

CIS 9440 - Data Warehousing and Analytics

Class #4

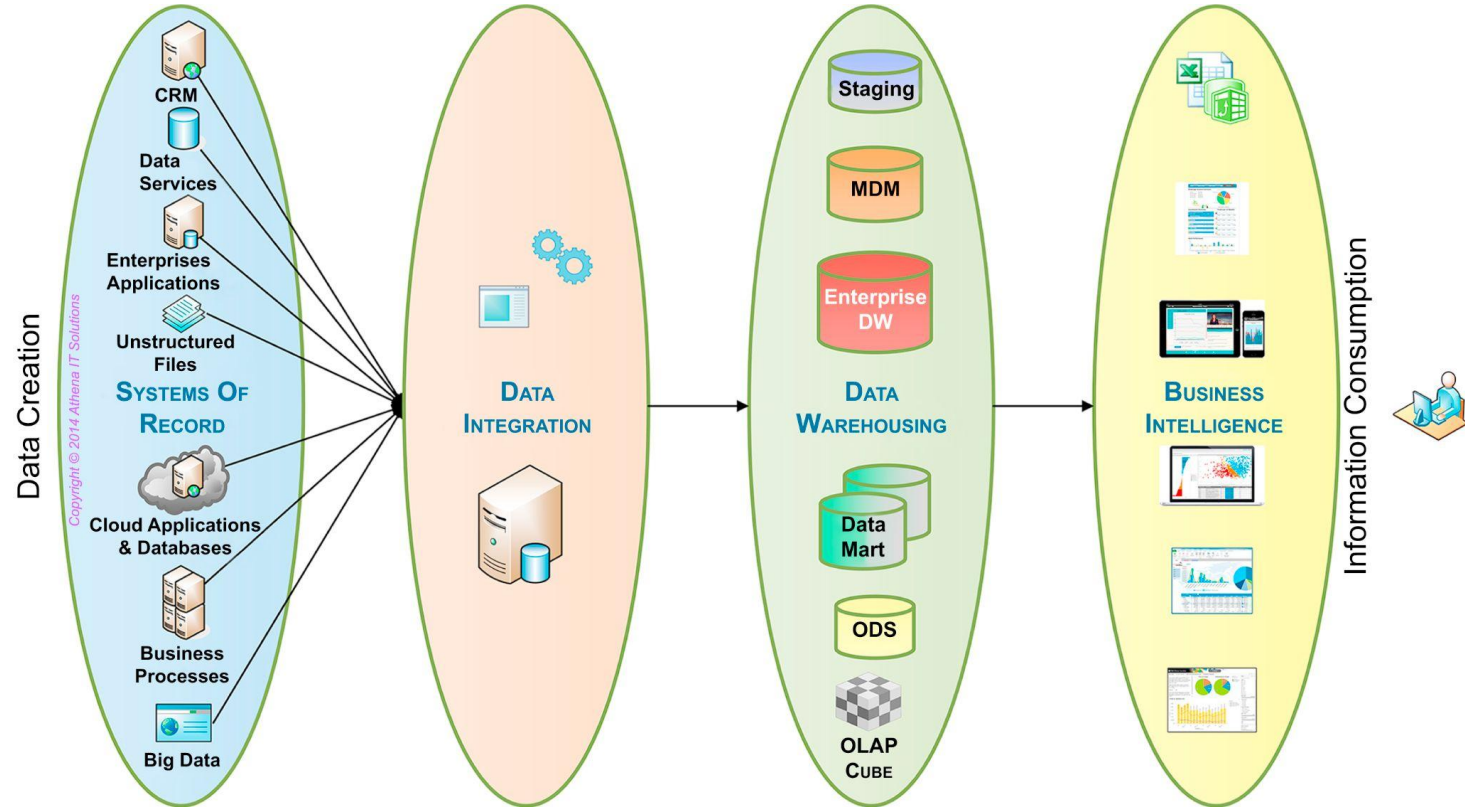




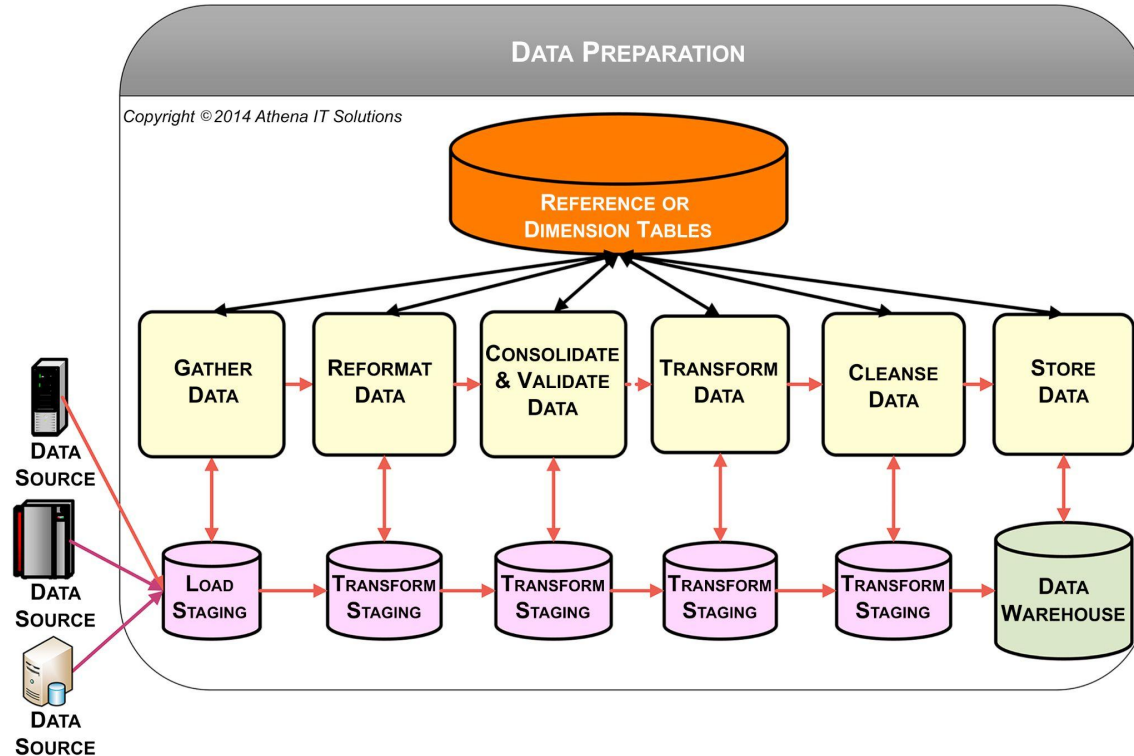
What's on Blackboard?

