Management Science
Business Intelligence

Dan Brandon, Ph.D., PMP

Business Intelligence

- As companies become larger, more complex, and more diverse (multinational), it becomes harder and harder for them to understand who their customers are, how to best serve them, and how to maximize their profits
- To make such decisions in todays' fast paced global marketplace, companies make extensive use of something called "business intelligence"
- **Data on every single aspect of a business is meticulously collected and then rigorously analyzed to make sure each action is optimal**

Business Intelligence (con't)

- This approach relies on large "data warehouses" and complex software that uses sophisticated algorithms to pore through endless amounts of data
- Business technologists have many names for this revolutionary technology; "business intelligence" (BI), "data analytics," and "data mining" are among the most common

Business Intelligence (con't)

- *The Economist* says it's "a golden vein", and business experts now call it "the new science of winning"
- FedEx, Capital One, and Amazon.com can't function without it
- It's been adopted by nearly every Fortune 500 company
- Even professional sports franchises like the Boston Red Sox, Oakland A's, and New England Patriots are being forced to use this technology
- A Gartner survey of 1,400 chief information officers suggests that business intelligence is the number one technology priority for IT organizations"

It started with Capital One...

- Back in the 1980s, consultants Richard Fairbank and Nigel Morris realized that by analyzing data, credit card companies -- like a tiny Virginia bank called Signet -- could systematically target the most lucrative customers, while leaving their competition to fight over the rest
- Their approach was so successful, Signet ended up spinning off its credit card division as a separate company, which became Capital One
- Today, Capital One runs approximately 300 data tests *per day,* and it credits data analysis with increasing the retention rate of its savings business by a whopping 87% while simultaneously slashing the cost of acquiring new customers by 83%
- BI has allowed Capital One to increase the value of its stock 1,000% over its first ten years as a public company -- outpacing the S&P 500 by a power of 10

Business Intelligence (con’t)

- Companies are not short on data
- The average large business stores more than 200 terabytes (10^{12}) from their daily transactions
- This tells them who is buying what, and also where and when
- But today business also need to know <u>why, or why not</u>
- How have companies traditionally done that ?

Business Intelligence (con't)

- Traditionally this was done with classical business research such as surveys, focus groups, etc.
- But today it also comes from tweets, videos, likes, clickstream data, and other social media sources
  - This is called "**Big Data**"
  - The average company in 15 of 17 US sectors now has more data stored than the Library of Congress

Know Thy Customer

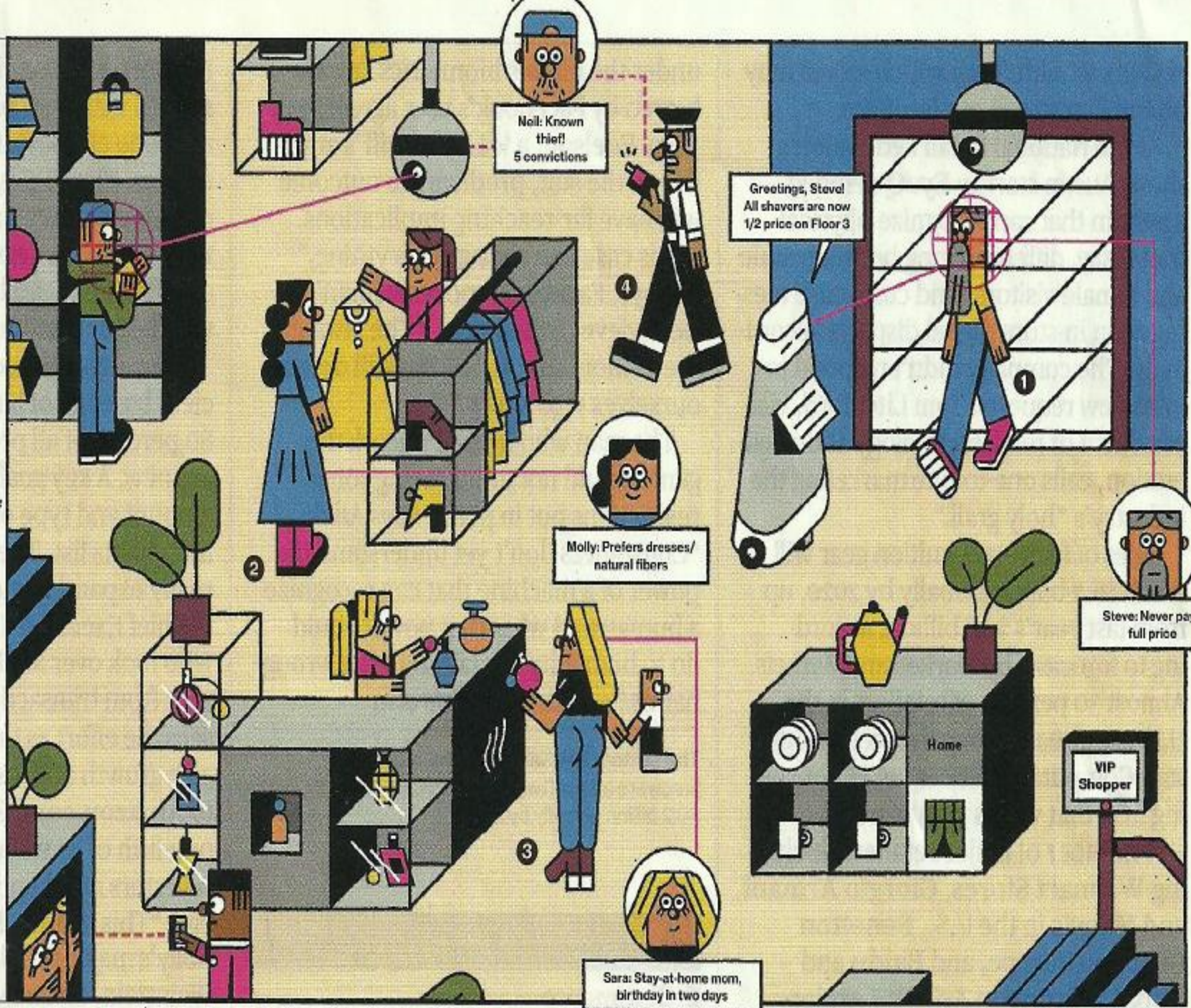
Biometrics and other surveillance devices will allow retailers and shoppers to interact in new ways.

❶ Facial recognition software triggers a personalized message to Steve, drawing on information from his loyalty program profile.

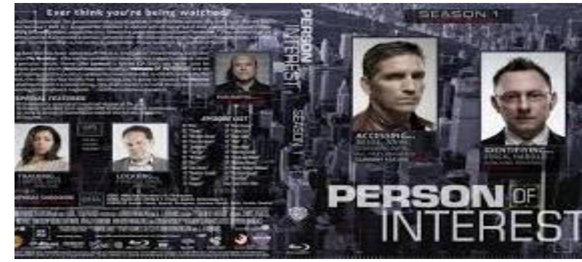
❷ A sales associate picks out a dress for Molly based on her purchase history.

❸ The attendant at the fragrance counter is notified that Sara's birthday is approaching, so she's eligible for a gift with purchase.

❹ Neil is flagged as an "unwanted person," and security is alerted.



Big Data Applications



- In addition to business applications, big data storage and processing requirements are also showing up in:

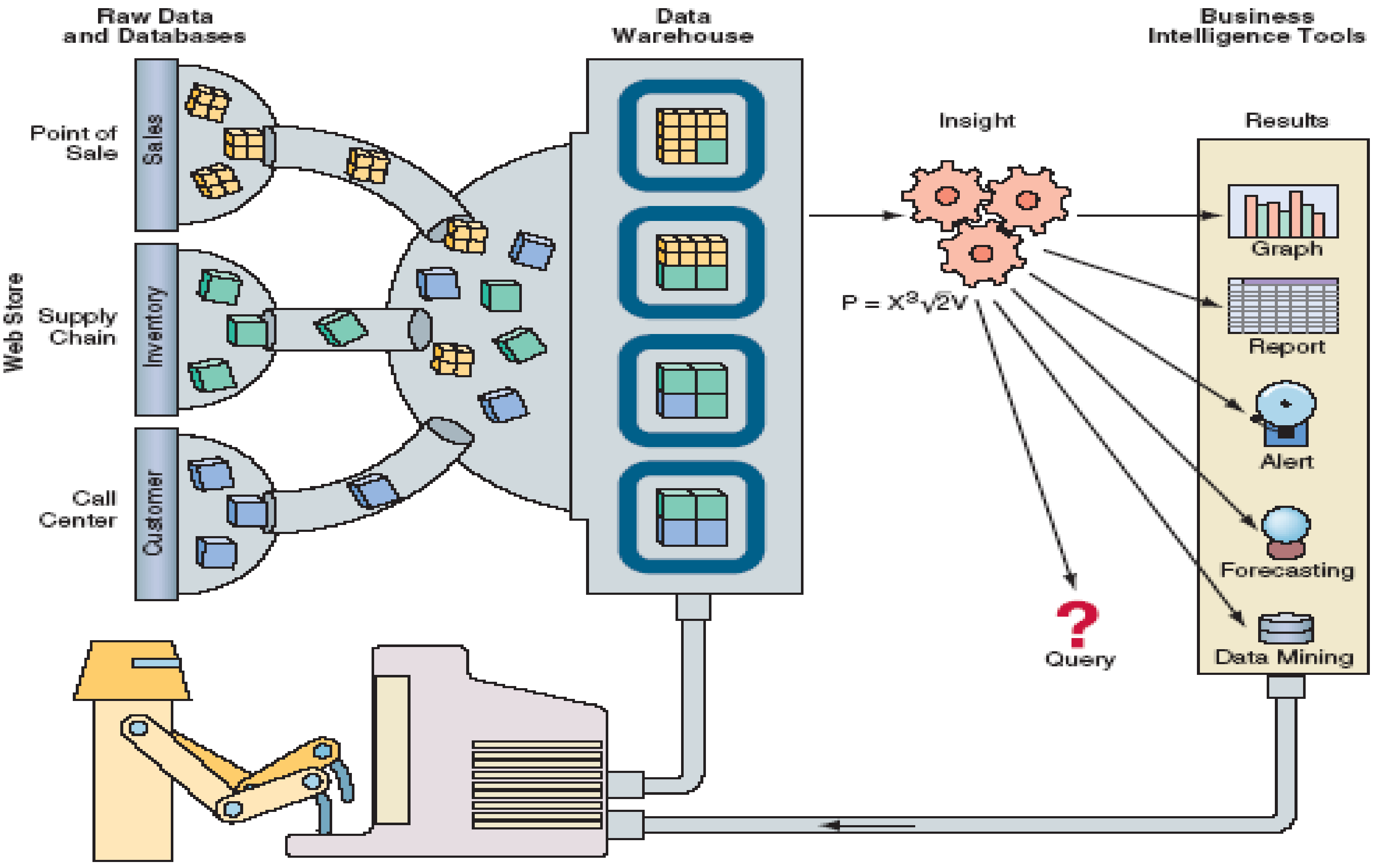
  - Search analysis
  - Genomes
  - Phone records (such as the metadata analysis by our NSA)
  - Video surveillance processing

  - Geological surveys
  - Climate data
  - Bioinformatics, disease (i.e. cancer) simulation

Wikipedia

- **Business intelligence** (**BI**) is a set of theories, methodologies, architectures, and technologies that transform raw data into meaningful and useful information for business purposes
- Making use of new opportunities and implementing an effective strategy can provide a competitive market advantage and long-term stability.
- Common functions of business intelligence technologies are reporting, online analytical processing, analytics, data mining, process mining, complex event processing, business performance management, benchmarking, text mining, predictive analytics and prescriptive analytics

Business Intelligence



Business Intelligence Skills

- Database – SQL
- Data Warehouse
- Statistics
- Analytics (quantitative methods)
- OLAP
- Data Mining
- Data Visualization
- Artificial Intelligence

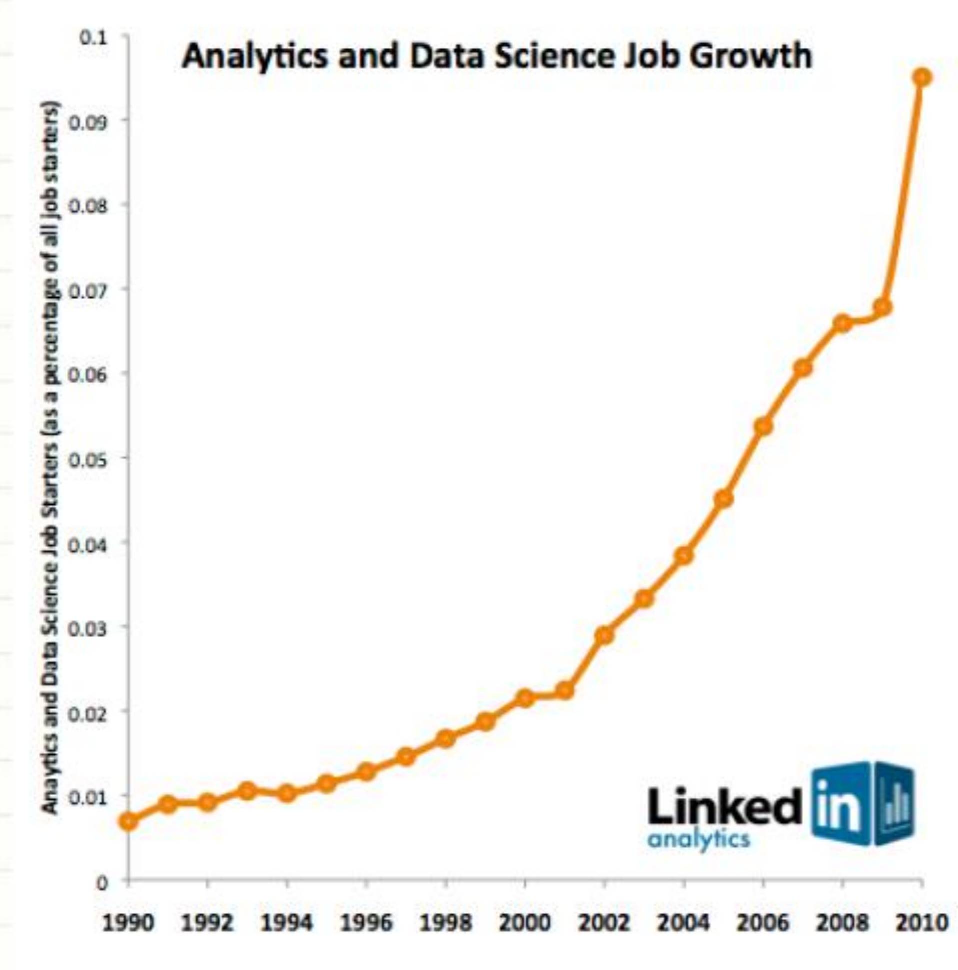
Top Tech Initiatives for 2015
[CIO Magazine Survey]

- Business Intelligence (analytics)
- Mobile Technologies
- Cloud Services
- Application Modernization
- Customer Experience Technologies
- Security and Risk Management

Data Analytics Jobs

A report released in 2016 by Glassdoor says that data scientists have the best jobs in the U.S., according to that company's analysis.

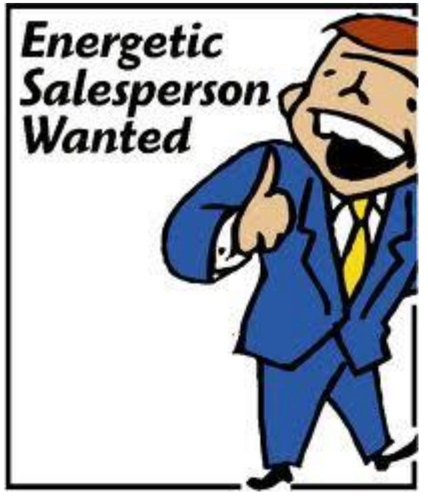
With a median base salary of $116,840, more than 1,700 job openings on Glassdoor's site, and a user-provided career opportunities rating of 4.1, "data scientist" took the prize for most highly rated job title in America.



SQL

Example Database Tables
[salesperson, product, sales]

- S (SID, SName, City)
- P (PID, PName, Size, Price)
- SP (SID, PID, Qty)



Keys ?

Salesperson Table (S)

SID	Sname	City
S1	Peterson	Aarhus
S2	Olsen	Copenhagen
S4	Hansen	Odense
S5	Jensen	Copenhagen

Product Table (P)

PID	PName	Size	Price
P1	Shirt	6	50
P3	Trousers	5	90
P4	Socks	7	20
P5	Blouse	6	50
P8	Blouse	8	60

SP Table (Intersection Table)

SID	PID	Qty
S2	P1	200
S2	P3	100
S4	P5	200
S4	P8	100
S5	P1	50
S5	P3	500
S5	P4	800
S5	P5	500
S5	P8	100

Access Exercise

- 1. Design each of three tables (S, P, SP), and <u>set primary keys</u>
- 2. Set up relationships (foreign keys)
- 3. Enter data

Don’t look ahead !

