



# Housekeeping

- Introductions
- Thanks to Encana
- Call for participation -Executive Meeting in January
  - Plan for 2006
  - Meeting / events
  - Web site/ Forums



# Enterprise Data Warehouse Using Data Vault

[bruce.mccartney@dbinfosystems.com](mailto:bruce.mccartney@dbinfosystems.com)



# Outline

- Introduction
- Challenges
- What is the Data Vault?
- Where does the Data Vault fit in?
- Data Vault Components
- Loading the Data Vault
- Query/ Data Extraction from the DV
- Issues
- References



# Introduction

- Background – “Information Architecture”
  - Data Integration
  - Data Mart builds (Star Schema)
  - Conformed Dimensions
- Problems
  - One source for facts “as the were at that time”
  - Duplicate effort building ETL/EAI



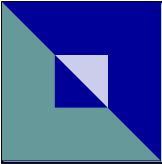
# Challenges in EDW

- Getting it right – “the truth”
- Integration
- Compliance
- Time dependency
- Modeling



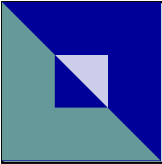
# Challenges – “the truth”

- There is no truth, only facts as they were
- Modeling things in advance forces you to do two hard jobs:
  1. Mind reader
    - Assume user meaning and business context
    - AF\_TYPE
  2. Fortune teller
    - Predict future
      - Business requirements
      - Relationship meaning/aging
- Yahoo vs. Google



# Challenges - Integration

- Unique Business Keys
- Multiple systems carry different parts of data
- Quality varies by source system
- Timing
- EAI Architecture



# Challenges - Compliance

- Require Audit history including before/after values
- Need to be able to reconstruct source data at any point in time
- ISO 9001 – Quality/Process
- SEI Capability Maturity Model (CMM) Level 5  
(Repeatable, consistent, redundant architecture)
- Manage and Enforce Compliance to Sarbanes-Oxley, HIPPA, and BASIL II in your Enterprise Data Warehouse

# Challenges - Time dependency

- Source systems often don't remember old values of data, changed keys etc.
- Near real-time OLTP mixed with Batch 'header file' updates
  - Transactions without customers
- Scaling loads to near real-time
  - EAI and SOA

# Challenges – Modeling a EDW

- **Adapt** some of these:
  - 3NF
    - Rework and inflexible
  - Star Schema Structure
    - Type-2 Dimensions
    - Aggregation and help tables
    - Snowflakes
  - Operational Data Store requirements
    - Audit ability
    - Scalability
  - Query
    - Performance
    - Flexibility