- HW #1, due at 6:00 PM on Wednesday, 9/28
- Final Project Milestone #1, due at 11:59 PM on Wednesday, 9/28
- Download Tableau
 - If you do not already have Tableau: [link](#)
 - If your student licence expired, you can request a new one here: [link](#)
 - If your license is still valid, you're all set

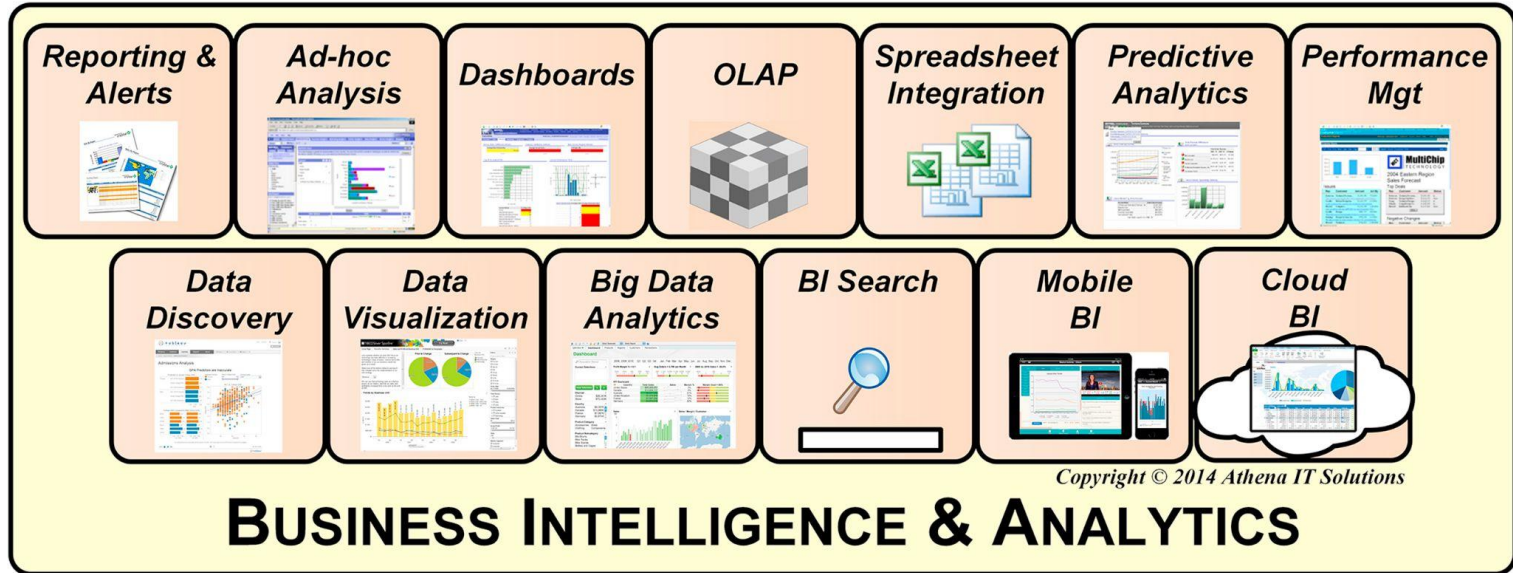
Textbook Reading Highlights:



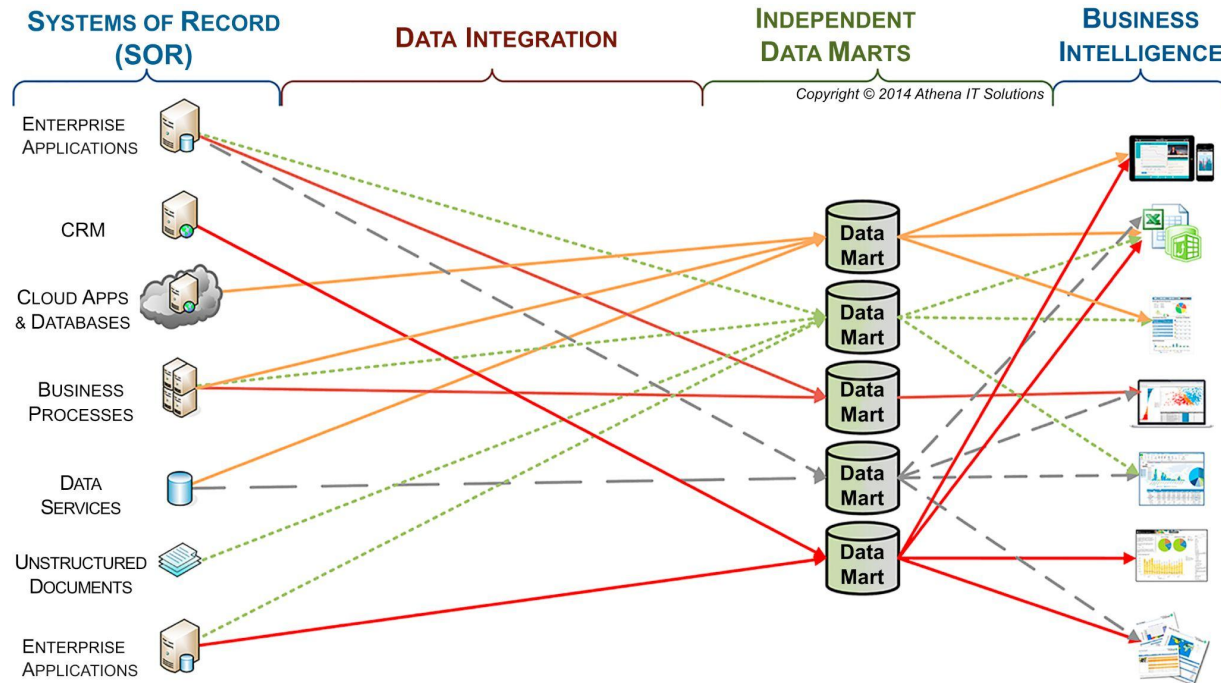
Textbook Reading Highlights:



Textbook Reading Highlights:




Textbook Reading Highlights:






Week 4 Class Overview:

1. Analytical SQL (continued)
 2. Final Project Milestone #1 Check-in
 3. Introduction to Dimensional Modeling
- 



Week 4 Class Overview:

- 1. Analytical SQL (continued)**
 2. Final Project Milestone #1 Check-in
 3. Introduction to Dimensional Modeling
- 



CASE statement in SQL

If you want an IF, ELSE condition in SQL, you use a CASE statement



CASE statement

Create an IF statement in SQL

```
SELECT gameId,  
homeTeamName,  
CASE  
  WHEN duration_minutes > 180 THEN "over 3 hours"  
  WHEN duration_minutes <= 180 THEN "3 hours of less"  
  ELSE "Unsure of game duration..."  
END AS game_duration  
FROM `bigquery-public-data.baseball.schedules`
```



