

E T L

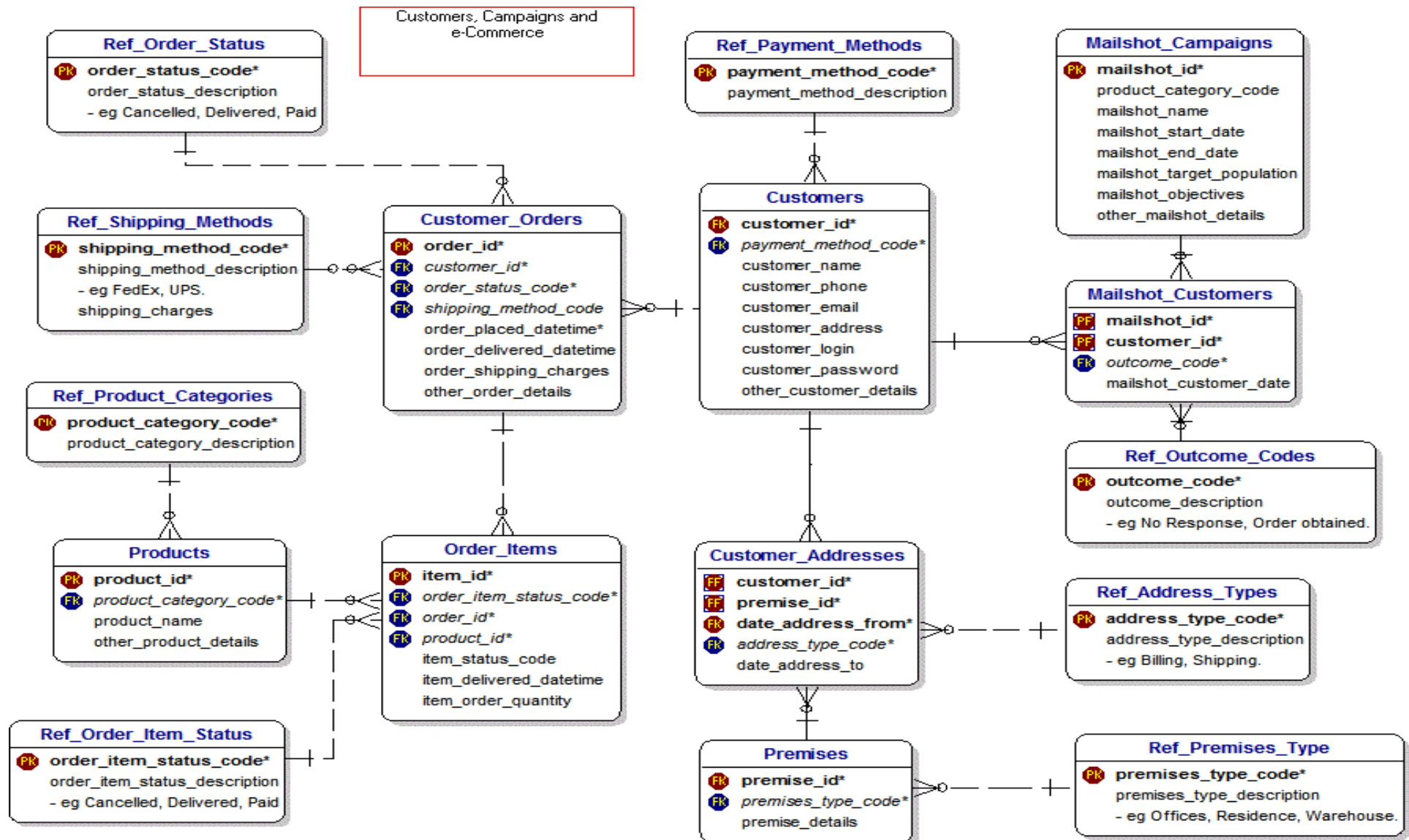
Delivery / Loading

Cyrus Lentin

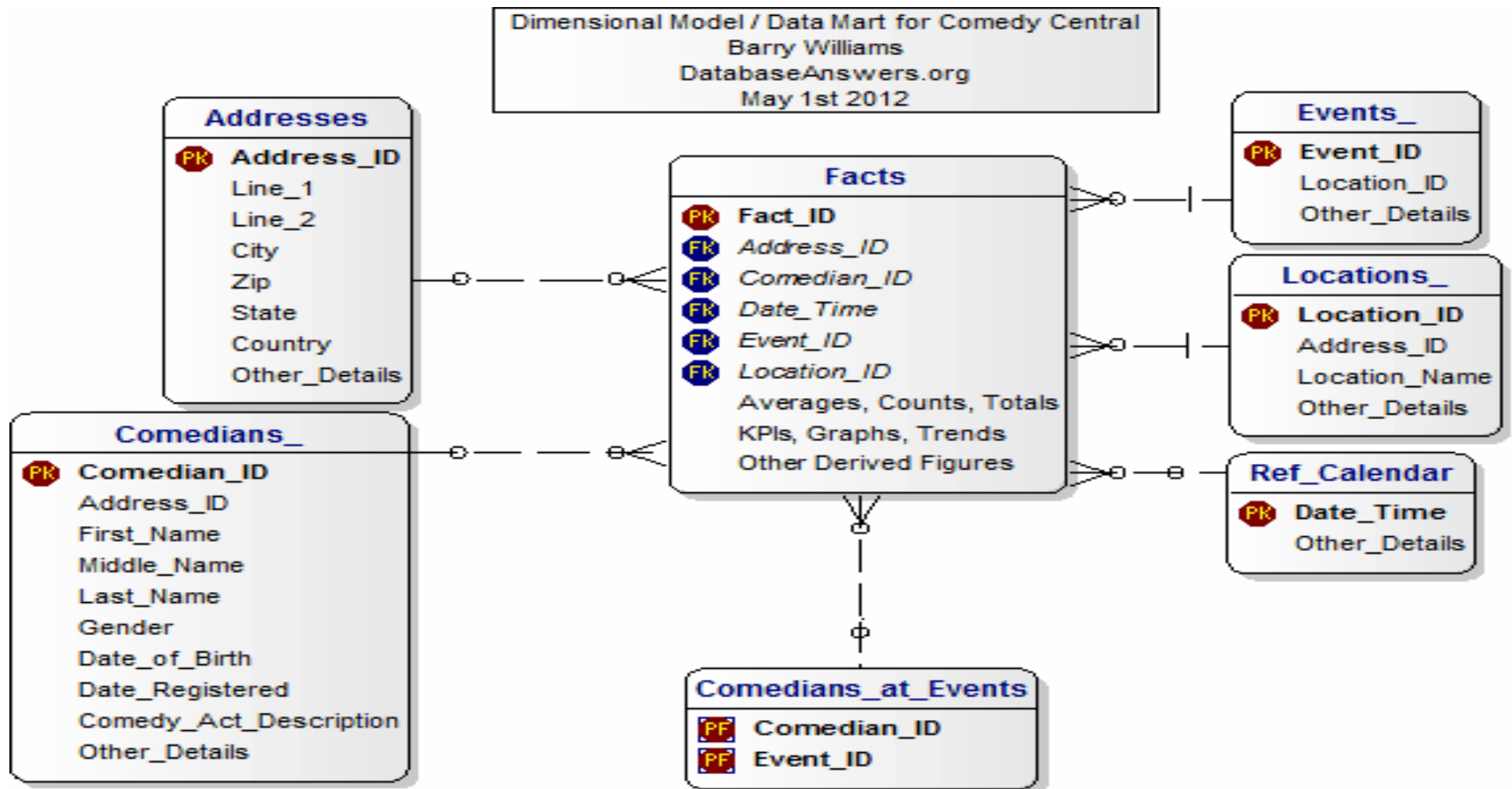
Schemas

- Snowflake Schema
- Star Schema
- Flat / Single Table

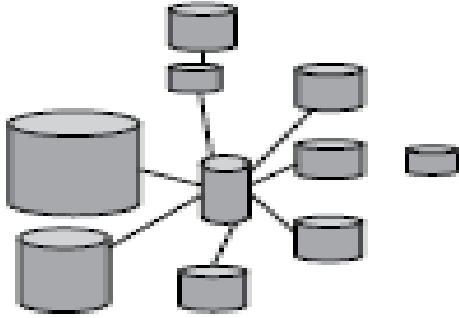
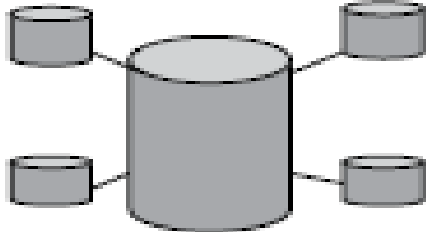
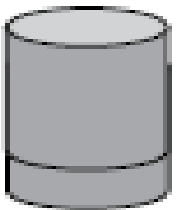
Snowflake Schema – Typical RDBMS Architecture



