Data 604 – Data Management

Week 8 – Data Warehouse and OLAP

Introduction

- Operational versus tactical/strategic decision-making
- Data warehouse definition
- Data warehouse schemas
- The extraction, transformation, and loading (ETL) process
- Data marts
- Virtual data warehouses and virtual data marts
- Operational data store
- Data warehouses versus data lakes
- Business intelligence

Descriptive Analytics

- The exploratory phase of Analytics and Business Intelligence

- explores, aggregates and visualizes data

- Management Information Reports

- Dashboards

- Graphs and charts

- Business Scorecards

- Data warehousing and OLAP are ways to optimize data for BI and Reporting

Traditional Data Warehousing

- Uses BI, relational databases and OLAP features of traditional relational database management systems

- Use ETL methods to transform relational data into a data warehouse

- Star schema and snowflake database design methods to create Dimension and Fact tables

- Data marts -subject oriented data warehouses

- Scale up with computing and data nodes on high end servers (how does this differ from big data?)

Decision-Making Levels

- Operational level
  - Day-to-day business decisions
  - Short time frame
  - On-Line Transaction Processing (OLTP)

- Tactical level
  - Middle management decisions
  - Medium-term focus

- Strategic level
  - Senior management decisions
  - Long-term focus

© Cambridge University Press 2018

Decision-Making Levels

- Operational systems
  - Operational level
  - Focus on simple INSERT, UPDATE, DELETE, and/or SELECT statements
  - Transaction throughput

- Decision support systems
  - Tactical + strategic level
  - Focus on data retrieval by answering complex ad-hoc queries (SELECT statements)
  - Represent data in a multidimensional way
  - Trend analysis

© Cambridge University Press 2018

Data Warehouse Definition

- *"A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process."* (Inmon, 1996)

- Subject oriented
  - Organized around subjects such as customers, products, sales

- Integrated
  - Integrates data from a variety of operational sources and a variety of formats

© Cambridge University Press 2018

Data Warehouse Definition

- Non-volatile
  - data are primarily read-only
  - data loading and data retrieval

- Time variant
 - time series of periodic snapshots

Data Warehouse Definition

	Transactional System	Data Warehouse
Usage	Day-to-day business operations	Decision support at the tactical/strategic level
Data latency	Real-time data	Periodic snapshots, incl. historical data
Design	Application oriented	Subject oriented
Normalization	Normalized data	(Sometimes also) denormalized data
Data manipulation	INSERT/UPDATE/DELETE/SELECT	INSERT/SELECT
Transaction management	Important	Less of a concern
Type of queries	Many simple queries	Fewer, but complex and ad-hoc queries

© Cambridge University Press 2018

Data Warehouse Schemas

- Star schema

- Snowflake schema

- Fact constellation

- Specific schema issues

© Cambridge University Press 2018

Star Schema

© Cambridge University Press 2018
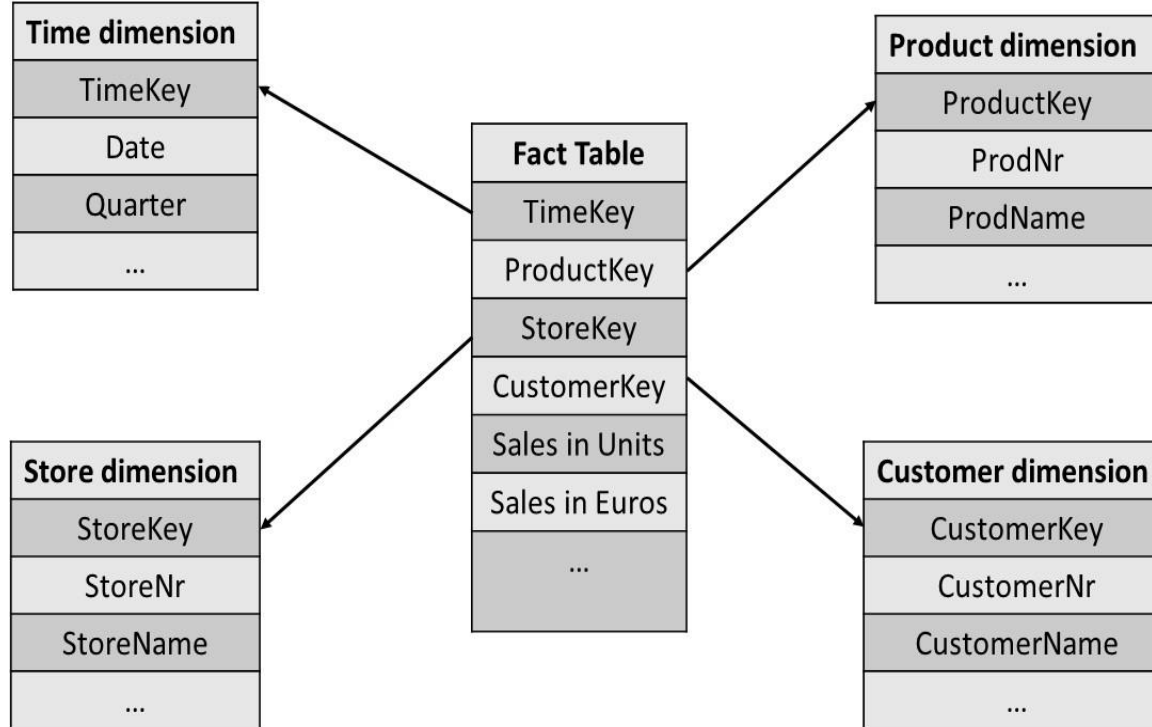
Star Schema

- Fact table
  - one tuple per transaction or event (i.e., a fact) and also measurement data (e.g., sales)

- Dimension table
  - further information about each of the facts in the fact table (e.g., Time, Store, Customer, Product)
  - criteria for aggregating the measurement data
  - often denormalized

Star Schema

Fact table

TimeKey	ProductKey	StoreKey	CustomerKey	Sales in Units	Sales in Euros
200	50	100	20006010	6	167.94
210	25	130	20006012	3	54
180	30	150	20006008	12	384
...					

Dimension tables

TimeKey	Date	Quarter	...
200	08/03/2017	1	...
210	09/11/2017	3	
180	27/02/2017	1	

CustomerKey	CustomerNr	CustName	...
20006008	20	Bart Baesens	...
20006010	10	Wilfried Lemahieu	
20006012	5	Seppe vanden Broucke	

StoreKey	StoreNr	StoreName	...
100	68	The Wine Depot	...
130	94	The Wine Crate	
150	69	Vinos del Mundo	

ProductKey	ProdNr	ProdName	...
25	0178	Meerdael, Methode Traditionnelle Chardonnay, 2014	...
30	0199	Jacques Selosse, Brut Initial, 2012	
50	0212	Billecart-Salmon, Brut Réserve, 2014	

