ETL

Data Structures

Cyrus Lentin

Data Structures

- Flat Files
- XML Files
- Relational Tables

Flat Files

- Data Is Stored In Columns And Rows Within A File On Your File System To Emulate A Database Table
- Data Stored In Standard Manner Known As American Standard Code For Information Interchange (ASCII).
- ASCII Flat Files Can Be Processed / Manipulated By ETL Tools Or Scripting Languages Very Fast!

Advantages

- No Overhead To Maintain Metadata About The Data Being Processed
- Processes Like Sorting, Merging, Deleting, Replacing, And Data-migration Functions Are Very Fast
- Many Utility Programs Are Dedicated To Text-file Manipulation.

Disadvantage

- Meta Data Not In-Built
- Formatting Not In-Built

Best Suited For

- Staging Source Data For Safekeeping And Recovery
- Sorting Data
- Filtering
- Aggregation
- Referencing Source Data

Flat Files – Types

Fixed Width Files

- All Rows Have Same Width
- All Fields (Columns) In The Rows Also Have The Same Width
- No Field Level Delimiter Required
- Row Level Delimiter Is \n or \r\n
- Sample

```
Tim  M 5014510/21/1978
Sara F 6518011/23/1965
```

Delimited Files

- Rows Have Variable Width
- Fields (Columns) Also Have Variable Width
- Field Level Delimiter Required; generally , for .csv
- Row Level Delimiter Is \n or \r\n
- Qualifier Character Is Required
- Sample
  Tim, M, 50, 145, 10/21/1978
  Sara, F, 65, 180, 11/23/1965

XML Files

- XML is a for data communication or data exchange / not generally used for persistent staging in ETL
- XML is very common format for both input to and output from the ETL system
- XML takes the form of plain text documents containing both data and metadata but no formatting info
- XML metadata consists of tags unambiguously identifying each item in an XML document.
- XML has capability for declaring hierarchical structures, such as complex forms with nested fields.
- For instance, an invoice coded in XML contains sequences such as:
  <Customer Name = "Bob" Address= "123 Main Street" City= "Philadelphia" />

Advantages

- Meta Data In-Built
- Processes Like Sorting, Merging, Deleting, Replacing, And Data-migration Functions Are Possible

Disadvantage

- Overheads To Store & Read Meta DATA (huge in large-volume data transfer)
- Formatting Not In-Built

Best Suited For

- XML defines a universal language for data sharing

Links

- http://www.w3schools.com/xml/xml_whatis.asp

JSON Files

- JSON is a for data communication or data exchange / not generally used for persistent staging in ETL
- JSON is very common format for both input to and output from the ETL system
- JSON takes the form of plain text files containing both data and metadata but no formatting info
- JSON Files consists key:value pair; the unambiguously identifying each item a document.
- For instance, an invoice coded in XML contains sequences such as:

  { "employee": {"name":"sonoo","salary":56000, "married":true} }

Advantages

- Meta Data In-built
- Better Mix Of Flat File & XML Capability

Disadvantage

- Overheads To Store & Read Meta DATA
- Formatting Not In-Built

Best Suited For

- JSON Format is designed to work with Java Script and is best suited for data transfer over HTTP

Links

- http://www.w3schools.com/js/js_json_intro.asp
- http://www.jsoneditoronline.org/

Relational Tables

- Staging Data Can Optionally Be Stored Within The Confines Of A Relational DBMS
- Using Database Tables Is Most Appropriate Especially When You Don't Have A Dedicated ETL Tool

Advantages

- Apparent Metadata
- Relational Abilities
- Open Repository
- DBA Support
- SQL Interface

Disadvantage

- RDBMS Overheads
- Dedicated Support

Best Suited For

- Creating Persistent Data For Other ETL Processes When Data Is Already In RDBMS

Links

- https://en.wikipedia.org/wiki/Relational_database_management_system
- https://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm

