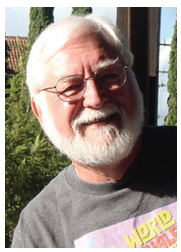
A data structure for customer insights

Received (in revised form): 17th February, 2017



Jim Porzak
is a semi-retired data scientist specialising in data-driven customer insights using customer behavioral and demographic data to predict propensity to purchase and/or churn, do uplift modeling, segment customers based on cluster analysis and undertake routine marketing analytics. Jim is very active in the open-source community, particularly with R-the open-source software environment for statistical computing and graphics. He is a frequent speaker at conferences in the US & Europe.

DS4CI.org, 7829 Terrace Drive, El Cerrito, CA 94530, USA
Email: Jim@DS4CI.org

Abstract Data-driven customer focused organisations front line business analysts and data scientists have an overwhelming collection of data about their customers. A simple data structure is presented to organise those disparate data in a way that is focused on enabling the easy discovery of customer insights by business analysts. Focusing on the customer when collecting and summarising source data ensures relevant data elements, with a general frame work for both subscription and product organisations described. A concrete example of the customer insights data mart built for one subscription business is followed by four examples of actual business questions that were answered with simple queries.

KEYWORDS: DBMS, SQL, data lake, data mart, data warehouse, customer analytics

INTRODUCTION

Some big-data proponents hype the idea of a data lake which 'is a storage repository that holds a vast amount of raw data in its native format, including structured, semi-structured, and unstructured data. *The data structure and requirements are not defined until the data is needed*.'[1] (Emphasis added.)

Contrast this with the metaphor of the data by Ralph Kimball about the warehouse manager as a publisher,[2] in which '… the correct job description for a data warehouse project manager is *publish the right data*.' (Emphasis his.) This work is inspired by the philosophy of Kimball.

Enabling a customer focused organisation to efficiently and reliably make data driven decisions requires a data source that is curated with the same rigor as any world-class newspaper: in short it must be relevant, accurate, complete, timely and accessible. Focusing on customer data, packaged in a business-friendly structure, provides business analysts and data scientists with the data to make timely business decisions and build advanced customer models.

The general data structure described here has been used to design and build customer insight (CI) data marts in two midsized B-to-C businesses, one subscription based and the other merchandise. Marketing, product, finance and operations teams queried their CI mart daily while e-mail teams at both companies depended on their CI mart for customer focused messaging and targeting data.

The CI mart data structure has two major benefits. First, business analysts can get consistent data driven answers to most of their customer related questions *by themselves*. Second, data scientists have a clean and rich data set to use as a basis for modeling and segmentation *that will sync-up with the business user view*.

ACHIEVING CUSTOMER FOCUS

The CI data mart is designed to model customer actions, decisions and eventually motivations, meaning that the data architect must put him or herself in the mind of the customer. Both the organisation and data elements of the CI data mart should mimic the interactions of the customer with the organisation, with the challenge distilling the raw data to surface these *customer initiated actions* (CIAs).

'Customer' in interpreted broadly as anyone who gives the organisation money, did so in the past or is expected to do so in the future. Similarly, the organisation could be a business selling products or services or even a charity.

CONSUMERS OF CI DATA

The well-designed CI data mart will be the go-to resource for all customer driven groups and services in the organisation. Different roles will have different needs.

Business analysts

These front-line goal focused analysts are tasked with understanding customers and possibly communicating with them. They are often embedded in departments and may not have 'analyst' in their job title: their analytic tools include Excel and, perhaps, Tableau or another front-end tool. They are the domain experts of the organisation and typically on a strict deadline.

Data scientists

Data scientists do the deep dive analysis of customer data producing cluster based segments, propensity models (purchase, churn, upgrade, etc) and recommendations for next best offer, message, and product.

Conventional wisdom holds that data scientists spend at least 80 per cent of their time on data preparation. The CI mart will greatly reduce their data preparation effort, a significant change given the high demand and relatively small number of data scientists. More importantly, as they are basing their models on the same CI data set used by the rest of the organisation, their results should be compatible with the business analysts' findings.

Data scientists also provide data back to the CI mart by loading individual customer level scores, segments and recommendations which can be used to enhance reports and for one-to-one messaging.