© Cambridge University Press 2018
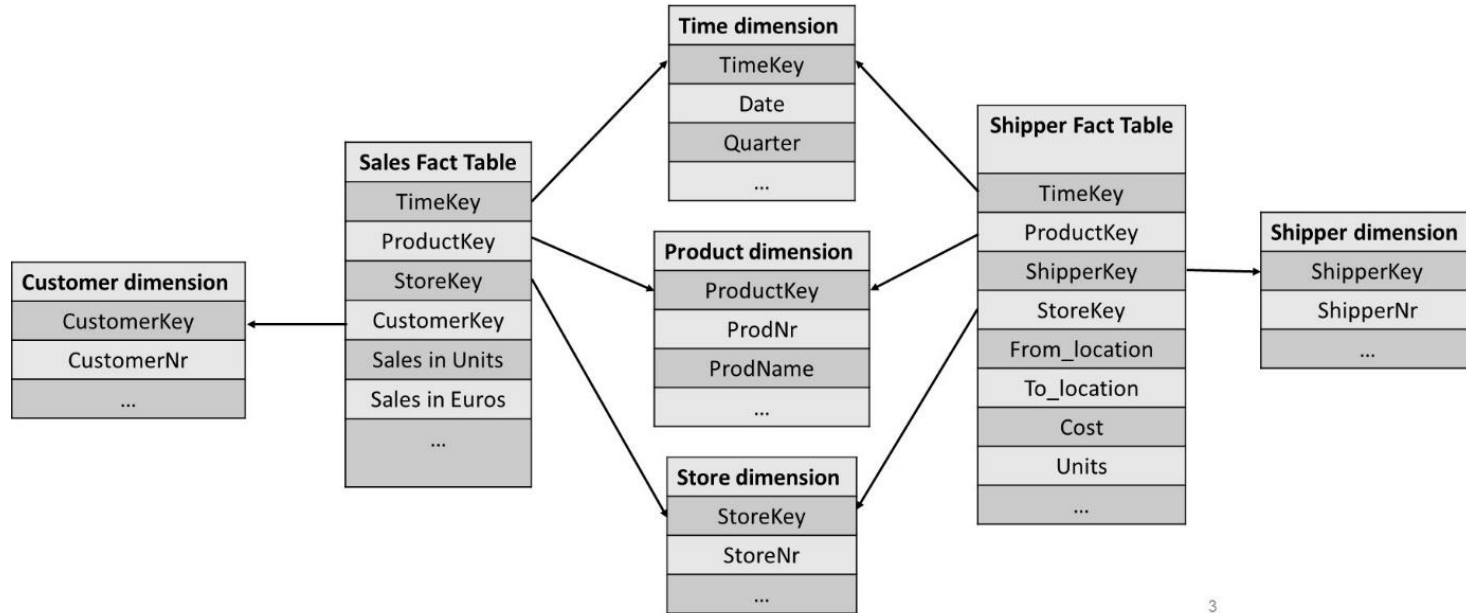
Snowflake Schema

© Cambridge University Press 2018
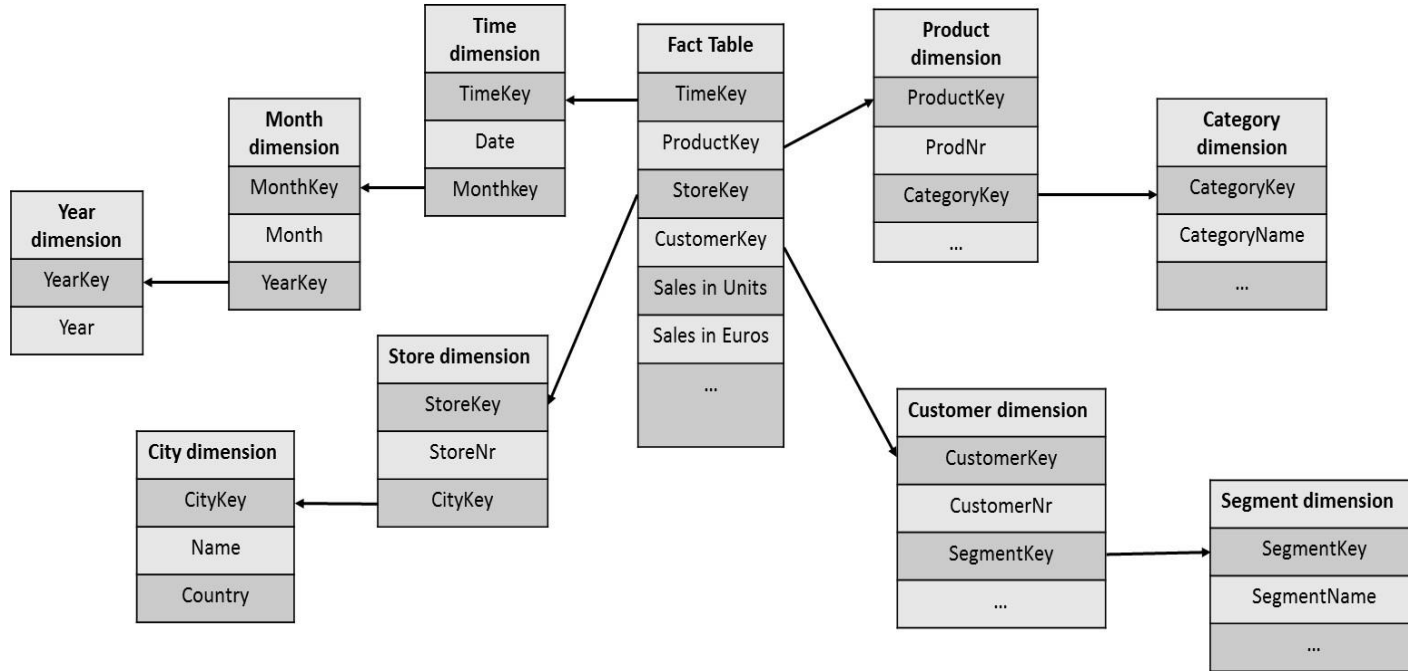
Snowflake Schema

- In case dimension tables grow too large

- In case most queries don't make use of the outer-level dimension tables

© Cambridge University Press 2018

Fact Constellation

Specific Schema Issues

- Surrogate keys
- Granularity of the fact table
- Factless fact tables
- Optimizing the dimension tables
- Defining junk dimensions
- Defining outrigger tables
- Slowly changing dimensions
- Rapidly changing dimensions

Surrogate Keys

- StoreKey, ProductKey, ShipperKey, …

- Meaningless integers

- Cannot use business keys since they usually have a business meaning

- Surrogate keys essentially buffer the data warehouse from the operational environment

- Business keys are usually larger in size

- Business keys are also often re-used over longer periods of time

18

Granularity of the Fact Table

- Level of detail of one row of the fact table

- Higher (lower) granularity implies more (fewer) rows

- Tradeoff between level of detailed analysis and storage requirements

- Examples:
  - One tuple of the fact table corresponds to one line on a purchase order
  - One tuple of the fact table corresponds to one purchase order
  - One tuple of the fact table corresponds to all purchase orders made by a customer

