AG Heterogene Informationssysteme
Prof . Dr.-Ing. Stefan Deßloch
Fachbereich Informatik
Technische Universität Kaiserslautern

# Recent Developments for Data Models – Exercise 4

Monday, June 28, 2012 – 15:30 to 17:00 – Room 36-336

## 1) Multi-dimensional Modeling

Multi-dimensional schemas are typically used for data warehousing. The design goals are query performance, user understandability (through simplicity), and resilience to changes.

Consider the following scenario: An online bookstore company wants to build a data warehouse to better understand its sales. Each book has a unique identifier and a title. Books are further classified into genres. The price on which books are purchased by the company is known as *cost price*. The price on which books are sold to customers is known as *sales price*. The difference between cost price and sales price is known as *gross profit*. The *gross margin* can be calculated by dividing the gross profit by the sales price.

Customers need to register before they can place orders. During registration customers have to enter their name, email address, age, city, state, and country. Each customer is assigned a unique identifier (customer key).

Orders may contain multiple order lines. Each order line refers to one or more copies of the same book. Each order is assigned a unique identifier (order key), order lines are numbered consecutively. For analyzing the sales it is important to understand when an order was placed, i.e. at what year, quarter, month, day of month, week of year, day of week, and whether the day was a holiday.

a. Design a star schema that captures the information to be analyzed described in the scenario above!
b. What are the differences between a star schema and a schema in third-normal-form? What are the differences between a star schema and a snowflake schema? What are the advantages and drawbacks of either approach?

## 2) Data Analysis in SQL

In SQL:1999 the GROUP BY clause was extended for improved data analysis. These extensions, invoked with the keywords CUBE, ROLLUP, and GROUPING SETS, provide multi-dimensional summaries for grouped data.

a. Reconsider the star schema designed in 1). Specify SQL queries to retrieve the following information. (Assume that a view named `sales` has been defined that joins the fact table and the dimension tables of the star schema).

   1) The revenue (sum of sales price) of book sales per genre and year.
   2) The revenue of book sales per genre and per year with subtotals for each year and the grand total.
   3) The revenue per city, state, and country with subtotals for each state and country, subtotals for each country, and the grand total.
   4) The revenue per city, state, and country with subtotals for each state and country, and subtotals for each country.
   5) The average order total per calendar month, the average order total per calendar year, the average order total per month and year, and the overall average order total.
   6) The revenue of book sales in 2009 per genre and per state and country (in combination) with subtotals for each genre and the grand total.
   7) The revenue of book sales in each country per year, per quarter, per year and quarter, and the grand total.
   8) The gross margin of book sales by genre and year and the grand total.

b. Rewrite the following queries using alternative grouping features.

   1) Use ordinary grouping to rephrase the following query.

```
select genre, year, sum(sales_price) as revenue
from sales
group by ROLLUP (year, genre)
```

   2) Use GROUPING SETS to rephrase the following query.

```
select month, day_of_month,
       avg(order_total) as avg_order_total
from (
    select month, day_of_month,
           sum(sales_price) as order_total
    from sales
    group by month, day_of_month, order_key)
group by CUBE(month, day_of_month);
```

3) Use GROUPING SETS to rephrase the following query.

```
select year, quarter, month, sum(gross_profit)
from sales
group by year, ROLLUP(quarter, month);
```

4) Use GROUPING SETS to rephrase the following query.

```
select genre, country, state, avg(quantity)
from sales
group by GROUPING SETS(ROLLUP(genre),
         ROLLUP(country, state));
```

## 3) Window Functions in SQL

The most fundamental enhancement that SQL/OLAP adds to the SQL language is the notion of windows – a user-defined selection of rows within a query that determines the set of rows used to perform certain calculations.

Reconsider the star schema designed in 1). Specify SQL queries to retrieve the following information. (Again, assume that a view named `sales` has been defined that joins the fact table and the dimension tables of the star schema).

a. The total amount of each order in 2009 together with the moving average over the last three orders of the respective customer.

b. The total amount of each order in 2009 together with the moving average over orders placed in the last three month by the respective customer.

c. The monthly sales and the cumulative sales for each genre in 2009 ordered by genre and month.

d. The actual sales in each state together with the average sales in that state in the previous three month.

e. A ranking of the bestselling books in 2009.

f. A ranking of the most valuable customers (highest order totals) in the last quarter of 2009 per country and per state.

g. The monthly order totals together with the cumulative order totals of customers from "Texas" in 2009.

h. The percentage contribution of cities to the total sales of the respective countries.