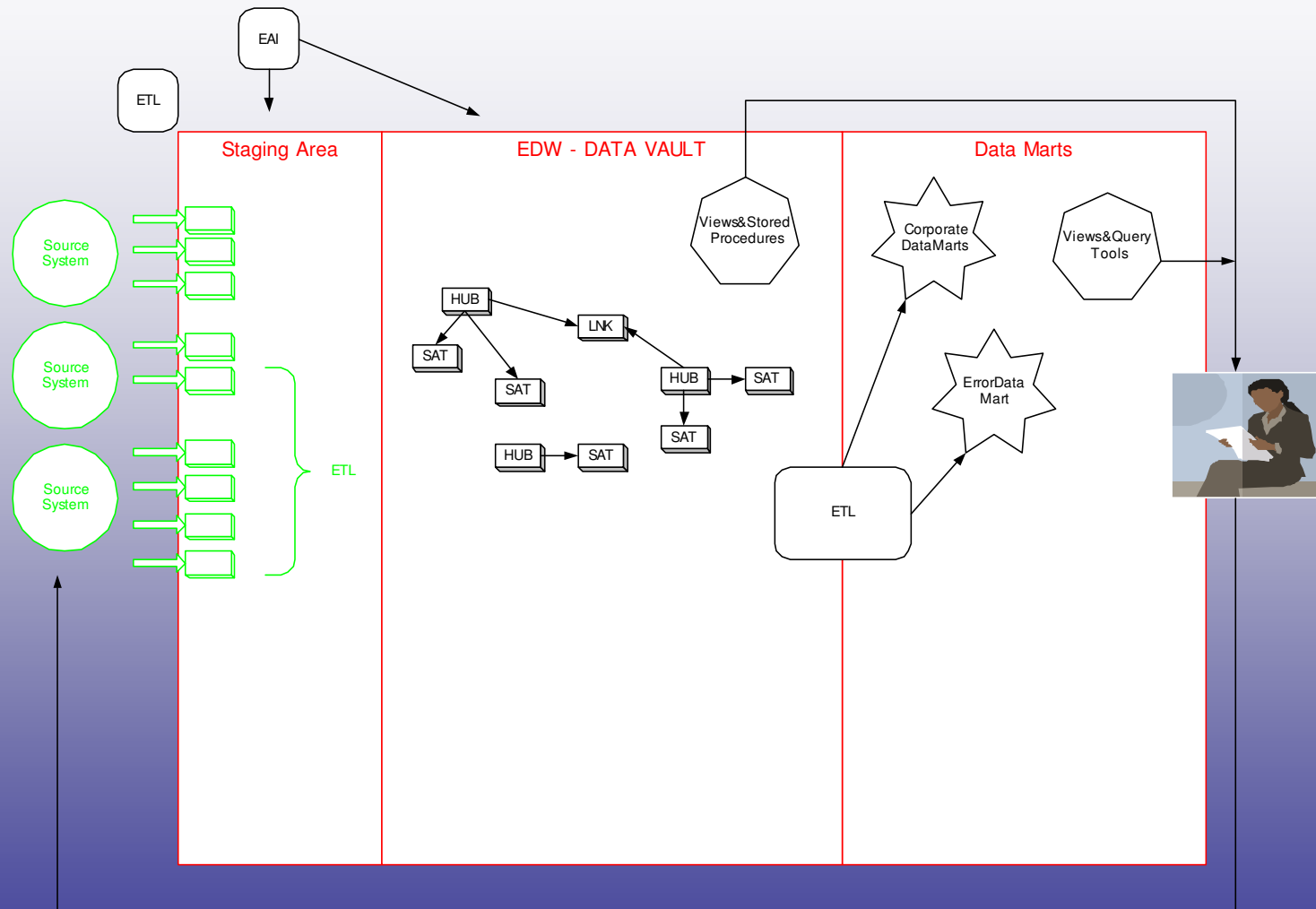
# What is the Data Vault?

- Definition

The Data Vault™ is a detailed, historically oriented, uniquely linked set of normalized tables that support one or more functional areas of business.

- Architected vs. Adapted
- Intended to address the Problems

# Where does the Data Vault fit in?



# How does DV compare to Star Schema Model?

<u>Star Schema Pros</u>	<u>Star Schema Cons</u>
•Good for Multi-Dimensional Analysis	•Not cross-business functional
•Subject Oriented Answers	•Not suited to data mining
•Excellent for Aggregation	•Begins to fail under very large loads
•Rapid Development / Deployment	•Difficult integrating enterprise information
	•Can't handle ODS or Exploration Warehouse Requirements

# How does DV compare to 3<sup>rd</sup>NF model?

<u>3<sup>rd</sup>NF Pros</u>	<u>3<sup>rd</sup>NF Cons</u>
•Many to Many Linkages	•Time Driven PK issues
•Handle lots of information	•Parent-Child Complexities
•Tightly integrated information	•Cascading Change Impacts
•Highly structured	•Top-down model results in rework
•Conducive to near-real time loads	•Not conducive to BI tools
•Relatively easy to extend	•Not conducive to Spiral/scope controlled implementation.
	•Not conducive to Drill-down

# Data Vault Pros/Cons?

<u>DV Pros</u>	<u>DV Cons</u>
•Supports Near-Real Time and Batch Feeds	•Complexity of SQL Joins
•Extensible, Flexible	•Not conducive to today's BI tools.
•Provides rapid build/delivery of Star Schema's	•Not conducive to OLAP processing.
•Supports VLDB/VLDW	•Not always friendly to aggregate levels.
•Designed for EDW	
•Provides granular detail.	
•Supports Data Mining and A.I.	