© Cambridge University Press 2018

Factless Fact Tables

© Cambridge University Press 2018
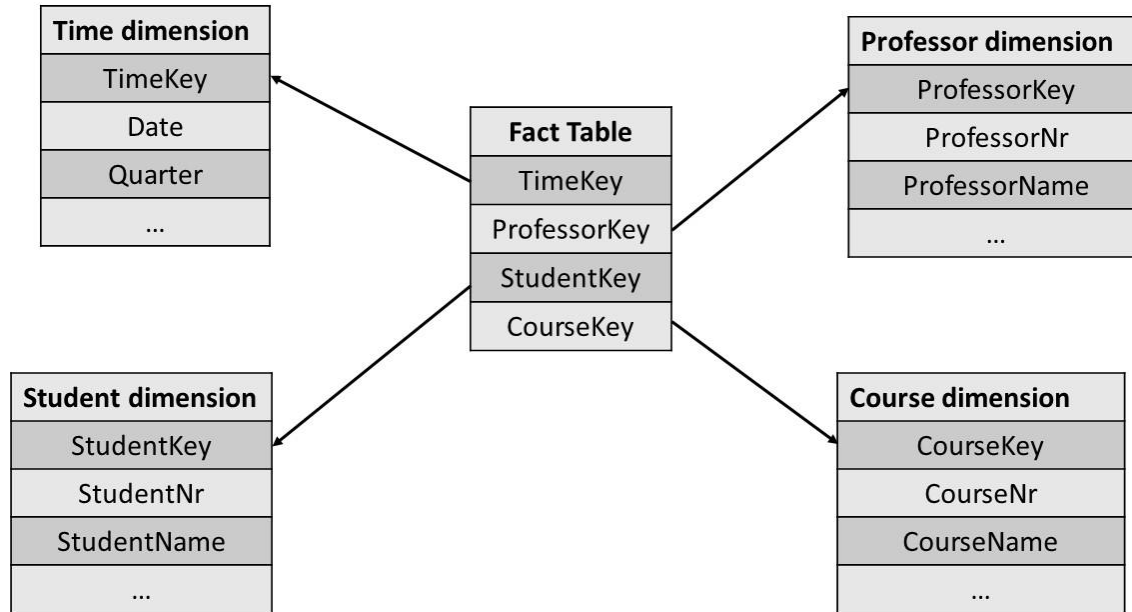
Optimizing the Dimension Tables

- Dimension tables should be heavily indexed to improve query execution time

- On average between five and ten

- E.g., Time
  - TimeKey, Date, DayOfWeek, DayOfMonth, DayOfYear, Month, MonthName, Year, LastDayInWeekFlag, LastDayInMonthFlag, FiscalWeek, HolidayFlag, …

© Cambridge University Press 2018

Junk Dimensions

- Deal with low cardinality attribute types such as flags or indicators

- Example: On-line Purchase (Y/N), Payment (cash or credit card), Discount (Y/N)

- Junk dimension is a dimension that simply enumerates all feasible combinations of values of the low cardinality attribute types

JunkKey1	On-line purchase	Payment	Discount
1	Yes	Credit card	Yes
2	Yes	Credit card	No
3	No	Credit card	Yes
4	No	Credit card	No
5	No	Cash	Yes
6	No	Cash	No

© Cambridge University Press 2018

22

Outrigger Tables

- Store a set of attribute types of a dimension table which are highly correlated, low in cardinality, and updated simultaneously

© Cambridge University Press 2018
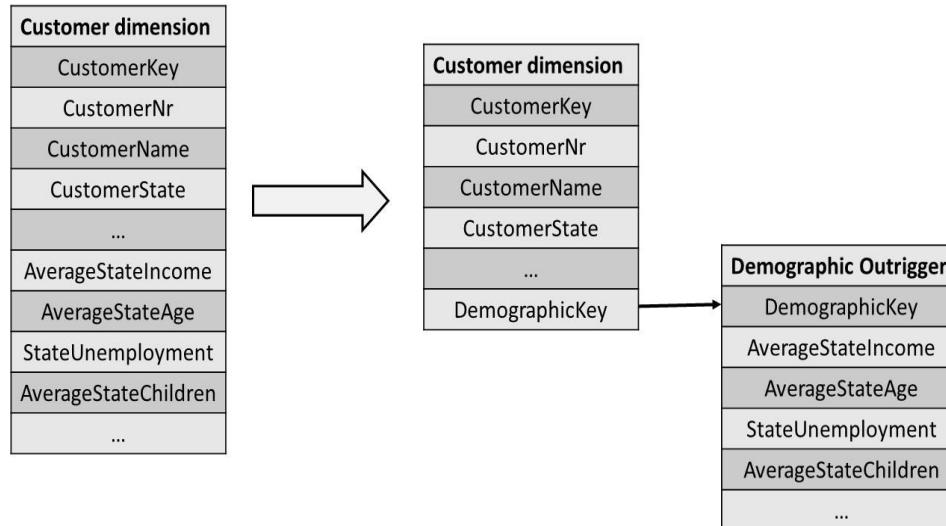
Slowly Changing Dimensions

- Dimensions that change slowly and irregularly over a period of time

- Example: customer segment ranging from AAA, AA, A, BBB, ... to C, determined on a yearly basis

© Cambridge University Press 2018

Slowly Changing Dimensions

- **Approach 1**

OLD STATUS

CustomerKey	CustomerNr	CustomerName	Segment
123456	ABC123	Bart Baesens	AA

NEW STATUS

CustomerKey	CustomerNr	CustomerName	Segment
123456	ABC123	Bart Baesens	AAA

© Cambridge University Press 2018

Slowly Changing Dimensions

- **Approach 2**

OLD STATUS

CustomerKey	CustomerNr	CustomerName	Segment	Start_Date	End_Date	Current_Flag
123456	ABC123	Bart Baesens	AA	27-02-2014	31-12-9999	Y

NEW STATUS

CustomerKey	CustomerNr	CustomerName	Segment	Start_Date	End_Date	Current_Flag
123456	ABC123	Bart Baesens	AA	27-02-2014	27-02-2015	N
123457	ABC123	Bart Baesens	AAA	28-02-2015	31-12-9999	Y

Slowly Changing Dimensions

- **Approach 3**

OLD STATUS

CustomerKey	CustomerNr	CustomerName	Segment
123456	ABC123	Bart Baesens	AA

NEW STATUS

CustomerKey	CustomerNr	CustomerName	Old Segment	New Segment
123456	ABC123	Bart Baesens	AA	AAA

© Cambridge University Press 2018

Slowly Changing Dimensions

- **Approach 4**

OLD STATUS

CustomerKey	CustomerNr	CustomerName	Segment
123457	ABC123	Bart Baesens	AAA

NEW STATUS

CustomerKey	CustomerNr	CustomerName	Segment	Start_Date	End_Date
123456	ABC123	Bart Baesens	AA	27-02-2014	27-02-2015
123457	ABC123	Bart Baesens	AAA	28-02-2015	31-12-9999

© Cambridge University Press 2018

Rapidly Changing Dimensions

- Dimensions that change rapidly and regularly over a period of time

- Example: customer segment ranging from AAA, AA, A, BBB, ... to C, determined on a daily basis

- Approaches 2 and 4 discussed in the previous section will result in a lot of rows

- Split customer information into stable (e.g., gender, marital status, ...) and rapidly changing information which is put in a mini-dimension table

