IT Real-Time training that work for your career.

PROVIDED TRAINING FOR THOUSANDS OF STUDENTS.

SUBJECT, MATERIAL & VIDEOS

VIDEOS PROVIDED

Trainings:

CLASS ROOM

ONLINE

FAST TRACK

ONE ON ONE

PROJECT TRAINING

DATA MODEL

- Business Process, Conceptual, Logical, Physical Model
- Entity, Relationship, Attribute, and Tuple
- Relationships [1:1, 1:Many, Many: Many]
- Models [Normalized model and Dimensional Model]
- Dimension Table and Types, Fact Table and Types
- Surrogate key and usages
- Star, Snow flake, Galaxy, and Mixed (hybrid)schema
- Active and Inactive relationships
- Single and Bidirectional Relationships

Address:

Flat No: 506/B
Nilgiri Block
Aditya Enclave
Mytrivanam Area
Hyderabad.

Website & Blog

www.vinaytechhouse.com
www.msbivinay.blogspot.in

Contact Information

+91 9573168449
040 66638869

MSBI

IS, AS, RS & MDS

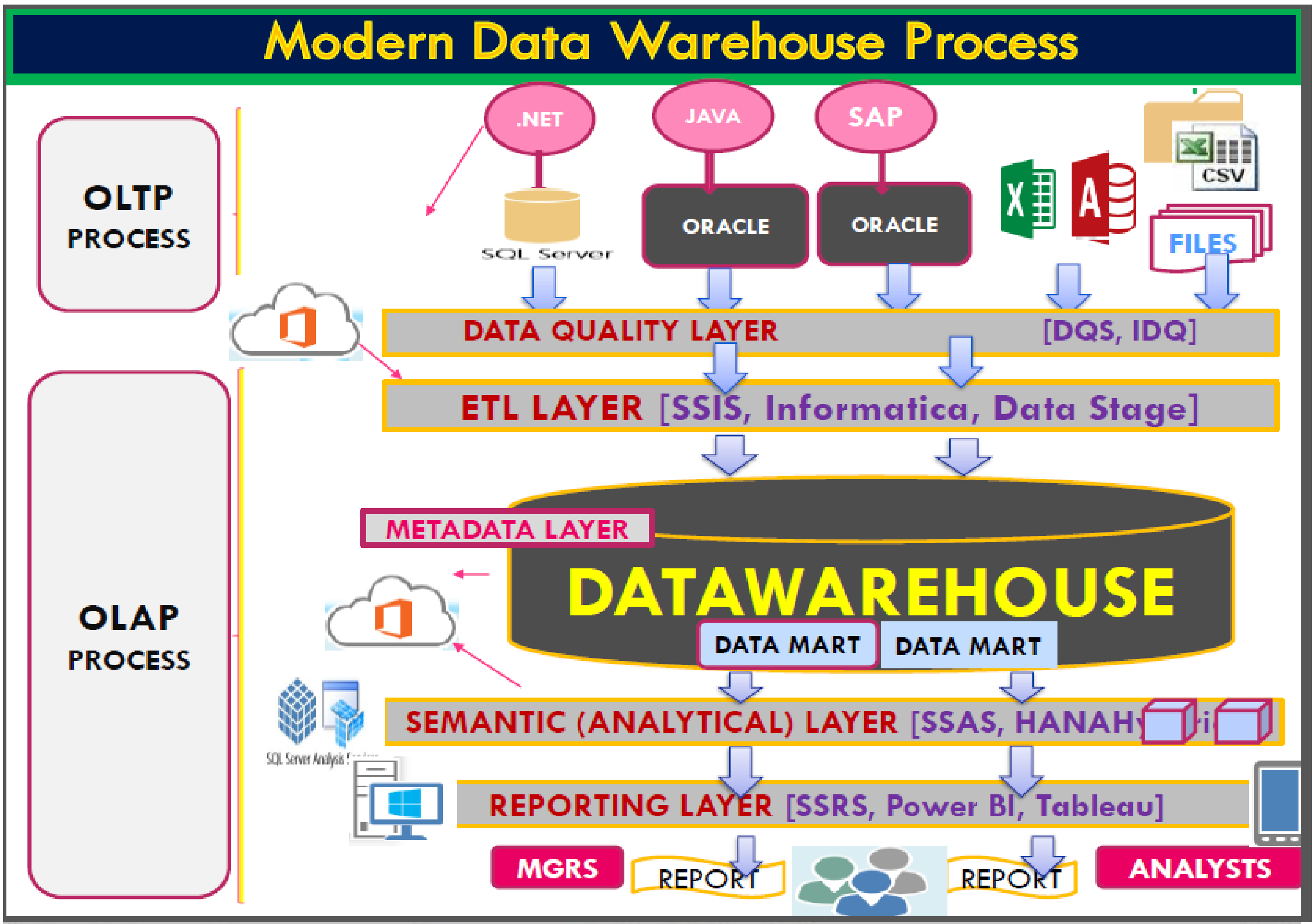
POWER BI

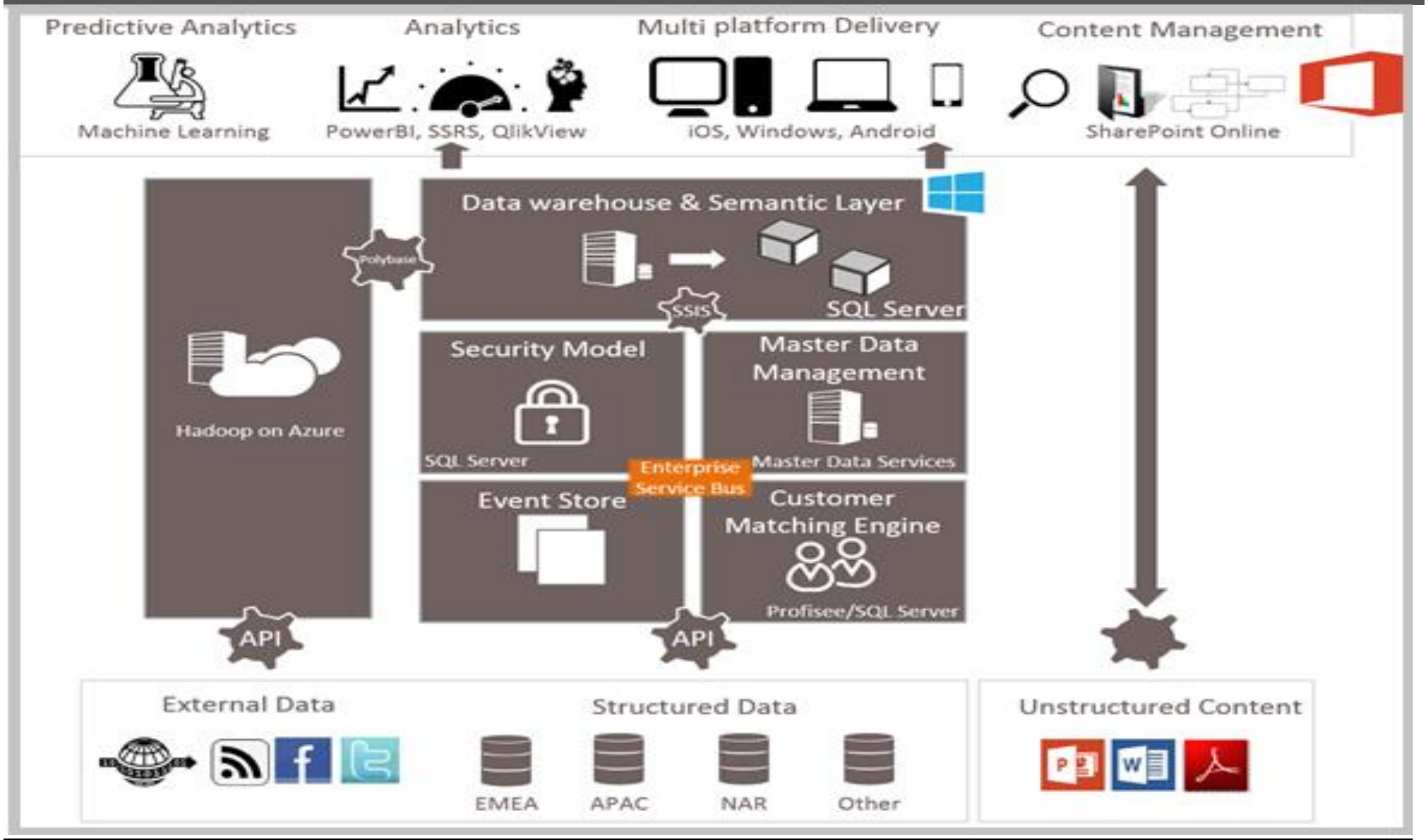
SERVER, DESKTOP & DAX

Sri VINAY TechHouse

WE'VE WORKED WITH A DIVERSE CUSTOMER BASE. HOW CAN WE HELP YOU?

IT Training, Support and Consulting.


Modern Data Warehouse Process
OLTP PROCESS
OLAP PROCESS
.NET
JAVA
SAP
SQL Server
ORACLE
ORACLE
CSV
FILES
DATA QUALITY LAYER [DQS, IDQ]
ETL LAYER [SSIS, Informatica, Data Stage]
METADATA LAYER
DATAWAREHOUSE
DATA MART
DATA MART
SEMANTIC (ANALYTICAL) LAYER [SSAS, HANAH
SQL Server Analysis
REPORTING LAYER [SSRS, Power BI, Tableau]
MGRS
REPORT
REPORT
ANALYSTS


Predictive Analytics
Machine Learning
Analytics
PowerBI, SSRS, QlikView
Multi platform Delivery
iOS, Windows, Android
Content Management
SharePoint Online
Data warehouse & Semantic Layer
Polybase
SSIS
SQL Server
Hadoop on Azure
Security Model
SQL Server
Master Data Management
Master Data Services
Enterprise Service Bus
Event Store
Customer Matching Engine
Profisee/SQL Server
API
API
External Data
Structured Data
EMEA
APAC
NAR
Other
Unstructured Content

DATA MODEL AND ESTABLISHING A PROPER MODEL

What is Data Model?

Presents the relationship between objects / tables / queries.

What is the need of a proper data model?

Proper relationships always provide valid data.

What are the basic advantages of data modeling in Power BI / MSBI/ AZURE BI?

a) Report accuracy

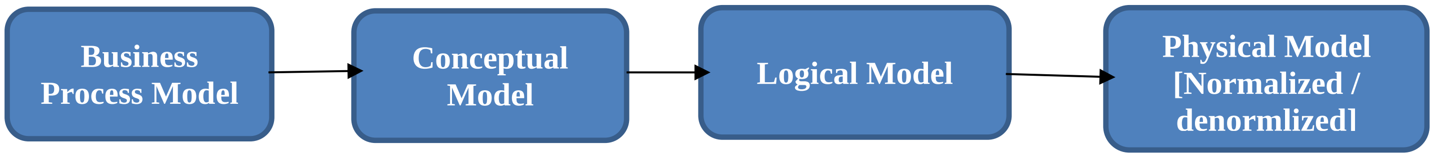
b) Multiple data sources data {files, tables, cloud sources etc. you can use in

A single view. Means an excel data can relate to a flat file, a flat can relate to a table with different cardinalities to use all in a single visual or report level.

OLTP SYSTEM	**OLAP SYSTEM**
Transactional processing	Decision making / analytical processing
Used to run the business	Used to analyze the business
	We should update using schedule
Volatile data (read, write Data is fresh / real-time**, modify, delete)**	**Nonvolatile data (read)**
Current / fresh data only	Historical data (years and years data)
More audiences / transaction audience	Limited audience [executive to management team]
Small to large database [**MBs-GBs**] [1 GB=1024 MB]	Large to very large database [**TBs –PBs**] [1 PB= 1024 TB]
Join time is more [because of more tables]	Joining time is less because of less tables
Physical Database models: **Normalized model** **[more tables, small tables]** **Basic relationships:** **a) 1:1 b) 1 : Many c) Many : Many** **Terminology:** **Entity, Relationship, table, primary key, Foreign key etc.**	**Physical Database models:** **Denormalized model** **[Less tables, Huge volume tables]** **Basic relationships:** **Star schema, Snow flake schema, Galaxy schema, Hybrid schema etc.** **Terminology:** **Dimension, Fact, surrogate key, composite key, role playing, etc.**

Vinay Tech House

Types of models [Business End-End Models]



A general understanding to the three data models is that business analyst uses a conceptual and logical model to model the business objects exist in the system, while database designer or database engineer elaborates the conceptual and logical ER model to produce the physical model that presents the physical database structure ready for database creation. The table below shows the difference between the three data models.

Conceptual model vs Logical model vs Physical model:

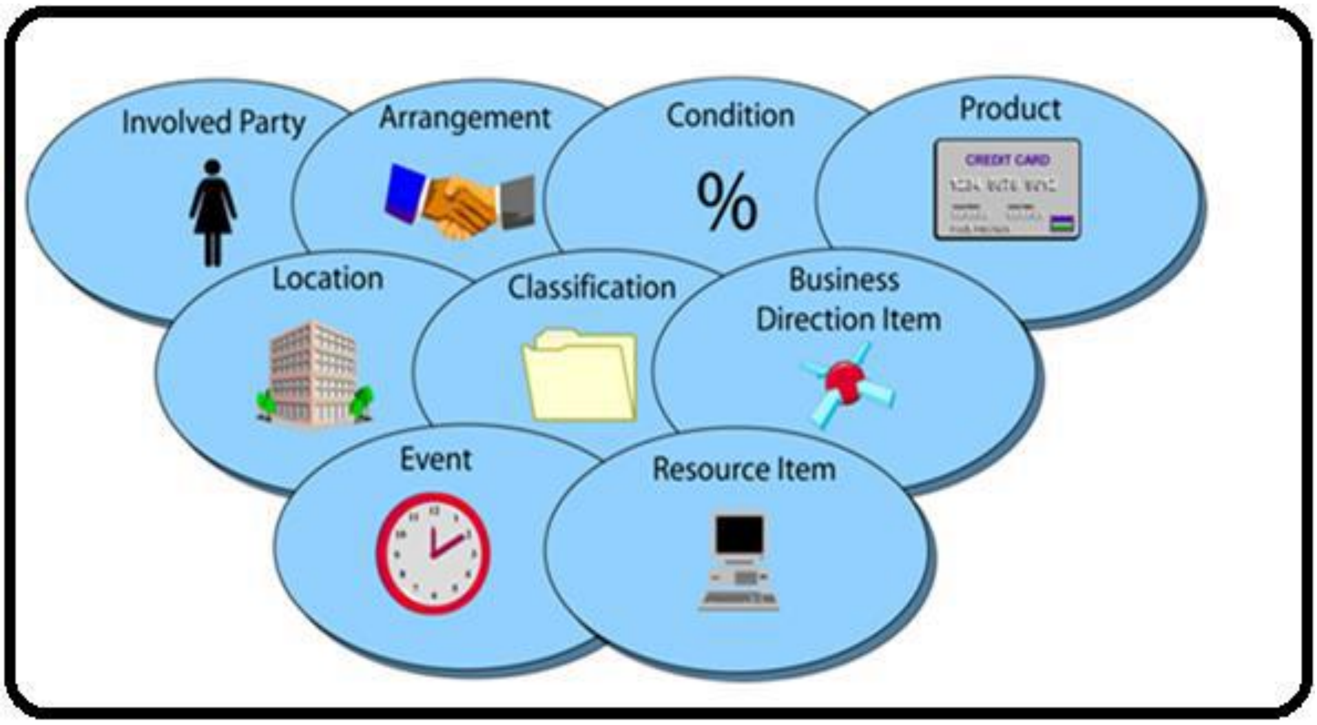
ERD features	Conceptual	Logical	Physical
Entity (Name)	Yes	Yes	Yes
Relationship	Yes	Yes	Yes
Columns		Yes	Yes
Column's Types		Optional	Yes
Primary Key			Yes
Foreign Key			Yes

Conceptual data model

Conceptual ERD models the **business objects that should exist in a system and the relationships between them**. **A conceptual model is developed to present an overall picture of the system** by recognizing the business objects involved. It defines what entities exist, NOT which tables. For example, 'many to many' tables may exist in a logical or physical data model but they are just shown as a relationship with no cardinality under the conceptual data model.

Conceptual data model example

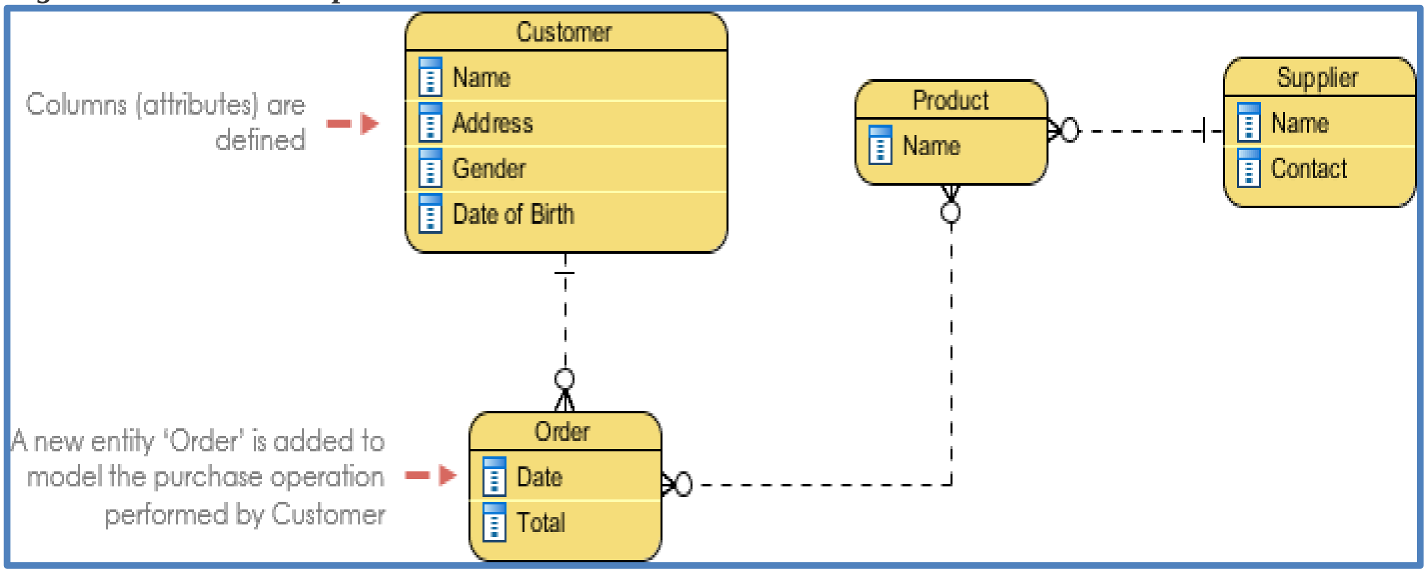
NOTE: Conceptual ERD supports the use of generalization in modeling the 'a kind of' relationship between two entities, for instance, Triangle, is a kind of Shape. The usage is like generalization in UML. Notice that only conceptual ERD supports generalization.



Logical data model

Logical ERD is a **detailed version of a Conceptual ERD**. A logical ER model is developed to enrich a conceptual model by defining explicitly the columns in each entity and introducing operational and transactional entities. Although a logical data model is still independent of the actual database system in which the database will be created, you can still take that into consideration if it affects the design.

