snowflake

AGILE DATA ENGINEERING

Introduction to Data Vault 2.0

KENT GRAZIANO, CHIEF TECHNICAL EVANGELIST

KentGraziano

Agenda

- Bio
- Agile & DW
- What is a Data Vault & Where does it fit?
- How to design a Data Vault model
- Foundational Keys
- Benefits of Data Vault
- Who is using Data Vault?
- References

My Bio

- ***Chief Technical Evangelist***, Snowflake Computing
- Blogger: *The Data Warrior*
- Certified Data Vault Master and DV 2.0 Practitioner
- Oracle ACE Director (Alumni)
- OakTable Member
- Member – DADA Houston & DAMA International
- Data Modeling, Data Architecture and Data Warehouse Specialist
  - 30+ years in IT
  - 25+ years of Oracle-related work
  - 20+ years of data warehousing experience
- Former-Member: Boulder BI Brain Trust (http://www.boulderbibraintrust.org/)
- Author & Co-Author of a bunch of books
- Past-President of ODTUG and Rocky Mountain Oracle User Group

3 years in stealth + 3 years GA

Founded 2012 by industry veterans with over 120 database patents

Over $850M in venture funding from leading investors

First customers 2014, general availability 2015

800+ employees
Over 2000 customers today

Fun facts:

Queries processed in Snowflake per day:	Largest single table:	Largest number of tables single DB:	Single customer most data:	Single customer most users:
100 million	**68 trillion rows**	**200,000**	**> 40PB**	**> 10,000**

Manifesto for Agile Software Development

http://agilemanifesto.org

"We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

★ Individuals and interactions over processes and tools
★ Working software over comprehensive documentation
★ Customer collaboration over contract negotiation
★ Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more."

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Applying Agile to DW

- User Stories instead of requirements documents
- Time-based iterations
  - Iteration has a standard length
  - Choose one or more user stories to fit in that iteration
- Rework is part of the game
  - There are no “Missed requirements”…only those that haven’t been discovered yet

Data Vault Model Definition

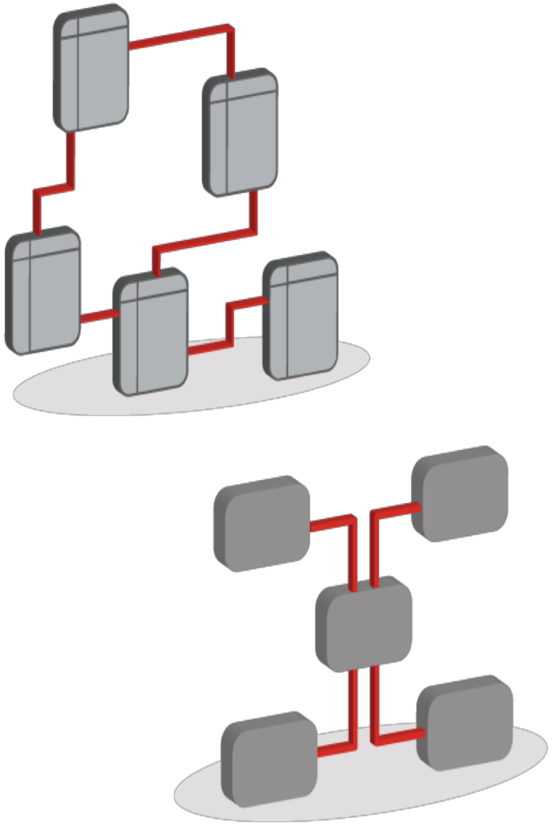
The Data Vault is a detail oriented, historical tracking and uniquely linked set of normalized tables that support one or more functional areas of business.

It is a hybrid approach encompassing the best of breed between 3rd normal form (3NF) and star schema. The design is flexible, scalable, consistent and adaptable to the needs of the enterprise.

Architected specifically to meet the needs of today's enterprise data warehouses

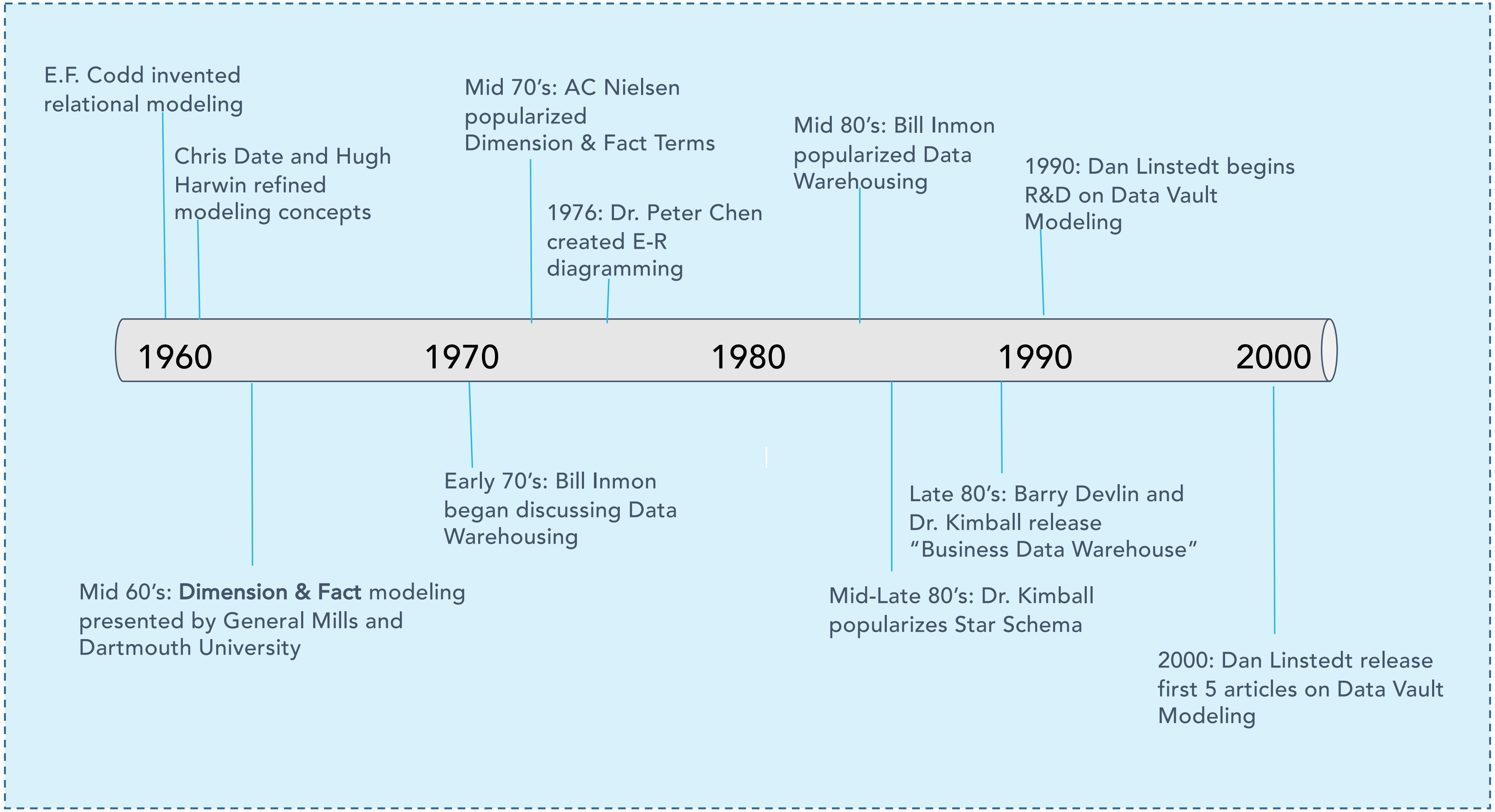
Dan Linstedt: Defining the Data Vault

What is Data Vault Trying to Solve?



