ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**Database Management System**

**Theory Lab Assignment #7**

**SUBMITTED BY:**

**Siddhant Rimal**

**013BSCCSIT039**

**SUBMITTED TO**

|  |  |
| --- | --- |
| **Er. Sanjay Kr. Yadav**  **( Lecturer )** |  |
| **Department of Computer Science** | |

Submission Date: September 14th 2015

**Tuple Relational Calculus**

Tuple relational calculus is a form of “pure” type of Query Language. Pure languages form underlying basis of query languages that people use.

In brief, TRC is a nonprocedural query language, where each query is of the form:- {t | P (t) }

* It is the set of all tuples t such that predicate P is true for t
* t is a tuple variable, t[A] denotes the value of tuple ‘t’ on attribute A
* t ∈ r denotes that tuple t is in relation r
* P is a formula similar to that of the predicate calculus

**Views**

* In some cases, it is not desirable for all users to see the entire logical model (i.e., all the actual relations stored in the database.)
* Consider a person who needs to know a customer’s loan number but has no need to see the loan amount. This person should see a relation described, in the relational algebra, by

∏customer-name, loan-number ()

* Any relation that is not of the conceptual model but is made visible to a user as a “virtual relation” is called a view.

**View Definition**

* A view is defined using the create view statement which has the form:

create view v as <query expression>

Where,

<query expression> is any legal relational algebra query expression. The view name is represented by v.

* Once a view is defined, the view name can be used to refer to the virtual relation that the view generates.
* View definition is not the same as creating a new relation by evaluating the query expression
  + Rather, a view definition causes the saving of an expression; the expression is substituted into queries using the view.

**Updates through view and Null values**

* Database modifications expressed as views must be translated to modifications of the actual relations in the database.
* Consider the person who needs to see all loan data in the loan relation except amount. The view given to the person, branchloan, is defined as:

create view branch-loan as

∏branch-name, loan-number (loan)

* Since we allow a view name to appear wherever a relation name is allowed, the person may write:

branch-loan ← branch-loan ∪ {(“Perryridge”, L-37)}

* The previous insertion must be represented by an insertion into the actual relation loan from which the view branch-loan is constructed.
* An insertion into loan requires a value for amount. The insertion can be dealt with by either.
  + rejecting the insertion and returning an error message to the user.
  + inserting a tuple (“L-37”, “Perryridge”, null) into the loan relation
* Some updates through views are impossible to translate into database relation updates
  + create view v as

σbranch-name = “Perryridge” (account))  
v ← v ∪ (L-99, Downtown, 23)

* Others cannot be translated uniquely
  + all-customer ← all-customer ∪ {(“Perryridge”, “John”)}
    - Have to choose loan or account, and create a new loan/account number!

**REFERENCE:**

* **<http://homepages.cwi.nl/~manegold/teaching/DBtech/slides/ch3-2.pdf>**