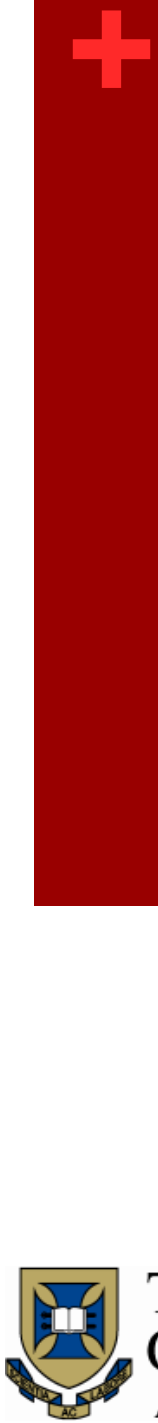
Tutorial 4:
Data Warehousing Design



+ Question 1

- Suppose that a data warehouse for *AllElectronics* consists of the following three dimensions ***time***, ***item***, and ***location***, and one measure ***sales***. The dimension hierarchies are as following:

Dimensions

Time
Day
Month
Quarter
Year

Item
Item Name
Brand
Type

Location
Street
City
Province or State
Country

Fact table

Time
Item
Location
Sales

Q1 Fact Table and Dimensions

Fact table

Time	Item	Location	Sales
12/2/17	INS157700HQ	JB Hi-Fi Garden City	25
15/5/17	IP6P128AU	Best Buy Streeterville	10

Dimensions

Time	Day	Month	Quarter	Year
12/2/17	12	2	Q1	2017
15/5/17	15	5	Q2	2017

Item	Item Name	Brand	Type
INS157700HQ	Inspiron 15	Dell	Computer
IP6P128AU	Iphone 6 plus	Apple	Phone

Location	Street	City	Province or State	Country
JB Hi-Fi Garden City	Logan Rd	Brisbane	Queensland	AUS
Best Buy Streeterville	Michigan Ave	Chicago	Illinois	USA

Q1 Star Schema

- (a) Please draw a design of the data warehouse on sales data using the Star Schema.

Time	Day	Month	Quarter	Year
12/2/17	12	2	Q1	2017
15/5/17	15	5	Q2	2017

Item	Item Name	Brand	Type
INS157700HQ	Inspiron 15	Dell	Computer
IP6P128AU	Iphone 6 plus	Apple	Phone

Fact table

Time	Item	Location	Sales
12/2/17	INS157700HQ	JB Hi-Fi Garden City	25
15/5/17	IP6P128AU	Best Buy Streeterville	10

Location	Street	City	Province or State	Country
JB Hi-Fi Garden City	Logan Rd	Brisbane	Queensland	AUS
Best Buy Streeterville	Michigan Ave	Chicago	Illinois	USA

- Which table is larger?
  - Fact table

+ Q1 Snowflake Schema

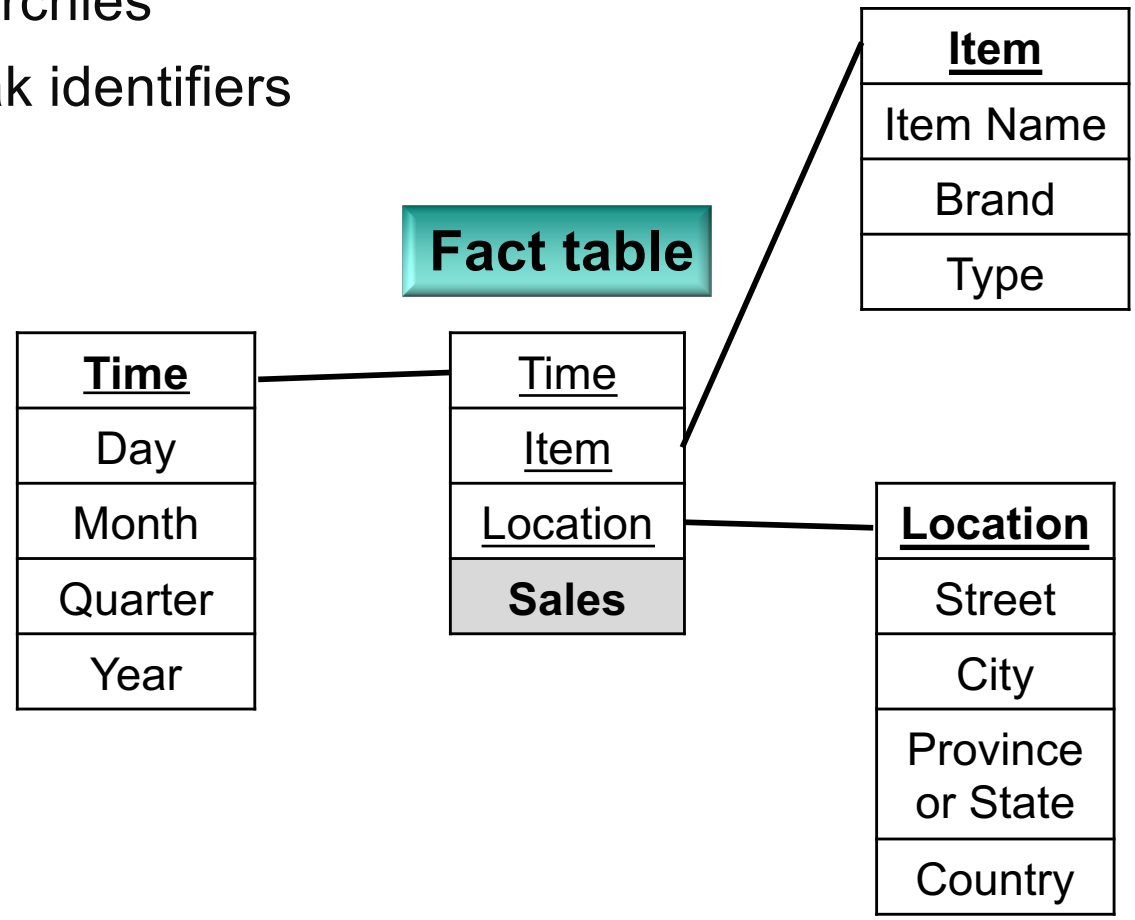
- (b) Construct a Snowflake Schema for the above data warehouse, with the dimension tables normalized to 3NF.

