Databases
Lectures 9 and 10

Timothy G. Griffin

Computer Laboratory
University of Cambridge, UK

Databases, Lent 2009

Lecture 09 and 10

Two Themes ...

- Redundancy can be a GOOD thing!
- Duplicates, aggregates, and group by in SQL, and evolution to "Data Cube"

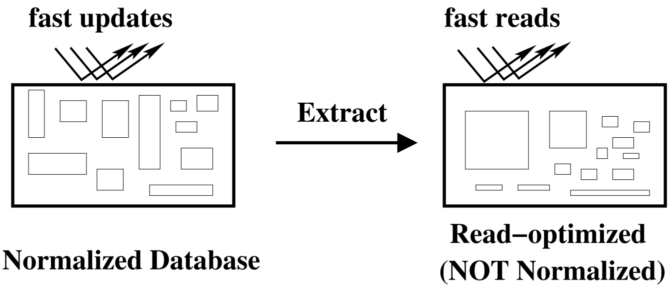
.... come together in OLAP

- OLTP : Online Transaction Processing (traditional databases)
  - Data is normalized for the sake of updates.
- OLAP : Online Analytic Processing
  - These are (almost) read-only databases.
  - Data is de-normalized for the sake of queries!
  - Multi-dimensional data cube emerging as common data model.
    - This can be seen as a generalization of SQL's group by

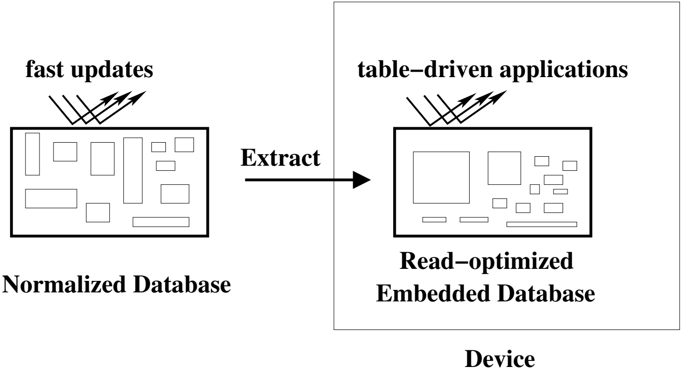
Materialized Views

- Suppose Q is a very expensive, and very frequent query.
- Why not de-normalize some data to speed up the evaluation of Q?
  - This might be a reasonable thing to do, or ...
  - ... it might be the first step to destroying the integrity of your data design.
- Why not store the value of Q in a table?
  - This is called a materialized view.
  - But now there is a problem: How often should this view be refreshed?

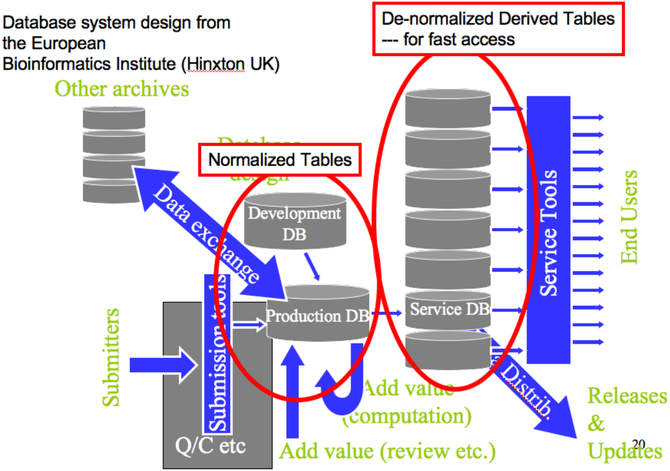
FIDO = Fetch Intensive Data Organization



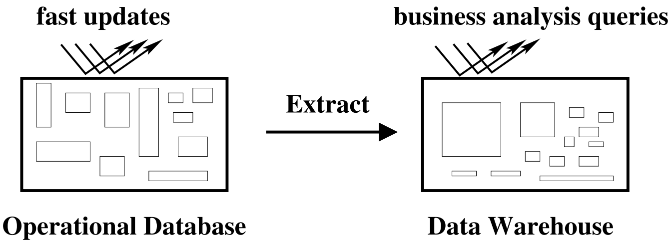
Example : Embedded databases



Example : Hinxton Bioinformatics



Example : Data Warehouse (Decision support)



OLAP vs. OLTP

OLTP Online Transaction Processing

OLAP Online Analytical Processing

- Commonly associated with terms like Decision Support, Data Warehousing, etc.