CASE statement

Create an IF statement in SQL

```
SELECT market,  
venue_name,  
CASE  
  WHEN venue_capacity > 15000 THEN 'large arena'  
  WHEN venue_capacity > 7500 THEN 'medium arena'  
  ELSE 'small arena'  
END AS arena_size  
FROM `bigquery-public-data.ncaa_basketball.mbb_teams`;
```



CASE statement example 2

```
SELECT a.gametime,  
a.market,  
CASE  
  WHEN a.attendance > b.venue_capacity*0.9 THEN 'Max Attendance'  
  ELSE 'Empty Seats'  
END AS attendance_level  
FROM `bigquery-public-data.ncaa_basketball.mbb_teams_games_sr` a  
JOIN `bigquery-public-data.ncaa_basketball.mbb_teams` b  
ON a.market = b.market  
where a.home_team = True;
```



CASE statement example 3

```
SELECT a.gametime,  
a.market,  
CASE  
  WHEN a.attendance > b.venue_capacity*0.9 THEN 'Max Attendance'  
  ELSE 'Empty Seats'  
END AS attendance_level  
FROM `bigquery-public-data.ncaa_basketball.mbb_teams_games_sr` a  
JOIN `bigquery-public-data.ncaa_basketball.mbb_teams` b  
ON a.market = b.market  
where a.home_team = True;
```

Practice Question:

(5 minutes)

Using the schedules table from the baseball dataset, write a query that returns the `gameId`, `homeTeamName`, and a case statement that returns "high attendance" if the `attendance` is above the average and "low attendance" is the `attendance` is below the average.



Subquery in SQL

In SQL, a Subquery is a query nested within a query



Subquery

```
SELECT gameId,  
homeTeamName,  
awayTeamName,  
duration_minutes  
FROM `bigquery-public-data.baseball.schedules`  
WHERE attendance >  
    (SELECT AVG(attendance)  
     FROM `bigquery-public-data.baseball.schedules`);
```




Subquery

```
SELECT homeTeamName,  
COUNT(gameId) AS number_of_games,  
FROM `bigquery-public-data.baseball.schedules`  
WHERE attendance >  
      (SELECT AVG(attendance)  
       FROM `bigquery-public-data.baseball.schedules`)  
GROUP BY homeTeamName  
ORDER BY number_of_games DESC;
```

Practice Question:

(5 minutes)

Using the schedules table from the baseball:
Write a query (that uses a subquery) to return
the homeTeamName, awayTeamName,
dayNight, duration_minutes, and
attendance of the game with the highest
attendance in the table.



Week 4 Class Overview:

1. Analytical SQL (continued)
 - 2. Final Project Milestone #1 Check-in**
 3. Introduction to Dimensional Modeling
- 



First, use NYC Open Data datasets to drive the opportunity you're aiming for

- Look at many datasets
 - Which interest you?
 - Which may be combined to uncover interesting insights?
 - Let's look at a couple!



Second, do you Business and Technical Justification?

- Does your project have a clear objective?
- Do you have the data to reach that objective?




Third, do your KPI's accomplish your objective?

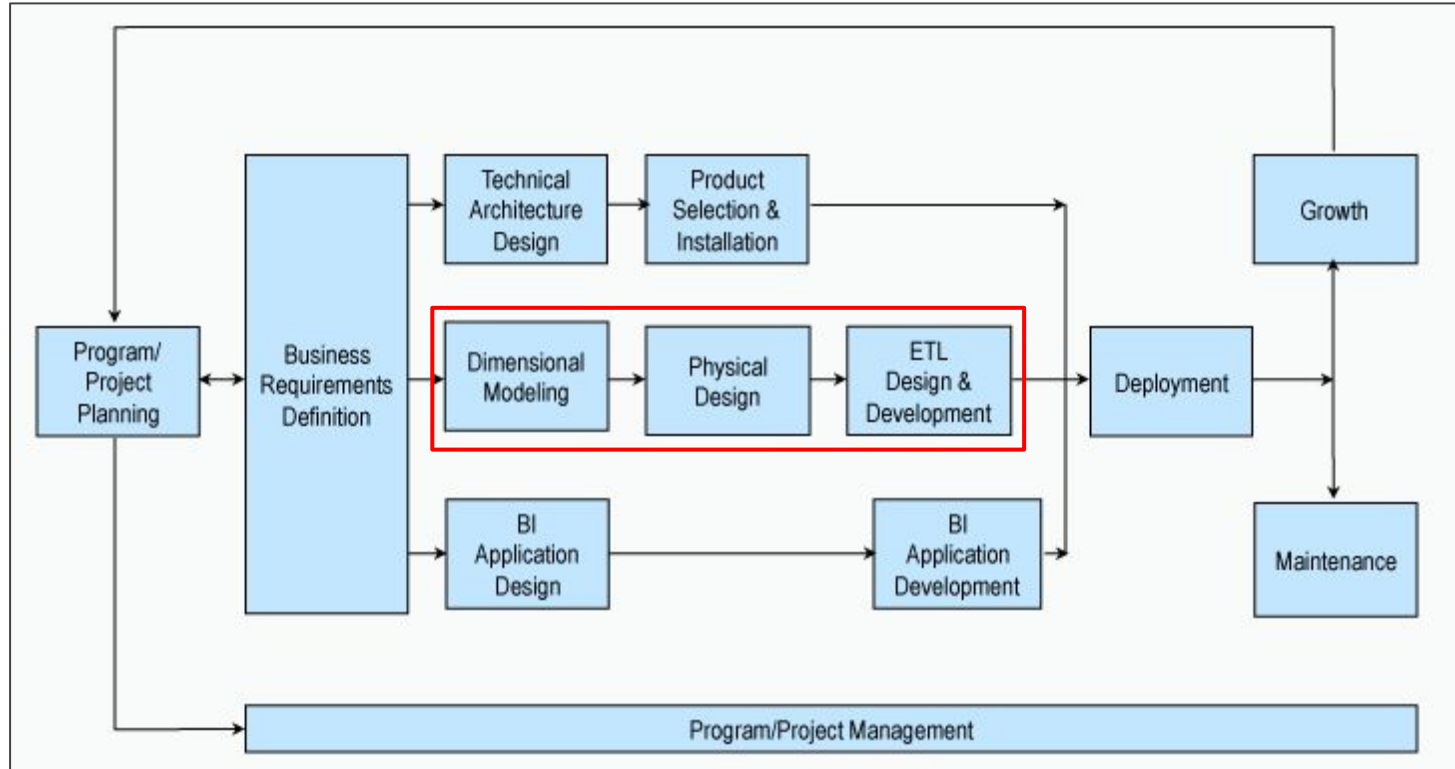
- If you could see the outcome of your KPI's, would that accomplish your project's goal?



