Data Modeling
===============

what is meant by Data Modeling?

A way to structure your data so that it fits your needs in the best possible way.

needs can be different based on what system we are modeling?

OLTP (online transactional processing) Database

OLAP (online analytical processing) Datawarehouse

Needs can be different based on who is the consumer - Data analyst / Data Engineer

OLTP - relational modeling (Designed for writing)

primary goal when designing a OLTP is to minimize the redundancy.

how do you minimize redundancy?

Normalization

it's a technique to devide one big table into multiple smaller tables with an intent to reduce the redundancy.

1NF

- a single cell must not hold more than one value (atomicity)
- there must be a primary key for identification of rows
- no duplicated rows or columns

2NF
- it should be in 1st NF
- Non primary key attributes of the table should depend on complete candiate key

3NF
- it should be in 2nd NF
- should not have any transitive dependencies

1. First Normal form

StudentID	Courses
1	Math, English, Music
2	Science, History

StudentID	Course
1	Math
1	English
1	Music
2	Science
2	History

2. Second Normal Form

CAND_ID	SUBJECT_NO	SUBJECT_FEE
111	S1	1000
222	S2	1500
111	S4	2000
444	S3	1000
444	S1	1000
222	S5	2000

CAND_NO	SUBJECT_NO
111	S1
222	S2
111	S4
444	S3
444	S1
222	S5

SUBJECT_NO	SUBJECT_FEE
S1	1000
S2	1500
S3	1000
S4	2000
S5	2000

3. Third Normal Form

Grades Table:

StudentID	Course	Instructor	Instructor Office
1	Math	Prof. A	Room 101
1	English	Prof. B	Room 102
2	Science	Prof. C	Room 103

Instructor	Instructor Office
Prof. A	Room 101
Prof. B	Room 102
Prof. C	Room 103

OLTP systems are not meant to do reporting?

It will involve a lot of joins

It will overload the OLTP systems

Datawarehouse (DWH) is best fit for reporting purpose (OLAP)

Databases (OLTP)

APPS -> Staging -> Transformations -> DWH -> Data Marts

Flat files

Extract Transform Load

what is a data warehouse?

it's like a Database but the objective is to make your analytical queries faster.

Data Model in a DWH

Dimensional Modeling -

"Dimensional Modeling is a design technique for Databases intended to support end user queries in a DWH"

Ralph Kimball (the data warehouse toolkit by Ralph Kimball)

- the process of modeling a business process into a series of facts and dimension tables designed for analysis

Transactional DB design vs Reporting DB design
==

Transactional DB design

performance - designed towards fast maintainance of data
inserting and updating is quick
very small sets of data is retrieved in a query
Data consistency is critical
Laws of Normalization
Focus is on customers who are entering the data

Reporting DB design

copy of transactional data (not exactly the same way)
as we are not worried about maintainence of data
the resulting model reflects the kind of questions business wants to ask rather than the functions of underlying operational system.
Descriptive data like customer name, customer address is separated from the quantity data such as order quantity, order amount.
larger datsets
insert and update speed is not relevant
performance focus is on retrieving the data quickly.

Features of Dimensional modeling
==================================

=> Data maintainence performance is secondary
=> Data is denormalized to support reporting

what is a fact and what is a dimension?

A fact is a measure, is a measurable metric

order quantity
order amount

total profit

A dimension is something which enhances the fact data

A dimension would be containing 95% of all these columns

what users would want to filter, group, sort on like dates, customer number, store number etc...

example of dimensions

product, customer, store

if you see a integer or a decimal... (it can be possible a fact)

whenever you see a string (dimension)

A customer bought for $1000

$1000 is a fact

who bought is?
where they bought it?
who was the sales person?
when they bought it?

generally there are very less number of facts?

the relationship is between a fact and a dimension

there is no connection between 2 dimensions.

the dimension tables are denormalized

fact table will be a high volume table

300 million active users on amazon (2022)

customers table (300 million)

transactions table (320 billion entries in a year)

=======

what is a surrogate key?

the dimension keys are not to be taken from the source systems (surrogate keys)

surrogate keys are artificial keys generated by you for performing joins.

=> the backend system can change the data

=> you want to take the control of the key

=> you can store your legacy key in dimensional table, but your primary key is a different column.

=> When we have multiple source systems (same key, or different key strucutures)

=> to support SCD

facts never change

dimensions can change slowly

Slowly changing dimensions (SCD)

=========

Arificially created key generally an integer used only by DWH to uniquely identify a row in a dimension table.

why surrogate keys
=================

=> required to implement history of SCD
=> avoid conflicts among backend application keys
=> insulates the DWH from backend application changes

Star Schema

Snowflake Schema

==================

when a dimension relates to another dimension

causes a lot of performance issues (due to more joins)

can be a good fit for OLTP but not for OLAP

Steps for Dimensional Modeling
=================================

1. Choose the business process - Model sales

2. Declare the Grain - what level of detail

orders

order - $1000 , 10
order line item - $80, $120

3. Identify the dimensions

4. Identify the facts

user stories

Sumit Mittal buys a Iphone for $800 which is iphone15 on Jan 28th 2024, at 4 pm via amazon.com using his mastercard to be delivered on Jan 30th 2024 by firstflight courier service.

how?
what?
where?
when?
who?
how much? $800
why?