Data engineers

Data engineers have a dual role. First of all they implement, maintain, and enhance of the CI mart. They are also consumers as the CI mart can provide data integratory signals and alarms when there are issues with other parts of the business data flow. For example, an abrupt change in the rate of orders or consumption could be a signal of a website issue.

Today, data engineers are typically the go-to group for business analysts when they need customer related data. With the CI mart the analysts can fulfill many of these data needs themselves or at most, with some hints from a data engineer.

Operational systems

Operational systems that communicate with the customer can use the CI mart as a source of scores, segments and targeting recommendations. Retention marketers can export individual customer properties to their email, site messaging and other marketing systems enabling more relevant messaging and targeting.

DATA SOURCES

The customer is an individual and hence in our data mart there should be one unique identifier for each customer. In practice, different data sources may have their own distinct individual identifiers (one challenge will be reliably mapping customer IDs between the various systems).

Some data sets, through negligence, may have no identifier, such as when the author

has been given survey and data append data lacking customer IDs. Unfortunately these data must be discarded and the effort that went into gathering them is wasted. Hopefully, this error of omission will not be repeated the next time a survey or data append is run.

Operational data

Most of what we know about the customer will come from operational systems. The systems that record purchases and deliveries (of product or services) will be most reliable as they are fundamental to the organisation. These systems however deal with the real-world complications associated with collecting money and shipment of goods or services, leading to a classic case of too much information.

Figure 1 illustrates the difference between the subscriber actions of starting a subscription and, perhaps, cancelling or up/downgrading their subscription (after cancelling the subscriber may subsequently make the decision to subscribe again). At most there are a few CIAs (customer initiated actions) while the subscription processing system, on the other hand, needs to deal with all the complexities of collecting money from the subscriber, resulting in many more events. Also, many subscription processing systems chop what the subscriber views as a continuous subscription into the individual payment cycles which are often confusingly called 'subscriptions.'

Occasionally, operational data sets include fields which promise background and qualitative insights about the customer: some examples are *jobtitle, introduced_by, reason_for_return*. The quality and usability of these fields is often severely lacking as they are not important for the main purpose of the operational system nor its users. An order processing system, for example, may have such field, but they are of no importance to the finance and operation groups who ignore them.

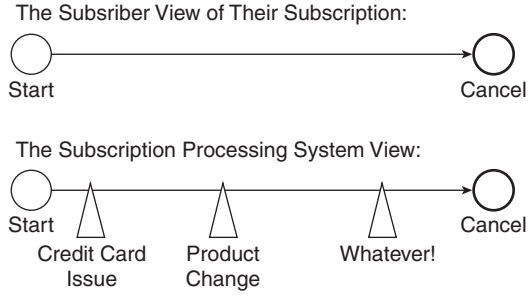


Figure 1: A subscription through eyes of subscriber vs. operational system

Interaction data

Today we get many signals directly from our customers: they browse or search websites, respond to emails, answer surveys, use social media and call or chat with customer service. If the operational systems tell us what customers do interaction data can tell us about their motivation and interests. But these data are very noisy and cluttered with measurement artifacts requiring a fair amount of pre-processing to get measures useful for customer insights. Web logs are the classic example in which most log events result from the mechanics of displaying the page and are not directly customer initiated.

Customers also respond to classical advertising and PR in the press, on television or other media, but these are one-way interactions that provide no direct signal that the customer actually saw the message. At best the prospect may respond to a 'How did you hear about us?' question.

Data appends

Individual-level appends based on customer name and address are available from Acxiom[3] and other data brokers[4].

Neighbourhood demographics are based on the geo-location of the customer and, in the best case, on their street address. A nine-digit US ZIP code can be mapped to the US Census block group, the smallest area for which American Community Survey[5] (ACS) data is available. The ACS has neighbourhood level demographic, economic, housing and social data.

The Claritas PRIZM[6] segments are also US geo-location based but adds 15 groups and

66 segments to characterise the neighbourhood for marketing users. PRIZM is now supplied by Nielsen.

When no street address is available the IP address[7] of the website visitor can give a rough geo-location estimate, although this raises several accuracy issues.

Analytic results

The data science team will be creating individual level propensity scores, cluster based segmentations and recommendations. These should be loaded back to the CI mart so they are available to the marketing email, site, banner, and direct mail systems.

