

CS 6715: Functional - Logic Programming

Project Proposal



A MEDIATOR FOR LEISURE TIME SERVICES

Instructor: Dr. Prof. Harold Boley

Students:

Le Thi Thu Thuy

Han Liang

Xiao Hui Zhou

Yong Liang

March 16, 2006

1. Motivation

Searching leisure time activities from the internet is time-consuming because:

- There are many leisure time services on the internet.
- Each leisure time service might provide different kinds of services, for example, some services can provide golf, fishing services while others can provide swimming, fitness services, etc.
- Different leisure time services have different regulations in computing prices.
- Different leisure time services have different input parameters and different output formats.

Therefore, our group decided to design and implement a mediator which could help users to search for leisure time activities over several services on the internet.

2. Background

- *Domain of the project:* Leisure time activities.
- *Tools:* Relfun, a functional-logic language developed for declarative programming.
- *Techniques:* recursion, non-ground structures, non-determinism.

3. Description of Problem

Consider that Bill is free on Saturday, and he plans to play a bowling game with his friends. Before making any decision, he searches leisure time activities from the internet in order to find a good place to go with his friends. The followings are requirements of Bill:

1. A bowling game for 12 people
2. Cost should be lower than \$300 for all of them

The query result should include: when and where they can go, and how much they have to pay. The result should return in the order of prices.

We will use a soft-constraint-like matching in our system, which means if no result can satisfy all above conditions, the system can also try to provide some close results according to users' strategies. For example, if the customer thinks the price is more important, the system will consider condition 2 and change to another game.

4. Possible Methodology

4.1 Tasks of the service engine

The engine must have the following abilities:

- Receive a query from a user
- Decompose this query into sub-queries, and each sub-query is used for a specific company
- Find information from local companies
- Compute prices and discounts depending on regulations of service companies
- Return search results
- If no result is returned, a consultant of the system will return other possible results if the user needs.

Figure 1 shows a service engine, acting like a mediator. In this case, this mediator is built for three companies A, B and B. It is used to decompose a user query into three sub-queries for these three companies.

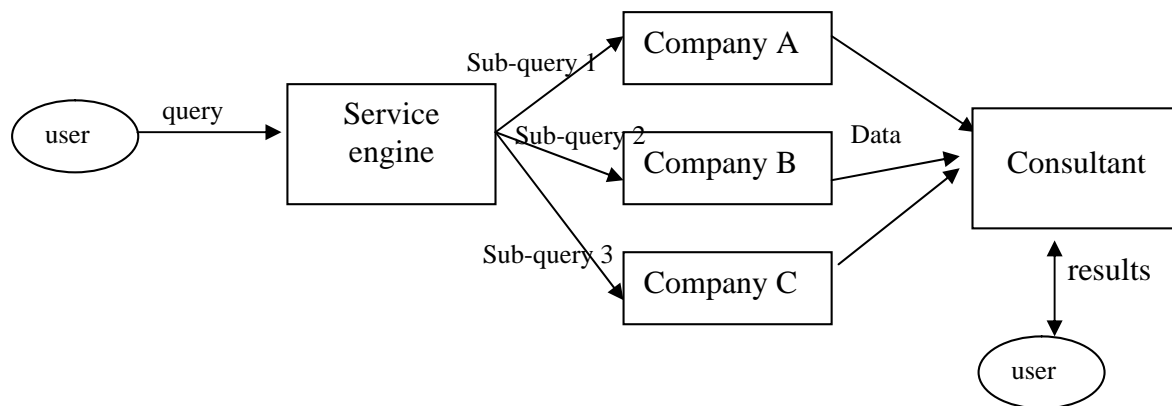


Figure 1. A mediator architecture for leisure time activities

4.2 User input query

In order to find leisure services efficiently, users have to show their identity information (e.g., userID, user name). They also have to indicate the name of leisure services they want to participate, dates and a number of people expecting to enjoy the services. An example of an input of a user is shown as follows:

leisureServiceQuery("0123", "Bowling", 12, "2006-09-09", CompanyName, Where, Price)