Access Model

S

SID	SName	City
S1	Peterson	Aarhus
S2	Olsen	Copenhagen
S4	Hansen	Odense
S5	Jensen	Copenhagen

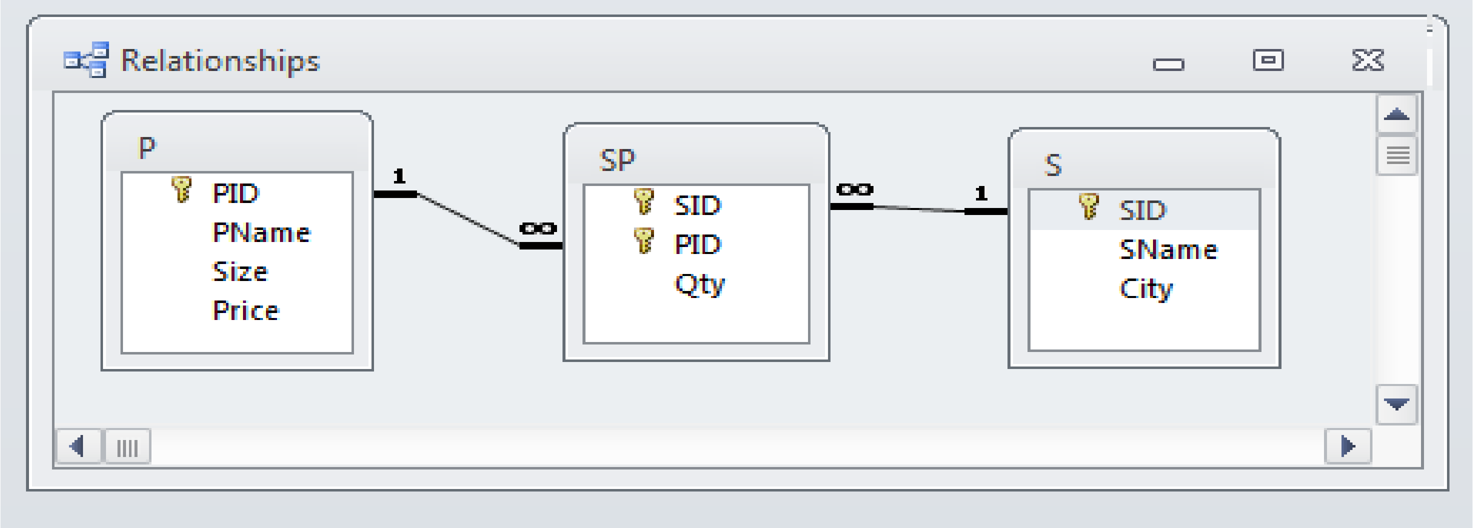
SP

SID	PID	Qty
S2	P1	200
S2	P3	100
S4	P5	200
S4	P8	100
S5	P1	50
S5	P3	500
S5	P4	800
S5	P5	500
S5	P8	100

P

PID	PName	Size	Price
P1	Shirt	6	$50.00
P3	Trousers	5	$90.00
P4	Socks	7	$20.00
P5	Blouse	6	$50.00
P8	Blouse	8	$60.00

Access Relationship View
[establish “referential integrity”]



Makes sure that the corresponding rows exists in S and P before adding entries to SP
Makes sure that entries in SP are deleted before deleting corresponding entries from S or P

Access "Query by Example"

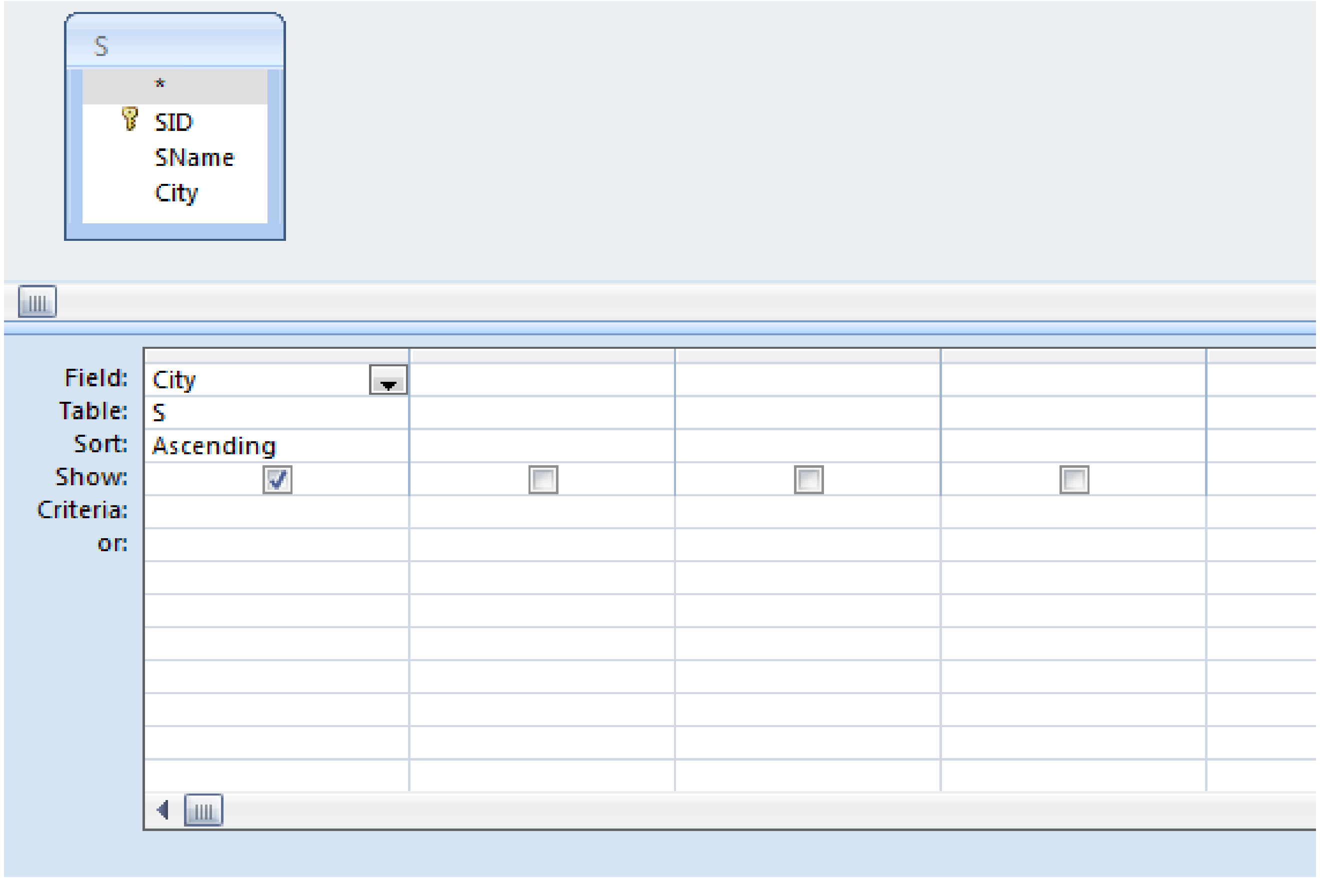
- Query-By-Example (QBE) is non-procedural
- There is no standard for QBE
- Not all queries can be done in QBE
- Create an Access QBE to answer this question:
  - "In which cities are salespersons located ?"

Don’t look ahead !

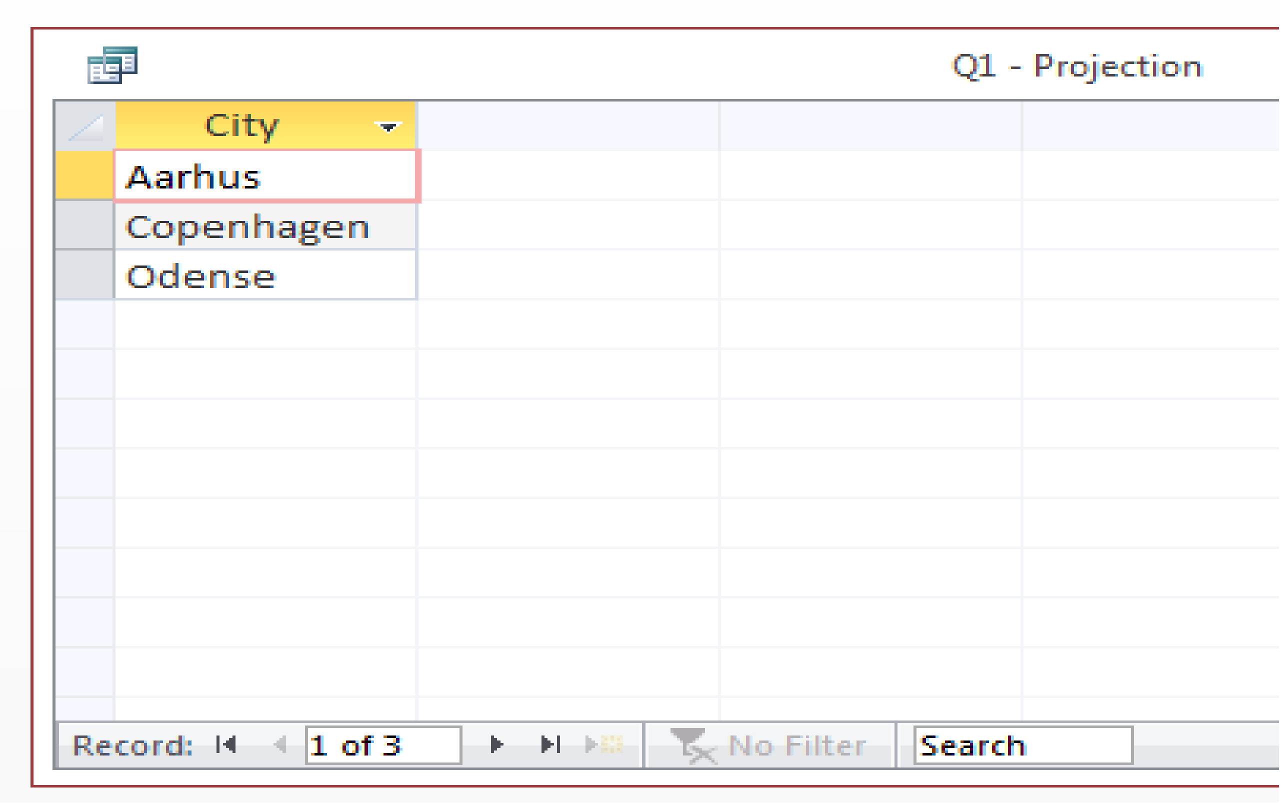
Query Results

City
Aarhus
Copenhagen
Odense

Access Query Grid



Access Datasheet View (Execute Query)



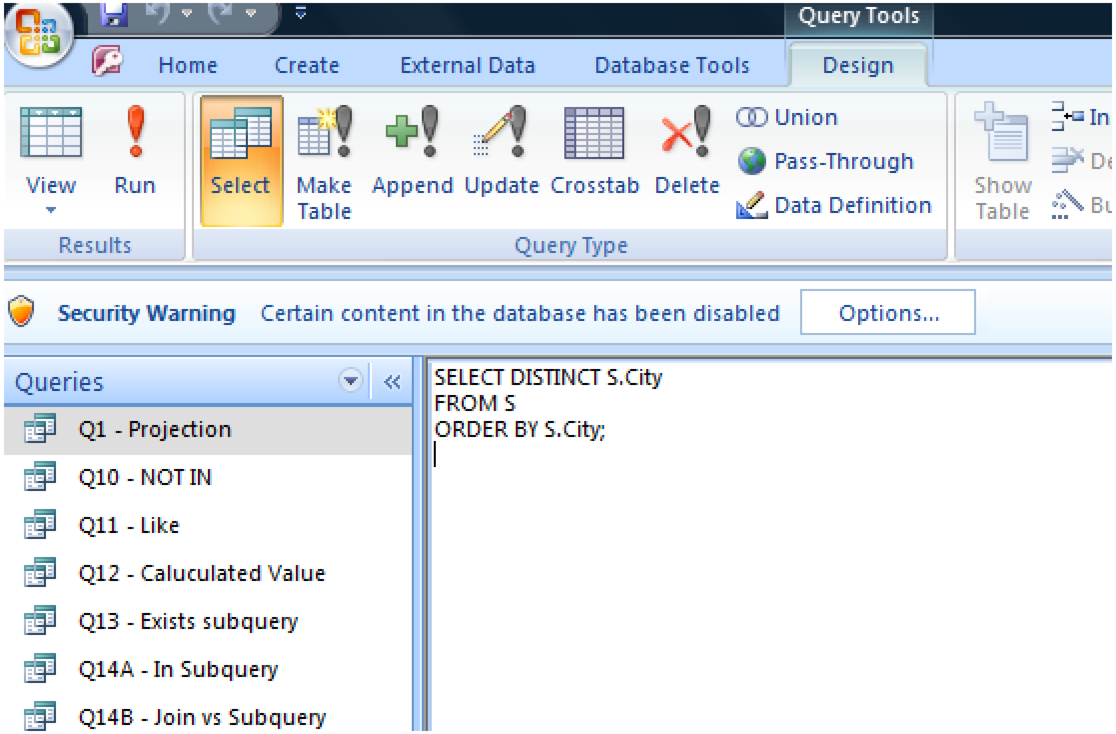
SQL
[structured query language]

- SQL is an international standard
- Many more queries can be done in SQL than in QBE
- Some queries can be done by one or more alternative methods in SQL
- SQL is more efficient than QBE
- SQL is used to interface to databases from programming languages

Retrieving Data via SQL

- SELECT columns-in-output-tables
  - FROM input-tables
  - WHERE logical-expression
  - ORDER BY columns -in-output-tables
- Output is always a table
  - may be a table with only one row and/or column
  - may be a NULL table

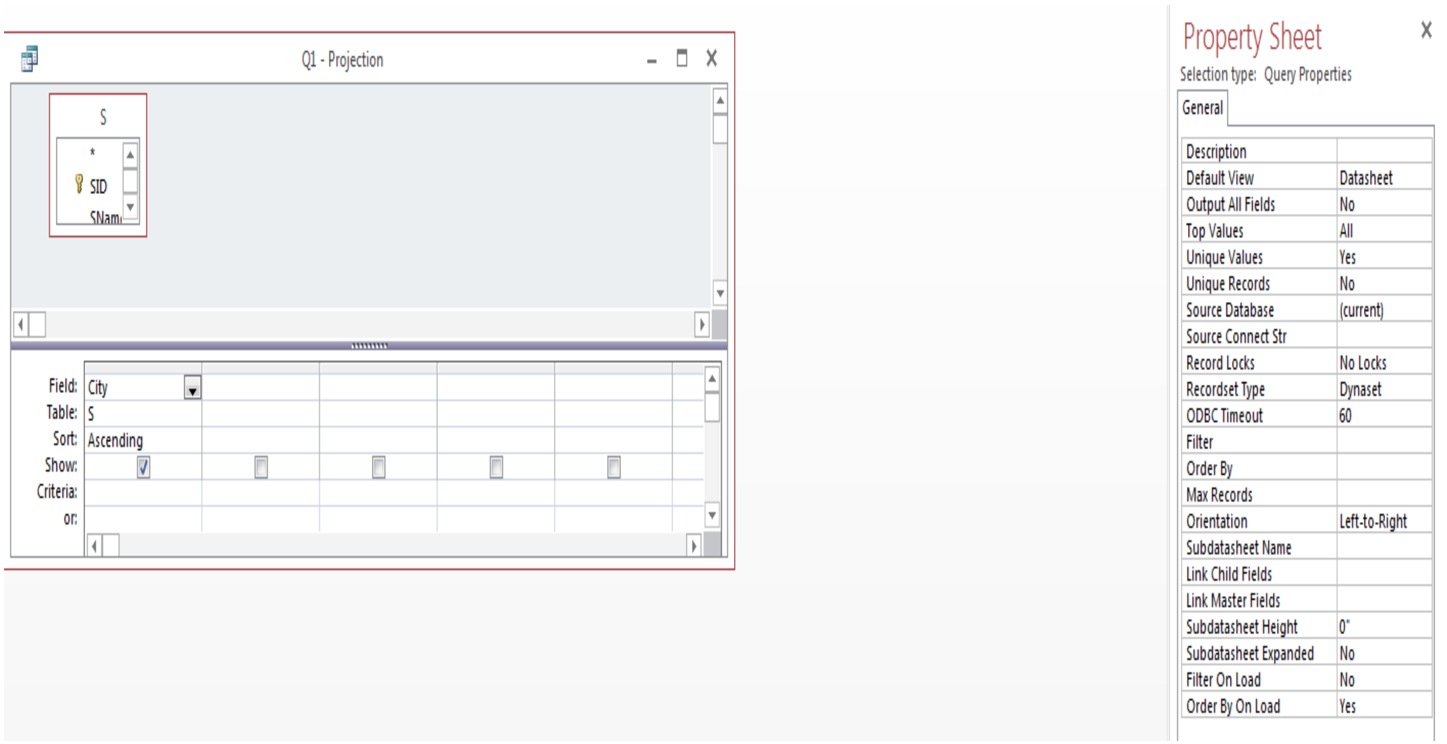
Access SQL View



Relational Algebra PROJECTION
("project" certain columns)

- SELECT DISTINCT City
  - FROM S
  - ORDER BY City
- *DISTINCT removes redundant columns**
- "In which cities are salespersons located ?"
- * In Access, select "Show Property Sheet" then; select "unique values" to "yes"

Query Properties



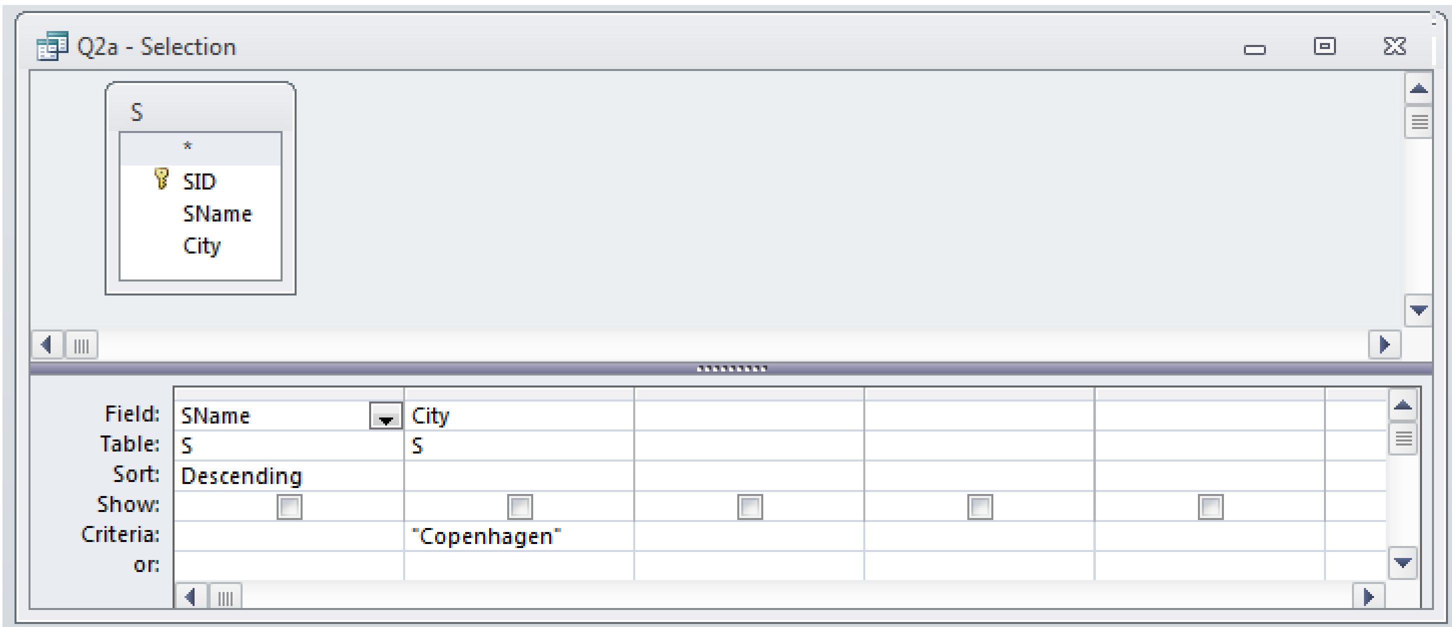
Distinct & Distinctrow

- In Access:
  - DISTINCT - Shows rows if selected columns are unique
  - DISTINCTROW - Shows rows if entire row from underlying table(s) are unique

Access Exercise

- Perform an Access query via the query grid (QBE) to answer this question:
  - "List info for salespersons in Copenhagen"
- What is the SQL for this query ?

