Technologies for Information Systems

Part II

prof. L. Tanca – February 13, 2012

Available Time 2 hours

Last Name

First Name

Student ID **Signature**

IceCompany is a European company that provides winter services (e.g., snow removal and de-icing of roads) in many European cities. The company has many snow removal vehicles and salt spreader vehicles. The management of *IceCompany* is interested in analyzing the income of the company and expanding the activities in new European cities. For these reasons, the management of *IceCompany* has asked you to design its data warehouse in order to be able to analyze the income of *IceCompany* and the weather conditions in European cities (e.g., daily snowfall, number of days with the presence of ice). The weather condition information will be used to select new cities and countries as prospective clients.

The following is the logical schema of the *IceCompany* operational database. It contains data about income and weather conditions for the past five years.

City(City-ID, Name, Country, Altitude, Official-Language)
WeatherObservation(City-ID, Date, Time, Temperature, Ice, Wind, Snowfall)
Vehicle(RegistrationNumber, Name, YearOfIssue, Manufacturer, Type)
VehicleRental(RegistrationNumber, Date, City-ID, CostPaidByCity)

The attribute *CostPaidByCity* in *VehicleRental* represents the amount paid by the city for the rented vehicle (i.e., it represents the income for *IceCompany*).
The attribute *Ice* in *WeatherObservation* can assume the values *true* or *false*.
The attribute *Snowfall* in *WeatherObservation* stores the snowfall in cm.
The attribute *Type* in *Vehicle* can assume the values '*snow removal vehicle*' or '*salt spreader*'.

1. Perform the reverse engineering of the given logical schema into a conceptual schema (Entity-Relationship model).

2. With respect to the produced conceptual schema and the analysis of interest:

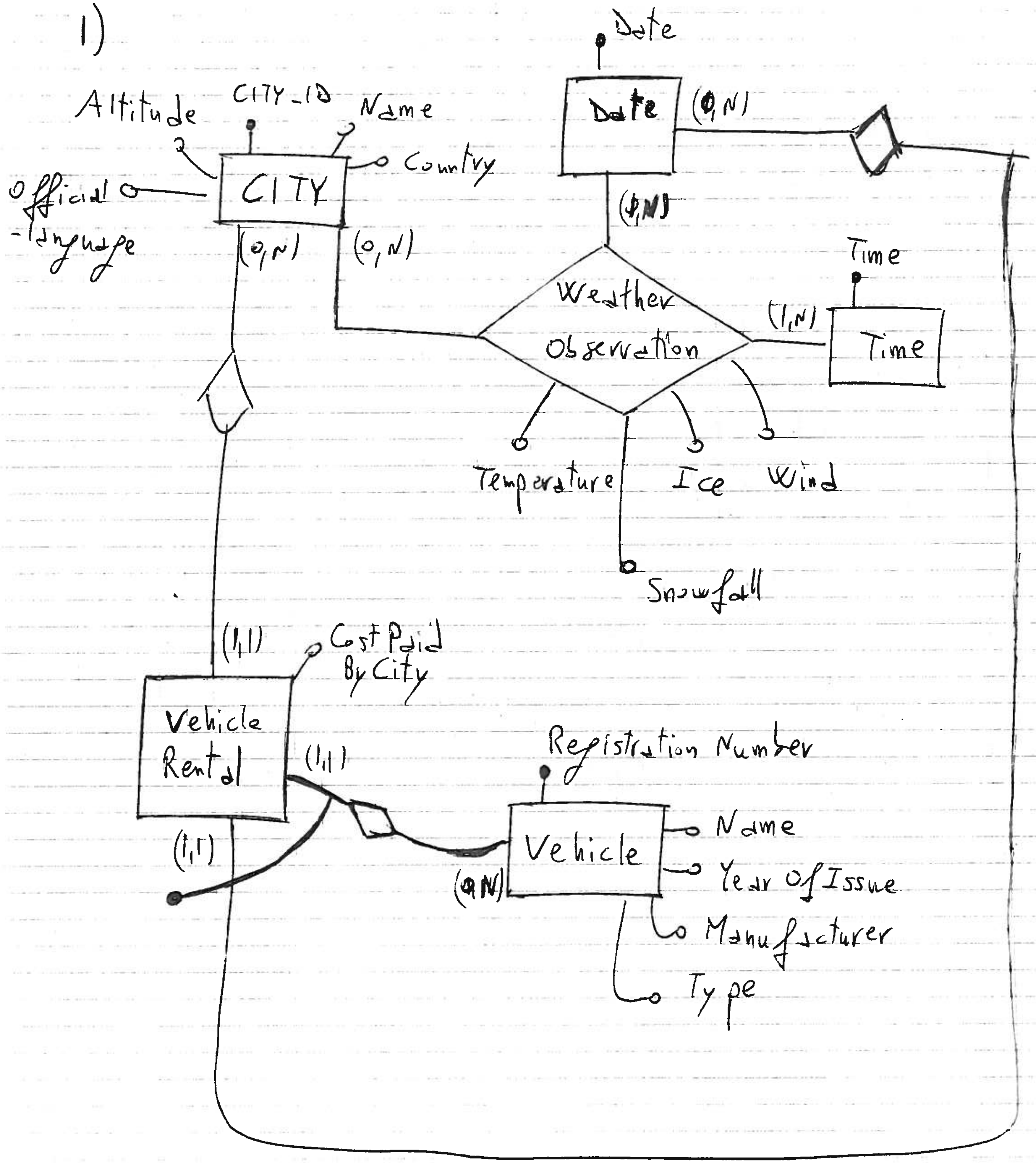
   a. Discover the fact(s) that are useful for the analysis of interest of *IceCompany*. For each of them:

i. Identify measures and dimensions (with their hierarchies) and produce the attribute tree (with pruning and grafting).
ii. Produce the conceptual schema (fact schema).

b. Produce a star schema or snowflake schema consistent with the conceptual schema and such that it allows performing the following queries:
   i. Considering only the winter season, find the name, the number of days with snowfall, and the number of days with the presence of ice for each Italian city.
   ii. Considering the first semester of year 2008, find for each city its name, the total cost for the rental of salt spreader vehicles, and the number of days in which the city rented salt spreader vehicles.
   iii. Find the French cities with the highest number of days with snowfall in the second three-month period of year 2010.
   iv. For each date of December 2009, find the country with the highest cost for snow removal.
   v. Find the Danish cities for which the cost for the rental of snow removal vehicles in year 2009 increased with respect to the cost of the previous year (i.e., with respect to year 2008).

c. Write the above queries in SQL.

1)



Altitude
CITY_ID
Name
Country
Official-language
CITY
(0,N)
(0,N)
Date
Date
(0,N)
(0,N)
Weather Observation
Time
(1,N)
Time
Temperature
Ice
Wind
Snowfall
(1,1)
Cost Paid By City
Vehicle Rental
(1,1)
(1,1)
Registration Number
Vehicle
(0,N)
Name
Year Of Issue
Manufacturer
Type

