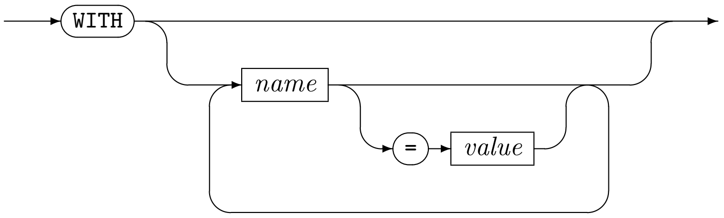
shift/reduce conflicts using yacc/bison. In fact we defined the grammar for *skyline_options*, just as simple as that:

skyline_options



While the definition given above describes exactly the intended semantics, this definition gives us great flexibility during the development. Using the simple definition we were able to introduce new options as we liked on the fly. The options are passed around from the parser, through query planning to the execution engine as a simple *name/value* list. So whenever we needed a new option we just had to query this list at the desired point in the code. If the equal sign and the value are omitted, this has the semantics of assigning the integer constant 1 to the name.

Note that the grammar for *select_clause* is defined a little bit differently[4] but only in the aspect of the associativity of the entire `SKYLINE OF` clause, this is because the SQL:1992 Standard [ISO/ANSI, 1991] requires the following statement:

`SELECT foo UNION SELECT bar ORDER BY baz` (4.2)

to be parsed as

`(SELECT foo UNION SELECT bar) ORDER BY baz` (4.3)

and not as

`SELECT foo UNION (SELECT bar ORDER BY baz)` (4.4)

For the `SKYLINE OF` clause we decided that the it should be left associative, i.e.

`SELECT * FROM foo UNION SELECT * FROM bar SKYLINE OF baz` (4.5)

will be parsed as

`SELECT * FROM foo UNION (SELECT * FROM bar SKYLINE OF baz)` (4.6)

We did so, because we believe the `SKYLINE OF` clause is closer related to the `GROUP BY` clause than to the `ORDER BY` clause, therefore we parse it in the same way.

4.1.2 Reserved Keywords

On the lexical level we introduced one new *reserved keyword*: “`SKYLINE`”[5]. Introducing a reserved keyword has the usual consequences: it cannot be used as normal identifier, like illustrated in the following example:

```
db=# CREATE DATABASE SKYLINE;
ERROR:  syntax error at or near "SKYLINE"
LINE 1: CREATE DATABASE SKYLINE;
                        ^
```

[4] see `src/backend/parser/gram.y` for details

[5] see `src/backend/parser/keywords.c`