Data Structure Consideration

- RDBMS Tables
  - Transaction Model
  - Dimensional Model
  - Independent
- Normalization
  - Required
  - Not Required
- Heterogeneous Data
  - Relational
  - Non Relational
- Impact Analysis
  - New Requirement
  - Change Request
- Metadata Capture
  - Data Lineage
  - Business Definitions
  - Technical Definitions
  - Process Metadata

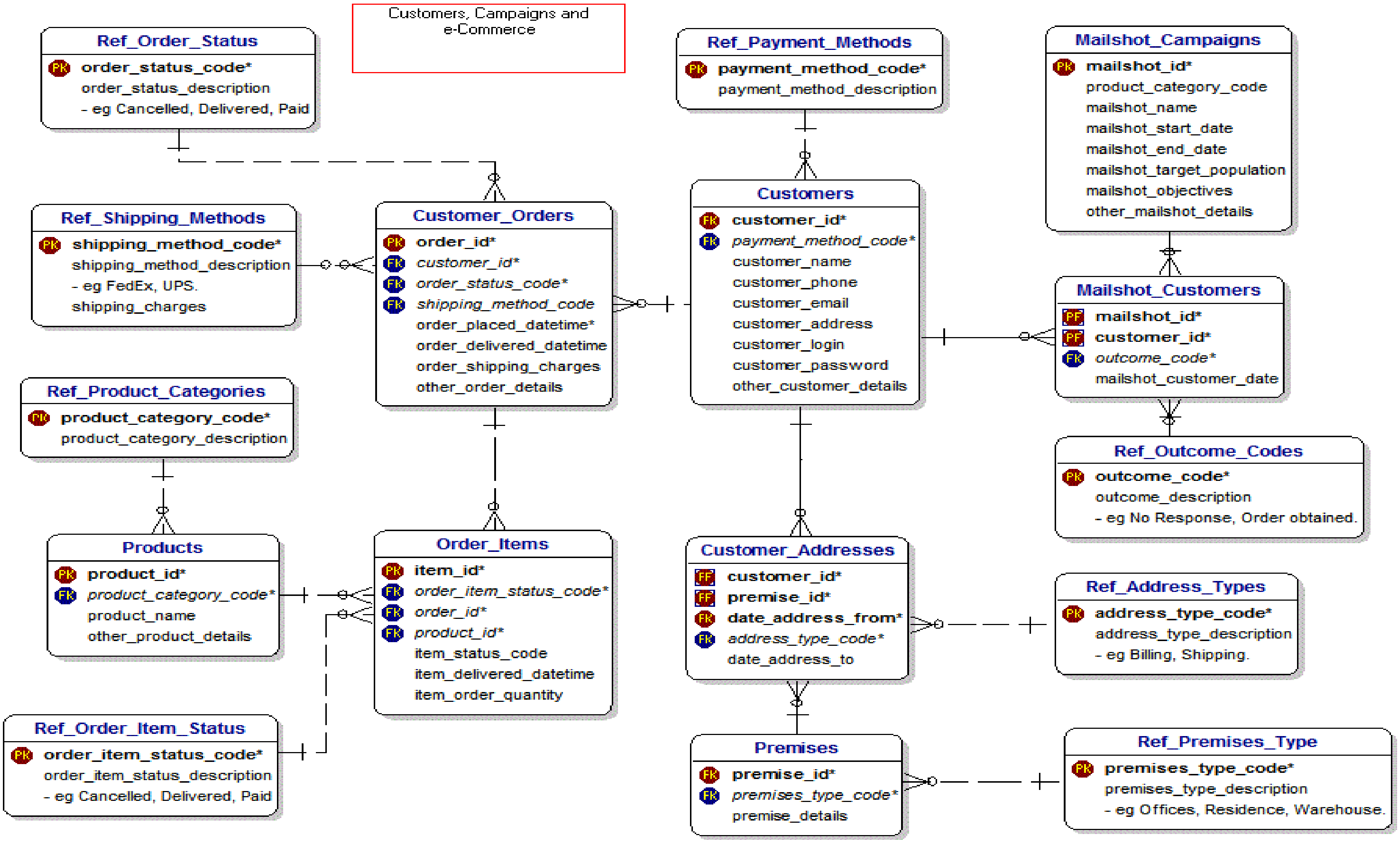
Fact Tables

- Fact Table Consists Of The Measurements, Metrics Or Facts Of A Business Process
- Fact Table Is Located At The Center Of A Star Schema Or A Snowflake Schema Surrounded By Dimension Tables
- Where Multiple Fact Tables Are Used, These Are Arranged As A Fact Constellation Schema
- A Fact Table Typically Has Two Types Of Columns:
  - Those That Contain Facts
  - Those That Are A Foreign Key To Dimension Tables
- Fact Tables Contain The Content Of The Data Warehouse And Store Different Types Of Measures Like Additive, Non Additive, And Semi Additive Measures

Dimension Table

- A Dimension Table Is One Of The Set Of Companion Tables To A Fact Table.
- Dimension Tables Contain Descriptive Attributes (Or Fields) That Are Typically Textual Fields (Or Discrete Numbers That Behave Like Text).
- These Fields Are Designed To Serve Two Critical Purposes:
  - Query Constraining And/Or Filtering
  - Query Result Set Labeling.
- Dimension Fields Should Be:
  - Verbose (Labels Consisting Of Full Words)
  - Descriptive
  - Complete (Having No Missing Values)
  - Discretely Valued (Having Only One Value Per Dimension Table Row)