Week 4 Class Overview:

1. Analytical SQL (continued)
 2. Final Project Milestone #1 Check-in
 - 3. Introduction to Dimensional Modeling**
- 

Where is Dimensional Modeling in the Kimball Lifecycle?





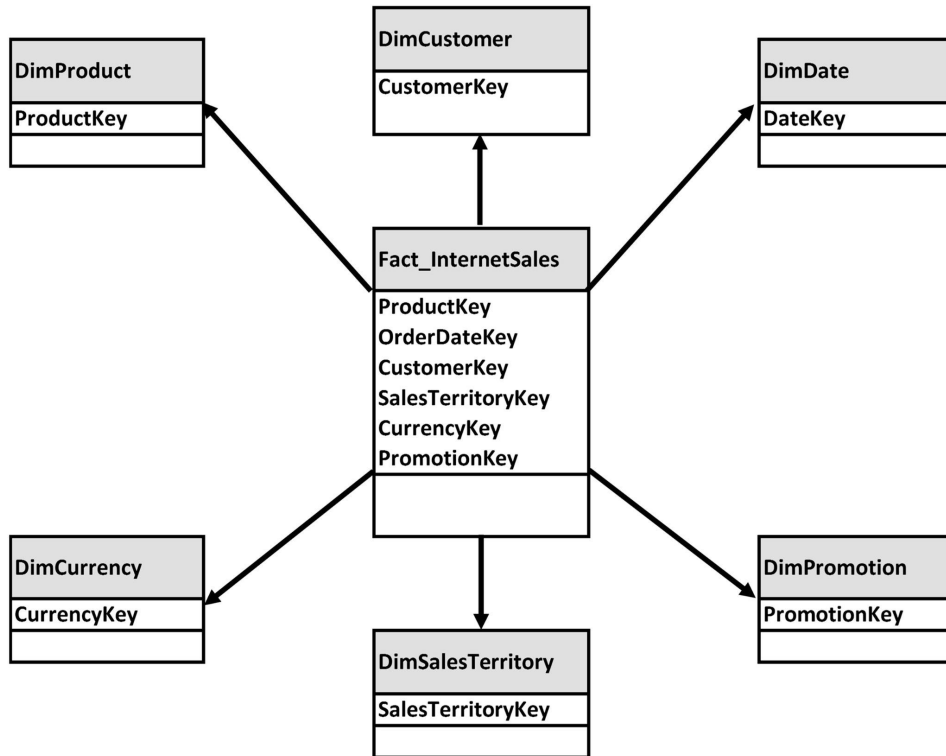
What is a Dimension Modeling?

Dimensional Modeling is a *logical* design technique optimal for **Business Intelligence applications**.

Other common data modeling techniques are ER modeling, and relationship modeling.

What does a Dimensional Model look like?

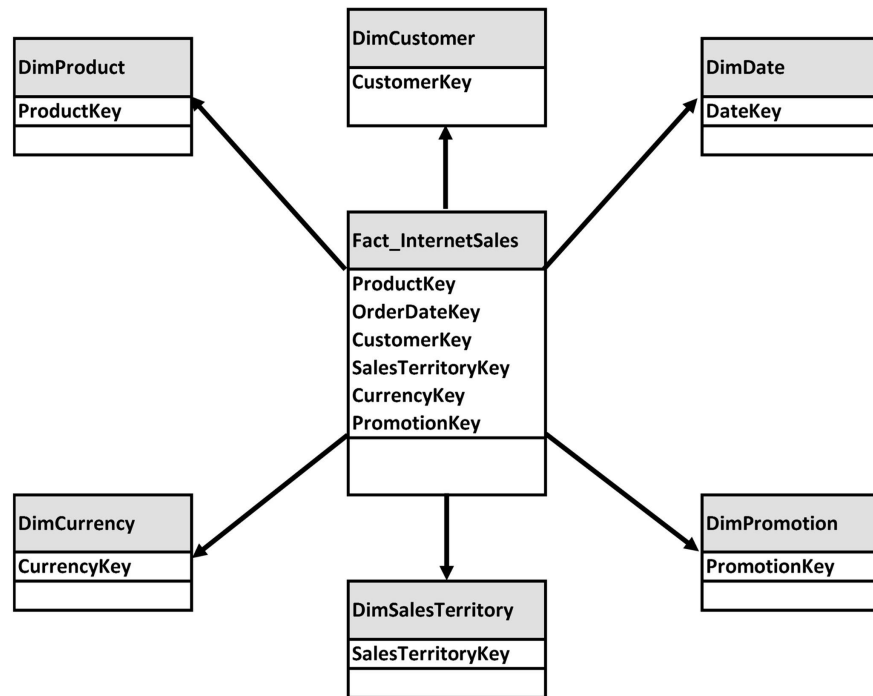
Dimensional Models are purposefully simple, consisting of **facts, dimensions, and attributes.**





We already have data in ERD models, why implement Dimension Modeling?

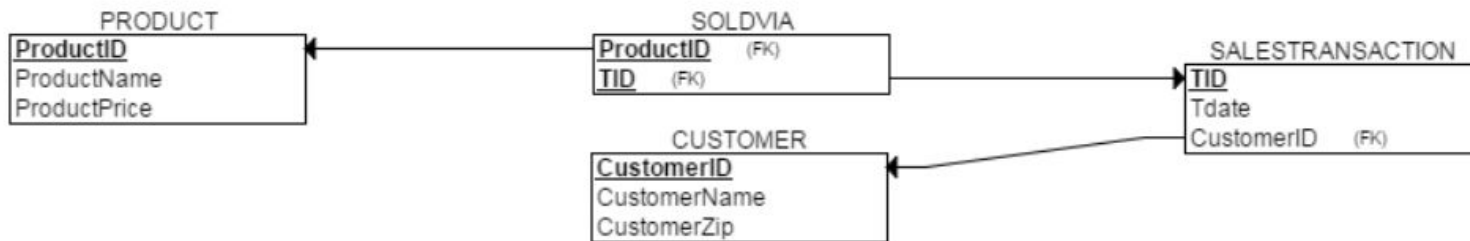
- Dimensional Models are structured for Data Analysis:
 - Very few joins between your values and context
 - Easy to update when hierarchies change
 - Fast, so you can quickly slice-and-dice your data to analyze many ways
 - “Conformed Dimensions” ensure data integrity
- What are ERD Models better at?
 - Normalization



