Week 2 - The Relational Data Model

Part A. The Relational Model

1. Discuss the following terms:

- Relation
- Attribute
- Domain
- Tuple
- Degree and Cardinality of a Relation

2. Consider the CUSTOMER and ORDER relations below:

CUSTOMER (CUSTOMER-ID, NAME, ADDRESS)
ORDER (ORDER-ID, DATE, CUSTOMER-ID)

Assume a single customer may have any number of orders.

(a) Identify the primary key and foreign key attributes in these relations.

(b) Can you think of a reason why we would not just store all the customer and order information in one relation so that we would not have to perform the join operation?

3. Choosing Primary Key.

(a) In any relation, tuples must be unique. However, in many cases the set of all the attributes in a relation is not considered a candidate key. Why not?

On the other hand, suppose we do have a relation where the set of all attributes is a candidate key. In this case, show that this set must therefore be the only candidate key and hence the primary key.

(b) Consider a relation that depicts a tutorial room booking in a university. Each faculty assigns a person to handle the booking for all tutorial classes for that faculty. The person's email address is given to the university's booking system as a contact person.

BOOKING(b_date, b_starttime, b_endtime, unit_code, contact_person, room_no, tutor_id)

(i) Identify candidate key(s) and primary key for the relation if the following business rules are applicable:

   a. More than one tutorial classes of the same unit may run at the same time (parallel sessions are possible).

   b. A tutor may teach several classes of the same unit.

   c. All tutorial classes are 2 hours long.

(ii) Identify candidate key(s) and primary key for the relation if the following business rules are applicable:

   a. Tutorial classes can be either 1 hour or 2 hours long.

   b. A tutor can only teach one tutorial class in a given unit.

c. There are no parallel sessions of tutorial classes.

The following exercises (questions 4 and 5) in Relational Algebra are optional. It is not examinable. However, if you are interested to understand the relational model theory in depth, these exercises will give you good foundations.

4. (Adapted from Exercise 3.6 of Connolly, Begg and Strachan)

Suppose we have the following 4 relations:

HOTEL(HOTEL-NO, NAME, ADDRESS)
ROOM(ROOM-NO, HOTEL-NO, TYPE, PRICE)
BOOKING(HOTEL-NO,GUEST-NO, DATE-FROM, DATE-TO, ROOM-NO)
GUEST(GUEST-NO, NAME, ADDRESS)

Generate the relational algebra for the following queries:

(a) List the names and addresses of all hotels
(b) List all single rooms with a price below $50
(c) List the names and addresses of all guests
(d) List the price and type of all rooms at the Grosvenor Hotel
(e) List all names and addresses of guests currently staying at the Grosvenor Hotel (assume that if the guest has a tuple in the BOOKING relation, then they are currently staying in the hotel)

5. In the lecture we have discussed 7 relational algebra operators, namely:

SELECTION, PROJECTION, JOIN, UNION, INTERSECTION, DIFFERENCE and CARTESIAN PRODUCT. In fact, 5 of these operators may be considered primitive operators in the sense that the others may be expressed in terms of the primitive operators. The primitive operators are:

SELECTION, PROJECTION, UNION, DIFFERENCE and CARTESIAN PRODUCT

Using the sample tables below, show how the JOIN operation can be expressed in terms of the fundamental operators by showing the process to do a natural join of customer and order.

– **CUSTOMER table:**

Cust_ID	Name
1	Green
2	Blue

ORDER table:

Ord_ID	Date	Cust_ID
1	23-Feb-2009	1
2	26-Feb-2009	1
3	26-Feb-2009	2

Part B. Practical work – The SQL Developer

Swimming Club Event Entry System

Create three tables as detailed below and load them with sample data using SQL insert commands - again we are ignoring the issues of entity and referential integrity here (ie. do not worry about primary and foreign keys, we will return to this later). This week's exercise is about further developing your familiarity with SQL Developer.

a. SWIMMER

Attributes:

- Swimmer registration code eg. CA101 (maximum of 5 characters)
- Name (maximum of 20 characters)
- Date of Birth
- Swimming Club Name (max 15 characters)

b. EVENT

Attributes:

- Event Number 1 .. 50
- Event Description (maximum of 20 chars)

c. ENTRY

Attributes:

- Swimmer registration code eg. CA101 (maximum of 5 characters)
- Event Number 1 .. 50
- Entry time (m.sshh) eg. 0.3506 (store as a number)
- Final time (m.sshh) eg. 0.3506 (store as a number)

Final time (m.sshh) - will not be known until after the race has been swum (in the insert file put the word null for this value) - ie. as an example: insert into ENTRY values ('CA101',5,0.3506, null)

You should create **three separate text files** and test run them within SQL

- a schema file called swimschm.sql to create the tables (swimmer, event and then entry)
- a file called swimload.sql containing several insert statements to load data into all three tables, and
- a file containing drop table statements (eg. drop table swimmer purge) to drop all.

Where multiple SQL statements exist in one file, be sure to terminate each with a semi-colon (;).

You can check the contents of your loaded tables by using the SQL select statement eg. select * from swimmer

Your tutor will check these files and watch you do a sample setup of your Swimming database.