  - Quality Assured (Having No Misspellings Or Impossible Values)

Snowflake Schema – Typical RDBMS Architecture



Customers, Campaigns and e-Commerce

Ref_Order_Status
PK order_status_code*
order_status_description
- eg Cancelled, Delivered, Paid

Ref_Payment_Methods
PK payment_method_code*
payment_method_description

Mailshot_Campaigns
PK mailshot_id*
product_category_code
mailshot_name
mailshot_start_date
mailshot_end_date
mailshot_target_population
mailshot_objectives
other_mailshot_details

Ref_Shipping_Methods
PK shipping_method_code*
shipping_method_description
- eg FedEx, UPS.
shipping_charges

Customer_Orders
PK order_id*
FK *customer_id**
FK *order_status_code**
FK *shipping_method_code*
order_placed_datetime*
order_delivered_datetime
order_shipping_charges
other_order_details

Customers
PK customer_id*
FK *payment_method_code**
customer_name
customer_phone
customer_email
customer_address
customer_login
customer_password
other_customer_details

Mailshot_Customers
PF mailshot_id*
PF customer_id*
FK *outcome_code**
mailshot_customer_date

Ref_Product_Categories
PK product_category_code*
product_category_description

Ref_Outcome_Codes
PK outcome_code*
outcome_description
- eg No Response, Order obtained.

Products
PK product_id*
FK *product_category_code**
product_name
other_product_details

Order_Items
PK item_id*
FK *order_item_status_code**
FK *order_id**
FK *product_id**
item_status_code
item_delivered_datetime
item_order_quantity

Customer_Addresses
PF customer_id*
PF premise_id*
PK date_address_from*
FK *address_type_code**
date_address_to