- What are our other Enterprise Data Warehouse Options?
  - Third-Normal Form (3NF): Complex primary keys (PK's) with cascading snapshot
  - Star Schema (Dimensional): Difficult to reengineer fact tables for granularity changes
- Difficult to get it right the first time
- Not adaptable to rapid business change
- NOT AGILE!

Data Vault Timeline



E.F. Codd invented relational modeling

Chris Date and Hugh Harwin refined modeling concepts

Mid 60's: **Dimension & Fact** modeling presented by General Mills and Dartmouth University

Early 70's: Bill Inmon began discussing Data Warehousing

Mid 70's: AC Nielsen popularized Dimension & Fact Terms

1976: Dr. Peter Chen created E-R diagramming

Mid 80's: Bill Inmon popularized Data Warehousing

Mid-Late 80's: Dr. Kimball popularizes Star Schema

Late 80's: Barry Devlin and Dr. Kimball release "Business Data Warehouse"

1990: Dan Linstedt begins R&D on Data Vault Modeling

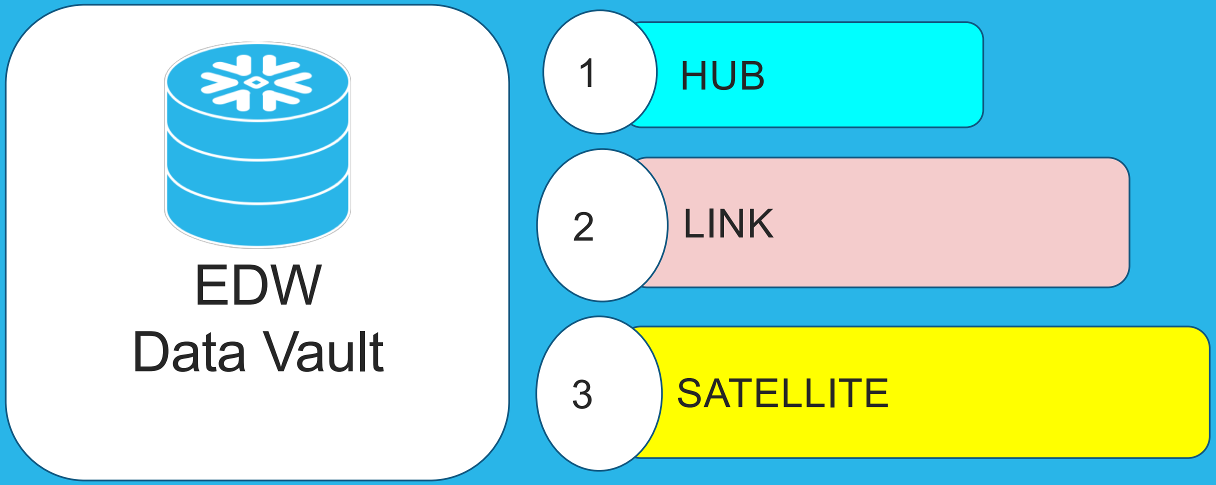
2000: Dan Linstedt release first 5 articles on Data Vault Modeling

Data Vault Evolution

- The work on the Data Vault approach began in the early 1990s; completed around 1999.
- Throughout 1999, 2000, and 2001, the Data Vault design was tested, refined, and deployed into specific customer sites.
- In 2002, the industry thought leaders were asked to review the architecture.
  - **Kent meets Dan at a Lunch & Learn in Denver!**
- In 2003, Dan began teaching the modeling techniques to the mass public.
  - **Kent & Team take their 1st Data Vault Modeling class**
- In 2014, Dan introduced DV 2.0!

WHAT DOES THE DATA VAULT MODEL LOOK LIKE?

Data Vault: 3 Simple Structures



Data Vault Core Architecture

HUBS

Unique List of Business Keys

LINKS

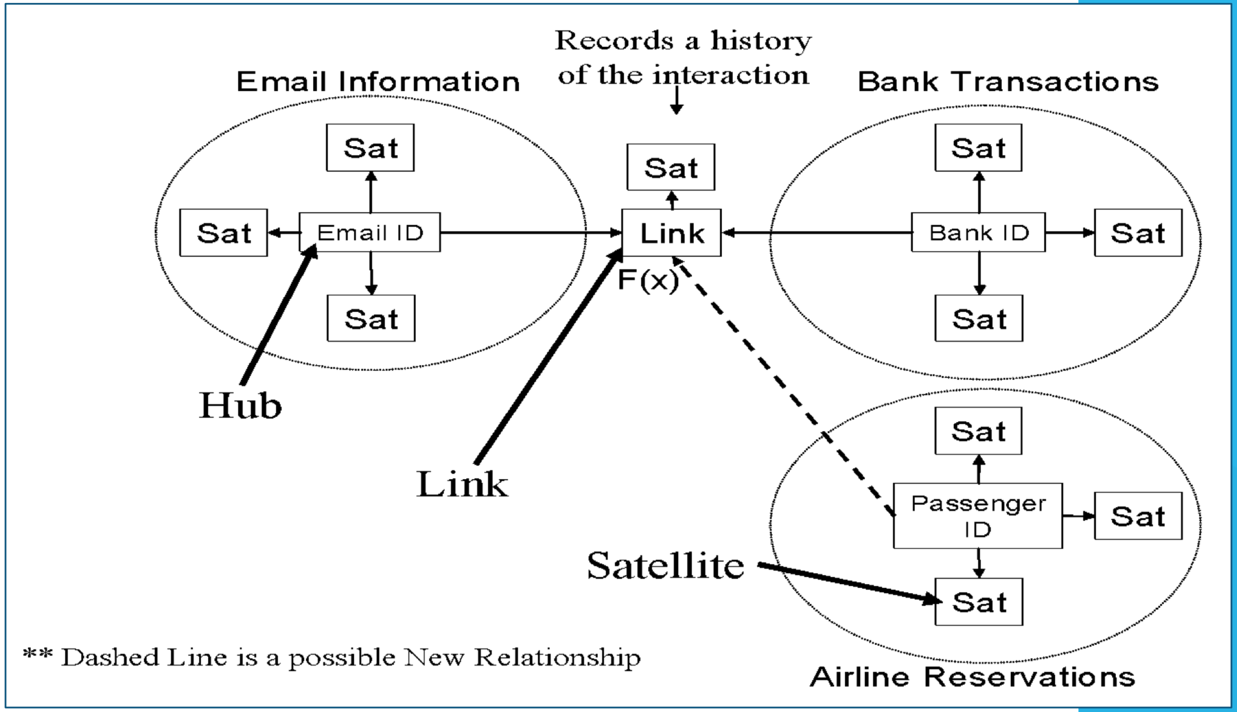
Unique List of Relationships across Keys

SATS

Descriptive Data

- Satellites have one and only one parent table
- Satellites cannot be parents to other tables
- Hubs cannot be child tables

