Submission Instructions:

Due Date: St. George Tues, Jan. 29, 6:15 p.m.

UTSC Wed Jan. 30, 7:15 p.m.

- Answer **both** of the questions provided in the assignment.
- All assignments must be submitted **electronically**.
- Assignments that are "on-time" must be handed-in in class by the date and time that they are due.
- Assignments not handed-in within the first five minutes of lecture are considered late, and will not be accepted later in class.
- The assignment must be **typed.** Acceptable electronic formats for submission are PDF and Postscript. Any other format will not be marked.
- If you are working in a team of two, submit only **one** assignment. (If two assignments are submitted, the assignment with the lower mark will be considered.)
- Include a cover page. A template for the cover page is available on the Assignment One web page.
- All pages must be **numbered** and **stapled** together. Do not put them in an envelope. Attach all pages well. Only those pages actually stapled to the front page will be marked – the markers cannot be responsible for pages that go astray. For this reason – don't use a paperclip, fold your pages together at the corner, or attach them by any other creative means.
- Late assignments must be emailed to your instructor (either 343@cs.toronto.edu, Professor Miller; or faye@cs.toronto.edu. Professor Baron).
- Late assignments must be emailed to your professor. 25% (absolute value) will be deducted for every day the assignment is late — to a maximum of two days.
- Assignments cannot be more than two days late – assignments received after 6:15p.m. Thursday (for St. George Students) and 7:15 Friday (for UTSC Students), will not be considered.
- There will be a 24 hour blackout prior to the first due date. This means that as of 6:15 p.m. on Monday, January 28th, any questions regarding this assignment will not be answered. The exception to this is if there is some kind of technical issue you may have with your electronic submission.

Question 1 – The Relational Model and DDL [25 marks]

GrocerEase.com is an online service that delivers groceries in the GTA (Greater Toronto Area). They are building a database to manage their business. They must capture the following kinds of information:

- GrocerEase sells to individual Customers. Each customer has a unique customer ID, customer name, customer address, postal code, and telephone number.
- Orders are placed. Every order has a unique order number; an order date; a delivery date; a delivery window; the customer; the payment ID for that order; the picker who packed the order; and the order status. The order status indicates whether the order is placed, canceled, packed for shipment, en-route, or delivered.
- Payment information includes the payment ID number which uniquely identifies that payment record; the credit card type, number, and expiry date; and the owner of the credit card. The company accepts three cards: American Express, MasterCard, and Visa (AX, MC and VS).
- There are three delivery windows: morning from 8 to 12 a.m.; afternoon from 12 – 5 p.m.; and evenings from 5 – 9 p.m. Customers identify the time window in which they wish to have their groceries delivered on delivery day.
- The company keeps a store of products for sale. Each product has a product name; unique product ID; a product measure; a sale price (per measure); a cost price; an inventory quantity; a quantity on-order; a vendor ID; and a temperature indicator. The product measure can be kilos, or units. The temperature indicator specifies if the product is normally kept at room temperature, refrigerated or frozen.
- The orders consist of one or more order items. Each order item indicates the product and the quantity of that product that has been ordered. When an order is removed, all items belonging to that order are also removed.
- Delivery routes are assigned to a specific truck. Each delivery route includes the date and delivery window of the shipment; and the number of the truck to which it is assigned for shipment.
- One or more postal codes are assigned to a delivery route. All orders for that postal code that must be delivered during the delivery date and window specified for the delivery route will be included in that delivery route shipment.
- The company has several delivery trucks in their fleet. The trucks have a truck number, for easy identification, a license number, a license-renewal date. The fleet is well-maintained. Every truck is inspected on a specific day of the month (the same day each month for that truck.) The truck maintains a status indicating whether it is "out of service" or "available". As well, the truck mileage is recorded at the end of each day.
- The company has several types of employees including drivers and pickers. For each employee, the company keeps their first and last name, their address, city, and postal code, social insurance number and their hourly pay-rate. For security reasons, they do not use the social insurance number to identify the employees – but rather, their employee id number.
- Drivers deliver the orders. Driver shift information includes the driver employee ID, the day, the truck number to which they are assigned, the start-time and the end-time of their shift.
- Pickers are workers who pack an order into bins and load them onto the appropriate truck. The picker shift information includes the picker employee ID, the start-time, and the end-time of their shift. (The orders which they have packed are tracked with the order information.)
- The company has streamlined their purchasing and has negotiated a purchase price through a single vendor for each product. (There may be more than one product sold by a single vendor, but only one vendor from which GrocerEase purchases a single product.) Vendor information includes the vendor name; contact name; address; city; province; postal code; telephone number; the unique ID which identifies the vendor; and the volume-purchase quantity (a dollar value) and discount rate. When they exceed the volume-purchase quantity threshold in a single order, their purchase is discounted by the discount rate.

Define a relational schema that can accommodate the preceding information **in DDL**. Remember: Enforce all key and integrity constraints.