Don’t look ahead !


Q2a - Selection
S
*
SID
SName
City
Field: SName | City
Table: S | S
Sort: Descending
Show:
Criteria: "Copenhagen"
or:

Query Results

SID	Sname	City
S2	Olsen	Copenhagen
S5	Jensen	Copenhagen

Relational Algebra SELECTION (WHERE) ("select" certain rows)

- SELECT *
  - FROM S
  - WHERE City = 'Copenhagen'
  - ORDER BY SName DESC
- "List info for salespersons in Copenhagen"
- * selects all columns
- DESC sorts in descending order

SQL "Join"

- Multiply the two tables together ("cross join" - finds all combinations):
  - Combine every row of the first table with every row of the second table
- Eliminate the rows that do not match the join criteria
  - Join criteria is usually a match between a foreign key in one table and the primary key in another table

SQL Join (inner) on Vendor ID

Checks (transaction)			
Check	Vendor ID	Date	Amount
1	B	4/11/2008	$ 451.58
2	D	4/14/2008	$ 4,483.99
3	B	4/15/2008	$ 848.48
4	A	4/18/2008	$ 8,564.99
5	E	4/19/2008	$ 1,941.80

Vendors (master)	
Vendor ID	Name
A	Adams Corp.
B	Blette, Inc.
C	Carlson Co.
E	ERT Corp.
F	Franks, Inc.

Dynaset (combined)				
Check	Vendor ID	Name	Date	Amount
1	B	Blette, Inc.	4/11/2008	$ 451.58
3	B	Blette, Inc.	4/15/2008	$ 848.48
4	A	Adams Corp.	4/18/2008	$8,564.99
5	E	ERT Corp.	4/19/2008	$1,941.80

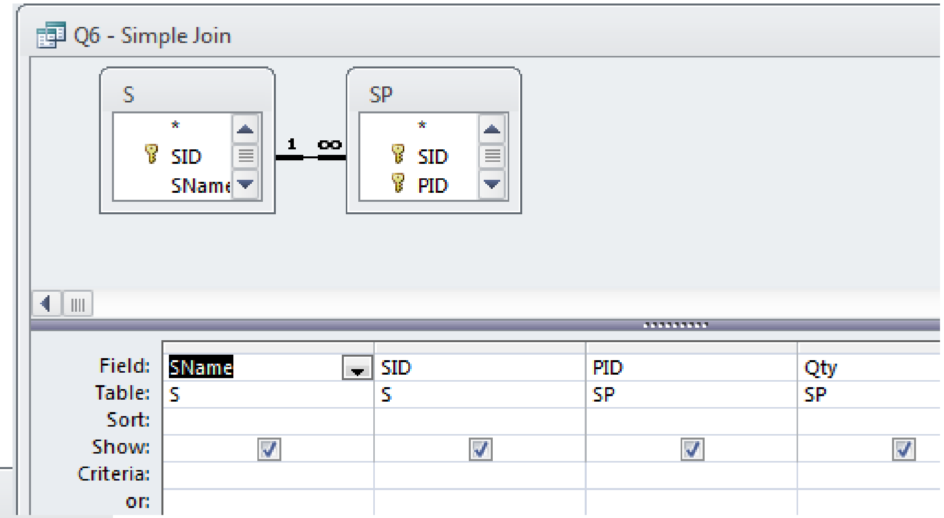
What about Check 2 ?

SQL Join Query

[show salespersons (name and ID) and how much they have sold]

Q6 - Simple Join

```
SELECT DISTINCTROW S.SName, S.SID, SP.PID, SP.Qty
FROM S INNER JOIN SP ON S.SID = SP.SID;
```



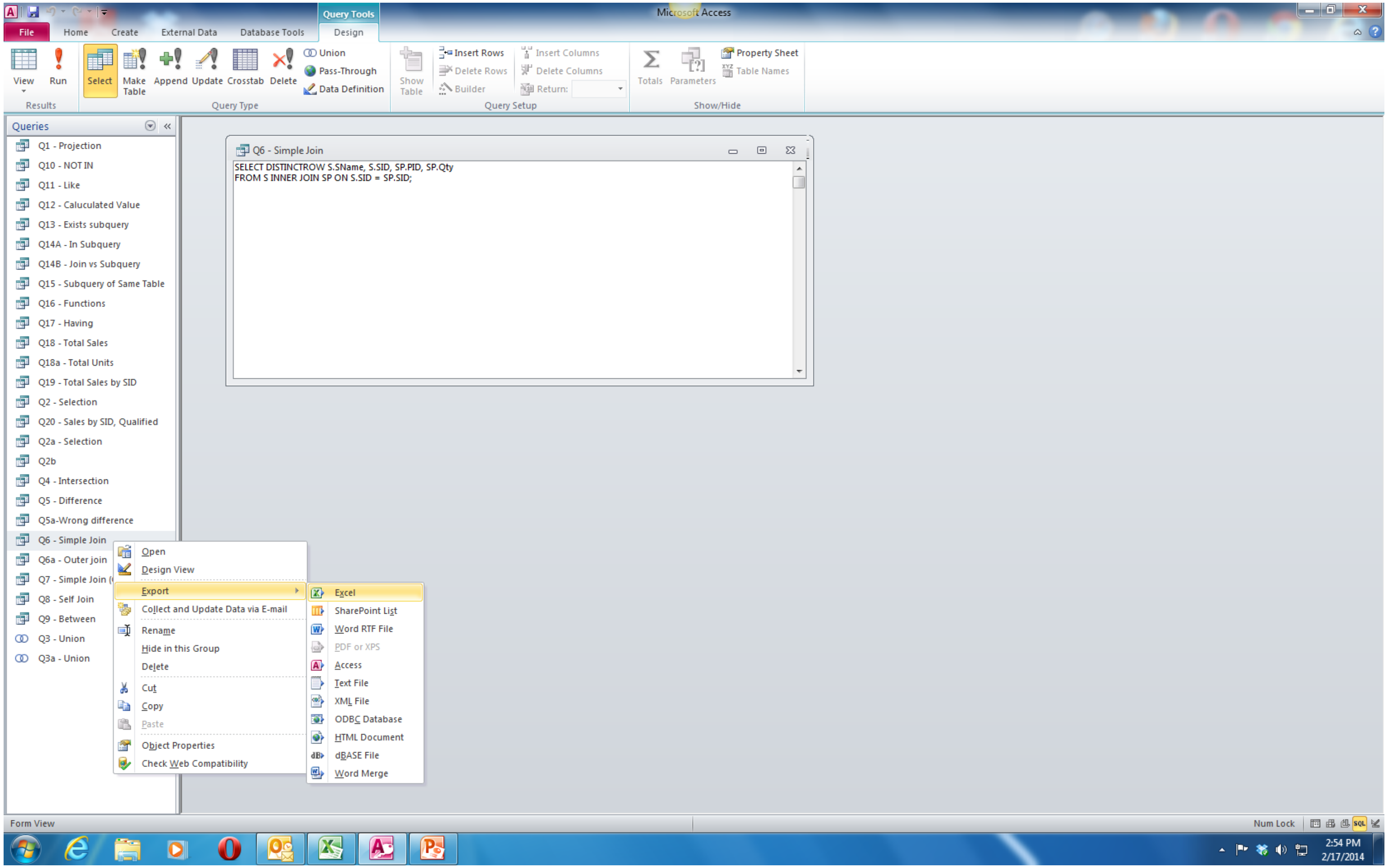
Q6 - Simple Join

SName	SID	PID	Qty
Olsen	S2	P1	200
Olsen	S2	P3	100
Hansen	S4	P5	200
Hansen	S4	P8	100
Jensen	S5	P1	50
Jensen	S5	P3	500
Jensen	S5	P4	800
Jensen	S5	P5	500
Jensen	S5	P8	100

Database → Spreadsheet

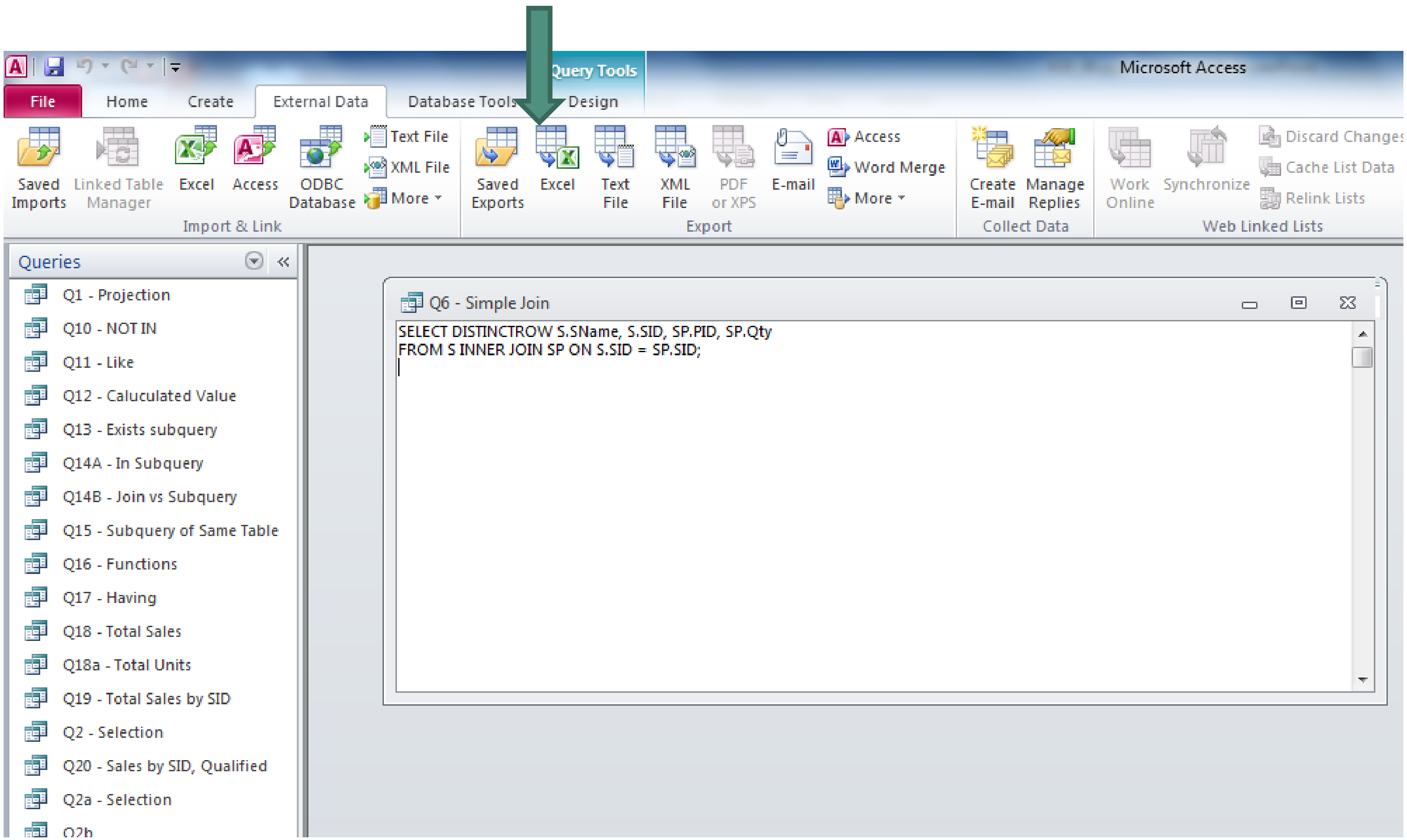
- Data is often exported from data warehouses or databases into Excel or another analysis tool
- Objects (such as queries) can be defined as the source of an export to Excel or other tool

Export Database Object to Excel [right click object]

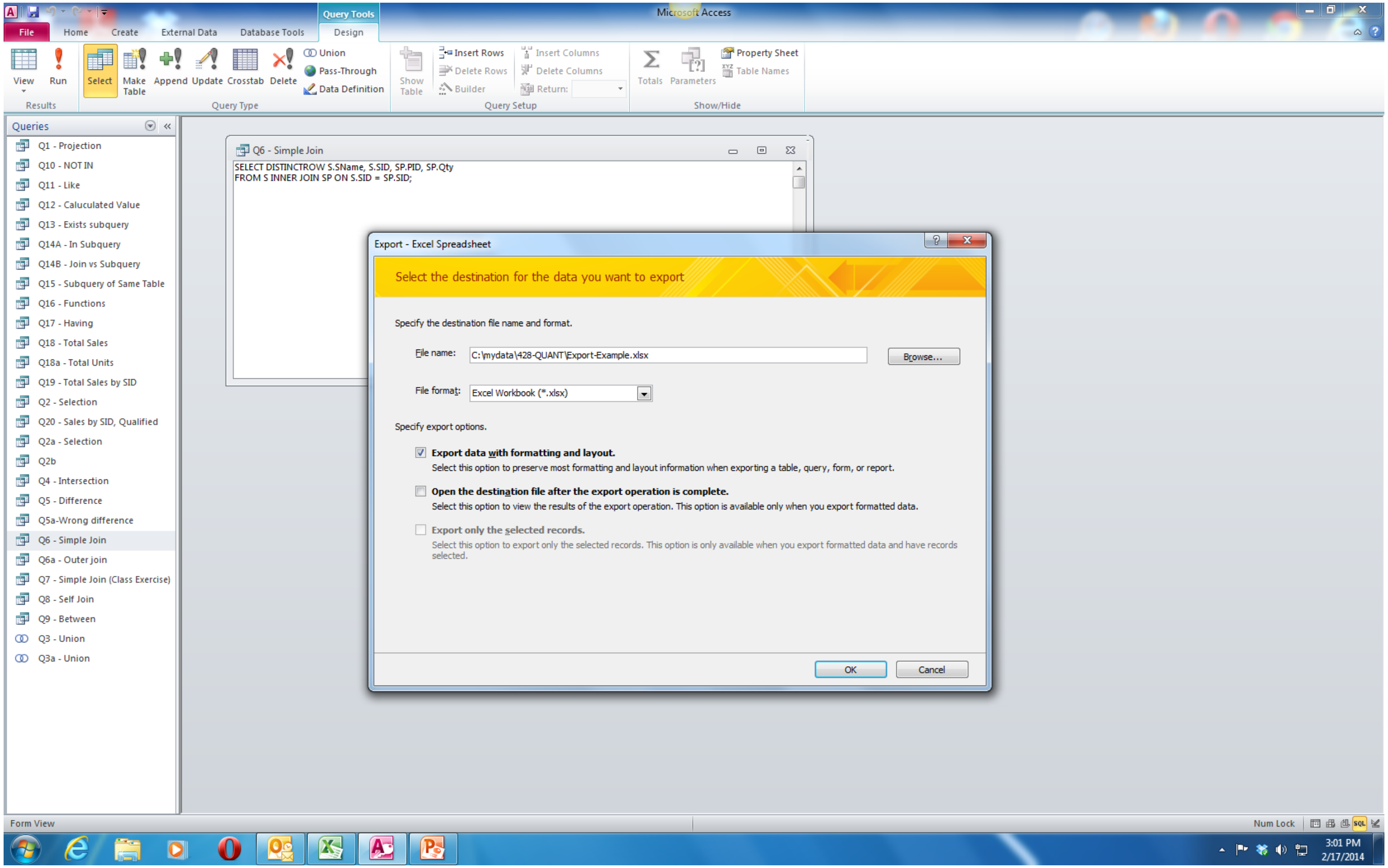


Or from ribbon:

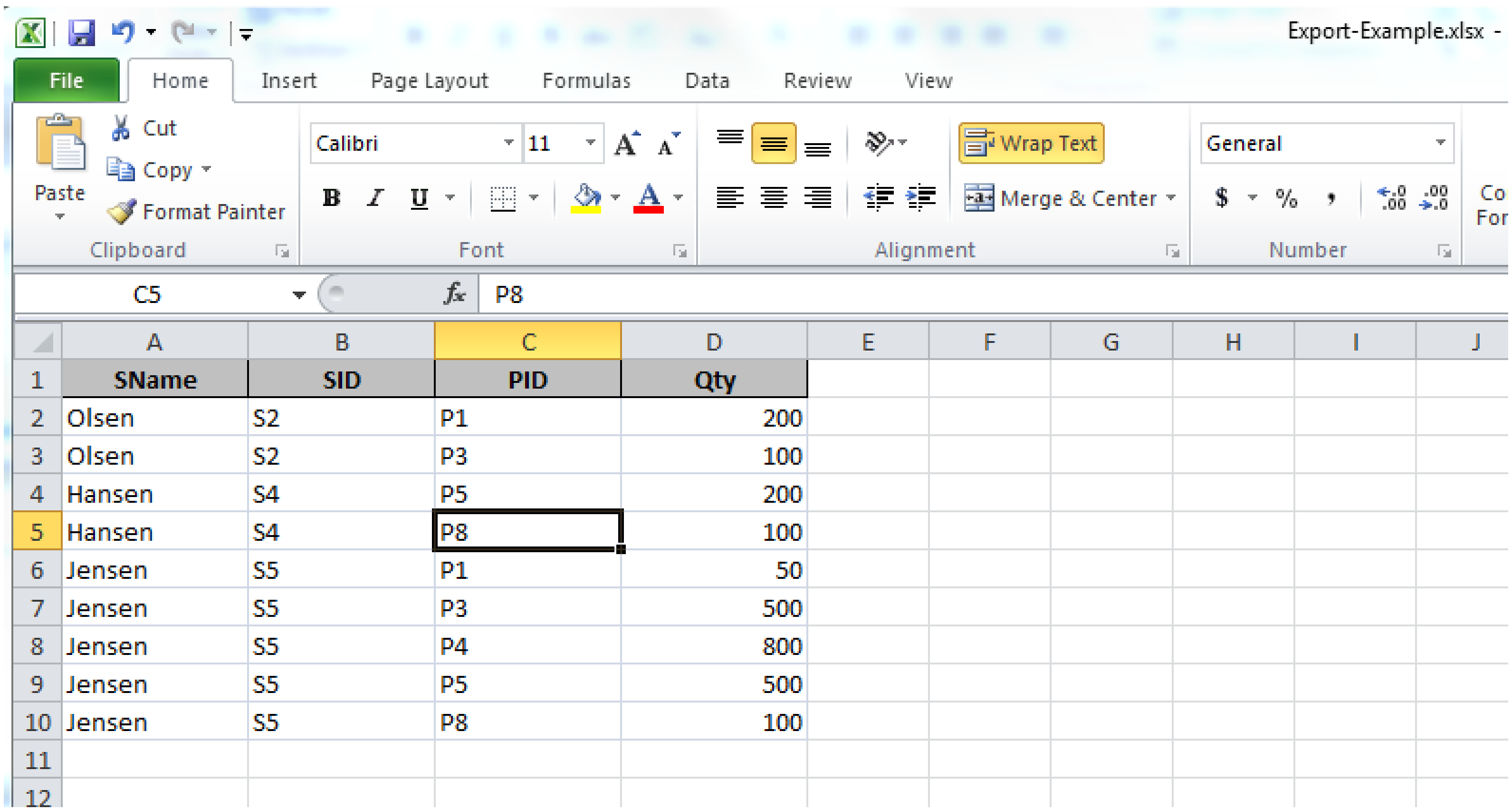
External Data Tab, Export Group, Excel



Export Database Object to Excel (con't)



Export Database Object to Excel (con't)



	A	B	C	D
1	SName	SID	PID	Qty
2	Olsen	S2	P1	200
3	Olsen	S2	P3	100
4	Hansen	S4	P5	200
5	Hansen	S4	P8	100
6	Jensen	S5	P1	50
7	Jensen	S5	P3	500
8	Jensen	S5	P4	800
9	Jensen	S5	P5	500
10	Jensen	S5	P8	100

Export Database Object to Excel (con't)

EXPORT	SOURCE OBJECT	FIELDS AND RECORDS	FORMATTING
Without formatting	Table or query NOTE Forms and reports cannot be exported without their formatting.	All fields and records in the underlying object are exported.	The **Format** property settings are ignored during the operation. For lookup fields, only the lookup ID values are exported. For hyperlink fields, the contents are exported as a text column that displays the links in the format **displaytext#address#**.
With formatting	Table, query, form, or report	Only fields and records that are displayed in the current view or object are exported. Filtered records, hidden columns in a datasheet, and fields not displayed on a form or report are not exported.	The wizard respects the **Format** property settings. For lookup fields, the lookup values are exported. For hyperlink fields, the values are exported as hyperlinks. For rich text fields, the text is exported but the formatting is not.

Export Database Object to Excel (con't)

IF THE DESTINATION WORKBOOK	AND THE SOURCE OBJECT IS	AND YOU WANT TO EXPORT	THEN
Does not exist	A table, query, form, or report	The data, with or without the formatting	The workbook is created during the export operation.
Already exists	A table or query	The data, but not the formatting	The workbook is not overwritten. A new worksheet is added to the workbook, and is given the name of the object from which the data is being exported. If a worksheet having that name already exists in the workbook, Access prompts you to either replace the contents of the corresponding worksheet or specify another name for the new sheet.
Already exists	A table, query, form, or report	The data, including the formatting	The workbook is overwritten by the exported data. All existing worksheets are removed, and a new worksheet having the same name as the exported object is created. The data in the Excel worksheet inherits the format settings of the source object.

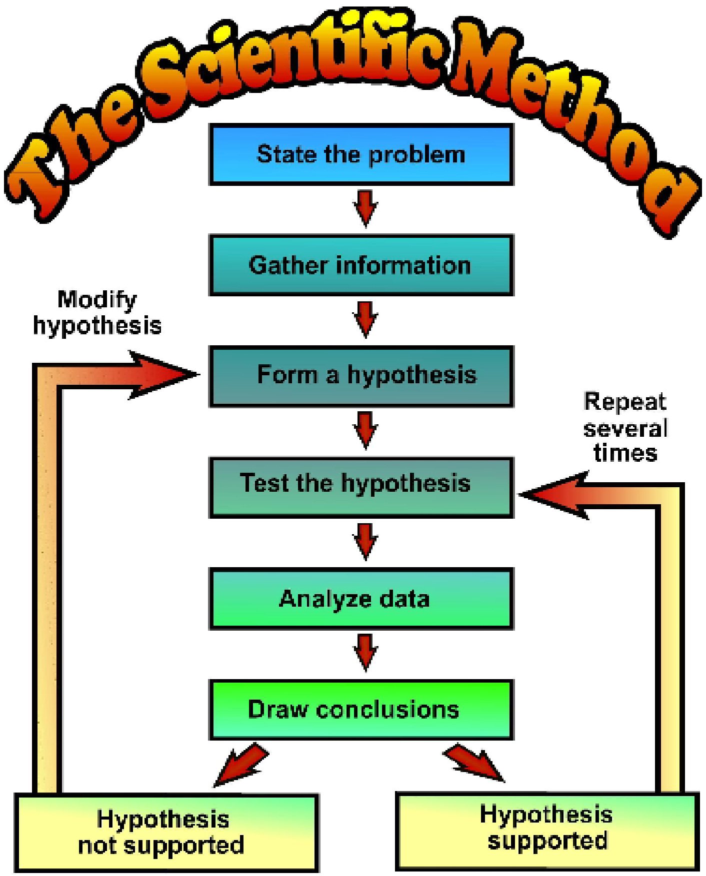
Data Mining

- What is the “scientific method” ?

Don’t look ahead !

The "Scientific Method"

- Formulate a hypothesis
- Gather data:
  - Experiments
  - Surveys
  - Observations
- Use inferential statistics to see if the data supports the hypothesis



Wikipedia

- **Data mining** is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems
- The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use
- Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating

Data Mining Techniques

- Association or affinity analysis uses a specialized set of algorithms that sort through large data sets and express statistical rules among items
- Nearest-neighbor and clustering method
- Text mining and context analysis
- Neural computing is a machine learning approach which examines historical data for patterns
- Intelligent agents retrieving information from the Internet or from intranet-based databases
- Genetic algorithms

Purchase Information

- Purchase patterns of customers (transaction data) contain a huge wealth of information that many business now use for a variety of purposes:
  - Marketing
  - Up selling
  - Cross selling
  - Recommendations
  - Inventory & logistics
  - Store management
  - *This is often combined with shopper ID information*

Affinity Analysis
[Market Basket Analysis]

- This is the most widely used and, in many ways, most successful data mining algorithm
- It essentially determines what products people purchase together
- Stores can use this information to place these products in the same area (particularly preferred brands)
- Direct marketers can use this information to determine which new products to offer to their current customers
- Inventory policies can be improved if reorder points reflect the demand for the complementary products

Association Rules for Market Basket Analysis

Rules are derived in the form "left-hand side implies right-hand side" and an example is:

Yellow Peppers IMPLIES Red Peppers, Bananas

Unidirectional Rules

- The rules are <u>unidirectional</u>
- The following is an "obvious" rule:
  - Caviar IMPLIES Vodka
- But the reverse is not true:
  - Vodka IMPLIES Caviar

Measures of Predictive Ability

[“left-hand side implies right-hand side”]

1. *Support* (prevalence) refers to the percentage of baskets where both left and right side products were present
2. *Confidence* measures what percentage of baskets that contained the left-hand product also contained the right
3. *Lift* measures how much more frequently the left-hand item is found with the right than pure chance (the product of their individual probabilities of occurrence)

Example rule:

- Green Peppers IMPLIES Bananas
  - Confidence – 85.96
    - About 86% of the baskets with green peppers also had bananas
  - Support – 3.77
    - About 4% of the baskets had both green peppers and bananas
  - Lift – 1.37
    - It is 1.37 times more likely to find green peppers with bananas than the product of their individual probabilities (probability of green peppers AND bananas)

Example Analysis

Rule:	Green Peppers IMPLIES Bananas	Red Peppers IMPLIES Bananas	Yellow Peppers IMPLIES Bananas
Lift	1.37	1.43	1.17
Support	3.77	8.58	22.12
Confidence	85.96	89.47	73.09

- The confidence suggests people buying any kind of pepper also buy bananas
- Green peppers sell in about the same quantities as red or yellow (lift), but are not as predictive (support)

Market Basket Analysis Methodology

- We first need a list of transactions of what was purchased - this is readily available with electronic cash registers
- Next, we use a list of products to analyze, and tabulate how many times each was purchased with the others
- The diagonals of the table shows how often a product is purchased in any combination, and the off-diagonals show which combinations were bought

A Small Simple Store Example

Consider the following simple example about five transactions at a convenience store:

Transaction 1: Frozen pizza, cola, milk
Transaction 2: Milk, potato chips
Transaction 3: Cola, frozen pizza
Transaction 4: Milk, pretzels
Transaction 5: Cola, pretzels

Cross Tabulation in a Table

Transaction 1: Frozen pizza, cola, milk
Transaction 2: Milk, potato chips
Transaction 3: Cola, frozen pizza
Transaction 4: Milk, pretzels
Transaction 5: Cola, pretzels

Product Bought	Pizza also	Milk also	Cola also	Chips also	Pretzels also
Pizza	2	1	2	0	0
Milk	1	3	1	1	1
Cola	2	1	3	0	1
Chips	0	1	0	1	0
Pretzels	0	1	1	0	2

- Pizza and Cola sell together more often than any other combo; a cross-marketing opportunity?
- Milk sells well with everything – people probably come here specifically to buy it

Market Basket Concepts

- Transaction – the purchase of one or more items by a customer at one point in time and space – a "shopping cart" or "market basket"
- Association Rule – a rule which suggests a relationship between items in the transaction, written as for single items A and B:
  - A IMPLIES B (or A → B)

Support

- Support – the % of transactions (baskets) where an association rule applies – where we see both item A and B in the same basket
  - For example, if 500 baskets contain both A and B out of a total of 1000 baskets, then the support is 50%
  - A→B and B→A both have the same support

Confidence

- Confidence – measures the predictive accuracy of a rule
- Confidence is the probability that item B is in the basket if item A is in the basket ("conditional probability") → $P(B|A) = P(AB)/P(A)$
- Calculated as:
  - Support (A & B)/P(A) where support (A) is the % of baskets containing A
  - For example, if 500 baskets contain both A and B out of a total of 1000 baskets, then the support of A & B is 50%
  - If A is in 75% of baskets, the confidence is 50/75 or 67%

Lift

- Lift - the ratio of support to a product to the individual probabilities of both sides
  - P(AB)/(P(A) * P(B))
- For example:
  - For example, if 500 baskets contain both A and B out of a total of 1000 baskets, then the support of A & B is 50%
  - If A is in 75% of baskets and B is in 20% of the baskets, then the lift is:
    - .50/(.75*.20) = 3.33

Computing Support

	Pizza	Milk	Cola	Chips	Pretzels
Pizza	2	1	2	0	0
Milk	1	3	1	1	1
Cola	2	1	3	0	1
Chips	0	1	0	1	0
Pretzels	0	1	1	0	2

The support measure for Cola IMPLIES Pizza is 40% (2/5).

Of the 5 transactions 2 have both cola and pizza.

Note support does not consider direction (Pizza IMPLIES Cola is also 40%).

Computing Confidence

	Pizza	Milk	Cola	Chips	Pretzels
Pizza	2	1	2	0	0
Milk	1	3	1	1	1
Cola	2	1	3	0	1
Chips	0	1	0	1	0
Pretzels	0	1	1	0	2

Milk IMPLIES Chips has a confidence of 33%, since the support of “Milk plus Chips” is 20% (1/5) and Milk is in 60% of baskets (3/5).
Thus 20%/60% is 33. Confidence is unidirectional !

Computing Lift

	Pizza	Milk	Cola	Chips	Pretzels
Pizza	2	1	2	0	0
Milk	1	3	1	1	1
Cola	2	1	3	0	1
Chips	0	1	0	1	0
Pretzels	0	1	1	0	2

Lift is the ratio of support of a product to the individual joint probabilities of both sides.

Cola IMPLIES Pizza lift is .40/(.60 * .40) = 1.67.

Using the Results

- The tabulations can immediately be translated into association rules and the numerical measures computed
- Comparing this week's table to last week's table can immediately show the effect of this week's promotional activities
- Some rules are going to be *trivial* (hot dogs and buns sell together) or *inexplicable (*toilet rings sell only when a new hardware store is opened)

Market Basket Illustration Tool

Market Basket Analysis

Shopping Carts

Basket	Item 1	Item 2	Item 3	Item 4
Basket 1	...	...	...	...
Basket 2	...	...	...	...
Basket 3	...	...	...	...
Basket 4	...	...	...	...
Basket 5	...	...	...	...

Available Items

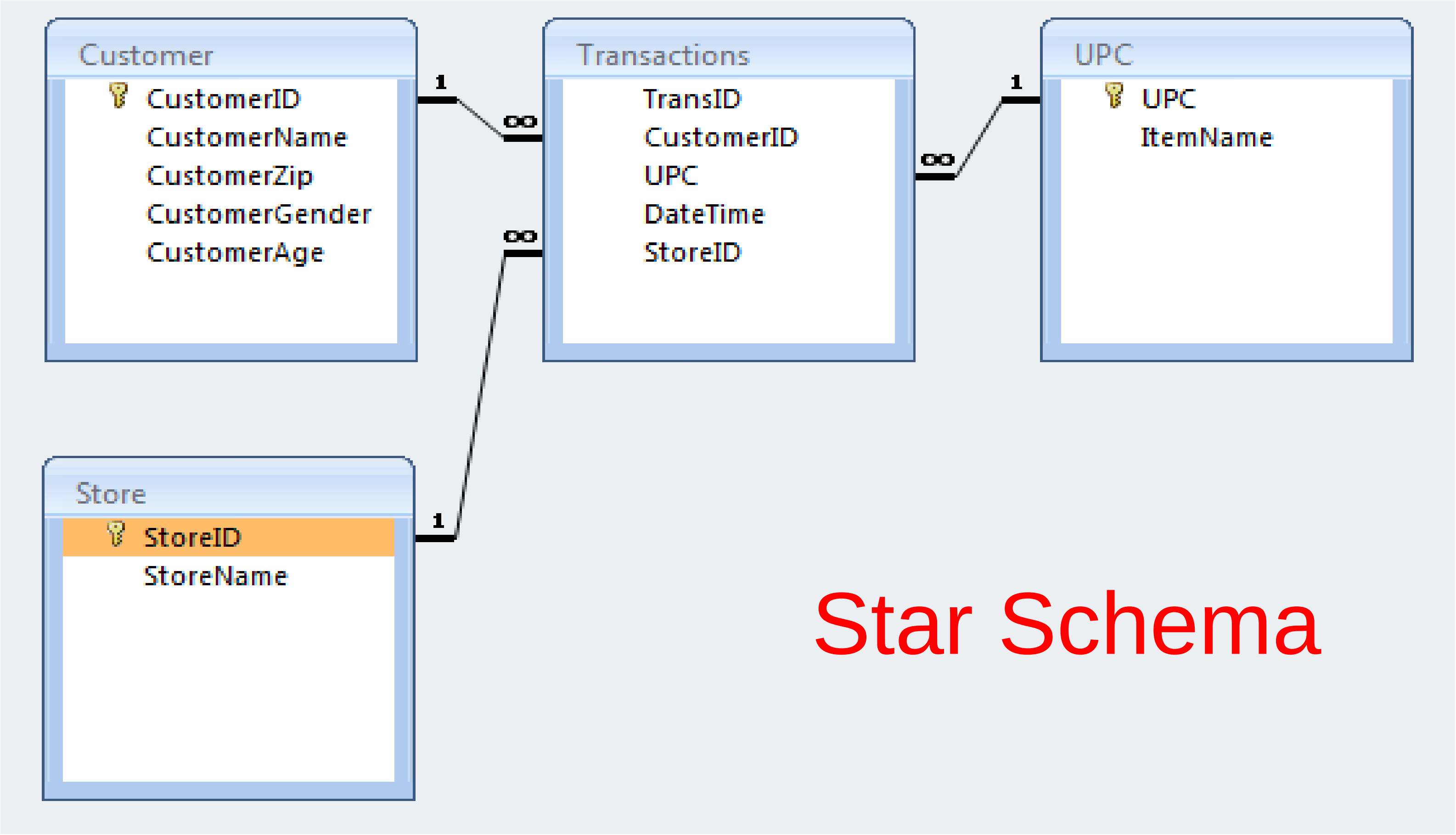
UPC	Name
1	Pizza
2	Milk
3	Cola
4	Chips
5	Pretzels

Current Basket: 0

Hit Auto-Fill button, or manually fill baskets by picking the "current" basket, then pick items to place in that basket.
When baskets are complete, hit calculate button.