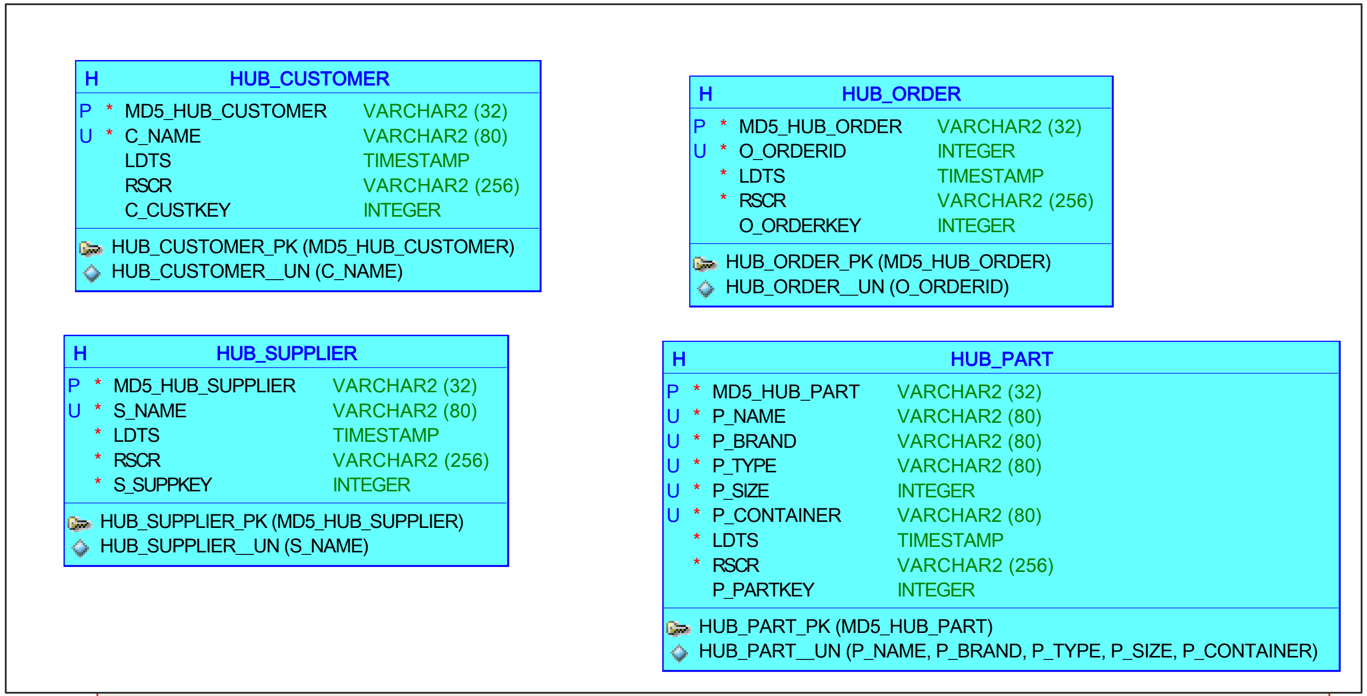
Standard Data Vault Model



** Dashed Line is a possible New Relationship

- *Hub*: List of UNIQUE business keys.
- *Link*: List of UNIQUE relationships across keys
- *Satellite*: Historical descriptive data.

1. Hub = Business Keys



H		HUB_CUSTOMER	
P	*	MD5_HUB_CUSTOMER	VARCHAR2 (32)
U	*	C_NAME	VARCHAR2 (80)
		LDTS	TIMESTAMP
		RSCR	VARCHAR2 (256)
		C_CUSTKEY	INTEGER

HUB_CUSTOMER_PK (MD5_HUB_CUSTOMER)
HUB_CUSTOMER__UN (C_NAME)

H		HUB_ORDER	
P	*	MD5_HUB_ORDER	VARCHAR2 (32)
U	*	O_ORDERID	INTEGER
	*	LDTS	TIMESTAMP
	*	RSCR	VARCHAR2 (256)
		O_ORDERKEY	INTEGER

HUB_ORDER_PK (MD5_HUB_ORDER)
HUB_ORDER__UN (O_ORDERID)

H		HUB_SUPPLIER	
P	*	MD5_HUB_SUPPLIER	VARCHAR2 (32)
U	*	S_NAME	VARCHAR2 (80)
	*	LDTS	TIMESTAMP
	*	RSCR	VARCHAR2 (256)
	*	S_SUPPKEY	INTEGER

HUB_SUPPLIER_PK (MD5_HUB_SUPPLIER)
HUB_SUPPLIER__UN (S_NAME)

H		HUB_PART	
P	*	MD5_HUB_PART	VARCHAR2 (32)
U	*	P_NAME	VARCHAR2 (80)
U	*	P_BRAND	VARCHAR2 (80)
U	*	P_TYPE	VARCHAR2 (80)
U	*	P_SIZE	INTEGER
U	*	P_CONTAINER	VARCHAR2 (80)
	*	LDTS	TIMESTAMP
	*	RSCR	VARCHAR2 (256)
		P_PARTKEY	INTEGER

HUB_PART_PK (MD5_HUB_PART)
HUB_PART__UN (P_NAME, P_BRAND, P_TYPE, P_SIZE, P_CONTAINER)

Hubs = Unique Lists of Business Keys
Business Keys are used to TRACK and IDENTIFY key information
DV 2.0 uses MD5 Hash of the BK for the PK
Natural Business Keys also acceptable for PK

What Does the MD5 Look Like?

- MD5 hash function – **Snowflake**
  - MD5 (UPPER(RTRIM(RMC.CAFCUSCHN)))
- MD5 hash function – **Oracle**
  - rawtohex(sys.utl_raw.cast_to_raw(dbms_obfuscation_toolkit.md5 (input_string => ...)
  - NEW: **dbms_crypto.HASH**(utl_raw.cast_to_raw(<input string>), 2);
    - 2 is for MD5 algorithm option
- MD5 hash function - **SQL Server**
  - CONVERT([Char](32),HASHBYTES('MD5', UPPER(RTRIM(RMC.CAFCUSCHN))))
- **Need to minimize chance of duplicates**
  - 12||3||45 and 1||2||345 hash to same value
  - Need a separator between each
  - Example: Col1||'^'||Col2||'^'||Col3
  - Need to account for NULLs too

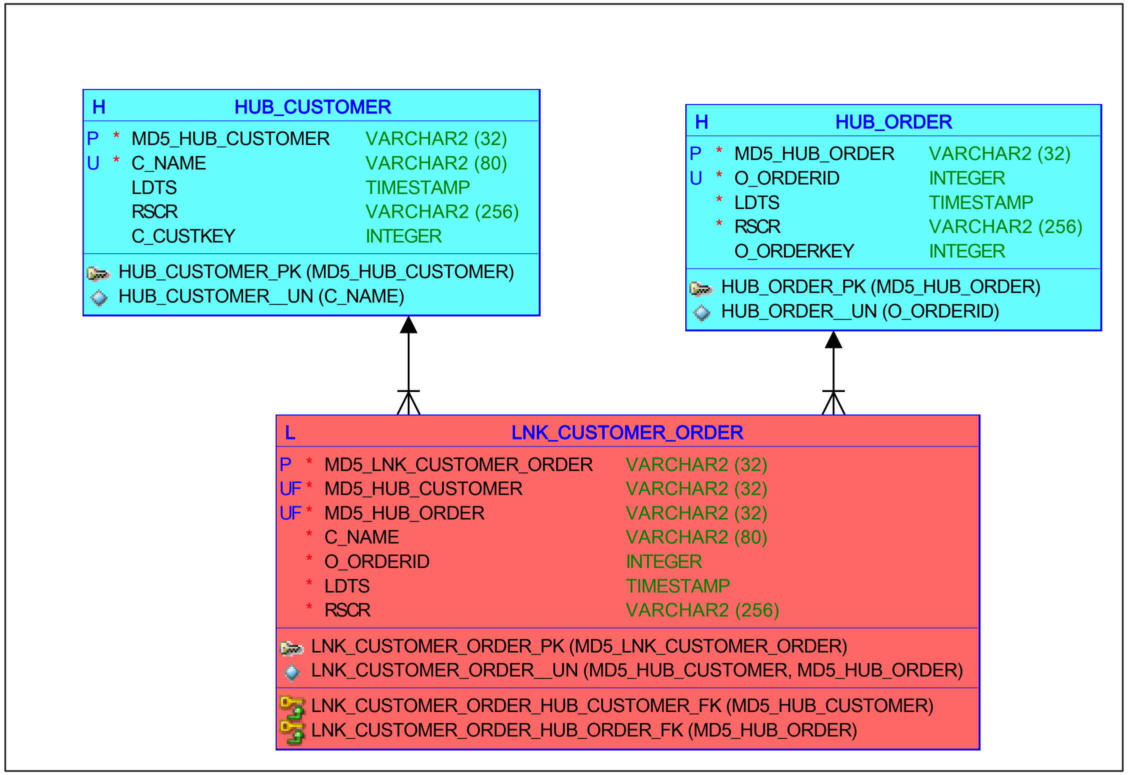
Other Considerations

- To generate most consistent string: standardize!
- Convert data types
- If 'NUMBER', 'VARCHAR', ‘TEXT’
  - THEN 'TO_CHAR(' || column_name || ‘)‘
- If LIKE 'DATE%‘
  - THEN 'TO_CHAR(' || column_name || ', "YYYY-MM-DD’’)‘
- If LIKE 'TIME%‘
  - THEN 'TO_CHAR(' || column_name || ', "YYYY-MM-DD HH24:MI:SS")'