# What is the Data Vault?

- Key concepts
  - Everything is MANY-TO-MANY
  - Time dependency on everything
  - Late BINDING for data – the LINK
- Uses Relational DBMS
  - Tables/Columns/Views
  - No object orientation



# Data Vault Components

- “Table types”:
  - *Hub* = List of Business Keys
  - *Satellite* = Descriptive Information
  - *Link* = Describes Relationship Between Business Keys

# The HUB

- Definition

- a single table carrying at a minimum a unique list of business keys.
- Other attributes in the Hub include:
  - Surrogate Key – Optional component, possibly a smart key or a sequential number.
  - Load Date Time Stamp – recording when the key itself first arrived in the warehouse.
  - Last Seen Time Stamp – recording when the key itself last arrived in the warehouse, used for data driven 'soft' delete.
  - Record Source – A recording of the source system utilized for data traceability.

- Example (PK Bold)

```
CREATE TABLE HUB_AFE
(
  AFE_NO          VARCHAR2(12 BYTE)          NOT NULL,
  AFE_ID         NUMBER(22)                NOT NULL,
  LOAD_DTTM      DATE                        DEFAULT sysdate          NOT NULL,
  LAST_SEEN_DTTM DATE                        DEFAULT sysdate          NOT NULL,
  DATA_SOURCE   VARCHAR2(240 BYTE)          DEFAULT 'DATA VAULT'      NOT NULL
)
```



# The HUB - Notes

- Unique Business Key is IMPARATIVE
- Insert only loads
- Last seen date can be used to track deletes or reuse of Key (then updates of LAST SEEN)

# The SATELLITE

- Definition

- Provide context (descriptive) information much like a Type-2 dimension, its information is subject to change over time;
- The Satellite is comprised of the following attributes:
  - Satellite Primary Key: Hub or Link Primary Key – migrated into the Satellite from the Hub or Link.
  - Satellite Primary Key: Load Date Time Stamp – recording when the context information is available in the warehouse (the new row is always inserted).
  - Satellite Optional Primary Key: Sequence Surrogate Number – utilized for Satellites that have multiple values (such as a billing and home address), or line item numbers, used to keep the Satellites sub-grouped and in order.
  - Record Source – A recording of the source system utilized for data traceability.

- Example

```
CREATE TABLE SAT_AFE_STATUS
(
  AFE_ID                NUMBER(22)           NOT NULL,
  LOAD_DTTM             DATE                 DEFAULT sysdate      NOT NULL,
  AFE_STATUS            VARCHAR2(4 BYTE)     NOT NULL,
  DATA_SOURCE          VARCHAR2(20 BYTE)    DEFAULT 'DATA VAULT'  NOT NULL,
  STATUS_DESCRIPTION    VARCHAR2(40 BYTE),
  START_DT              DATE,
  ...
  LOAD_END_DTTM         DATE
)
```



# The SATELLITE - Notes

- Many satellites possible per HUB/LINK
  - Group by frequency of change to avoid data explosion due to rapid change
- Date effective
  - FROM LOAD\_DATE to END\_DATE
- Primary Key includes Date and a possible sequence id

# The LINK

- Definition

- Link Entities or Links, are a physical representation of a many-to-many 3NF relationship.
- The Link contains the following attributes:
  - Surrogate Key – Optional component, possibly a smart key or a sequential number.
  - Hub 1 Key to Hub N Key – Hub Keys migrated into the Link to represent the composite key or relationship between two Hubs.
  - Load Date Time Stamp – recording when the relationship/transaction was first created in the warehouse.
  - Last Seen Time Stamp – recording when the relationship/transaction was last seen in the warehouse, used for terminating relationships.
  - Record Source – A recording of the source system utilized for data traceability.

- Example

```
CREATE TABLE LNK_AFE_ACCOUNT
(
  L_AFE_ACC_ID      NUMBER(22)           NOT NULL,
  ACC_ID            NUMBER(22)           NOT NULL,
  AFE_ID            NUMBER(22)           NOT NULL,
  LOAD_DTTM         DATE                 DEFAULT sysdate           NOT NULL,
  LAST_SEEN_DTTM    DATE                 DEFAULT sysdate           NOT NULL,
  DATA_SOURCE      VARCHAR2(20 BYTE)    DEFAULT 'DATA VAULT'     NOT NULL
)
```