The engine will divide this input into the two following parts:

An internal global query:

ask4Services(CompanyName, Where, "bowling", 12, "20060909",Price)

A customer query: (this query will be sent to service companies)

customerQuery("0123", CompanyName)

An example of three decomposed queries of the global query is shown as follows:

ask4AServices(ComanyName, Place, "bowling", 12, "20060909",ServicePrice)

ask4BServices(Address, "bowling", 12, "20060909",Money)

ask4CServices(Name, Location, "bowling", 12,"20060909", Cost)

4.3 Facts

We assume that a company has its own database and can manage one or more leisure time services. Besides, different companies have different data structure formats. The followings list examples of data structures of companies A, B and C.

1. Company A:

db1(Name, Location, ContactInfo, NameOfLeisure, Price).

2. Company B:

db2(Name, Telephone, Address, Links, TypeOfLeisure, Price).

3. Company C:

db3(CompanyName, Where, Phone #, Website, Event Support, Cost).

Companies' databases (i.e., facts) are built followed data structures of companies' schemas. Besides the above facts, each company also has other facts (e.g., customer information, dates of discounts of a company). These facts will help that company to improve their services such as having a discount for a VIP customer.

- Facts of customers

These facts must include Customer ID, Customer Name, How much money they spent last year in company, Customer's address, Which company they belongs to.

customer (ID, Name, Money, Address, Company).

- Facts of holidays

holiday(Date).

- Facts of users' priorities

In these facts, we will define what users prefer if they do not get their results by their constraints in their inputs. For example, h(X) is defined, where X representing the users' requirements.

While each company has its own database and data structure, the mediator works based on a global structure. For example:

globalschema(Name, Location, ContactInfo, Website, NameOfLeisure, Price).