Client Dimension

client_key	client_id	Name	City	Sector	Profession
101	7892	Sumit Mittal	Bangalore	IT	Educator

SCD (slowly changing dimension)

SCD 0 - never changes

SCD 1 - overwrite, easy to implement, lose the history

SCD 2 - maintain full history

client_key	client_id	Name	City	Sector	Profession	start_date	end_date
101	7892	Sumit Mittal	Bangalore	IT	Educator	1st jan 2013	31st Dec 2016
102	7892	Sumit Mittal	Hyderabad	IT	Educator	1st jan 2017	31st Dec 2018
103	7892	Sumit Mittal	Pune	IT	Educator	1st jan 2019	Null

SCD 3 - Partial history, keep extra column to store previous value, little easy to implement, limited history.

client_key	client_id	Name	previous_city	current_City	Sector	Profession
101	7892	Sumit Mittal	Hyderabad	Pune	IT	Educator

what if your dimension is very frequently changing

like once every day

then its better you take daily production snapshot

monthly snapshot

Facts
======

but the volume is very high

300 million * 20 = 6 billion

60 billion * 5 = 300 billion rows

you can think of the right grain

13th feb 2014
sumit mittal 100
sumit mittal 200
sumit mittal 300
sumit mittal 100
sumit mittal 200

aggregation of orders per day per user.

sumit mittal, 5, 900

Fact - Dimension

Join

Wide transformation (Shuffle)

32 buckets - fact

4 buckets - dimension

One Big Table
===============

One big table is a concept that has gained popularity in recent years. The idea is to store all data in one single massive table.

Advantages -

Improved query performance
Reduced development and maintainence effors
Simplified data model

Disadvantage -

Increased storage requirement
complexity in data updates

row based file formats

id name age salary id name age salary id name age salary id name age salary
id name age salary

column based file formats

id
5
7
9
2

names
sumit
kapil
rahul
sachin

age
30
31
32
33

salary
10000
20000
30000
40000

sumit mittal
sumit mittal
sumit mittal
kapil Prasad
kapil Prasad

Sumit Mittal 3
Kapil Prasad 2

SCD Implementation

/user/itv005857/scd_demo
/user/itv005857/scd_demo/source
/user/itv005857/scd_demo/target

Customer Dimension

```
CustomerID,FirstName,LastName,Email,Phone,Address,City,State,ZipCode
1,John,Doe,johndoe@email.com,555-1234,123 Main St,Anytown,CA,12345
2,Jane,Smith,janesmith@email.com,555-5678,456 Oak Ave,Sometown,NY,67890
3,Robert,Johnson,robertjohnson@email.com,555-8765,789 Pine Ln,Othercity,TX,34567
4,Alice,Williams,alicewilliams@email.com,555-4321,234 Cedar Dr,Yourtown,FL,89012
5,Michael,Brown,michaelbrown@email.com,555-9876,567 Elm Blvd,Theirtown,IL,45678
6,Emily,Miller,emilymiller@email.com,555-6543,890 Birch Rd,Newcity,WA,23456
7,David,Jones,davidjones@email.com,555-2345,678 Maple Ave,Yourcity,GA,78901
8,Sarah,Anderson,sarahanderson@email.com,555-5432,901 Pine St,Heretown,OH,56789
9,Christopher,Taylor,christophertaylor@email.com,555-8765,234 Oak Ln,Thistown,PA,12345
10,Olivia,Clark,oliviaclark@email.com,555-3456,567 Cedar Ave,Thatcity,TN,67890
```

```
CustomerID,FirstName,LastName,Email,Phone,Address,City,State,ZipCode
1,John,Doe,johndoe@gmail.com,555-1234,123 Main St,Anytown,CA,12345
2,Jane,Smith,janesmith@email.com,555-5679,456 Oak Ave,Sometown,NY,67890
3,Robert,Johnson,robertjohnson@email.com,555-8765,123 Elm Ln,Harborcity,FL,87654
4,Alice,Williams,alicewilliams@email.com,555-4321,234 Cedar Dr,Yourtown,FL,89012
5,Michael,Brown,michaelbrown@email.com,555-9876,567 Elm Blvd,Theirtown,IL,45678
6,Emily,Miller,emilymiller@email.com,555-6543,890 Birch Rd,Newcity,WA,23456
7,David,Jones,davidjones@email.com,555-2345,678 Maple Ave,Yourcity,GA,78901
8,Sarah,Anderson,sarahanderson@email.com,555-5432,901 Pine St,Heretown,OH,56789
```

```
9,Christopher,Taylor,christophertaylor@email.com,555-8765,234 Oak Ln,Thistown,PA,12345
11,Grace,Turner,graceturner@email.com,555-1122,567 Oak St,Cityview,CA,98765
12,Connor,Evans,connorevans@email.com,555-2233,890 Pine Ave,Townsville,TX,54321
```