	OLAP	**OLTP**
Supports	analysis	day-to-day operations
Data is	historical	current
Transactions mostly	reads	updates
optimized for	query processing	updates
Normal Forms	not important	important

OLAP Databases : Data Models and Design

The big question

Is the relational model and its associated query language (SQL) well suited for OLAP databases?

- Aggregation (sums, averages, totals, ...) are very common in OLAP queries
  - Problem : SQL aggregation quickly runs out of steam.
  - Solution : Data Cube and associated operations (spreadsheets on steroids)
- Relational design is obsessed with normalization
  - Problem : Need to organize data well since all analysis queries cannot be anticipated in advance.
  - Solution : Multi-dimensional fact tables, with hierarchy in dimensions, star-schema design.

Let's start by looking at aggregate queries in SQL ...

An Example ...

```
mysql> select * from marks;
  +-------+------------+------+
  | sid   | course     | mark |
  +-------+------------+------+
  | ev77  | databases  |   92 |
  | ev77  | spelling   |   99 |
  | tgg22 | spelling   |    3 |
  | tgg22 | databases  |  100 |
  | fm21  | databases  |   92 |
  | fm21  | spelling   |  100 |
  | jj25  | databases  |   88 |
  | jj25  | spelling   |   92 |
  +-------+------------+------+
```

... of duplicates

```
mysql> select mark from marks;
+------+
| mark |
+------+
|   92 |
|   99 |
|    3 |
|  100 |
|   92 |
|  100 |
|   88 |
|   92 |
+------+
```

Why Multisets?

Duplicates are important for aggregate functions.

```
mysql> select min(mark),
              max(mark),
              sum(mark),
              avg(mark)
       from marks;
+-----------+-----------+-----------+-----------+
| min(mark) | max(mark) | sum(mark) | avg(mark) |
+-----------+-----------+-----------+-----------+
|         3 |       100 |       666 |   83.2500 |
+-----------+-----------+-----------+-----------+
```

The `group by` clause

```
mysql> select course,
              min(mark),
              max(mark),
              avg(mark)
       from marks
       group by course;
+-----------+-----------+-----------+-----------+
| course    | min(mark) | max(mark) | avg(mark) |
+-----------+-----------+-----------+-----------+
| databases |        88 |       100 |   93.0000 |
| spelling  |         3 |       100 |   73.5000 |
+-----------+-----------+-----------+-----------+
```

Visualizing group by

sid	course	mark
ev77	databases	92
ev77	spelling	99
tgg22	spelling	3
tgg22	databases	100
fm21	databases	92
fm21	spelling	100
jj25	databases	88
jj25	spelling	92

`group by` $\Longrightarrow$

course	mark
spelling	99
spelling	3
spelling	100
spelling	92

course	mark
databases	92
databases	100
databases	92
databases	88

Visualizing group by

course	mark
spelling	99
spelling	3
spelling	100
spelling	92

course	mark
databases	92
databases	100
databases	92
databases	88

min(**mark**) $\Longrightarrow$

course	min(**mark**)
spelling	3
databases	88

The `having` clause

How can we select on the aggregated columns?

```
mysql> select course,
              min(mark),
              max(mark),
              avg(mark)
       from marks
       group by course
       having min(mark) > 60;
+-----------+-----------+-----------+-----------+
| course    | min(mark) | max(mark) | avg(mark) |
+-----------+-----------+-----------+-----------+
| databases |        88 |       100 |   93.0000 |
+-----------+-----------+-----------+-----------+
```

Use renaming to make things nicer ...

```
mysql> select course,
              min(mark) as minimum,
              max(mark) as maximum,
              avg(mark) as average
       from marks
       group by course
       having minimum > 60;
+-----------+---------+---------+---------+
| course    | minimum | maximum | average |
+-----------+---------+---------+---------+
| databases |      88 |     100 | 93.0000 |
+-----------+---------+---------+---------+
```

Limits of SQL aggregation

sale	prodId	storeId	amt
	p1	c1	12
	p2	c1	11
	p1	c3	50
	p2	c2	8

⟷

	c1	c2	c3
p1	12		50
p2	11	8	

- Flat tables are great for processing, but hard for people to read and understand.
- Pivot tables and cross tabulations (spreadsheet terminology) are very useful for presenting data in ways that people can understand.
- SQL does not handle pivot tables and cross tabulations well.