THE CI DATA STRUCTURE

The basic design principles are:

- Be analyst ready
- Be front end tool agnostic
- Model customer decision events
- Have three levels of abstraction: Customer, Orders, and Consumption

Being *analyst ready* is inspired by the Kimball publisher model.[8] The CI mart must be easy to use, both technically and conceptually: any contemporary business analyst should already possess the skills to use it. The tables and columns must be intuitive and well labeled so any customer focused business analyst will know what to expect: it should be complete in the sense that most CI questions can be answered by the contents. It goes without saying that it must be accurate and timely, with the longest query response time taking no more than a minute. Finally, it must easily evolve with changing business environments and requirements.

The CI mart is a pure data layer and front-end tool agnostic. Any modern front end tool with JDBC or ODBC connectivity can access it. Experienced analysts and data scientists have their preferred tool sets: if the Ci mart is to become widely adopted, it must support almost any front-end tool.[9]

To remove the barriers to discovering customer insights the CI data elements must be as close to the customer decisions as possible: all noise introduced by operational systems and event logs must be removed.

THREE LEVELS OF ABSTRACTION

Having a simple data structure makes the CI mart easy to understand by everyone in the organisation. Levels modeling the customer, payments and consumption are implemented by these tables:

1. Customer Summary – *everything* about the individual customer.
2. Order Summary – *everything* about individual orders.
3. Consumption Summary – *everything* about consumption of what was ordered.

Here *everything* is an evolving goal, not initial requirement. As we add new data sources the scope of *everything* will broaden.

In practice these are very wide tables or views. In the worst case only two joins are needed to query across the full data structure.

The big idea is the abstraction level of the business question should be answerable at the corresponding CI mart level. For example, if the business question involves properties of individuals then data in the Customer Summary table should answer the question.

Each lower level should be summarised in the upper level. Order Summary will have details of consumptions associated with the order. Customer Summary will have a summary of orders.

While we focus on these three levels of abstraction in practice additional 'helper tables' are often useful. The example data mart discussed later has two helper tables.

TUNE SPECIFICS TO THE ORGANISATION

The names and, of course, content should be tuned to the specific nature of the business. The names should be as much as possible based on the specific vocabulary used in the

Table 1: Naming levels of abstraction based on type of business

Generic	Subscription	Product
1. Customer	1. Subscriber	1. Customer
2. Order	2. Subscription chain	2. Order
3. Consumption	3. Usage stint	3. Order detail

organisation. Table 1 shows the default level names for the generic model, a subscription based business and a product based business.

There can be many order rows per customer and many consumption rows per order. *Subscription Chain* indicates many operational system subscription events may be combined into a contiguous chain to model the decision points of the subscription: 1) 'I start a subscription.' and, perhaps, 2) 'I stop or change my subscription.'

Usage Stint indicates many usage events may be combined into a consumption session, such as one visit to the exercise club (using various machines) or one online session of viewing a number of training videos. Again, we are modeling the decision points of the customer. Refer to Figure 1.

The columns in each table are a mix of general measures with specifics for the organisation. The following tables compare typical column content for subscription versus product organisations at the three levels of abstraction. The exact mix of columns evolved as the two actual CI marts were rolled-out.

Level 1 – customers

This top level of abstraction is all about the individual customer. See Table 2.

The subscription and product variants differ mostly in the naming of the data elements. Substantively, subscribers are active over a time period while product

Table 2 – Level 1: Subscriber vs. customer level table content

Subscriber summary	Customer summary	Notes
Subscriber ID	Customer ID	The primary key
First name	First name	As a sanity check, no other PII in CI mart
Is current subscriber	Is current customer	Per business rule
Initial & last subscription dates	Initial & last order dates	
Initial & last subscription type	Initial & last primary product	
Last subscription chain end on	Last order on	Recency
# Chains, # Payments, # Products, & # Stints	# Orders, # Product groups, & # SKUs	Frequencies
RTD, RTD 1st Chain, RTD <initial x months>	RTD, RTD <initial x months>, average order value	Revenue to date (monetary)
# Days, # Days subscribed, % Coverage	# Days (from first order)	Tenure
Stints per [year, qtr, month], frequency variance	Orders per [year, qtr, month], frequency variance	Consumption rates
Usage type RFMs	Seasonal & life-stage RFMs	Recency, frequency, & monetary by product focus.
Initial channel, offer, search, …	Initial channel, offer, search, …	Acquisition details
Usage type	Product group	Initial consumption
Site, email, call centre, …	Site, email, call centre, …	General engagement
Neighbourhood & appended	Neighbourhood & appended	Demographic appends
YYYY, YYQQ, YYMM	YYYY, YYQQ, YYMM	Cohort segments based on initial payment
Segment(s) & score(s)	Segment(s) & score(s)	From the data science team
Other identifiers	Other identifiers	Links to other systems, eg email, CRM
???	???	Additional organisation specific customer level metrics