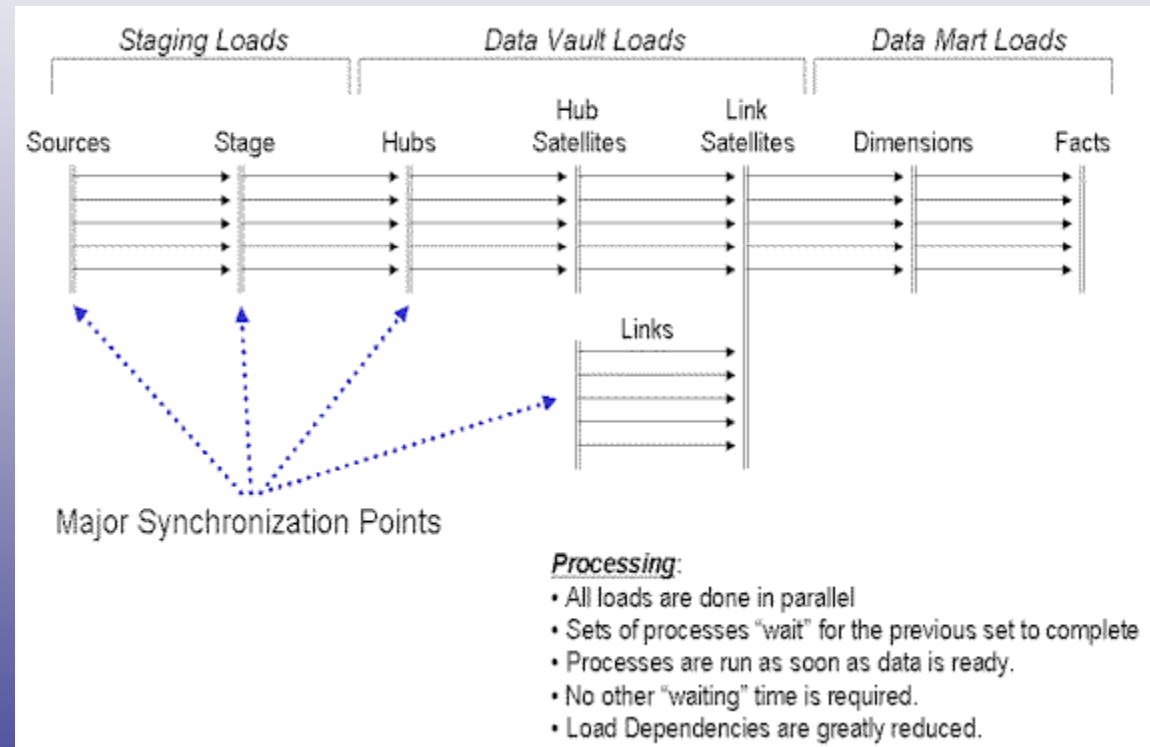


# The LINK - Notes

- Resolves 3<sup>rd</sup>NF relationship(s) amongst HUBs
- Determines GRAIN or level of detail
- Date effective
  - FROM LOAD\_DATE to LAST\_SEEN\_DATE
- Complicated by 'optional' relationships
  - Default may be required to avoid NULL keys