A very influential paper [G+1997]

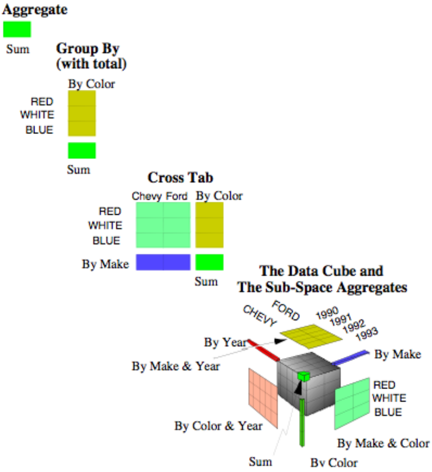
Data Mining and Knowledge Discovery 1, 29–53 (1997)

Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals*

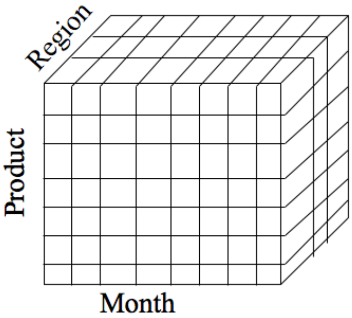
JIM GRAY Gray@Microsoft.com
SURAJIT CHAUDHURI SurajitC@Microsoft.com
ADAM BOSWORTH AdamB@Microsoft.com
ANDREW LAYMAN AndrewL@Microsoft.com
DON REICHART DonRei@Microsoft.com
MURALI VENKATRAO MuraliV@Microsoft.com
Microsoft Research, Advanced Technology Division, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

FRANK PELLOW Pellow@vnet.IBM.com
HAMID PIRAHESH Pirahesh@Almaden.IBM.com
IBM Research, 500 Harry Road, San Jose, CA 95120

From aggregates to data cubes



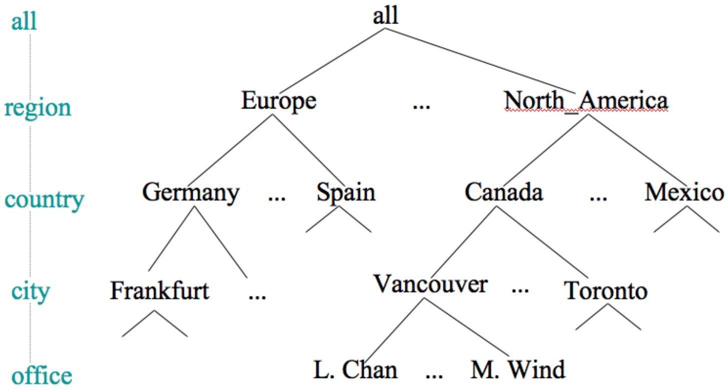
The Data Cube



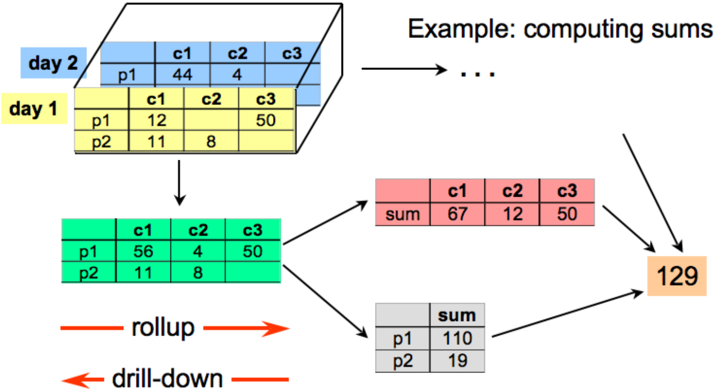
Dimensions:
Product,
Location,
Time

- Data modeled as an *n*-dimensional (hyper-) cube
- Each dimension is associated with a hierarchy
- Each "point" records facts
- Aggregation and cross-tabulation possible along all dimensions

Hierarchy for **Location** Dimension



Cube Operations



The Star Schema as a design tool