Logical data model example

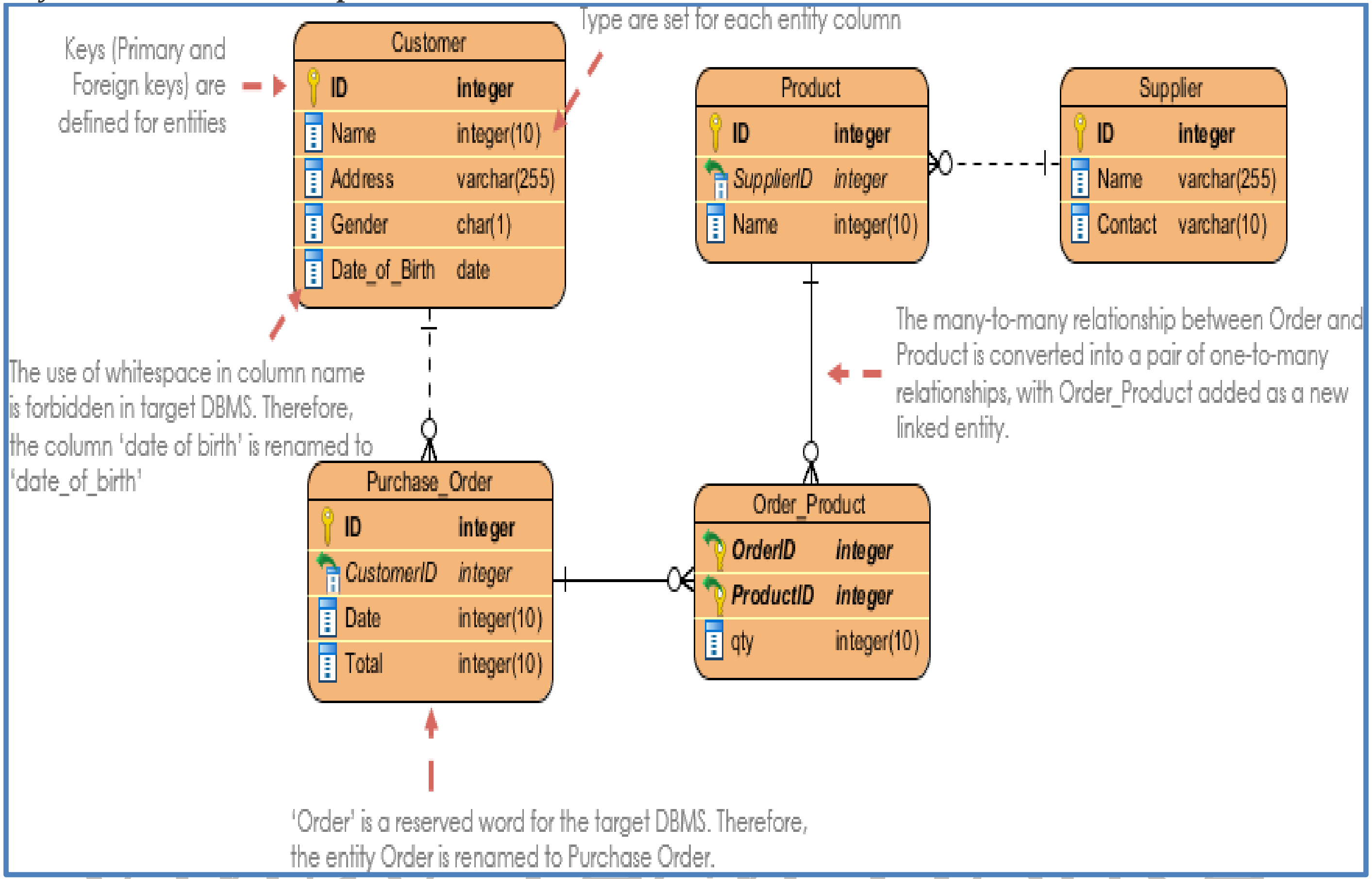

Columns (attributes) are defined
Customer
Name
Address
Gender
Date of Birth
Product
Name
Supplier
Name
Contact
A new entity 'Order' is added to model the purchase operation performed by Customer
Order
Date
Total

NOTE: MANY –MANY LOGICALLY MENTIONED BETWEEN ORDER AND PRODUCT

Physical data model

Physical ERD represents the **actual design blueprint of a relational database**. A physical data model elaborates on the logical data model by assigning each column with type, length, nullable, etc. Since a physical ERD represents how data should be structured and related in a specific DBMS it is important to consider the convention and restriction of the actual database system in which the database will be created. Make sure the column types are supported by the DBMS and reserved words are not used in naming entities and columns.

Physical data model example


Keys (Primary and Foreign keys) are defined for entities
Type are set for each entity column
Customer
ID integer
Name integer(10)
Address varchar(255)
Gender char(1)
Date_of_Birth date
Product
ID integer
SupplierID integer
Name integer(10)
Supplier
ID integer
Name varchar(255)
Contact varchar(10)
The many-to-many relationship between Order and Product is converted into a pair of one-to-many relationships, with Order_Product added as a new linked entity.
The use of whitespace in column name is forbidden in target DBMS. Therefore, the column 'date of birth' is renamed to 'date_of_birth'
Purchase_Order
ID integer
CustomerID integer
Date integer(10)
Total integer(10)
Order_Product
OrderID integer
ProductID integer
qty integer(10)
'Order' is a reserved word for the target DBMS. Therefore, the entity Order is renamed to Purchase Order.

Vinay Tech House

NOTE: MANY –MANY PHYSICALLY MENTIONED BETWEEN ORDER AND PRODUCT WITH ONE BRIDGE OR INTERMEDIATE TABLE AND TWO 1-MANY RELATIONSHIPS.

Two different types of physical data models:

Transactional systems model [OLTP]	**Analytical systems model [OLAP]**
Normalization model	Denormalization model [Dimensional] model
Less duplicates, More tables, Small sized tables, Aggregates less	More duplicates, Less tables, Complex sized tables, aggregates more
Normal forms [Form 1- Form 6]	Schemas [Star, Snow Flake, Galaxy, Mixed etc.]
Relationships [**1:1,1:*,*:***] One-One, One-Many, Many-Many	The above schemas in **many-many** relationships.
All in single direction	Modern OLAP applications having these relationships in **both the directions**
Entity, Tuple, Attribute, Primary Key, Foreign Key etc…	Dimension, Fact, Measure, Measure group, Primary Key, Foreign Key, Surrogate Key or BI Key etc…

OLTP Terminology:

Entity: A row / table

Tuple: Collections of columns / row

Attribute: Column

Primary Key: Which take only unique values

Foreign Key: One place it is unique and another place is it normal, then it is foreign key

Dimension: Textual attribute. [Does not support aggregations {sum, avg, min, etc.]

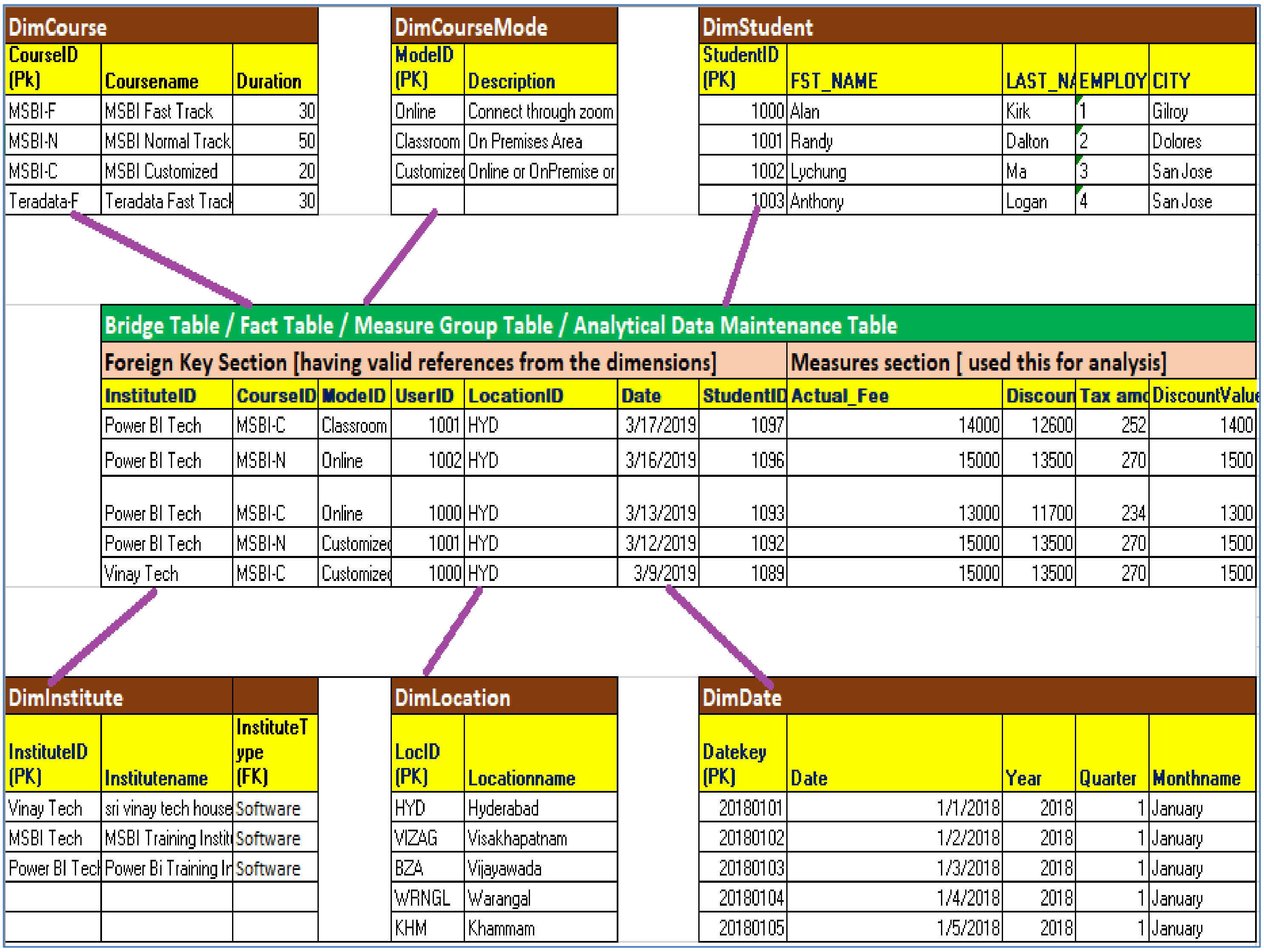
Fact: Measurable attribute. [Support aggregations]

SalesID[Numeric]: 1000,2000, 3000 --Never used for aggregations, **so dimension**

Salesincome[Numeric]: 1000, 2000, 3000 –Support aggregations, **so measure / fact**

Vinay Tech House

VINAY TECH BUSINESS DETAILS DWH DB DIAGRAM



DimCourse

CourseID (Pk)	Coursename	Duration
MSBI-F	MSBI Fast Track	30
MSBI-N	MSBI Normal Track	50
MSBI-C	MSBI Customized	20
Teradata-F	Teradata Fast Track	30

DimCourseMode

ModeID (PK)	Description
Online	Connect through zoom
Classroom	On Premises Area
Customized	Online or OnPremise or

DimStudent

StudentID (PK)	FST_NAME	LAST_NA	EMPLOY	CITY
1000	Alan	Kirk	1	Gilroy
1001	Randy	Dalton	2	Dolores
1002	Lychung	Ma	3	San Jose
1003	Anthony	Logan	4	San Jose

Bridge Table / Fact Table / Measure Group Table / Analytical Data Maintenance Table

Foreign Key Section [having valid references from the dimensions]							Measures section [used this for analysis]			
InstituteID	CourseID	ModeID	UserID	LocationID	Date	StudentID	Actual_Fee	Discoun	Tax amo	DiscountValue
Power BI Tech	MSBI-C	Classroom	1001	HYD	3/17/2019	1097	14000	12600	252	1400
Power BI Tech	MSBI-N	Online	1002	HYD	3/16/2019	1096	15000	13500	270	1500
Power BI Tech	MSBI-C	Online	1000	HYD	3/13/2019	1093	13000	11700	234	1300
Power BI Tech	MSBI-N	Customized	1001	HYD	3/12/2019	1092	15000	13500	270	1500
Vinay Tech	MSBI-C	Customized	1000	HYD	3/9/2019	1089	15000	13500	270	1500

DimInstitute

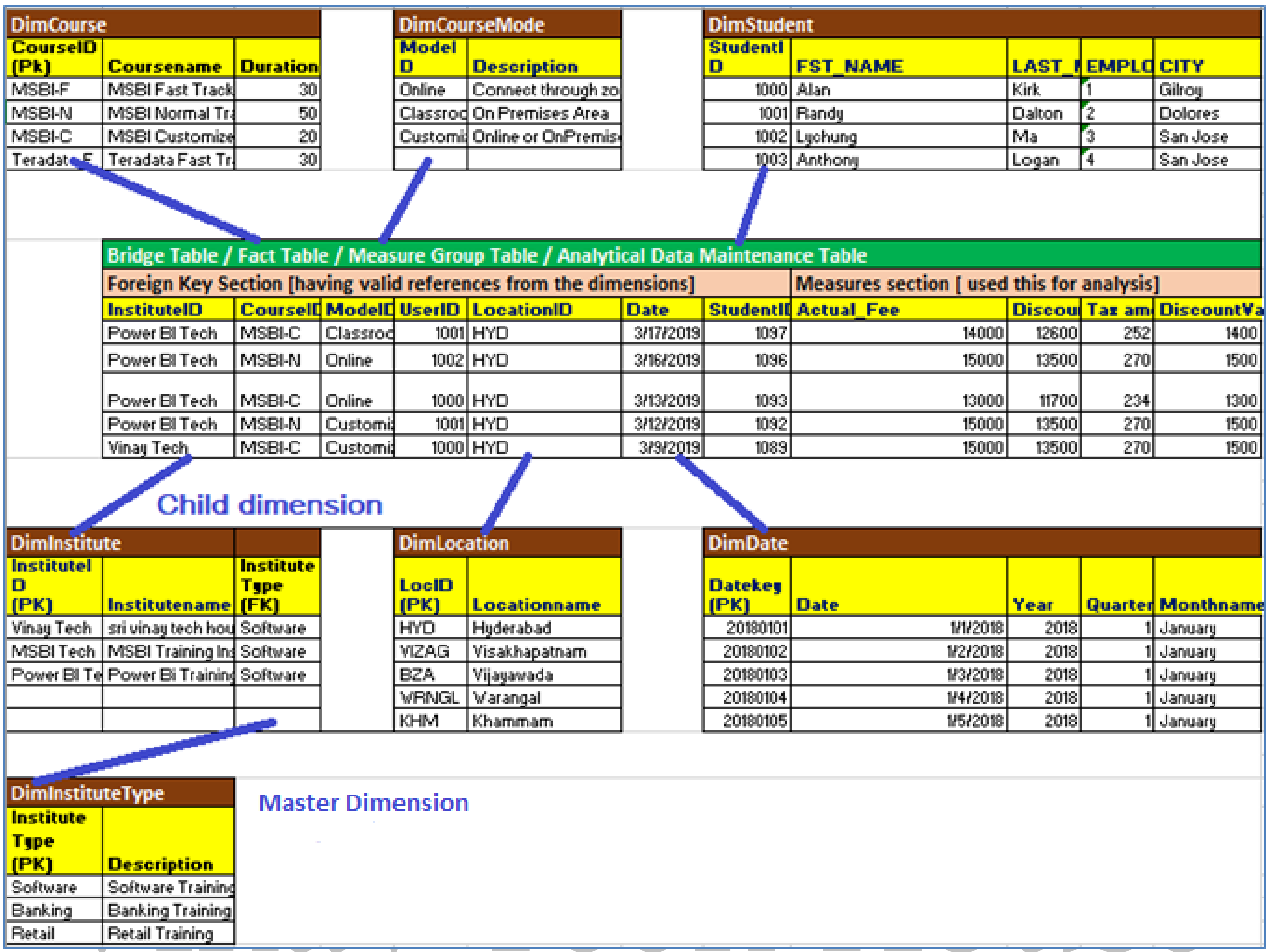
InstituteID (PK)	Institutename	InstituteType (FK)
Vinay Tech	sri vinay tech house	Software
MSBI Tech	MSBI Training Instit	Software
Power BI Tech	Power Bi Training In	Software

DimLocation

LocID (PK)	Locationname
HYD	Hyderabad
VIZAG	Visakhapatnam
BZA	Vijayawada
WRNGL	Warangal
KHM	Khammam

DimDate

Datekey (PK)	Date	Year	Quarter	Monthname
20180101	1/1/2018	2018	1	January
20180102	1/2/2018	2018	1	January
20180103	1/3/2018	2018	1	January
20180104	1/4/2018	2018	1	January
20180105	1/5/2018	2018	1	January



DimCourse		
CourseID (Pk)	Coursename	Duration
MSBI-F	MSBI Fast Track	30
MSBI-N	MSBI Normal Tr	50
MSBI-C	MSBI Customize	20
Teradata-F	Teradata Fast Tr	30

DimCourseMode	
ModelD	Description
Online	Connect through zo
Classroo	On Premises Area
Customi	Online or OnPremis

DimStudent				
StudentID	FST_NAME	LAST_N	EMPLO	CITY
1000	Alan	Kirk	1	Gilroy
1001	Randy	Dalton	2	Dolores
1002	Lychung	Ma	3	San Jose
1003	Anthony	Logan	4	San Jose

Bridge Table / Fact Table / Measure Group Table / Analytical Data Maintenance Table										
Foreign Key Section [having valid references from the dimensions]							Measures section [used this for analysis]			
InstituteID	CourseID	ModelD	UserID	LocationID	Date	StudentID	Actual_Fee	Discou	Tax am	DiscountVa
Power BI Tech	MSBI-C	Classroo	1001	HYD	3/17/2019	1097	14000	12600	252	1400
Power BI Tech	MSBI-N	Online	1002	HYD	3/16/2019	1096	15000	13500	270	1500
Power BI Tech	MSBI-C	Online	1000	HYD	3/13/2019	1093	13000	11700	234	1300
Power BI Tech	MSBI-N	Customi	1001	HYD	3/12/2019	1092	15000	13500	270	1500
Vinay Tech	MSBI-C	Customi	1000	HYD	3/9/2019	1089	15000	13500	270	1500

DimInstitute		
InstituteID (PK)	Institutename	Institute Type (FK)
Vinay Tech	sri vinay tech hou	Software
MSBI Tech	MSBI Training Ins	Software
Power BI Te	Power Bi Training	Software

DimLocation	
LocID (PK)	Locationname
HYD	Hyderabad
VIZAG	Visakhapatnam
BZA	Vijayawada
VRNGL	Warangal
KHM	Khammam

DimDate				
Datekey (PK)	Date	Year	Quarter	Monthname
20180101	1/1/2018	2018	1	January
20180102	1/2/2018	2018	1	January
20180103	1/3/2018	2018	1	January
20180104	1/4/2018	2018	1	January
20180105	1/5/2018	2018	1	January

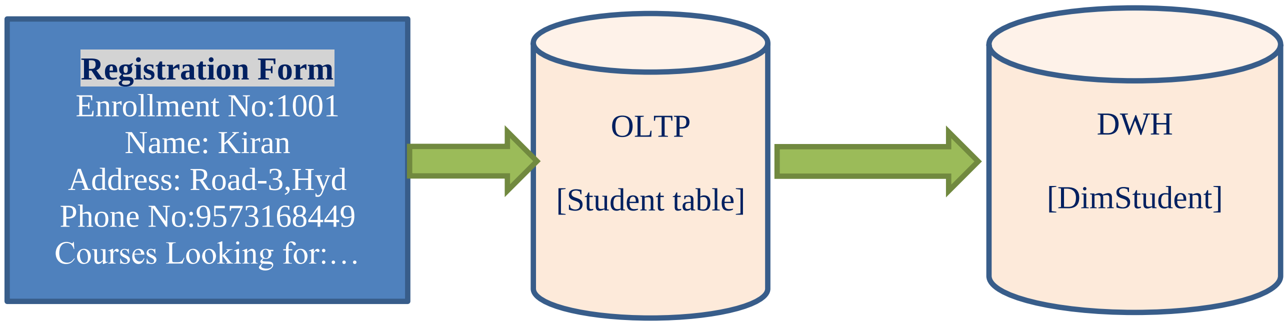
DimInstituteType	
Institute Type (PK)	Description
Software	Software Training
Banking	Banking Training
Retail	Retail Training

Data Loading into Dimension and Facts:

a) Dimension Loading is Direct Loading

Example:

Today there are four students registered. Then four entries will go to DimStudent table.



b) Fact Loading is two-step process

Step-1) Dimensions validation

Step-2) Loading data into measures

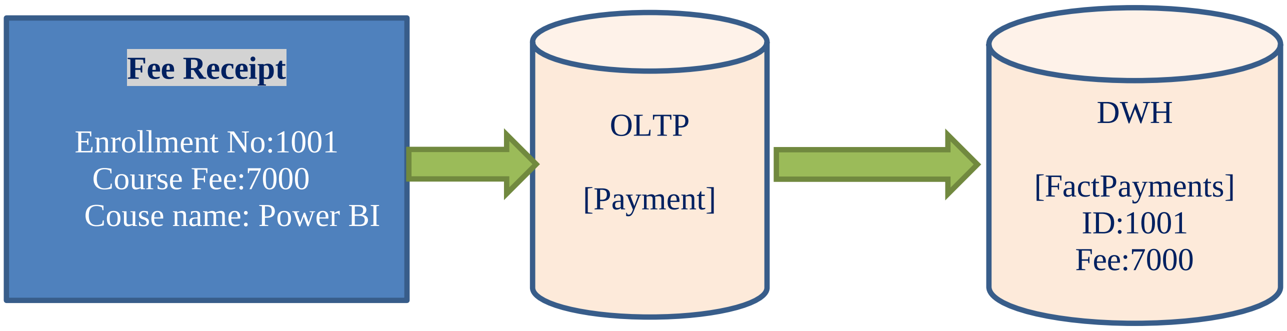
Example:

In the above four students, 3 students took multiple courses and paid course fee.