Final Input String

```
(
    UPPER(TRIM(T1.GENERICNAME)) ||'^'||
    UPPER(TRIM(TO_CHAR(T1.MED_STRNG_AMT))) ||'^'||
    UPPER(TRIM(T1.UOM_CD)) ||'^'||
    UPPER(TRIM(T1.MED_FORM_NM)) ||'^'
)
```

2: Links = Associations



H		HUB_CUSTOMER	
P	*	MD5_HUB_CUSTOMER	VARCHAR2 (32)
U	*	C_NAME	VARCHAR2 (80)
		LDTS	TIMESTAMP
		RSCR	VARCHAR2 (256)
		C_CUSTKEY	INTEGER

HUB_CUSTOMER_PK (MD5_HUB_CUSTOMER)
HUB_CUSTOMER__UN (C_NAME)

H		HUB_ORDER	
P	*	MD5_HUB_ORDER	VARCHAR2 (32)
U	*	O_ORDERID	INTEGER
	*	LDTS	TIMESTAMP
	*	RSCR	VARCHAR2 (256)
		O_ORDERKEY	INTEGER

HUB_ORDER_PK (MD5_HUB_ORDER)
HUB_ORDER__UN (O_ORDERID)

L		LNK_CUSTOMER_ORDER	
P	*	MD5_LNK_CUSTOMER_ORDER	VARCHAR2 (32)
UF	*	MD5_HUB_CUSTOMER	VARCHAR2 (32)
UF	*	MD5_HUB_ORDER	VARCHAR2 (32)
	*	C_NAME	VARCHAR2 (80)
	*	O_ORDERID	INTEGER
	*	LDTS	TIMESTAMP
	*	RSCR	VARCHAR2 (256)

LNK_CUSTOMER_ORDER_PK (MD5_LNK_CUSTOMER_ORDER)
LNK_CUSTOMER_ORDER__UN (MD5_HUB_CUSTOMER, MD5_HUB_ORDER)

LNK_CUSTOMER_ORDER_HUB_CUSTOMER_FK (MD5_HUB_CUSTOMER)
LNK_CUSTOMER_ORDER_HUB_ORDER_FK (MD5_HUB_ORDER)

Links = Transactions and Associations
They are used to hook together multiple sets of information
In DV 2.0 the BK attributes may migrate to the Links for faster query

Modeling Links - 1:1 or 1:M?

Today

Relationship is a 1:1
so
why model a Link?

Tomorrow

The business rule can change to a 1:M

You discover new data later

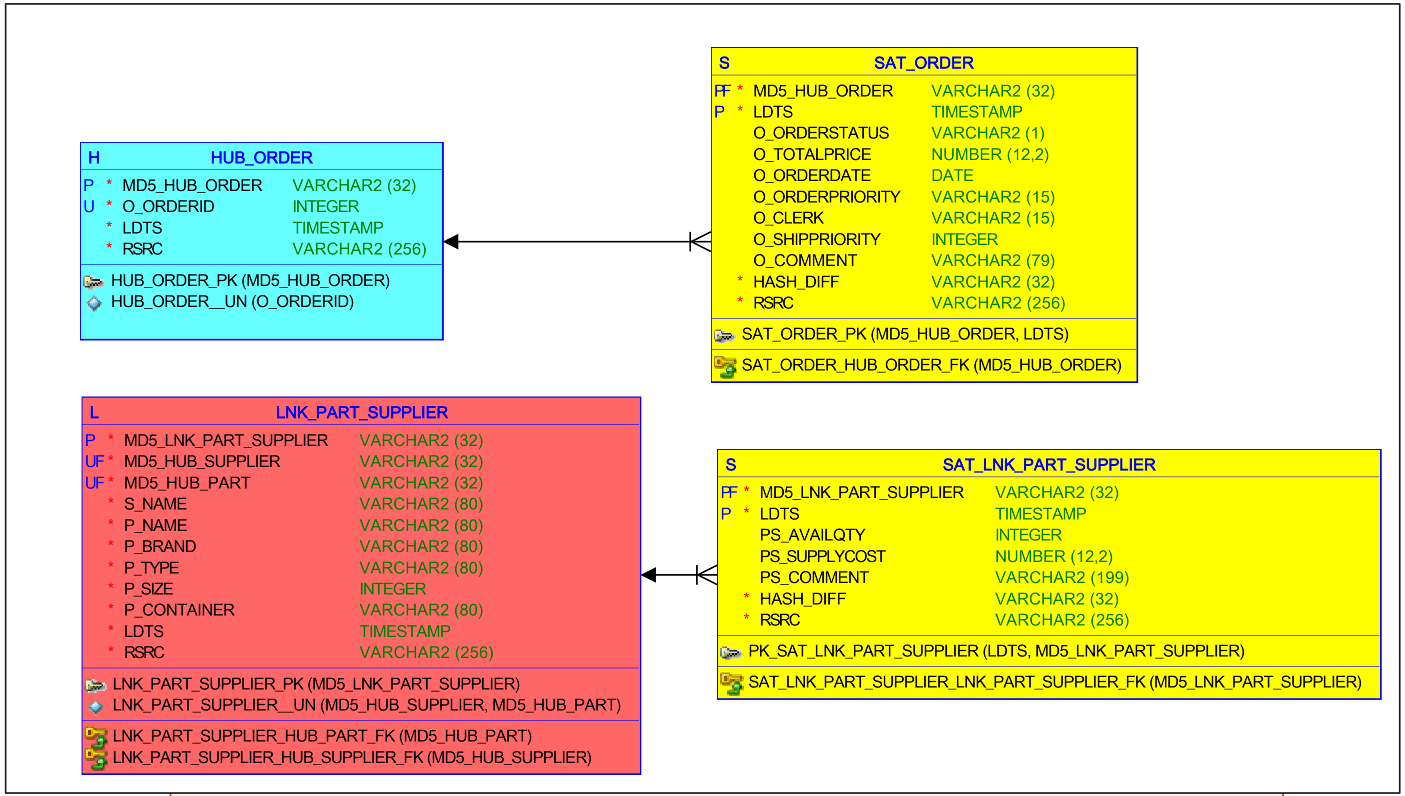
With DV

No need to change EDW structure

Existing data is fine

New data is added

3. Satellites = Descriptors



H HUB_ORDER

P	*	MD5_HUB_ORDER	VARCHAR2 (32)
U	*	O_ORDERID	INTEGER
	*	LDTS	TIMESTAMP
	*	RSRC	VARCHAR2 (256)

HUB_ORDER_PK (MD5_HUB_ORDER)
HUB_ORDER__UN (O_ORDERID)

S SAT_ORDER

PF	*	MD5_HUB_ORDER	VARCHAR2 (32)
P	*	LDTS	TIMESTAMP
		O_ORDERSTATUS	VARCHAR2 (1)
		O_TOTALPRICE	NUMBER (12,2)
		O_ORDERDATE	DATE
		O_ORDERPRIORITY	VARCHAR2 (15)
		O_CLERK	VARCHAR2 (15)
		O_SHIPPRIORITY	INTEGER
		O_COMMENT	VARCHAR2 (79)
	*	HASH_DIFF	VARCHAR2 (32)
	*	RSRC	VARCHAR2 (256)