2.a) There are two facts

Fact: Weather Conditions

Measures: - Num Of Days With Ice
- Num of Days With Snowfall

Dimensions: - City
- Date

Fact: Vehicle Rental

Measures: - Cost
- Num Of Days With Rental

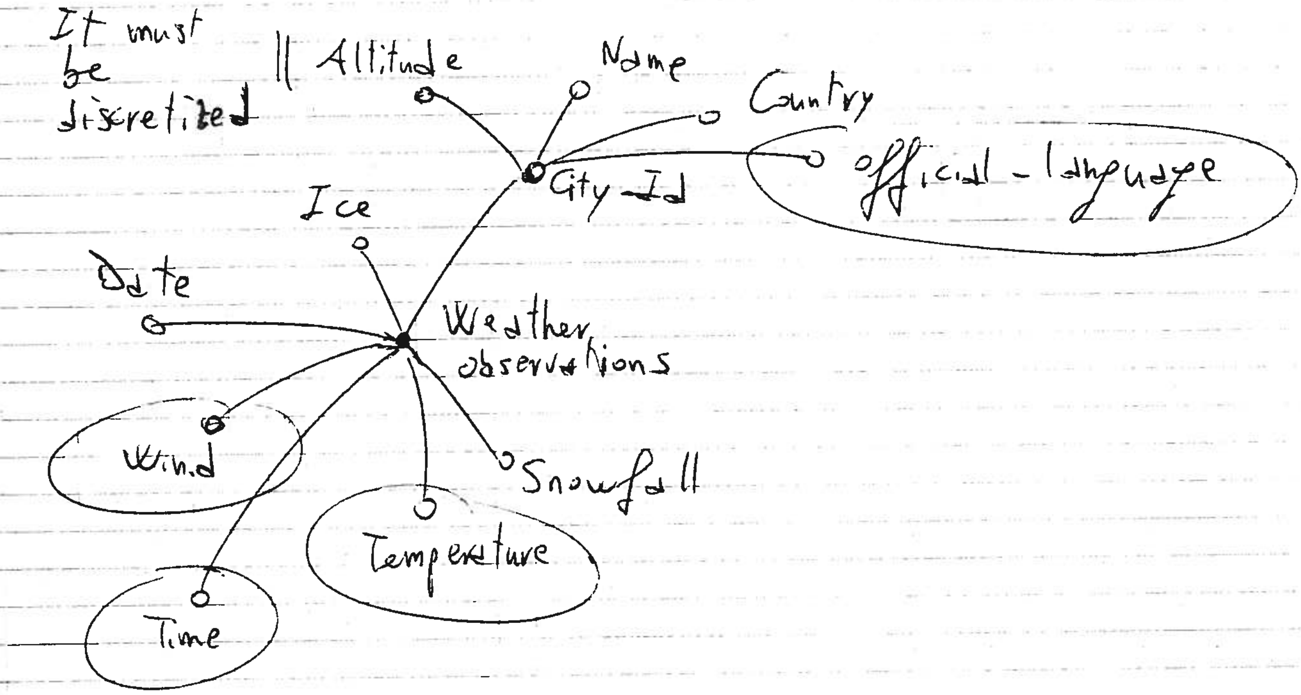
Dimensions: - City
- Date
- Vehicle Type