Auto-Fill Calculate

Example Database



Example Access Data

UPC

UPC	ItemName
11111	Pizza
22222	Milk
33333	Cola
44444	Chips
55555	Pretzels

Store

StoreID	StoreName
100	Memphis
200	Nashville
300	Jackson

Customer

CustomerID	CustomerName	CustomerZip	CustomerGender	CustomerAge
1	Jones	12345	M	34
2	Adams	23456	F	67
3	Dodd	34567	M	19
4	Zed	45678	F	43
5	Johnson	56789	M	52

Transaction Example Data

Transaction 1: Frozen pizza, cola, milk

Transaction 2: Milk, potato chips

Transaction 3: Cola, frozen pizza

Transaction 4: Milk, pretzels

Transaction 5: Cola, pretzels

Transactions

TransID	CustomerID	UPC	DateTime	StoreID
1	5	11111		100
1	5	33333		100
1	5	22222		100
2	4	22222		100
2	4	44444		100
3	3	33333		100
3	3	11111		100
4	2	22222		100
4	2	55555		100
5	1	33333		100
5	1	55555		100

SQL “Baskets” View

[SELECT Transactions.TransID, UPC.ItemName AS Item
FROM UPC INNER JOIN Transactions ON UPC.UPC=Transactions.UPC;]

Baskets

TransID	Item
1	Pizza
3	Pizza
1	Milk
2	Milk
4	Milk
1	Cola
3	Cola
5	Cola
2	Chips
4	Pretzels
5	Pretzels

Might have many UPC’s
for same product,
such as different sizes.

Cross Product to Find Products Selling Together

[each row in the first table combined with each row in the second table]

Table TABA

Field 1	Field 2
1	Text 1
2	Text 2

Table TABB

Field 3	Field 4	Field 5
1	A	Text 3
1	B	Text 4
2	A	Text 5
2	B	Text 6

Cross product of tables TABA and TABB

Field 1	Field 2	Field 3	Field 4	Field 5
1	Text 1	1	A	Text 3
1	Text 1	1	B	Text 4
1	Text 1	2	A	Text 5
1	Text 1	2	B	Text 6
2	Text 2	1	A	Text 3
2	Text 2	1	B	Text 4
2	Text 2	2	A	Text 5
2	Text 2	2	B	Text 6

SQL “Pairs” View (same basket [transaction])

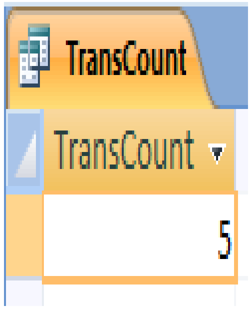
[SELECT T1.Item AS Item1, T2.Item AS Item2
FROM Baskets AS T1, Baskets AS T2
WHERE T1.transID=T2.transID And T1.Item<>T2.Item;]

Pairs

Item1	Item2
Cola	Pizza
Milk	Pizza
Pizza	Cola
Milk	Cola
Pizza	Milk
Cola	Milk
Chips	Milk
Milk	Chips
Pizza	Cola
Cola	Pizza
Pretzels	Milk
Milk	Pretzels
Pretzels	Cola
Cola	Pretzels

SQL Transaction Count

- Standard SQL
  - SELECT count(DISTINCT TransID) AS TransCount
    FROM Baskets
- Access SQL
  - SELECT count(*) AS TransCount
    FROM (SELECT DISTINCT TransID FROM baskets)



SQL Grouping View

- SELECT Item, count(*) AS ItemCount
- FROM Baskets
- GROUP BY Item;

Transaction 1: Frozen pizza, cola, milk
Transaction 2: Milk, potato chips
Transaction 3: Cola, frozen pizza
Transaction 4: Milk, pretzels
Transaction 5: Cola, pretzels

Count of Baskets Containing Items

Item	ItemCount
Chips	1
Cola	3
Milk	3
Pizza	2
Pretzels	2

SQL Support Count

- From the transactions, how many are for each pair:
  - SELECT Item1, Item2, count(*) AS SupportCount
  - FROM Pairs
  - GROUP BY Item1, Item2;

SupportCount

Item1	Item2	SupportCount
Chips	Milk	1
Cola	Milk	1
Cola	Pizza	2
Cola	Pretzels	1
Milk	Chips	1
Milk	Cola	1
Milk	Pizza	1
Milk	Pretzels	1
Pizza	Cola	2
Pizza	Milk	1
Pretzels	Cola	1
Pretzels	Milk	1

SQL Support

- From the transactions, how many are for each pair as a percentage of the total transactions:
  - SELECT Item1, Item2, count(*) AS SupportCount, count(*)/(SELECT count(*) AS TransCount FROM (SELECT DISTINCT transID FROM transactions)) AS Support
  - FROM Pairs
  - GROUP BY Item1, Item2;

Support (con't)

Support

Item1	Item2	SupportCount	Support
Chips	Milk	1	0.2
Cola	Milk	1	0.2
Cola	Pizza	2	0.4
Cola	Pretzels	1	0.2
Milk	Chips	1	0.2
Milk	Cola	1	0.2
Milk	Pizza	1	0.2
Milk	Pretzels	1	0.2
Pizza	Cola	2	0.4
Pizza	Milk	1	0.2
Pretzels	Cola	1	0.2
Pretzels	Milk	1	0.2

Cola IMPLIES Pizza support is 40%; of the 5 transactions, 2 have both Cola and Pizza.

Pizza IMPLIES Cola is also 40% (support does not consider direction)

SQL Confidence

- Support divided by % of baskets containing the first product in the rule
  - SELECT Item1, Item2, count(*) AS SupportCount, count(*)/(SELECT count(*) AS TransCount FROM (SELECT DISTINCT transID FROM baskets)) AS Support, (select count(*) from baskets where Item=Item1)/(SELECT count(*) AS TransCount FROM (SELECT DISTINCT transID FROM baskets)) AS Item1inBaskets, Support/Item1inBaskets AS Confidence
  - FROM Pairs
  - GROUP BY Item1, Item2;

Confidence (con’t)

Support-Confidence

Item1	Item2	SupportCount	Support	Item1inBaskets	Confidence
Chips	Milk	1	0.2	0.2	1
Cola	Milk	1	0.2	0.6	0.333333333333333
Cola	Pizza	2	0.4	0.6	0.666666666666667
Cola	Pretzels	1	0.2	0.6	0.333333333333333
Milk	Chips	1	0.2	0.6	0.333333333333333
Milk	Cola	1	0.2	0.6	0.333333333333333
Milk	Pizza	1	0.2	0.6	0.333333333333333
Milk	Pretzels	1	0.2	0.6	0.333333333333333
Pizza	Cola	2	0.4	0.4	1
Pizza	Milk	1	0.2	0.4	0.5
Pretzels	Cola	1	0.2	0.4	0.5
Pretzels	Milk	1	0.2	0.4	0.5

Milk IMPLIES Chips has a confidence of .33 [.2 divided by .6]
Chips IMPLIES Milk has a confidence of 1

SQL Lift

- Lift is the ratio of support to the product of the individual probabilities
  - SELECT Item1, Item2, count(*) AS SupportCount, count(*)/(SELECT count(*) AS TransCount FROM (SELECT DISTINCT transID FROM baskets)) AS Support, (select count(*) from baskets where Item=Item1)/(SELECT count(*) AS TransCount FROM (SELECT DISTINCT transID FROM baskets)) AS Item1inBaskets, Support/Item1inBaskets AS Confidence, (select count(*) from baskets where Item=Item2)/(SELECT count(*) AS TransCount FROM (SELECT DISTINCT transID FROM baskets)) AS Item2inBaskets, Support/(Item1inBaskets*Item2inBaskets) AS Lift
  - FROM Pairs
  - GROUP BY Item1, Item2;

Lift (con't)

Support-Confidence-Lift

Item1	Item2	SupportCount	Support	Item1inBaskets	Confidence	Item2inBaskets	Lift
Chips	Milk	1	0.2	0.2	1	0.6	1.66666666666667
Cola	Milk	1	0.2	0.6	0.333333333333333	0.6	0.555555555555556
Cola	Pizza	2	0.4	0.6	0.666666666666667	0.4	1.66666666666667
Cola	Pretzels	1	0.2	0.6	0.333333333333333	0.4	0.833333333333333
Milk	Chips	1	0.2	0.6	0.333333333333333	0.2	1.66666666666667
Milk	Cola	1	0.2	0.6	0.333333333333333	0.6	0.555555555555556
Milk	Pizza	1	0.2	0.6	0.333333333333333	0.4	0.833333333333333
Milk	Pretzels	1	0.2	0.6	0.333333333333333	0.4	0.833333333333333
Pizza	Cola	2	0.4	0.4	1	0.6	1.66666666666667
Pizza	Milk	1	0.2	0.4	0.5	0.6	0.833333333333333
Pretzels	Cola	1	0.2	0.4	0.5	0.6	0.833333333333333
Pretzels	Milk	1	0.2	0.4	0.5	0.6	0.833333333333333

The lift for the rule "Cola IMPLIES Pizza" is .4/ (.6 * .4) = 1.67

Selecting Rules → "Mining"

- To select the relevant rules, one would select rows from the previous table where the support, confidence, and lift met minimum criteria
  - SELECT Item1, Item2, Support, Confidence, Lift
  - FROM [Support-Confidence-Lift]
  - WHERE Support>=0.4 AND Confidence>=1 AND Lift>=1;

Rules

Item1	Item2	Support	Confidence	Lift
Pizza	Cola	0.4	1	1.66666666666667

Performing Analysis with Virtual Items

- The sales data can be augmented with the addition of “virtual items” -- For example, we could record that the customer was new to us, or had children
- The transaction record might look like:

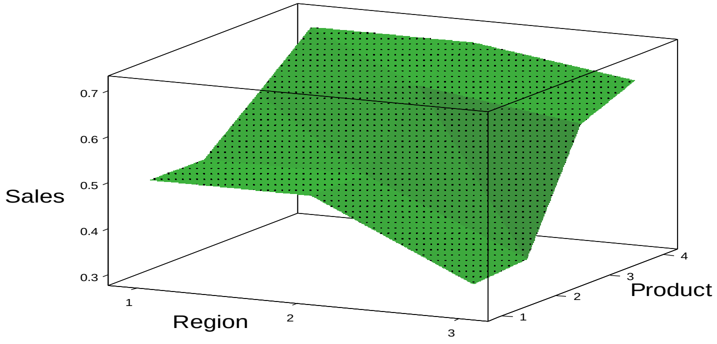
  Item 1: Sweater   Item 2: Jacket   Item 3: New

- This might allow us to see what patterns new customers have versus old customers

Multidimensional Market Basket Analysis

- Rules can involve more than two items, for example Plant and Clay Pot IMPLIES Soil
- These rules are built iteratively -- first, pairs are found, then relevant sets of three or four
- In our example here, one would join the “pairs” table to itself, to formulate a “triples” table
- These are then pruned by removing those that occur infrequently
- In an environment like a grocery store, where customers commonly buy over 100 items, rules could involve as many as 10 items

Online Analytical Processing



Traditional SQL Queries

Queries allow users to request information from the computer that is not available in periodic reports

Query systems are often based on menu/GUI based programs (which generate SQL) or via direct structured query language (SQL) or using a query-by-example (QBE) method

- User requests are stated in a query language and the results are subsets of the data in the relational tables:
  - Sales by department by customer type for specific period
  - Weather conditions for specific date
  - Sales by day of week
  - ...

Wikipedia

- In computing, **online analytical processing**, or **OLAP,** is an approach to answering multi-dimensional analytical queries swiftly
- OLAP tools enable users to analyze multidimensional data interactively from multiple perspectives
- OLAP consists of three basic analytical operations: consolidation (roll-up), drill-down, and slicing and dicing

OLAP

- On Line Analytical Processing (OLAP) is a relatively new way of storing, viewing, and presenting information
- With it, data is viewed in cubes
- A two dimensional cube can be viewed as a table
- A three dimensional cube as a “cube”
- A multidimensional cube as a “hypercube”
- These cubes have axes, dimensions, measures, slices, and levels

Example: Relational Source Data

Category	Type	City	State	Date	Sales Price	Asking Price
New	Single Family	San Francisco	California	1/1/2000	679,000	685,000
Existing	Condo	Los Angeles	California	3/5/2001	327,989	350,000
Existing	Single Family	Elko	Nevada	7/17/2001	105,675	125,000
New	Condo	San Diego	California	12/22/2000	375,000	375,000
Existing	Single Family	Paradise	California	11/19/2001	425,000	449,000
Existing	Single Family	Las Vegas	Nevada	1/19/2001	317,000	325,000
New	Single Family	San Francisco	California	1/1/2000	679,000	685,000
Existing	Condo	Los Angeles	California	3/5/2001	327,989	350,000
Existing	Condo	Las Vegas	Nevada	6/19/2001	297,000	305,000
Existing	Single Family	Los Angeles	California	4/1/2000	579,000	625,000
New	Condo	Los Angeles	California	8/5/2001	321,000	320,000
Etc.						

What is the average sales price for new single family homes in LA in the 2QT of 2001 ?

Example: OLAP Cube for Average Sales Price

[2 “axes” (rows and columns): date “dimensions” and type “dimensions”]

Average Sales Price of Single-Family Dwellings ($thousands)										
			Existing Structures				New Construction			
			California			Nevada	California			Nevada
			San Francisco	Los Angeles	San Diego		San Francisco	Los Angeles	San Diego	
2000	Q1	Jan	408	465	375	179	418	468	371	190
		Feb	419	438	382	180	429	437	382	185
		Mar	427	477	380	195	426	471	387	198
	Q2		433	431	382	188	437	437	380	193
	Q3		437	437	380	190	438	439	382	190
	Q4		435	439	377	193	432	434	370	198
2001	Q1	Jan	452	454	368	198	450	457	367	197
		Feb	450	467	381	187	457	464	388	191
		Mar	432	444	373	188	436	446	371	201
	Q2		437	437	368	190	444	432	363	196
	Q3		436	452	388	196	447	455	385	199
	Q4		441	455	355	198	449	455	355	202

What is the average sales price for new single family homes in LA in the 2QT of 2001 ?

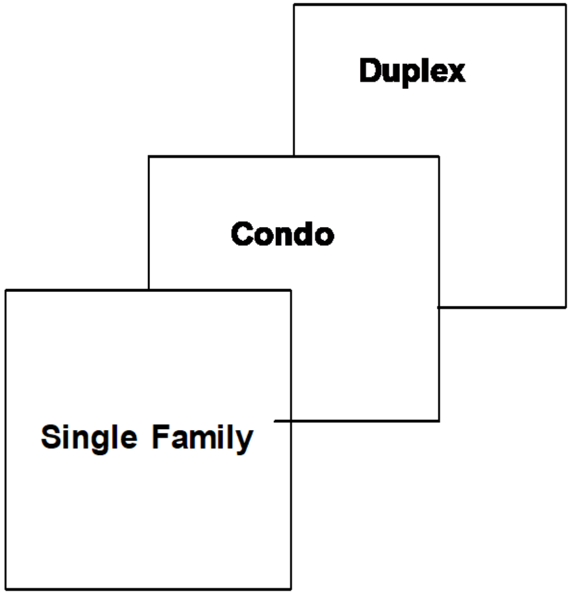
OLAP (con't)

Average Sales Price of Single-Family Dwellings ($thousands)										
			Existing Structures				New Construction			
			California			Nevada	California			Nevada
			San Francisco	Los Angeles	San Diego		San Francisco	Los Angeles	San Diego	
2000	Q1	Jan	408	465	375	179	418	468	371	190
		Feb	419	438	382	180	429	437	382	185
		Mar	427	477	380	195	426	471	387	198
	Q2		433	431	382	188	437	437	380	193
	Q3		437	437	380	190	438	439	382	190
	Q4		435	439	377	193	432	434	370	198
2001	Q1	Jan	452	454	368	198	450	457	367	197
		Feb	450	467	381	187	457	464	388	191
		Mar	432	444	373	188	436	446	371	201
	Q2		437	437	368	190	444	432	363	196
	Q3		436	452	388	196	447	455	385	199
	Q4		441	455	355	198	449	455	355	202

- When 2 or more dimensions are shown on one axis, then <u>every combination (relational column) of one must be shown with the other</u>
- Notice the same sub categories under both existing and new construction categories, and same categories under 2000 and 2001
- The cells of the OLAP cube hold the "measures" (the data); here the measure is sales price for single family homes

OLAP Slices

- This OLAP cube is just for the average sales price of single family homes; there would be another cube for the average sales price of condos
- You could think of these two cubes as one behind the other, or as "slices"
- We could also have slices for "sales price" and "asking price"



“Members” and “Levels”

- The values of a dimension are called “members”
- The members of the type dimension are single and condo
- The members of the category dimension are new and existing
- For this data set, the members of the state dimension are CA an NV
- Some members may be computed such as date and/or time
- The “level” of a dimension is its position in the hierarchy; the levels of the date dimension are year, quarter, and month

Average Sales Price of Single-Family Dwellings ($thousands)										
			Existing Structures				New Construction			
			California			Nevada	California			Nevada
			San Francisco	Los Angeles	San Diego		San Francisco	Los Angeles	San Diego	
2000	Q1	Jan	408	465	375	179	418	468	371	190
		Feb	419	438	382	180	429	437	382	185
		Mar	427	477	380	195	426	471	387	198
	Q2		433	431	382	188	437	437	380	193
	Q3		437	437	380	190	438	439	382	190
	Q4		435	439	377	193	432	434	370	198
2001	Q1	Jan	452	454	368	198	450	457	367	197
		Feb	450	467	381	187	457	464	388	191
		Mar	432	444	373	188	436	446	371	201
	Q2		437	437	368	190	444	432	363	196
	Q3		436	452	388	196	447	455	385	199
	Q4		441	455	355	198	449	455	355	202