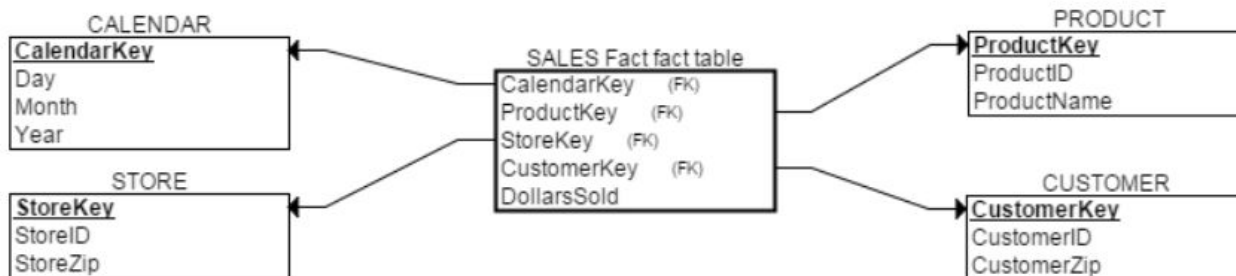


Dimensional Modeling vs ER Diagram

ER
Diagram



Dimensional
Model





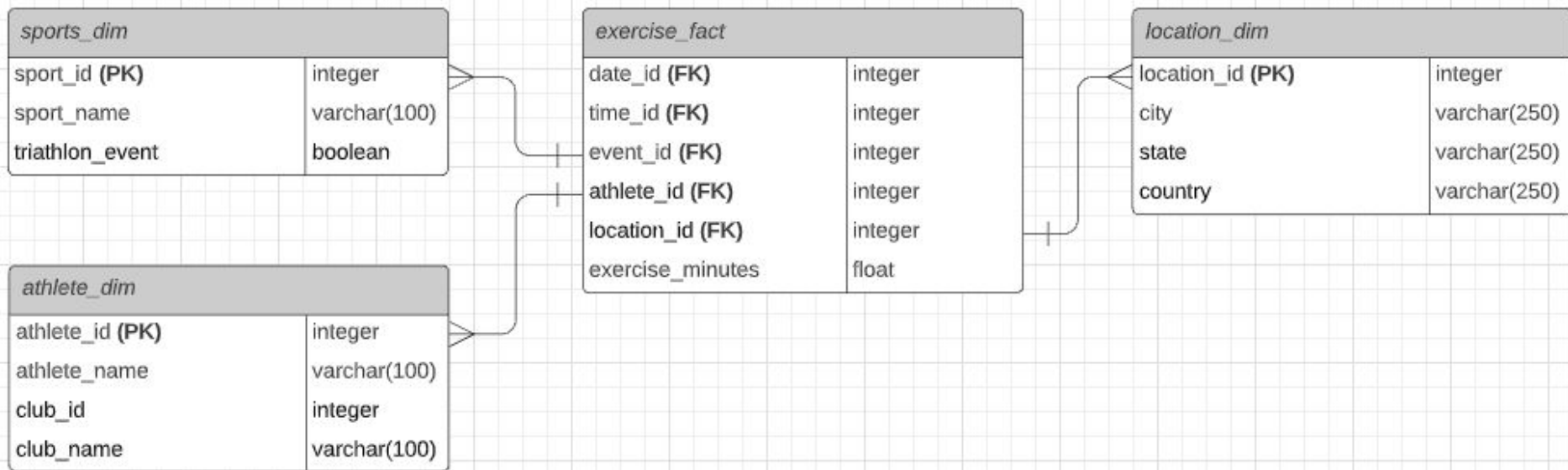
How do we start Dimension Modeling?

- We follow the Kimball Lifecycle! The previous step is to gather your KPI's
 - Dimensional Modeling flows naturally from your KPI's
 - For example, KPI's:
 - Total Exercise Minutes per Sport
 - Average Exercise Minutes per Athlete
 - Total Exercise Minutes per State
 - Fact: Exercise Minutes
 - Dimensions: Sports, Athletes, Location

- KPI's: Total Exercise Minutes per Sport and Average Exercise Minutes per Athlete and Total Exercise Minutes by State



- **Fact:** Exercise Minutes
- **Dimension:** Sports
- **Dimension:** Athletes
- **Dimension:** Location





Dimensions vs Facts

Fact: a measurement of a business activity, a specific occurrence of transaction. These are (generally) numeric.

- Inventory level
- An online sale
- A book checked out from the library
- A single tweet
- A recorded exercise

Dimension: an entity that establishes context for a measurement. The who, what, where, and why.

- Geography
- Customers
- Dates
- Stores



Dimensions vs Facts (simplified)

Fact: values

Dimensions: Context: who, what, where, how, and why.

- Sales per City
- Inventory Level per Week
- Price per Product Category
- House Cost per Realtor

Dimensions vs Facts (continued)

DIMENSIONS

Top Salespeople



Top Resellers

BusinessType	ResellerName	Value Added Reseller
	Brakes and Gears	\$877,107
	Excellent Riding Supplies	\$853,849
	Totes & Baskets Company	\$816,756
	Corner Bicycle Supply	\$787,773
	Thorough Parts and Repa..	\$740,986
Warehouse	Vigorous Exercise Compa..	\$841,909
	Retail Mall	\$799,278
	Outdoor Equipment Store	\$746,318
	Health Spa, Limited	\$730,799
	Fitness Toy Store	\$727,273

Reseller \$ by Country

SalesTerritor..	2011	2012	2013	2014
Australia			\$701,483	\$720,294
Canada	\$1,511,332	\$2,237,913	\$1,378,310	\$570,414
France		\$848,354	\$2,286,719	\$1,374,860
Germany			\$961,126	\$829,512
United States	\$5,106,532	\$12,625,496	\$10,678,412	\$5,596,123

Reseller Business Type



DIMENSIONS

Dimensions vs Facts (continued)

The screenshot shows an Excel PivotTable titled "Internet Sales Amount". The PivotTable Fields task pane on the right shows the following configuration:

- Filters:** Date.Calendar
- Columns:** (Empty)
- Rows:** Sales Territory, Product Categories
- Values:** Internet Sales Amount

Annotations on the screenshot:

- A red dashed arrow points from the text "DIMENSIONS" to the "Sales Territory" and "Product Categories" fields in the Rows area.
- A green dashed arrow points from the text "FACTS" to the "Internet Sales Amount" field in the Values area.