Attribute trees

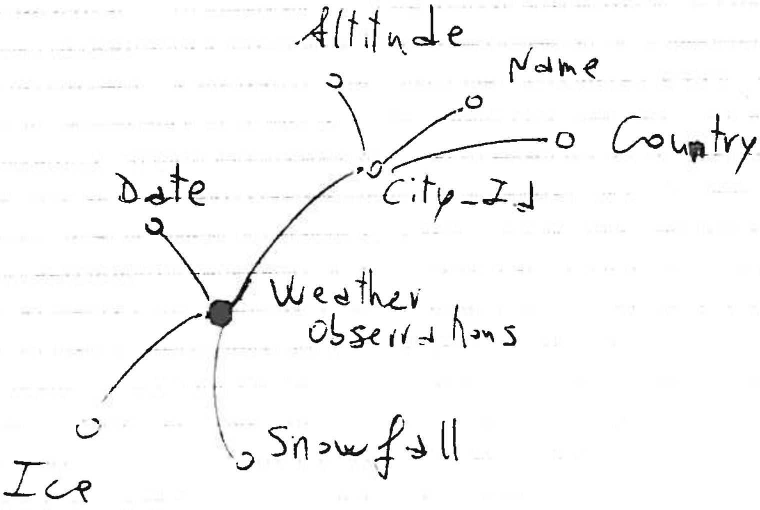
⬭ = pruning

Weather Conditions

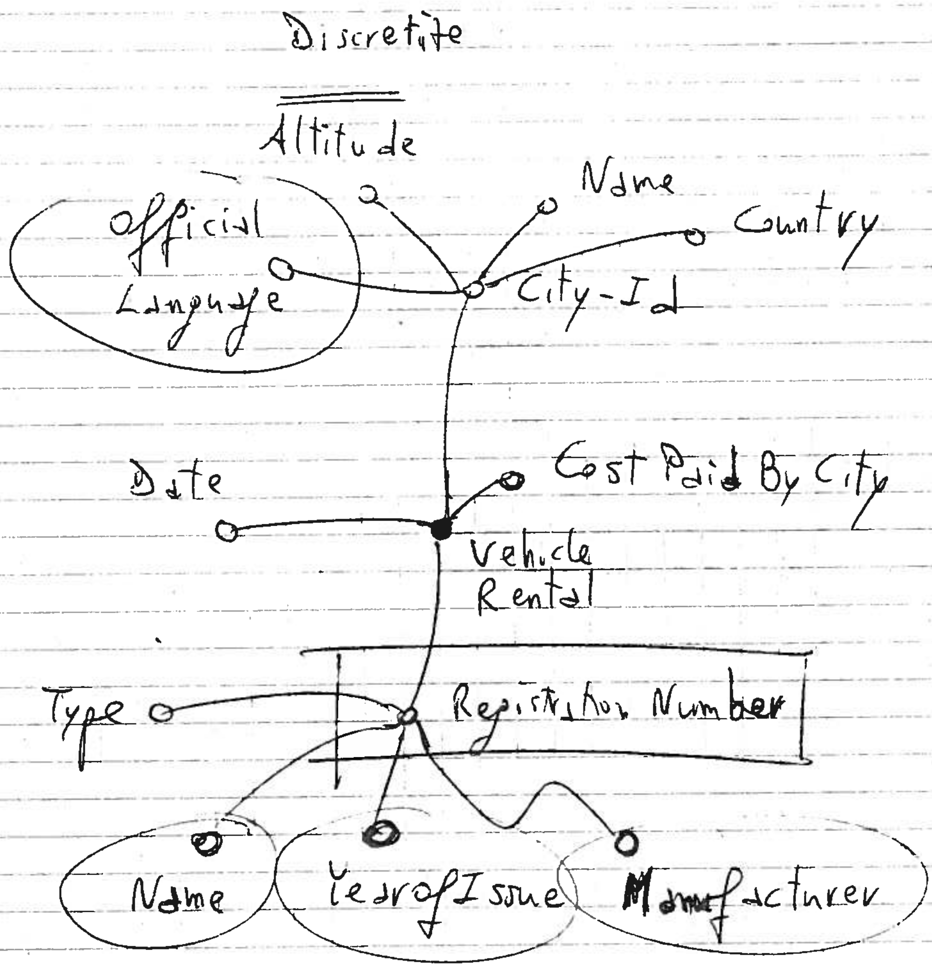
▭ = grafting


It must be discretized ||

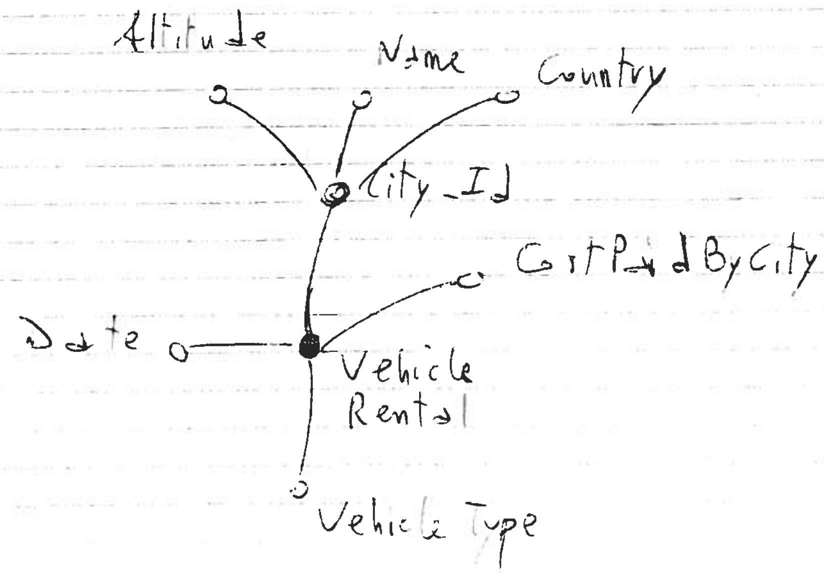
⇓ After pruning & grafting



Vehicle rental

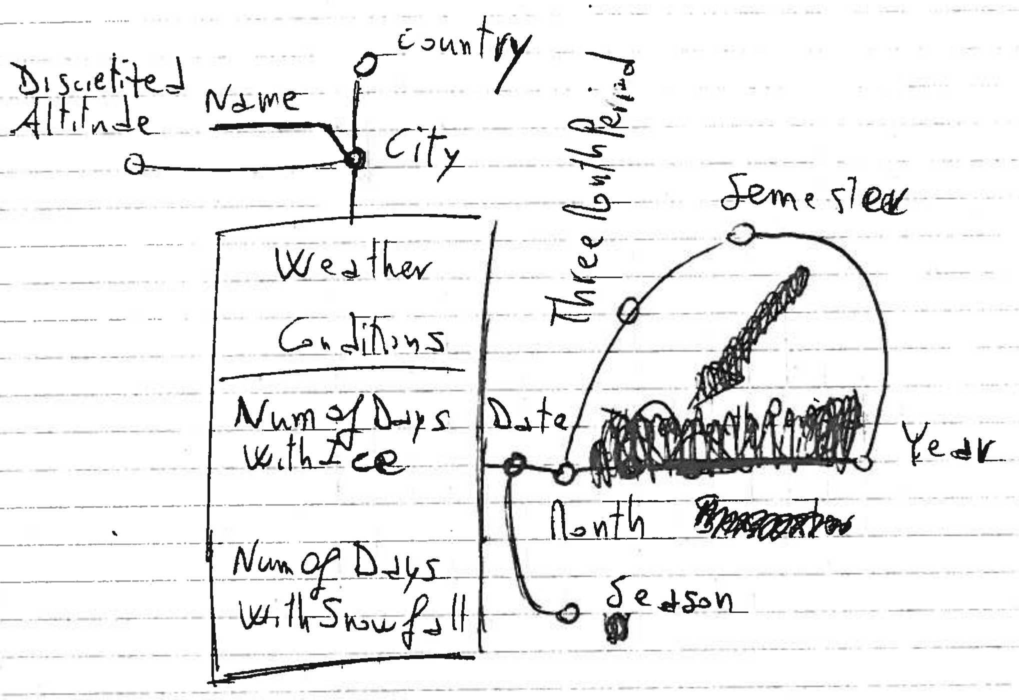