Step-1) Then 3 students valid or not verified in the dimension tables [DimStudent]

Step-2) Then course fee values loaded to fact table.

Vinay Tech House



```
Create table dimstudent
(studentid integer primary key, sname varchar(30), address
varchar(30))

create table factpayments
(studentid integer references dimstudent(studentid), fee
decimal(9,2))

insert into dimstudent
values(1001,'vinay','hyd'),(1002,'madhu','mum')
--Two people registered in the institute

insert into factpayments values(1001,10000);
--One student joined, so entries loaded to factpayments

insert into factpayments values(1004,11000);
--failed because the student not enrolled
```

Vinay Tech House

Why are we maintaining two different tables (Dimension, Fact)?

If we maintain textual and numerical data in a single table, the below are drawbacks

a) **Table Size increase**
b) **More columns in the table**
c) **Retrieval slow**
d) **Business values history maintenance not possible**
e) **Analytical operations not so easy**

Advantages of two tables with different data

a) **We can get one table with only numerical / measurable values**
b) **Separate measurable values provide faster calculations**
c) **Easy to maintain business values history**

DIMENSION TABLE & FEATURES

DEFINITION:

Dimension is a textual attribute and set of textual attributes available table is dimension table.

FEATURES:

a) Dimension table is **master table**

b) Dimension table **contain Primary key**

c) Dimension table level **surrogate keys highly supported**

d) Dimension table **support hierarchies** [so that **drilldown and drill up** possible]

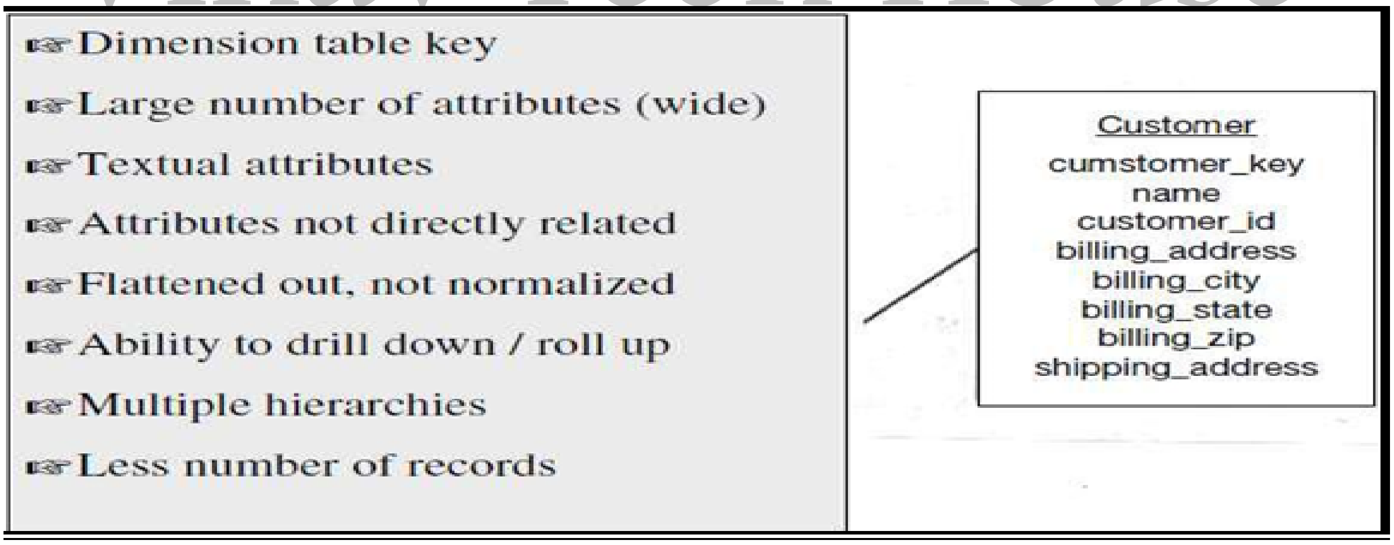
e) **Less number of records** available compared to Fact table

f) More columns available [Wide Table]

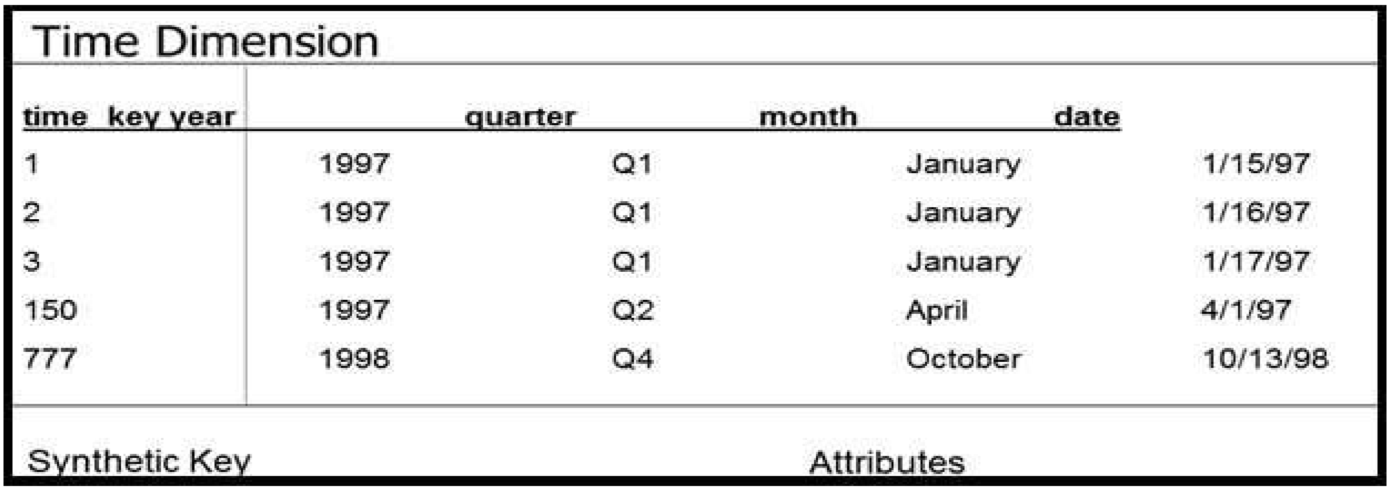
g) Support answers for **What, Where, When etc. textual questions**

h) Rarely **changed data table** [except few situations]

Vinay Tech House



- Dimension table key
- Large number of attributes (wide)
- Textual attributes
- Attributes not directly related
- Flattened out, not normalized
- Ability to drill down / roll up
- Multiple hierarchies
- Less number of records



Time Dimension

time key	year	quarter	month	date
1	1997	Q1	January	1/15/97
2	1997	Q1	January	1/16/97
3	1997	Q1	January	1/17/97
150	1997	Q2	April	4/1/97
777	1998	Q4	October	10/13/98

Synthetic Key Attributes

Attributes not directly related

Frequently you will find that some of the attributes in a dimension table are not directly related to the other attributes in the table.

For example, package size is not directly related to product brand; nevertheless, package size and product brand could both be attributes of the product dimension table.

Vinay Tech House

Not normalized

The attributes in a dimension table are used over and over again in queries. An attribute is taken as a constraint in a query and applied directly to the metrics in the fact table.

For efficient query performance, it is best if the query picks up an attribute from the dimension table and goes directly to the fact table and not through other intermediary tables. If you normalize the dimension table, you will be creating such intermediary tables and that will not be efficient. Therefore, a dimension table is flattened out, not normalized.

Drilling down, rolling up

The attributes in a dimension table provide the ability to get to the details from higher levels of aggregation to lower levels of details.

For example, the three attributes zip, city, and state form a hierarchy. You may get the total sales by state, then drill down to total sales by city, and then by zip. Going the other way, you may first get the totals by zip, and then roll up to totals by city and state.

Multiple hierarchies

In the example of the customer dimension, there is a single hierarchy going up from individual customer to zip, city, and state. But dimension tables often provide for multiple hierarchies, so that drilling down may be performed along any of the multiple hierarchies.

Fewer number of records.

A dimension table typically has fewer number of records or rows than the fact table. A product dimension table for an automaker may have just 500 rows. On the other hand, the fact table may contain millions of rows.

Vinay Tech House

FACT OR MEASURE & MEASURE GROUP OR FACT TABLE

DEFINITION:

Fact is measurable attribute. Set of measurable attributes table is Fact table /Measure Group Table. Many - many relationships providing table.

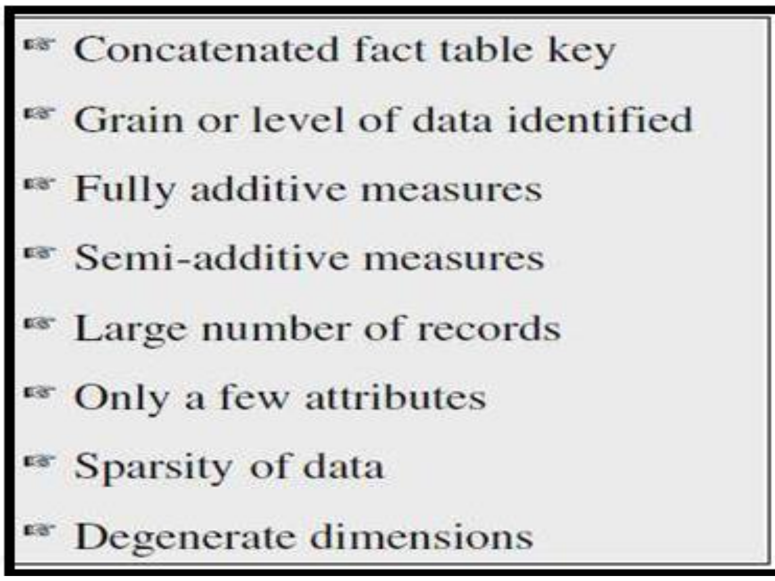
It contains two sections

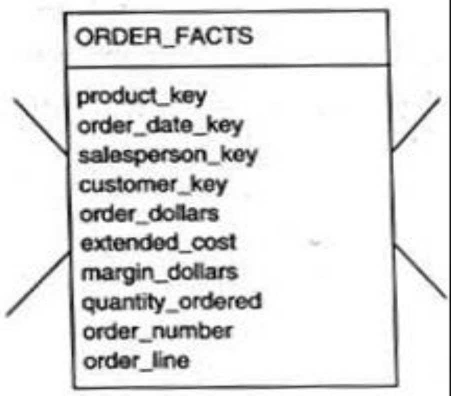
a) Foreign key section b) Measures section

FEATURES:

a) Fact table is **Child table / Transaction / Operational / Business Analysis Table**
b) Fact table contains 1. Foreign key references 2. Measures
c) Fact also **support surrogate keys**
d) Fact table **doesn't support hierarchies** [because of numerical values]
e) **More number of records** available compared to Dimension table
f) **Less columns and more rows available** [**Deep Table**]
g) Support answers for **Top, Bottom, How much, how many etc. questions**
h) Frequently changed data table [from **OLTP data loaded regularly in to the table**]

Vinay Tech House


☞ Concatenated fact table key
☞ Grain or level of data identified
☞ Fully additive measures
☞ Semi-additive measures
☞ Large number of records
☞ Only a few attributes
☞ Sparsity of data
☞ Degenerate dimensions



Concatenated Key(Combination of foreign keys)

A row in the fact table relates to a combination of rows from all the dimension tables.

The primary key of the fact table must be the concatenation of the primary keys of all the dimension tables.

Data Grain

This is an important characteristic of the fact table. As we know, the data grain is the level of detail for the measurements or metrics.

If we keep the quantity ordered as the quantity of a specific product for each month, then the data grain is different and is at a higher level.

Fully Additive Measures.

Let us look at the attributes *order dollars, extended cost,* and *quantity ordered.* Each of these relates to a particular product on a certain date for a specific customer procured by an individual sales representative.

Vinay Tech House

When we run queries to aggregate measures in the fact table, we will have to make sure that these measures are fully additive. Otherwise, the aggregated numbers may not show the correct totals.

Semi additive Measures

Consider the margin dollars attribute in the fact table.

For example, if the *order dollars* is 120 and *extended cost* is 100, the *margin percentage* is 20. This is a calculated metric derived from the *order dollars* and *extended cost.*

Derived attributes such as *margin percentage* are not additive. They are known as semi additive measures. Distinguish semi additive measures from fully additive measures when you perform aggregations in queries.

Table Deep, Not Wide

Typically a fact table contains fewer attributes than a dimension table.

If you lay the fact table out as a two-dimensional table, you will note that the fact table is narrow with a small number of columns, but very deep with a large number of rows.

Sparse Data

It is important to realize this type of sparse data and understand that the fact table could have gaps.

Vinay Tech House

DIFFERENCES BETWEEN DIMENSION AND FACT TABLES

1) Master table / Dimension table Vs. Child table / Fact table / Transaction table

Master table / Dimension table	**Child / transaction / Fact table**
Contains primary key	Contains foreign key
Hold Textual values	More measurable values
No measurable information and does not support for aggregate storage	Aggregate and analytical columns more
More columns	Less columns
Maintain less rows	More rows
Support Hierarchies creation [Country→State→City]	No support
Drill down or drill up analysis possible	Not supported
Answer for what, where, when etc… textual questions	How much, how many etc… numerical questions.
Ex: What is the location details for the locationid 'HYD'	What is the growth rate from previous year to current year?
Rarely changed table {except few business types} Ex: if a new student joined, then studentid specified in Student dimension with student details.	**Frequently changed tables** **{Hourly, daily, weekly etc…}** Ex: If the student joined in multiple intervals multiple courses, then those many rows added to fact table.

Vinay Tech House

TYPES OF DIMENSIONS

Types of dimensions

Various types of dimensions available.

Conformed dimension

Sharable dimension, where the dimension is used by the multiple fact tables.

Ex: Time, Location etc… tables can be shared, so we call them as confirmed dimensions.

Degenerated dimension

This is neither dimension nor fact, but available in fact table.

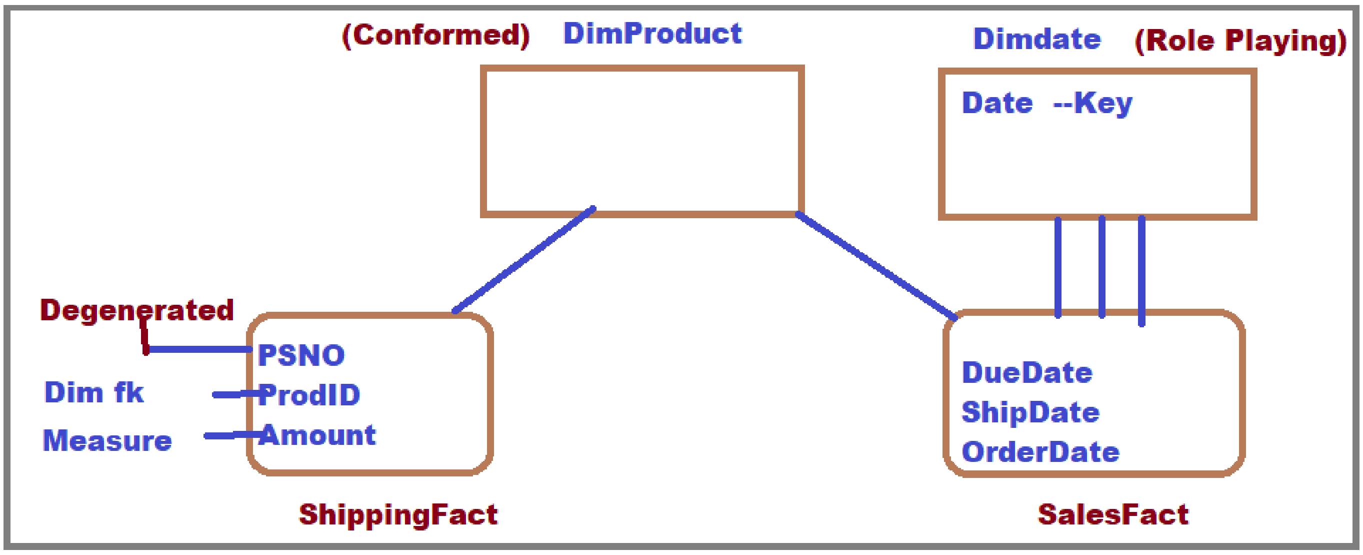
Ex: InvoiceNo, TransactionID, SalesOrderNO etc…

Vinay Tech House

Role playing dimension

If a dimension has multiple foreign keys in the fact table, then it is role playing dimension.

Ex: If DimDate table has Enquiry_Date, Join_Date, CourseStart_Date keys in the fact tanle, then the Date table is Role playing dimension table.



Dirty dimension

If a dimension table has multiple non key values for the same business key and difficult to identify business operation uniquely, then it is dirty dimension.

Ex:

ID,NAME,LOC

1,VINAY,HYD

1,VINAY,MUM

1,VINAY,CHE

Junk Dimension

If a dimension **has non business data and that is used to store status / indicator / flags, or any other kind of information, then it is Junk dimension.**

Ex:

Country and Currency code table→Junk table

Gender information table—>Junk table

Status or flag table→ On/ Off, Current / Expired etc...

Vinay Tech House

Rapidly Changing Dimensions

A dimension attribute that changes frequently is a rapidly changing attribute. If you don't need to track the changes, the rapidly changing attribute is no problem, but if you do need to track the changes, using a standard slowly changing dimension technique can result in a huge inflation of the size of the dimension. One solution is to move the attribute to its own dimension, with a separate foreign key in the fact table. This new dimension is called a rapidly changing dimension.

Slowly Changing Dimensions

Attributes of a dimension that would undergo changes over time. It depends on the business requirement whether particular attribute history of changes should be preserved in the data warehouse. This is called a slowly changing attribute and a dimension containing such an attribute is called a slowly changing dimension.