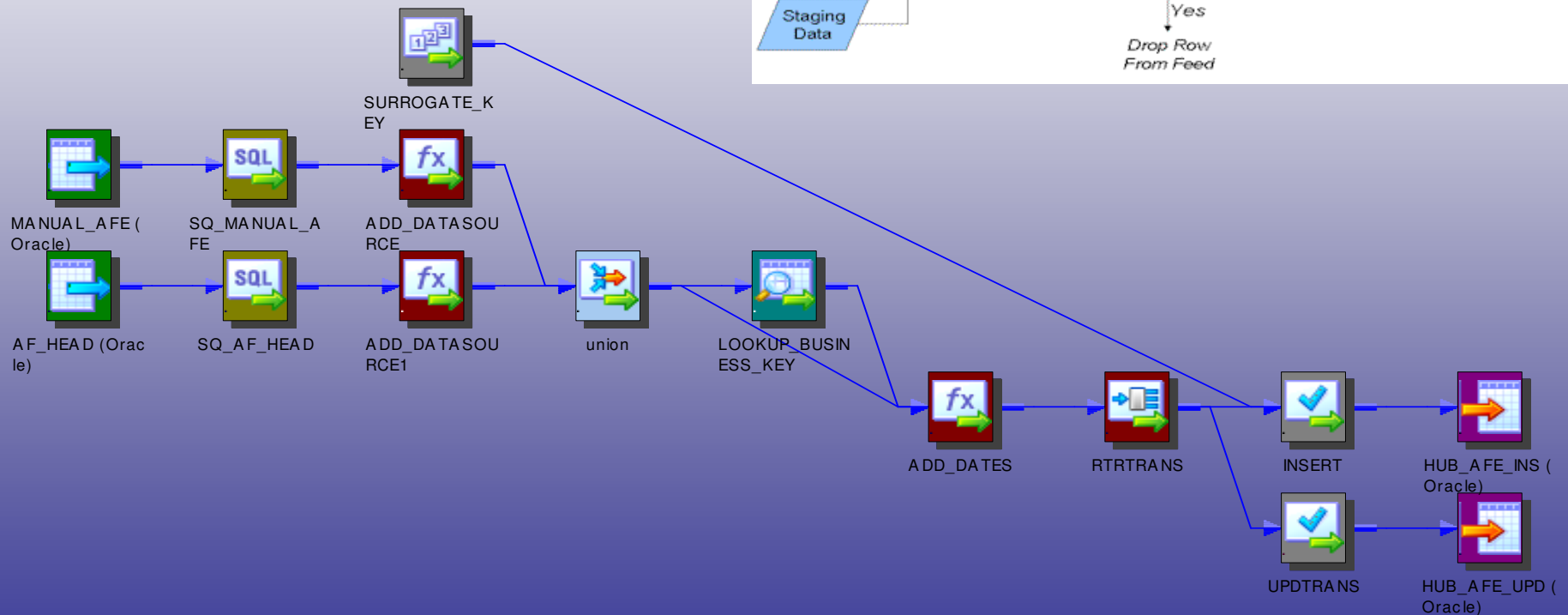
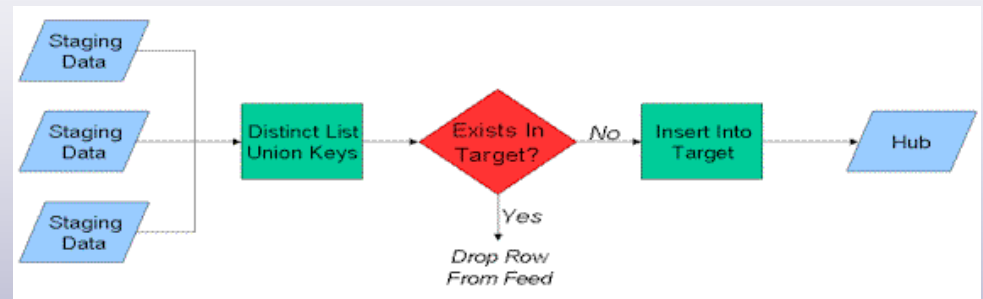
# Loading the Data Vault

- Process overview/goals
  - Scale
  - Near real time
- Issues
  - Business deletes
  - Aging relationships