4.4 Rules

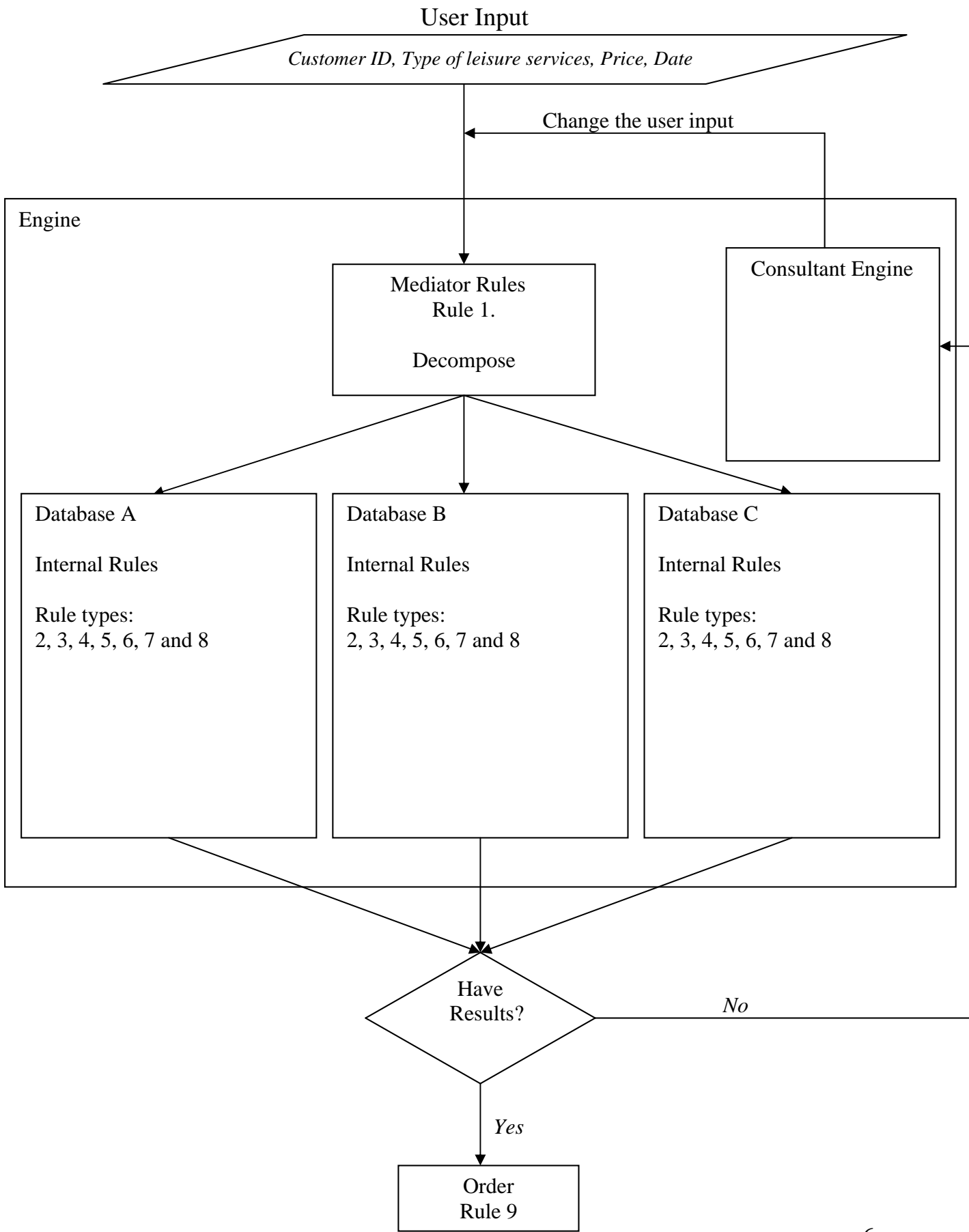
Every process of our system must be performed by using rules written in the form of Relfun. Some kinds of rules of the system are shown as follows:

1. Mediator rules used to transform input queries into sub queries for companies
Input: a customer's query
Output: different sub-queries for corresponding companies
2. Get Price of a specific leisure service in a company
Input: a leisure service type, a company name, a user ID, a date
Output: a price including discounts
3. Get holiday discount rate
Input: a leisure date
Output: a discount rate