Rapidly Changing Dimensions

© Cambridge University Press 2018
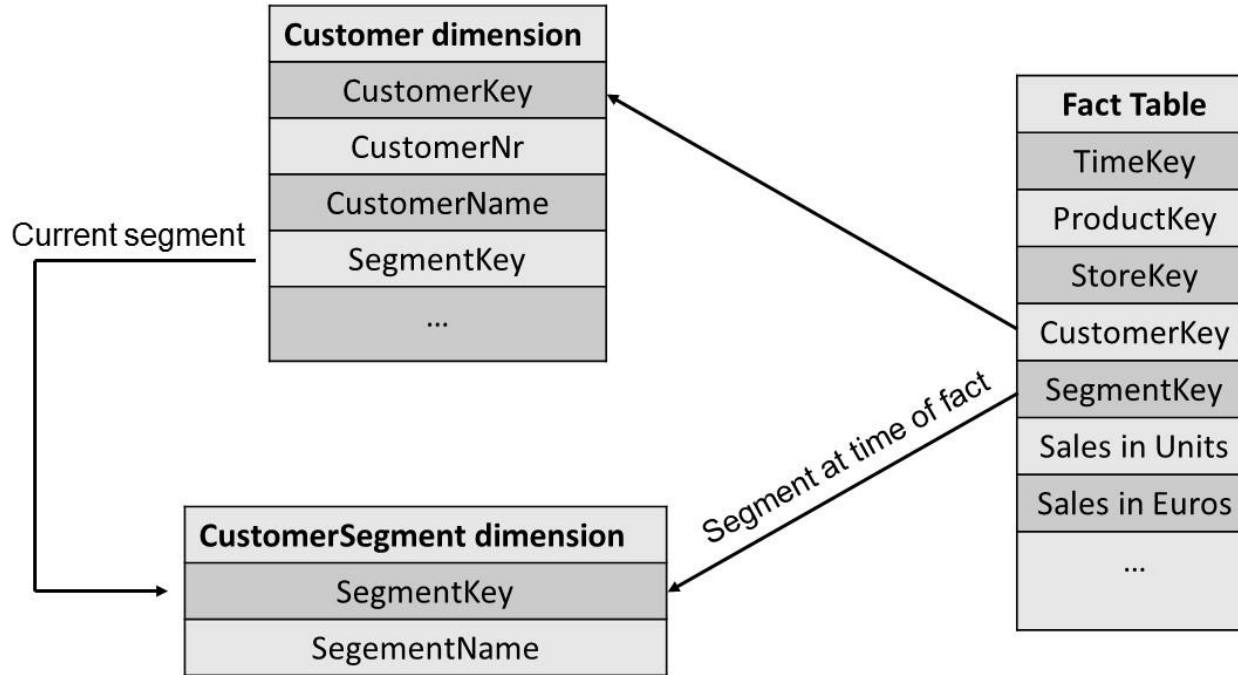
Rapidly Changing Dimensions

Fact table

TimeKey	ProductKey	StoreKey	CustomerKey	SegmentKey	Sales in Units	Sales in Euros
200	50	100	1200	1	6	167.94
210	25	130	1400	4	3	54
180	30	150	1000	3	12	384
...						

Customer Dimension

CustomerKey	CustomerNr	CustName	SegmentKey
1000	20	Bart Baesens	2
1200	10	Wilfried Lemahieu	1
1400	5	Seppe vanden Broucke	1

CustomerSegment Dimension

SegmentKey	SegmentName
1	A
2	B
3	C
4	D

© Cambridge University Press 2018

Rapidly Changing Dimensions

© Cambridge University Press 2018
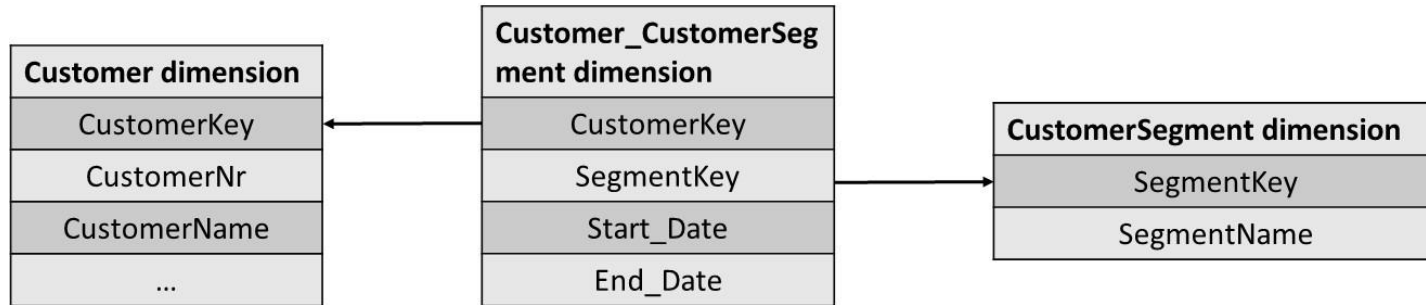
The Extraction, Transformation, and Loading (ETL) Process

- Can consume up to 80% of all efforts needed to set up a data warehouse

- Dump the data in a so-called staging area where all the ETL activities can be executed

© Cambridge University Press 2018

33
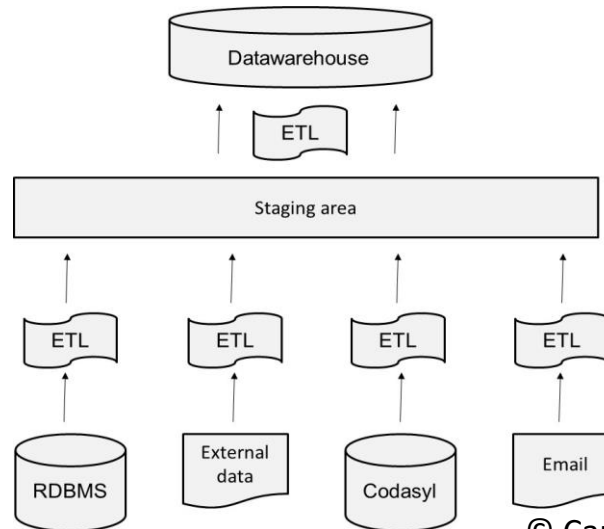
The Extraction, Transformation, and Loading (ETL) Process

- Extraction
  - Full or incremental (changed data capture)
- Transformation
  - formatting
  - cleansing
  - aggregation and merging
  - enrichment
- Loading
  - Fill the fact and dimension tables
- Documentation and metadata

© Cambridge University Press 2018

Data Marts

- A **data mart** is a scaled-down version of a data warehouse aimed at meeting the information needs of a homogeneous small group of end-users such as a department or business unit

- Provide focused content and improved query performance

Data Marts

- **Dependent data marts** pull their data directly from a central data warehouse

© Cambridge University Press 2018
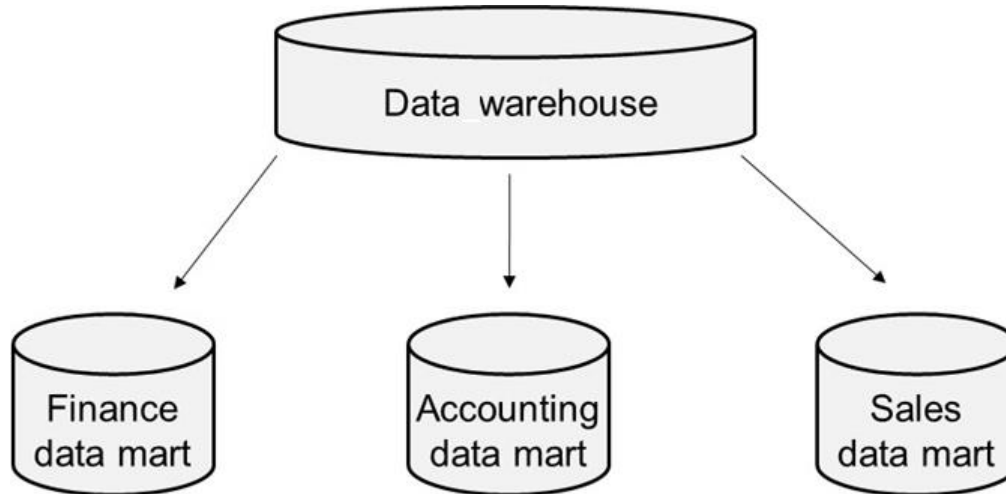
Data Marts