Notes: CI: Customer Insight, PII: Personally Identifiable Information, RFM: Recency, Frequency & Monetary and RTD: Revenue To Date.

customers place one order at a time. For tenure, the basic measure is how many days since the first payment: in addition, for subscribers there are the number of days (within the tenure period) the subscriber is subscribed and the percentage of tenure period they are subscribed.

The first name, while strictly a 'personally identifiable information' (PII) data element, is included as a reality check that the record is for the expected individual. Other PII data is only kept in those operational systems that need it such as fulfilment, billing, email etc.

RFM (Recency, Frequency, and Monetary) measures are classic direct marketing measures[10] for predicting response to a mailing. RFM measures are still important today as they are typically very important predictors when building propensity models (especially recency and frequency). Additionally, breadth of engagement with the organisation is a top predictor when modeling.

To account for subscriber interests and customer seasonal and life-stage patterns RFM is also broken out for those variables. For example, RFM for a highly seasonal product line such as holiday gifts will be quite different from RFM for a commodity product like groceries.

The initial channel, offer, search etc are important for the marketing acquisition team. Looking at RTD as a function of initial acquisition conditions can guide allocation of marketing spend. It is also important for modeling to get an idea of what initially motivated the customer: the initial consumption, usage type or product group, based on summarising the consumption details from the third level of abstraction, also hints at the motivation of the customer at acquisition time.

The cohort columns (YYYY, YYQQ, YYMM) are technically redundant since they can be calculated from the initial subscription or order dates. We include them as 'convenience' columns that make data extraction easier for everyone and, importantly, standardise the representation of cohorts across all reports and models.

Level 2 – orders

The order level, shown in Table 3, is fairly consistent between the two uses with exception of column names.

Table 3 – Level 2: Subscription chain vs. order table contents

Subscription chain summary	Order summary	Notes
Subscriber ID, chain sequence #	Customer ID, order sequence #	The primary key pair
Subscription system ID	Order system ID	Reference back to operational system
Is chain active	Is most recent order	Status flags
This, prior, & next chain	This, prior, & next order	Primary products for ….
Chain start & end	Order placed & complete	Timestamps
Days to prior & next stint	Days to prior & next order	Intervals (days w/fraction)
Number & dollars	Order value, tax, shipping, …	Payments
Dollars & reason code	Dollars & reason code	Discounts w/reason
Status, promo, offer, …	Status, promo, offer, …	Initial conditions
Requested on, stated reason, is voluntary flag	Requested on, amount, reason	Cancels & returns
Level, type, breadth, acceleration over chain	Top product group, % dominance	Consumption
Device, geo location	Billing & shipment geo	Location
Chat or help	Order satisfaction survey	Follow-up
???	???	Additional organisation specific customer level metrics

Notice that the second part of the primary key is a sequence number within the individual. This trick makes it easy to 'walk' the chains or orders from the first one to the last or in reverse order. For convenience, the primary products for the previous and next chain or order are included; as are the intervals from the previous, and to the next, chain or order. The idea of 'primary product' type is for a chain or order with a variety of product types, with the primary product is the most important type. This attempts to get at the motivating product for the subscriber or customer: 'Importance' is measured by the amount of consumption time spent for subscribers or the product monetary value for customers.

Level 3 – consumption

The differences between subscription and product businesses are most marked at the consumption level, Table 4. The specifics of a given business will dictate the exact content included at this level.

For a subscription business the primary key element for subscriptions is the usage stint sequence number. Sometimes it is useful to know for a given subscription chain the associated usage stints, which the *Chain Sequence #* reveals. Stints have a start and end timestamp, a count of the number of usage events in the stint and a context including a geo location and, perhaps, a platform. Also recorded is the interval after the end of the prior stint to the start of this one and the interval after the end of this stint to the beginning of the next one: these intervals are useful for finding patterns in usage frequency. The usage profile should summarise the nature of what was consumed during the stint to get at the primary focus of the subscriber and reason for the stint.

