Data warehouse
Data analysis

Elena Baralis

Politecnico di Torino

Data analysis

- OLAP analysis: complex aggregate function computation
  - support to different types of aggregate functions (e.g., moving average, top ten)
- Comparison operations, exploited to compare business trends (example: sale figure comparison for different time periods)
  - difficult by exploiting plain SQL
- Data analysis by means of data mining techniques

User interface

Users may query the data warehouse by means of various tools:

- controlled query environments
- query and report generation tools
- data mining tools

Controlled query environment

- It encompasses
  - complex queries with predefined structure (usually parametric)
  - ad hoc analysis procedures
  - predefined reports
- Techniques and knowledge of a specific economic area may be exploited
- It requires ad hoc code development
  - stored procedures, application packages, predefined joins and aggregations
  - flexible tools for report management are available, which allow defining
    - report layout
    - publication periodicity
    - distribution list

Ad hoc query environment

- Arbitrary OLAP queries may be defined
- Queries are designed on demand by users
  - query is defined by point and click techniques, which automatically generate SQL instructions
  - (typically) complex queries may be defined
  - spreadsheet is the user interface paradigm
- An OLAP session allows successive refinements of the same query
- Used when predefined reports are not enough

OLAP analysis

Elena Baralis

Politecnico di Torino

OLAP analysis

- Available query operations
  - roll up, drill down
  - slice and dice
  - (table) pivot
  - sorting
- Operations may be
  - used together in the same query
  - exploited in sequence to refine the same query which builds up the OLAP session

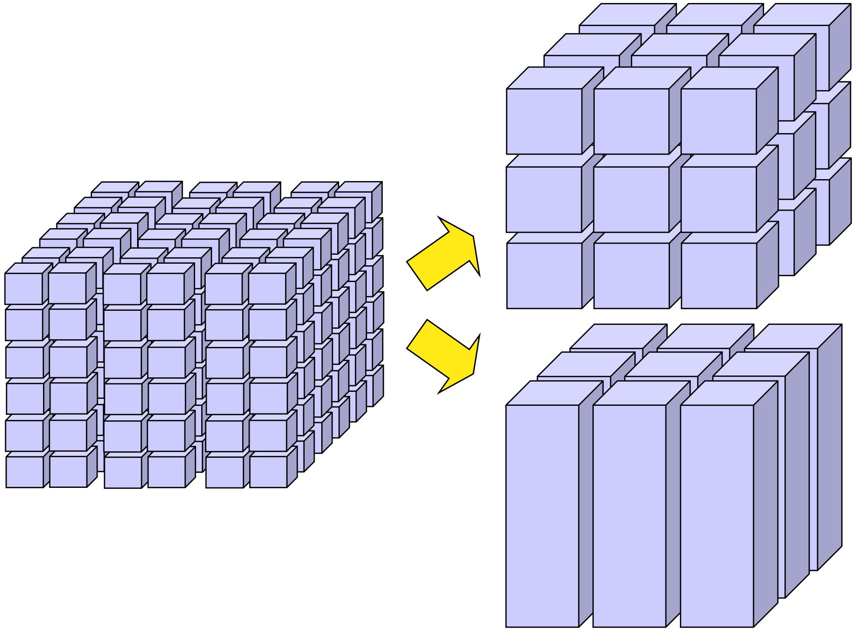
Roll up

- Data detail reduction by
  - decreasing detail in a dimension, by climbing up a hierarchy
    - example

      group by store, month → group by city, month
  - dropping a whole dimension
    - example

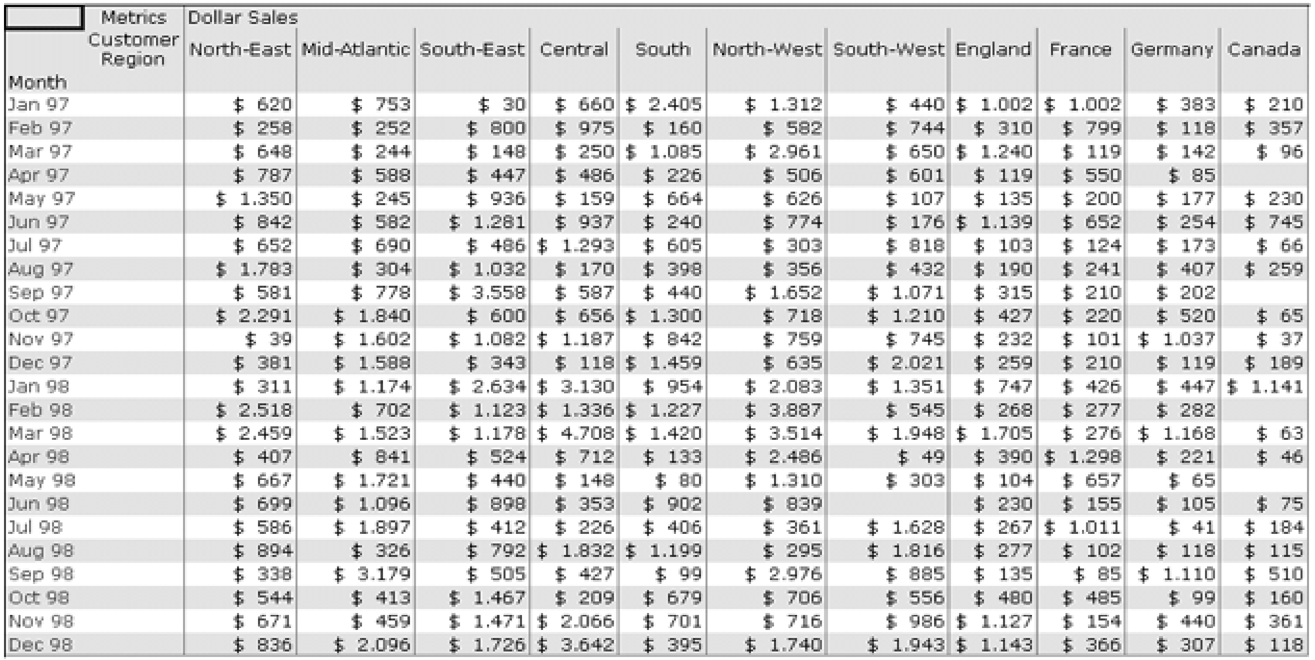
      group by product, city → group by product