⇓ After Pruning & Grafting

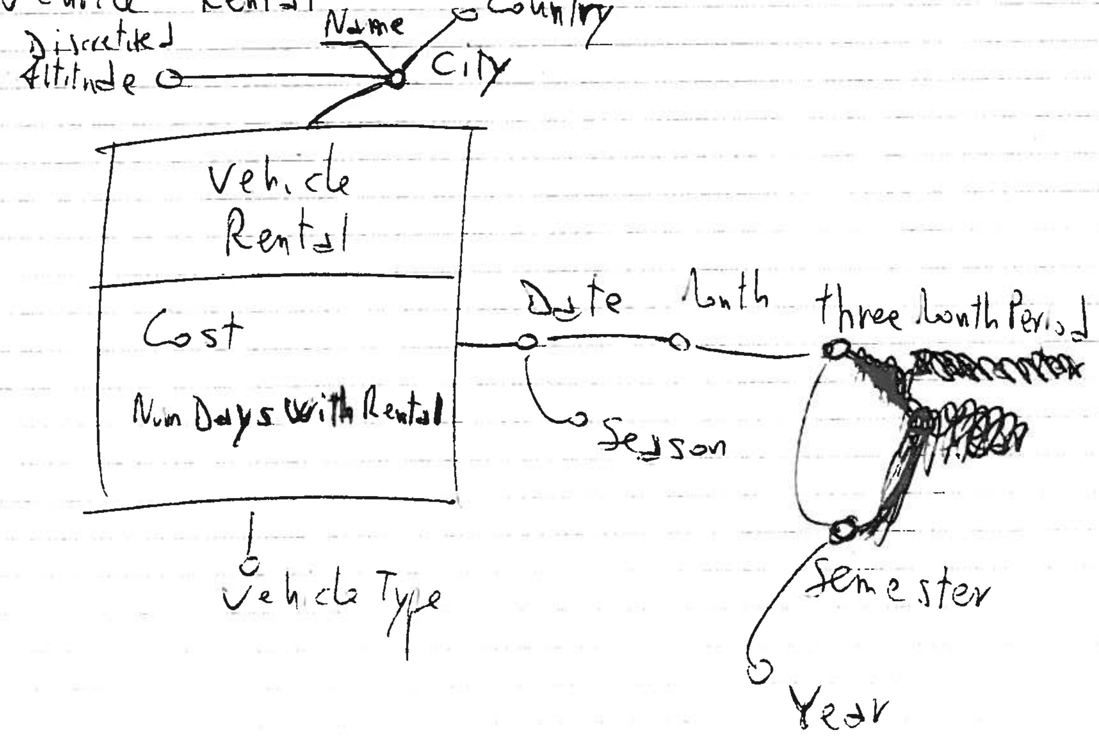


Fact schema = Conceptual model

Weather Conditions



Vehicle Rental



Logical star schema

The dimensions City and Date
are shared by the two facts

FACT WEATHERCOND (CityID, DateId, NumDaysIce,
NumDaysSnowfall)