For a product organisation the content of the third level starts with the order detail line: the SKU, quantity and price with any

Table 4 – Level 3: Usage stint vs. order line detail table contents

Usage stint summary	**Stint notes**	**Order line detail**	**Line notes**
Subscriber ID, usage stint sequence #	The primary key pair	Customer ID, order sequence #, order line #	Primary key triplet
Chain sequence #	Foreign key back to chain		
Usage system initial event ID	Reference back to operational system	Order system line ID	*Same as stint*
Stint start & end	Time stamp		
Hours to prior & next	Interval (hours w/fraction)		
What used	Usage profile:	SKU, description	What ordered
Count	How many?	Quantity	How many?
Where	Geo, platform, …	Offer, discount	SKU specific
		Kind, group, class, …	Taxonomy membership
		Gross & net price	Of order line
		Prices of components & options	To roll-up across orders.
		Size, colour, …	Standard options
???	As needed to characterise usage	???	As needed to describe the order line

Note: SKU: Stock Keeping Unit (the product identifier).

order line level discounts. If the product organisation has complex offerings it will also have a strong merchandising team: to support them the order line needs to have a complete breakdown of the characteristics of what was ordered including things like size, colour, options and any other features helpful to the merchants.

It may also be useful to break down the base price into its components for options and add-ons. This level can get quite complex and may be challenging to build from the source data but the merchandising team benefits greatly by slicing profitability reports by the various product attributes.

AN EXAMPLE CI DATA MART

Figure 2 shows the high-level data structure diagram of an actual CI data mart implemented by the author for a subscription business.

In addition to the three main tables corresponding to the three levels of abstraction (shown in red), there are two helper tables. The Renewals table gives