OLAP Terminology

- OLAP hypercube: means a data display with an unlimited number of axes

Term	Description	Example in Figure
Axis	A coordinate of the hypercube	Rows, columns
Dimension	A feature of the data to be placed on an axis	Time, Housing Type, Location
Level	A (hierarchical) subset of a dimension	{California, Nevada} {San Francisco, Los Angeles, Other} {Q1, Q2, Q3, Q4}
Member	A data value in a dimension	{New, Existing}, {Jan, Feb, Mar}
Measure	The source data for the hypercube	Sales Price, Asking Price
Slice	A dimension or measure held constant for the display	Housing Type—all shown are for Single Family—another cube exists for Condo

OLAP Cube Data Definition

[4 dimensions, 2 slices (sales and asking price)]

```
CREATE CUBE HousingSalesCube (
    DIMENSION Time TYPE TIME,
        LEVEL Year TYPE YEAR,
        LEVEL Quarter TYPE QUARTER,
        LEVEL Month TYPE MONTH,
    DIMENSION Location,
        LEVEL USA TYPE ALL,
        LEVEL State,
        LEVEL City,
    DIMENSION HousingCategory,
    DIMENSION HousingType,
    MEASURE SalesPrice,
        FUNCTION AVG
    MEASURE AskingPrice,
        FUNCTION AVG
        )
```

Compare to SQL Data Definition Language: “create table”

Multidimensional SELECT Statement [produces this "view"]

SELECT CROSSJOIN

({Existing Structure, New Construction},

{California.Children, Nevada})

ON COLUMNS,

{2000.Q1.Children, 2000.Q2, 2000.Q3, 2000.Q4,

2001.Q1.Children, 2001.Q2, 2001.Q3, 2001.Q4}

ON ROWS

FROM HousingSalesCube

WHERE (SalesPrice, HousingType = 'SingleFamily')

Average Sales Price of Single-Family Dwellings ($thousands)										
			Existing Structures				New Construction			
			California			Nevada	California			Nevada
			San Francisco	Los Angeles	San Diego		San Francisco	Los Angeles	San Diego	
2000	Q1	Jan	408	465	375	179	418	468	371	190
		Feb	419	438	382	180	429	437	382	185
		Mar	427	477	380	195	426	471	387	198
	Q2		433	431	382	188	437	437	380	193
	Q3		437	437	380	190	438	439	382	190
	Q4		435	439	377	193	432	434	370	198
2001	Q1	Jan	452	454	368	198	450	457	367	197
		Feb	450	467	381	187	457	464	388	191
		Mar	432	444	373	188	436	446	371	201
	Q2		437	437	368	190	444	432	363	196
	Q3		436	452	388	196	447	455	385	199
	Q4		441	455	355	198	449	455	355	202

The OLAP "crossjoin" ({X,Y}, {A,B}) creates a view

where X and Y are the main categories

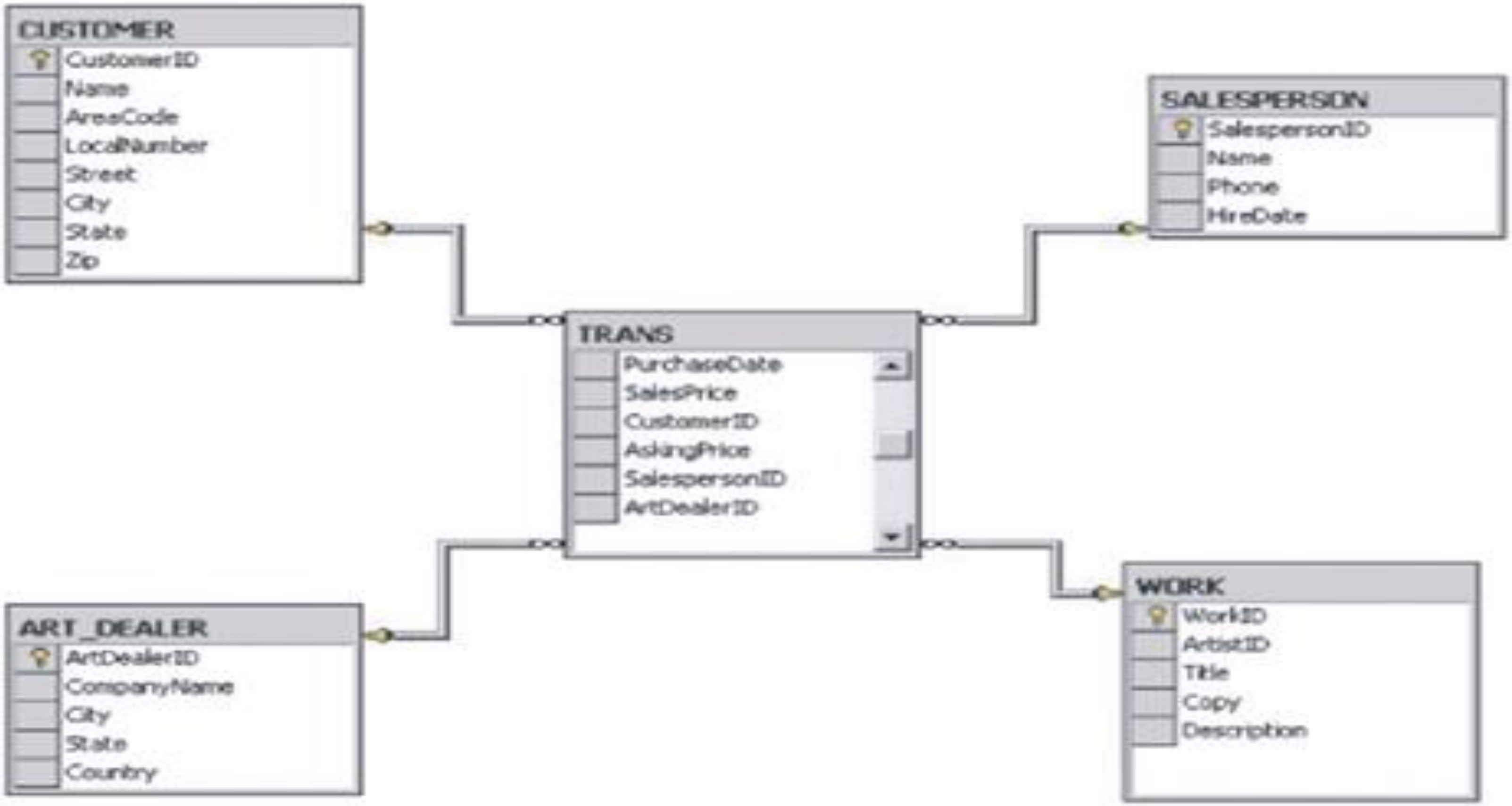
A and B are sub categories under both X and Y

Crossjoins are created on the columns or rows

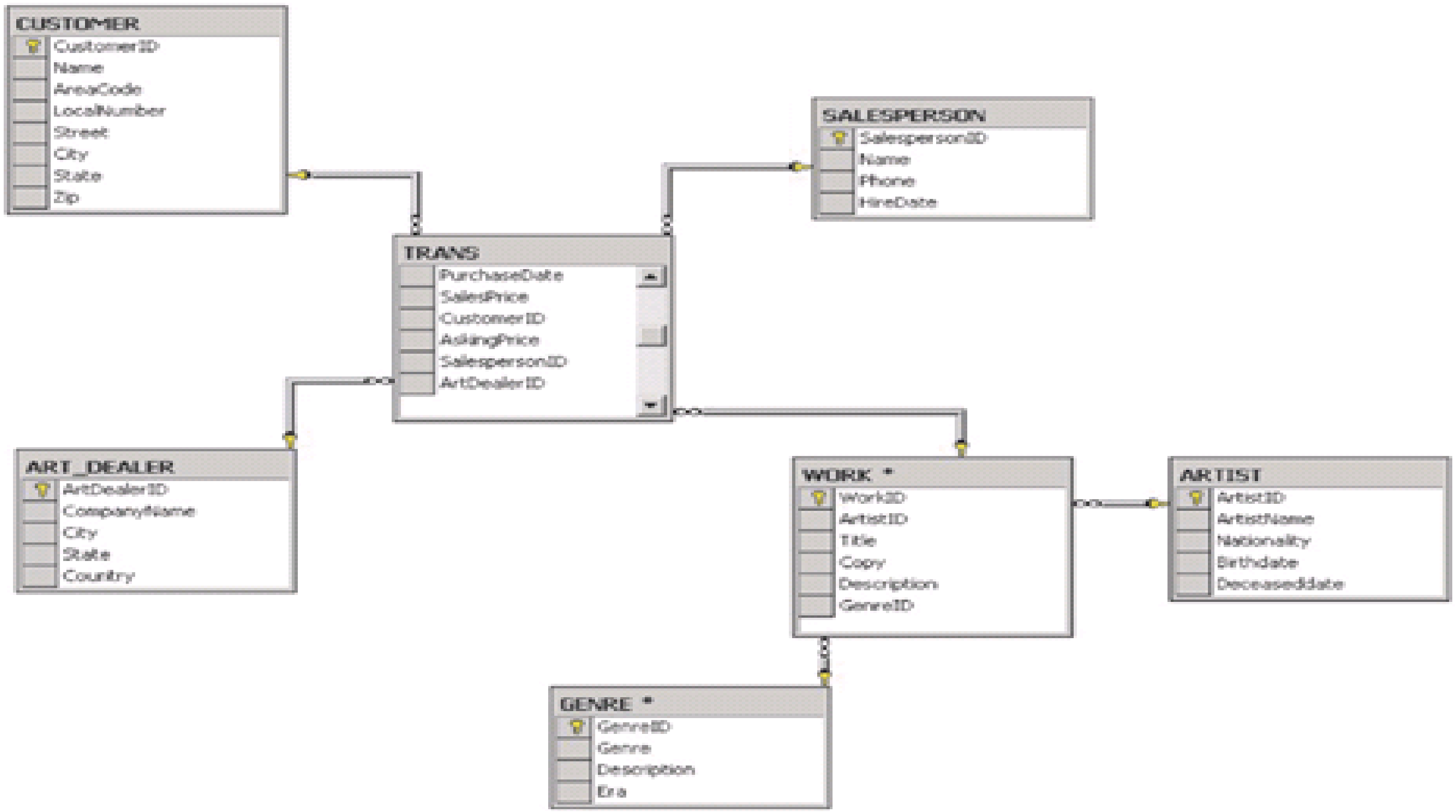
Related Tables

- The previous example only had one relational table
- Most databases have multiple tables with primary/foreign key relationships
- Often the dimensions are held as foreign keys in the “cube” table, with relations to other tables (“member data”) with the details about each dimension

Example: Star Schema



Example: Snowflake Schema



MOLAP & ROLAP

- OLAP servers typically come in two basic flavors
- Some servers have specialized data stores which store data in a form which is highly effective for multidimensional analysis
  - These servers are termed MOLAP and they tend to have exceptional performance due to their specialized data store
- Loading data into a MOLAP server usually takes a very long time because many of the answers in the cube must be calculated (the extra time spent during the load is usually called “processing” time)
- A relational OLAP (or ROLAP) server uses data stored in an RDBMS
  - These systems trade the performance of a multidimensional store for the convenience of an RDBMS. These servers almost always query over a database which is structured as a star or snowflake type schema.
- An OLAP server usually returns information to the user as a ‘pivot table‘ or ‘pivot report’ which are now supported in both Excel and Access

Pivot Table

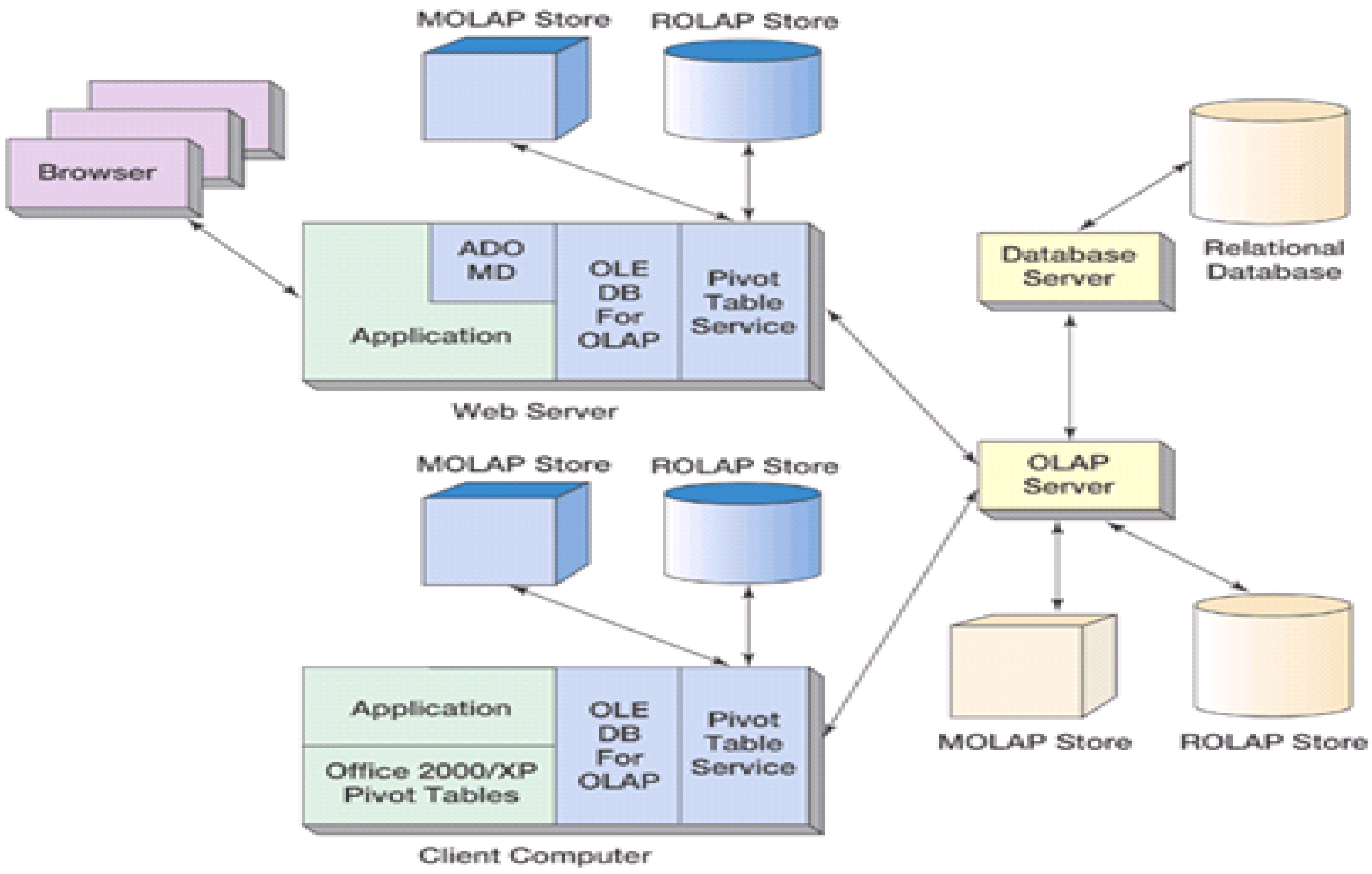
Data before pivoting:

	A	B	C	D	E	F	G
1	**Region**	**Gender**	**Style**	**Ship Date**	**Units**	**Price**	**Cost**
2	East	Boy	Tee	1/31/2005	12	11.04	10.42
3	East	Boy	Golf	1/31/2005	12	13	12.6
4	East	Boy	Fancy	1/31/2005	12	11.96	11.74
5	East	Girl	Tee	1/31/2005	10	11.27	10.56
6	East	Girl	Golf	1/31/2005	10	12.12	11.95
7	East	Girl	Fancy	1/31/2005	10	13.74	13.33
8	West	Boy	Tee	1/31/2005	11	11.44	10.94
9	West	Boy	Golf	1/31/2005	11	12.63	11.73
10	West	Boy	Fancy	1/31/2005	11	12.06	11.51
11	West	Girl	Tee	1/31/2005	15	13.42	13.29
12	West	Girl	Golf	1/31/2005	15	11.48	10.67

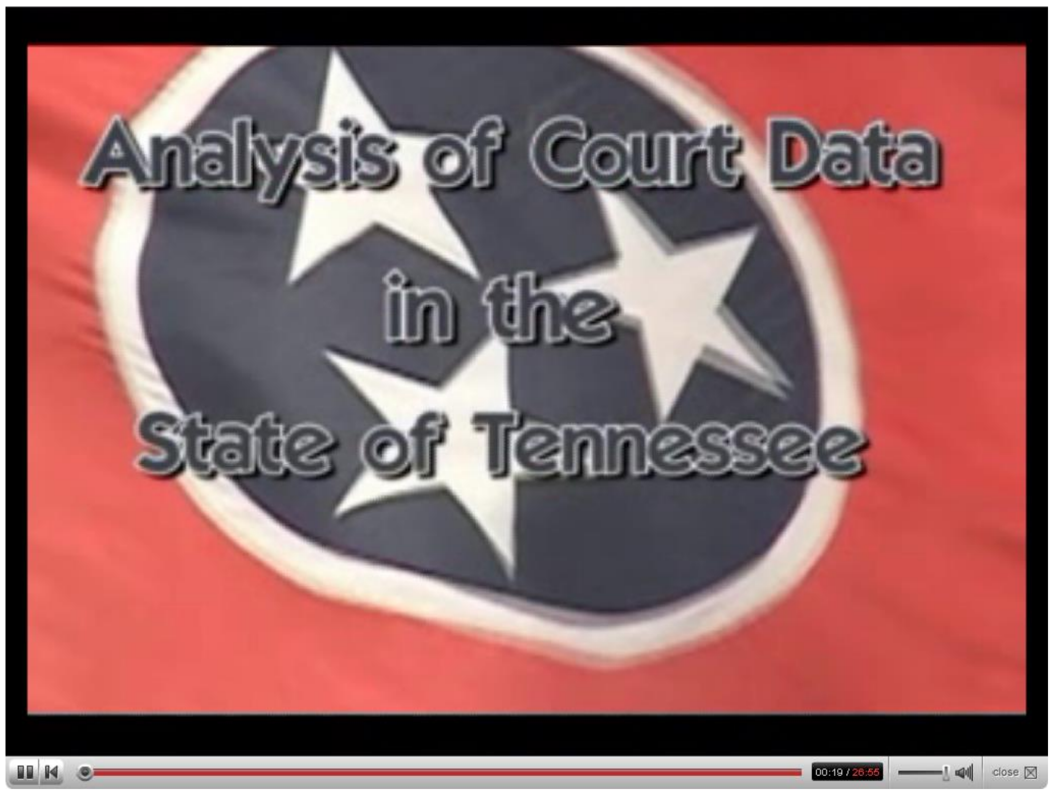
Data summarized in pivot form:

Sum of Units	Ship Date					
Region	1/31/2005	2/28/2005	3/31/2005	4/30/2005	5/31/2005	6/30/2005
East	66	80	102	116	127	125
North	96	117	138	151	154	156
South	123	141	157	178	191	202
West	78	97	117	136	150	157
(blank)						
Grand Total	363	435	514	581	622	640

Microsoft OLAP Architecture



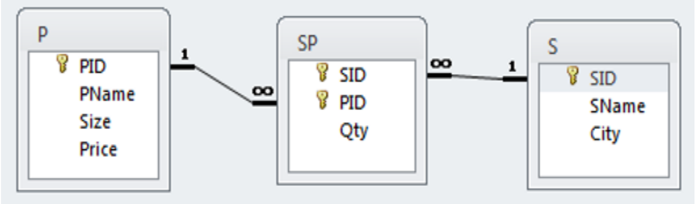
TN OLAP Application



OLAP Vendors

- SQL Server
- Oracle
- IBM
- SAS
- Teradata
- 1010 Data
- Information Builders
- Open Source (interface to MySQL), such as Mondrian

Earlier Access Model



S

SID	SName	City
S1	Peterson	Aarhus
S2	Olsen	Copenhagen
S4	Hansen	Odense
S5	Jensen	Copenhagen

P

PID	PName	Size	Price
P1	Shirt	6	$50.00
P3	Trousers	5	$90.00
P4	Socks	7	$20.00
P5	Blouse	6	$50.00
P8	Blouse	8	$60.00

SP

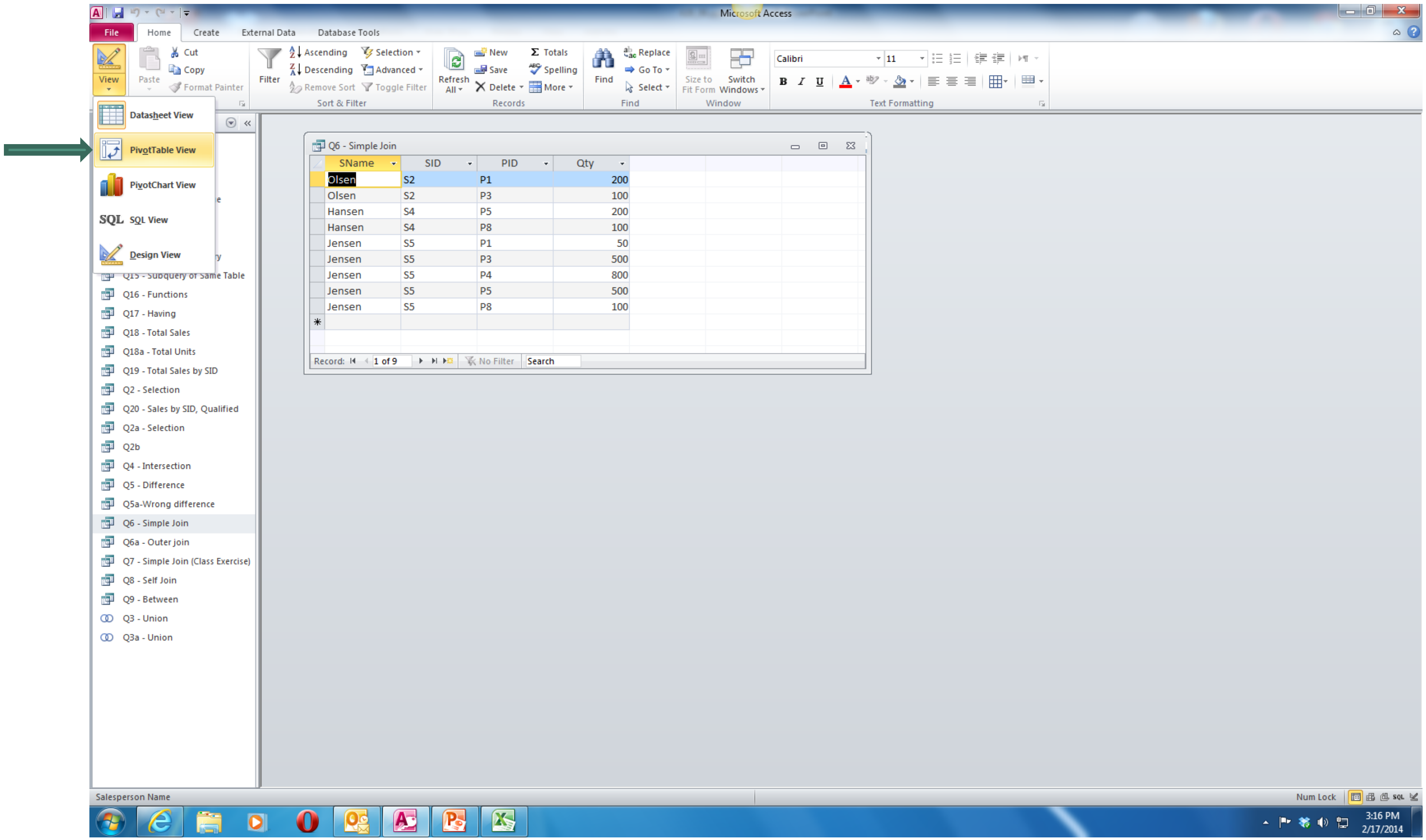
SID	PID	Qty
S2	P1	200
S2	P3	100
S4	P5	200
S4	P8	100
S5	P1	50
S5	P3	500
S5	P4	800
S5	P5	500
S5	P8	100

How many shirts have been sold ?

How many items has Jensen sold ?

How many items have been sold in Copenhagen ?

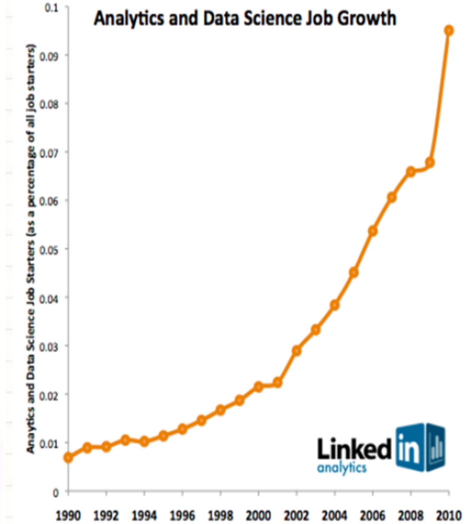
Pivot Table (OLAP) View



Access Pivot Table

		PName				
		Blouse	Shirt	Socks	Trousers	Grand Total
City	SName	Sum of Qty	Sum of Qty	Sum of Qty	Sum of Qty	Sum of Qty
Copenhagen	Jensen	600	50	800	500	1950
	Olsen		200		100	300
	Total	600	250	800	600	2250
Odense	Hansen	300				300
	Total	300				300
Grand Total		900	250	800	600	2550

How many shirts have been sold ?
How many items has Jensen sold ?
How many items have been sold in Copenhagen ?



ANALYTICS

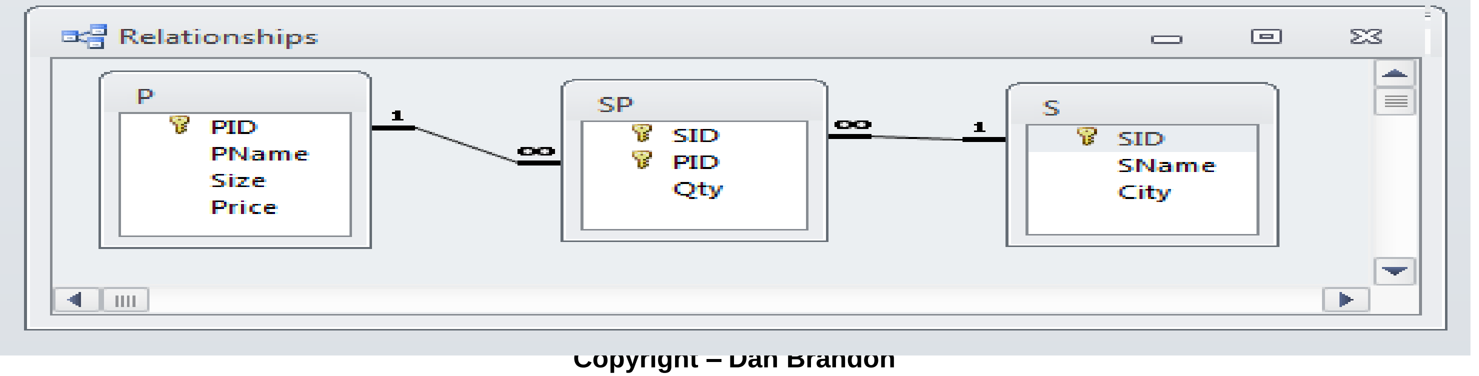
Know what's hot.

References

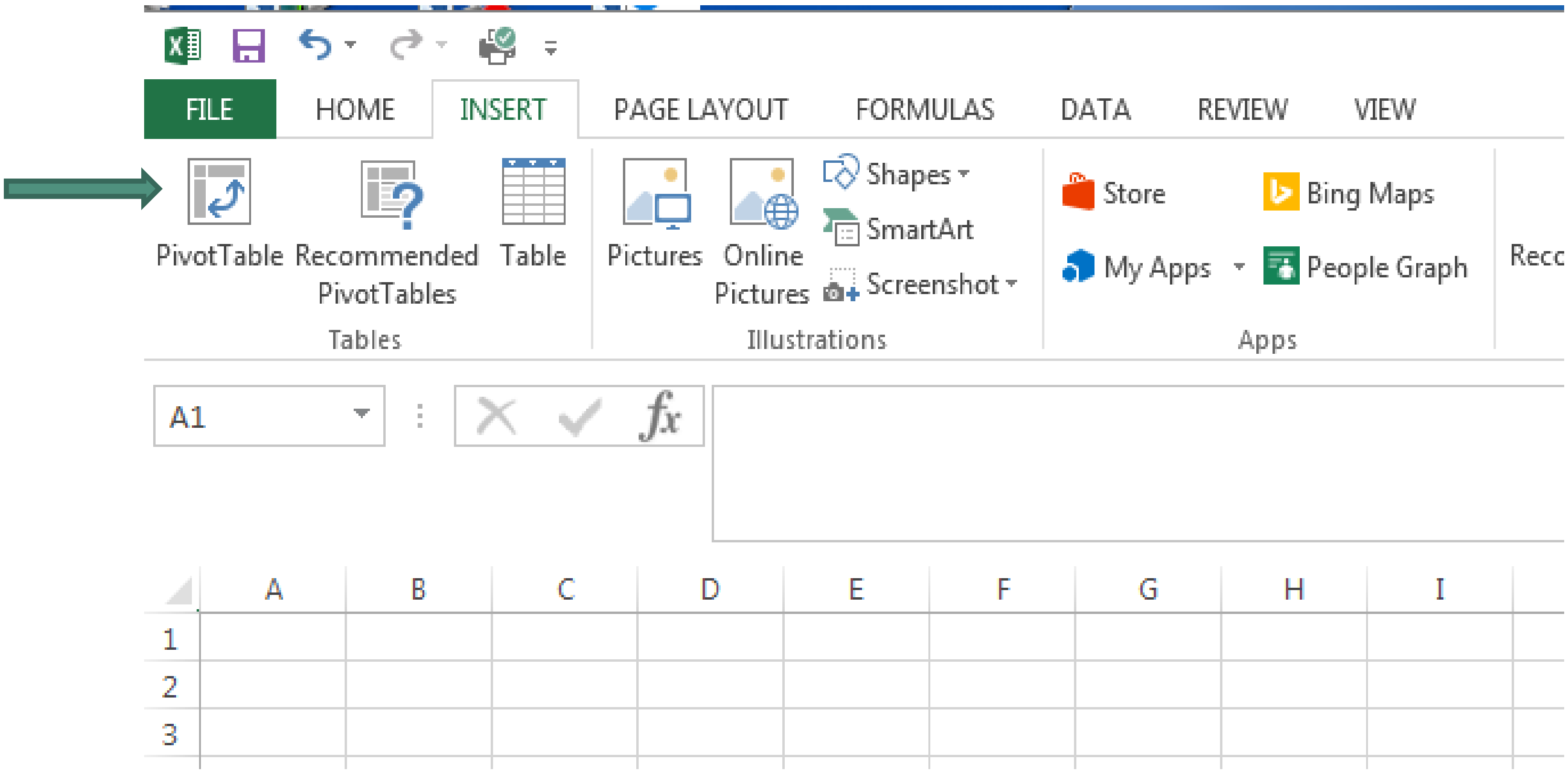
- The Language of SQL: How to Access Data in Relational Databases by Larry Rockoff (Jun 3, 2010)
- SQL All-in-One For Dummies by Allen G. Taylor (Apr 5, 2011)
- The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling by Ralph Kimball and Margy Ross (Jul 1, 2013)
- Joe Celko's Analytics and OLAP in SQL (The Morgan Kaufmann Series in Data Management Systems) by Joe Celko (Aug 7, 2006)
- Business Intelligence For Dummies by Swain Scheps (Dec 21, 2007)
- Introduction to Data Mining by Pang-Ning Tan (Nov 13, 2014)
- Data Mining: Concepts and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems) by Jiawei Han, Micheline Kamber and Jian Pei (Jul 6, 2011)
- Data Science for Business: What you need to know about data mining and data-analytic thinking by Foster Provost and Tom Fawcett (Aug 16, 2013)

Project 5

- Create an OLAP model (pivot table) for the Access S-P-SP problem with one dimension for products (by name) and the other dimension for city/salesperson (by name) – use the quantity as the measure (metric)
  - Note: Microsoft Access (not SQLServer) dropped pivot tables starting with Office 2013 – need to export query to Excel and use Excel pivot table, or use new Get & Transform tools in Excel→

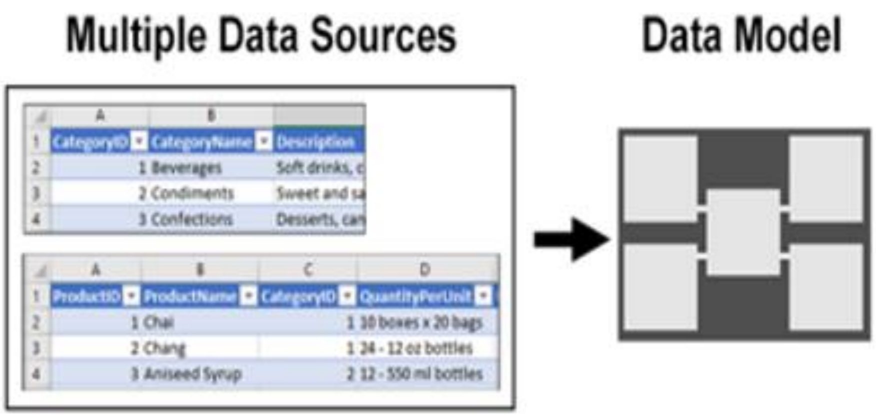


Excel Traditional Pivot Table



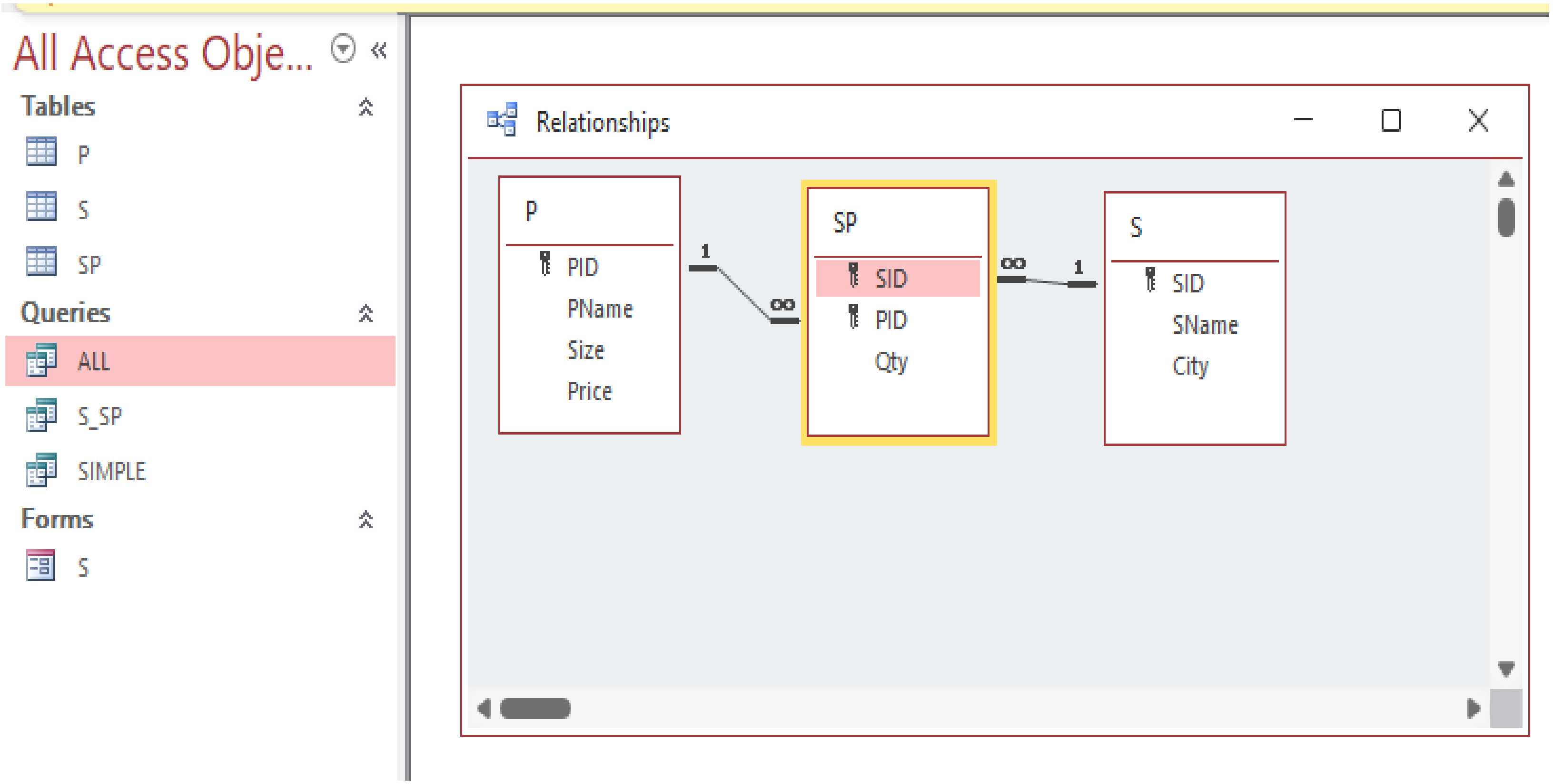
Excel OLAP (Data Models)