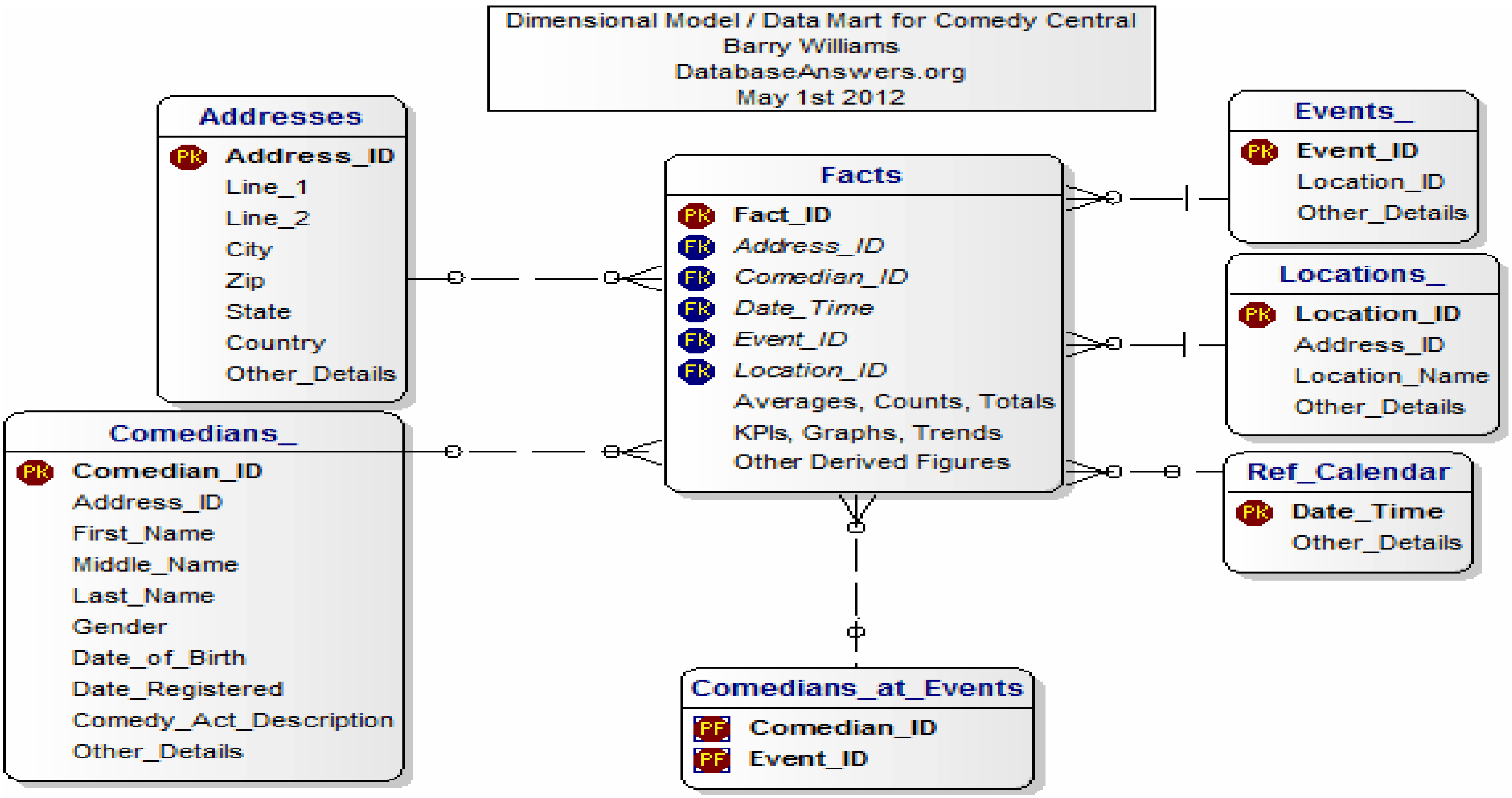
Ref_Address_Types
PK address_type_code*
address_type_description
- eg Billing, Shipping.

Ref_Order_Item_Status
PK order_item_status_code*
order_item_status_description
- eg Cancelled, Delivered, Paid

Premises
PK premise_id*
FK *premises_type_code**
premise_details

Ref_Premises_Type
PK premises_type_code*
premises_type_description
- eg Offices, Residence, Warehouse.