Roll up

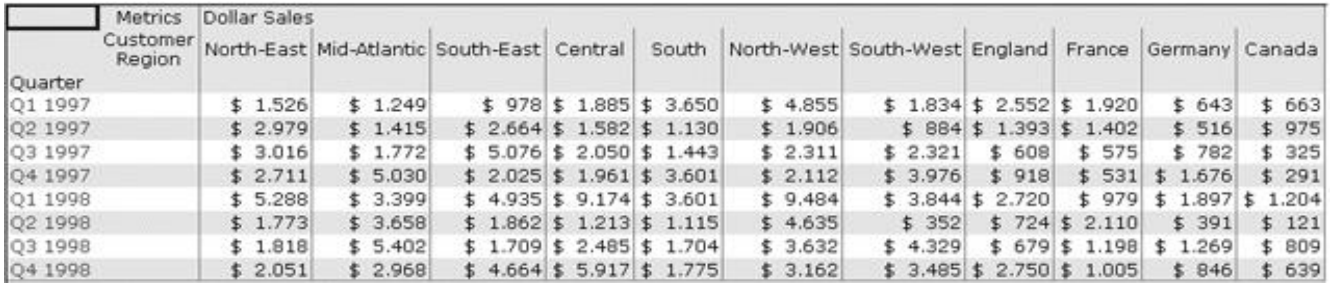


From Golfarelli, Rizzi,”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006

Roll up



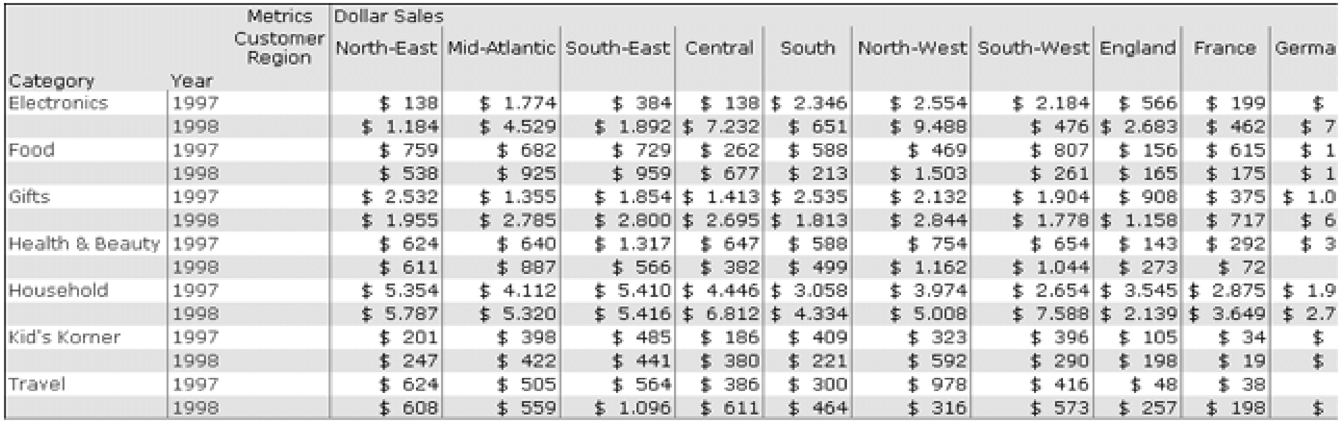
Metrics	Dollar Sales										
Customer Region	North-East	Mid-Atlantic	South-East	Central	South	North-West	South-West	England	France	Germany	Canada
Month											
Jan 97	$ 620	$ 753	$ 30	$ 660	$ 2.405	$ 1.312	$ 440	$ 1.002	$ 1.002	$ 383	$ 210
Feb 97	$ 258	$ 252	$ 800	$ 975	$ 160	$ 582	$ 744	$ 310	$ 799	$ 118	$ 357
Mar 97	$ 648	$ 244	$ 148	$ 250	$ 1.085	$ 2.961	$ 650	$ 1.240	$ 119	$ 142	$ 96
Apr 97	$ 787	$ 588	$ 447	$ 486	$ 226	$ 506	$ 601	$ 119	$ 550	$ 85	
May 97	$ 1.350	$ 245	$ 936	$ 159	$ 664	$ 626	$ 107	$ 135	$ 200	$ 177	$ 230
Jun 97	$ 842	$ 582	$ 1.281	$ 937	$ 240	$ 774	$ 176	$ 1.139	$ 652	$ 254	$ 745
Jul 97	$ 652	$ 690	$ 486	$ 1.293	$ 605	$ 303	$ 818	$ 103	$ 124	$ 173	$ 66
Aug 97	$ 1.783	$ 304	$ 1.032	$ 170	$ 398	$ 356	$ 432	$ 190	$ 241	$ 407	$ 259
Sep 97	$ 581	$ 778	$ 3.558	$ 587	$ 440	$ 1.652	$ 1.071	$ 315	$ 210	$ 202	
Oct 97	$ 2.291	$ 1.840	$ 600	$ 656	$ 1.300	$ 718	$ 1.210	$ 427	$ 220	$ 520	$ 65
Nov 97	$ 39	$ 1.602	$ 1.082	$ 1.187	$ 842	$ 759	$ 745	$ 232	$ 101	$ 1.037	$ 37
Dec 97	$ 381	$ 1.588	$ 343	$ 118	$ 1.459	$ 635	$ 2.021	$ 259	$ 210	$ 119	$ 189
Jan 98	$ 311	$ 1.174	$ 2.634	$ 3.130	$ 954	$ 2.083	$ 1.351	$ 747	$ 426	$ 447	$ 1.141
Feb 98	$ 2.518	$ 702	$ 1.123	$ 1.336	$ 1.227	$ 3.887	$ 545	$ 268	$ 277	$ 282	
Mar 98	$ 2.459	$ 1.523	$ 1.178	$ 4.708	$ 1.420	$ 3.514	$ 1.948	$ 1.705	$ 276	$ 1.168	$ 63
Apr 98	$ 407	$ 841	$ 524	$ 712	$ 133	$ 2.486	$ 49	$ 390	$ 1.298	$ 221	$ 46
May 98	$ 667	$ 1.721	$ 440	$ 148	$ 80	$ 1.310	$ 303	$ 104	$ 657	$ 65	
Jun 98	$ 699	$ 1.096	$ 898	$ 353	$ 902	$ 839		$ 230	$ 155	$ 105	$ 75
Jul 98	$ 586	$ 1.897	$ 412	$ 226	$ 406	$ 361	$ 1.628	$ 267	$ 1.011	$ 41	$ 184
Aug 98	$ 894	$ 326	$ 792	$ 1.832	$ 1.199	$ 295	$ 1.816	$ 277	$ 102	$ 118	$ 115
Sep 98	$ 338	$ 3.179	$ 505	$ 427	$ 99	$ 2.976	$ 885	$ 135	$ 85	$ 1.110	$ 510
Oct 98	$ 544	$ 413	$ 1.467	$ 209	$ 679	$ 706	$ 556	$ 480	$ 485	$ 99	$ 160
Nov 98	$ 671	$ 459	$ 1.471	$ 2.066	$ 701	$ 716	$ 986	$ 1.127	$ 154	$ 440	$ 361
Dec 98	$ 836	$ 2.096	$ 1.726	$ 3.642	$ 395	$ 1.740	$ 1.943	$ 1.143	$ 366	$ 307	$ 118



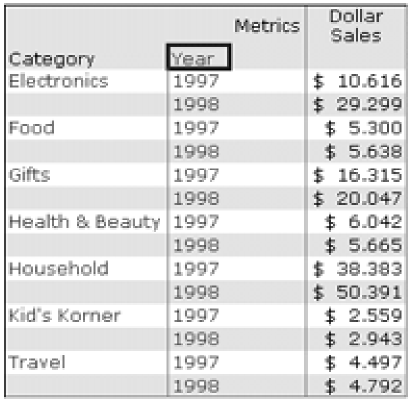
Metrics	Dollar Sales										
Customer Region	North-East	Mid-Atlantic	South-East	Central	South	North-West	South-West	England	France	Germany	Canada
Quarter											
Q1 1997	$ 1.526	$ 1.249	$ 978	$ 1.885	$ 3.650	$ 4.855	$ 1.834	$ 2.552	$ 1.920	$ 643	$ 663
Q2 1997	$ 2.979	$ 1.415	$ 2.664	$ 1.582	$ 1.130	$ 1.906	$ 884	$ 1.393	$ 1.402	$ 516	$ 975
Q3 1997	$ 3.016	$ 1.772	$ 5.076	$ 2.050	$ 1.443	$ 2.311	$ 2.321	$ 608	$ 575	$ 782	$ 325
Q4 1997	$ 2.711	$ 5.030	$ 2.025	$ 1.961	$ 3.601	$ 2.112	$ 3.976	$ 918	$ 531	$ 1.676	$ 291
Q1 1998	$ 5.288	$ 3.399	$ 4.935	$ 9.174	$ 3.601	$ 9.484	$ 3.844	$ 2.720	$ 979	$ 1.897	$ 1.204
Q2 1998	$ 1.773	$ 3.658	$ 1.862	$ 1.213	$ 1.115	$ 4.635	$ 352	$ 724	$ 2.110	$ 391	$ 121
Q3 1998	$ 1.818	$ 5.402	$ 1.709	$ 2.485	$ 1.704	$ 3.632	$ 4.329	$ 679	$ 1.198	$ 1.269	$ 809
Q4 1998	$ 2.051	$ 2.968	$ 4.664	$ 5.917	$ 1.775	$ 3.162	$ 3.485	$ 2.750	$ 1.005	$ 846	$ 639

From Golfarelli, Rizzi,”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006

Roll up



	Metrics	Dollar Sales									
	Customer Region	North-East	Mid-Atlantic	South-East	Central	South	North-West	South-West	England	France	Germa
Category	Year										
Electronics	1997	$ 138	$ 1.774	$ 384	$ 138	$ 2.346	$ 2.554	$ 2.184	$ 566	$ 199	$
	1998	$ 1.184	$ 4.529	$ 1.892	$ 7.232	$ 651	$ 9.488	$ 476	$ 2.683	$ 462	$ 7
Food	1997	$ 759	$ 682	$ 729	$ 262	$ 588	$ 469	$ 807	$ 156	$ 615	$ 1
	1998	$ 538	$ 925	$ 959	$ 677	$ 213	$ 1.503	$ 261	$ 165	$ 175	$ 1
Gifts	1997	$ 2.532	$ 1.355	$ 1.854	$ 1.413	$ 2.535	$ 2.132	$ 1.904	$ 908	$ 375	$ 1.0
	1998	$ 1.955	$ 2.785	$ 2.800	$ 2.695	$ 1.813	$ 2.844	$ 1.778	$ 1.158	$ 717	$ 6
Health & Beauty	1997	$ 624	$ 640	$ 1.317	$ 647	$ 588	$ 754	$ 654	$ 143	$ 292	$ 3
	1998	$ 611	$ 887	$ 566	$ 382	$ 499	$ 1.162	$ 1.044	$ 273	$ 72	
Household	1997	$ 5.354	$ 4.112	$ 5.410	$ 4.446	$ 3.058	$ 3.974	$ 2.654	$ 3.545	$ 2.875	$ 1.9
	1998	$ 5.787	$ 5.320	$ 5.416	$ 6.812	$ 4.334	$ 5.008	$ 7.588	$ 2.139	$ 3.649	$ 2.7
Kid's Korner	1997	$ 201	$ 398	$ 485	$ 186	$ 409	$ 323	$ 396	$ 105	$ 34	$
	1998	$ 247	$ 422	$ 441	$ 380	$ 221	$ 592	$ 290	$ 198	$ 19	$
Travel	1997	$ 624	$ 505	$ 564	$ 386	$ 300	$ 978	$ 416	$ 48	$ 38	
	1998	$ 608	$ 559	$ 1.096	$ 611	$ 464	$ 316	$ 573	$ 257	$ 198	$