Normal Forms
https://en.wikipedia.org/wiki/Database_normalization#Normal_forms

	UNF (1970)	1NF (1971)	2NF (1971)	3NF (1971)
Primary key	✓	✓	✓	✓
No repeating groups	✓	✓	✓	✓
Atomic columns	✗	✓	✓	✓
No partial dependencies	✗	✗	✓	✓
No transitive dependencies	✗	✗	✗	✓

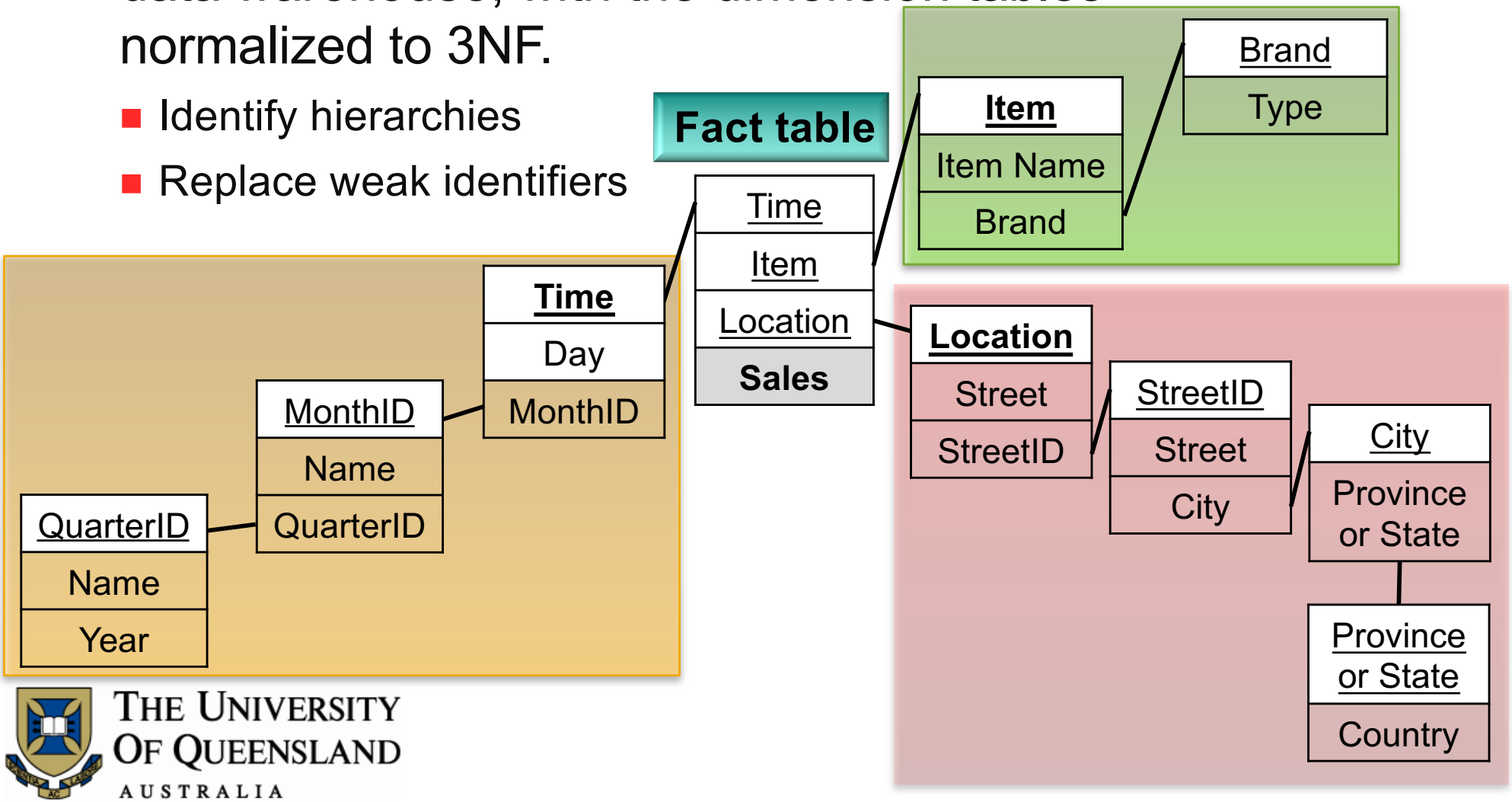
+ Q1 Snowflake Schema

- (b) Construct a Snowflake Schema for the above data warehouse, with the dimension tables normalized to 3NF.
  - Identify hierarchies
  - Replace weak identifiers



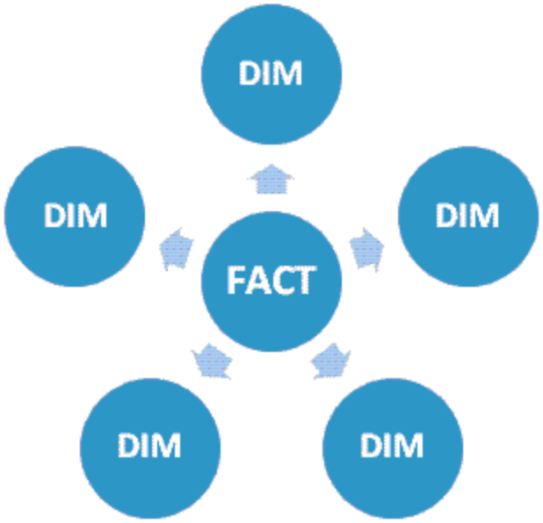
+ Q1 Snowflake Schema

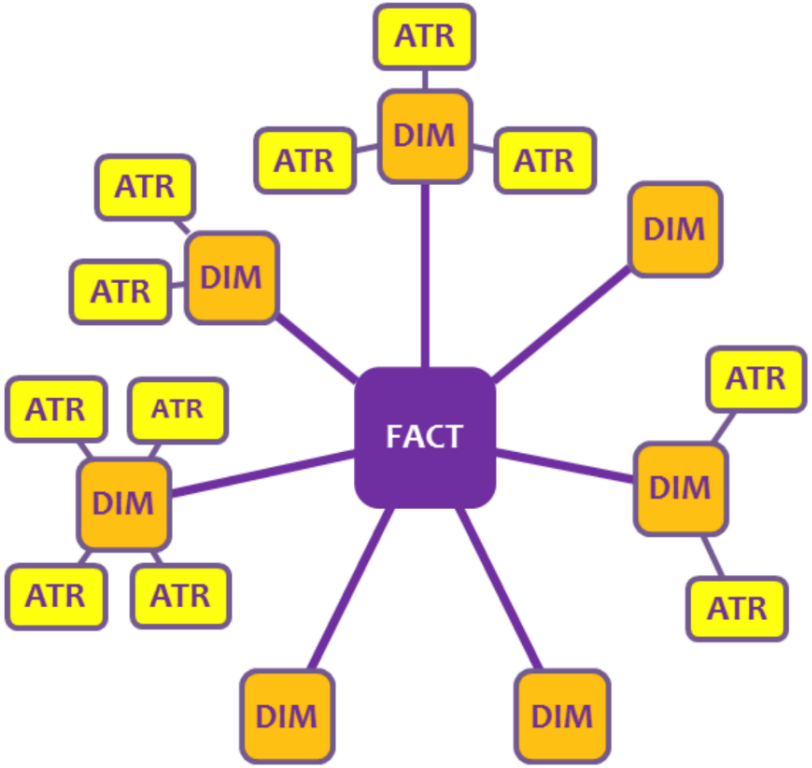
- (b) Construct a Snowflake Schema for the above data warehouse, with the dimension tables normalized to 3NF.
  - Identify hierarchies
  - Replace weak identifiers



+ Q1 Star vs Snowflake Schema

- (c) Compare and contrast the above two schemas, Star and Snowflake
- STAR schema
  - Faster query processing speed (less join)
- SNOWFLAKE schema
  - Less space consumption
  - Less data integrity problems





+ Question 2

Typical Functionality of DW

- The following table is a sample of the *AllElectronics* sales data for one year. This report identifies a multi-dimensional model using dimension hierarchies defined in Question 1.