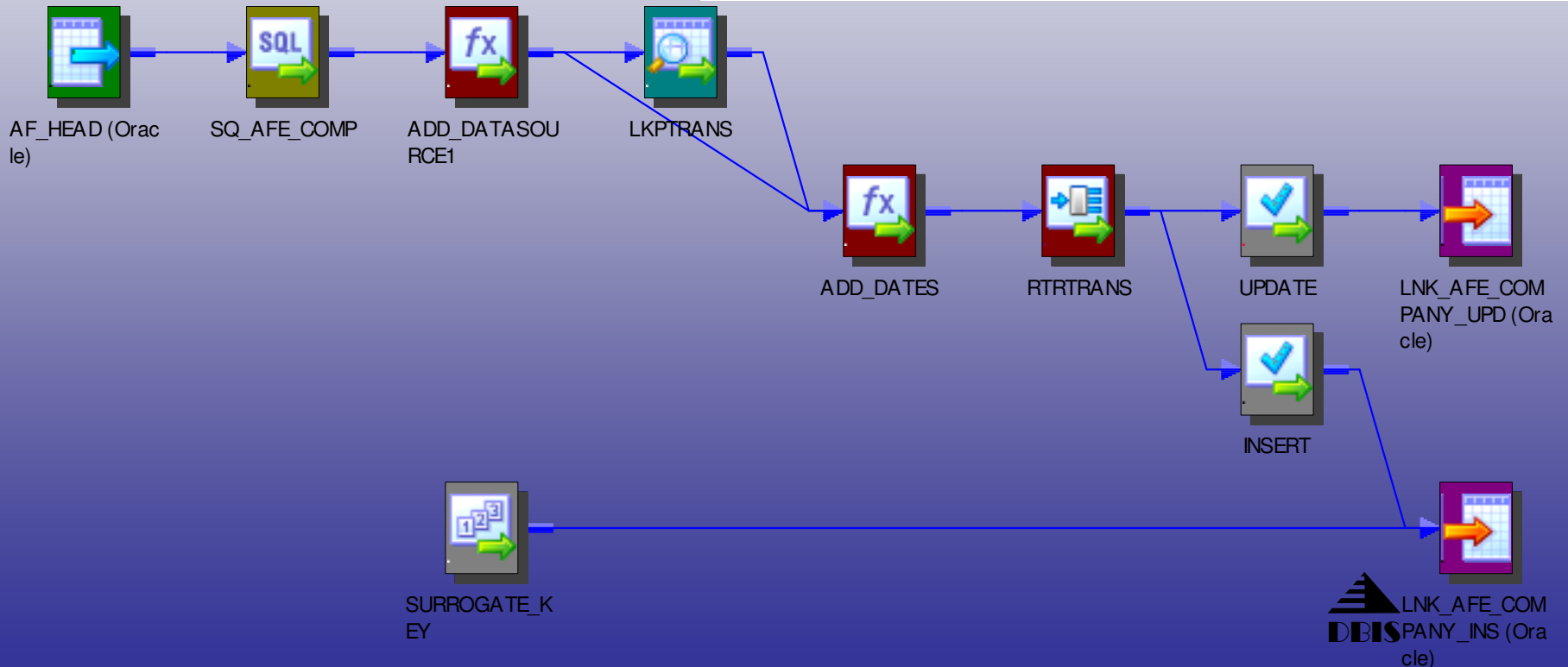
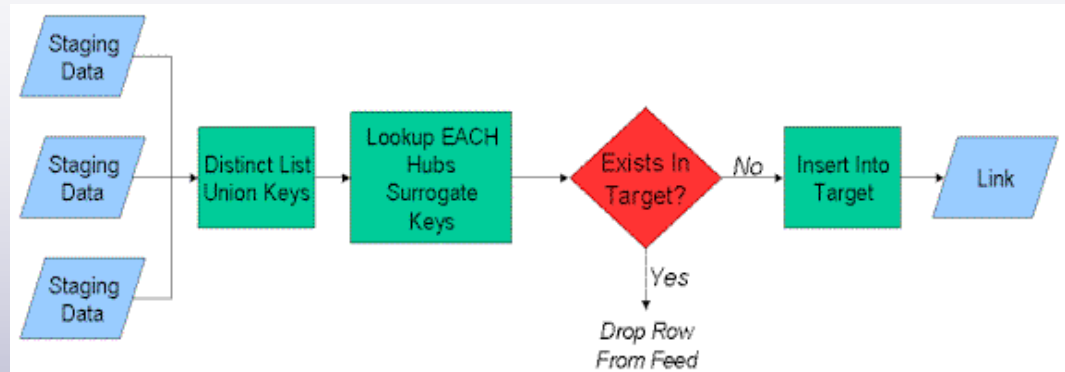
# Loading HUBS