	Metrics	Dollar Sales
Category	Year	
Electronics	1997	$ 10.616
	1998	$ 29.299
Food	1997	$ 5.300
	1998	$ 5.638
Gifts	1997	$ 16.315
	1998	$ 20.047
Health & Beauty	1997	$ 6.042
	1998	$ 5.665
Household	1997	$ 38.383
	1998	$ 50.391
Kid's Korner	1997	$ 2.559
	1998	$ 2.943
Travel	1997	$ 4.497
	1998	$ 4.792

From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

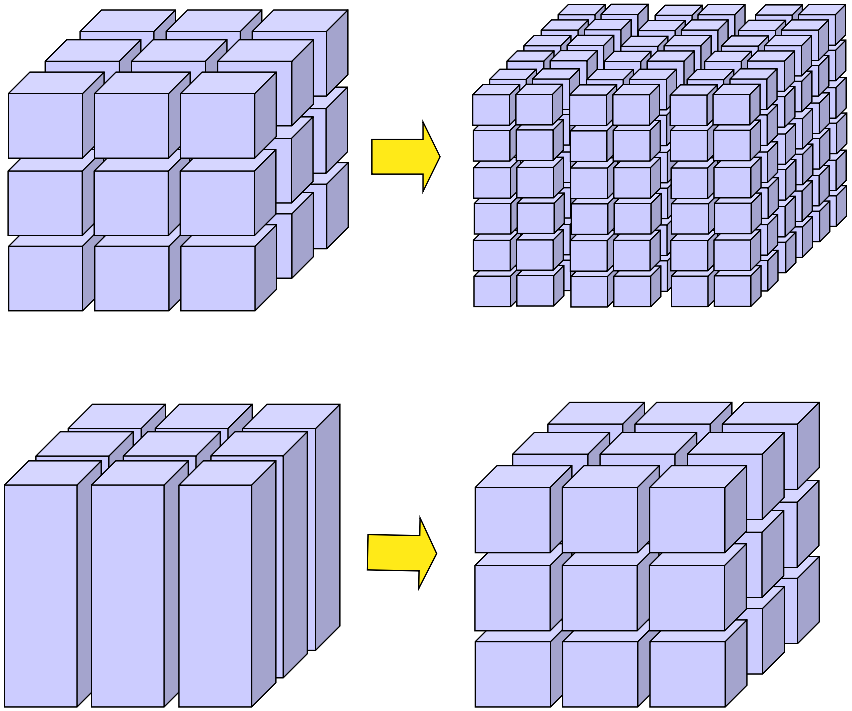
Drill down

- Data detail increase by
  - increasing detail in a dimension, by walking down a hierarchy
    - example

      group by city, month → group by store, month
  - adding a whole dimension
    - example

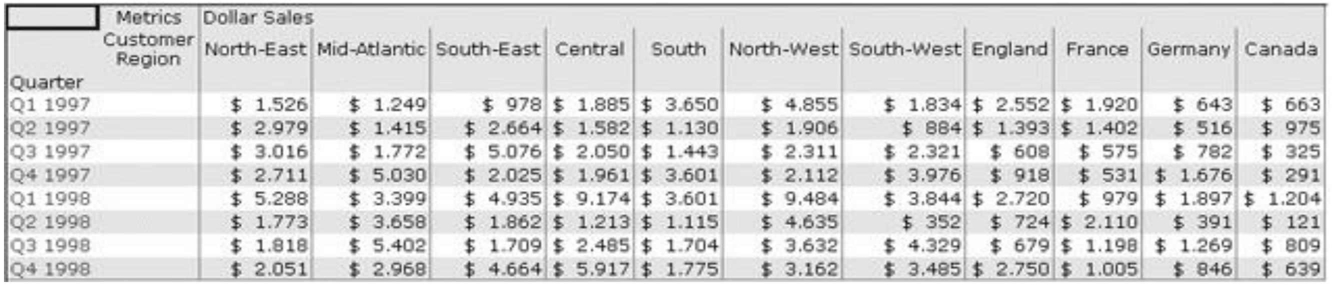
      group by product → group by product, city
- Frequently drill down operates on a subset of data produced by the initial query

Drill down

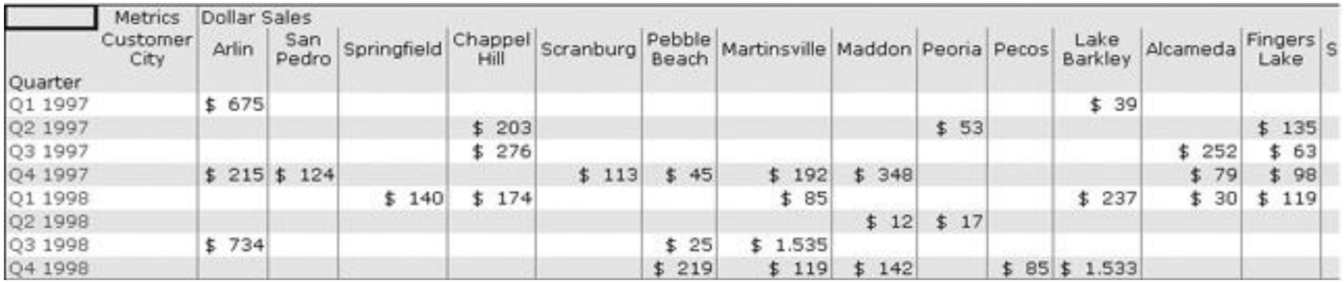


From Golfarelli, Rizzi,”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006

Drill down



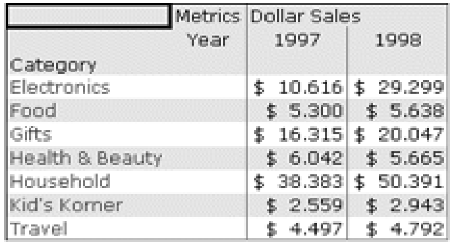
	Metrics	Dollar Sales										
	Customer Region	North-East	Mid-Atlantic	South-East	Central	South	North-West	South-West	England	France	Germany	Canada
Quarter												
Q1 1997		$ 1.526	$ 1.249	$ 978	$ 1.885	$ 3.650	$ 4.855	$ 1.834	$ 2.552	$ 1.920	$ 643	$ 663
Q2 1997		$ 2.979	$ 1.415	$ 2.664	$ 1.582	$ 1.130	$ 1.906	$ 884	$ 1.393	$ 1.402	$ 516	$ 975
Q3 1997		$ 3.016	$ 1.772	$ 5.076	$ 2.050	$ 1.443	$ 2.311	$ 2.321	$ 608	$ 575	$ 782	$ 325
Q4 1997		$ 2.711	$ 5.030	$ 2.025	$ 1.961	$ 3.601	$ 2.112	$ 3.976	$ 918	$ 531	$ 1.676	$ 291
Q1 1998		$ 5.288	$ 3.399	$ 4.935	$ 9.174	$ 3.601	$ 9.484	$ 3.844	$ 2.720	$ 979	$ 1.897	$ 1.204
Q2 1998		$ 1.773	$ 3.658	$ 1.862	$ 1.213	$ 1.115	$ 4.635	$ 352	$ 724	$ 2.110	$ 391	$ 121
Q3 1998		$ 1.818	$ 5.402	$ 1.709	$ 2.485	$ 1.704	$ 3.632	$ 4.329	$ 679	$ 1.198	$ 1.269	$ 809
Q4 1998		$ 2.051	$ 2.968	$ 4.664	$ 5.917	$ 1.775	$ 3.162	$ 3.485	$ 2.750	$ 1.005	$ 846	$ 639



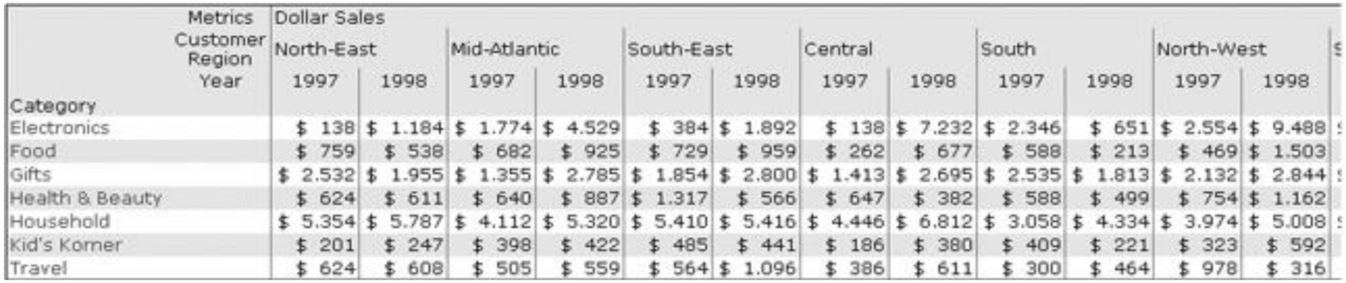
	Metrics	Dollar Sales													
	Customer City	Arlin	San Pedro	Springfield	Chappel Hill	Scranburg	Pebble Beach	Martinsville	Maddon	Peoria	Pecos	Lake Barkley	Alcameda	Fingers Lake	S
Quarter															
Q1 1997		$ 675										$ 39			
Q2 1997					$ 203					$ 53				$ 135	
Q3 1997					$ 276								$ 252	$ 63	
Q4 1997		$ 215	$ 124			$ 113	$ 45	$ 192	$ 348				$ 79	$ 98	
Q1 1998				$ 140	$ 174			$ 85				$ 237	$ 30	$ 119	
Q2 1998									$ 12	$ 17					
Q3 1998		$ 734					$ 25	$ 1.535							
Q4 1998							$ 219	$ 119	$ 142		$ 85	$ 1.533			