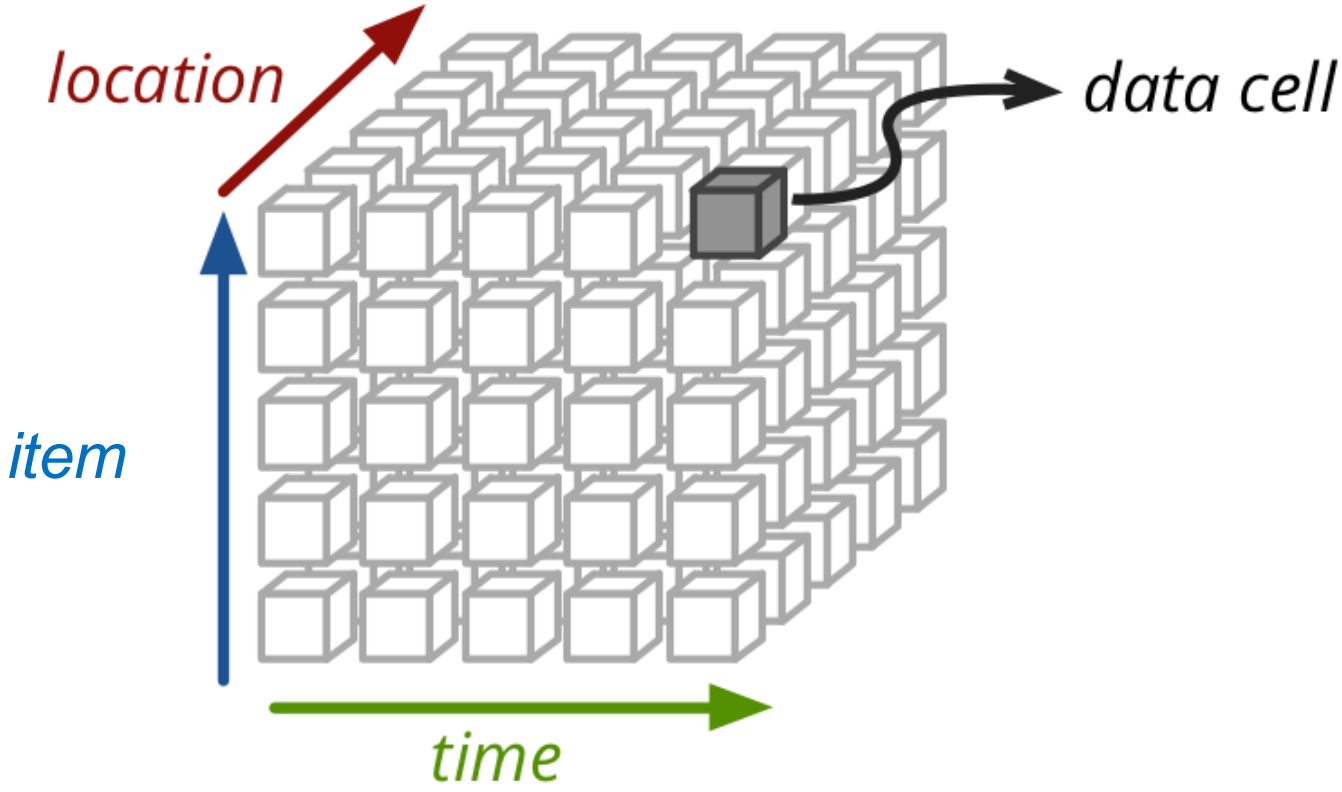
	location = "Chicago"				***location*** = "New York"				***location*** = "Toronto"				***location*** = "Vancouver"			
	item				***item***				***item***				***item***			
	home				*home*				*home*				*home*			
time	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

+ Q2 Multidimensional vs Relational

- How is the data stored?
  - Multidimensional: Data Cube
  - Relational: Relational table
    - STAR schema
    - SNOWFLAKE schema

+ Q2 Data Cube

- Three dimensions: location, item and time



+ Q2 Cube Operations

- (a) Perform a Roll-up operation on the *location* dimension from **cities** to **countries**.
  - Location -> Street -> City -> Province or State -> Country

location ="USA" *location ="CAN"*

	location = "Chicago"				***location*** = "New York"				***location*** = "Toronto"				***location*** = "Vancouver"			
	item				***item***				***item***				***item***			
time	*home ent.*	*comp.*	*phone*	*sec.*	*home ent.*	*comp.*	*phone*	*sec.*	*home ent.*	*comp.*	*phone*	*sec.*	*home ent.*	*comp.*	*phone*	*sec.*
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

