

Politecnico di Torino

Database Management Systems

February 6th 2012

1. (7 Points) The following relations are given (primary keys are underlined):

```
TECHNICIAN(TId, Surname, BirthDate, Qualification, Laboratory)
PATIENT(PId, BirthDate, Disease, HId)
ANALYSIS(Date, PId, TId, Type, Results)
HOSPITAL(HId, Name, Address, City)
```

Assume the following cardinalities:

- $\text{card}(\text{TECHNICIAN}) = 10^4$ tuples,
MIN(BirthDate) = 1-1-1960, MAX(BirthDate) = 31-12-1989,
distinct values of Qualification $\simeq 10$,
- $\text{card}(\text{PATIENT}) = 10^6$ tuples,
MIN(BirthDate) = 1-1-1912, MAX(BirthDate) = 31-12-2011,
distinct values of Disease $\simeq 100$
- $\text{card}(\text{ANALYSIS}) = 10^7$ tuples,
MIN(Date) = 1-1-2011, MAX(Date) = 31-12-2011
distinct values of Type $\simeq 10$
- $\text{card}(\text{HOSPITAL}) = 10^3$ tuples,
distinct values of City $\simeq 10$

Furthermore, assume the following reduction factor for the group by condition:

- $\text{having count} (*) > 10 \simeq \frac{1}{100}$.

Consider the following SQL query:

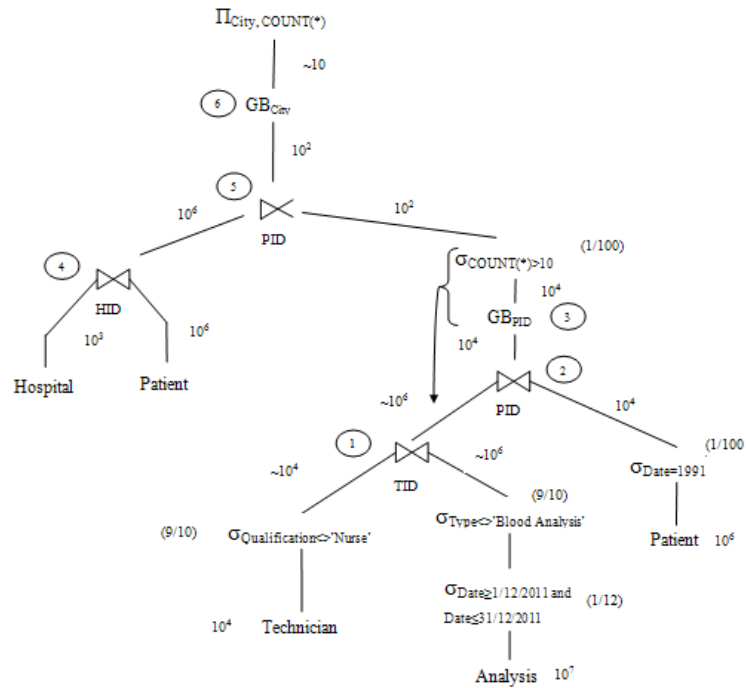
```
select City, Count(*)
from HOSPITAL H, PATIENT P
where H.PId=P.PId and
      and PId in (select PId from PATIENT P1, ANALYSIS A, TECHNICIAN T
                 where P1.PId=A.PId and A.TId=T.TId
                 and P1.BirthDate = 1991 and Qualification <>'Nurse'
                 and A.Date ≥ 1/12/2011 and A.Date ≤ 31/12/2011
                 and Type <> 'Blood Analysis'
                 group by PId
                 having count (*) > 10)
group by City;
```

For the SQL query:

- Report the corresponding algebraic expression and specify the cardinality of each node (representing an intermediate result or a leaf). If necessary, assume a data distribution. Also analyze the group by anticipation.
- Select one or more secondary physical structures to increase query performance. Justify your choice and report the corresponding execution plan (join orders, access methods, etc.).