From Golfarelli, Rizzi,”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006

Drill down



Metrics	Dollar Sales	
Year	1997	1998
Category		
Electronics	$ 10.616	$ 29.299
Food	$ 5.300	$ 5.638
Gifts	$ 16.315	$ 20.047
Health & Beauty	$ 6.042	$ 5.665
Household	$ 38.383	$ 50.391
Kid's Korner	$ 2.559	$ 2.943
Travel	$ 4.497	$ 4.792



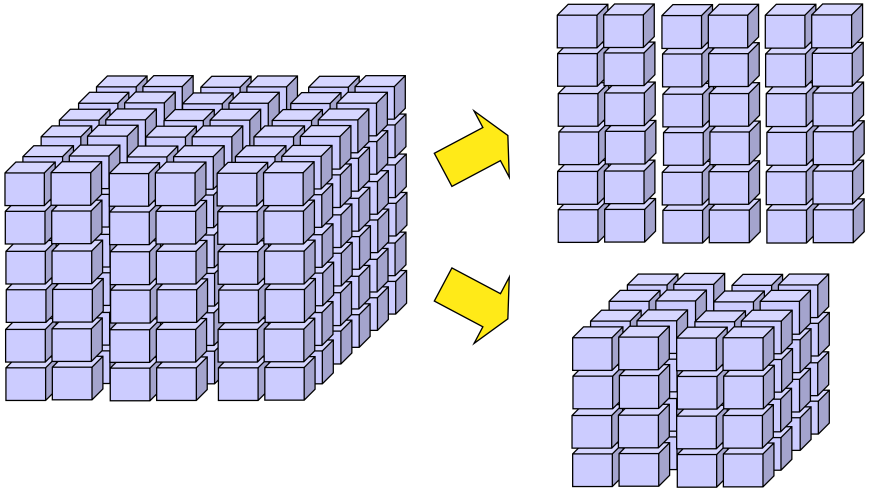
Metrics	Dollar Sales											
Customer Region	North-East		Mid-Atlantic		South-East		Central		South		North-West	
Year	1997	1998	1997	1998	1997	1998	1997	1998	1997	1998	1997	1998
Category												
Electronics	$ 138	$ 1.184	$ 1.774	$ 4.529	$ 384	$ 1.892	$ 138	$ 7.232	$ 2.346	$ 651	$ 2.554	$ 9.488
Food	$ 759	$ 538	$ 682	$ 925	$ 729	$ 959	$ 262	$ 677	$ 588	$ 213	$ 469	$ 1.503
Gifts	$ 2.532	$ 1.955	$ 1.355	$ 2.785	$ 1.854	$ 2.800	$ 1.413	$ 2.695	$ 2.535	$ 1.813	$ 2.132	$ 2.844
Health & Beauty	$ 624	$ 611	$ 640	$ 887	$ 1.317	$ 566	$ 647	$ 382	$ 588	$ 499	$ 754	$ 1.162
Household	$ 5.354	$ 5.787	$ 4.112	$ 5.320	$ 5.410	$ 5.416	$ 4.446	$ 6.812	$ 3.058	$ 4.334	$ 3.974	$ 5.008
Kid's Korner	$ 201	$ 247	$ 398	$ 422	$ 485	$ 441	$ 186	$ 380	$ 409	$ 221	$ 323	$ 592
Travel	$ 624	$ 608	$ 505	$ 559	$ 564	$ 1.096	$ 386	$ 611	$ 300	$ 464	$ 978	$ 316

From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Slice and dice

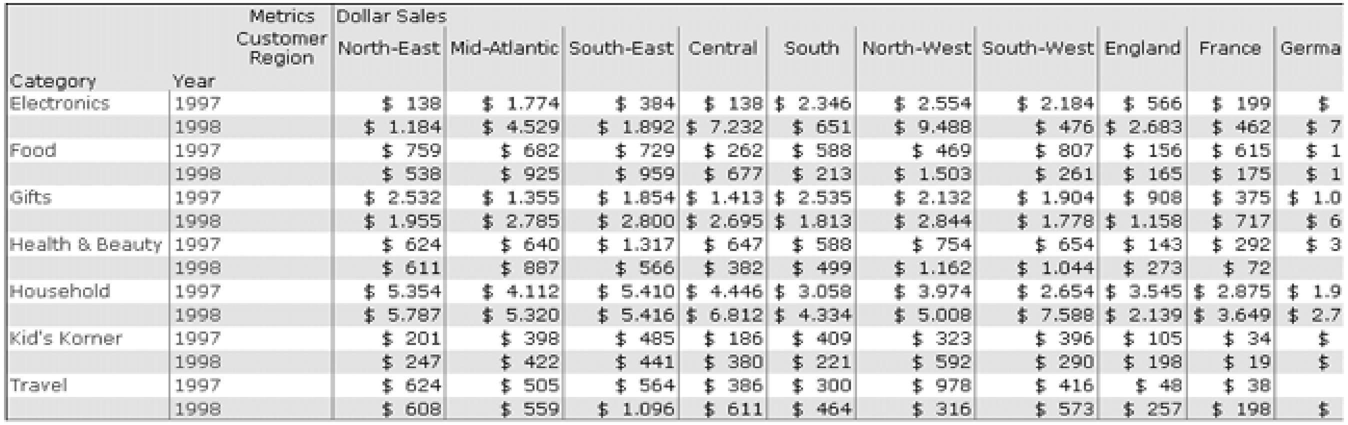
- Selection of a data subset by means of selection predicates
  - slice: equality predicate selecting a “slice”
    - example: Year=2005
  - dice: predicate expression selecting a “dice”
    - example: Category=‘Food’ and City=‘Torino’

Slice and dice

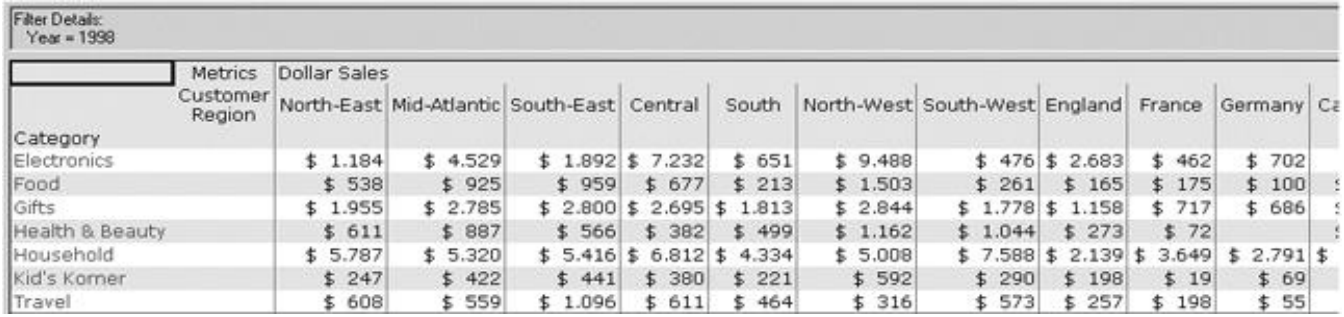


From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Slice and dice



	Metrics	Dollar Sales									
	Customer Region	North-East	Mid-Atlantic	South-East	Central	South	North-West	South-West	England	France	Germa
Category	Year										
Electronics	1997	$ 138	$ 1.774	$ 384	$ 138	$ 2.346	$ 2.554	$ 2.184	$ 566	$ 199	$
	1998	$ 1.184	$ 4.529	$ 1.892	$ 7.232	$ 651	$ 9.488	$ 476	$ 2.683	$ 462	$ 7
Food	1997	$ 759	$ 682	$ 729	$ 262	$ 588	$ 469	$ 807	$ 156	$ 615	$ 1
	1998	$ 538	$ 925	$ 959	$ 677	$ 213	$ 1.503	$ 261	$ 165	$ 175	$ 1
Gifts	1997	$ 2.532	$ 1.355	$ 1.854	$ 1.413	$ 2.535	$ 2.132	$ 1.904	$ 908	$ 375	$ 1.0
	1998	$ 1.955	$ 2.785	$ 2.800	$ 2.695	$ 1.813	$ 2.844	$ 1.778	$ 1.158	$ 717	$ 6
Health & Beauty	1997	$ 624	$ 640	$ 1.317	$ 647	$ 588	$ 754	$ 654	$ 143	$ 292	$ 3
	1998	$ 611	$ 887	$ 566	$ 382	$ 499	$ 1.162	$ 1.044	$ 273	$ 72	
Household	1997	$ 5.354	$ 4.112	$ 5.410	$ 4.446	$ 3.058	$ 3.974	$ 2.654	$ 3.545	$ 2.875	$ 1.9
	1998	$ 5.787	$ 5.320	$ 5.416	$ 6.812	$ 4.334	$ 5.008	$ 7.588	$ 2.139	$ 3.649	$ 2.7
Kid's Korner	1997	$ 201	$ 398	$ 485	$ 186	$ 409	$ 323	$ 396	$ 105	$ 34	$
	1998	$ 247	$ 422	$ 441	$ 380	$ 221	$ 592	$ 290	$ 198	$ 19	$
Travel	1997	$ 624	$ 505	$ 564	$ 386	$ 300	$ 978	$ 416	$ 48	$ 38	
	1998	$ 608	$ 559	$ 1.096	$ 611	$ 464	$ 316	$ 573	$ 257	$ 198	$