SAT_ORDER_PK (MD5_HUB_ORDER, LDTS)
SAT_ORDER_HUB_ORDER_FK (MD5_HUB_ORDER)

L LNK_PART_SUPPLIER

P	*	MD5_LNK_PART_SUPPLIER	VARCHAR2 (32)
UF	*	MD5_HUB_SUPPLIER	VARCHAR2 (32)
UF	*	MD5_HUB_PART	VARCHAR2 (32)
	*	S_NAME	VARCHAR2 (80)
	*	P_NAME	VARCHAR2 (80)
	*	P_BRAND	VARCHAR2 (80)
	*	P_TYPE	VARCHAR2 (80)
	*	P_SIZE	INTEGER
	*	P_CONTAINER	VARCHAR2 (80)
	*	LDTS	TIMESTAMP
	*	RSRC	VARCHAR2 (256)

LNK_PART_SUPPLIER_PK (MD5_LNK_PART_SUPPLIER)
LNK_PART_SUPPLIER__UN (MD5_HUB_SUPPLIER, MD5_HUB_PART)
LNK_PART_SUPPLIER_HUB_PART_FK (MD5_HUB_PART)
LNK_PART_SUPPLIER_HUB_SUPPLIER_FK (MD5_HUB_SUPPLIER)

S SAT_LNK_PART_SUPPLIER

PF	*	MD5_LNK_PART_SUPPLIER	VARCHAR2 (32)
P	*	LDTS	TIMESTAMP
		PS_AVAILQTY	INTEGER
		PS_SUPPLYCOST	NUMBER (12,2)
		PS_COMMENT	VARCHAR2 (199)
	*	HASH_DIFF	VARCHAR2 (32)
	*	RSRC	VARCHAR2 (256)

PK_SAT_LNK_PART_SUPPLIER (LDTS, MD5_LNK_PART_SUPPLIER)
SAT_LNK_PART_SUPPLIER_LNK_PART_SUPPLIER_FK (MD5_LNK_PART_SUPPLIER)

Satellites provide context for the Hubs and the Links
Tracks changes over time - Like SCD 2
In DV 2.0 use HASH_DIFF to detect changes

MD5-Based Change Detection

- **Think Type 2 SCD (Slowly Changing Dimensions)**
- **Old Way:**
  - Compare column by column
    - Source value != Current value in DW table
  - 20 columns, then 20 compares
- **New Way:**
  - Concatenate all columns to one string
  - Convert to one char(32) string with hash function
  - Compare to hashed value (HASH_DIFF) in target table
  - Does not matter how many columns

Easily Getting Current Rows

- Use the Hub/Link PK columns
- Filter on LEAD of LOAD_DTS

Example – Virtual Expire Date!

```
CASE
  WHEN LEAD(stg.LOAD_DTS)
        OVER (PARTITION BY stg.CDC_KEY
        ORDER BY stg.LOAD_DTS) IS NULL
     THEN 'Y'
   ELSE 'N'
END CURR_FLG,

LEAD(stg.LOAD_DTS) OVER (PARTITION BY
stg.CDC_KEY ORDER BY stg.LOAD_DTS) EXPR_DTS
```

Foundational Keys

Data Vault Model Flexibility (Agility)

Goes beyond standard 3NF

Highly normalized

- Hubs and Links only hold keys and meta data
- Satellites split by rate of change and/or source

Enables Agile data modeling

- Easy to add to model without having to change existing structures and load routines
  - Relationships (links) can be dropped and created on-demand.
- No more reloading history because of a missed requirement

Based on natural business keys

Not system surrogate keys

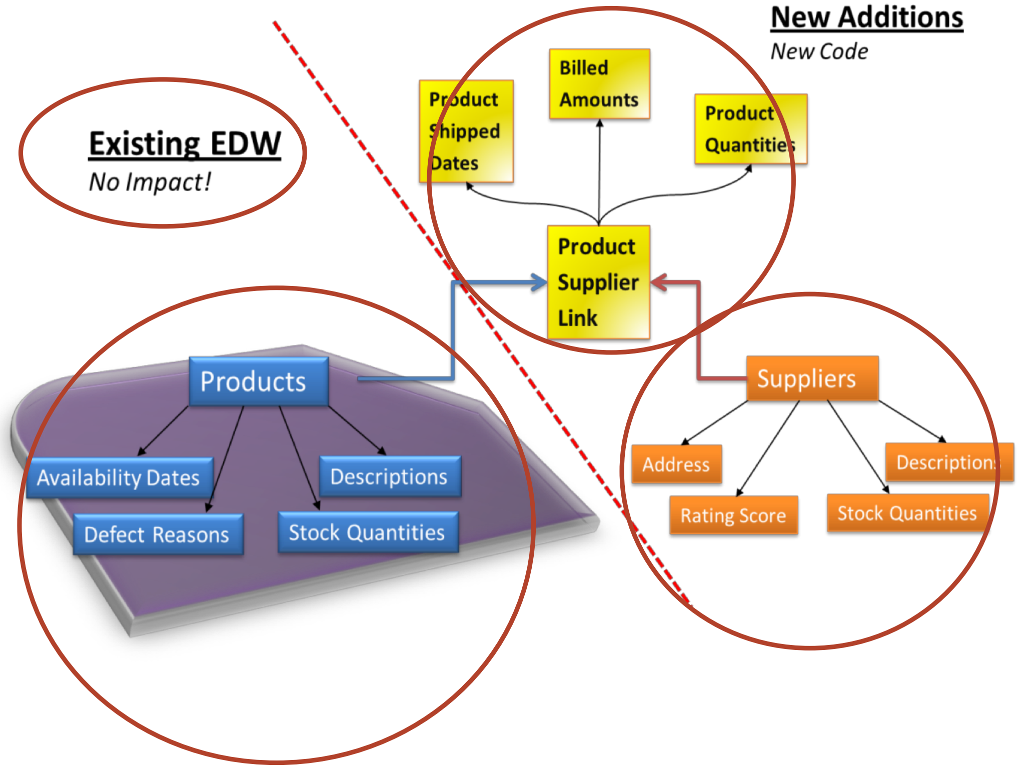
Allows for integrating data across functions and source systems more easily

- All data relationships are key driven

Data Vault Agility

Adding new components to the EDW has NEAR ZERO impact to existing:

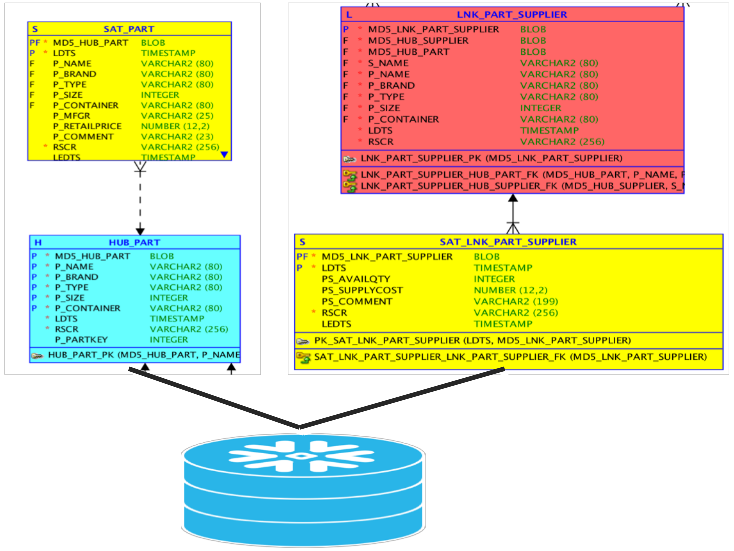
- Loading Processes
- Data Model
- Reports & BI Functions
- Downstream Systems
- Star Schemas or Data Marts



Perhaps You Wish to Split for Security Reasons?