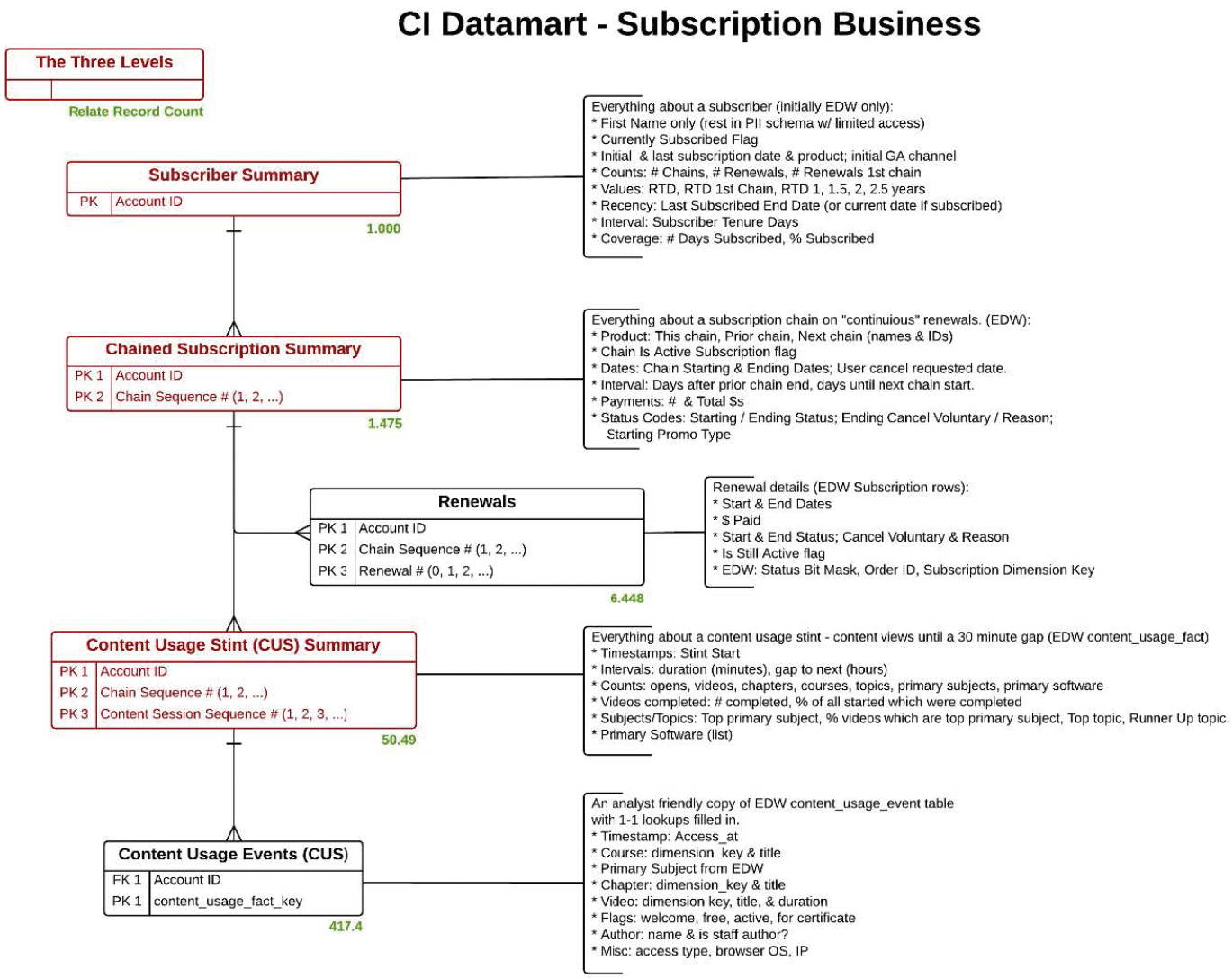


Figure 2: High level data structure diagram for a subscription business

```
Words
      Count the number of subscriber summary rows with the
      is_subscribed flag being TRUE.

SQL
      SELECT COUNT(*) AS Number_Subscribed
        FROM subscriber_summary AS ss
       WHERE ss.is_subscribed;
```

Figure 3: Solution to business question 1

access to the individual renewal events making up a subscription chain which allows the finance and retention marketing teams to look at the fine structure of renewals.

The Content Usage Events table is rolled up to the Content Usage Stint table. It is an analyst friendly version of the star schema[11] fact and dimension tables on the enterprise data warehouse of the organisation. This table allowed the content team to easily examine usage patterns across all subscribers or types of subscribers.

The green numbers at the right under each table show the relative number of rows per subscriber row. On the average, there are 1.5 subscription chains, 50 content usage stints, and 417 usage events per subscriber. The CI mart design not only simplifies analysis but also provides a high degree of data summarisation.

THE STRUCTURE IN ACTION

The following four examples are based on actual business questions and queries from the subscription based company at which the author was consulting. The queries are both shown in words (pseudo-code) and SQL, the former a guide for setting up front end tools like Tableau. SQL would be used by more advanced business analysts and data scientists.

Note we are using the AWS Redshift dialect of standard SQL.

Business question 1: how many subscribers do we have?

This is one of the first questions asked by anyone in the organisation, from the chairman of the board on down.

It is about subscribers so we should be able to use the top level of abstraction, Subscriber Summary, to get our answer. The query is trivial: (Figure 3).

This query is trivial only as the business rules defining what it means to be subscribed are encapsulated in the is_subscribed column. The rules themselves are typically complex. For example, there are decisions to be made about credit card processing failures, payment grace periods and special customer service considerations.

This is an example of the dangers in the 'data lake' approach that contains only raw data. Imagine the number of different interpretations that 'subscribed' could have when each analyst and data scientist is left with creating their own interpretation of the business rules. The curated columns of the CI mart ensure uniform business logic is used throughout the business.

Business question 2: by monthly cohort how many and what percentage of paying subscribers are currently subscribed?

Senior management uses this to discover inflection points in subscriber retention; these may be due to changes in business approach, acquisition methods or economic conditions.

```
Words
      Group the subsubscriber_summary rows having positive RTD by
      monthly cohort. Calculate:

        1. Number of subscriber starts as the number of rows,
        2. Number still subscribed as the number with is_subscribed
           being TRUE, and
        3. Percentage that are still subscribed from 1) and 2).

      Arrange in monthly cohort order.

SQL
      SELECT ss.cohort_yymm,
             COUNT(*) AS Number_Starts,
             SUM(ss.is_subscribed::INT) AS Number_Subscribed,
             (100.0 * SUM(ss.is_subscribed::INT) /
               COUNT(*))::NUMERIC(10, 1) AS Per cent_Subscribed
        FROM subscriber_summary AS
       WHERE ss.revenue_to_date > 0
       GROUP BY 1
       ORDER BY 1;
```

Figure 4: Solution to business question 2

Again, this is a subscriber question so the Subscriber Summary table should be sufficient. 'Paying subscribers' have a RTD (revenue to date) greater than zero. The convenience column Cohort YYMM makes it easy to group by monthly cohorts (Figure 4).