- Not 'incremental'
- Informatica template



# Loading LINKS

- Not 'incremental'
- Informatica template

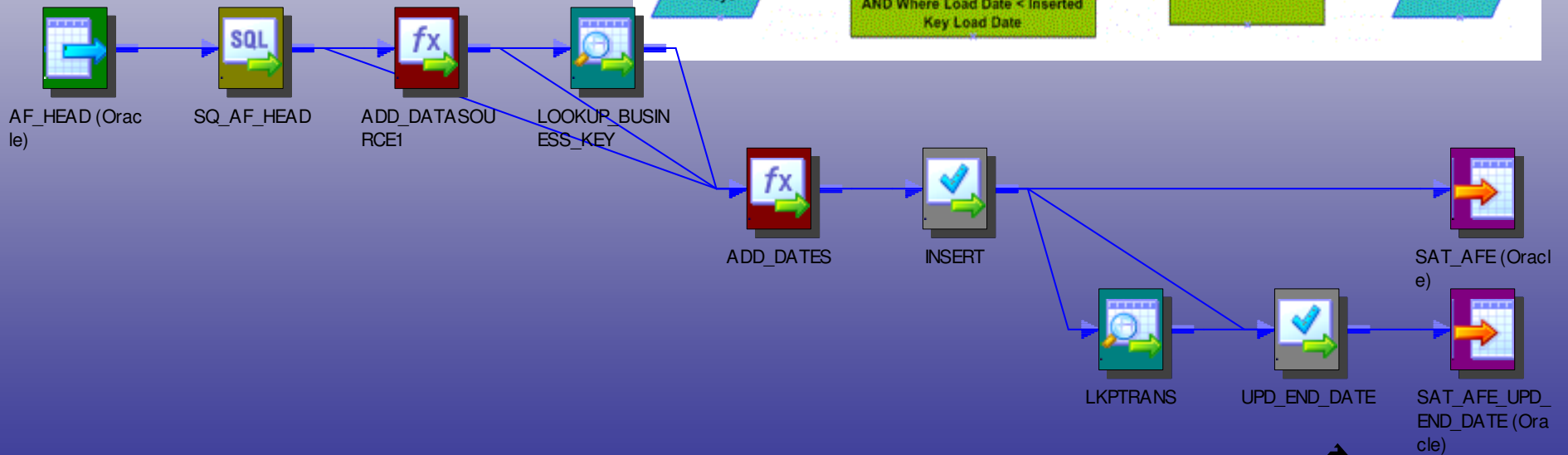
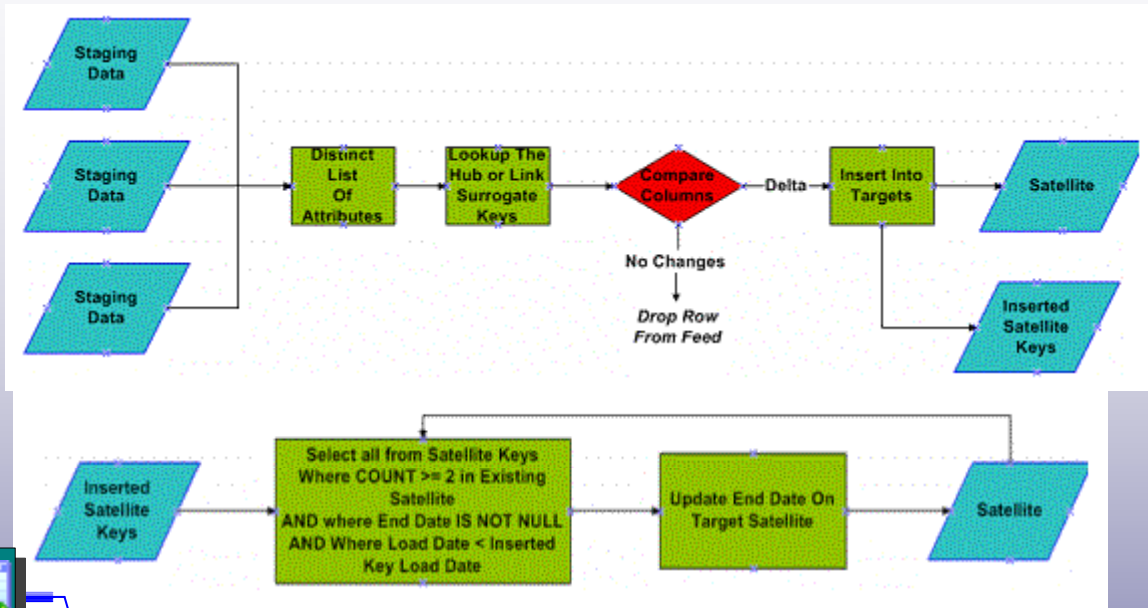