- Provide default values where appropriate.
- Avoid duplicating information (except for foreign keys used to link relations.)
- Indent your DDL statements consistently for readability.
- You may make text fields any (reasonable) size you want.
- If you are using a descriptive field that occurs in more than one tuple, you may want to encode it in a separate table.

Question 2 – Relational Algebra [35 marks]

For this problem we use the schema of Musselman's Fitness Club database. It includes the following relations. Keys are underlined and each tuple is described:

- Member(memberID, name, address, telephone) A member is either an active member or a potential member (i.e., they have contacted the club sales representatives and asked for information). Member information includes the member name, address and phone number.
- Employee(employeeID, type, name, address, start-date) An employee including their employee number, name, address, start-date and employee type. Employee types include values such as "personal trainer", "desk clerk", "fitness instructor", "sales representative", and "maintenance".
- Court(number, type, description) Every court in the club is uniquely identified by a number, has a detailed location description. The court type, a value of *S, R,* or *T,* identifies the court as being a squash, racketball, or tennis court respectively.
- CourtReservation(court, date, hour, member) A court may be reserved for an hour (on the hour), each reservation includes the court number, date and hour of the reservation, and the ID of the member who is reserving the court.
- MembershipType(typeCode, description, visitLimit, duration, price) there are three membership types, a trial membership which has a visitLimit which limits the number of visits to 3 and is offered at no cost by the sales reps to entice new members; a 10 visit membership with no fixed duration; and a yearly membership, which has no visit limit and the duration set to one year.
- Membership(membershipID, member, type, startDate, endDate, salesRep) Members have memberships which are identified by membershipID. The type of membership is identified by type, as well the memberID, the date the membership starts and ends as well as the sales rep who is credited for the membership are captured.
- ClassType(typeCode, description) Fitness class types are assigned a unique code and described. For example, the club offers "tummy tightener", "stretch and flexibility", "step", "kickboxing", and "spinning" classes. A full description of each class type is provided.
- Class(classID, type, level, instructor, startDate, endDate, dayOfWeek, hourOfDay, capacity) Each class scheduled is assigned a unique classID. Each class has a specific class type and occurs at a specific hour on a specific day of the week (1 – 7) . There are currently three class levels, *beginner*, *intermediate* and *advanced*. An instructor is assigned to teach that class. As well, the start and end dates, indicating when the class is offered, is captured. The capacity is the maximum number of members that may join each class. Classes start at the beginning of each session. Typically there are four sessions (spring, winter, fall, and

summer). So, for example, all classes starting for Fall, 2008 begin on the same day, with the same startDate though they may not actually occur til later in the week.

- ClassReservation(classID, member, date)Every time a member attends a class, he must reserve. This ensures that the classes aren't overcrowded. The id of the Class and member as well as the date they are reserving for, are captured.

The following inclusion dependencies (foreign keys) hold:

CourtReservation(court) $\subseteq$ Court(number)

CourtReservation(member) $\subseteq$ Member(memberID)

Membership(member) $\subseteq$ Member(memberID)

Membership(salesRep) $\subseteq$ Employee(employeeID)

Membership(type) $\subseteq$ MembershipType(typeCode)

Class(type) $\subseteq$ ClassType(typeCode)

Class(instructor) $\subseteq$ Employee(employeeID)

ClassReservation(ClassID) $\subseteq$ Class(classID)

ClassReservation(member) $\subseteq$ Member(memberID)

Write the following queries in Relational Algebra. You are not allowed to use the renaming of relations as described in the text though you may create interim tables as shown in tutorial, nor can you use the extended relational algebra. You may use renaming of attributes, as well as numerical and date comparisons (e.g., R.A > 5, or R.A $\geq$ YYYY-MM-DD). You may specify a date as YYYY-MM-DD and time as HH (using a 24 hour clock). Also assume that you can subtract one date from another to get the number of days. So if a first date minus a second date equals 5, then the first date is five days after the second date. Use *CurrentDate* to represent today's date and be sure to clearly state any assumptions that you make.

1. Find the name and telephone numbers of all members who reserved both a "squash court" and a "tennis court" on the same day.
2. Find the names all of the instructors who have ever taught a "tummy tightener" class in the past but are not currently teaching it.
3. Find the first name of the sales representative who did not sell a membership in December but sold at least two memberships in January. The membership starts on the day it is sold. (Hint: to sell a membership, the membership price must be greater than zero – therefore, trial memberships are not considered.)
4. Find the name of the fitness instructor who is currently teaching both a "spinning" and a "step" class.
5. Find the name and telephone numbers of all members who played squash and took a spinning class on the same day.
6. Find the number and description of all squash courts that were not reserved from January 10, 2008 to January 12, 2008.
7. Find the name of the instructors who taught the first classes the first week that they were ever offered. (Hint: start-date identifies the week when a class that is offered starts)