DIMCITY (CityID, Name, Country, DiscrAltitude)

DIMDATE (DateId, Date, Season, Month, ThreeMonthPeriod,
Semester, Year);

FACTVEHICLERENTAL(CityID, DateId, VTypeId, Cost,
NumDaysRental)

DIMVEHICLETYPE(VTypeId, Type)

2. c

Query i

```
Select   Name, SUM(NumDaysSnowfall),
                SUM( NumDaysIce )
From     FACTWEATHERCOND  FW
         DIMDATE     D,
         DIMCITY     C
Where    FW.CityID = C.CityID
   AND   FW.DateId = D.DateID
   AND   D.Season = 'Winter'  AND  C.Country = 'Italy'
Group by  C.CityID, C.Name;
```

Query ii

```
Select   Name, SUM(Cost), SUM(NumDaysRental),
From     FACTVEHICLERENTAL  FV,
         DIMDATE    D,  DIMVEHICLETYPE VT, DIMCITY C
Where   FV.CityID = C.CityID
  AND    FV.VTypeId = VT.VTypeId
  AND    FV.DateId = D.DateId
  AND     VT.Type = 'Salt spreader'
  AND     D.Semester = 'Frst 2008'
Group by   C.CityID, C.Name;
```

Query iii

```
Create VIEW SNOWFALLSECOND AS
Select SUM(NumDaysSnowfall) TOTALSNOWFALL, CITYID
From FACTWEATHERCOND FW,
     DIMDATE D,
     DIMCITY C

where -join conditions-
AND  C.Country = 'France'
AND  D.ThreeMontPeriod = '2° 2010'
Group by C.CityID;

=

Select CityID, Name
From DIMCITY C, SNOWFALLSECOND S
where S.CityID = C.CityID
and S.TOTALSNOWFALL =
         ( Select MAX(TOTALSNOWFALL)
           From SNOWFALLSECOND S2 );
```

Query iv

```
Creat VIEW   COUNTRYDATECOST
 AS

Select   SUM(Cost) TOTALCOST, Country, Date
from     FACTVEHICLERENTAL  FV,
         DIMDATE   D,
         DIMCITY    C , DIMVEHICLETYPE   VT

Where    -join-
 AND  D.Month = 'December  2009'
  AND    VT.Type = 'Snow removal vehicle'
 Group by C.Country , D.Date ;

=
Select   Country, Date
FROM     COUNTRYDATECOST   CD1
WHERE    CD1.TOTALCOST =
              (SELECT  MAX(TOTALCOST)
               FROM COUNTRYDATECOST  CD2
               WHERE     CD1.Date = CD2.Date )
```

Query V

```
CREATE VIEW COST2008
AS
Select CityID, SUM(COST) TOTALCOST2008
from   FACTVEHICLERENTAL FV,
       DIMDATE D,
       DIMVEHICLETYPE VT,
       DIMCITY C
where  -join conditions-
AND    C.Country ='
AND    VT.Type ='Snow removal vehicle'
AND    D.Year = 2008.
Group by CityID;
=
CREATE VIEW COST2009 AS
Select cityID, SUM(COST) TOTALCOST 2009
from   FACTVEHICLERENTAL FV,
       DIMDATE D, DIMVEHICLETYPE VT,
       DIMCITY C
 where -join conditions-
 and   C.Country ='
 and   VT.Type = 'Snow removal vehicle'
 and   D.Year = 2009
 Group by cityID;
```

=

```
Select C.CityID,   C.Name
 from   DIMCITY  C,
        COST2008   C08
        COST2009   C09

 where   C. CityID = C08. CityID

  and    C09. CityID =  C08. CityID

  and      ~~[illegible]~~

 .C09. TotalCost2009 > C08. TotalCost2008;
```