Inferred Dimensions

While loading fact records, a dimension record may not yet be ready. One solution is to generate a surrogate key with null for all the other attributes. This should technically be called an inferred member, but is often called an inferred dimension.

Shrunken Dimensions

A shrunken dimension is a subset of another dimension. For example, the orders fact table may include a foreign key for product, but the target fact table may include a foreign key only for productcategory, which is in the product table, but much less granular. Creating a smaller dimension table, with productcategory as its primary key, is one way of dealing with this situation of heterogeneous grain. If the product dimension is snowflaked, there is probably already a separate table for productcategory, which can serve as the shrunken dimension.

Static Dimensions

Static dimensions are not extracted from the original data source, but are created within the context of the data warehouse. A static dimension can be loaded manually — for example with status codes — or it can be generated by a procedure, such as a date or time dimension.

Vinay Tech House

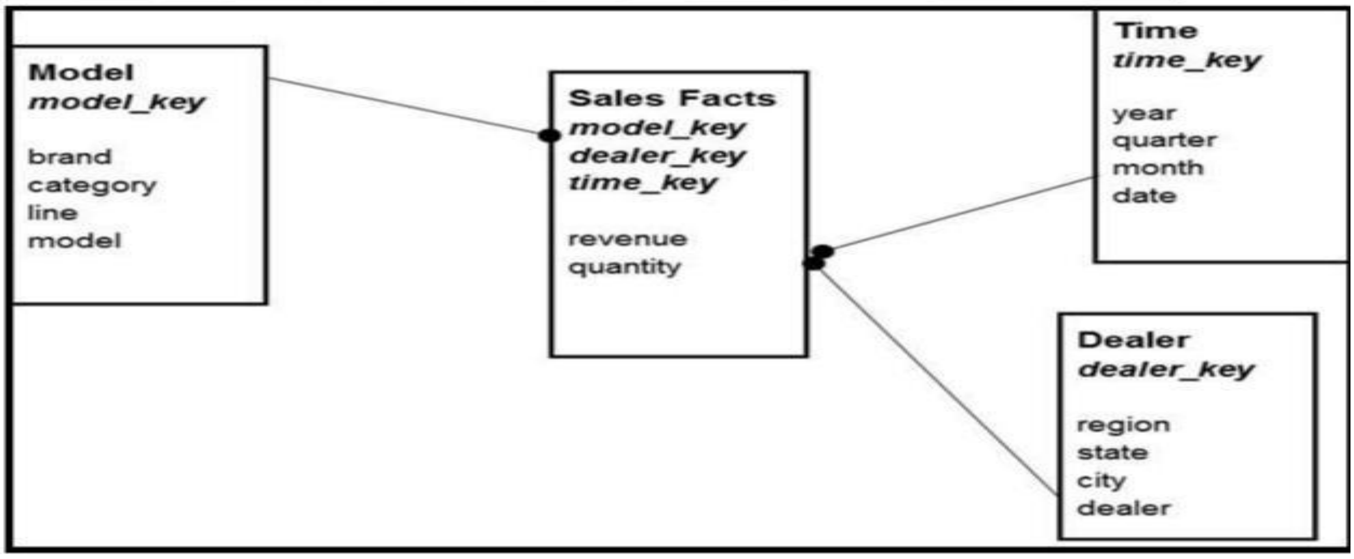
TYPES OF FACTS / MEASURES [Three types of facts]

Type1: Fully additive Facts:

- Can be summed across any and all dimensions
- Stored in fact table
- Examples: revenue, quantity

Tip: Usually SUM, AVG, COUNT etc. come under fully additive measure

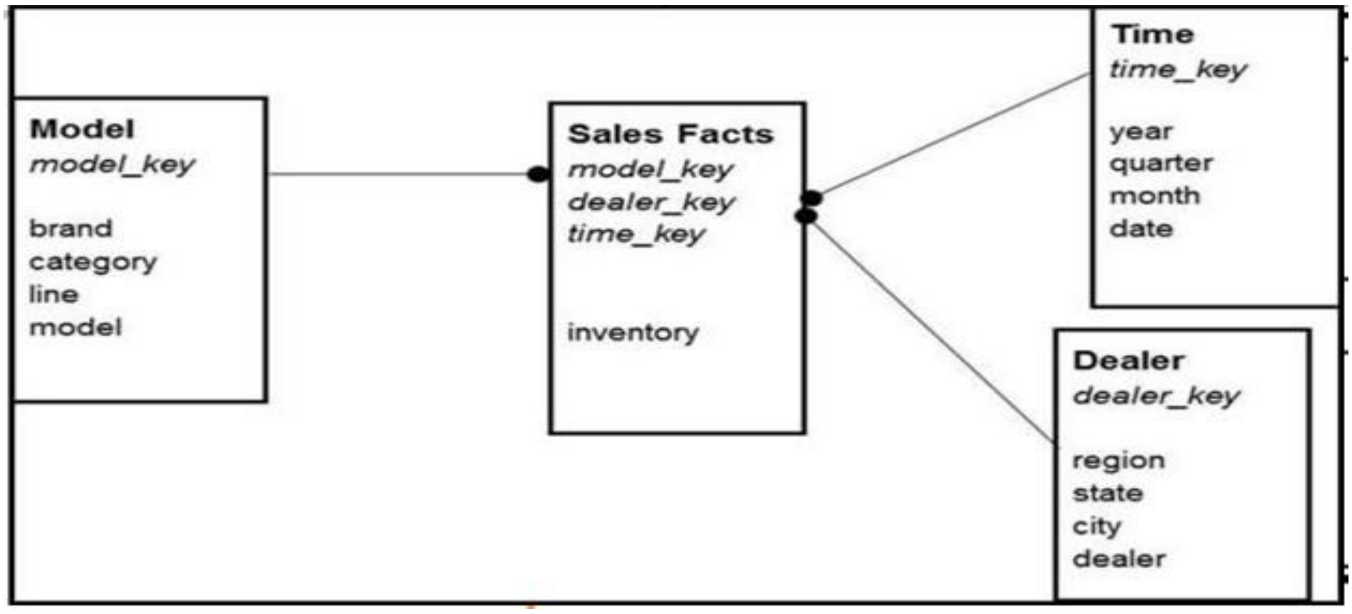
Eg: **Because all values participated in the calculation**



Type2: Semi Additive Facts:

- Can be summed across most dimensions but not all
- Examples: Inventory quantities, account balances, or personnel counts
- Anything that measures a "level" Must be careful with ad-hoc reporting often aggregated across the "forbidden dimension" by averaging

Tip: Usually TOP, BOTTOM, FIRST, LAST etc. come under semi additive measure
Because of few values participated.

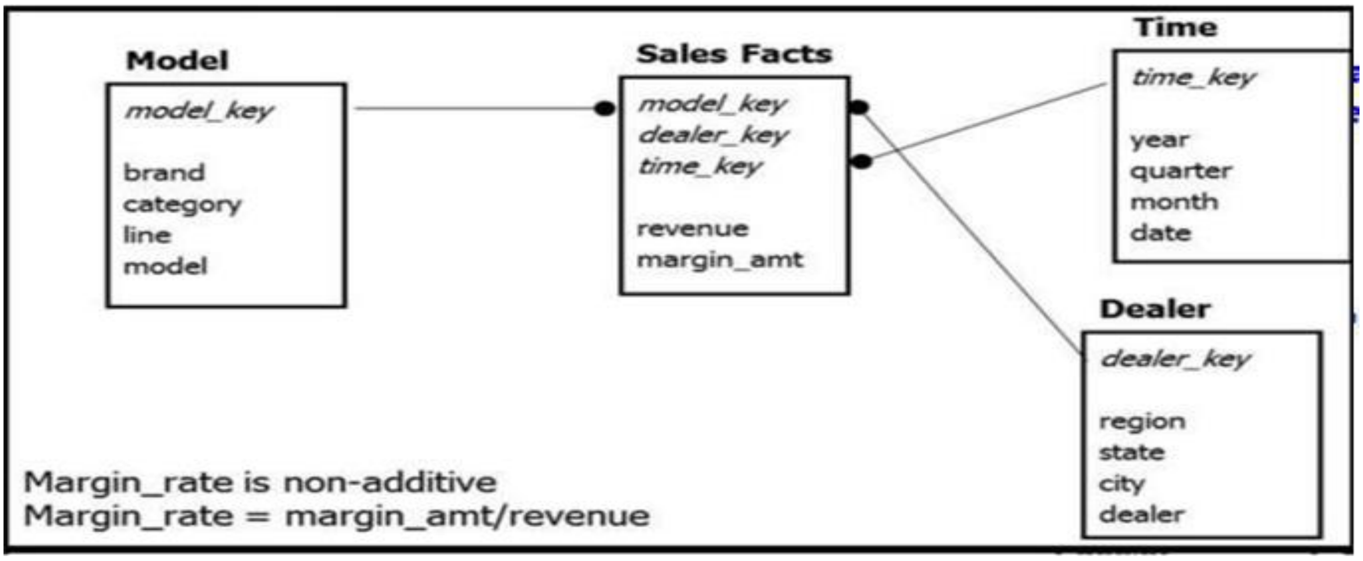


Type 3: Non Additive Facts:

– Cannot be summed across any dimension
– All ratios are non-additive
– Break down to fully additive components, store them in fact table

Vinay Tech House

Tip: Usually Growth, Growth Percentage etc. come under Non additive measure Because existing measures we are using to perform calculation.



Additive

Additive facts are facts that can be summed up through all of the dimensions in the fact table.
Eg: Sales fact

Semi-Additive

Semi-additive facts are facts that can be summed up for some of the dimensions in the fact table, but not the others.
Eg: Daily balances fact can be summed up through the customers dimension but not through the time dimension.

Non-Additive

Non-additive facts are facts that cannot be summed up for any of the dimensions present in the fact table.
Eg: Facts which have percentages, Ratios calculated.

Vinay Tech House

TYPES OF FACT TABLES

The Types of Fact Table are

- Snapshot
- Cumulative
- **Factless Fact Table**

Snapshot

This type of fact table describes the state of things in a particular instance of time, and usually includes more semi-additive and non-additive facts. The second example presented here is a snapshot fact table.

Eg: Daily balances fact can be summed up through the customers dimension but not through the time dimension.

Cumulative

This type of fact table describes what has happened over a period of time. For example, this fact table may describe the total sales by product by store by day. The facts for this type of fact tables are mostly additive facts. The first example presented here is a cumulative fact table.
Eg: Sales fact

Vinay Tech House

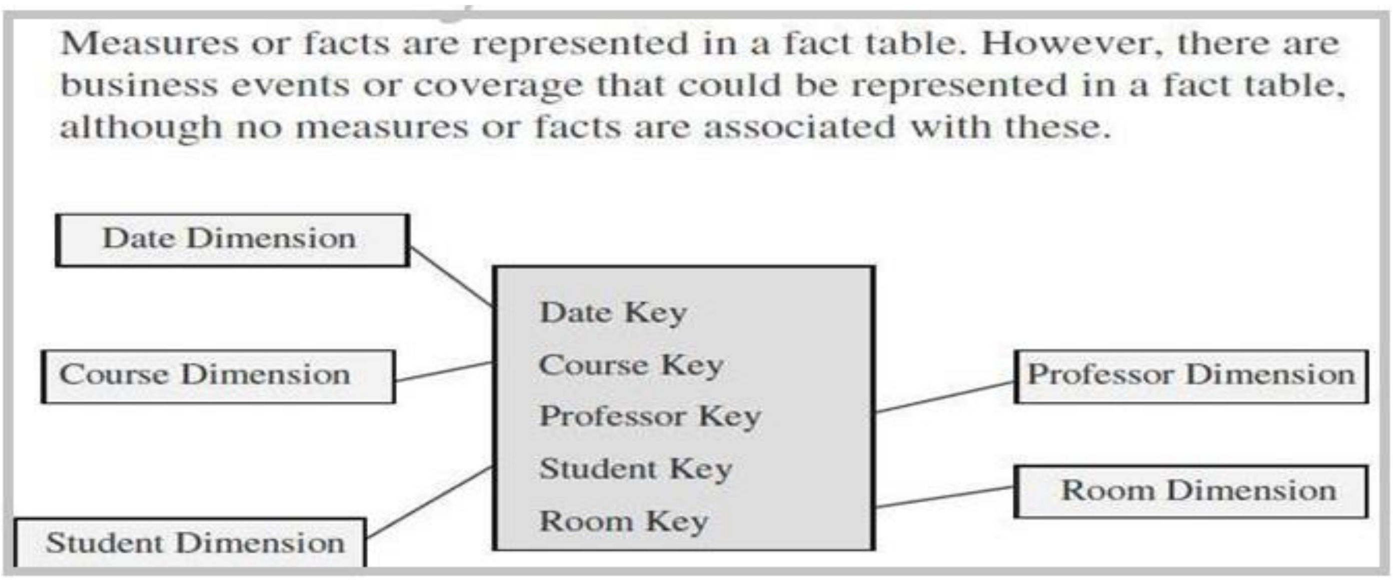
Factless Fact Table

In the real world, it is possible to have a fact table that contains no measures or facts. These tables are called “Factless Fact tables”.

Eg: A fact table which has only product key and date key is a factless fact. There are no measures in this table. But still you can get the number products sold over a period of time.

FACTLESS FACT TABLE

A table without any value added measures (additive measure) is called factless fact table.
Features: Used for covering an event / recording an event.


Measures or facts are represented in a fact table. However, there are business events or coverage that could be represented in a fact table, although no measures or facts are associated with these.

Vinay Tech House

SURROGATE KEY AND REAL-TIME USAGE

1. A surrogate key is a system generated (could be GUID, sequence, etc.) value with no business meaning that is used to uniquely identify a record in a table.

2. **The key itself could be made up of one or multiple columns.**

3. Usually created at dimension table and placed in Fact table to recognize dimension data quickly

4. Using **regular sequence generation mechanisms and custom algorithms** we create them.

Ex: SQL Server (Identity and Sequence), Oracle (Sequence)

5. Surrogate key values are non-changeable.

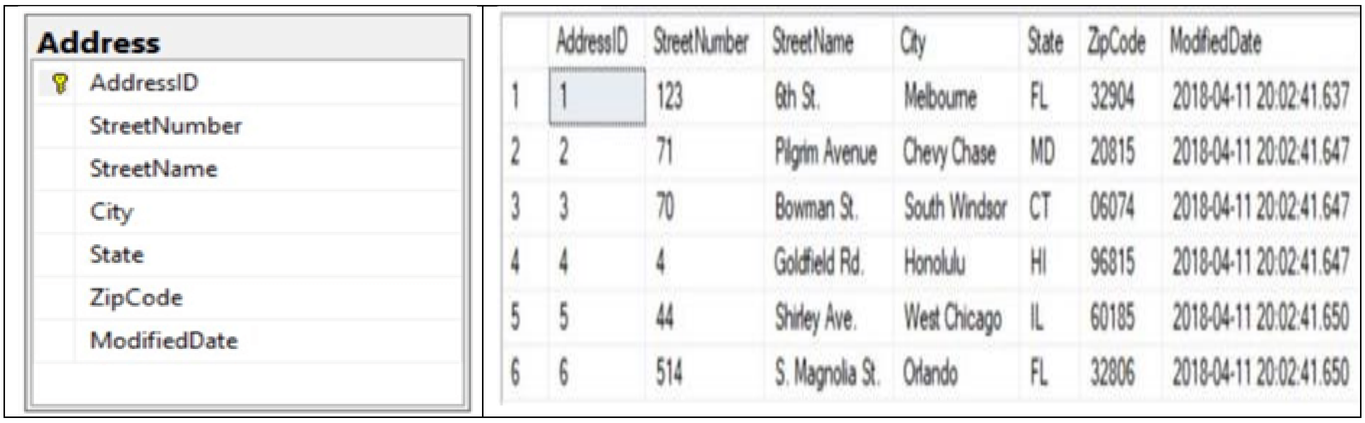
6. We also use at standalone tables to have uniqueness of values

The following diagram shows an example of a table with a surrogate key (AddressID column) along with some sample data. **Notice the key itself has no business meaning, it's just a sequential integer.**

Example: A standalone table [dimension table having surrogate key on AddressID]

Vinay Tech House

Here AddressID recognize the records uniquely



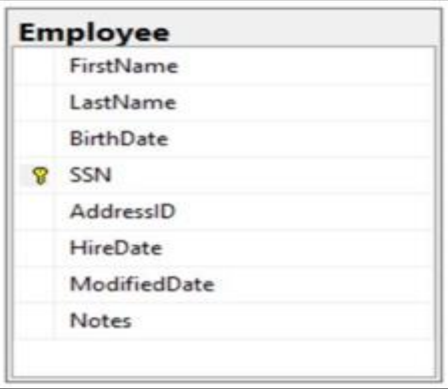
Address
- AddressID
- StreetNumber
- StreetName
- City
- State
- ZipCode
- ModifiedDate

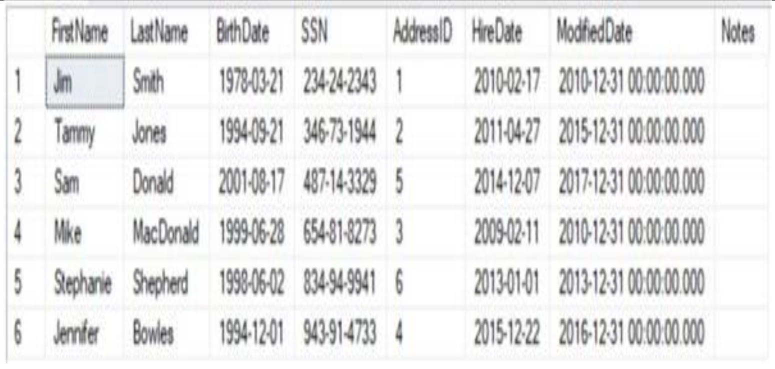
	AddressID	StreetNumber	StreetName	City	State	ZipCode	ModifiedDate
1	1	123	6th St.	Melbourne	FL	32904	2018-04-11 20:02:41.637
2	2	71	Pilgrim Avenue	Chevy Chase	MD	20815	2018-04-11 20:02:41.647
3	3	70	Bowman St.	South Windsor	CT	06074	2018-04-11 20:02:41.647
4	4	4	Goldfield Rd.	Honolulu	HI	96815	2018-04-11 20:02:41.647
5	5	44	Shirley Ave.	West Chicago	IL	60185	2018-04-11 20:02:41.650
6	6	514	S. Magnolia St.	Orlando	FL	32806	2018-04-11 20:02:41.650

Natural Key Overview

A natural key is a column or set of columns that already exist in the table (e.g. they are attributes of the entity within the data model) and uniquely identify a record in the table. Since these columns are attributes of the entity they obviously have business meaning.

The following is an example of a table with a natural key (SSN column) along with some sample data. **Notice that the key for the data in this table has business meaning.**





	FirstName	LastName	BirthDate	SSN	AddressID	HireDate	ModifiedDate	Notes
1	Jim	Smith	1978-03-21	234-24-2343	1	2010-02-17	2010-12-31 00:00:00.000	
2	Tammy	Jones	1994-09-21	346-73-1944	2	2011-04-27	2015-12-31 00:00:00.000	
3	Sam	Donald	2001-08-17	487-14-3329	5	2014-12-07	2017-12-31 00:00:00.000	
4	Mike	MacDonald	1999-06-28	654-81-8273	3	2009-02-11	2010-12-31 00:00:00.000	
5	Stephanie	Shepherd	1998-06-02	834-94-9941	6	2013-01-01	2013-12-31 00:00:00.000	
6	Jennifer	Bowles	1994-12-01	943-91-4733	4	2015-12-22	2016-12-31 00:00:00.000	

Natural Key Pros

- Key values have business meaning and can be used as a search key when querying the table
- Column(s) and primary key index already exist so no disk extra space is required for the extra column/index that would be used by a surrogate key column
- Fewer table joins since join columns have meaning.
  For example, this can reduce disk IO by not having to perform extra reads on a lookup table

Natural Key Cons

- May need to change/rework key if business requirements change. For example, if you used SSN for your employee as in the example above and your company expands outside of the United States not all employees would have a SSN so you would have to come up with a new key.
- More difficult to maintain if key requires multiple columns. It's much easier from the application side dealing with a key column that is constructed with just a single column.
- Poorer performance since key value is usually larger and/or is made up of multiple columns. Larger keys will require more IO both when inserting/updating data as well as when you query.
- Can't enter record until key value is known. It's sometimes beneficial for an application to load a placeholder record in one table then load other tables and then come back and update the main table.
- Can sometimes be difficult to pick a good key. There might be multiple candidate keys each with their own trade-offs when it comes to design and/or performance.