From This

DV “Logical” Partitioning



S SAT_PART

PF	*	MD5_HUB_PART	BLOB
P	*	LDTS	TIMESTAMP
F		P_NAME	VARCHAR2 (80)
F		P_BRAND	VARCHAR2 (80)
F		P_TYPE	VARCHAR2 (80)
F		P_SIZE	INTEGER
F		P_CONTAINER	VARCHAR2 (80)
		P_MFGR	VARCHAR2 (25)
		P_RETAILPRICE	NUMBER (12,2)
		P_COMMENT	VARCHAR2 (23)
	*	RSCR	VARCHAR2 (256)
		LEDTS	TIMESTAMP

H HUB_PART

P	*	MD5_HUB_PART	BLOB
P	*	P_NAME	VARCHAR2 (80)
P	*	P_BRAND	VARCHAR2 (80)
P	*	P_TYPE	VARCHAR2 (80)
P	*	P_SIZE	INTEGER
P	*	P_CONTAINER	VARCHAR2 (80)
	*	LDTS	TIMESTAMP
	*	RSCR	VARCHAR2 (256)
		P_PARTKEY	INTEGER

HUB_PART_PK (MD5_HUB_PART, P_NAME

L LNK_PART_SUPPLIER

P	*	MD5_LNK_PART_SUPPLIER	BLOB
F	*	MD5_HUB_SUPPLIER	BLOB
F	*	MD5_HUB_PART	BLOB
F	*	S_NAME	VARCHAR2 (80)
F	*	P_NAME	VARCHAR2 (80)
F	*	P_BRAND	VARCHAR2 (80)
F	*	P_TYPE	VARCHAR2 (80)
F	*	P_SIZE	INTEGER
F	*	P_CONTAINER	VARCHAR2 (80)
	*	LDTS	TIMESTAMP
	*	RSCR	VARCHAR2 (256)

LNK_PART_SUPPLIER_PK (MD5_LNK_PART_SUPPLIER)

LNK_PART_SUPPLIER_HUB_PART_FK (MD5_HUB_PART, P_NAME, F

LNK_PART_SUPPLIER_HUB_SUPPLIER_FK (MD5_HUB_SUPPLIER, S_I

S SAT_LNK_PART_SUPPLIER

PF	*	MD5_LNK_PART_SUPPLIER	BLOB
P	*	LDTS	TIMESTAMP
		PS_AVAILQTY	INTEGER
		PS_SUPPLYCOST	NUMBER (12,2)
		PS_COMMENT	VARCHAR2 (199)
	*	RSCR	VARCHAR2 (256)
		LEDTS	TIMESTAMP

PK_SAT_LNK_PART_SUPPLIER (LDTS, MD5_LNK_PART_SUPPLIER)

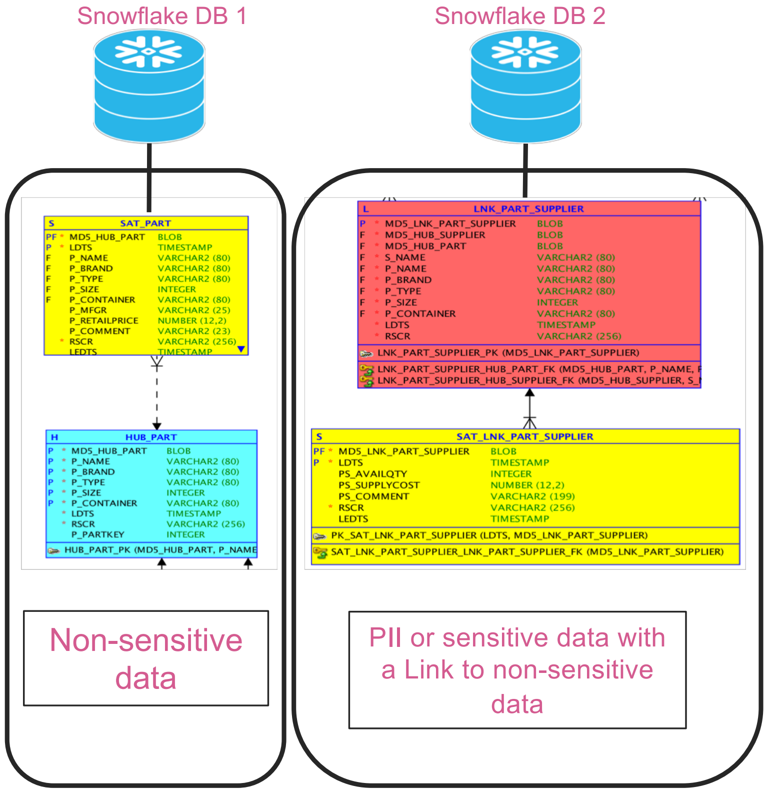
SAT_LNK_PART_SUPPLIER_LNK_PART_SUPPLIER_FK (MD5_LNK_PART_SUPPLIER)

Single Snowflake Database

In Snowflake – might split for security reasons

To THIS!

DV “Physical” Partitioning



S SAT_PART

PF	*	MD5_HUB_PART	BLOB
P	*	LDTS	TIMESTAMP
F		P_NAME	VARCHAR2 (80)
F		P_BRAND	VARCHAR2 (80)
F		P_TYPE	VARCHAR2 (80)
F		P_SIZE	INTEGER
F		P_CONTAINER	VARCHAR2 (80)
		P_MFGR	VARCHAR2 (25)
		P_RETAILPRICE	NUMBER (12,2)
		P_COMMENT	VARCHAR2 (23)
	*	RSCR	VARCHAR2 (256)
		LEDTS	TIMESTAMP

H HUB_PART

P	*	MD5_HUB_PART	BLOB
P	*	P_NAME	VARCHAR2 (80)
P	*	P_BRAND	VARCHAR2 (80)
P	*	P_TYPE	VARCHAR2 (80)
P	*	P_SIZE	INTEGER
P	*	P_CONTAINER	VARCHAR2 (80)
	*	LDTS	TIMESTAMP
	*	RSCR	VARCHAR2 (256)
		P_PARTKEY	INTEGER

HUB_PART_PK (MD5_HUB_PART, P_NAME

L LNK_PART_SUPPLIER

P	*	MD5_LNK_PART_SUPPLIER	BLOB
F	*	MD5_HUB_SUPPLIER	BLOB
F	*	MD5_HUB_PART	BLOB
F	*	S_NAME	VARCHAR2 (80)
F	*	P_NAME	VARCHAR2 (80)
F	*	P_BRAND	VARCHAR2 (80)
F	*	P_TYPE	VARCHAR2 (80)
F	*	P_SIZE	INTEGER
F	*	P_CONTAINER	VARCHAR2 (80)
	*	LDTS	TIMESTAMP
	*	RSCR	VARCHAR2 (256)

LNK_PART_SUPPLIER_PK (MD5_LNK_PART_SUPPLIER)

LNK_PART_SUPPLIER_HUB_PART_FK (MD5_HUB_PART, P_NAME, F

LNK_PART_SUPPLIER_HUB_SUPPLIER_FK (MD5_HUB_SUPPLIER, S_I

S SAT_LNK_PART_SUPPLIER

PF	*	MD5_LNK_PART_SUPPLIER	BLOB
P	*	LDTS	TIMESTAMP
		PS_AVAILQTY	INTEGER
		PS_SUPPLYCOST	NUMBER (12,2)
		PS_COMMENT	VARCHAR2 (199)
	*	RSCR	VARCHAR2 (256)
		LEDTS	TIMESTAMP

PK_SAT_LNK_PART_SUPPLIER (LDTS, MD5_LNK_PART_SUPPLIER)