- **Independent data marts** are standalone systems drawing data directly from the operational systems, external sources or a combination of both


© Cambridge University Press 2018
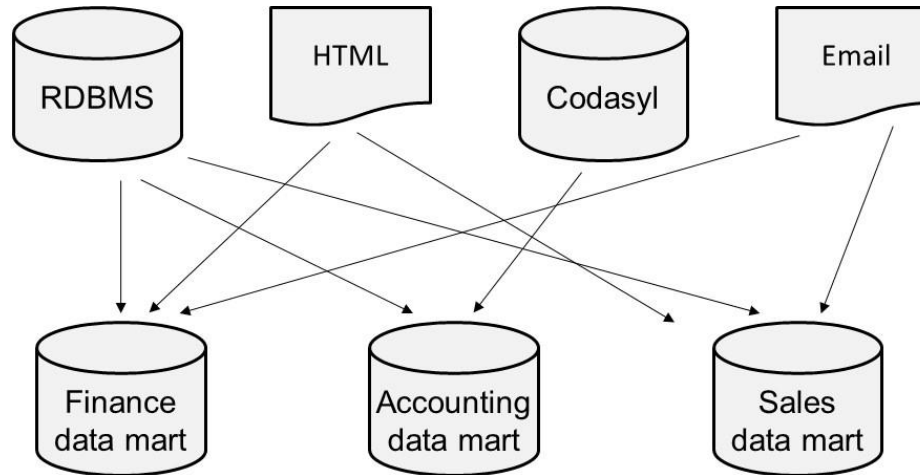
Virtual Data Warehouses and Virtual Data Marts

- Idea of virtualization is to use middleware to create a logical or **virtual data warehouse** or **virtual data mart** which has no physical data but provides a single point of access to a set of underlying physical data stores.

- Data is only accessed ("pulled") at query time.

© Cambridge University Press 2018

Virtual Data Warehouses and Virtual Data Marts

- Virtual Data warehouse can be built as a set of SQL views directly on the underlying operational data sources


© Cambridge University Press 2018
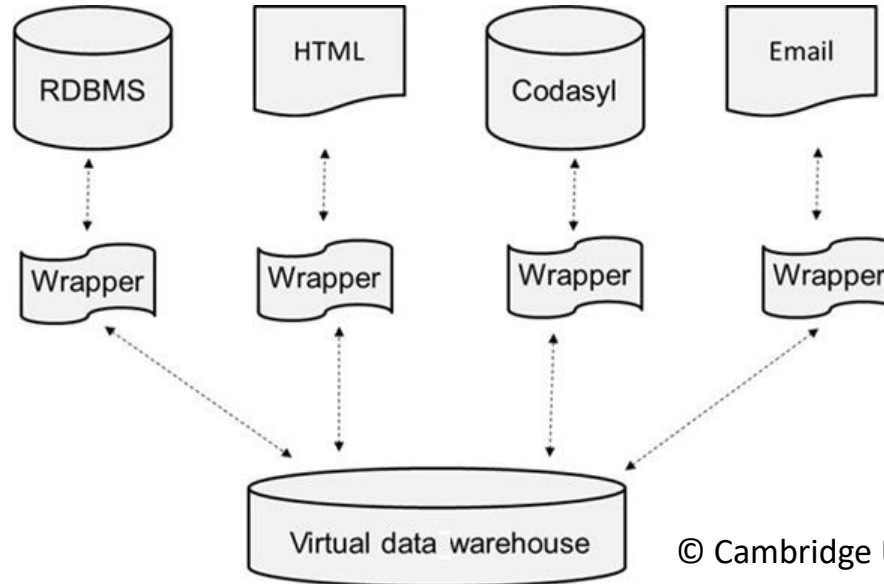
Virtual Data Warehouses and Virtual Data Marts

- Virtual data warehouse can be built as an extra layer on top of a collection of physical independent data marts

© Cambridge University Press 2018
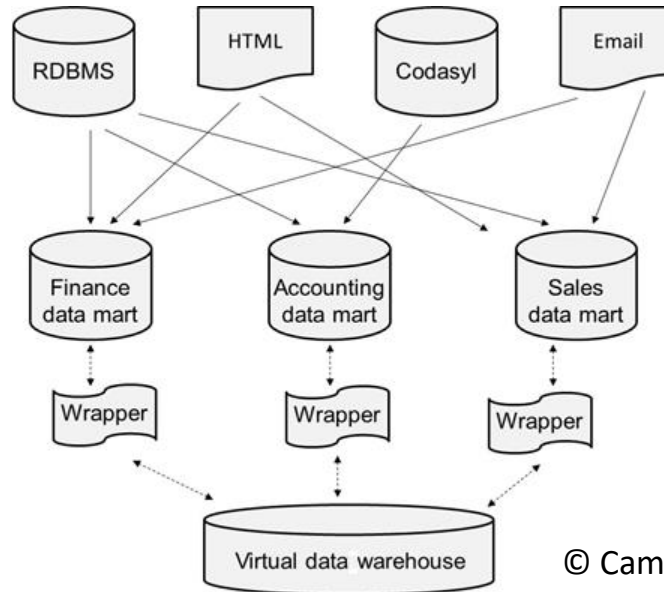
Virtual Data Warehouses and Virtual Data Marts

- Metadata model contains the schema mappings between the schemas of the underlying data stores and the schema of the virtual data warehouse

- Query reformulation

© Cambridge University Press 2018
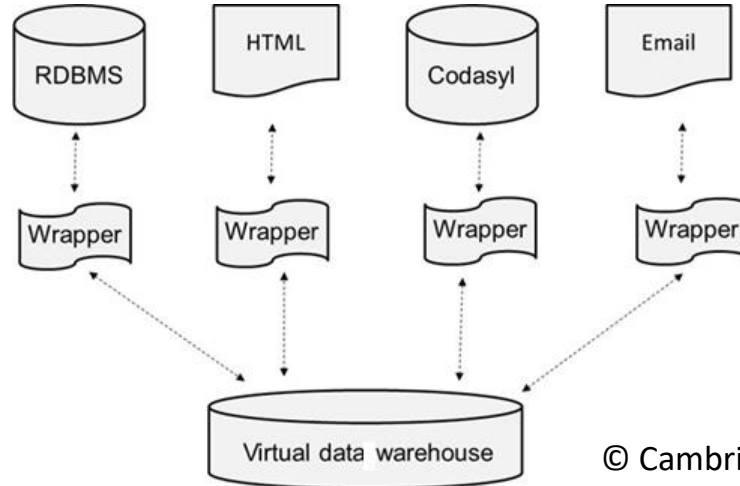
Virtual Data Warehouses and Virtual Data Marts