- Latest versions of Excel have "Get & Transform" which can import directly from relational databases such as Access, SQLServer, MySQL, Oracle, etc.
- Import can be to a table to build a traditional pivot table) , data model (Power PivotTable), or PivotChart
- Tables <u>and</u> relationships can be imported



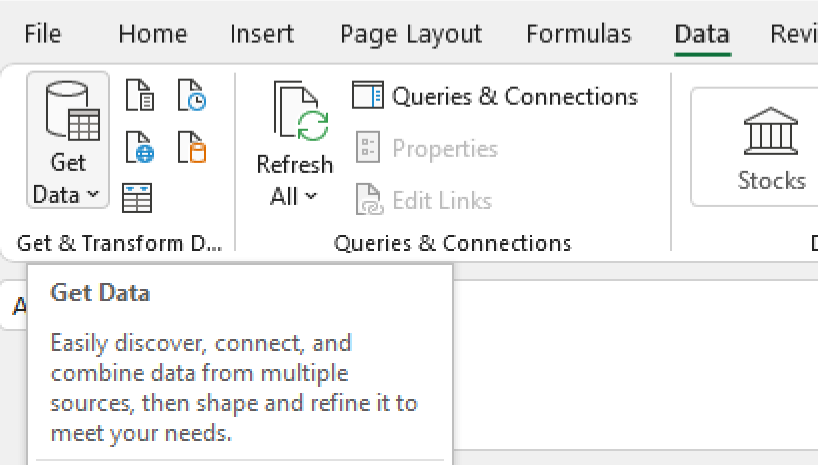
Data Models (con't)

- Example Access database:



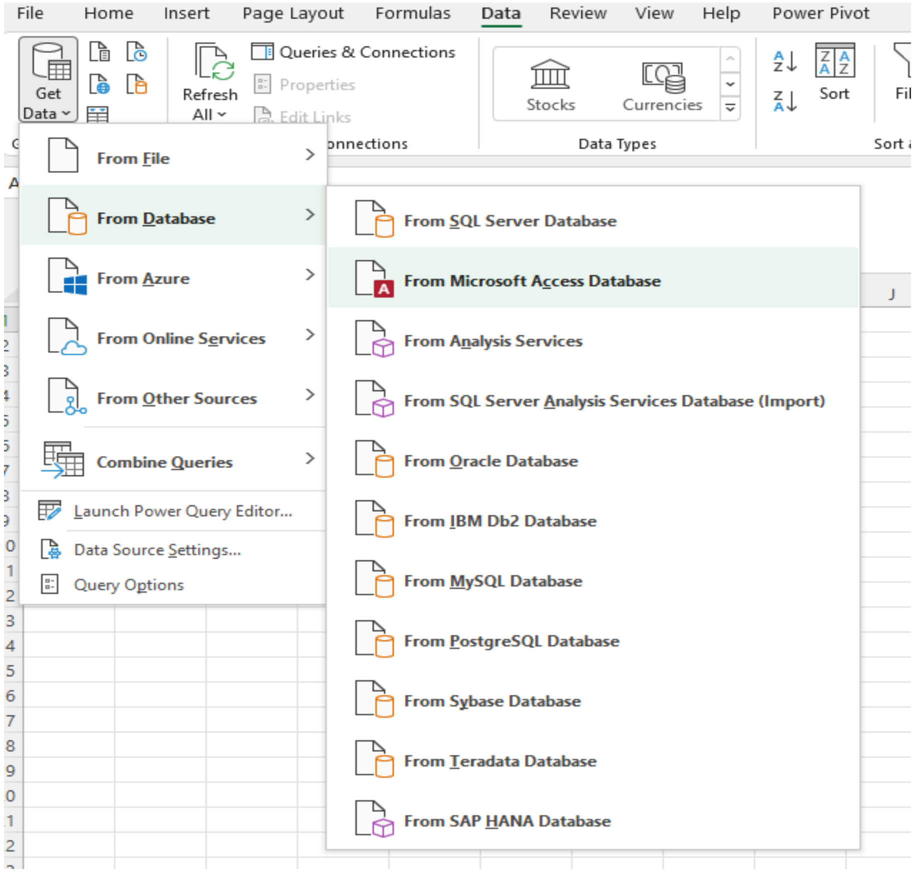
Data Models (con't)

- Open a new Excel workbook, then:
  - Data → Get & Transform → Get Data → From Database → from Microsoft Access Database



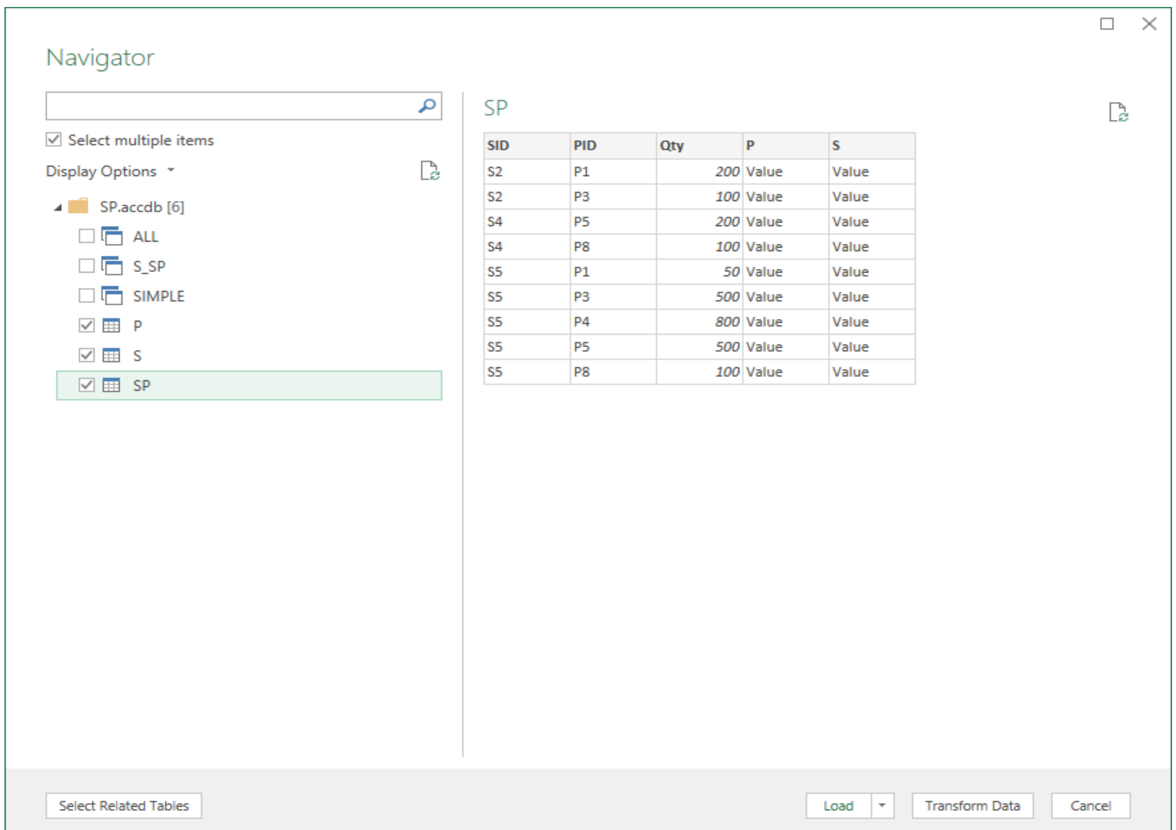
Data Models (con’t)

- From Microsoft Access Database then chose database file in Windows file selection window



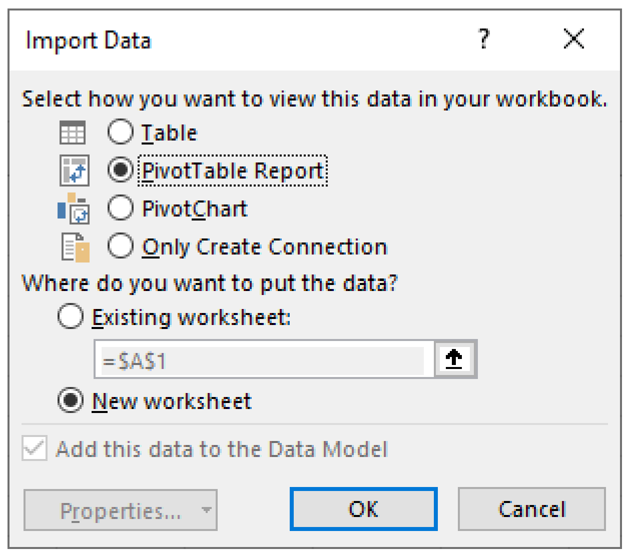
Data Models (con't)

- Navigator window opens – select multiple items – click load button



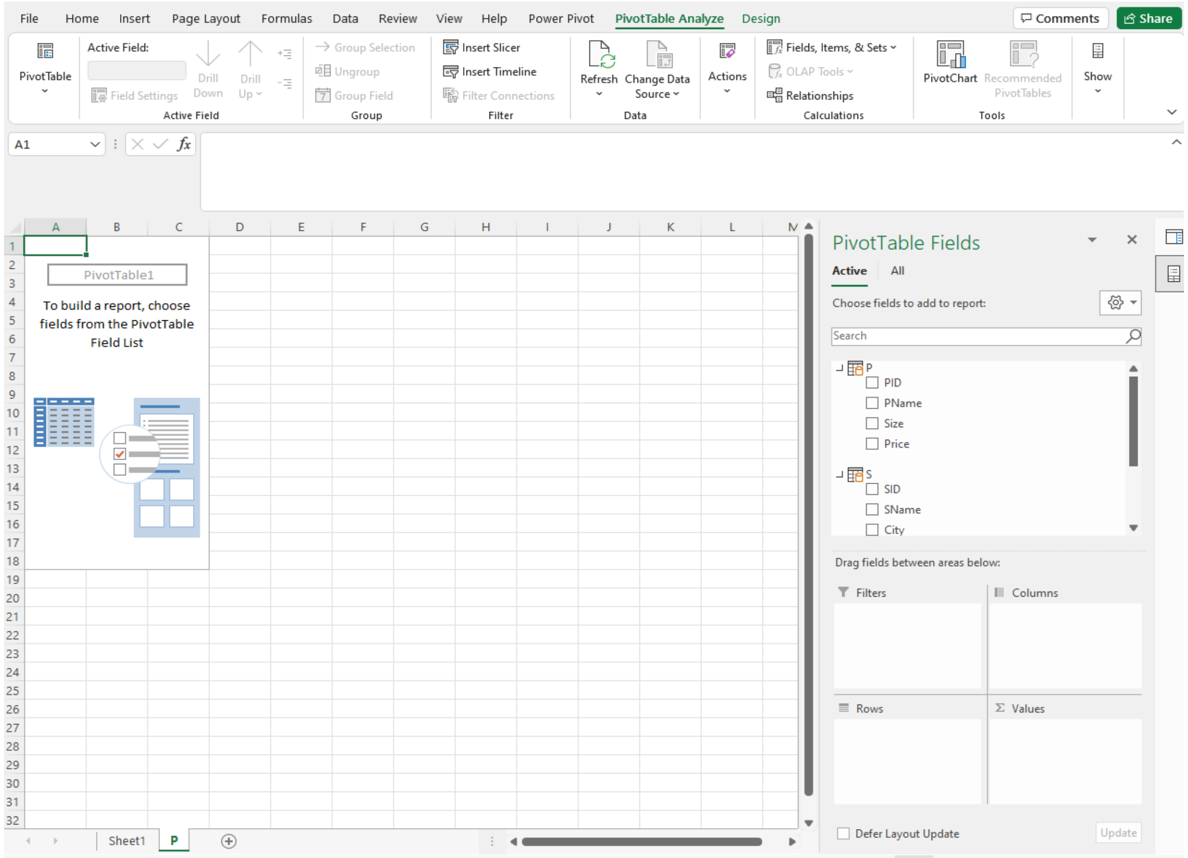
Data Models (con't)

- Click Load → Load To …
  - Click Pivot Table Report (Add this to the Data Model is automatically checked)
  - Click OK



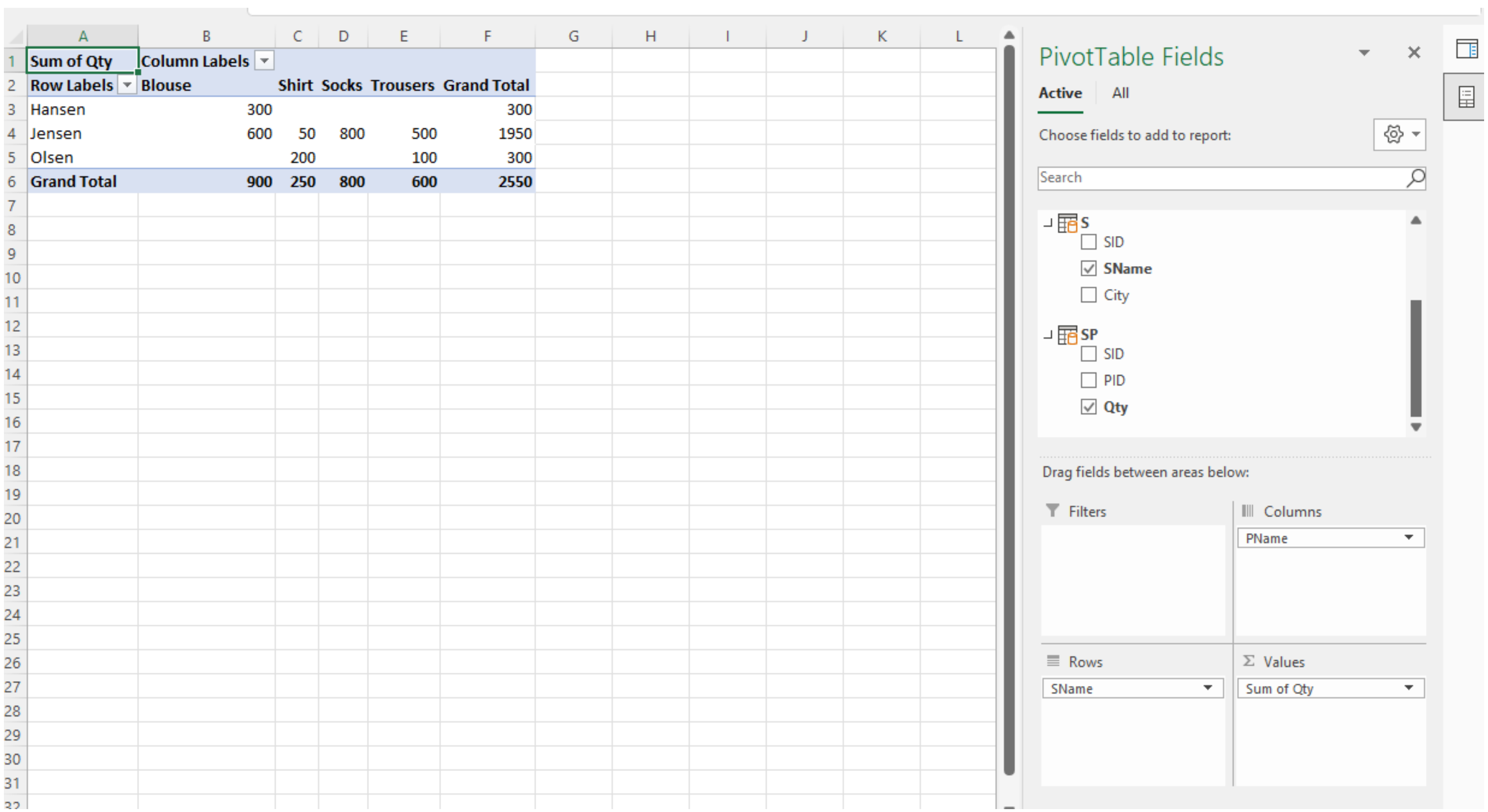
Data Models (con’t)

- An empty OLPA pivot table is created, and you can now design your pivot table



Data Models (con't)

- Qty by Salesperson name and Product name



Sum of Qty	Column Labels				
Row Labels	Blouse	Shirt	Socks	Trousers	Grand Total
Hansen	300				300
Jensen	600	50	800	500	1950
Olsen		200		100	300
Grand Total	**900**	**250**	**800**	**600**	**2550**

Data Models (con’t)

- One can see the imported Data Model:
  - Power Pivot → Data Model → Manage
  - The Power Pivot window opens
  - Home → View → Diagram View