+ Q2 Cube Operations

- (a) Perform a Roll-up operation on the *location* dimension from **cities** to **countries**.
  - Location -> Street -> City -> Province or State -> Country

location ="USA"

	location = "Chicago"				location = "New York"			
	item				item			
time	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872
Q2	943	890	64	698	1130	1024	41	925
Q3	1032	924	59	789	1034	1048	45	1002
Q4	1129	992	63	870	1142	1091	54	984

	location = "USA"			
	item			
time	home ent.	comp.	phone	sec.
Q1	1941	1850	127	1495
Q2				
Q3				
Q4				

+ Q2 Cube Operations

- (b) Perform a Drill-down operation on the *time* dimension from **quarters** to **months** (You may assume same sales at each month).
  - Time -> Day -> Month -> Quarter -> Year

	location = "Chicago"				location = "New York"				location = "Toronto"				location = "Vancouver"			
	item				item				item				item			
	home				home				home				home			
time	ent.	comp.	phone	sec.	ent.	comp.	phone	sec.	ent.	comp.	phone	sec.	ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

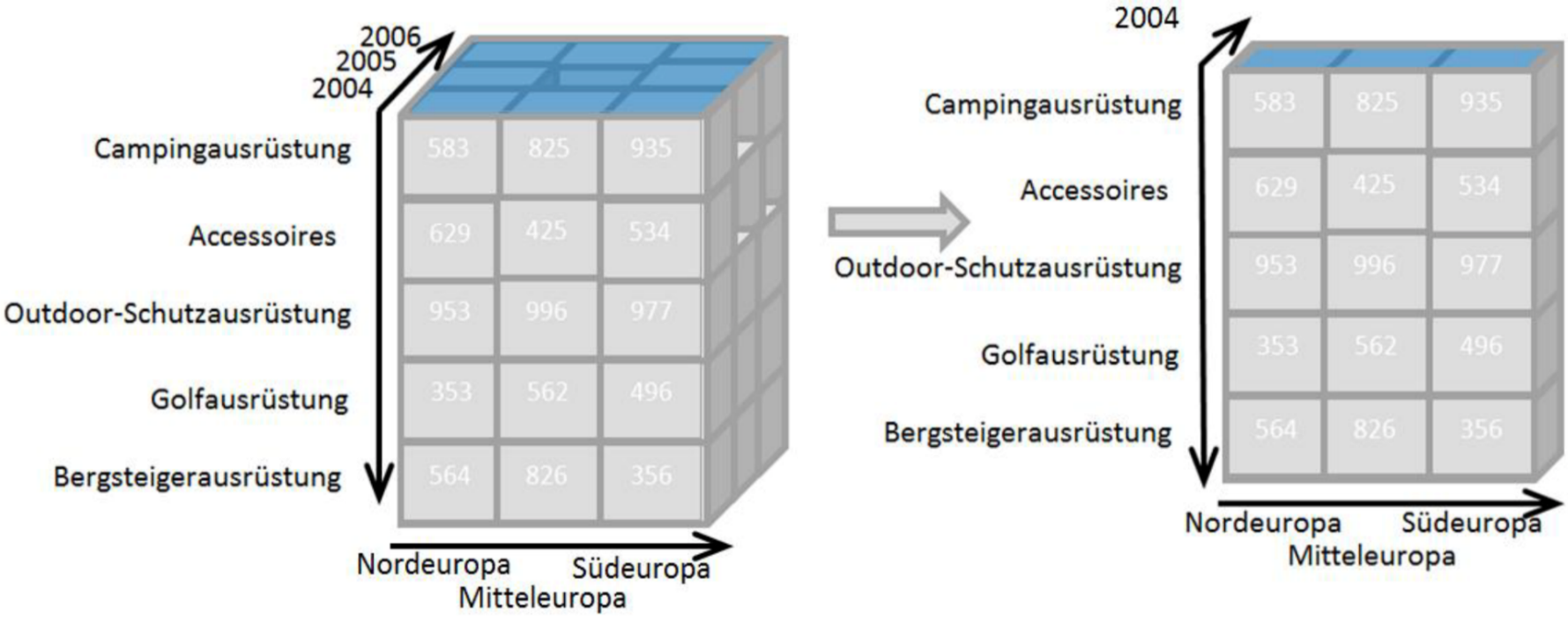
Jan
Feb
March

	location = "Chicago"				location = "New York"				location = "Toronto"				location = "Vancouver"			
	item				item				item				item			
	home				home				home				home			
time	ent.	comp.	phone	sec.	ent.	comp.	phone	sec.	ent.	comp.	phone	sec.	ent.	comp.	phone	sec.
Jan	284	294	29	207												
Feb	285	294	30	208												
Mar	285	294	30	208												

Q2 Cube Operations

- Slice and dice
  - Perform projection operations on the dimensions

- (c) Perform a Slice operation for *time* = “Q1”.