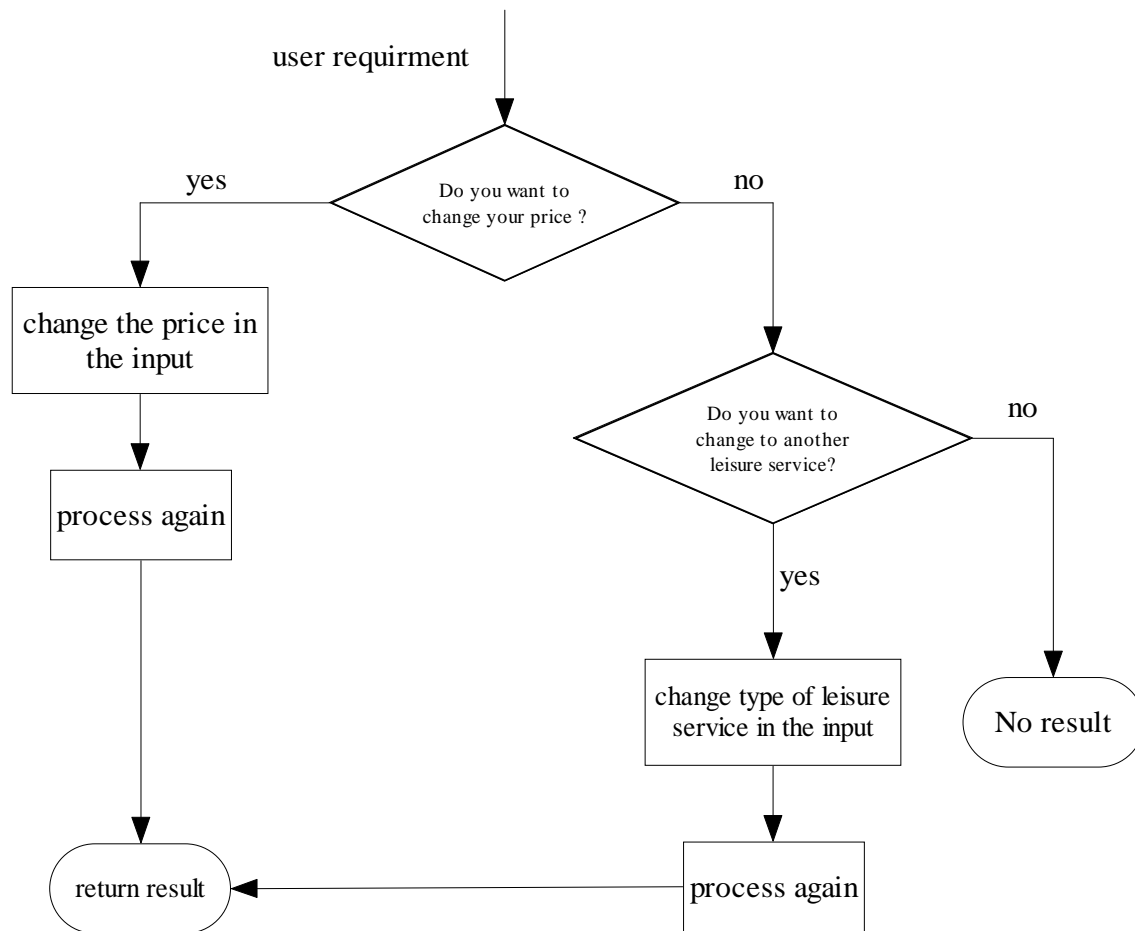
Example:

```
getHolidayDiscount(Date):-holiday(Date)!&0.1.  
getHolidayDiscount(Date):&0.
```

4. Get a discount rate of a group of participant
Input: a company name, Type of leisure service, Number of participants
Output: a discount rate
5. Get discount rate of membership
Input: a membership type
Output: a discount rate
6. Check if a customer is VIP member or a regular member
Input: a customer ID
Output: a membership type
7. Precise search
Input: a sub-query for a company
Output: Company Name, Address, Price
8. Advanced search
Input: a sub-query for a company
Output: Company Name, Type of leisure, Address, Price
9. Order the result
Input: a list
Output: an ordered list



Consultant engine



References

1. Felix Naumann. Quality-Driven Query Answering for Integrated Information System. Springer Book (2002) LNCS2261.
2. Harold Boley. A Tight, Practical Integration of Relations and Functions. Springer Book (1999) LNAI1712.
3. Harold Boley. Lecture Notes. CS6715: Functional and Logic Programming.
4. UNB Recreation Inc. <http://www.unbf.ca/CampusRec/index.html>
5. Royal Oaks country Club. http://www.royaloaks.nb.ca/welcome_en.htm
6. Kingswood Park. <http://www.kingswoodpark.com>.

Appendix

% data of company A.

% Database 1(Name, Location, ContactInfo, NameOfLeisure, Price)

db1("UNB Recreation Inc",
"UNB LB Gym",
"1-506- 453-4579",
"Yoga", 30).

db1("UNB Recreation Inc",
"UNB LB Gym",
"1-506- 453-4578",
"Strength Training", 40).

db1("UNB Recreation Inc",
"UNB LB Gym",
"1-506- 453-4578",
"Dancing Class," 50).

db1("UNB Recreation Inc",
"UNB LB Gym",
"1-506- 453-4522",
"Swimming", 20).

db1("UNB Recreation Inc",
"UNB LB Gym",
"1-506- 453-4533",
"Karate Class", 60).

db1("UNB Recreation Inc",
"UNB LB Gym",
"1-506- 453-4545",
"Climbing Club", 40).

db1("UNB Recreation Inc",
"UNB LB Gym",
"1-506- 453-4545",
"Badminton Club", 10).

db1("UNB Recreation Inc",
"UNB LB Gym",
"1-506- 453-4545",
"Squash", 20).