Surrogate Key Pros

- No business logic in key so no changes based on business requirements. For example, if the Employee table above used a integer surrogate key you could simply add a separate column for SIN if you added an office in Canada (to be used in place of SSN)
- Less code if maintaining same key strategy across all entities. For example, application code can be reused when referencing primary keys if they are all implemented as a sequential integer.
- Better performance since key value is smaller. Less disk IO is required on when accessing single column indexes.

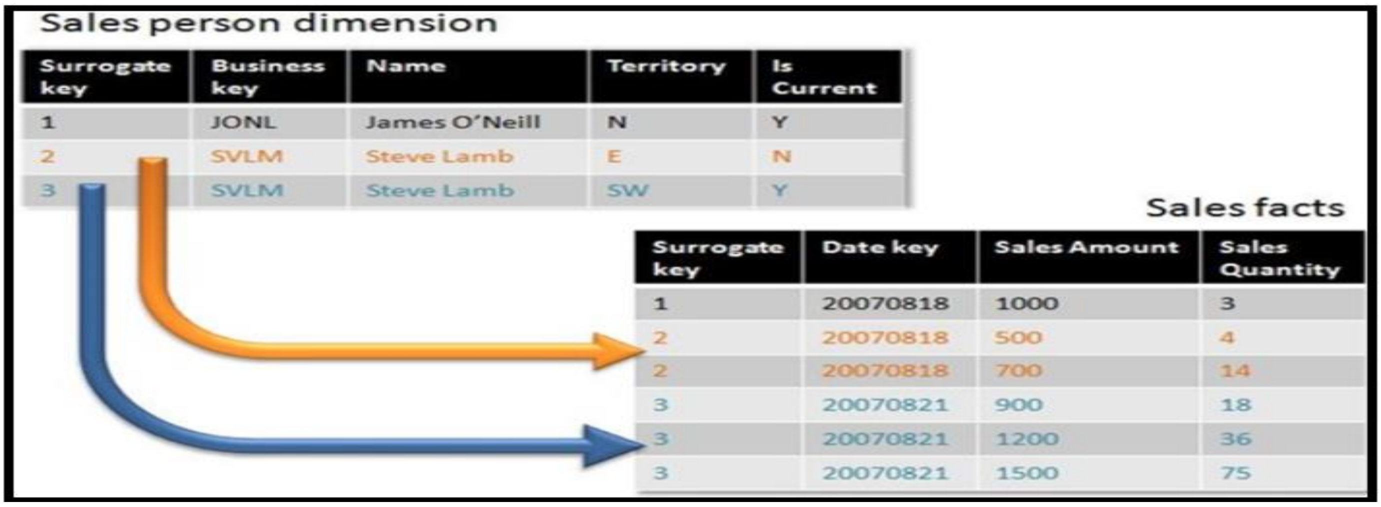
- Surrogate key is guaranteed to be unique.

  For example, when moving data between test systems you don't have to worry about duplicate keys since new key will be generated as data is inserted.
- If a sequence used then there is little index maintenance required since the value is ever increasing which leads to less index fragmentation.

Surrogate Key Cons

- Extra column(s)/index for surrogate key will require extra disk space
- Extra column(s)/index for surrogate key will require extra IO when insert/update data
- Requires more table joins to child tables since data has no meaning on its own.
- Can have duplicate values of natural key in table if there is no other unique constraint defined on the natural key
- Difficult to differentiate between test and production data. For example, since surrogate key values are just auto-generated values with no business meaning it's hard to tell if someone took production data and loaded it into a test environment.
- Key value has no relation to data so technically design breaks 3NF
- The surrogate key value can't be used as a search key
- Different implementations are required based on database platform.

  For example, SQL Server identity columns are implemented a little bit different than they are in Postgres or DB2.


Sales person dimension

Surrogate key	Business key	Name	Territory	Is Current
1	JONL	James O'Neill	N	Y
2	SVLM	Steve Lamb	E	N
3	SVLM	Steve Lamb	SW	Y

Sales facts

Surrogate key	Date key	Sales Amount	Sales Quantity
1	20070818	1000	3
2	20070818	500	4
2	20070818	700	14
3	20070821	900	18
3	20070821	1200	36
3	20070821	1500	75

Differences between Primary Key and Surrogate Key:

Primary Key:

a) **Database physical key**

b) **Unique values taking in a column**

Surrogate Key:

a) **Not a database key, we take in DWH and BI projects while working with records to address them uniquely and properly**

b) **Already you have a primary key in the table, still you need uniqueness to recognize records, then we go for surrogate key**

Vinay Tech House

E-R MODEL and RELATIONSHIPS [Entity, Attribute, Cardinality etc…]

Entity

An ERD entity is a **definable thing or concept within a system**, such as a person/role (e.g. Student), object (e.g. Invoice), concept (e.g. Profile) or event (e.g. Transaction) (note: In ERD, the term "entity" is often used instead of "table", but they are the same).

Entity Attribute Also known as a column, an attribute is a **property or characteristic of the entity that holds it**. An attribute has a name that describes the property and a type that describes the kind of attribute it is, such as varchar for a string, and int for integer.

Attribute	Data Type
CourseID	Integer
Courname	Varchar(30)

	CourseID (Key)	Coursename
	Power BI-F	Power BI Fast Track
	Power BI-N	Power BI Normal Track
	Power BI-C	Power BI Customized

Tech House

Primary Key

Also known as PK, a primary key is a special kind of entity attribute that **uniquely defines a record in a database table**.

Foreign Key

Also known as FK, a foreign key is a **reference to a primary key in a table**. It is used to identify the relationships between entities. Note that foreign keys need not be unique. Multiple records can share the same values.

Relationship

A relationship between two entities signifies that the **two entities are associated with each other somehow**.

For example, a student might enroll in a course. The entity Student is therefore related to Course, and a relationship is presented as a connector connecting between them.

Cardinality

Cardinality defines the **possible number of occurrences in one entity which is associated with the number of occurrences in another**.

For example, one student can join multiple courses [I: Many].

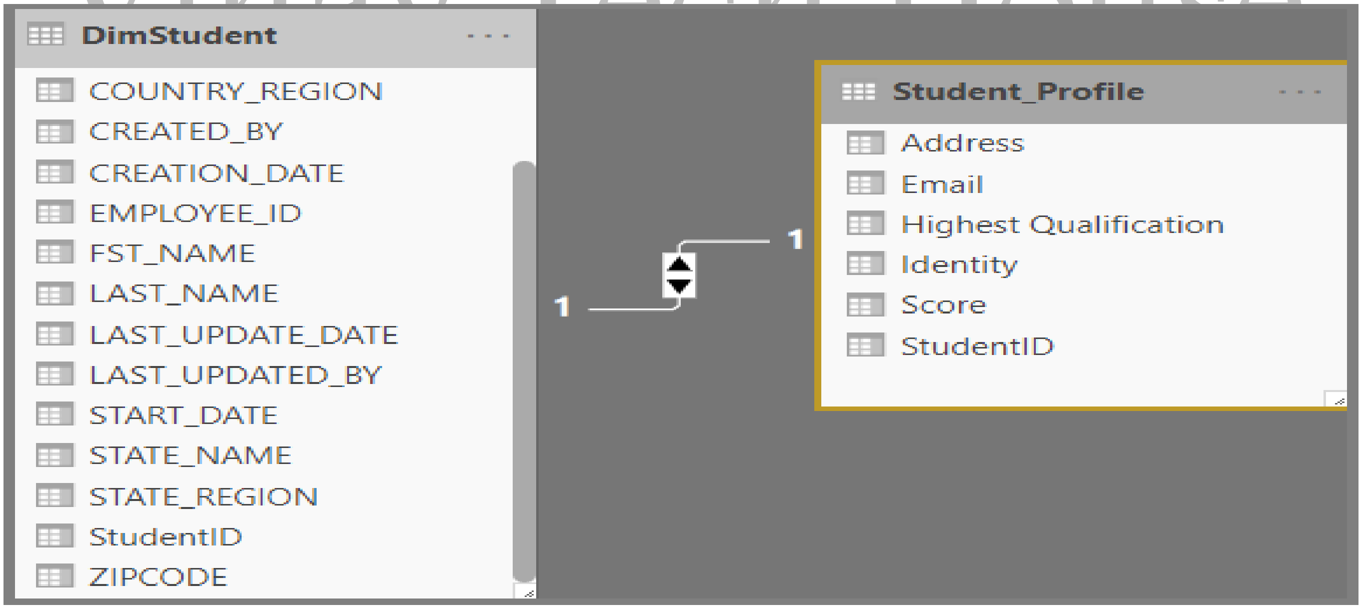
Multiple **cardinalities available (1:1, 1: Many, Many: Many).**

One-to-One cardinality example

A one-to-one relationship is mostly used to split an entity in two to provide information concisely and make it more understandable.

Example: One student having one profile

Vinay Tech House



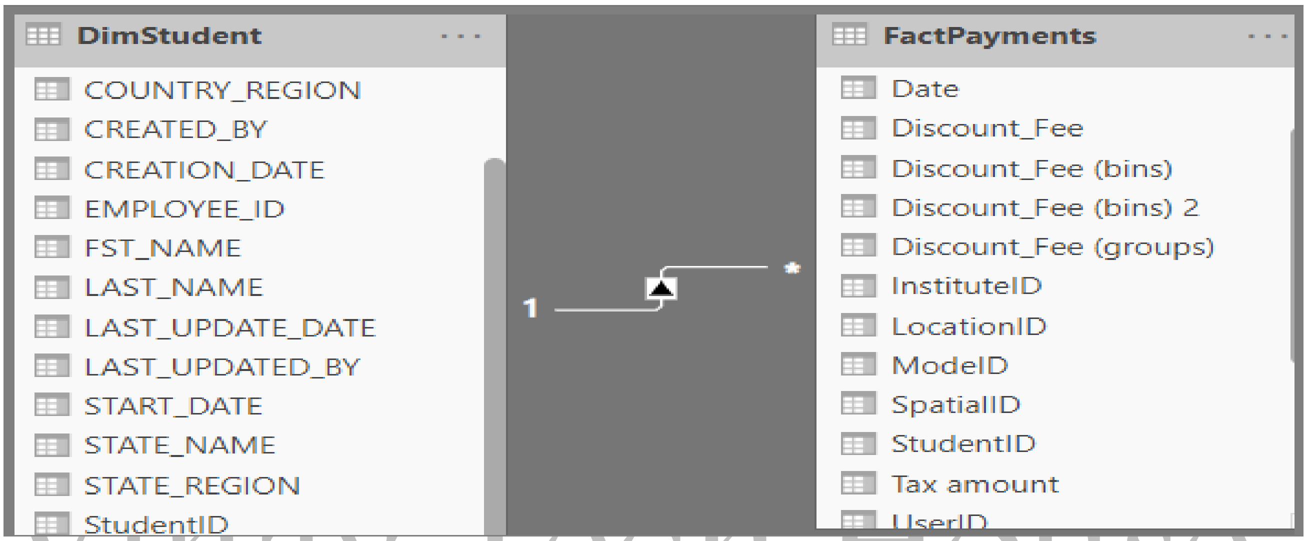
1. One to one cardinality, **single and bidirectional indicate same.**

2. No Bridge table is required

One-to-Many relationship

A one-to-many relationship refers to the relationship between two entities X and Y in which an instance of X may be linked to many instances of Y, but an instance of Y is linked to only one instance of X.

Ex: One Student may join multiple courses



Vinay Tech House

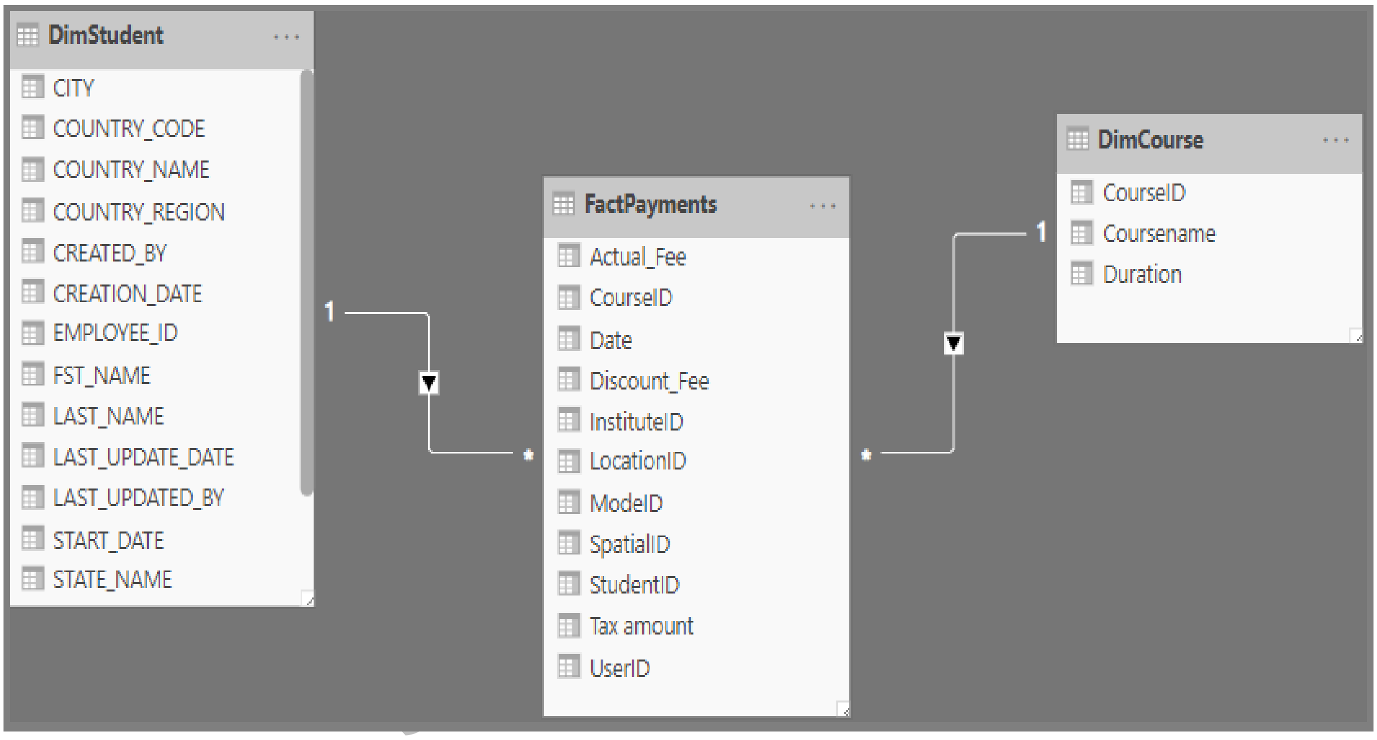
Note:

1. One to Many cardinality, **single and bidirectional are different.**

2. No Bridge table is required

Many-to-Many relationship

A many-to-many relationship refers to the relationship between two entities X and Y in which X may be linked to many instances of Y and vice versa. The figure below shows an example of a many-to-many relationship.



Note:

1. Many to Many cardinality, **single and bidirectional are different.**

2. Bridge table is required

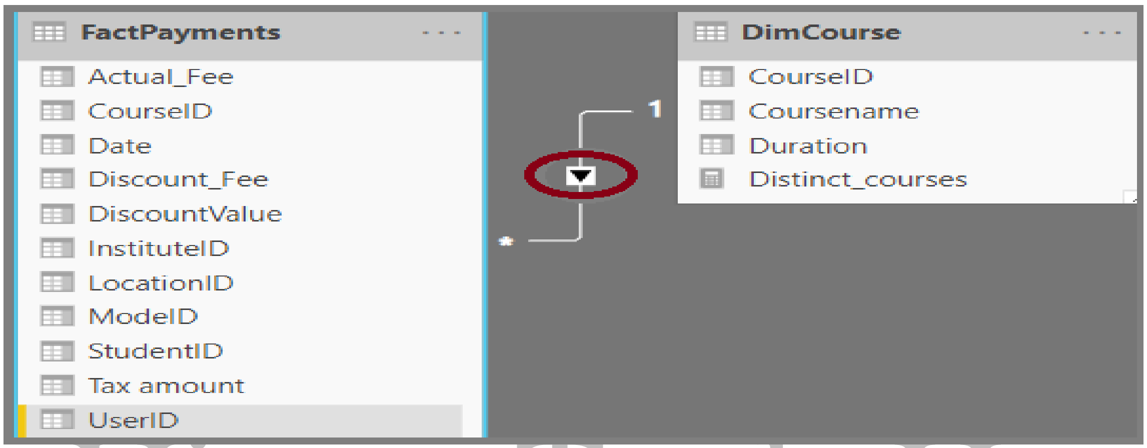
3. Note that a many-to-many relationship is split into a pair of **one-to-many relationships.**

SINGLE AND BI-DIRECTIONAL RELATIONSHIPS

1-Many Single Direction

DimCourse to Fact having one to many relationships, If I would like to know how many unique courses used in a year in the Fact table, you will get all courses count as a result however you slice (year or location or any…)

In this situation Fact table evaluation happened based on dimension entries (Dimensions→Fact)