+ Q2 Cube Operations

- (c) Perform a Slice operation for *time* = "Q1".

	location = "Chicago"				***location*** = "New York"				***location*** = "Toronto"				***location*** = "Vancouver"			
	item				***item***				***item***				***item***			
	home				*home*				*home*				*home*			
time	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

	Home Ent.	**Comp.**	**Phone**	**Sec.**
Chicago	854	882	89	623
New York	1087	968	38	872
Toronto	818	746	43	591
Vancouver	825	825	14	400

+ Q2 Cube Operations

- (d) Perform a Dice operation for (*location* = "Toronto" or "Vancouver") and (*time* ="Q1" or "Q2") and (*item* = "home entertainment" or "computer").

	location = "Chicago"				***location*** = "New York"				***location*** = "Toronto"				***location*** = "Vancouver"			
	item				***item***				***item***				***item***			
	home				*home*				*home*				*home*			
time	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

	location = "Toronto"		***location*** = "Vancouver"	
	item		***item***	
	home		*home*	
time	*ent.*	*comp.*	*ent.*	*comp.*
Q1	818	746	605	825
Q2	894	769	680	952

+ Q2 Cube Operations

- (e) Perform a Pivot operation on *location* and *item* dimensions.

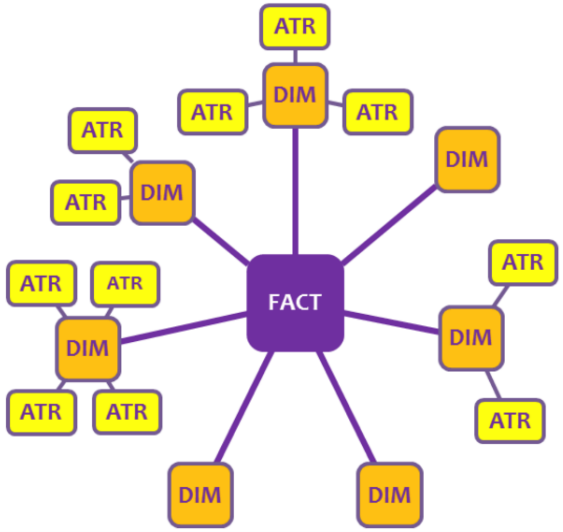
	location = "Chicago"				***location*** = "New York"				***location*** = "Toronto"				***location*** = "Vancouver"			
	item				***item***				***item***				***item***			
	home				*home*				*home*				*home*			
time	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

	Home Ent.	Comp.	Phone	Sec.	Total Item
Chicago	3958				
New York					
Toronto					
Vancouver					
Total Location					

+ Question 3

- OLAP queries can be implemented using standard SQL queries. Consider the sample *AllElectronics* data in Question 2 and answer the questions.
- (a) Write SQL queries to implement
  - The Roll-up operation in Question 2(a)
  - The Dice operation in Question 2(d)
  - The Pivot operation in Question 2(e).

+ Q3 OLAP Queries



- The Roll-up operation in Question 2(a)
  - Location -> Street -> City
    -> Province or State -> Country

	Home Ent.				Comp.				…
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	…
Chicago	854	943	1032	1129	882	890	924	992	…
New York									
Total USA									
Toronto									
Vancouver									
Total CAN									…

SELECT a.item, a.time, b.country, sum(a.sales)
FROM AllElectronics a, location b
WHERE a.location = b.city
GROUP BY a.item, a.time, b.country

+ Q3 OLAP Queries

- The Dice operation in Question 2(d)

	location = "Chicago"				***location*** = "New York"				***location*** = "Toronto"				***location*** = "Vancouver"			
	item				***item***				***item***				***item***			
	home				*home*				*home*				*home*			
time	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

	location = "Toronto"		***location*** = "Vancouver"	
	item		***item***	
	home		*home*	
time	*ent.*	*comp.*	*ent.*	*comp.*
Q1	818	746	605	825
Q2	894	769	680	952

SELECT item, time, location, sales
FROM AllElectronics
WHERE location ***IN*** ('Toronto', 'Vancouver')
    AND time ***IN*** ('Q1', 'Q2')
    AND item ***IN*** ('home ent.', 'comp.');

+ Q3 OLAP Queries

- The Pivot operation in Question 2(e).

	Home Ent.	Comp.	Phone	Sec.	Total Location
Chicago	3958				
New York					
Toronto					
Vancouver					
Total Item					

1.