Star Schema – Typical BI-Tool Architecture



Schemas

	Option 1 Snowflake	Option 2 Star Schema	Option 3 Single Table
			
Response Time	Satisfactory	Good	Excellent
RAM consumption	Good	Good	Bad
Script run time	Good	Excellent	Bad
Flexibility Model	Poor	Excellent	Excellent
Complexity Script	Poor	Excellent	Excellent

Star Schema v/s Snowflake Schema

	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

Normalization

First Normal Form

- each set of column must have a unique value
- no two Cols can contain repeating group of information

Adam 20 Maths, Science

Adam	20	Maths
Adam	20	Science

Normalization

First Normal Form

- each set of column must have a unique value
- no two Cols can contain repeating group of information

Adam 20 Maths, Science

Adam 20

Maths

Adam 20

Science

Second Normal Form

- no rows of data must contain repeating data
- use primary key and segregate data in a table

Adam 20 Maths

S001 Adam 20

S001 Maths

Adam 20 Science

S001 Science

Normalization

First Normal Form

- each set of column must have a unique value
- no two Cols can contain repeating group of information

Adam	20	Maths, Science	Adam	20	Maths
			Adam	20	Science

Second Normal Form

- no rows of data must contain repeating data
- use primary key and segregate data in a table

Adam	20	Maths	S001 Adam	30	S001 Maths
Adam	20	Science			S001 Science

Third Normal Form

- every **non-prime attribute of table** must be dependent on primary key
- again separate non-prime attributes to a separate table

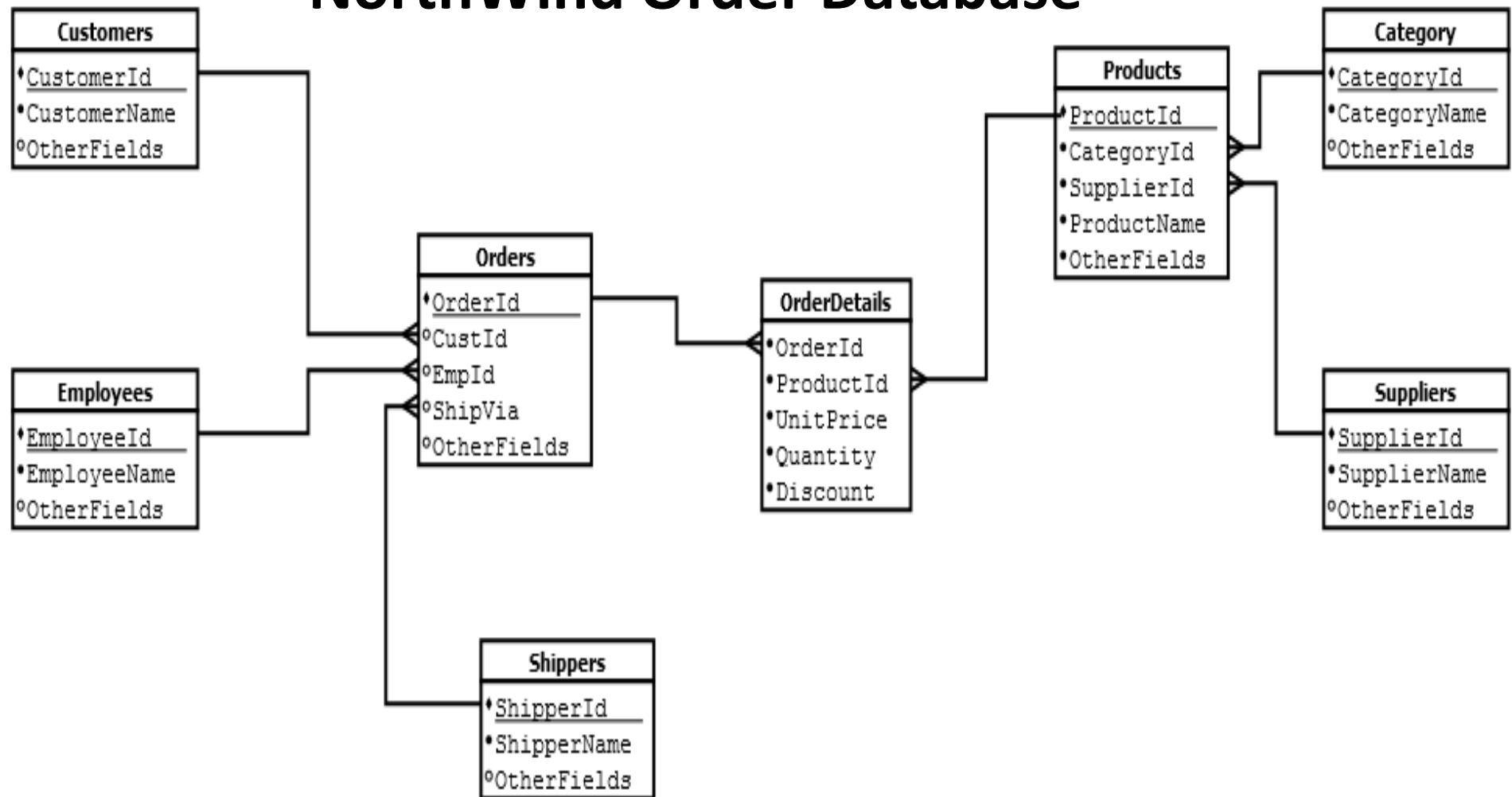
S001 Adam 30 Bldg Name, Road, City State, Country, Pin

