

FOUNDATION FOR FUTURE DATABASE SYSTEMS:

THE THIRD MANIFESTO

Second Edition

by

C.J.Date and Hugh Darwen

Known errors in **first printing**

Date of this list: **August 3rd, 2002**

Page 61, last line above the footnote should read:

more spaces before and/or after the comma).

Page 62, BNF production for `<exp>`, last line should read:

| *<with exp>*

Page 77, last para., 2nd line: If "COUNT" has been replaced by "COUNT or COUNTD", replace "COUNT or COUNTD" by "COUNT".

Page 102, 1st footnote: Replace "an example might be INTERVAL, where each individual (generated) interval type" by "e.g., an INTERVAL type generator might allow the generation of individual interval types, where each such generated type".

Page 115, last two lines above heading RM PRESCRIPTION 4: Delete "in most respects".

Page 137, 1st block of cp material, 3rd line should read:

QTY QTY } } ... ;

Page 137, subsection **GROUP and UNGROUP operators**: 1st cp line should read:

RELATION { S# S#, PQ RELATION { P# P#, QTY QTY } }

Page 147, last line of cp material on the page should read:

ZIP CHAR } }

Page 148, 12th line of cp material on the page should read:

LOC POINT } } ;

Page 168, para. beginning "For a final example", 3rd cp line should read:

QTY QTY } }

Page 177, 3rd line of cp material on the page should read:

QTY QTY } } ... ;

Page 252, IM Prescription 14: Replace the 1st para. (up to but not including the indented para. labeled a.) by the following:

If X is an expression and T is a subtype of $DT(X)$, then an operator of the form

TREAT_DOWN_AS_T (X)

(or logical equivalent thereof) shall be supported. We refer to such operators generically as "TREAT DOWN" operators; their semantics are as follows. First, if $MST(X)$ is not a subtype of T , then a type error shall occur. Otherwise:

Page 253, IM Prescription 15 (*Definition assuming single inheritance only*): Replace the 1st two sentences by:

If X is an expression and T is a subtype of $DT(X)$, then an operator of the form

IS_T (X)

(or logical equivalent thereof) shall be supported.

Page 261, subsection "Syntax", 1st sentence: Replace "with little by way of" by "without".

Page 261, 1st BNF production rule should read:

```
<user scalar type def>
 ::= TYPE <user scalar type name> [ UNION ]
    [ <ne possrep def list or is def> ]
```

Page 261: Delete para. beginning "Note:".

Page 263, 2nd bullet item should read:

- Distinct subtypes of the same supertype are disjoint (unless one is a subtype of the other). In particular, distinct leaf types are disjoint.

Page 280, algorithm FIND_MST (cp text): Delete lines 2-5 (from "let the POSSREP ..." to "END IF ;").

Page 288, IM Prescription 14: Replace the 1st para. (up to but not including the indented para. labeled a.) by the following:

If X is an expression and T is a subtype of DT(X), then an operator of the form

TREAT_DOWN_AS_T (X)

(or logical equivalent thereof) shall be supported. We refer to such operators generically as "TREAT DOWN" operators; their semantics are as follows. First, if MST(X) is not a subtype of T, then a type error shall occur. Otherwise:

Page 291, sentence beginning "And this assignment", replace "the following simpler one:" by "the following:".

Page 291, 2nd chunk of cp text (beginning "E :="): Replace by:

```
IF IS_CIRCLE ( E )
  THEN E := CIRCLE ( LENGTH ( 5.0 ), THE_CTR ( E ) ) ;
  ELSE signal error ;
END IF ;
```

Page 291: Delete sentence immediately following foregoing cp text ("In other words, ... just shorthand.").

Page 292, IM Prescription 15: Replace the 1st two sentences by:

If X is an expression and T is a subtype of DT(X), then an operator of the form

IS_T (X)

(or logical equivalent thereof) shall be supported.

Page 311, para. beginning "Why might", last sentence should read simply "For example:" (i.e., delete "(to invent some syntax on the fly)").

Page 319, last line of cp text should read:

```
D = THE_D ( RECTANGLE ) } ;
```

(i.e., insert a closing parenthesis before the closing brace).

Page 333, IM Prescription 15: Replace the 1st two sentences by:

15. *If X is an expression and T is a subtype of DT(X), then an operator of the form*

IS_T (X)

(or logical equivalent thereof) shall be supported.

Page 333, footnote: Replace "would probably" by "might possibly".

Page 429, Section 9 (*Tuples*), lines 5-6: Replace "the row value constructor of SQL" by "the row constructor of SQL".

Page 501, reference [45]: Replace "M. Pratini" by "M. Piattini". Also, replace "(2000, to appear)" by "(2000)".

Page 543, index entry for "row constructor (SQL)": Replace "428" by "428,429".

***** End *** End *** End *****