% data of company B.

% Database 2(Name, Telephone, Address, Links, TypeOfLeisure, Price)

db2("Royal Oaks country Club",
"1-866-769-6257",
"1746 Elmwood Drive Moncton, New Brunswick E1H 2H6",

“http://www.royaloaks.nb.ca/welcome_en.htm”,
“Golf”, 60).

db2(“Royal Oaks country Club”,
“1-866-769-6252”,
“1746 Elmwood Drive Moncton, New Brunswick E1H 2H6”,
“http://www.royaloaks.nb.ca/welcome_en.htm”,
“Swimming”, 20).

db2(“Royal Oaks country Club”,
“1-866-769-6253”,
“1746 Elmwood Drive Moncton, New Brunswick E1H 2H6”,
“http://www.royaloaks.nb.ca/welcome_en.htm”,
“Dinner”, 30).

db2(“Royal Oaks country Club”,
“1-866-769-6254”,
“1746 Elmwood Drive Moncton, New Brunswick E1H 2H6”,
“http://www.royaloaks.nb.ca/welcome_en.htm”,
“Biking”, 20).

db2(“Royal Oaks country Club”,
“1-866-769-6255”,
“1746 Elmwood Drive Moncton, New Brunswick E1H 2H6”,
“http://www.royaloaks.nb.ca/welcome_en.htm”,
“Hotel”, 100).

db2(“Royal Oaks country Club”,
“1-866-769-6256”,
“1746 Elmwood Drive Moncton, New Brunswick E1H 2H6”,
“http://www.royaloaks.nb.ca/welcome_en.htm”,
“Tennis”, 20).

db2(“Royal Oaks country Club”,
“1-866-769-6233”,
“1746 Elmwood Drive Moncton, New Brunswick E1H 2H6”,
“http://www.royaloaks.nb.ca/welcome_en.htm”,
“Hunting”, 50).

db2 (“Royal Oaks country Club”,
“1-866-769-6223”,
“1746 Elmwood Drive Moncton, New Brunswick E1H 2H6”,
“http://www.royaloaks.nb.ca/welcome_en.htm”,
“Kayaking”, 30).

% data of company C.
% DB3 (CompanyName, Where, Phone #, Website, Event Support, Cost)

db3(“Kingswood Park”,
“31 Kingswood Park Fredericton, New Brunswick E3C 2L4”,
“1-506-444-9503”,
“<http://www.kingswoodpark.com/>”,
“Bowling”, 20).

db3(“Kingswood Park”,
“31 Kingswood Park Fredericton, New Brunswick E3C 2L4”,
“1-506-444-9503”,

“http://www.kingswoodpark.com/”,
“Fitness”, 10).

db3(“Kingswood Park”,
“31 Kingswood Park Fredericton, New Brunswick E3C 2L4”,
“1-506-444-9503”,
“http://www.kingswoodpark.com/”,
“Cinema”, 10).

db3(“Kingswood Park”,
“31 Kingswood Park Fredericton, New Brunswick E3C 2L4”,
“1-506-444-9503”,
“http://www.kingswoodpark.com/”,
“Laser game”, 30).

db3(“Kingswood Park”,
“31 Kingswood Park Fredericton, New Brunswick E3C 2L4”,
“1-506-444-9503”,
“http://www.kingswoodpark.com/”,
“Golf”, 60).