Row Labels	CY 2005	CY 2006	CY 2007	CY 2008	Grand Total
Europe	\$709,947.20	\$1,627,759.71	\$3,382,979.27	\$3,209,356.08	\$8,930,042.26
Accessories			\$84,458.16	\$117,811.25	\$202,269.41
Bikes	\$709,947.20	\$1,627,759.71	\$3,262,529.32	\$3,044,696.49	\$8,644,932.72
Mountain Bikes	\$128,749.62	\$381,595.88	\$1,211,932.53	\$1,343,763.94	\$3,066,041.98
Road Bikes	\$581,197.58	\$1,246,163.83	\$1,522,513.66	\$940,618.37	\$4,290,493.44
Touring Bikes			\$528,083.13	\$760,314.18	\$1,288,397.31
Clothing			\$35,991.79	\$46,848.34	\$82,840.13
North America	\$1,247,379.26	\$2,748,298.93	\$3,374,296.82	\$3,997,659.37	\$11,367,634.37
Accessories			\$151,870.08	\$207,929.84	\$359,799.92
Bikes	\$1,247,379.26	\$2,748,298.93	\$3,148,783.82	\$3,676,699.92	\$10,821,161.92
Mountain Bikes	\$152,474.55	\$528,881.06	\$1,621,726.41	\$1,729,816.12	\$4,032,898.14
Road Bikes	\$1,094,904.71	\$2,219,417.87	\$994,096.19	\$917,123.42	\$5,225,542.18
Touring Bikes			\$532,961.22	\$1,029,760.38	\$1,562,721.60
Clothing			\$73,642.92	\$113,029.61	\$186,672.53
Pacific	\$1,309,047.20	\$2,154,284.88	\$3,033,784.21	\$2,563,884.29	\$9,061,000.58
Accessories			\$57,381.47	\$81,309.16	\$138,690.63
Bikes	\$1,309,047.20	\$2,154,284.88	\$2,947,789.48	\$2,440,928.44	\$8,852,050.00
Mountain Bikes	\$304,749.10	\$651,979.82	\$1,155,979.53	\$741,111.00	\$2,853,819.45
Road Bikes	\$1,004,298.10	\$1,502,305.07	\$1,435,419.37	\$1,062,525.88	\$5,004,548.42
Touring Bikes			\$356,390.58	\$637,291.56	\$993,682.14
Clothing			\$28,613.26	\$41,646.69	\$70,259.95
Grand Total	\$3,266,373.66	\$6,530,343.53	\$9,791,060.30	\$9,770,899.74	\$29,358,677.22

- Facts are the **values**
- Dimensions are the **context**

Label the following as Fact or Dimension:

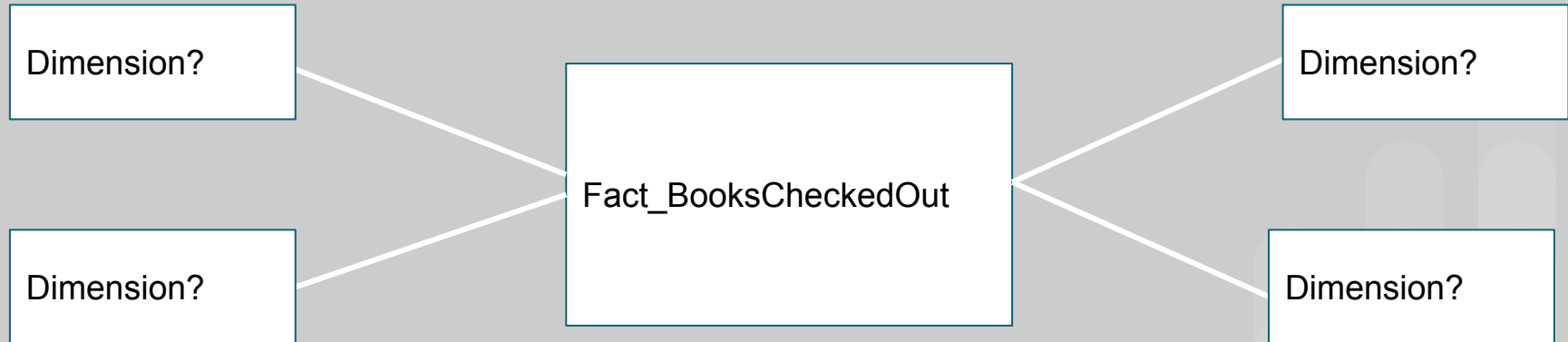
Poll

1. Sales dollars
2. Internet Clicks
3. Daily High Temperature
4. Country
5. Time
6. Date
7. Customer
8. Color

Let's practice building a Dimensional Model:

We're building a Data Warehouse for the NYC Library System to analyze checked out books. Our first fact table is "Books Checked Out"

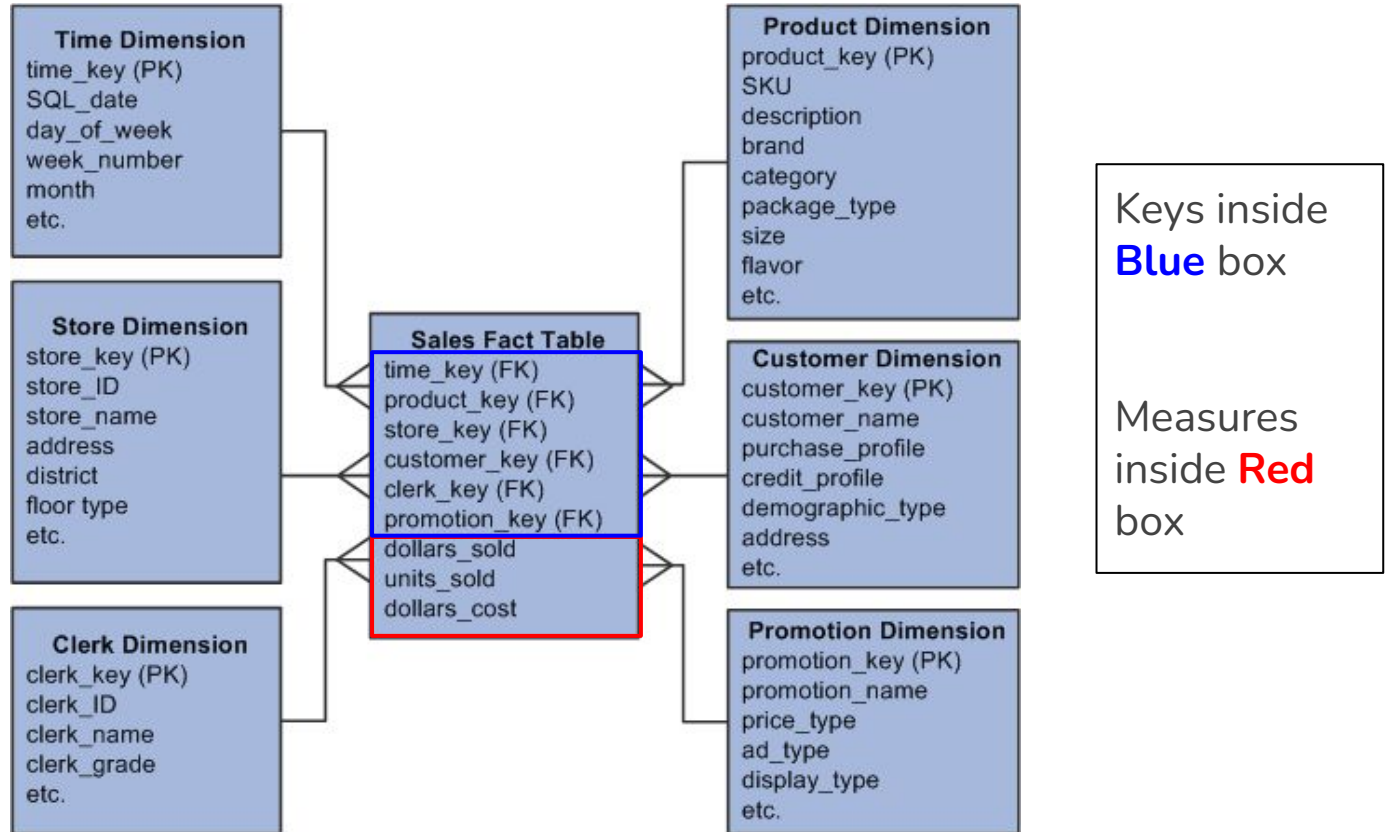
What are some possible Dimensions that could add context to our Fact Table? See how many you can add!