- A virtual data mart is usually defined as a single SQL view

- Virtual independent versus virtual dependent data mart

- Disadvantages
  - Extra processing capacity from the underlying (operational) data sources
  - Not possible to keep track of historical data

© Cambridge University Press 2018

Operational Data Store

- Operational data store (ODS) can be considered as a staging area that provides query facilities

- Good for analysis tools that need data that are closer to real time

- More complex analyses are still conducted on the actual data warehouse

© Cambridge University Press 2018

Data Warehouses versus Data Lakes

	Data Warehouse	Data Lake
Data	Structured	Often unstructured
Processing	Schema-on-write	Schema-on-read
Storage	Expensive	Low cost
Transformation	Before entering the DW	Before analysis
Agility	Low	High
Security	Mature	Maturing
Users	Decision-makers	Data scientists

© Cambridge University Press 2018

Business Intelligence

- **Business intelligence (BI)** is referred to as the set of activities, techniques, and tools aimed at understanding patterns in past data and predicting the future.

- Garbage in, garbage out (GIGO)

- BI techniques
  - Query and reporting
  - Pivot tables
  - OLAP

Query and Reporting

- Business user can graphically and interactively design a query and corresponding report

- Self-service BI

- Query by example (QBE)
  - a query is composed in a user-friendly and visual way

- Report can be refreshed at any time

- Innovative visualization techniques

Pivot Tables

- A **pivot or cross-table** is a popular data summarization tool. It essentially cross-tabulates a set of dimensions in such a way that multidimensional data can be represented in a two-dimensional tabular format.

Sales		Quarter				Total
		Q1	Q2	Q3	Q4	
Region	Europe	100	200	50	100	450
	Africa	50	100	200	50	400
	Asia	20	50	10	150	230
	America	50	10	100	100	260
	Total	220	360	360	400	1340

© Cambridge University Press 2018   47

On-Line Analytical Processing (OLAP)

- OLAP allows you to interactively analyze the data, summarize it, and visualize it in various ways

- Provides the business user with a powerful tool for ad-hoc querying

- Types
  - MOLAP
  - ROLAP
  - HOLAP

© Cambridge University Press 2018

MOLAP

- Multidimensional OLAP (MOLAP) stores the multidimensional data using a multidimensional DBMS (MDBMS), whereby the data are stored in a multidimensional array-based data structure optimized for efficient storage and quick access.

Array (key, value)	Q1	Q2	Q3	Q4	Total
Product A	(1,1) 10	(1,2) 20	(1,3) 40	(1,4) 10	(1,5) 80
Product B	(2,1) 20	(2,2) 40	(2,3) 10	(2,4) 30	(2,5) 100
Product C	(3,1) 50	(3,2) 20	(3,3) 40	(3,4) 30	(3,5) 140
Product D	(4,1) 10	(4,2) 30	(4,3) 20	(4,4) 20	(4,5) 80
Total	(5,1) 90	(5,2) 110	(5,3) 110	(5,4) 90	(5,5) 400

© Cambridge University Press 2018

MOLAP

- Fast in terms of data retrieval

- More storage space needed

- Scales poorly when the number of dimensions increases

- No universal SQL-like standard is provided

- Not optimized for transaction processing

ROLAP

- **Relational OLAP (ROLAP)** stores the data in a relational data warehouse, which can be implemented using a star, snowflake, or fact constellation schema

- ROLAP scales better to more dimensions than MOLAP

- ROLAP query performance may be inferior to MOLAP

© Cambridge University Press 2018

HOLAP

- **Hybrid OLAP (HOLAP)** tries to combine the best of both MOLAP and ROLAP

- An RDBMS is used to store the detailed data in a relational data warehouse, whereas the pre-computed aggregated data are kept as a multidimensional array managed by an MDBMS

- OLAP analysis first starts from the multidimensional database

- Combines the performance of MOLAP with the scalability of ROLAP

OLAP Operators

© Cambridge University Press 2018
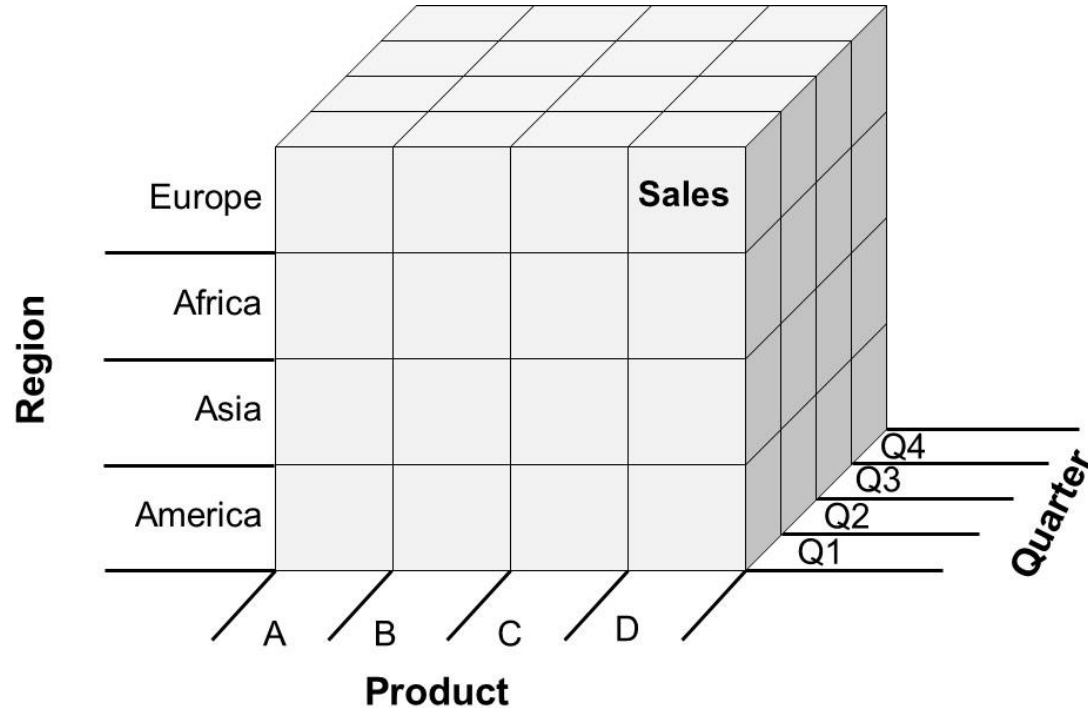
OLAP Operators

- Roll-up refers to aggregating the current set of fact values within or across one or more dimensions

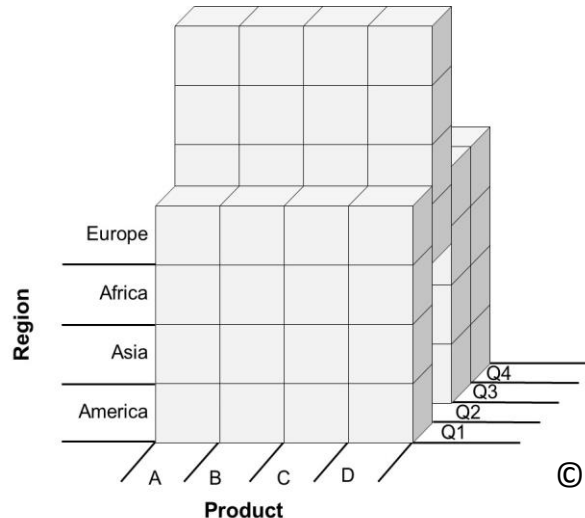- Hierarchical versus dimensional roll-up

Reverse:
drill-down!