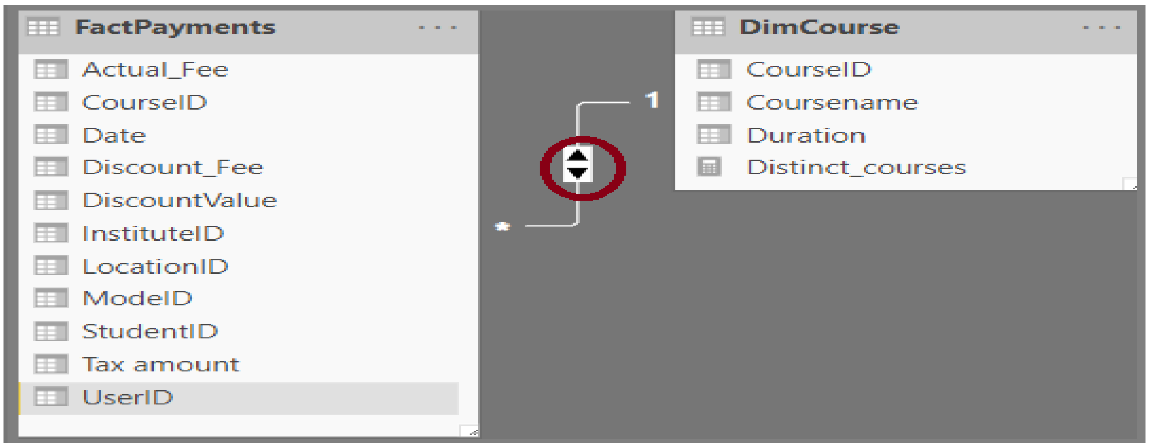
Vinay Tech House

Best Question For Single Direction: Courses and the total discount fee values

Bidirectional relationship

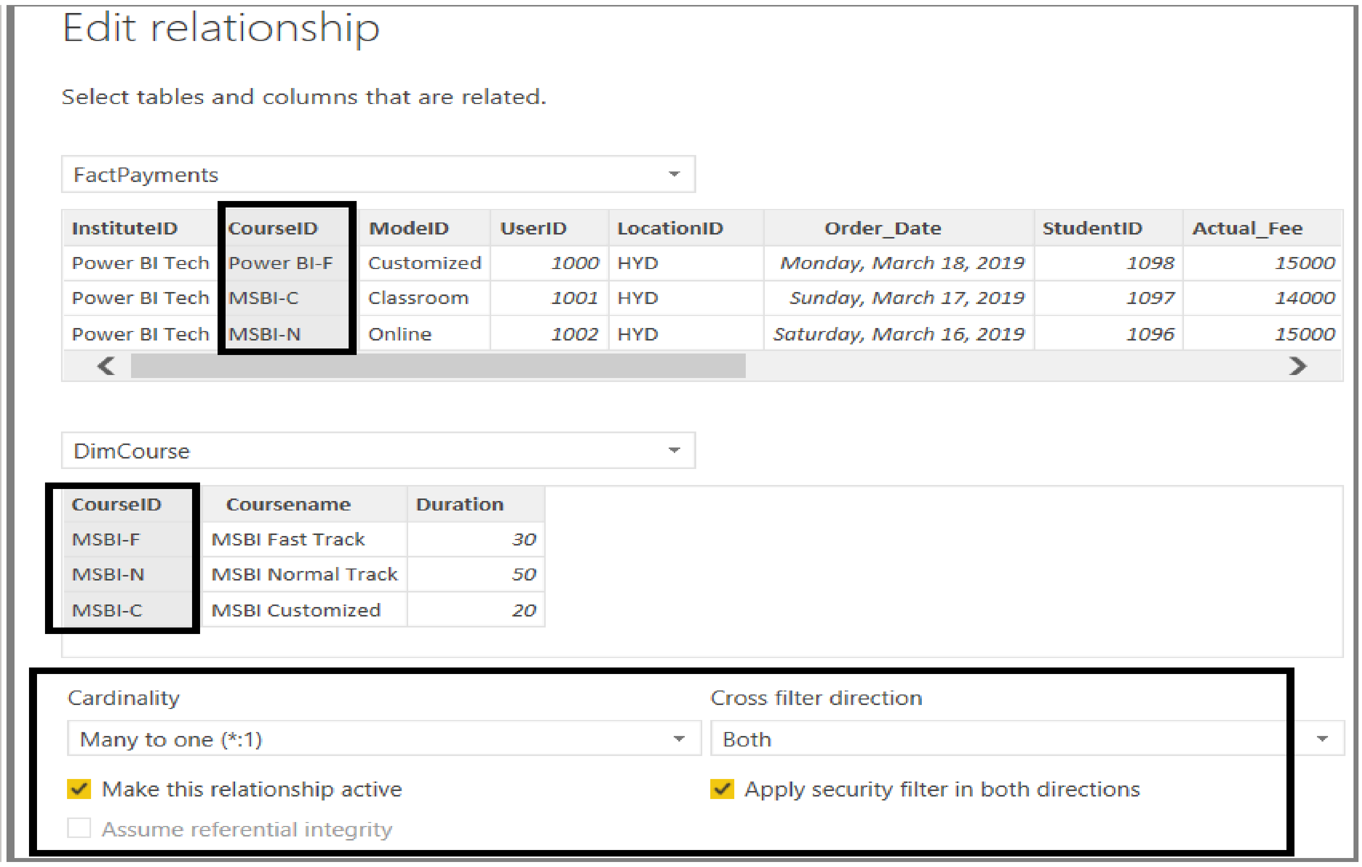
Here Dimcourse to Fact relationship both the directions, so if you slice on any column (Ex: year), then respective courses used in the fact table you will find.

In this situation DimCourse evaluation happened based on fact table entries (Fact→Dimensions)



Best Question for Both Direction: Number of Courses Out of Total Courses having Business?

Modifying relationships in Analysis Services and Power BI



Edit relationship

Select tables and columns that are related.

FactPayments

InstituteID	CourseID	ModelD	UserID	LocationID	Order_Date	StudentID	Actual_Fee
Power BI Tech	Power BI-F	Customized	1000	HYD	Monday, March 18, 2019	1098	15000
Power BI Tech	MSBI-C	Classroom	1001	HYD	Sunday, March 17, 2019	1097	14000
Power BI Tech	MSBI-N	Online	1002	HYD	Saturday, March 16, 2019	1096	15000

DimCourse

CourseID	Coursename	Duration
MSBI-F	MSBI Fast Track	30
MSBI-N	MSBI Normal Track	50
MSBI-C	MSBI Customized	20

Cardinality: Many to one (*:1)

Cross filter direction: Both

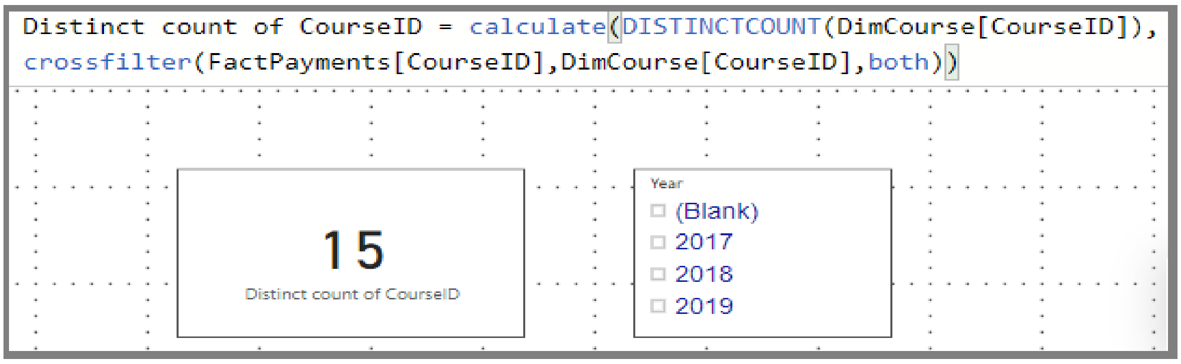
☑ Make this relationship active

☑ Apply security filter in both directions

☐ Assume referential integrity

Testing with a measure in DAX and place it in a Card

a) Create a measure like below to find DistinctCount of courses
b) Use a card on the report and place the measure
c) Use Year slicer like below and check the result



```
Distinct count of CourseID = calculate(DISTINCTCOUNT(DimCourse[CourseID]),
crossfilter(FactPayments[CourseID],DimCourse[CourseID],both))
```

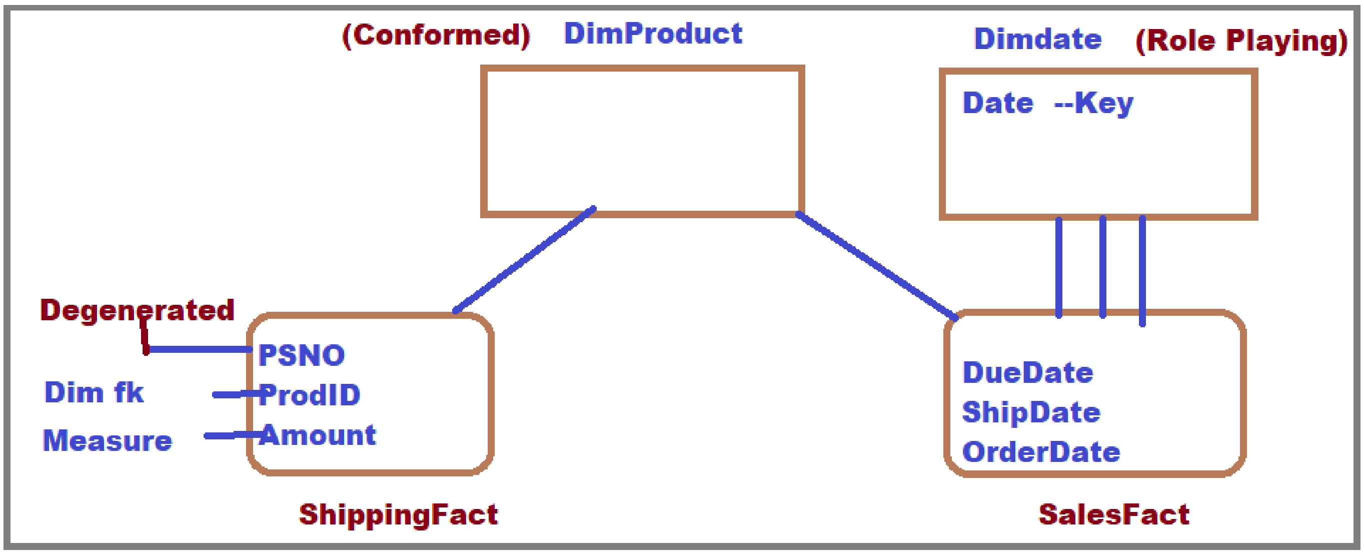
15

Distinct count of CourseID

Year

☐ (Blank)
☐ 2017
☐ 2018
☐ 2019

ACTIVE AND INACTIVE RELATIONSHIPS



Here the Date table is role playing dimension table. At a time one column of Date table is active (thick line) and remaining are inactive (dashed lines) and connected to Fact table. The column which is active only participated in the analysis.

Example:

Vinay Tech House

If OrderDate is active, order date analysis only possible.

In case you want to analyze DueDate or ShipDate, then make the OrderDate inactive and the require column put active.

Making relationship inactive:

Uncheck the above box to make relationship inactive.

Making relationship active

Specify the columns between tables, check the relationship box to make it active

NOTE: In the class room we play with this in case of role-playing dimension

DIMENSIONAL MODEL RELATIONSHIPS [STAR, SNOW FLAKE, GALAXY ETC…]

Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates.

Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses Star, Snowflake, and Fact Constellation schema. In this chapter, we will discuss the schemas used in a data warehouse.

Star Schema relationship and creating it:

Centralized fact table with surrounding dimensions is called star schema.

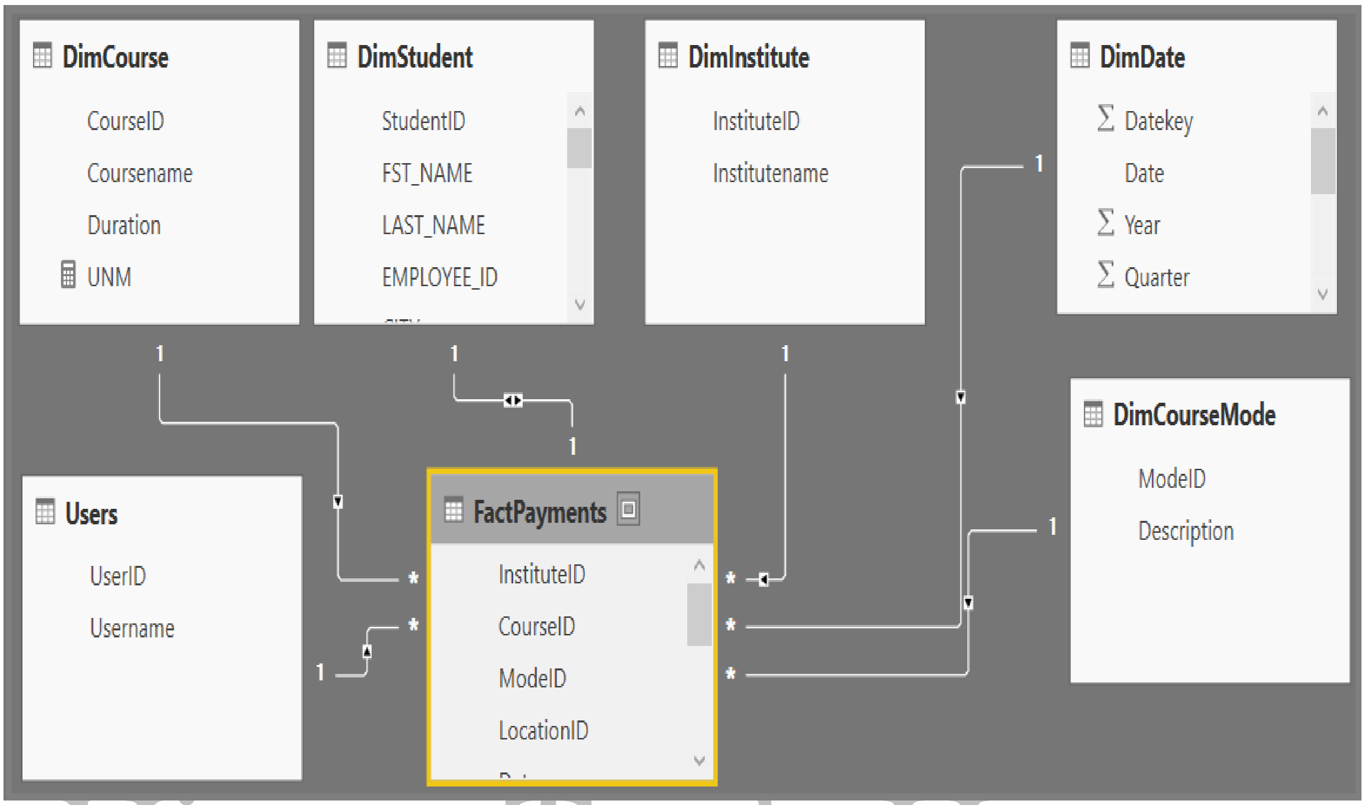
Star schema characteristics:

- The dimension table is connected to the fact table by using the foreign key.
- Dimension tables are not joined to each other.
- The schema is supported by BI tools.
- The dimension tables are not normalized.
- Star schema is very easy to understand.
- Star schema gives optimal disk usage.

Advantages of star schema:

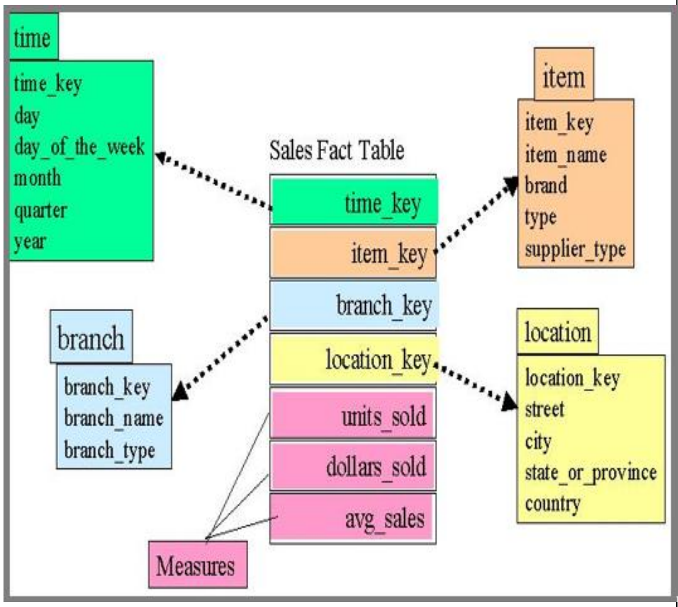
- Star schemas simplify the method of pulling business reports live period-over-period reports.
- Star schema gives data to Online Analytic Processing systems.
- As the complete data connects through a single fact table, the various dimension tables are considered as one huge table of data, and that makes queries more comfortable to perform.
- The performance of read-only commands is very high.
- The speed of the query is very high.

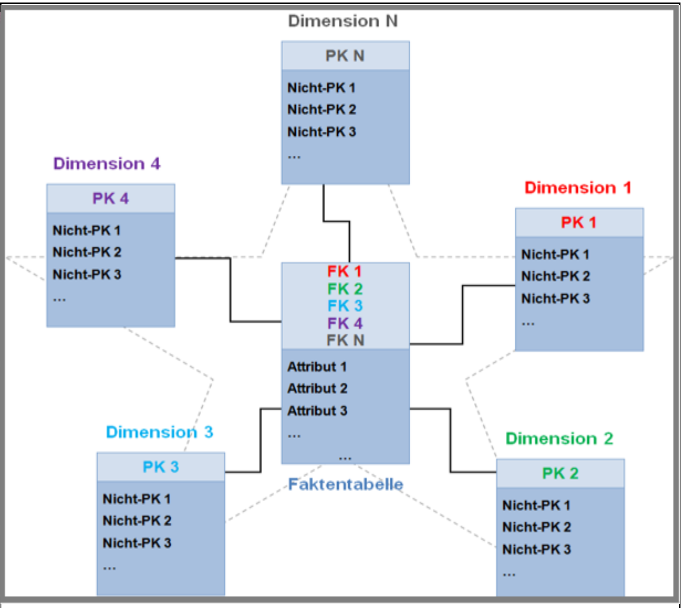
Classroom pic:



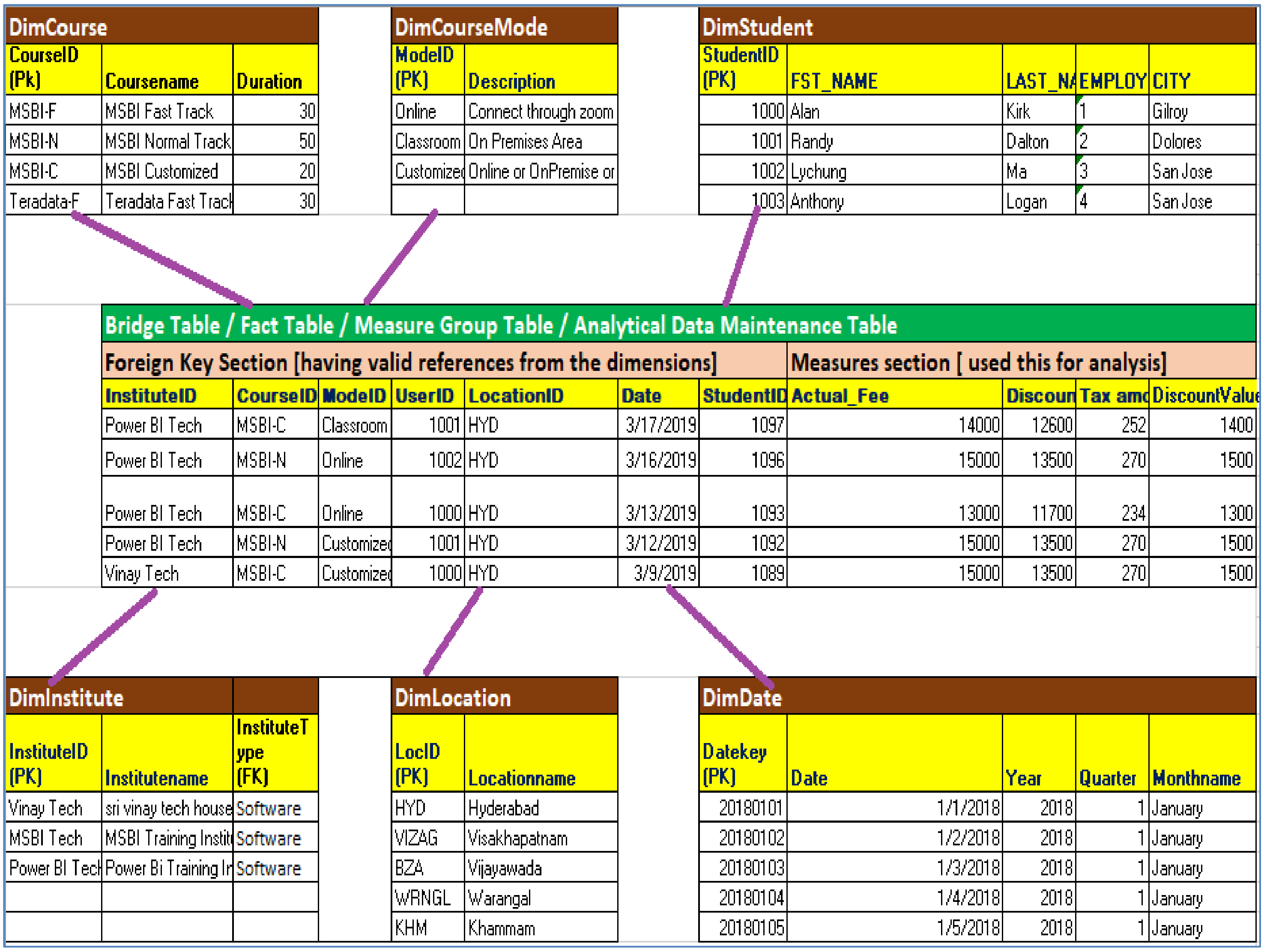
Vinay Tech House

Other images:





Business data store: Sales information
Dimensions: **Date, Branch, Location, and Item**
Measures: Units_sold, Dollars_Sold, and Avg_Sales



DimCourse

CourseID (Pk)	Coursename	Duration
MSBI-F	MSBI Fast Track	30
MSBI-N	MSBI Normal Track	50
MSBI-C	MSBI Customized	20
Teradata-F	Teradata Fast Track	30

DimCourseMode

ModeID (PK)	Description
Online	Connect through zoom
Classroom	On Premises Area
Customized	Online or OnPremise or

DimStudent

StudentID (PK)	FST_NAME	LAST_NA	EMPLOY	CITY
1000	Alan	Kirk	1	Gilroy
1001	Randy	Dalton	2	Dolores
1002	Lychung	Ma	3	San Jose
1003	Anthony	Logan	4	San Jose

Bridge Table / Fact Table / Measure Group Table / Analytical Data Maintenance Table

Foreign Key Section [having valid references from the dimensions]							Measures section [used this for analysis]			
InstituteID	**CourseID**	**ModeID**	**UserID**	**LocationID**	**Date**	**StudentID**	**Actual_Fee**	**Discoun**	**Tax amo**	**DiscountValue**
Power BI Tech	MSBI-C	Classroom	1001	HYD	3/17/2019	1097	14000	12600	252	1400
Power BI Tech	MSBI-N	Online	1002	HYD	3/16/2019	1096	15000	13500	270	1500
Power BI Tech	MSBI-C	Online	1000	HYD	3/13/2019	1093	13000	11700	234	1300
Power BI Tech	MSBI-N	Customized	1001	HYD	3/12/2019	1092	15000	13500	270	1500
Vinay Tech	MSBI-C	Customized	1000	HYD	3/9/2019	1089	15000	13500	270	1500

DimInstitute

InstituteID (PK)	Institutename	InstituteType (FK)
Vinay Tech	sri vinay tech house	Software
MSBI Tech	MSBI Training Instit	Software
Power BI Tech	Power Bi Training In	Software

DimLocation

LocID (PK)	Locationname
HYD	Hyderabad
VIZAG	Visakhapatnam
BZA	Vijayawada
WRNGL	Warangal
KHM	Khammam

DimDate

Datekey (PK)	Date	Year	Quarter	Monthname
20180101	1/1/2018	2018	1	January
20180102	1/2/2018	2018	1	January
20180103	1/3/2018	2018	1	January
20180104	1/4/2018	2018	1	January
20180105	1/5/2018	2018	1	January

Snow flake Schema relationship and creating it:

Centralized fact table with surrounding dimensions is called star schema.

In snow flake schema dimensions contain sub dimensions [dimensional hierarchies available].

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.

Note − Due to normalization in the Snowflake schema, the redundancy is reduced and

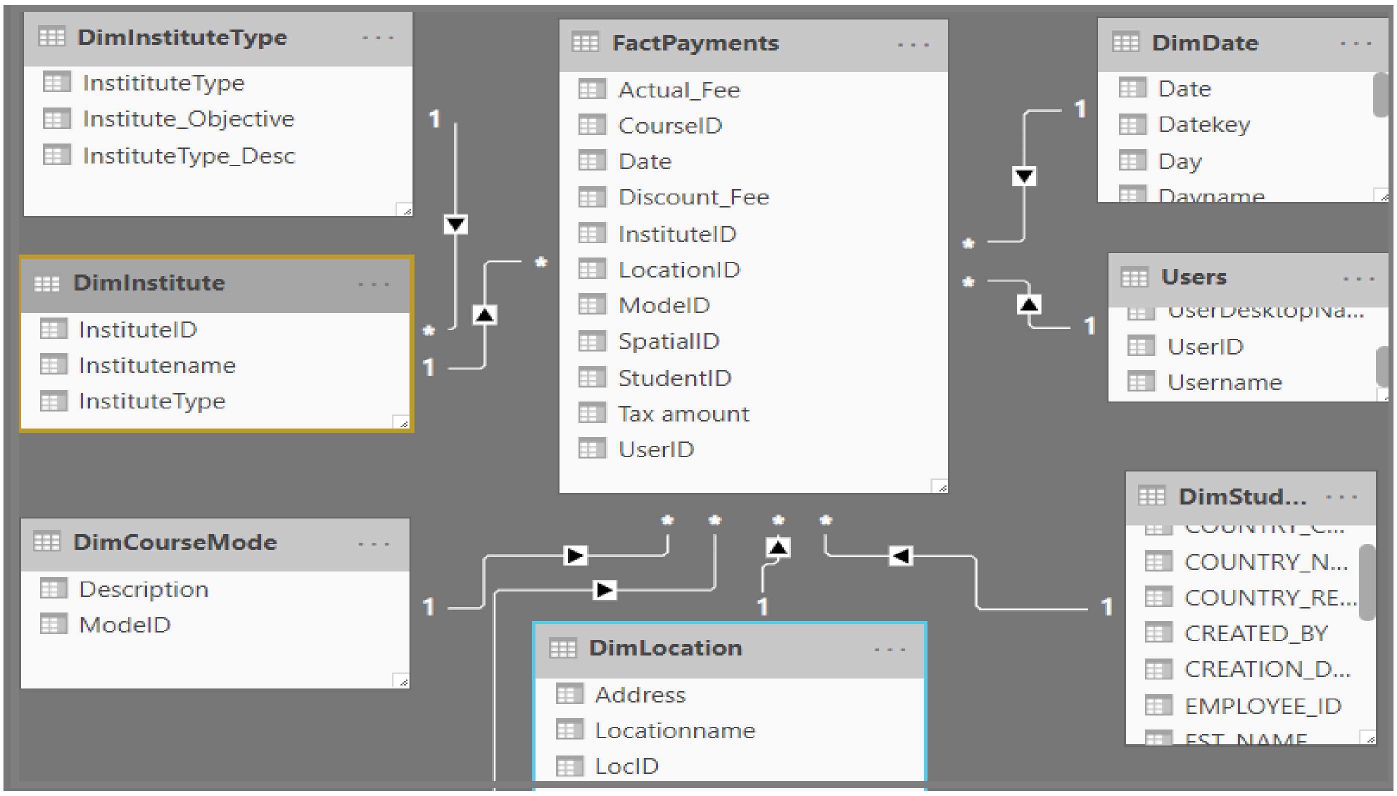
therefore, it becomes easy to maintain and the save storage space.

Snowflake Schema Characteristics:

- The biggest advantage of the snowflake schema is it uses minimal disk space.
- Because of multiple tables, the query performance will be reduced.
- Query performance is reduced due to multiple tables.
- Maintenance efforts are required because of more lookup tables.

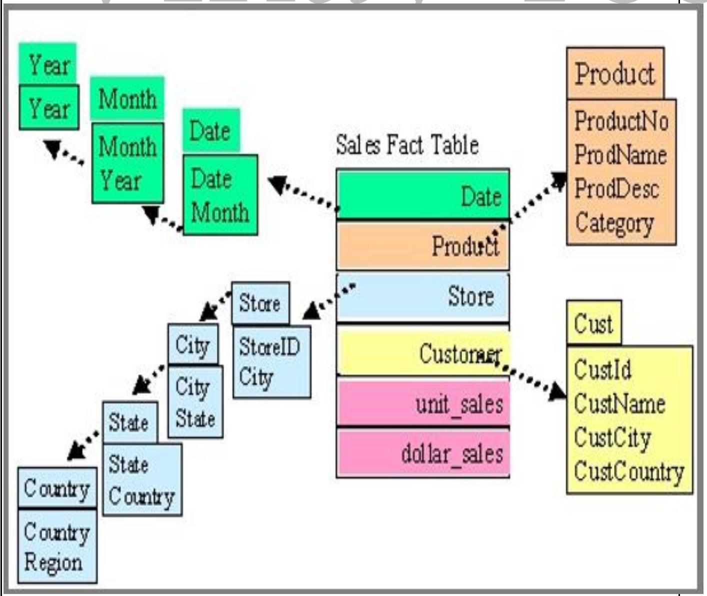
Advantages of Snowflake Schema:

- Some OLAP database tools data scientists utilize for data modeling and analysis are mainly designed to work snowflake data schemes.
- Normalizing the data which would typically get denormalized within a star schema can give an enormous reduction in disk space requirements. This is because you are converting long strings of non-numerical data within numerical keys which are dramatically abrupt taxing from a storage viewpoint.

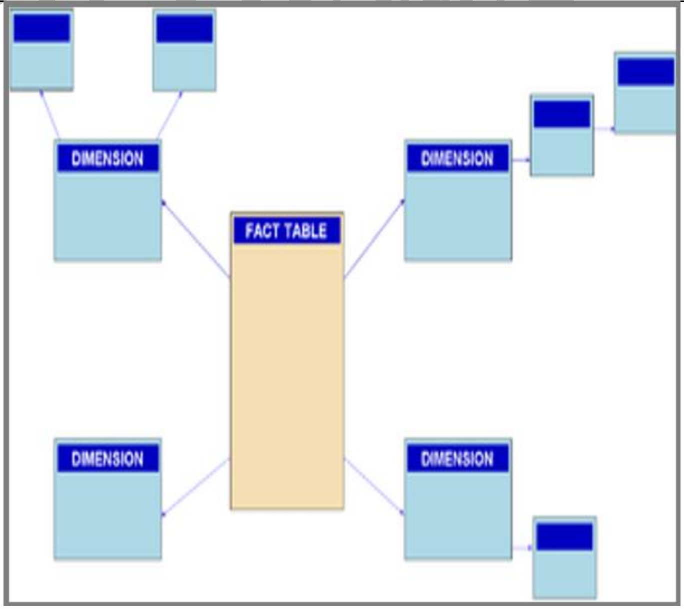


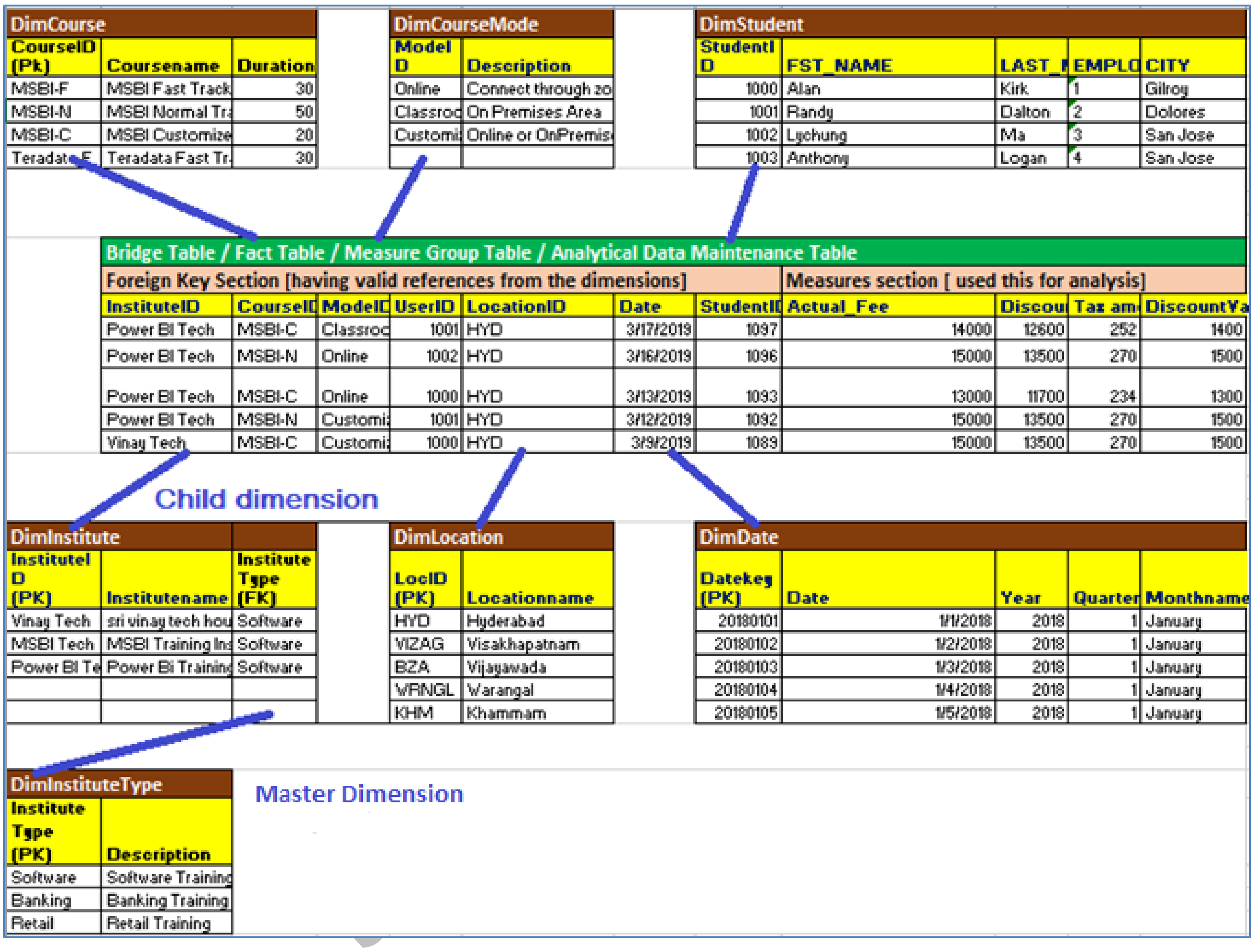
Other Images:

Vinay Tech House



Store dimension having city, state, and country parent dimensions.





DimCourse

CourseID (Pk)	Coursename	Duration
MSBI-F	MSBI Fast Track	30
MSBI-N	MSBI Normal Tr	50
MSBI-C	MSBI Customize	20
Teradata-F	Teradata Fast Tr	30

DimCourseMode

ModelD	Description
Online	Connect through zo
Classroc	On Premises Area
Customi	Online or OnPremis

DimStudent

StudentID	FST_NAME	LAST_I	EMPLO	CITY
1000	Alan	Kirk	1	Gilroy
1001	Randy	Dalton	2	Dolores
1002	Lychung	Ma	3	San Jose
1003	Anthony	Logan	4	San Jose

Bridge Table / Fact Table / Measure Group Table / Analytical Data Maintenance Table

Foreign Key Section [having valid references from the dimensions]							Measures section [used this for analysis]			
InstituteID	CourseII	ModelC	UserID	LocationID	Date	StudentII	Actual_Fee	Discou	Tax am	DiscountVa
Power BI Tech	MSBI-C	Classroc	1001	HYD	3/17/2019	1097	14000	12600	252	1400
Power BI Tech	MSBI-N	Online	1002	HYD	3/16/2019	1096	15000	13500	270	1500
Power BI Tech	MSBI-C	Online	1000	HYD	3/13/2019	1093	13000	11700	234	1300
Power BI Tech	MSBI-N	Customi	1001	HYD	3/12/2019	1092	15000	13500	270	1500
Vinay Tech	MSBI-C	Customi	1000	HYD	3/9/2019	1089	15000	13500	270	1500

Child dimension

DimInstitute

InstituteID (PK)	Institutename	Institute Type (FK)
Vinay Tech	sri vinay tech hou	Software
MSBI Tech	MSBI Training In	Software
Power BI Te	Power Bi Training	Software

DimLocation

LocID (PK)	Locationname
HYD	Hyderabad
VIZAG	Visakhapatnam
BZA	Vijayawada
VRNGL	Warangal
KHM	Khammam

DimDate

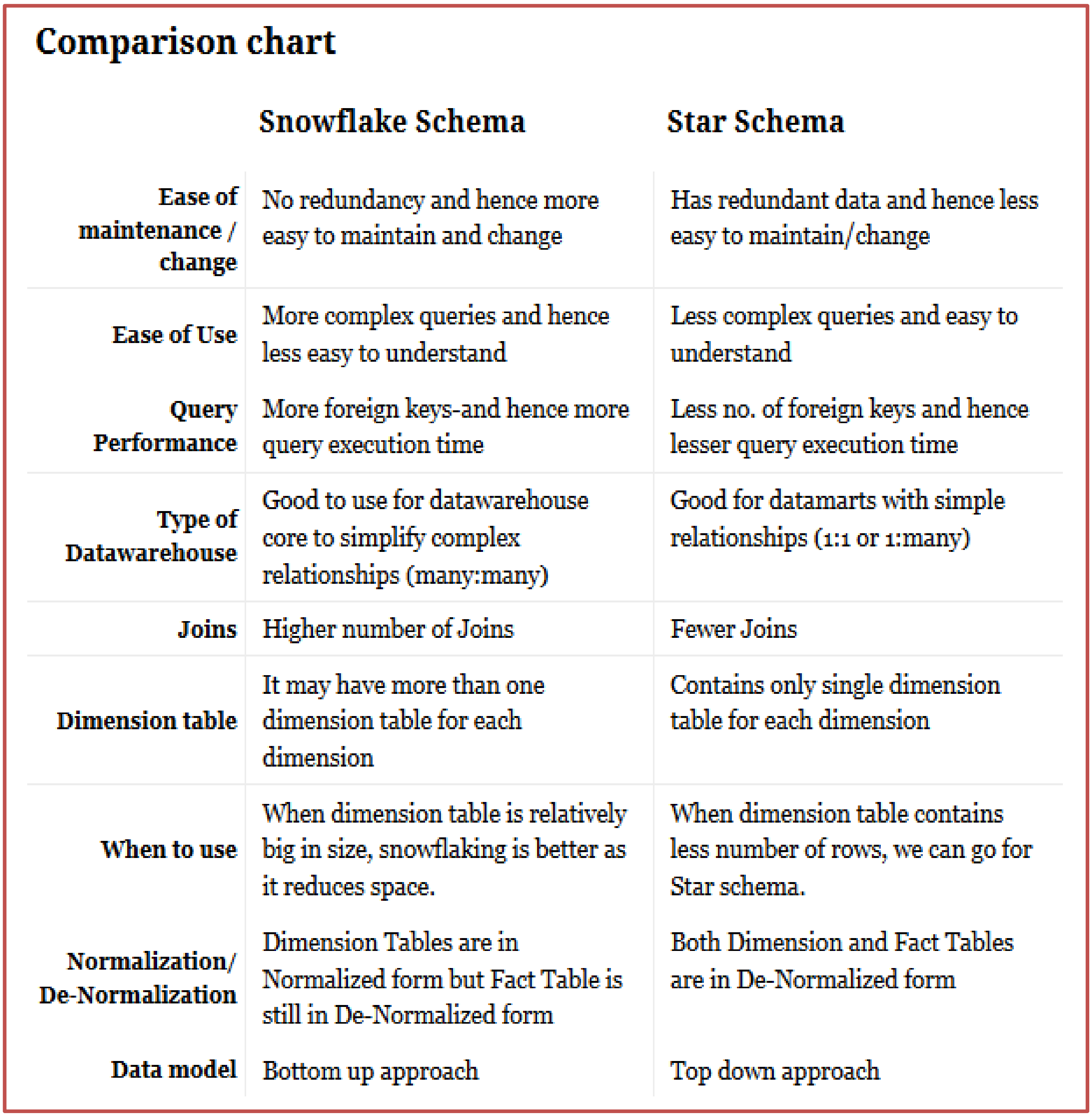
Datekey (PK)	Date	Year	Quarter	Monthname
20180101	1/1/2018	2018	1	January
20180102	1/2/2018	2018	1	January
20180103	1/3/2018	2018	1	January
20180104	1/4/2018	2018	1	January
20180105	1/5/2018	2018	1	January