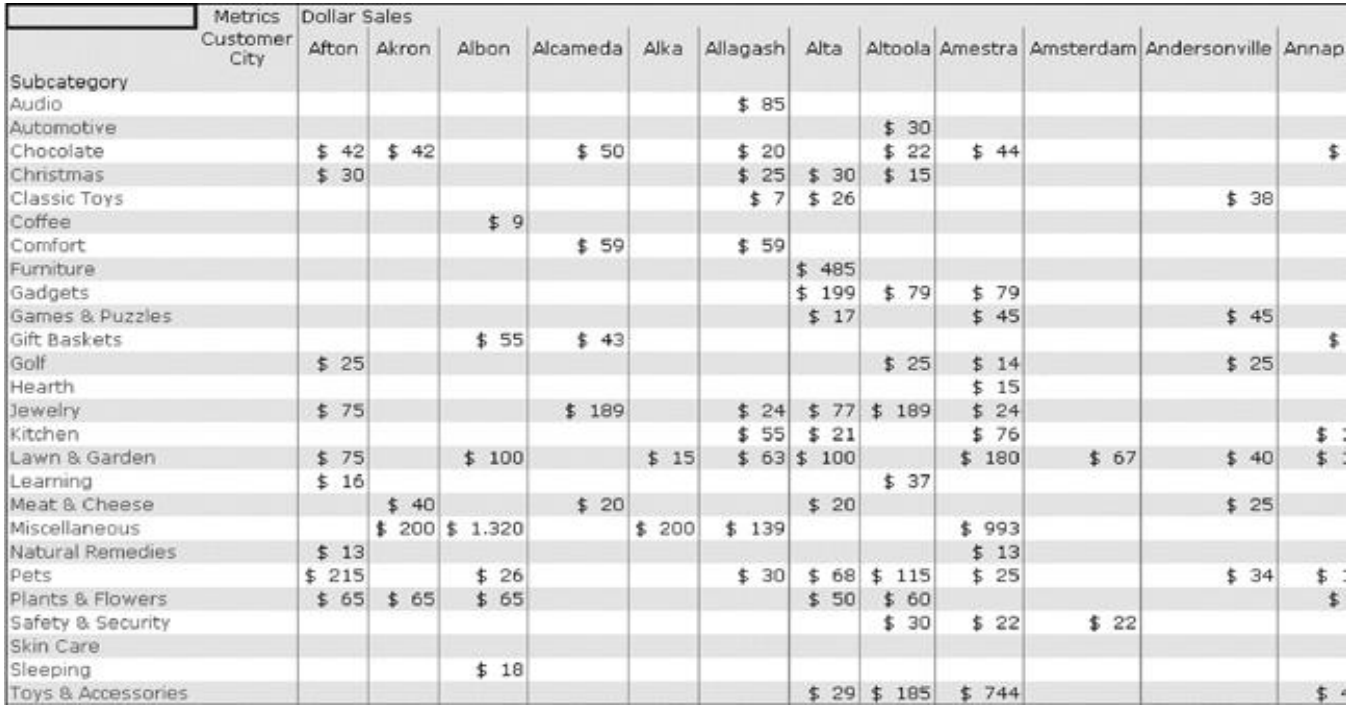


Filter Details:
Year = 1998

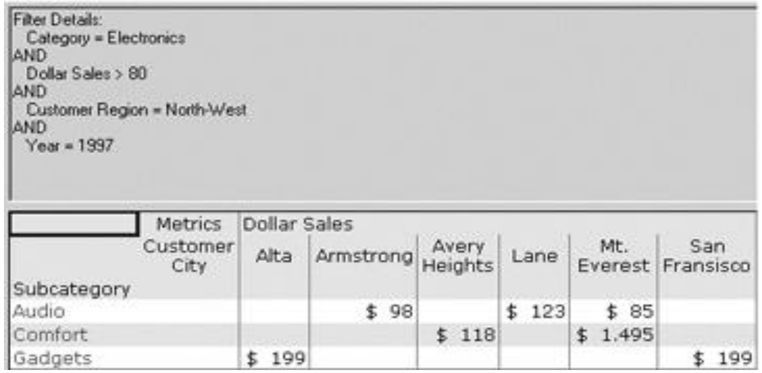
Metrics	Dollar Sales										
Customer Region	North-East	Mid-Atlantic	South-East	Central	South	North-West	South-West	England	France	Germany	Ca
Category											
Electronics	$ 1.184	$ 4.529	$ 1.892	$ 7.232	$ 651	$ 9.488	$ 476	$ 2.683	$ 462	$ 702	
Food	$ 538	$ 925	$ 959	$ 677	$ 213	$ 1.503	$ 261	$ 165	$ 175	$ 100	$
Gifts	$ 1.955	$ 2.785	$ 2.800	$ 2.695	$ 1.813	$ 2.844	$ 1.778	$ 1.158	$ 717	$ 686	$
Health & Beauty	$ 611	$ 887	$ 566	$ 382	$ 499	$ 1.162	$ 1.044	$ 273	$ 72		$
Household	$ 5.787	$ 5.320	$ 5.416	$ 6.812	$ 4.334	$ 5.008	$ 7.588	$ 2.139	$ 3.649	$ 2.791	$
Kid's Korner	$ 247	$ 422	$ 441	$ 380	$ 221	$ 592	$ 290	$ 198	$ 19	$ 69	
Travel	$ 608	$ 559	$ 1.096	$ 611	$ 464	$ 316	$ 573	$ 257	$ 198	$ 55	

From Golfarelli, Rizzi,”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006

Slice and dice



Metrics	Dollar Sales											
Customer City	Afton	Akron	Albon	Alcameda	Alka	Allagash	Alta	Altoola	Amestra	Amsterdam	Andersonville	Annap
Subcategory												
Audio						$ 85						
Automotive								$ 30				
Chocolate	$ 42	$ 42		$ 50		$ 20		$ 22	$ 44			$
Christmas	$ 30					$ 25	$ 30	$ 15				
Classic Toys						$ 7	$ 26				$ 38	
Coffee			$ 9									
Comfort				$ 59		$ 59						
Furniture							$ 485					
Gadgets							$ 199	$ 79	$ 79			
Games & Puzzles							$ 17		$ 45		$ 45	
Gift Baskets			$ 55	$ 43								$
Golf	$ 25							$ 25	$ 14		$ 25	
Hearth									$ 15			
Jewelry	$ 75			$ 189		$ 24	$ 77	$ 189	$ 24			
Kitchen						$ 55	$ 21		$ 76			$
Lawn & Garden	$ 75		$ 100		$ 15	$ 63	$ 100		$ 180	$ 67	$ 40	$
Learning	$ 16							$ 37				
Meat & Cheese		$ 40		$ 20			$ 20				$ 25	
Miscellaneous		$ 200	$ 1.320		$ 200	$ 139			$ 993			
Natural Remedies	$ 13								$ 13			
Pets	$ 215		$ 26			$ 30	$ 68	$ 115	$ 25		$ 34	$
Plants & Flowers	$ 65	$ 65	$ 65				$ 50	$ 60				$
Safety & Security								$ 30	$ 22	$ 22		
Skin Care												
Sleeping			$ 18									
Toys & Accessories							$ 29	$ 185	$ 744			$



Filter Details:
Category = Electronics
AND
Dollar Sales > 80
AND
Customer Region = North-West
AND
Year = 1997

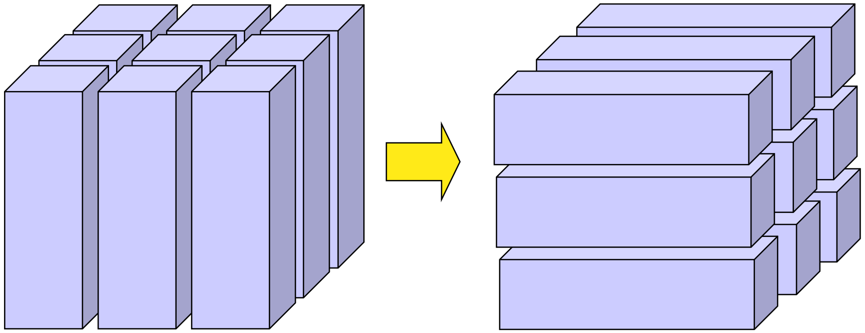
Metrics	Dollar Sales					
Customer City	Alta	Armstrong	Avery Heights	Lane	Mt. Everest	San Fransisco
Subcategory						
Audio		$ 98		$ 123	$ 85	
Comfort			$ 118		$ 1.495	
Gadgets	$ 199					$ 199

From Golfarelli, Rizzi,"Data warehouse, teoria e pratica della progettazione", McGraw Hill 2006