SAT_LNK_PART_SUPPLIER_LNK_PART_SUPPLIER_FK (MD5_LNK_PART_SUPPLIER)

Non-sensitive data

PII or sensitive data with a Link to non-sensitive data

Productivity

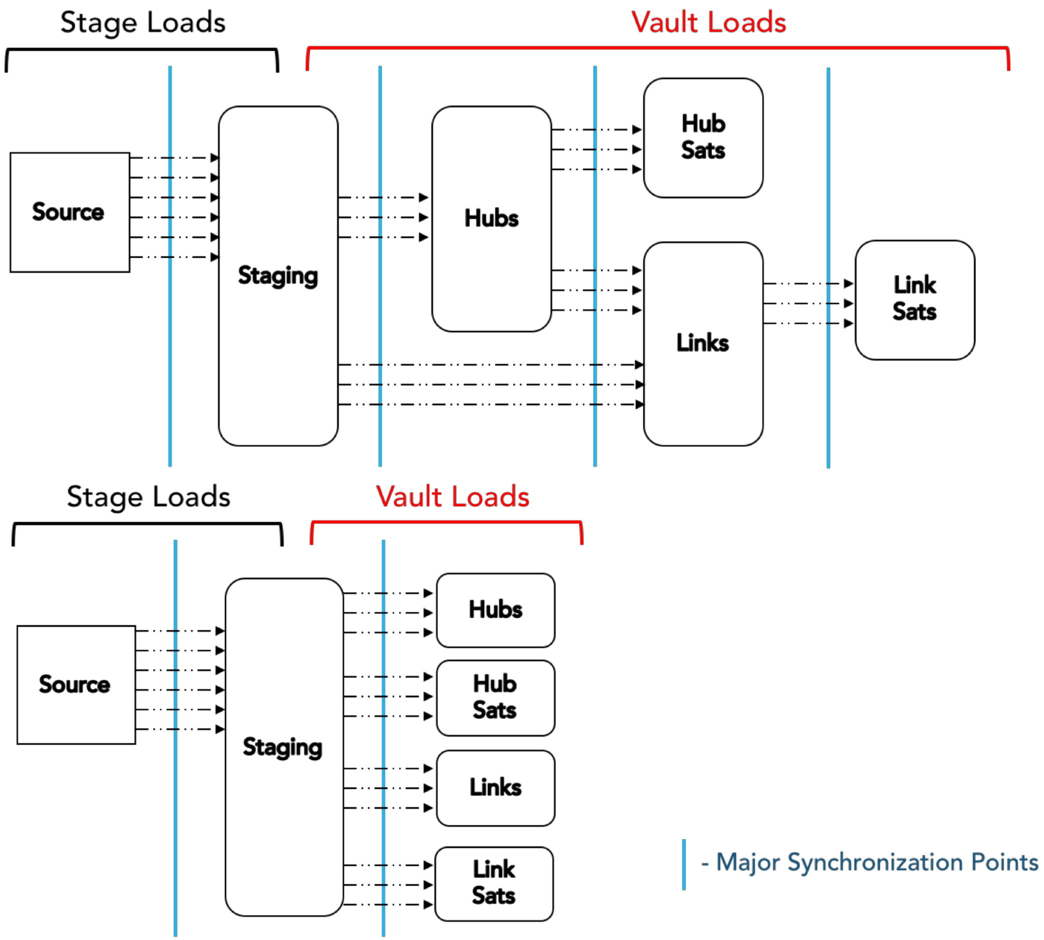
- **Standardized modeling rules**
  - Highly repeatable and learnable modeling technique
  - Can standardize load routines
    - Delta Driven process
    - Re-startable, consistent loading patterns.
    - **Load multiple objects in parallel!**
  - Can standardize extract routines
    - Rapid build of new or revised Data Marts
  - Can be automated (e.g. **WhereScape**)
- **Can use a BI-meta layer to virtualize the reporting structures**
  - Example: Looker using LookML semantic layer
  - Example: BOBJ Universe Business Layer
- **Can put views on the DV structures as well**
  - Simulate ODS/3NF or Star Schemas

Productivity - Loading

Less Scheduling

Non-Deterministic Keys

Deterministic Keys



Other Benefits of a Data Vault

- Modeling the EDW as a DV forces integration of the Business Keys upfront
  - Good for organizational alignment
- An integrated data set with raw data extends it’s value beyond BI:
  - Source for data quality projects
  - Source for master data
  - Source for data mining
  - Source for Data as a Service (DaaS) in an SOA (Service Oriented Architecture)

Other Benefits of Data Vault

- Upfront Hub integration simplifies the data integration routines required to load data marts
  - Helps divide the work a bit
- Much easier to implement security on these granular pieces
- Granular, re-startable processes enable pin-point failure correction
- Designed and optimized for real-time loading in its core architecture (without any tweaks or mods)

Organizations Using Data Vault

- University of Texas, MD Anderson Cancer Center
- Denver Public Schools
- Micron
- Independent Purchasing Cooperative (IPC, Miami)
- Kaplan
- US Defense Department
- Colorado Springs Utilities
- State Court of Wyoming
- Federal Express
- US Dept. of Agriculture

Snowflake Customers using Data Vault

- Aptus Health
- ResearchNow
- F+W Media
- Sainsbury’s

Data Vault Training & Certification

- Several Snowflake Partners offer Data Vault classes
  - ScaleFree - in EMEA
  - PerformanceG2 – in USA
  - Empowered Holdings (Dan Linstedt) – globally
- Talk to you Snowflake account rep for contact information

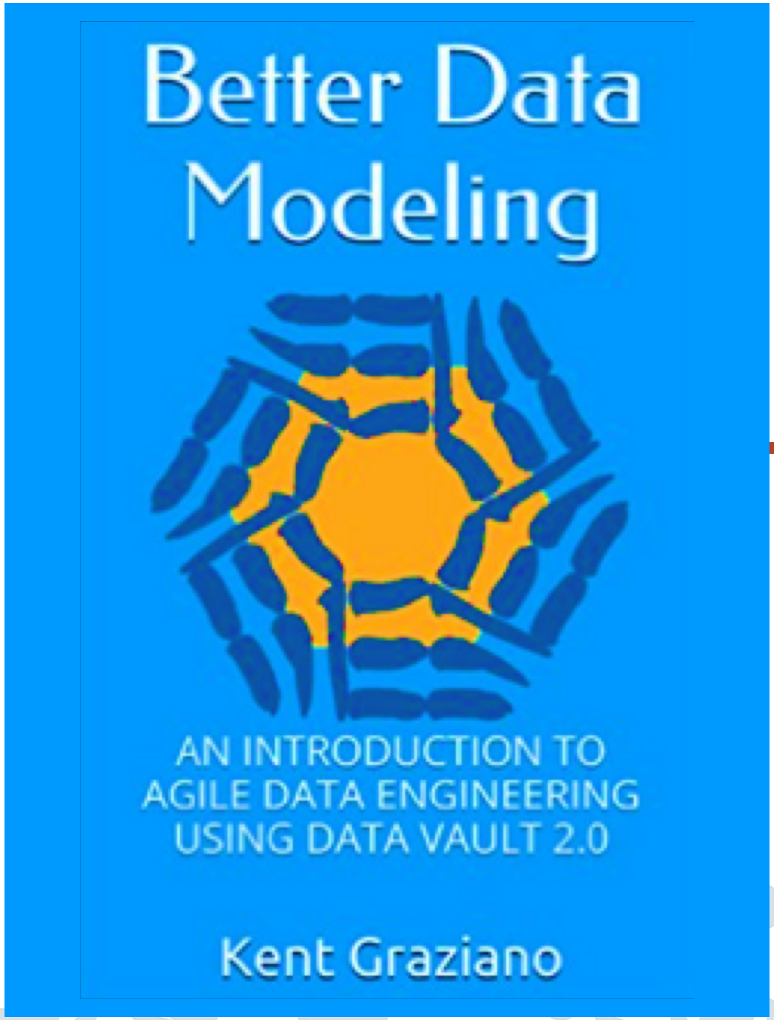
The Experts Say…

"The Data Vault is the optimal choice for modeling the EDW in the DW 2.0 framework." Bill Inmon