© Cambridge University Press 2018

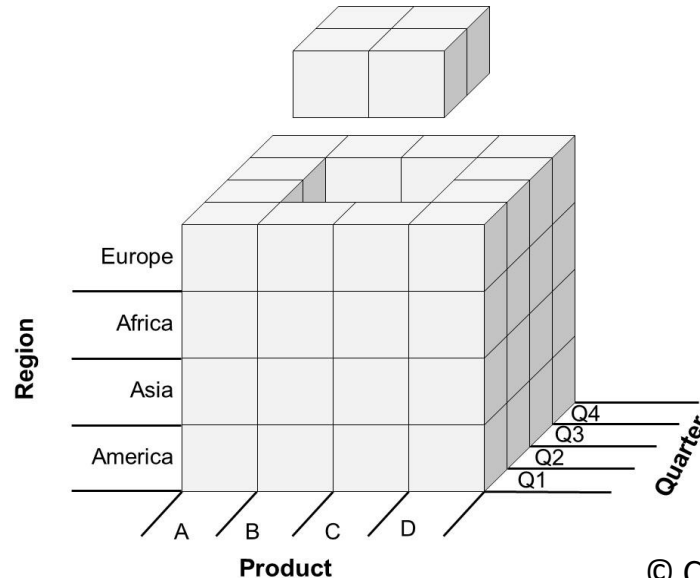# OLAP Operators

- **Drill-across** is another OLAP operation, in which information from two or more connected fact tables is accessed

- **Slicing** represents the operation in which one of the dimensions is set at a particular value

© Cambridge University Press 2018

OLAP Operators

- **Dicing** corresponds to a range selection on one or more dimensions

© Cambridge University Press 2018

OLAP Queries in SQL

- SQL-99 introduced three extensions to the GROUP BY statement: the CUBE, ROLLUP, and GROUPING SETS operators

- The **CUBE** operator computes a union of GROUP BYs on every subset of the specified attribute types

OLAP Queries in SQL

PRODUCT	QUARTER	REGION	SALES
A	Q1	Europe	10
A	Q1	America	20
A	Q2	Europe	20
A	Q2	America	50
A	Q3	America	20
A	Q4	Europe	10
A	Q4	America	30
B	Q1	Europe	40
B	Q1	America	60
B	Q2	Europe	20
B	Q2	America	10
B	Q3	America	20
B	Q4	Europe	10
B	Q4	America	40

© Cambridge University Press 2018

OLAP Queries in SQL

```
SELECT QUARTER, REGION, SUM(SALES)
FROM SALESTABLE
GROUP BY CUBE (QUARTER, REGION)
```

- this query computes the union of 2^2 = 4 groupings of the SALESTABLE, being: {(quarter,region), (quarter), (region), ()}, where () denotes an empty group list

- resulting multiset will have 4 × 2 + 4 × 1 + 1 × 2 + 1 or 15 tuples

© Cambridge University Press 2018

OLAP Queries in SQL

QUARTER	REGION	SALES
Q1	Europe	50
Q1	America	80
Q2	Europe	40
Q2	America	60
Q3	Europe	NULL
Q3	America	40
Q4	Europe	20
Q4	America	80
Q1	NULL	130
Q2	NULL	100
Q3	NULL	40
Q4	NULL	90
NULL	Europe	110
NULL	America	250
NULL	NULL	360

```
CASE WHEN grouping(QUARTER) = 1
THEN 'All' ELSE QUARTER END AS QUARTER and
CASE WHEN grouping(REGION) = 1 THEN
'All' ELSE REGION END AS REGION
```

© Cambridge University Press 2018

OLAP Queries in SQL

- **ROLLUP** operator computes the union on every prefix of the list of specified attribute types, from the most detailed up to the grand total

- The key difference between the ROLLUP and CUBE operators is that the former generates a result set showing the aggregates for a hierarchy of values of the specified attribute types, whereas the latter generates a result set showing the aggregates for all combinations of values of the selected attribute types

© Cambridge University Press 2018

OLAP Queries in SQL

```
SELECT QUARTER, REGION, SUM(SALES)
FROM SALESTABLE
GROUP BY ROLLUP (QUARTER, REGION)
```

- This query generates the union of three groupings {(quarter,region), (quarter}, ()} where () again represents the full aggregation

- Resulting multiset will thus have 4 × 2 + 4 + 1 or 13 rows

© Cambridge University Press 2018

OLAP Queries in SQL

QUARTER	REGION	SALES
Q1	Europe	50
Q1	America	80
Q2	Europe	40
Q2	America	60
Q3	Europe	NULL
Q3	America	40
Q4	Europe	20
Q4	America	80
Q1	NULL	130
Q2	NULL	100
Q3	NULL	40
Q4	NULL	90
NULL	NULL	360

© Cambridge University Press 2018

OLAP Queries in SQL

- GROUP BY ROLLUP construct can also be applied to attribute types that represent different aggregation levels along the same dimension

```
SELECT REGION, COUNTRY, CITY, SUM(SALES)
FROM SALESTABLE
GROUP BY ROLLUP (REGION, COUNTRY, CITY)
```

© Cambridge University Press 2018

64

OLAP Queries in SQL

- **GROUPING SETS** operator generates a result set equivalent to that generated by a UNION ALL of multiple simple GROUP BY clauses

```
SELECT QUARTER, REGION, SUM(SALES)
FROM SALESTABLE
GROUP BY GROUPING SETS ((QUARTER), (REGION))


SELECT QUARTER, NULL, SUM(SALES)
FROM SALESTABLE
GROUP BY QUARTER
UNION ALL
SELECT NULL, REGION, SUM(SALES)
FROM SALESTABLE
GROUP BY REGION
```

© Cambridge University Press 2018

OLAP Queries in SQL

QUARTER	REGION	SALES
Q1	NULL	130
Q2	NULL	100
Q3	NULL	40
Q4	NULL	90
NULL	Europe	110
NULL	America	250

© Cambridge University Press 2018

OLAP Queries in SQL

```
SELECT QUARTER, REGION, SUM(SALES)
FROM SALESTABLE
GROUP BY CUBE (QUARTER, REGION)


SELECT QUARTER, REGION, SUM(SALES)
FROM SALESTABLE
GROUP BY GROUPING SETS ((QUARTER, REGION),
(QUARTER), (REGION), ())
```

© Cambridge University Press 2018

OLAP Queries in SQL

```
SELECT QUARTER, REGION, SUM(SALES)
FROM SALESTABLE
GROUP BY ROLLUP (QUARTER, REGION)


SELECT QUARTER, REGION, SUM(SALES)
FROM SALESTABLE
GROUP BY GROUPING SETS ((QUARTER, REGION),
(QUARTER),())
```

© Cambridge University Press 2018

OLAP Queries in SQL

- SQL2003 introduced support for two types of other activities: ranking and windowing