Table 1 Name, Age, Address, Road

Table 2 City State, Country, Pin

Data Preparation [Snowflake Schema ➔ Star Schema]

NorthWind Order Database



Measures (Also Referred To As Facts)

- Fields Which Contain Numeric Values
- Reside In Fact Tables
- Calculations Made Over Number Of Records In A Data Set
- Measures Get Aggregated ... Sum, Count, Min, Max, Etc
- Measure Alone Cannot Provide Insight With Point Of Reference Or Contextual Reference
- A Measure Is A Property On Which Calculations & Aggregations (Sum, Count, Average, Minimum, Maximum) Can Be Made
- For Example If Retail Store Sold A Specific Product, The Quantity And Prices Of Each Item Sold Could Be Added Or Averaged To Find The Total Number Of Items Sold And Total Or Average Price Of The Goods Sold

Dimensions

- Categories / Classification Of Data
- Reside In Dimension Tables
- Dimensions Provide Structured Labeling Information To Otherwise Unordered Numeric Measures
- A Dimension Is A Structure That Categorizes Measures To Enable Users To Answer Business Questions
- A Dimension Is A Dataset Composed Of Individual, Non-overlapping Data Elements
- The Primary Functions Of Dimensions Are Threefold:
 - Labelling
 - Filtering
 - Grouping
- Commonly Used Dimensions Are People, Products, Place, Time, Category or Classification
- Contains Textual Description To Provide Context To The Measures
 - Quantity Sold By Department X
 - Revenue By Product A
- Measures Can Aggregated For Each Dimension
 - Quantity Sold By Each Department
 - Revenue By Each Product

Dimension Considerations

- Date / Time
- Big Dimensions
- Small Dimensions
- One Dimension or Two
- Dimensional Roles
- Degenerate Dimensions
- Slowly Changing Dimensions

All Data Structures In The ETL, Including Flat Files, XML Data Sets, And Entity-relation Schemas, We Transform The Structures Into Dimensional Schemas To Prepare For The Final Data-presentation Step

Dimension Tables

- Dimension Tables Provide The Context For Fact Tables
- Dimension Tables Are Usually Much Smaller Than Fact Tables
- Dimension Tables Are Crux Of The Data Warehouse
- They Provide Entry Points To Listing / Filtering / Grouping Data
- A Data Warehouse Is Only As Good As Its Dimensions