Star Schema – Typical BI-Tool Architecture



Dimensional Model / Data Mart for Comedy Central
Barry Williams
DatabaseAnswers.org
May 1st 2012

Star Schema v/s Snowflake Schema Comparison – Typical BI-Tool Architecture

	Star Schema	Snowflake Schema
Ease of maintenance	Has redundant data and hence difficult to maintain / change	No redundancy, so snowflake schemas are easier to maintain and change
Ease of Use	Less query complexity and easy to understand	More complex queries and hence difficult to understand
Query Performance	Less number of foreign keys and hence shorter query execution time (faster)	More foreign keys and hence longer query execution time (slower)
Type of Data Warehouse	Good for DataMart's with simple relationships (1:1 or 1:many)	Good to use for data warehouse with complex relationships (many:many)
Joins	Less Joins	Higher number of Joins
Normalization / DeNormalization	Both Dimension and Fact Tables are in De-Normalized form	Dimension Tables are in Normalized Fact Tables Are De-Normalized

Staging Tables Volumetric Worksheet

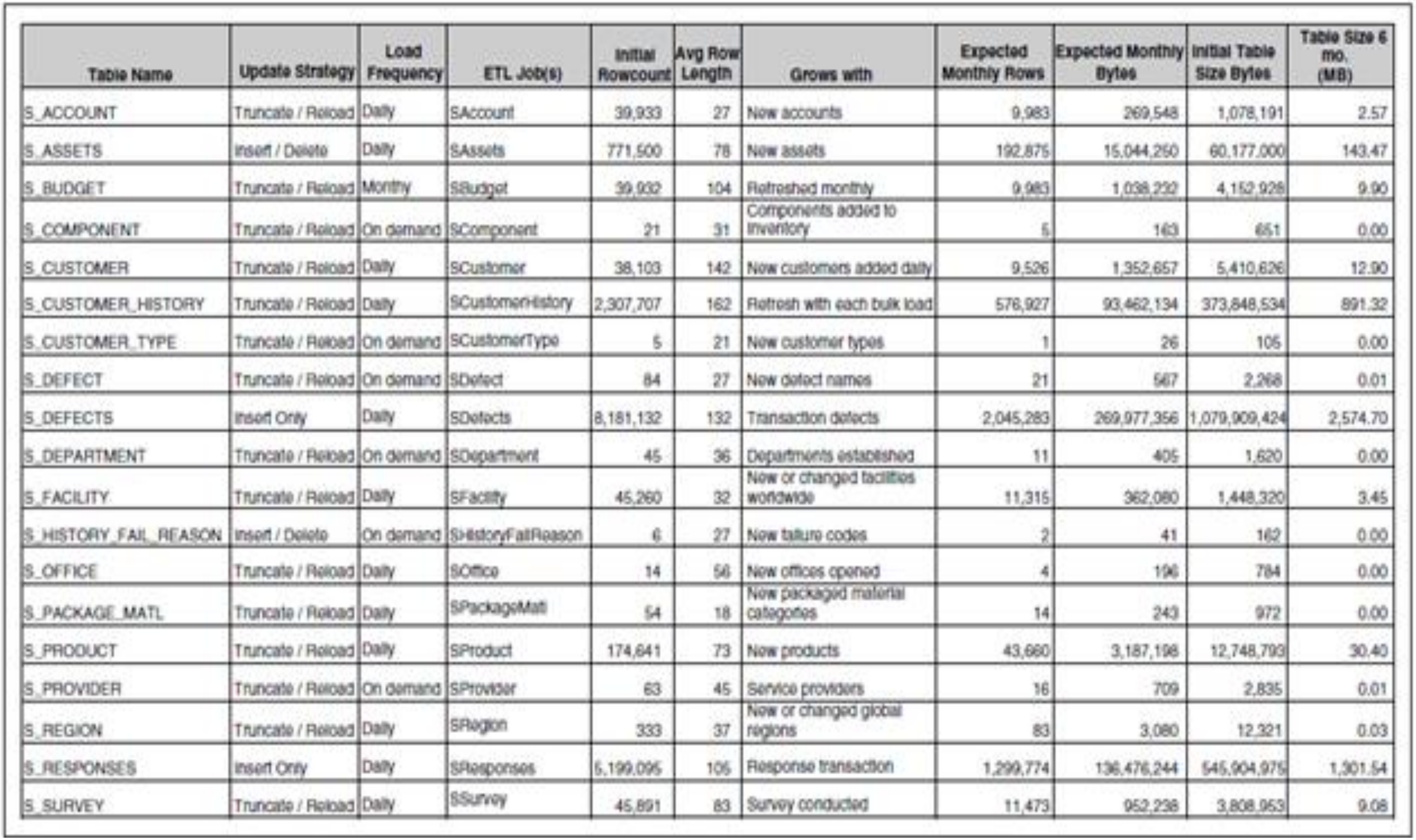


Table Name	Update Strategy	Load Frequency	ETL Job(s)	Initial Rowcount	Avg Row Length	Grows with	Expected Monthly Rows	Expected Monthly Bytes	Initial Table Size Bytes	Table Size 6 mo. (MB)
S_ACCOUNT	Truncate / Reload	Daily	SAccount	39,933	27	New accounts	9,983	269,548	1,078,191	2.57
S_ASSETS	Insert / Delete	Daily	SAssets	771,500	78	New assets	192,875	15,044,250	60,177,000	143.47
S_BUDGET	Truncate / Reload	Monthly	SBudget	39,932	104	Refreshed monthly	9,983	1,038,232	4,152,928	9.90
S_COMPONENT	Truncate / Reload	On demand	SComponent	21	31	Components added to inventory	5	163	651	0.00
S_CUSTOMER	Truncate / Reload	Daily	SCustomer	38,103	142	New customers added daily	9,526	1,352,657	5,410,626	12.90
