TY: Language Reference Manual

Tam Yuen
Due: 10/21/2004

1. **Introduction**

   This manual describes the ty language specified by Tam Yuen for his course in Programming Languages and Translators (COMS W4115).

   ty compliments the Structured Query Language (SQL) by offering simplified syntax for commonly used SQL statements of JOIN, SELECT, UPDATE, INSERT, AND DELETE. Some knowledge of SQL is needed to fully exploit the flexibility and ease-of-use of ty.

2. **Lexical Conventions**

   a. **Line Terminators**

      Line terminators are used to signal end of instruction. Instructions are terminated by the ASCII characters CR ("carriage return"), LF ("line feed"), or CR LF.

   b. **Whitespace**

      Whitespace is defined as the ASCII space, horizontal tab, and form feed characters. Line terminators are treated as whitespaces.

   c. **Tokens**

      There are three tokens: identifiers, keywords, and separators. Whitespace is ignored but is used as token separator. Whitespace is required between identifiers and keywords.

   d. **Identifiers**

      An identifier is a sequence of letters and digits. Identifiers must start with a letter and may be followed by letters or digits. Identifiers are case-sensitive.

      *Identifier -> ALPHA (ALPHA | DIGIT)**

   e. **Keywords**

      The following identifiers are reserved keywords:

      s, u, i, d, j, w

      s stands for the SELECT command in SQL; *u* for UPDATE; *i* for INSERT; *d* for DELETE; *j* for JOIN; and *w* for WHERE.

   f. **Numbers**

      Integers are a sequence of digits. Floating-point numbers are a sequence of digits with a decimal point. Exponential floating-point numbers are not allowed.

      Number -> (DIGIT)+ ('.' (DIGIT))?*

g. Logical Operators
There are two conditional operators: AND, OR.
These operators are used in the *w* clause of the SQL select statement.

h. Relational Operators
There are six relational operators: >, >=, <, <=, =, !=.
These operators are used in the *w* clause of the SQL select statement. The = operator may be used for column assignment.

3. Tutorial

The following tutorial highlights the differences between ty and SQL commands by providing some examples.

JOIN *(j)*:

a. ty: j emp(empid), salary(empid)
SQL: select * from emp, salary where emp.empid=salary.empid

b. ty: j emp(empid, lname, fname), salary(empid)
SQL select fname, lname from emp, salary
where emp.empid=salary.empid

SELECT *(s)*:

a. ty: s emp(*)
SQL: select * from emp

b. ty: s emp(fname)
SQL: select fname from emp

c. ty: s emp(fname, lname)
SQL: select fname, lname from emp

d. ty: s emp(lname, fname, w city=’New York’)
SQL: select lname, fname from emp where city=’New York’

e. ty: s emp(lname, fname, w city=’New York’ OR zip=10027)
SQL: select lname, fname from empl where city=’New York’ OR zip=10027

UPDATE *(u)*:

a. ty: u emp(salary=100000, w empid=123)
SQL: update emp set salary=100000 where empid=123

b. ty: u emp(lname=’Yuen’, fname=’Tam’, w empid=123)
SQL: update emp set lname=’Yuen’, fname=’Tam’ where empid=123

c. ty: u emp(lname='Yuen', fname='Tam', w zip=123 AND state='NY'
SQL: update emp set lname='Yuen', fname='Tam' where zip=123 AND state='NY'

INSERT *(i)*:

a. ty: i emp(lname='Yuen', fname='Tam')
SQL: insert into emp set lname='Yuen', fname='Tam'

b. ty: i emp(lname='Yuen', fname='Tam', w empid=123)
SQL: insert into emp set lname='Yuen', fname='Tam' where empid=123

DELETE *(d)*:

a. ty: d emp(w empid=123)
SQL: delete emp where empid=123

b. ty: d emp(w salary<100000 OR city='NY')
SQL: delete emp where salary<100000 OR city='NY'

c. ty: d emp(w zip=10027 and salary>10000)
d. SQL: delete emp where zip=10027 and salary>100000