Join and group by discussion:

- (1) Hash Join
- (2) Hash Join



- (3) GB (NO) Hash according to previous join
- (4) Hash Join /Nested Loop
- (5) Nested Loop
- (6) GB Sort

Indexes:

- Table ANALYSIS: B⁺-Tree on Date
- Table PATIENT: B⁺-Tree on BirthDate

2. (8 Points) The following relations are given (primary keys are underlined, optional attributes are denoted with *):

```
PRODUCT (ProductCode, PName, Price, ProductPoints)
LOYALTY_CARD (CardCode, CustomerName, TotalPoints)
PURCHASE (PurchaseCode, ProductCode, Date, CardCode*, NumberOfItems)
PRIZE (PrizeCode, PrizeDescription, NeededPoints)
NOTIFICATION_REQUEST (NCode, CardCode, PrizeCode, PrizeDescription)
```

A supermarket wants to manage some activities relating to loyalty cards. For each loyalty card, the LOYALTY_CARD table contains the total points (attribute TotalPoints) acquired by a customer. The PRIZE table describes the available prizes. For each prize, the NeededPoints attribute defines the value of the prize in points. Write the triggers managing the following activities.

(1) *Assignment of points for a purchase and possible selection of the prize.* Write the trigger to update the state of the loyalty card of the customer who made the purchase. When a new purchase is made (insert in the PURCHASE table), the total points achieved by the customer must be updated. For each product, the ProductPoints attribute contains the points related to the purchase of a single item. To calculate the total points achieved in the purchase, you should consider the total number of purchased items (attribute NumberOfItems). When the product is not worth the acquisition of any points, the attribute ProductPoints is equal to zero. Note that purchases are not necessarily associated with a loyalty card. When the purchase is not associated with a loyalty card, the value of the CardCode attribute is NULL and the LOYALTY_CARD table should not be updated.

Next, you must check if the total points accumulated by the customer allow her to obtain a prize, i.e., if there is at least one prize having value (attribute NeededPoints) less than or equal to the total points accumulated by the customer. If so, among the prizes that can be received with the accumulated points, the prize with the maximum value must be chosen. Assume that there is at most one prize satisfying this condition. Finally, a request to notify the possibility of receiving the selected prize for the cardholder must be inserted in the NOTIFICATION_REQUEST table. The NCode primary key is a counter, which should be incremented each time a new notification request is inserted (note that fully processed notification requests could be removed from database).

```
create trigger PointsUpdate
after insert on purchase
for each row
when (NEW.CardCode IS NOT NULL)
declare
  myPrizeDescription varchar(10);
  N, P, Z, PCode number;

begin

--read points for the product
  select ProductPoints into P
  from PRODUCT
  where ProductCode = :NEW.ProductCode;
```

```

if ( P > 0) then
  --update loyalty card
  update LOYALTY_CARD
  set TotalPoints = TotalPoints + P * :NEW.NumberOfItems
  where CardCode = :NEW.CardCode;

  --read maximum needed points for a prize
  select max(NeededPoints) INTO Z
  from PRIZE
  where NeededPoints <= (select TotalPoints
                        from LOYALTY_CARD
                        where CardCode = :NEW.CardCode);
  if Z IS NOT NULL then
    -- select the prize
    select PrizeCode, PrizeDescription into PCode, myPrizeDescription
    from PRIZE
    where NeededPoints = Z;

    --read maximum NCode
    select max(NCode) into
    from NOTIFICATION_REQUEST;

    if N IS NULL then
      N := 0;
    end if;

    --insert notification
    insert into NOTIFICATION_REQUEST (NCode, CardCode, PrizeCode, PrizeDescription)
    values (N+1, :NEW.CardCode, PCode, myPrizeDescription);
  end if;
end if;

end;

```

(2) *Integrity constraint on the maximum value in points for a product.* For each product, the value in points (attribute `ProductPoints` in the `PRODUCT` table) can not exceed 1/10 of the sale price (attribute `Price`). Write the trigger managing the integrity constraint, by assigning the maximum value, equal to 1/10 of the sale price, when the maximum allowed value is exceeded.

```

create trigger checkProductPoints
before insert or update of ProductPoints, Price on PRODUCT
for each row
when (NEW.ProductPoints > NEW.Price/10)
begin
  :NEW.ProductPoints := :NEW.Price/10;
end;

```

3. Data Warehouse design

Problem specifications

A company providing services for the promotion of international trade wishes to assess the quantity and the quality of the services delivered by its offices all over the world by analyzing the volume of business in each office on an international scale.