Pivot

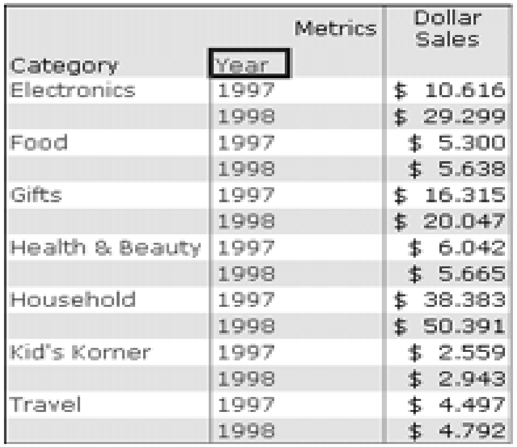
- Reorganization of the multidimensional structure without varying the detail level
  - increases readability of the same information
  - multidimensional representation is always based on a “grid” (hierarchical spreadsheet)
    - two dimensions are the main grid axes
    - position of dimensions in the grid are changed

Pivot

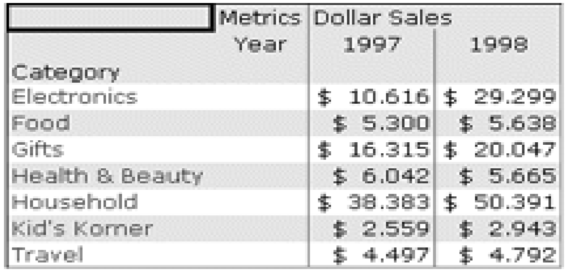


From Golfarelli, Rizzi,”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006

Pivot



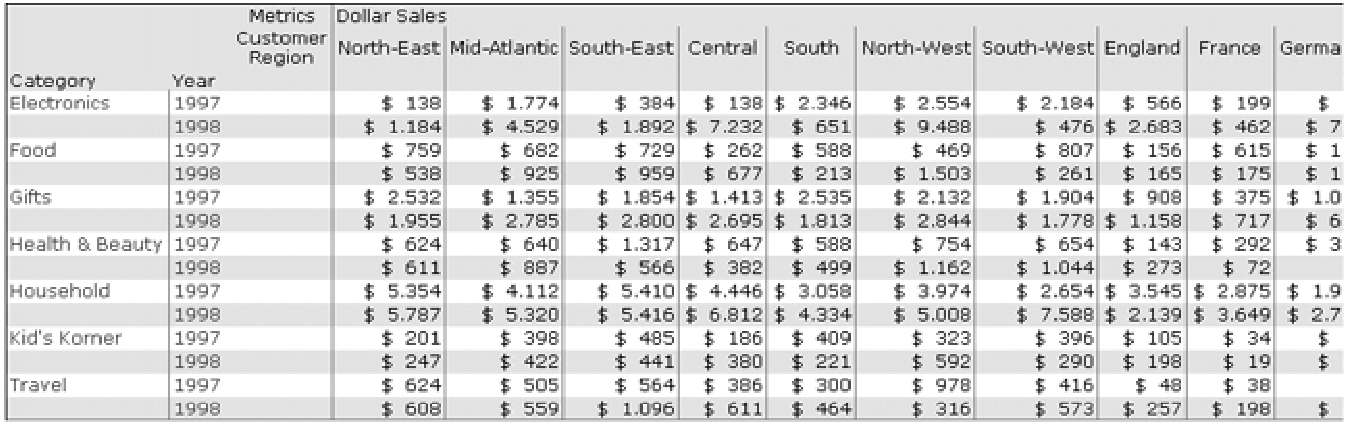
	Metrics	Dollar Sales
Category	Year	
Electronics	1997	$ 10.616
	1998	$ 29.299
Food	1997	$ 5.300
	1998	$ 5.638
Gifts	1997	$ 16.315
	1998	$ 20.047
Health & Beauty	1997	$ 6.042
	1998	$ 5.665
Household	1997	$ 38.383
	1998	$ 50.391
Kid's Korner	1997	$ 2.559
	1998	$ 2.943
Travel	1997	$ 4.497
	1998	$ 4.792



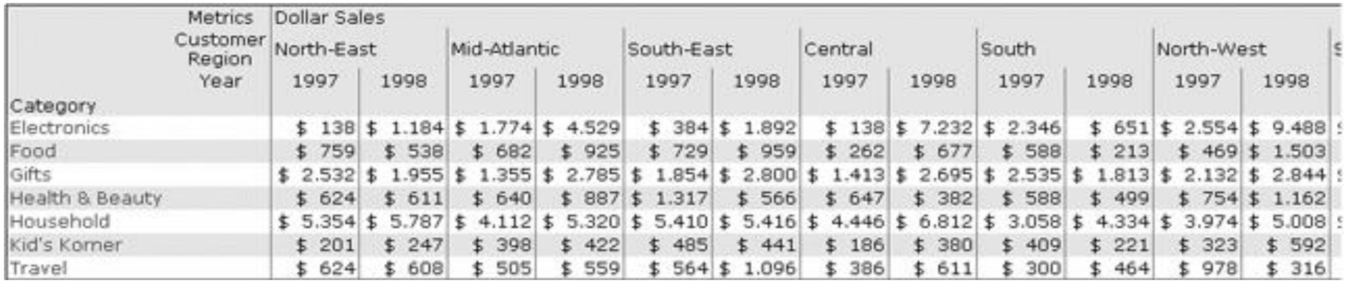
	Metrics	Dollar Sales	
	Year	1997	1998
Category			
Electronics		$ 10.616	$ 29.299
Food		$ 5.300	$ 5.638
Gifts		$ 16.315	$ 20.047
Health & Beauty		$ 6.042	$ 5.665
Household		$ 38.383	$ 50.391
Kid's Korner		$ 2.559	$ 2.943
Travel		$ 4.497	$ 4.792

From Golfarelli, Rizzi,”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006

Pivot



	Metrics	Dollar Sales									
	Customer Region	North-East	Mid-Atlantic	South-East	Central	South	North-West	South-West	England	France	Germa
Category	Year										
Electronics	1997	$ 138	$ 1.774	$ 384	$ 138	$ 2.346	$ 2.554	$ 2.184	$ 566	$ 199	$
	1998	$ 1.184	$ 4.529	$ 1.892	$ 7.232	$ 651	$ 9.488	$ 476	$ 2.683	$ 462	$ 7
Food	1997	$ 759	$ 682	$ 729	$ 262	$ 588	$ 469	$ 807	$ 156	$ 615	$ 1
	1998	$ 538	$ 925	$ 959	$ 677	$ 213	$ 1.503	$ 261	$ 165	$ 175	$ 1
Gifts	1997	$ 2.532	$ 1.355	$ 1.854	$ 1.413	$ 2.535	$ 2.132	$ 1.904	$ 908	$ 375	$ 1.0
	1998	$ 1.955	$ 2.785	$ 2.800	$ 2.695	$ 1.813	$ 2.844	$ 1.778	$ 1.158	$ 717	$ 6
Health & Beauty	1997	$ 624	$ 640	$ 1.317	$ 647	$ 588	$ 754	$ 654	$ 143	$ 292	$ 3
	1998	$ 611	$ 887	$ 566	$ 382	$ 499	$ 1.162	$ 1.044	$ 273	$ 72	
Household	1997	$ 5.354	$ 4.112	$ 5.410	$ 4.446	$ 3.058	$ 3.974	$ 2.654	$ 3.545	$ 2.875	$ 1.9
	1998	$ 5.787	$ 5.320	$ 5.416	$ 6.812	$ 4.334	$ 5.008	$ 7.588	$ 2.139	$ 3.649	$ 2.7
Kid's Korner	1997	$ 201	$ 398	$ 485	$ 186	$ 409	$ 323	$ 396	$ 105	$ 34	$
	1998	$ 247	$ 422	$ 441	$ 380	$ 221	$ 592	$ 290	$ 198	$ 19	$
Travel	1997	$ 624	$ 505	$ 564	$ 386	$ 300	$ 978	$ 416	$ 48	$ 38	
	1998	$ 608	$ 559	$ 1.096	$ 611	$ 464	$ 316	$ 573	$ 257	$ 198	$



	Metrics	Dollar Sales												
	Customer Region	North-East		Mid-Atlantic		South-East		Central		South		North-West		S
	Year	1997	1998	1997	1998	1997	1998	1997	1998	1997	1998	1997	1998	
Category														
Electronics		$ 138	$ 1.184	$ 1.774	$ 4.529	$ 384	$ 1.892	$ 138	$ 7.232	$ 2.346	$ 651	$ 2.554	$ 9.488	$
Food		$ 759	$ 538	$ 682	$ 925	$ 729	$ 959	$ 262	$ 677	$ 588	$ 213	$ 469	$ 1.503	
Gifts		$ 2.532	$ 1.955	$ 1.355	$ 2.785	$ 1.854	$ 2.800	$ 1.413	$ 2.695	$ 2.535	$ 1.813	$ 2.132	$ 2.844	$
Health & Beauty		$ 624	$ 611	$ 640	$ 887	$ 1.317	$ 566	$ 647	$ 382	$ 588	$ 499	$ 754	$ 1.162	
Household		$ 5.354	$ 5.787	$ 4.112	$ 5.320	$ 5.410	$ 5.416	$ 4.446	$ 6.812	$ 3.058	$ 4.334	$ 3.974	$ 5.008	$
Kid's Korner		$ 201	$ 247	$ 398	$ 422	$ 485	$ 441	$ 186	$ 380	$ 409	$ 221	$ 323	$ 592	
Travel		$ 624	$ 608	$ 505	$ 559	$ 564	$ 1.096	$ 386	$ 611	$ 300	$ 464	$ 978	$ 316	