Fact Tables

- Every Fact Table Is Defined By The Measures
- The Fact Table **Must State How The Measurement** Is Taken In The Physical World
- All Fact Tables Possess **A Set Of Foreign Keys Connected To The Dimensions** That Provide The Context Of The Fact Table Measurements
- Most Fact Tables Also **Possess One Or More Numerical Measurement Fields**, Which We Call Facts
- Fact Tables Almost Always Have At Least Three Dimensions, But Most Fact Tables Have More
- Virtually Every Fact Table Has **A Primary Key Defined By A Subset Of The Fields In The Table**

Star Scheme Compatibility

- Dimension tables should be denormalized flat tables.
- All **hierarchies & normalized structures** that may be present in earlier staging tables should be flattened
- All attributes in a dimension must take on a single value in the presence of the dimension's primary key.
- If all the proper data relationships have been enforced in the data-cleaning step, these relationships are preserved perfectly in the flattened dimension table.

Basic Structure

- Every Fact Table Is Defined By The Grain (Functionality) Of The Table
- The Grain Of The Fact Table Is The Definition Of The Measurement Event
- The Grain Of The Fact Table Shows How The Measurement Is Taken In The Physical World.
- This Grain Can Be Expressed As Dimension Foreign Keys And Possibly Other Fields In The Fact Table
- All Fact Tables Possess A Set Of Foreign Keys Connected To The Dimensions That Provide The Context Of The Fact Table Measurements
- Most Fact Tables Also Possess One Or More Numerical Measurement Fields, Which We Call Facts
- Some Fact Tables Possess One Or More Special Dimension Like Fields Known As Degenerate Dimensions. Degenerate Dimensions Exist In The Fact Table, But They Are Not Foreign Keys
- Types Of Fact Tables
 - Transactional
 - Snapshot

Guaranteeing Referential Integrity

- No Fact Table Record Should Contains Corrupt Or Unknown Foreign Key References
- There Are Only Two Ways To Violate Referential Integrity In A Dimensional Schema:
 - Load A Fact Record With One Or More **Bad Foreign Keys**
 - **Delete A Dimension Record** Whose Primary Key Is Being Used In The Fact Table
- Areas Where Referential Integrity Can Be Enforced:
 - **Just Before Loading The Fact** Table Records Into The Final Tables, Coupled With Deleting Any Dimension Records
 - Enforcement Of Referential Integrity **In The Database Itself** At The Moment Of Every Fact Table Insertion And Every Dimension Table Deletion
 - Discovery And Correction Of Referential Integrity Violations After Loading Has Occurred By Regularly Scanning The Fact Table, **Looking For Bad Foreign Keys**

Loading Data

Preparation For Loading Fact Tables

- Managing Indexes
- Managing Partitions
- Rollback Log

Loading the Data

- Inserting Facts
- Updating and Correcting Facts
 - Negating Facts
 - Updating Facts
 - Deleting Facts
- Incremental Loading

Structural Modifications

Structural Modifications:

- Adding A Fact To An Existing Fact Table
- Adding A Dimension To An Existing Fact Table
- Adding An Attribute To An Existing Dimension
- Increasing The Granularity Of Existing Fact Tables

Aggregations

- The single most dramatic way to affect performance in a large data warehouse is to provide a proper set of aggregate (summary) records that coexist with the primary base records.
- Aggregates can have a very significant effect on performance, in some cases speeding queries by a factor of a hundred or even a thousand.
- No other means exist to harvest such spectacular gains.
- IT owners of a data warehouse should exhaust the potential for performance gains with aggregates before investing in major new hardware purchases.
- The benefits of a comprehensive aggregate-building program can be realized with almost every data warehouse

Design Requirements

- Aggregates Must Be Stored In Their Own Fact Tables, Separate From Base-level Data.
Each Distinct Aggregation Level Must Occupy Its Own Unique Fact Table
- Aggregate Fact Tables Must Be Same Or Shrunk (Less Columns) Versions Of The Base Fact Table
Some Dimensions Should Be Removed. Definitely No New Dimensions Added
- The Base Fact Table And All Its Related Aggregate Fact Tables Can Be Associated Together As A Family
Generally Done Using A Common Prefix To The Base Fact & Aggregated Fact Tables
- End User Applications Must Refer Exclusively To The Base Fact Table
Never The Aggregated Fact Tables

Thank you!

Contact:

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