

Tutorial D: Approved Variations

Hugh Darwen and C.J. Date
Date of this first edition: May 10th, 2017

The language **Tutorial D** is defined in a PDF document available in the References section at www.thethirdmanifesto.com. It has been subject to some minor revisions since its latest publication in book form (Chapter 11 in *Database Explorations*), but revisions run the risk of invalidating examples in various published texts, not necessarily all by us (jointly or individually). However, the language is still subject to comment, especially in the email forum ttm@thethirdmanifesto.com, and occasionally such comments suggest revisions or variations that we would probably agree to if the language was still in its development phase (it was originally developed in the mid-1990s). The present document describes and defines such variations so as to provide a convenient reference point for implementers and writers who might prefer to use them, in which case the chosen variant should be referred to as “**Tutorial D** (*name dialect*)”.

We do not include variations arising from suggestions that are merely harmless (such as using a different name for some key word), nor those that would be beyond the scope of **Tutorial D**'s main purpose as indicated by its name: to assist in the teaching of relational database theory.

1. Omission of DIVIDEBY and/or SUMMARIZE

Relational “division”—DIVIDEBY in **Tutorial D**—is only occasionally useful and is in any case difficult to understand. SUMMARIZE uses a device—appearing in the BNF as *<summary>*—which some have questioned as being a little too *ad hoc*. The addition of the operator IMAGE_IN to **Tutorial D** in September 2016 significantly reduces the motivation for both of these operators. Either or both of the following BNF productions can therefore be optionally ignored.

```
<divide>
 ::= <relation exp> DIVIDEBY <relation exp> <per>

<summarize>
 ::= SUMMARIZE <relation exp> [ <per or by> ] :
                               { <attribute assign commalist> }
```

2. Infix notation for EXTEND and SUMMARIZE

It has been observed that the syntax for invoking the operators EXTEND and SUMMARIZE uses prefix notation, whereas infix notation is used for all the other relational operators (apart from projection, which follows most of the literature in not using an operator name, and TCLOSE, which is monadic). For example, compare the BNF for *<extend>* and *<summarize>* versus *<rename>*, *<where>*, and *<dyadic join>*:

```
<extend>
 ::= EXTEND <relation exp> : { <attribute assign commalist> }

<summarize>
 ::= SUMMARIZE <relation exp> [ <per or by> ] :
                               { <attribute assign commalist> }

<rename>
 ::= <relation exp> RENAME { <renaming commalist> }
```

```
<where>
 ::= <relation exp> WHERE <bool exp>
```

```
<dyadic join>
 ::= <relation exp> JOIN <relation exp>
```

The following alternatives for *<extend>* and *<summarize>* may therefore be used instead of, or as well as, those above.

```
<extend>
 ::= <relation exp> EXTEND [ : ] { <attribute assign commalist> }
```

```
<tuple extend>
 ::= <tuple exp> EXTEND [ : ] { <attribute assign commalist> }
```

```
<summarize>
 ::= <relation exp> SUMMARIZE [ <per or by> ] [ : ]
                               { <attribute assign commalist> }
```

All of these alternatives must be used if any of them are (though *<summarize>* can be omitted altogether as noted in Section 1. **Omission of DIVIDEBY and/or SUMMARIZE**).

The colons are included for consistency with the prefix forms, should those be retained, and also with the colons that appear in connection with the WITH construct and UPDATE operators.

**** END ****