SELECT SUM(sales)
FROM AllElectronics
GROUP BY location, item;

2.

SELECT SUM(sales)
FROM AllElectronics
GROUP BY location;

3.

SELECT SUM(sales)
FROM AllElectronics
GROUP BY item;

4.

SELECT SUM(sales)
FROM AllElectronics;

+ Q3 CUBE Operation

- (b) Oracle supports CUBE extension to standard SQL for building a data cube. For the following SQL query, it *equivalents* to how many SQL queries with GROUP BY clauses? Why?

```
SELECT item, time, location, sum(sales)
FROM AllElectronics
GROUP BY CUBE(item, time, location);
```

+ Q3 CUBE Operation

SELECT item, time, location, sum(sales)
FROM AllElectronics
GROUP BY CUBE(item, time, location);

- 2^3=8 GROUP BY queries, namely:
  - (item, time, location)
  - (item, time)
  - (time, location)
  - (item, location)
  - (item)
  - (time)
  - (location)
  - (NULL)

+ Q3 CUBE Operation

SELECT item, time, location, sum(sales)
FROM AllElectronics
GROUP BY CUBE(item, time, location);

- After creating the cube, it supports all types of aggregation queries. Therefore, any combination of the three dimensions should be **pre-processed**.

+ Q4 Difference between Operational DB and DW

- Operational DB run business

  - update data in ***real-time*** (insert/update/delete/select)

  - transactions that guarantee ACID properties

  - optimized for faster transaction/processing

  - effectiveness measured by # of transactions per sec./min.

  - detailed and current data

  - schema usually 3NF

+ Q4 Difference between Operational DB and DW

- DW planning, decision support, etc.

  - complex queries with aggregations

  - effectiveness measured by response time

  - current and historical data

  - schema in multi-dimensions

  - materialized view

Order table

Partition key	Row key	Order date	Shipping address	Total invoice	Order status
001 (Customer ID)	1 (Order ID)	11082013	One Microsoft way Redmond, WA 98052	$400	In process
005	2	11082013	One Microsoft way Redmond, WA 98052	$200	Shipped

OrderItem table

Partition key	Row key	Product	Unit Price	Amount	Total
1 (Order ID)	001_1 (OrderItem ID)	XX	$100	2	$200
1	001_2	YY	$40	5	$200
2	002_1	ZZ	$200	1	$200

Customer table

Partition key	Row key	Billing Information	Shipping address	Gender	Age
US East (region)	001 (Customer ID)	*****0001	One Microsoft way Redmond, WA 98052	Female	30
US East	002	*****2006	One Microsoft way Redmond, WA 98052	Male	40

Materialized View

Partition key	Row key	Product Name	Total sold	Number of customers
Electronics (Product category)	001 (Product ID)	XX	$30,000	500
Electronics	002	YY	$100,000	400

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

+ Q4 Difference between Operational DB and DW

further reading:

https://en.wikipedia.org/wiki/Operational_database
https://www.quora.com/What-is-the-difference-between-OLTP-and-OLAP
https://www.guru99.com/oltp-vs-olap.html
https://stackoverflow.com/questions/21900185/