# Loading LINKS – SQL

```
SELECT DISTINCT ha.afe_id, hc.comp_id, 'ENVISION' ds
      FROM envision_src.af_head a, hub_afe ha,
      hub_company hc
      WHERE ha.afe_no = a.afe_no
            AND hc.company_no = a.orig_comp
UNION
SELECT DISTINCT ha.afe_id, hc.comp_id, 'MANUALAFE' ds
      FROM afecontrol_src.manual_afe m,
      hub_afe ha,
      hub_company hc
      WHERE
            AND ha.afe_no = m.afe_number
            AND hc.company_no = m.originating_company
```

# Loading SATELLITES

- May be 'incremental'
- Informatica template



# Loading SATELLITES

```
SELECT DISTINCT envision_src.af_head.afe_no,  
                envision_src.af_head.descr,  
                envision_src.af_head.val_code,  
                ...  
FROM envision_src.af_head, hub_afe hub, sat_afe sat  
WHERE hub.afe_no = envision_src.af_head.afe_no  
AND sat.afe_id(+) = hub.afe_id  
AND (DECODE(envision_src.af_head.DESCR, sat.DESCRPTION , 1, 0) = 0  
     OR  
     DECODE (envision_src.af_head.atype, sat.AFE_TYPE , 1, 0) = 0 OR  
     ...  
)
```



# Query/ Data Extraction from the DV

- Views to mask complexity
- Point-in-time tables
- CURRENT tables
- SQL Examples/Challenges
- BO Universe
- Performance issues

# Sample View – “Latest AFE Data”

```
CREATE OR REPLACE VIEW AFE
(AFE_ID, AFE_NO, COMPANY_NO, AFE_COMP_TYPE, DESCRIPTION,
ACCRUABLE_FLAG, ACTIVITY, AFE_TYPE, AFE_TYPE_DESCRIPTION, APPROVAL_REQUIRED_FLAG,
...
LAST_USED_DT, RECLASSIFICATION_CODE, START_DT)
AS
SELECT h.afe_id, h.afe_no, h2.company_no, s3.afe_comp_type, s1.description,      s1 accruable_flag, s1.activity, s1.afe_type,
      s1.afe_type_description,
      s1.approval_required_flag, s1.blanket_afe_flag, s1.data_source,      s1.doi_group, s1.gross_interest, s1.net_interest,
      s1.supplemented_flag,
      ...
      s2.afe_status, s2.status_description, s2.estimated_completion_dt,      s2.closeout_code, s2.closed_dt, s2.distribution_dt,
      s2.last_closed_dt, s2.last_used_dt,      s2.reclassification_code, s2.start_dt
FROM hub_afe h,
      sat_afe s1,
      sat_afe_status s2,
      lnk_afe_company l1,
      sat_afe_comp s3,
      hub_company h2
WHERE h.afe_id = s1.afe_id(+)
      AND h.afe_id = s2.afe_id(+)
      AND h.afe_id = l1.afe_id(+)
      AND s3.l_afe_comp_id = l1.l_afe_comp_id
      AND h2.comp_id = l1.comp_id
      AND (s1.load_end_dttm IS NULL)
      AND (s2.load_end_dttm IS NULL)
      AND (s3.load_end_dttm IS NULL);
```



# Point-in-time tables

- Optional
- Use to populate values as specific “points in time” e.g. Monthly Production Volumes
- Join to a TIME table



# CURRENT tables

- Optional
- Most recent copy of everything related to a HUB
  - LOAD\_END\_DATE is NULL
- Managed via triggers/etl



# Issues

- Source system delete strategy
  - Managed by “last seen date”
- Scaling Performance
  - Lots of outer joins and semi joins in queries
  - Not intended for ad hoc end user access
  - Best practice to encapsulate complexity via PIT/VIEW or API/Web Service



# References

- Dan Linstedt (<http://www.danlinstedt.com>)
  - Forums – Dan's site
  - Papers <http://www.myersholum.com/whitepapers.htm>
  - Services/Training
    - Myers-Holum + Certified Consultants
    - TDWI Conference - one day seminar