S_CUSTOMER_HISTORY	Truncate / Reload	Daily	SCustomerHistory	2,307,707	162	Refresh with each bulk load	576,927	93,462,134	373,848,534	891.32
S_CUSTOMER_TYPE	Truncate / Reload	On demand	SCustomerType	5	21	New customer types	1	26	105	0.00
S_DEFECT	Truncate / Reload	On demand	SDefect	84	27	New defect names	21	567	2,268	0.01
S_DEFECTS	Insert Only	Daily	SDefects	8,181,132	132	Transaction defects	2,045,283	269,977,356	1,079,909,424	2,574.70
S_DEPARTMENT	Truncate / Reload	On demand	SDepartment	45	36	Departments established	11	405	1,620	0.00
S_FACILITY	Truncate / Reload	Daily	SFacility	45,260	32	New or changed facilities worldwide	11,315	362,080	1,448,320	3.45
S_HISTORY_FAIL_REASON	Insert / Delete	On demand	SHistoryFailReason	6	27	New failure codes	2	41	162	0.00
S_OFFICE	Truncate / Reload	Daily	SOffice	14	56	New offices opened	4	196	784	0.00
S_PACKAGE_MATL	Truncate / Reload	Daily	SPackageMatl	54	18	New packaged material categories	14	243	972	0.00
S_PRODUCT	Truncate / Reload	Daily	SProduct	174,641	73	New products	43,660	3,187,198	12,748,793	30.40
S_PROVIDER	Truncate / Reload	On demand	SProvider	63	45	Service providers	16	709	2,835	0.01
S_REGION	Truncate / Reload	Daily	SRegion	333	37	New or changed global regions	83	3,080	12,321	0.03
S_RESPONSES	Insert Only	Daily	SResponses	5,199,095	105	Response transaction	1,299,774	136,476,244	545,904,975	1,301.54
S_SURVEY	Truncate / Reload	Daily	SSurvey	45,891	83	Survey conducted	11,473	952,238	3,808,953	9.08

Staging Tables Volumetric Worksheet - Information

- Table Name
- Load Frequency
- ETL Job
- Average Row Length
- GrowsWith
- Expected Monthly Rows
- Initial Table Size
- Table Size 6 Months

Thank you!

Contact:

Cyrus Lentin
cyrus@lentins.co.in
+91-98200-94236