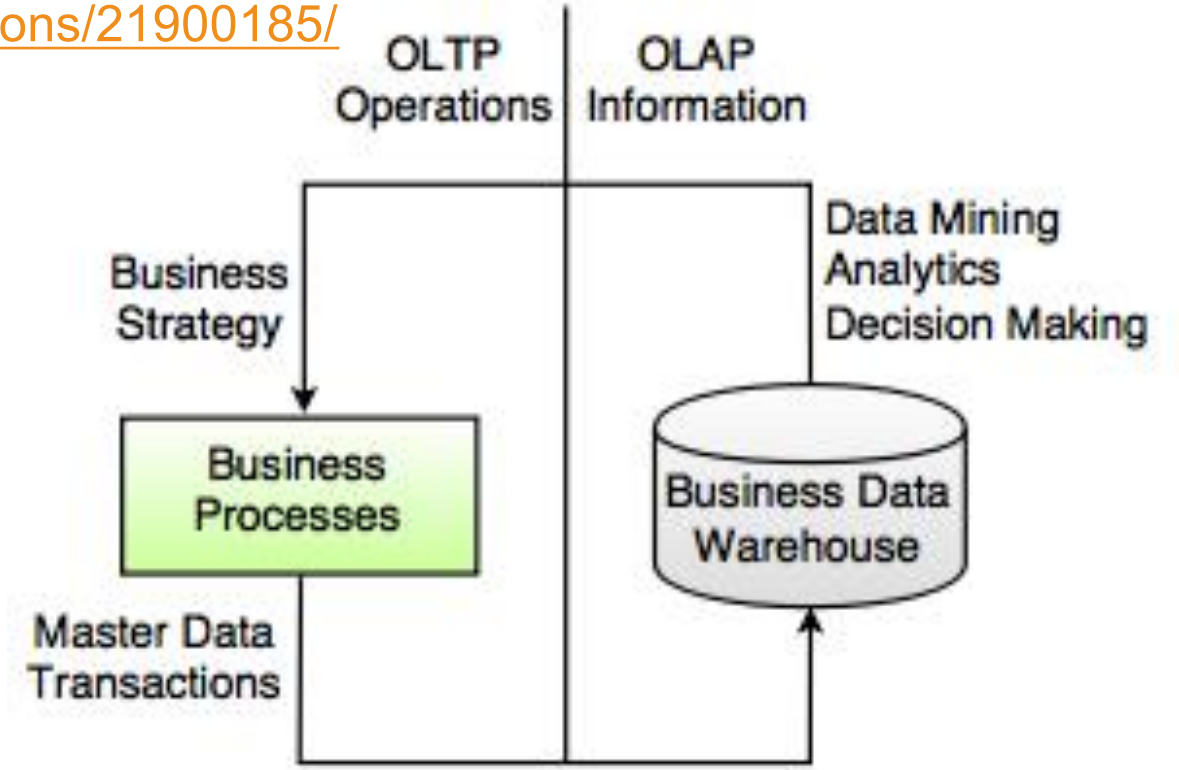


Fig. OLTP and OLAP

+ Q4 Difference between Operational DB and DW

http://datawarehouse4u.info/OLTP-vs-OLAP.html

	OLTP System Online Transaction Processing (Operational System)	OLAP System Online Analytical Processing (Data Warehouse)
Source of data	Operational data; OLTPs are the original source of the data.	Consolidation data; OLAP data comes from the various OLTP Databases
Purpose of data	To control and run fundamental business tasks	To help with planning, problem solving, and decision support
What the data	Reveals a snapshot of ongoing business processes	Multi-dimensional views of various kinds of business activities
Inserts and Updates	Short and fast inserts and updates initiated by end users	Periodic long-running batch jobs refresh the data
Queries	Relatively standardized and simple queries Returning relatively few records	Often complex queries involving aggregations
Processing Speed	Typically very fast	Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by Maintaining materialized view
Space Requirements	Can be relatively small if historical data is archived	Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP
Database Design	Highly normalized with many tables	Typically de-normalized with fewer tables; use of star and/or snowflake schemas
Backup and Recovery	Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability	Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method

+ Q4 Heterogeneous

- **Multiple sources → formatted → cleaned → fitted & loaded**
  - Syntactic data integration
    - Must access data from a variety of source formats and repositories
  - Semantic data integration
    - When getting data from multiple sources, must eliminate mismatches, e.g., different currencies
  - Load, refresh and purge
    - Must load data, periodically refresh it, and purge too-old data
  - Metadata management
    - Must keep track of source, loading time, and other information for all data in the warehouse

+ Q4 Heterogeneous

- **Quality**

consistency, duplicates, logic conflicts, missing data

- **Performance and cost**

fit for the needs of the organization

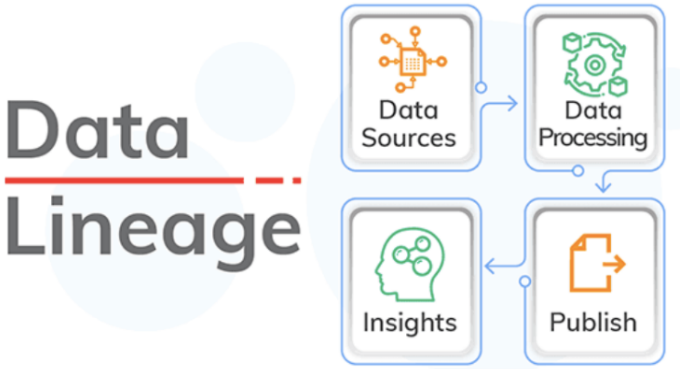
effectiveness and efficiency

- **User Acceptance**

provide comprehensive info to non-SQL experts

+ Q4 Heterogeneous

- **Meta data – Lineage**
  - The origin and the transformation that data goes through over time.



- **What's timeliness (latency) of the data (instantaneous, daily, weekly, monthly ..)?**
  - Monitoring, capturing and interpreting real-time data to ensure the best optimisation of decision marking.

- **How to detect and handle data inconsistency?**
  - Semantic e.g. Major(Faculty, program) -> (ITEE, Marketing)
  - Representational e.g. Queesland vs. QLD

More challenges and questions of data processing…

Missing data, noisy data, data linkage (data stemming/stopwords removal), more data mining techniques…