SCD Type 2 implementation in pyspark
=====================================

Customer Dimension

```
CustomerID,FirstName,LastName,Email,Phone,Address,City,State,ZipCode
1,John,Doe,johndoe@email.com,555-1234,123 Main St,Anytown,CA,12345
2,Jane,Smith,janesmith@email.com,555-5678,456 Oak Ave,Sometown,NY,67890
3,Robert,Johnson,robertjohnson@email.com,555-8765,789 Pine Ln,Othercity,TX,34567
4,Alice,Williams,alicewilliams@email.com,555-4321,234 Cedar Dr,Yourtown,FL,89012
5,Michael,Brown,michaelbrown@email.com,555-9876,567 Elm Blvd,Theirtown,IL,45678
6,Emily,Miller,emilymiller@email.com,555-6543,890 Birch Rd,Newcity,WA,23456
7,David,Jones,davidjones@email.com,555-2345,678 Maple Ave,Yourcity,GA,78901
8,Sarah,Anderson,sarahanderson@email.com,555-5432,901 Pine St,Heretown,OH,56789
9,Christopher,Taylor,christophertaylor@email.com,555-8765,234 Oak Ln,Thistown,PA,12345
10,Olivia,Clark,oliviaclark@email.com,555-3456,567 Cedar Ave,Thatcity,TN,67890
```

```
CustomerID,FirstName,LastName,Email,Phone,Address,City,State,ZipCode
1,John,Doe,johndoe@gmail.com,555-1234,123 Main St,Anytown,CA,12345
2,Jane,Smith,janesmith@email.com,555-5679,456 Oak Ave,Sometown,NY,67890
3,Robert,Johnson,robertjohnson@email.com,555-8765,123 Elm Ln,Harborcity,FL,87654
4,Alice,Williams,alicewilliams@email.com,555-4321,234 Cedar Dr,Yourtown,FL,89012
5,Michael,Brown,michaelbrown@email.com,555-9876,567 Elm Blvd,Theirtown,IL,45678
6,Emily,Miller,emilymiller@email.com,555-6543,890 Birch Rd,Newcity,WA,23456
```

```
7,David,Jones,davidjones@email.com,555-2345,678 Maple Ave,Yourcity,GA,78901
8,Sarah,Anderson,sarahanderson@email.com,555-5432,901 Pine St,Heretown,OH,56789
9,Christopher,Taylor,christophertaylor@email.com,555-8765,234 Oak Ln,Thistown,PA,12345
11,Grace,Turner,graceturner@email.com,555-1122,567 Oak St,Cityview,CA,98765
12,Connor,Evans,connorevans@email.com,555-2233,890 Pine Ave,Townsville,TX,54321
```

updates (1,2,3)
insert (11,12)
delete (10)
unchanged (all other records)

```
/user/itv005857/scd_demo
/user/itv005857/scd_demo/source
/user/itv005857/scd_demo/target
```

```
hadoop fs -put customers.csv /user/itv005857/scd_demo/source
```

effective start date	end date	active_flag
1st jan 2013	31st dec 2017 (history)	false
31st dec 2017	31st dec 9999 (current)	true

```
customers_source_schema = "customerid long,firstname string, lastname string, email string, phone string, address string, city string, state string, zipcode long"
```

```
customers_target_schema = "customerid long,firstname string, lastname string, email string, phone string, address string, city string, state string, zipcode long, customer_skey long, effective_date date, end_date date, active_flag boolean"
```

```
customers_source_df = spark.read \
.format("csv") \
.option("header",True) \
.schema(customers_source_schema) \
.load("/user/itv005857/scd_demo/source")
```

row_num

target - DWH

10 records with 4 extra columns

I am dividing this into 2 dataframes based on active_flag

true - active_customers_target_df

false - inactive_customers_target_df

10 records in the source - 9 columns

10 records in the target - 9 columns + 4 additional columns

surrogate key
effective date
end date
active flag

join

active_customers_target_df (DWH only active records)

customers_source_df (complete source dataframe)

if it's null in target and not null in source then it means a insert should happen

Insert
target - null
source - not null

Delete
target - not null
source - null

Updates

we have to check all the 6 keys if there is any change
we can take a hash of the 6 keys (single big string)

if the hash key is different then we have to update

if the hash key in both source and target is same then no change

INSERT
UPDATE
DELETE
NO CHANGE

column_renamer(customers_source_df, "_source", True):

firstname_source

column_renamer(df, suffix, append)

get_hash(df, keys_list)

active_customers_target_df_hash =

column_renamer(get_hash(active_customers_target_df,
slowly_changing_cols), suffix="_target", append=True)

customers_source_df_hash =

column_renamer(get_hash(customers_source_df, slowly_changing_cols),
suffix="_source", append=True)

|customerid_source firstname_source lastname_source
email| phone| address| city|state|zipcode| hash_md5

target_active

updates?

end date the previous record
insert a new record