This query is more interesting than the first one. Without the cohort convenience column, cohort_yymm, as well as the is_subscribed Boolean flag, this query would be much more complicated and difficult to understand.

Business question 3: by cancel reason what per cent of first chain cancels re-subscribe? What is the average lag between end of the first subscription and start of the second? Only look at subscribers starting on or after 1/1/2014

This question was asked by the reactivation team in the marketing group. One key performance indicator (KPI) for the team is the reactivation rate of subscribers who cancel voluntarily.

As this is a question about subscriptions it can be answered at the second level of abstraction, Subscription Chain Summary. 'First chain cancels' are the first chains (chain_sequence_number = 1) with the Is Active flag = FALSE. The subscriber has a second chain if the Days Until Next Start field is present (NOT NULL), (Figure 5).

Admittedly this query may be beyond the novice SQL coder. The 'SQL coach' can use it a teaching moment. In the experience of this authorience the more quantitate business analysts can understand the logic and code queries of similar complexity after a few weeks.

Business question 4: what are the top five content areas viewed over the last 90 days?

The content team tracks this to discover new trends in subscriber interest which helps guide their content creation efforts.

This is a consumption question so it is so it can be answered at the third level with the table Content Usage Stints. The column Top Area is set to one of a few dozen content areas. The function GETDATE() returns the current date (Figure 6).

This query is quite straightforward: most business analysts with basic SQL skills can code this without help.

Words

Group the initial subscriber_summary chains for each subscriber which have a start date from 2014 onwards and are no longer active by the cancel reasons given by the subscribers. Calculate:

1. Number of first chain cancels as the number of rows,
2. Number of re-subscribers as the number of rows with non-NULL days_until_next_start field,
3. Percentage of re-subscribers from 1) and 2), and
4. Average days before re-subscribing as the average of days_until_next_start.

Arrange results in decreasing number of first chain cancels.

SQL

```
SELECT css.ending_cancel_reason,
       COUNT(*) AS Number_1st_Chain_Cancels,
       COUNT(css.days_until_next_start) AS Number_Resubs,
       (100.0 * COUNT(css.days_until_next_start) /
         COUNT(*))::NUMERIC(10, 1) AS Per cent_Resub,
       AVG(css.days_until_next_start) AS Avg_Days_Before_Resub
  FROM chained_subscription_summary AS css
 WHERE css.chain_sequence_number = 1
   AND NOT css.is_active
   AND css.starting_date >= '2014-01-01'
 GROUP BY 1
 ORDER BY 2 DESC;
```

Figure 5: Solution to business question 3

Words

Group the content usage stints in the last 90 days by the top content area. Calculate:

1. Number of stints as the number of records.

Arrange by the number of stints in decreasing order.
Show only the first five records.

SQL

```
SELECT cuss.top_area AS Top_Area_In_Stint,
       COUNT(*) AS Number_Stints
  FROM content_usage_stint_summary AS cuss
 WHERE cuss.stint_start_at >= DATEADD(day, -90, GETDATE())
 GROUP BY 1
 ORDER BY 2 DESC
 LIMIT 5;
```

Figure 6: Solution to business question 4

IMPLEMENTATION CONSIDERATIONS AND EXPERIENCE

While the purpose of this paper is to put forward a data structure design philosophy some implementation notes are called for.

The data presented to the CI mart users should be three very wide tables which could be implemented as database views on top of a classical star-schema data warehouse. The author prefers to 'materialise' the tables

in a columnar data base system[12] such as Vertica, Aster, or AWS Redshift. Columnar databases are optimised for wide tables with many categorical columns which compress repeated data and do very fast queries against column based criteria. The Redshift SQL dialect is based on PostgreSQL which is known for its analytic capabilities. As an added benefit the event rollups and sessionisation steps needed to build the summary tables can be run on Redshift as part of the update process.

Most of front end tools today have AWS Redshift connectivity built in. Amazon supplies Redshift optimised JDBC drivers for most other cases while PostgreSQL ODBC is the fall back for connectivity.