From Golfarelli, Rizzi,”Data warehouse, teoria e pratica della progettazione”, McGraw Hill 2006

Extensions of the SQL language

Elena Baralis

Politecnico di Torino

Extensions of the SQL language

- Interface tools require
  - new aggregate functions
    - aggregate functions exploited for economic analysis (moving average, median, ...)
    - position in the sort order (i.e., rank)
  - functions for report generation
    - partial and cumulative totals
- New OLAP functions in the ANSI standard
  - implemented starting from DB2 UDB 7.1, Oracle 8i v2

Extensions of the SQL language

- Interface tools require
  - operators for the computation of different group bys at the same time
- The SQL-99 (SQL3) standard has extended the SQL group by clause

Example data base

Sales(City,Month,Amount)

City	Month	Amount
Milano	7	110
Milano	8	10
Milano	9	70
Milano	10	90
Milano	11	35
Milano	12	135
Torino	7	70
Torino	8	35
Torino	9	80
Torino	10	95
Torino	11	50
Torino	12	120

SQL OLAP functions

- New class of aggregate functions (OLAP functions) characterized by
  - – computation window, inside which the computation of aggregate functions is performed
    - cumulative totals and moving average can be computed
  - – new aggregate functions to compute the rank in a given sort order

Computation window

- New **`window`** clause, characterized by
  - *partitioning*: Rows are grouped without collapsing them (different from **`group by`**)
    - no partitioning: a single group is defined
  - *row ordering*, separately in each partition (similar to **`order by`**)
  - *aggregation window*: For each row in the partition, it defines the row group on which the aggregate function is computed

Example

- Show, for each city and month
  - sale amount
  - average on the current month and the two previous months, separately for each city

Example

- Partitioning on city
  - average computation is reset when the city changes
- Ordering by month, to compute the moving average on the current month and the two preceding months
  - without ordering the computation is meaningless
- Aggregation window size: the current row and the two preceding rows

Example

```
SELECT City, Month, Amount,
       AVG(Amount) OVER Wavg AS MovingAvg
FROM Sales
WINDOW Wavg AS (PARTITION BY City
                ORDER BY Month
                ROWS 2 PRECEDING)
```

Example

```
SELECT City, Month, Amount,
       AVG(Amount) OVER (PARTITION BY City
                          ORDER BY Month
                          ROWS 2 PRECEDING)
       AS MovingAvg
FROM Sales
```

Result

City	Month	Amount	MovingAvg
Milano	7	110	110
Milano	8	10	60
Milano	9	90	70
Milano	10	80	60
Milano	11	40	60
Milano	12	140	90
Torino	7	70	70
Torino	8	30	50
Torino	9	80	60
Torino	10	100	70
Torino	11	50	60
Torino	12	150	100

Partition 1

Partition 2

Observations

- Sort order is required, because the computation of the moving average considers rows in an ordered fashion
  - the window sort order does not enforce a predefined output sort order
- When the window is not complete, the computation takes place on the available rows
  - it is possible to require a **`NULL`** result for each incomplete window
- Several different computation windows may be specified

Aggregation window

- The moving window on which the aggregate function is computed may be defined
  - at the *physical level*: It builds the group by counting rows
    - example: the current row and the two preceding rows
  - at the *logical level*: It builds the group by defining an interval on the sort key
    - example: the current month and the two preceding months

Physical interval definition

- Between a lower bound and the current row

```
ROWS 2 PRECEDING
```

- Between lower and upper bounds

```
ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING
ROWS BETWEEN 3 PRECEDING AND 1 PRECEDING
```

- Between the beginning (or the end) of a partition and the current row

```
ROWS UNBOUNDED PRECEDING (o FOLLOWING)
```

Physical interval

- Appropriate for sequence data with no gaps
  - example: no month is missing in the sequence
  - more than a sort key can be specified
    - computation ignores breaks due to change in any sort key value
    - example: order by month and year
  - no mathematical expressions are needed to compute the window

Logical interval definition

- The **`range`** clause is used, with the same syntax as the physical interval
- A distance on the sort key between the interval bounds and the current value should be defined
- Example

```
RANGE 2 MONTH PRECEDING
```

Logical interval

- Appropriate for “sparse” data, with gaps in the sequence
  - example: a month is missing in the sequence
  - only a single sort key can be specified
  - the sort key can only be alphanumeric or date type (arithmetic expressions are allowed)

Applications

- Moving aggregate computations
  - computations on a window which moves over data
  - examples: moving average, moving sum
- Cumulative total computations
  - the (cumulative) total is incremented by adding an instance at a time
- Comparison between detailed data and aggregated data

Computation of a cumulative total

- Show, for each city and month
  - sale amount
  - cumulative sale amount for increasing months, separately for each city

Computation of a cumulative total

- Partition by city
  - the cumulative total is reset when the city changes
- Order by (ascending) month to compute the sum for increasing months
  - without sorting, the computation would be meaningless
- Size of the aggregation window
  - from the starting row of the partition to the current row

Computation of a cumulative total

```
SELECT City, Month, Amount,
       SUM(Amount) OVER (PARTITION BY City
                              ORDER BY Month
                              ROWS UNBOUNDED PRECEDING)
       AS CumeTot
FROM Sales
```

Computation of a cumulative total

City	Month	Amount	CumeTot
Milano	7	110	110
Milano	8	10	120
Milano	9	90	210
Milano	10	80	290
Milano	11	40	330
Milano	12	140	470
Torino	7	70	70
Torino	8	30	100
Torino	9	80	180
Torino	10	100	280
Torino	11	50	330
Torino	12	150	480

Partition 1

Partition 2

Comparison between detailed data and total data

- Show, for each city and month
  - sale amount
  - total sale amount on the whole time period for the current city

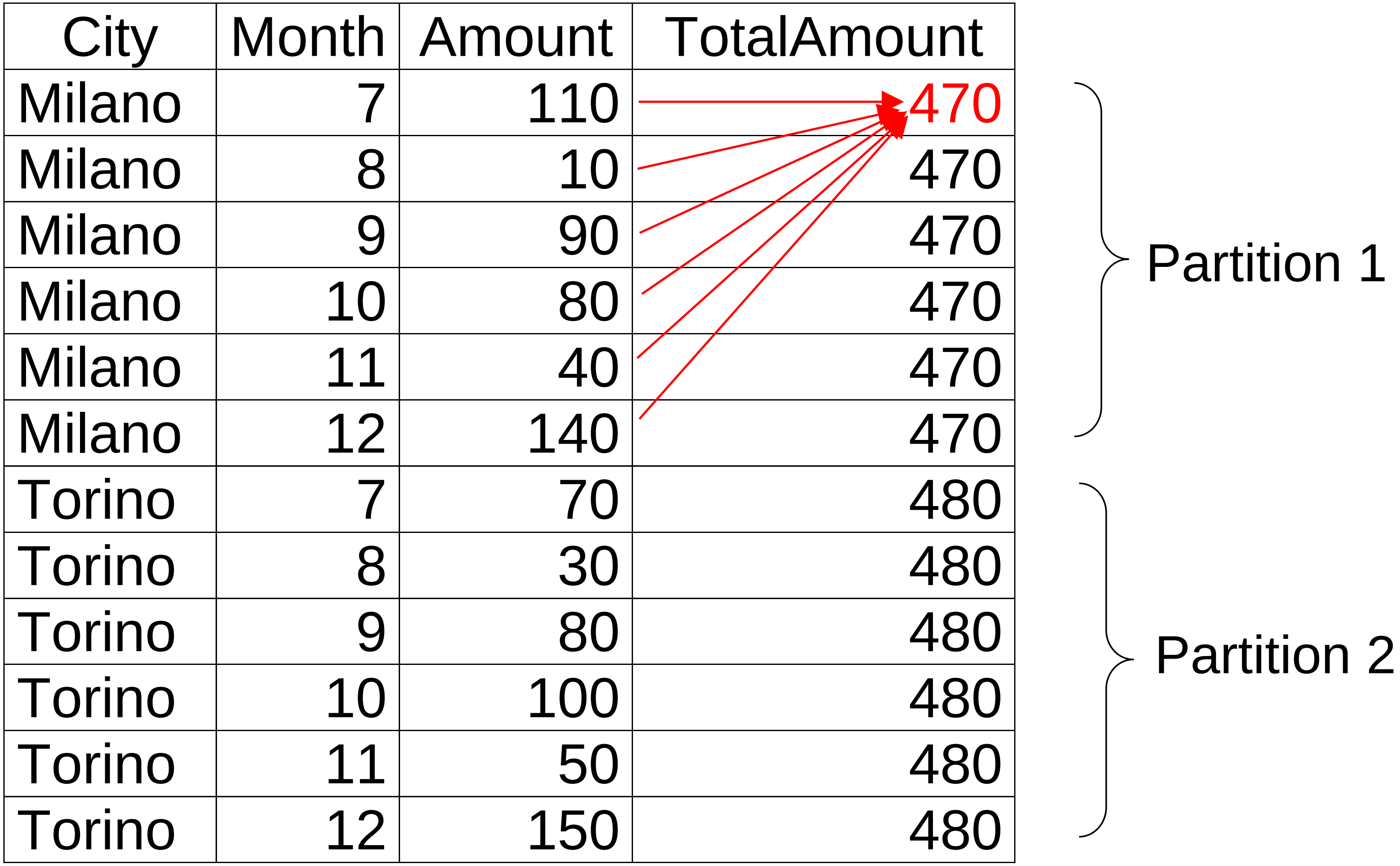
Comparison between detailed data and total data