db3(“Kingswood Park”,
“31 Kingswood Park Fredericton, New Brunswick E3C 2L4”,
“1-506-444-9503”,
“http://www.kingswoodpark.com/”,
“Tennis”, 20).

db3(“Kingswood Park”,
“31 Kingswood Park Fredericton, New Brunswick E3C 2L4”,
“1-506-444-9503”,
“http://www.kingswoodpark.com/”,
“Dinner”, 30).

db3(“Kingswood Park”,
“31 Kingswood Park Fredericton, New Brunswick E3C 2L4”,
“1-506-444-9503”,
“http://www.kingswoodpark.com/”,
“Kid Zone”, 10).

% Customer data.

% Customer (ID, Name, Money, Address, Company)

customer (“003”, “Allen Iverson”, 100, “8 Wilmot Street Fredericton”, “UNB Recreation
Inc.”).

customer (“003”, “Allen Iverson”, 250, “8 Wilmot Street Fredericton”, “Royal Oaks
Country Cub”).

customer (“003”, “Allen Iverson”, 150, “8 Wilmot Street Fredericton”, “Kingswood Park”).

customer (“056”, “Han Liang”, 360, “602 Graham Ave.”, “Kingswood Park”).

customer (“032”, “Jason Kidd”, 80, “475 Beaverbrook Street Fredericton”, “UNB
Recreation Inc.”).

customer (“032”, “Jason Kidd”, 180, “475 Beaverbrook Street Fredericton”, “Royal Oaks
Country Club”).

customer (“032”, “Jason Kidd”, 130, “475 Beaverbrook Street Fredericton”, “Kingswood
Park”).

customer (“020”, “Nicolas Cage”, 80, “155 Regent Street”, “Royal Oaks Country Club”).

customer (“021”, “Tim Duncan”, 120, “321 Main Street Fredericton”, “UNB Recreation

Inc.”).
customer (“021”, “Tim Duncan”, 500, “321 Main Street Fredericton”, “Royal Oaks
Country Club”).
customer (“021”, “Tim Duncan”, 240, “321 Main Street Fredericton”, “Kingswood Park”).
customer (“008”, “Tony Parker”, 150, “111 Goodine Street Fredericton”, “UNB Recreation
Inc.”).
customer (“008”, “Tony Parker”, 600, “111 Goodine Street Fredericton”, “Royal Oaks
Country Club”).
customer (“008”, “Tony Parker”, 300, “111 Goodine Street Fredericton”, “Kingswood
Park”).
customer (“006”, “Tommy Jiang”, 800, “15 Woodbridge Street”, “UNB Recreation Inc.”).
customer (“007”, “Yong Liang”, 500, “602 Graham Ave.”, “UNB Recreation Inc.”).

% Holiday(Date)

holidayA("2006-01-01").
holidayA("2006-02-14").
holidayA("2006-03-15").
holidayA("2006-07-01").
holidayA("2006-11-01").
holidayA("2006-10-01").
holidayA("2006-05-01").
holidayA("2006-01-14").
holidayA("2006-02-12").
holidayA("2006-08-01").
holidayA("2006-09-09").
holidayA("2006-12-25").

holidayB("2006-01-01").
holidayB("2006-02-14").
holidayB("2006-03-15").
holidayB("2006-07-01").
holidayB("2006-11-01").
holidayB("2006-10-01").
holidayB("2006-05-01").
holidayB("2006-01-14").
holidayB("2006-02-12").
holidayB("2006-08-01").
holidayB("2006-09-09").
holidayB("2006-12-25").

holidayC("2006-01-01").
holidayC("2006-02-14").
holidayC("2006-03-15").
holidayC("2006-07-01").
holidayC("2006-11-01").
holidayC("2006-10-01").
holidayC("2006-05-01").
holidayC("2006-01-14").
holidayC("2006-02-12").
holidayC("2006-08-01").
holidayC("2006-09-09").
holidayC("2006-12-25").

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.