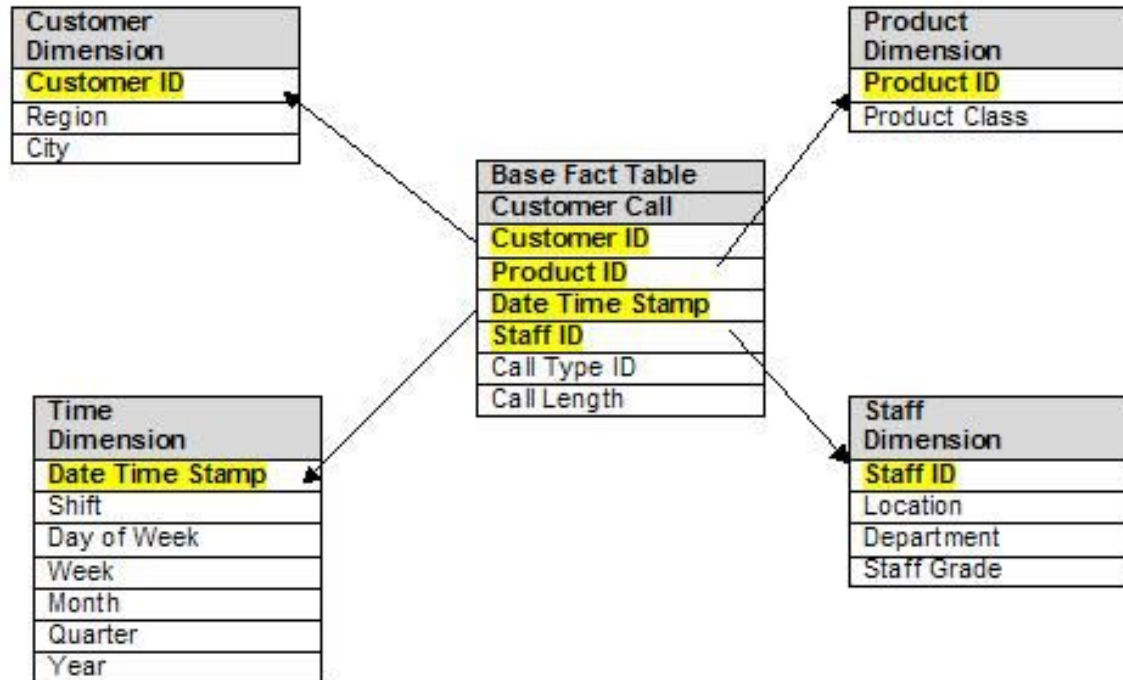


Fact Tables

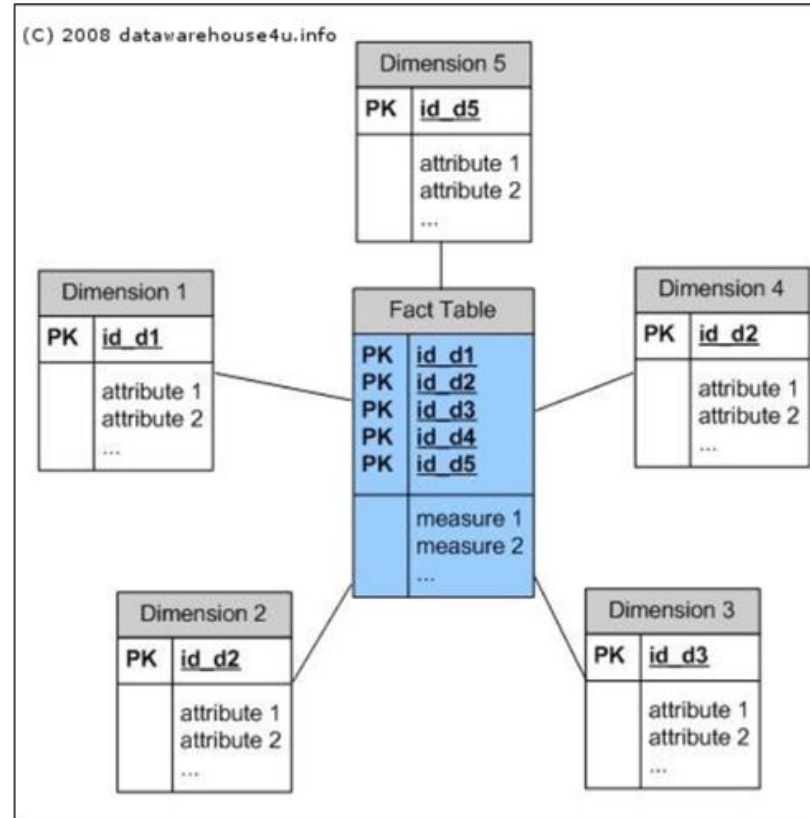
Fact Table Components - Keys and Measures



Fact Table Components - Keys and Measures



Fact Table - Primary Key





Fact Tables - Define the Grain

- **Grain:** level of detail in the measurement of an event, such as a unit of measure
- Examples, related to sales at Barnes and Noble:
 - Sales by transaction by product by store
 - Sales by Day by Product by Store
 - Sales by Week by product by Store
 - Sales by Week by Store



Fact Tables - Grain

Sales by transaction by Day by Store - Transactional Fact Table

date_id	time_id	store_id	product_id	sales_amount	sales_quantity
20210201	1058	74	58285	\$ 28	1
20210201	1058	27	55101	\$ 44	2
20210201	1059	53	73784	\$ 48	2
20210201	1100	89	78230	\$ 20	2
20210201	1104	82	28918	\$ 22	1
20210201	1104	42	77802	\$ 23	1
20210201	1106	14	11354	\$ 34	2
20210201	1107	48	76993	\$ 24	1
20210201	1109	92	34019	\$ 16	1
20210201	1115	39	25483	\$ 25	1
20210201	1116	44	65516	\$ 11	1
20210201	1116	99	11589	\$ 26	1
20210201	1121	73	14127	\$ 30	3
20210201	1125	54	33436	\$ 68	2



