, + profit for every conceivable slice + dice provided by the dimensions**
* **Can answer *what* is profitable, but also answer “*why*” because you can see all the components of the P&L**, including:
* How profitable is each channel (web sales, tele-sales, + store sales)? Why?
* How profitable are your customer segments? Why?
* How profitable is each product line? Why?
* How profitable are your promotions? Why?
* When is your business most profitable? Why?
* The **symmetric dimensional approach enables you to combine constraints from many dimensions, allowing compound versions of the profitability analyses** like:
* Who are the profitable customers in each channel? Why?
* Which promotions work well on the web but do not work well in other channels? Why?