DimInstituteType

Institute Type (PK)	Description
Software	Software Training
Banking	Banking Training
Retail	Retail Training

Master Dimension

Differences between star and snow flake schema

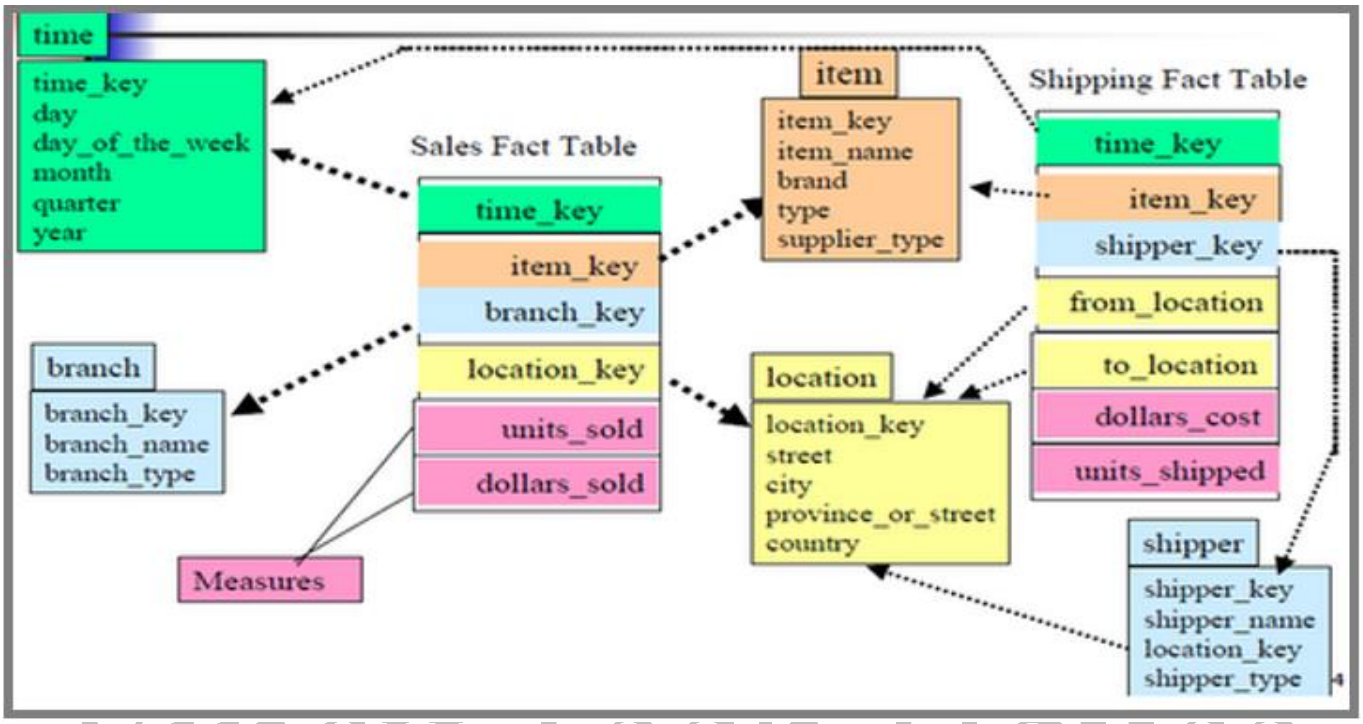


Comparison chart

	Snowflake Schema	Star Schema
Ease of maintenance / change	No redundancy and hence more easy to maintain and change	Has redundant data and hence less easy to maintain/change
Ease of Use	More complex queries and hence less easy to understand	Less complex queries and easy to understand
Query Performance	More foreign keys-and hence more query execution time	Less no. of foreign keys and hence lesser query execution time
Type of Datawarehouse	Good to use for datawarehouse core to simplify complex relationships (many:many)	Good for datamarts with simple relationships (1:1 or 1:many)
Joins	Higher number of Joins	Fewer Joins
Dimension table	It may have more than one dimension table for each dimension	Contains only single dimension table for each dimension
When to use	When dimension table is relatively big in size, snowflaking is better as it reduces space.	When dimension table contains less number of rows, we can go for Star schema.
Normalization/ De-Normalization	Dimension Tables are in Normalized form but Fact Table is still in De-Normalized form	Both Dimension and Fact Tables are in De-Normalized form
Data model	Bottom up approach	Top down approach

Galaxy Schema or Fact Constellation Schema

Multiple fact tables using common dimensions



Vinay Tech House

Explanation of the above image:

1. In this picture, we have two fact tables a) Sales (which stores product sales information) b) Shipping (which store shipping or delivery details)

2. Both the fact tables using common dimensions such as Item and Time. So, we called them as **"conformed" dimensions**.

3. Location table having multiple keys in the shipping fact table, so location table is called as **"Role Playing"** dimension table.

MIXED SCHEMA / HYBRID SCHEMA

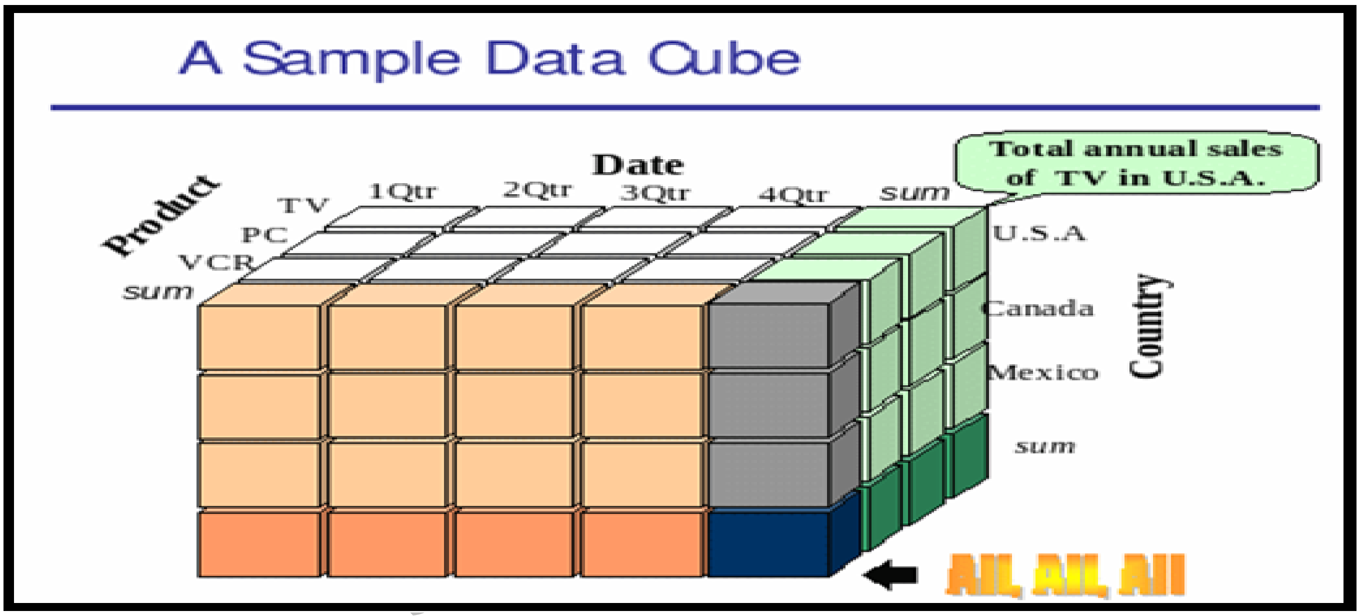
Combination of star and snow flake schema is called as mixed schema

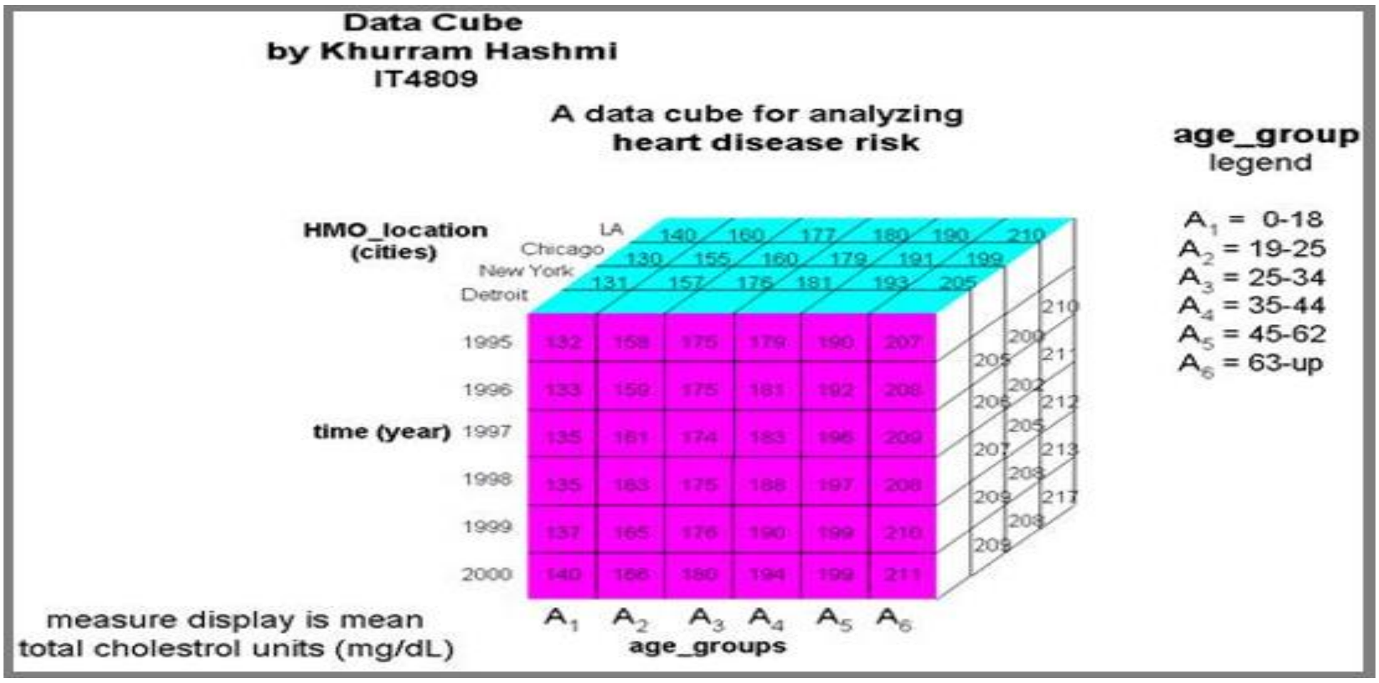
CUBE:

1.It is a multidimensional object which contains dimensions and facts and arranged in a schema model (star / snow flake / mixed (star+ snow flake) / Galaxy schema).

2. Help us to work with multiple dimensions data and to create aggregations for quick analysis.

3. Cube cells hold measurable data and corner cells hold aggregate data





NOTE: FOR FAQS (WWW.VINAYTECHHOUSE.COM-->OTHERS--> DATA MODEL FAQS)

Our Database "Vinaytech_Dev_Business_Details" having multiple dimensions and one fact table.

1-1: One value to one value mapping

Ex:
Each Course should contain one ID
[One Student--> One Student Profile]

1-Many: One value to many values mapping

Ex:
One date can be used by many people in the business.
DimDate (Date:PK)----> FactPayments(Date: FK)

One course can be joined by multiple students
DimCourse[CourseID:PK]----> FactPayments(CourseID: FK)
Etc...

Primary key column: No null value + No Duplicate values in the column

Vinay Tech House

Foreign Key column [references]: It refers to the master table (another) primary key or unique key values. It will not allow other than master table values.

Data Integrity Issues between dimension tables and fact tables:

LocationID(PK): HYD, BZA and Vizag

FactPayment (LocID:FK)

Correct LocID to Fact: HYD / BZA / Vizag
Wrong LocID to Fact: MUM [as it is not in the master table] as it is not FK violation
Note:This issue data integrity issue.

Dimension / Master tables in our class room database

DimCourse [CourseID:Pk]
DimMode [ModeID:Pk]
DimLocation [LocationID:Pk]
DimStudent [StudentID:Pk]
DimDate [Date:Pk]
DimUsers [UserID:Pk]
DimStudentProfile [StudentID;PK]
DimBranch [BranchID:PK]

Child / Transaction / fact table

FactPayments

FactPayment Columns

a) Foreign Keys: LocID, CourseID, ModeID, StudentID, UserID, InstituteID
b) Measures / Business Values: Actual_Fee, Discount_Fee and TaxAmount
Note: These measures suitable for aggregate and non-aggregate calculations.

Vinay Tech House

Many-Many:

Many courses can be handled by many students in many locations in different modes.

Always one bridge table (in the above picture FactPayments) is required to provide many-many relationship between all these dimension tables.

Practical:

Getdata→ Specify Sqlserver Instance and Database name (VINAYTECH_BUSINESS_DETAILS)→ Import

Choose tables DimCourse, DimMode, DimLocation, DimStudent, DimDate, Users, DimInstitute and FactPayment.

Load

Go to relationships view, observe the model, due to column names mismatch DimLocation and Factpayments are not connected.

Impact of DimLocation and FactPayents not connected.

Take PieChart→ Take vales (Drag DiscountFee from FactPayment) and Details (Loationname from DimLocation).

The chart displays same value for all locations.

Impact of DimLocation and FactPayents are connected.

a) In Relationships view, connect DimLocation LocationID to LocID of FactPayments

b) Take PieChart→ Take vales (Drag DiscountFee from FactPayment) and Details (Loationname from DimLocation).

The chart displays different value for all locations.

Vinay Tech House

SIMPLE UNDERSTANDING VINAYTECH BUSINESS DETAILS DWH DB

a) This is **DWH database** [not OLTP data store]

b) It holds **dimensions and facts**. All dimensions having 1:Many relationship with Fact table.

1: Many--> Cardinality

Cardinality: The number of times it appears

c) It is in the **Snow Flake Schema** format, because Dimension tables having hierarchies

[DimInstitute-->DimInstituteType]

d) Date table is **Role Playing Dimension table**, because it has **three foreign keys** in the Fact table We have only one column as active and remaining as inactive.

e) Date table has **surrogate key called DateKey** along with Primary Key (Date)

Ex: Date Key : 20191223

Date: 2019-12-23

Vinay Tech House

f) DimCourse has **bidirectional relationship** with **FactPayments Table**

DimStudent: Student Information [studentid,firstname, last name, dob etc...]

DimLocation: Location information [Locid, locname etc...]

DimInstitute: Institute information [Instid, name, etc...]

DimInstituteType: Master table for DimInstitute to store type of institute etc...

DimMode: Course Mode information [Modeid, Mode]

DimCourse: Courses information [CourseID, Coursename etc...]

DimDate: Dates Information

DimUsers: Store users information

FactPayments:

(Bridge table) which has Many:1 cardinality with all dimensions.

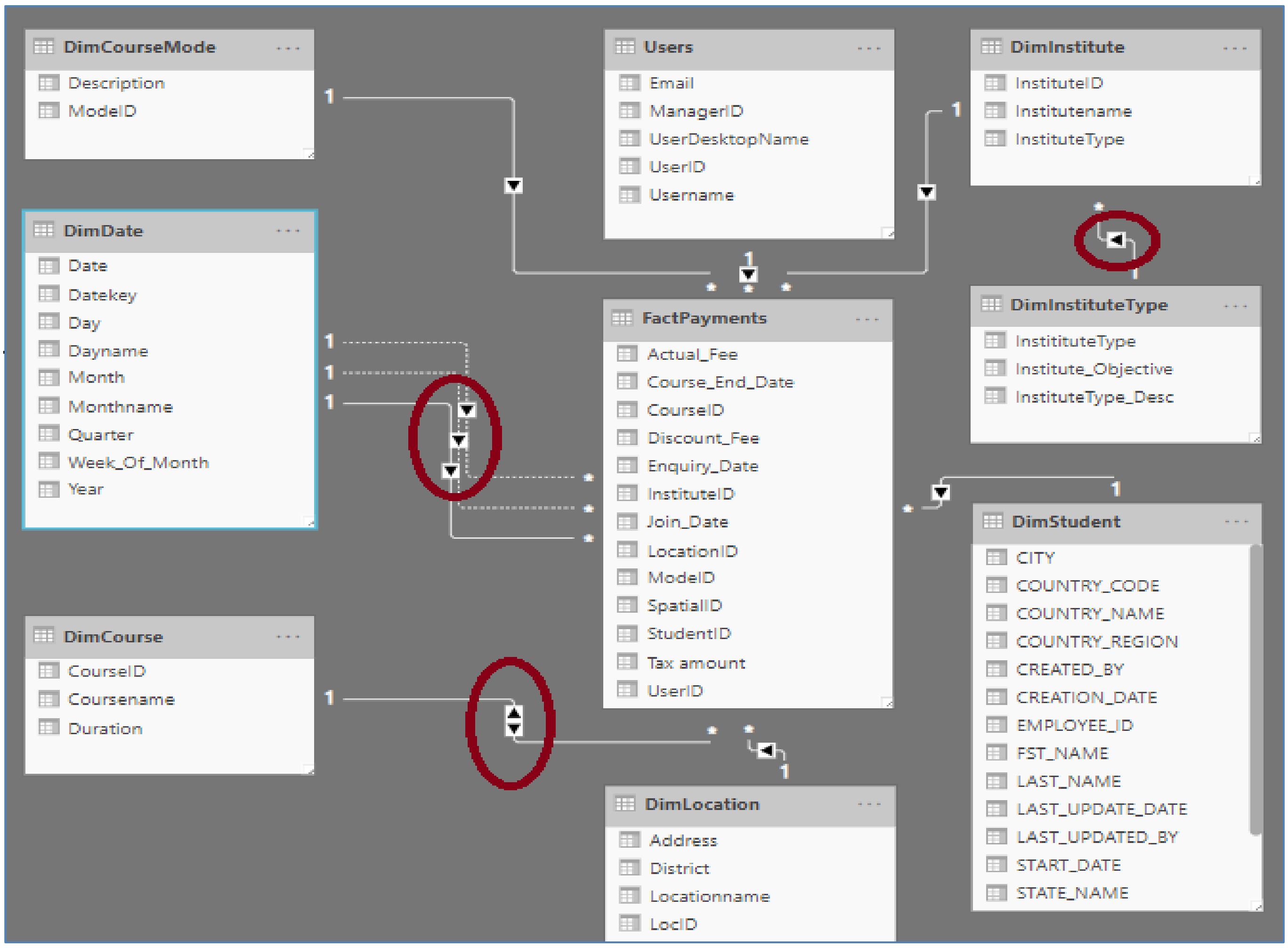
It also has measurable data for analysis.

Ex:

ActualFee

DiscountFee

TaxAmount



FOR POWER BI PEOPLE:

Get Data from the below places and generate a model as mentioned

[Heterogenous applications]

a) Get Csv file data [DimStudent]

b) Get Text file data [DimCourse]

c) Get Excel data   [DimDate, DimLocation, DimCourseMode, DimInstitute, DimInstituteType]

d) Get Table data [FactPayements data from VINAYTECH_DEV_BUSINESS_DETAILS database]

Retrieve like above and establish relationships in modeling area of Power BI

Vinay Tech House