© Cambridge University Press 2018

OLAP Queries in SQL

PRODUCT	SALES
A	50
B	20
C	10
D	45
E	40
F	30
G	60
H	20
I	15
J	25

© Cambridge University Press 2018

OLAP Queries in SQL

```
SELECT PRODUCT, SALES,
RANK() OVER (ORDER BY SALES ASC) as RANK_SALES,
DENSE_RANK() OVER (ORDER BY SALES ASC) as
DENSE_RANK_SALES, PERCENT_RANK() OVER (ORDER BY
SALES ASC) as PERC_RANK_SALES,
CUM_DIST() OVER (ORDER BY SALES ASC) as
CUM_DIST_SALES,
FROM SALES
ORDER BY RANK_SALES ASC
```

© Cambridge University Press 2018

UMBC

OLAP Queries in SQL

Product	Sales	RANK_SALES	DENSE_RANK_SALES	PERC_RANK_SALES	CUM_DIST_SALES
C	10	1	1	0	0.1
I	15	2	2	1/9 = 0.11	0.2
B	20	3	3	2/9 = 0.22	0.4
H	20	3	3	2/9 = 0.22	0.4
J	25	5	4	4/9 = 0.44	0.5
F	30	6	5	5/9 = 0.55	0.6
E	40	7	6	6/9 = 0.66	0.7
D	45	8	7	7/9 = 0.77	0.8
A	50	9	8	8/9 = 0.88	0.9
G	60	10	9	9/9 = 1	1

© Cambridge University Press 2018

OLAP Queries in SQL

- **Windowing** allows calculating cumulative totals or running averages based on a specified time window.

QUARTER	REGION	SALES
1	America	10
2	America	20
3	America	10
4	America	30
1	Europe	10
2	Europe	20
3	Europe	10
4	Europe	20

© Cambridge University Press 2018

OLAP Queries in SQL

```
SELECT QUARTER, REGION, SALES,
AVG(SALES) OVER (PARTITION BY REGION ORDER BY
QUARTER ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING) AS
SALES_AVG
FROM SALES
ORDER BY REGION, QUARTER, SALES_AVG
```
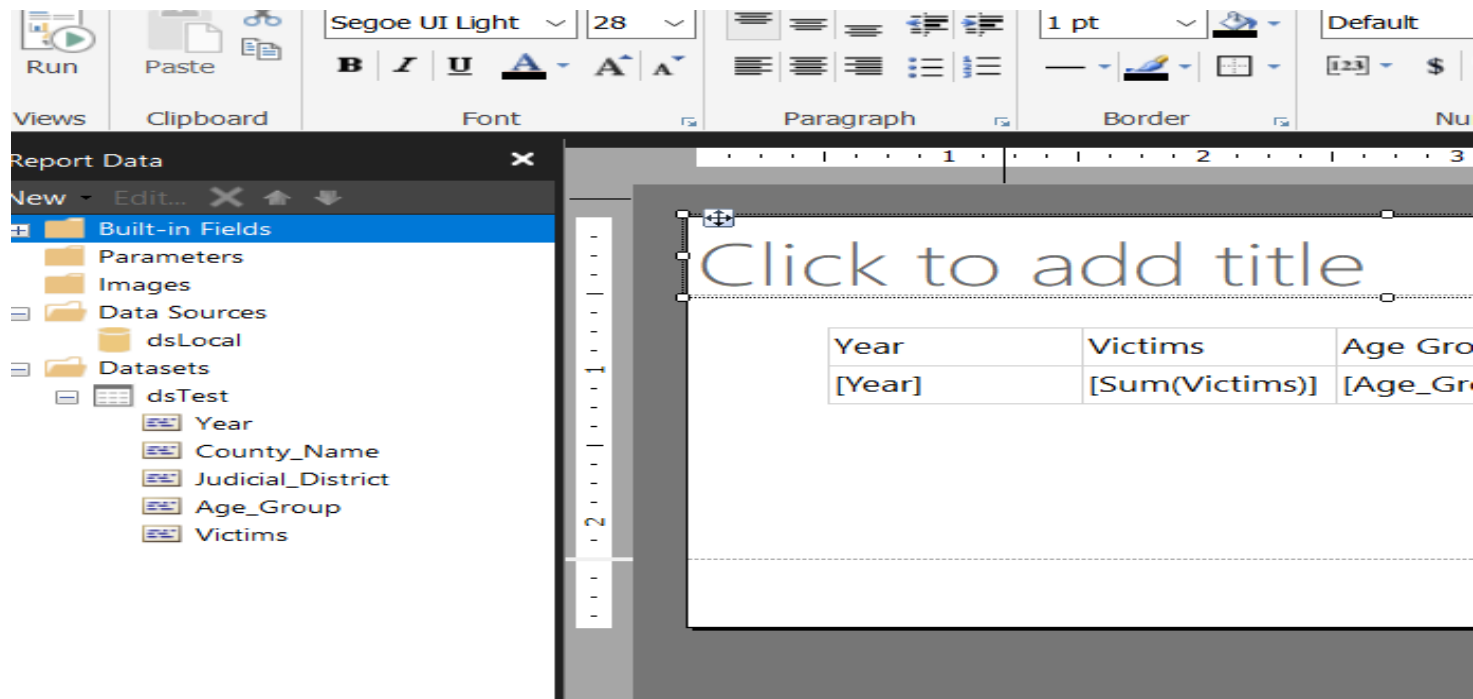
OLAP Queries in SQL

QUARTER	REGION	SALES	SALES_AVG
1	America	10	15
2	America	20	13,33
3	America	10	20
4	America	30	20
1	Europe	10	15
2	Europe	20	13,33
3	Europe	10	16,67
4	Europe	20	15

© Cambridge University Press 2018

Conclusions

- Operational versus tactical/strategic decision-making
- Data warehouse definition
- Data warehouse schemas
- The extraction, transformation, and loading (ETL) process
- Data marts
- Virtual data warehouses and virtual data marts
- Operational data store
- Data warehouses versus data lakes
- Business intelligence

© Cambridge University Press 2018

Reporting

Common Reporting Tools

- Crystal Reports
- Microsoft Access
- Microsoft Excel
- Microsoft SQL Server Reporting Services
- Oracle OBIEE
- IBI WebFocus
- Custom Developed Reports within Applications

Basics of Report Creation

- Create Data source
  - Establishes connection to database
  - Some tools support more than one data source
- Create Dataset
  - Defines what data (tables, columns) you want to pull into report
- Display Data
  - Display the data you want the users to see and how you want them to see it (e.g. matrix or table view)
  - Can include user interactivity to enable filters and drilldowns
  - Can include charts and graphs
- Publish or Distribute

Data Visualizations

- Representation of data in graphical and multimedia formats for human analysis

- "A picture tells a thousand words"

- Without showing precise values, graphs and charts can depict relationships in the data

- Often used in dashboards, as shown in next slide

Common Data Visualization Tools

- Tableau
- Microsoft PowerBI
- QuikView
- Oracle OBIEE
- SAS Visual Analytics
- SAS JMP
- R
- Python

Sample Dashboard

Business Performance Management (BPM) systems allow managers to measure, monitor, and manage key activities and processes to achieve organizational goals.

Dashboards are often used to provide an information system in support of BPM.

Data 604: Data Management   83