

---

Subject: Sql: New 'Of' 'syntax'

Posted by [mirek](#) on Thu, 01 Dec 2011 23:20:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Recently I have found I am doing a lot of joins in SQL, which leads to pretty verbose 'syntax' to get columns in Select list:

```
Select(ID.Of(FOO), NAME.Of(FOO), SURNAME.Of(FOO),  
      ID.Of(BAR), CODE.Of(BAR)).From(FOO).InnerJoin(BAR)...
```

after some thinking, I have introduced 'group synonym' for Of via function call, so now we can write:

```
Select(FOO(ID, NAME, SURNAME), BAR(ID, CODE)).From(FOO).InnerJoin(BAR)...
```

---

---

Subject: Re: Sql: New 'Of' 'syntax'

Posted by [unodgs](#) on Fri, 02 Dec 2011 19:21:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

mirek wrote on Thu, 01 December 2011 18:20 Recently I have found I am doing a lot of joins in SQL, which leads to pretty verbose 'syntax' to get columns in Select list:

```
Select(ID.Of(FOO), NAME.Of(FOO), SURNAME.Of(FOO),  
      ID.Of(BAR), CODE.Of(BAR)).From(FOO).InnerJoin(BAR)...
```

after some thinking, I have introduced 'group synonym' for Of via function call, so now we can write:

```
Select(FOO(ID, NAME, SURNAME), BAR(ID, CODE)).From(FOO).InnerJoin(BAR)...
```

That's really cool. In one of my apps I simply generated ids that started with table shortcut. For PERSON table I've got:

PE\_NAME, PE\_AGE and so on. Disadvantage was too many global id objects.

---

---

Subject: Re: Sql: New 'Of' 'syntax'

Posted by [mirek](#) on Sat, 03 Dec 2011 08:34:44 GMT

unodgs wrote on Fri, 02 December 2011 14:21mirek wrote on Thu, 01 December 2011 18:20Recently I have found I am doing a lot of joins in SQL, which leads to pretty verbose 'syntax' to get columns in Select list:

```
Select(ID.Of(FOO), NAME.Of(FOO), SURNAME.Of(FOO),  
       ID.Of(BAR), CODE.Of(BAR)).From(FOO).InnerJoin(BAR)...
```

after some thinking, I have introduced 'group synonym' for Of via function call, so now we can write:

```
Select(FOO(ID, NAME, SURNAME), BAR(ID, CODE)).From(FOO).InnerJoin(BAR)...
```

That's really cool. In one of my apps I simply generated ids that started with table shortcut. For PERSON table I've got:

PE\_NAME, PE\_AGE and so on. Disadvantage was too many global id objects.

BTW, I have two more plans in this area:

InnerJoinRef(TABLE) will automatically create On condition based on .sch file foreign key.

And I consider creating some predefined SqlIds to be used as As ids (like aA ... aZ, aA1 ... aZ1, aA2 ... aA2). And maybe use ^ as As synonym:

```
Select(aF(ID, NAME, SURNAME), aB(ID,  
CODE)).From(FOO^aF).InnerJoinRef(BAR^aB).Where(...)
```

Mirek

---