Fact Tables - Grain

Sales by Product by Day by Store - Periodic Fact Table

date_id	store_id	product_id	sales_amount	sales_quantity
20210201	57	84929	\$ 735	21
20210201	57	22395	\$ 341	11
20210201	57	82688	\$ 210	14
20210201	17	43619	\$ 336	24
20210201	17	20237	\$ 527	17
20210201	17	61555	\$ 336	16
20210201	17	62843	\$ 351	13
20210201	17	49042	\$ 432	16
20210201	17	67785	\$ 396	12
20210201	78	87942	\$ 360	18
20210201	78	35479	\$ 195	15
20210201	78	69936	\$ 696	24
20210201	78	29242	\$ 510	17
20210201	78	48111	\$ 325	25



Fact Tables - Grain

Sales by Day by Store - Periodic Fact Table

date_id	store_id	sales_amount	sales_quantity
20210201	57	\$ 2,184	182
20210202	57	\$ 11,970	399
20210203	57	\$ 4,446	234
20210204	57	\$ 2,926	133
20210201	17	\$ 3,496	184
20210202	17	\$ 10,362	314
20210203	17	\$ 6,622	301
20210204	17	\$ 10,268	302
20210201	78	\$ 5,940	297
20210202	78	\$ 6,422	247
20210203	78	\$ 8,760	292
20210204	78	\$ 3,000	100
20210205	78	\$ 3,675	147
20210206	78	\$ 6,832	427



Fact Tables - Grain

Customer Sales by Order - Accumulating Fact Table

customer_id	order_id	order_date	ship_date	receive_date	sales_amount	sales_quantity
209861	3946	20210201	20210204	20210210	\$ 20	2
603654	3153	20210207	20210208	20210215	\$ 30	1
240988	1604	20210207	20210209	20210214	\$ 24	1
557310	1007	20210210	20210212		\$ 28	2
860448	7076	20210211			\$ 20	1



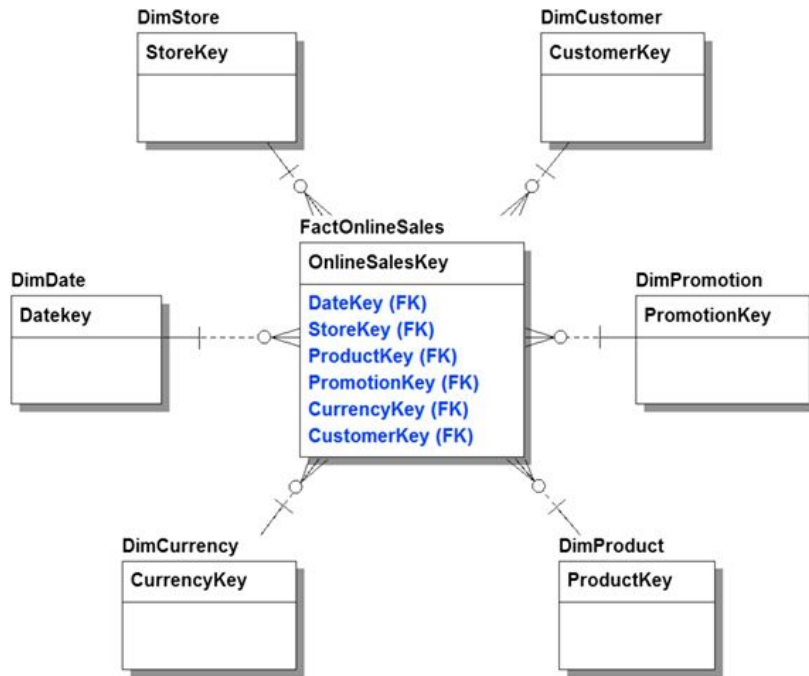
Fact Tables - Transaction, Periodic, Accumulating

Comparison of Fact Tables Types			
Property	Transaction	Periodic	Accumulating
Grain	One row per transaction	One row per time period	One row per lifetime of an event
Date dimension	Lowest level of granularity	End-of-period granularity	Multiple per row
Number of dimensions	Least	Average	Most
Facts	Transaction related	Period related	Numerous events over lifetime
Measurement	Additive	Not additive, average	Need to derive
Conforming dimensions	Yes	Yes	Yes
Conforming facts	Yes	Yes	Yes
Database size	Largest	Smaller	Smallest

Why not always use transaction rather than periodic?

Inmon chooses periodic while Kimball prefers transaction.

Reminder - Fact Table is Center of your Dimensional Model



You are still designing a Data Warehouse for NYC Public Libraries. The Data Warehouse will be used to run analyses on books checked out.

- If you define the Fact Table as:
Books checked out by Library by Day

Is that a transaction, periodic, or accumulating fact table?



Fact Tables - Type of Measurements

- **Additive**: a measures that can be added across ALL dimensions.
- **Semi-additive**: measurements that can be added across SOME dimensions but NOT others.
- **Non-additive**: measures that can't be added across ANY dimension whatsoever.



Type of Measurements - Additive

- **Additive:** Easiest one; It's a measures that can be added across ALL dimensions.
 - Example, number of items you bought in an online store. You can add this across customer, store, product, date. It makes sense any way you aggregate.
 - What would be another example?



Type of Measurements - Semi-additive

- **Semi-additive:** measurements that can be added across SOME dimensions but NOT others.
 - Example, Bank Account Balances. Does it make sense to add a bank balance across all 12 months? No. But you could add all customers balances for one month to get the total balance at the bank.
 - What would be another example?



Type of Measurements - Non-additive

- **Non-additive:** measures that can't be added across ANY dimension whatsoever.
 - Example, unit prices, ratios, temperatures; Does it make sense to add temperatures across dimensions? No.
 - What would be another example?

Label the following facts as additive, semi-additive, or non-additive:

1. Sales Dollars
2. End of Week Inventory Levels
3. Discount Percentage
4. Sales Units

Dimensional Modeling practice 2:

We're building a Data Warehouse for Netflix to better manage their streaming movies.

Given the following KPI's, attempt to fill out the needed Fact table and Dimension Tables in a [Dimensional Model](#):

1. Total Minutes Watched per Genre
2. Streams per Movie
3. Total Minutes Watched by Hour of Day
4. Average Streams per Month

Homework:

1. HW #1, due at 6:00 PM on Wednesday, 9/28
2. Final Project Milestone #1, due at 11:59 PM on Wednesday, 9/28
3. Reading - Business Intelligence Guidebook Chapters 9, 10