"The Data Vault is foundationally strong and exceptionally scalable architecture." Stephen Brobst

"The Data Vault is a technique which some industry experts have predicted may spark a revolution as the next big thing in data modeling for enterprise warehousing...." Doug Laney

References



Better Data Modeling

AN INTRODUCTION TO AGILE DATA ENGINEERING USING DATA VAULT 2.0

Kent Graziano

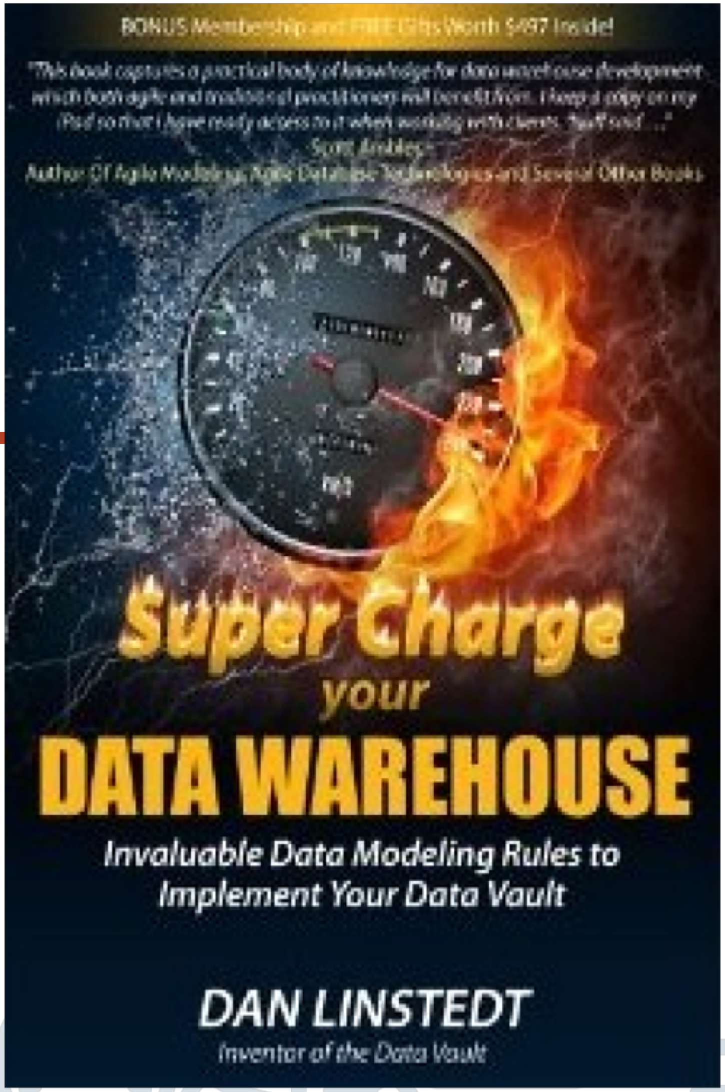
Intro ebook by Kent Graziano:

- **Available on Amazon:**

  http://www.amazon.com/Better-Data-Modeling-Introduction-Engineering-ebook /dp/ B018BREV1C/

Super Charge
Your Data Warehouse

- Available on ***Amazon.com***
- Soft Cover or Kindle Format
- Now also available in PDF at ***LearnDataVault.com***
- Kent was the Technical Editor



BONUS Membership and FREE Gifts Worth $497 Inside!

"This book captures a practical body of knowledge for data warehouse development which both agile and traditional practitioners will benefit from. I keep a copy on my iPad so that I have ready access to it when working with clients. 'Nuff said ..."

Scott Ambler -

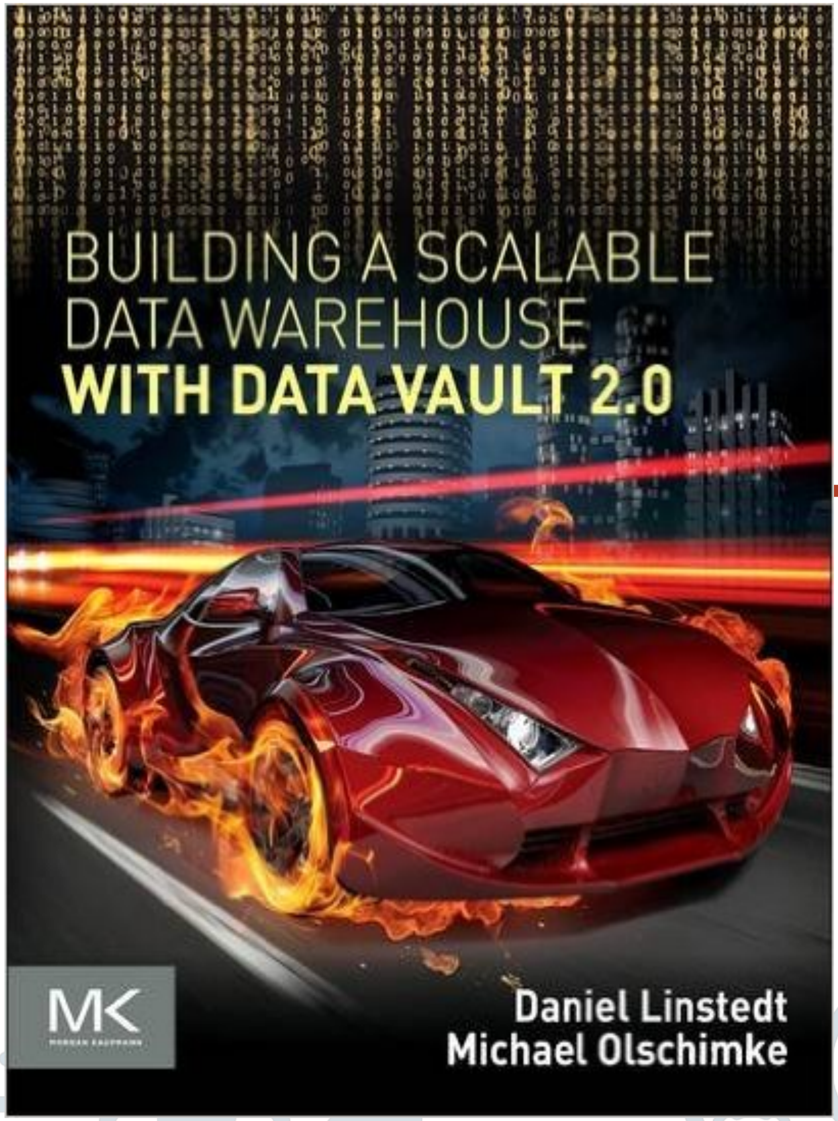
Author Of Agile Modeling, Agile Database Techniques and Several Other Books

Super Charge your DATA WAREHOUSE

Invaluable Data Modeling Rules to Implement Your Data Vault

DAN LINSTEDT

Inventor of the Data Vault



BUILDING A SCALABLE DATA WAREHOUSE WITH DATA VAULT 2.0

MK

Daniel Linstedt
Michael Olschimke

New DV 2.0 Book from Dan Linstedt

- **Available on Amazon:**

  http://www.amazon.com/Building-Scalable-Data-Warehouse-Vault/dp/0128025107/

Contact Information

Kent Graziano
Snowflake Computing
Kent.graziano@snowflake.com
On Twitter @KentGraziano

More info at
http://snowflake.com

Visit my blog at
http://kentgraziano.com

snowflake

THANK YOU