The astute reader will have noticed many of the columns will need to be updated with new order or consumption events: the implementation trick is to just process those customers with new events since the last update. The update frequency can be increased to achieve near real-time reporting if needed.

Although data governance has not been discussed the CI data mart requires serious governance to be accurate and believable. Adequate resources must be built into any CI data mart planning to ensure proper governance and sort out any issues discovered along the way: see Earley and Maislin[13] for an excellent guide.

No business is static and the CI data mart must evolve as business rules and requirements change. The simple three-table structure implemented on a columnar database engine makes it easy to add columns as needed. Columns encapsulating business rules (such as *is_subscribed* in the first example), should be coded in distinct modules. All changes must be coordinated with the data governance team.

The product CI mart

Implementation at the product company took two months for the customer and order levels and an additional three months to get all the product options and costs into the order detail level[14]. This included two-way integration with the new email system which was being brought up at the same time. Updates were done nightly.

Finance, marketing, operations and merchandising groups worked with the business intelligence team to develop targeted reports for tactical and strategic data driven decisions. The two-way integration with the email system gave the email marketing team up-to-date customer triggers, scores and segments enabling highly customised messaging.

The mart was also the source of CI data for propensity models, geo-targeting models and customer segmentation built by the data scientist.

The subscription CI mart

The primary driver for the CI mart was a data science effort to assist retention marketing with customer segments and propensity to churn models. This mart was easier since there was a rudimentary star-schema data warehouse for most of the source data. All levels were completed in three months.

Once completed the mart was also used to deliver insights to the acquisition marketing, content development, product and management teams. Some of this work was done by the data science team but many *ad hoc* queries were done by the business teams themselves.

CONCLUSION

The author contends that the benefits to a customer focused organisation far outweigh the costs of the CI data mart. The benefit of having a curated single version of customer 'truth', usable by business analysts and data scientists alike, cannot be minimised. Then there is the

accrued time savings when working from the CI data mart rather than needing to stitch together disparate data sets or wait for help from the overworked business intelligence or data engineering teams. Finally, senior management will be able to get timely and accurate answers to their *ad hoc* customer insight questions with minimal disruption to the front-line analysts and data engineers.

References

1. The Data Lake Debate: Pro is Up First, available at: http://www.smartdatacollective.com/tamaradull/306046/data-lake-debate-pro-s-first (accessed on 17th February, 2017).
2. Kimball, R. and Ross, M. (2010) The Kimball Group Reader; Relentlessly Practical Tools for Data Warehousing and Business Intelligence, Sections 3. 1–2. Wiley, Indianapolis.
3. Acxiom - Wikipedia, available at: https://en.wikipedia.org/wiki/Acxiom (accessed on 17th February, 2017).
4. Information broker - Wikipedia, available at: https://en.wikipedia.org/wiki/Information_broker (accessed on 17th February, 2017).
5. American Community Survey (ACS), available at: https://www.census.gov/programs-surveys/acs/ (accessed on 17th February, 2017).
6. Claritas Prizm - Wikipedia, available at: https://en.wikipedia.org/wiki/Claritas_Prizm (accessed on 17th February, 2017).
7. Geolocation software - Wikipedia, available at: https://en.wikipedia.org/wiki/Geolocation_software (accessed on 17th February, 2017).
8. Kimball & Ross. *Ibid*.
9. While it is tempting to use a front-end tool's data integration features for expediency, doing so is a dangerous trap locking the organization into a specific vendor's technology and forcing it's use by anyone wishing to leverage the integrated data. Even worse, the integrated data may not be easily exported.
10. Libey, D. and Pickering, C. (2005) RFM and Beyond. MerritDirect Press, White Plains.
11. Star schema – Wikipedia, available at: https://en.wikipedia.org/wiki/Star_schema (accessed on 17th February, 2017).
12. Column-oriented DBMS - Wikipedia, available at: https://en.wikipedia.org/wiki/Column-oriented_DBMS (accessed on 17th February, 2017).
13. Earley, S. and Maislin, S. (2016) 'Data governance and digital transformation: Using the customer journey to define a framework', *Applied Marketing Analytics,* Vol. 2, No. 1, pp. 25–40.
14. The final roll-up query of the source system MySQL tables had 14 joins.