- Partition by city
  - the total amount is reset when the city changes
- Sorting is not needed
  - the total amount is computed independently of the sort order of tuples
- The aggregation window is not needed
  - it is the whole partition

Comparison between detailed data and total data

```
SELECT City, Month, Amount,
       SUM(Amount) OVER (PARTITION BY City)
       AS TotalAmount
FROM Sales
```

Comparison between detailed data and total data



City	Month	Amount	TotalAmount
Milano	7	110	470
Milano	8	10	470
Milano	9	90	470
Milano	10	80	470
Milano	11	40	470
Milano	12	140	470
Torino	7	70	480
Torino	8	30	480
Torino	9	80	480
Torino	10	100	480
Torino	11	50	480
Torino	12	150	480

Partition 1

Partition 2

Comparison between detailed data and total data

- Show, for each city and month
  - sale amount
  - ratio between current row amount and grand total
  - ratio between current row amount and total amount by city
  - ratio between current row amount and total amount by month

Comparison between detailed data and total data

- Three different computation windows
  - grand total: no partitioning
  - total by city: partition by city
  - total by month: partition by month
- No sort is needed in any window
  - totals are independent of the sort order of tuples
- The aggregation window is always the whole partition

Comparison between detailed data and total data

```
SELECT City, Month, Amount
  Amount/SUM(Amount) OVER ()
  AS TotalFract
  Amount/SUM(Amount) OVER (PARTITION BY City)
  AS CityFract
  Amount/SUM(Amount) OVER (PARTITION BY Month)
  AS MonthFract
FROM Sales
```

Comparison between detailed data and total data

City	Month	Amount	TotalFract	CityFract	MonthFrct
Milano	7	110	110/950	110/470	110/180
Milano	8	10	10/950	10/470	10/40
Milano	9	90	90/950	90/470	90/170
Milano	10	80	80/950	80/470	80/180
Milano	11	40	40/950	40/470	40/90
Milano	12	140	140/950	140/470	140/290
Torino	7	70	70/950	70/480	70/180
Torino	8	30	30/950	30/480	30/40
Torino	9	80	80/950	80/480	80/170
Torino	10	100	100/950	100/480	100/180
Torino	11	50	50/950	50/480	50/90
Torino	12	150	150/950	150/480	150/290

`Group by` and `window`

- Windows can be used together with grouping performed by **`group by`**
- The “temporary table” generated by the execution of the **`group by`** clause (possibly with aggregate function computation) becomes the operand to which the computations in the **`window`** clause are applied

Example

- Assume that the **`Sales`** table contains information on sales with daily granularity
- Show, for each city and month
  - – sale amount
  - – average sale with respect to the current month and the two preceding months, separately for each city

Example

- Grouping by month is needed to compute the total amount by month before computing the moving average
  - the group by clause is used for computing the monthly total
- The temporary table generated by the group by computation is the operand on which the computation window is defined

Example

```
SELECT City, Month, SUM(Amount) AS TotMonth,
       AVG(SUM(Amount)) OVER (PARTITION BY City
                               ORDER BY Month
                               ROWS 2 PRECEDING)
       AS MovingAvg
FROM Sales
WHERE <join conditions>
GROUP BY City, Month
```

Ranking functions

- Functions computing the rank of a value inside a partition
  - **`rank()`** function: computes the rank by leaving an empty slot after a tie
    - example: after 2 first, the next rank is third
  - **`denserank()`** function: computes the rank by leaving an empty slot after a tie
    - example: after 2 first, the next rank is second

Example

- Show, for each city in december
  - sale amount
  - rank on amount

Example

- Partitioning is not needed
  - a single partition including all cities
- Order by amount to perform ranking
  - without sorting, the computation would be meaningless
- The aggregation window is the whole partition

Example

```
SELECT City, Amount,
       RANK() OVER (ORDER BY Amount DESC)
       AS Ranking
FROM Sales
WHERE Month = 12
```

Result

City	Amount	Ranking
Torino	150	1
Milano	140	2

Sorting the result

- A sorted result is obtained by means of the **`order by`** clause
  - may be different from the sort order in the computation window
- Example: sort the result in the former example by increasing city

Example

```
SELECT City, Amount,
       RANK() OVER (ORDER BY Amount DESC)
       AS Ranking
FROM Sales
WHERE Month = 12
ORDER BY City
```

City	Amount	Ranking
Milano	140	2
Torino	150	1

`group by` clause extensions

- Multidimensional spreadsheets compute several partial totals “in one shot”
  - total sale amount by month and city
  - total sale amount by month
  - total sale amount by city
- For the sake of efficiency avoid
  - multiple data reads
  - redundant data sorts

`group by` clause extensions

- SQL-99 standard extended the syntax of the **`group by`** clause
  - **`rollup`** computes aggregations on all groups obtained by removing one by one the columns in the grouping clause
  - **`cube`** computes aggregations on all combinations of the columns in the grouping clause
  - **`grouping sets`** computes aggregations on the group list in the grouping clause (grouping sets different from the previous clauses may be specified)
    - **`()`** for grand totals (no grouping)

`Rollup`: example

- Consider the following tables

  `Time(`<u>`Tkey`</u>`,Day,Month,Year,…)`

  `Shop(`<u>`Skey`</u>`,City,Region,…)`

  `Product(`<u>`Pkey`</u>`,PName,Brand,…)`

  `Sales(`<u>`Skey`</u>`,`<u>`Tkey`</u>`,`<u>`Pkey`</u>`,Amount)`

- Compute total sales in the year 2000 for the following attribute combinations
  - product, month, city
  - month, city
  - city

Rollup: example

```
SELECT City, Month, Pkey,
       SUM(Amount) AS TotSales
FROM Time T, Shop S, Sales V
WHERE T.Tkey = V.Tkey
  AND S.Skey = V.Skey
  AND Year = 2000
GROUP BY ROLLUP (City,Month,Pkey)
```

- The column sort order in **`rollup`** determines which aggregates are computed

Rollup: result

City	Month	Pkey	TotSales
Milano	7	145	110
Milano	7	150	10
Milano	…	…	…
Milano	7	**NULL**	8500
Milano	8	…	…
Milano	**NULL**	**NULL**	150000
Torino	…	…	150
Torino	…	**NULL**	2500
Torino	**NULL**	**NULL**	135000
…	…	…	…
NULL	**NULL**	**NULL**	25005000

- "Superaggregates" are represented by **NULL**

Cube: example

- Compute total sales in the year 2000 for *all* combinations of the following attributes
  - product, month, city
- The following aggregations should be computed
  - product, month, city
  - product, month
  - month, city
  - product, city
  - product
  - month
  - city
  - no grouping

Cube: example

```
SELECT City, Month, Pkey,
       SUM(Amount) AS TotSales
FROM Time T, Shop S, Sales V
WHERE T.Tkey = V.Tkey
  AND S.Skey = V.Skey
  AND Year = 2000
GROUP BY CUBE (City,Month,Pkey)
```

- The sort order of columns in **cube** is irrelevant

Cube computation

- Consider distributive and algebraic properties of aggregate functions
  - *distributive* aggregate functions (**`min`**, **`max`**, **`sum`**, **`count`**) may be computed from aggregations on a larger set of attributes (i.e., with larger granularity)
    - Example: from total sales by product and month, total sales by month may be computed
  - algebraic aggregate functions (**`avg`**, …) may be computed from aggregations on a larger set of attributes (i.e., with larger granularity), if appropriate support aggregations are stored
    - Example: average requires
      - the average value in the group
      - the cardinality of the group

Cube computation

- To increase the efficiency of cube computation, the distributive/algebraic properties of the aggregate functions are exploited
  - previously computed **`group by`** are exploited
  - **`rollup`** requires a single sort operation
  - the cube is a combination of several **`rollup`** operations (in the appropriate order)
  - previously executed sort operations are exploited (also partially)
    - it is possible to exploit sort on (A,B) to sort by (A,C)

Grouping Set: example

- Compute total sales in the year 2000 for the following groups
  - month
  - month, city, product
- A roll up would perform the computation of unnecessary groupings and aggregations

Grouping Set: example

```
SELECT City, Month, Pkey,
       SUM(Amount) AS TotSales
FROM Time T, Shop S, Sales S
WHERE T.Tkey = S.Tkey
  AND S.Skey = S.Skey
  AND Year = 2000
GROUP BY GROUPING SETS
      (Month, (City,Month,Pkey))
```