The company is interested in designing and developing a data warehouse offering an overview on the company business, allowing the analysis of the average daily volume of business and the average number of requested services in its offices, according to:

- the month, the 4-month period and the year in which the volume of business has been achieved
- the date, the day of the week, the day of the month,
- the office,
- the service and the type of service (e.g., financial, trade counseling, insurance),
- the class of service, i.e., whether the service is available on the web or on the phone,
- the type of customer (e.g., private, professional, small enterprise, etc.)
- payment type,
- the province, the region, the country and the continent where the office is based.

The data warehouse will store information about years 2004-2011. To correctly implement the data warehouse, the following statistics are known:

- number of offices \simeq 1500
- number of types of customers \simeq 5
- number of services \simeq 300

Business (OId, SId, DateId, Payment_Type, Customer_Type, BusinessVol, #Scheduled_Services)

Office (OId, Office, Province, Region, Country, Continent)

Service (SId, Service, Service_type, Class)

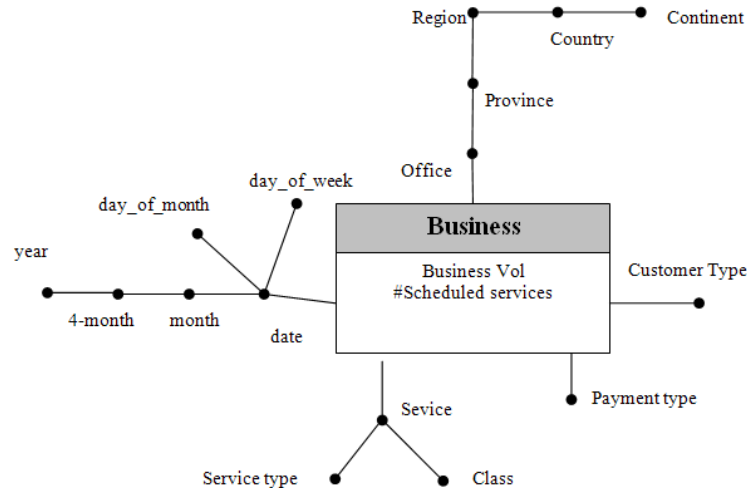
Time (DateId, Date, Day_of_Week, Day_of_Month, Month, 4-Month, Year)

The following are some of the frequent analyses the company is interested in:

- (a) For each type of service and considering only the volume of business on 2009, select for each office and 4-month period the total volume of business, the total number of provided services, the average daily volume of business and the average number of requested services.

```
SELECT ServiceType, Office, 4-Month
       SUM (BusinessVol), Sum(#RequestedServices),
       SUM (BusinessVol)/COUNT(Distinct Date),
       SUM (#RequestedServices)/COUNT(Distinct Date)
FROM Business AS B, Service AS S, Office AS O,Time AS T
WHERE B.OId = O.Oid AND B.SId = S.SId AND B.DateId = T.DateId AND Year =2009
GROUP BY ServiceType, Office, 4-Month;
```

- (b) For each pair (office city, year) select the total volume of business and the total number of requested services.



- (c) For the offices based in France, select the quarterly volume of business, the average daily volume of business for each trimester and the total cumulative quarterly volume of business since the beginning of the year, separately for each type of service.

```

SELECT ServiceType, Office, 4-Month, Year
      SUM (BusinessVol)
      SUM (BusinessVol)/COUNT(Distinct Date),
      SUM( SUM (BusinessVol)) OVER (PARTITION BY Office, ServiceType, Year
                                   ORDER BY 4-Month
                                   ROWS UNBOUNDED PRECEDING)
FROM Business AS B, Service AS S, Office AS O,Time AS T
WHERE B.OId = O.Oid AND B.SId = S.SId AND B.DateId = T.DateId AND Country = 'France'
GROUP BY ServiceType, Office, 4-Month, Year;

```

Design

- (a) (6 Points) Design the data warehouse to address the described issues. In particular, the designed data warehouse must allow efficient execution of all the queries described in the specifications.
- (b) (8 Points) Write frequent queries (a) and (c) of the “problem specifications” using the extended SQL language.