Subject: Sql: *JoinRef
Posted by mirek on Fri, 09 Dec 2011 13:14:50 GMT
View Forum Message <> Reply to Message

It took some time and effort, but SqlExp is now able to create join conditions based on .sch file REFERENCES and PRIMARY_KEY attributes:

```
Select(ID(ID, NAME, LASTNAME))
.From(TABLE1).LeftJoinRef(TABLE2)
.Where(BDATE == Date(2011, 12, 9))
```

produces

```
select ID.ID, ID.NAME, ID.LASTNAME
 from TABLE1 left outer join TABLE2 on TABLE2.TABLE1_ID = TABLE1.NAME
 where BDATE = date '2011-12-09'
```

it is also possible to use On clause anyway - it gets combined with 'and':

```
Select(ID(ID, NAME, LASTNAME))
.From(TABLE1).LeftJoinRef(TABLE2).On(IsNull(BDATE))
.Where(BDATE == Date(2011, 12, 9))
```

```
select ID.ID, ID.NAME, ID.LASTNAME
 from TABLE1 left outer join TABLE2 on TABLE2.TABLE1_ID = TABLE1.NAME and BDATE is NULL
 where BDATE = date '2011-12-09'
```

Subject: Re: Sql: *JoinRef
Posted by Didier on Wed, 21 Dec 2011 23:22:54 GMT
View Forum Message <> Reply to Message

Hi,

What syntax can I use to serialize more than one join at a time.

Take the following stupid .sch example:

```
TABLE_(TABLE_B)
 INT_   (B_ID) PRIMARY_KEY AUTO_INCREMENT
 INT_   (AGE)
END_TABLE

TABLE_(TABLE_C)
 INT_   (C_ID) PRIMARY_KEY AUTO_INCREMENT
 INT_   (WEIGTH)
END_TABLE

TABLE_(TABLE_A)
 INT_   (A_ID) PRIMARY_KEY AUTO_INCREMENT
 INT_   (A_REF_B) REFERENCES(TABLE_B)
 INT_   (A_REF_C) REFERENCES(TABLE_C)
 INT_   (A_SIZE)
END_TABLE
```

I would like to write something like:

```
Sql * Select(myData).From(TABLE_A)
     .LeftJoinRef(TABLE_B).Where(AGE == xxxx)
     .LeftJoinRef(TABLE_C).Where(WEIGHT == yyyy)
```

Is this possible ??

Subject: Re: Sql: *JoinRef
Posted by mirek on Thu, 22 Dec 2011 07:17:55 GMT
View Forum Message <> Reply to Message

Didier wrote on Wed, 21 December 2011 18:22Hi,

What syntax can I use to serialize more than one join at a time.

Take the following stupid .sch example:

```
TABLE_(TABLE_B)
 INT_   (B_ID) PRIMARY_KEY AUTO_INCREMENT
 INT_   (AGE)
END_TABLE

TABLE_(TABLE_C)
```

```
 INT_    (C_ID) PRIMARY_KEY AUTO_INCREMENT
 INT_    (WEIGTH)
END_TABLE

TABLE_(TABLE_A)
 INT_    (A_ID) PRIMARY_KEY AUTO_INCREMENT
 INT_    (A_REF_B) REFERENCES(TABLE_B)
 INT_    (A_REF_C) REFERENCES(TABLE_C)
 INT_    (A_SIZE)
END_TABLE
```

I would like to write something like:

```
Sql * Select(myData).From(TABLE_A)
      .LeftJoinRef(TABLE_B).Where(AGE == xxxx)
      .LeftJoinRef(TABLE_C).Where(WEIGHT == yyyy)
```

Is this possible ??

This one no. The answer depends on what kind of SQL do you want to produce.

I think the most sense makes this:

```
Sql * Select(myData).From(TABLE_A)
      .LeftJoinRef(TABLE_B).On(AGE == xxxx)
      .LeftJoinRef(TABLE_C).On(WEIGHT == yyyy)
```

(but it is not 'where', because we have left joins, so AGE or WEIGHT would yield null, which would exclude such records when placed in where).

Mirek

Subject: Re: Sql: *JoinRef
Posted by Didier on Thu, 22 Dec 2011 16:26:47 GMT
View Forum Message <> Reply to Message

OK thank's
I will try it.

Subject: Re: Sql: *JoinRef
Posted by Didier on Thu, 22 Dec 2011 22:10:10 GMT
View Forum Message <> Reply to Message

Hi mirek,

My example was mixed up (I posted it late ...)
What I'm trying to do is the following is use 'join' to
replace the following expression :

```
Sql * Select(myData).From(TABLE_A)
      .Where(    A_REF_B == Select(B_ID).From(TABLE_B).Where(AGE == xxxxxx)
             && A_REF_C == yyyyyyy
            );
```

With the same schema :

```
TABLE_(TABLE_B)
 INT_    (B_ID) PRIMARY_KEY AUTO_INCREMENT
 INT_    (AGE)
END_TABLE

TABLE_(TABLE_C)
 INT_    (C_ID) PRIMARY_KEY AUTO_INCREMENT
 INT_    (WEIGTH)
END_TABLE

TABLE_(TABLE_A)
 INT_    (A_ID) PRIMARY_KEY AUTO_INCREMENT
 INT_    (A_REF_B) REFERENCES(TABLE_B)
 INT_    (A_REF_C) REFERENCES(TABLE_C)
 INT_    (A_SIZE)
END_TABLE
```

I wan't to do it because I have quite bad performances and maybe using join can speed up things

Subject: Re: Sql: *JoinRef
Posted by mirek on Fri, 23 Dec 2011 07:40:30 GMT
View Forum Message <> Reply to Message

To replace "in" (nested select), you have to use normal InnerJoin, not LeftJoin.

I suppose that this should work:

```
Select(myData.Of(TABLE_A)).From(TABLE_A)
.InnerJoinRef(TABLE_B).On(AGE.Of(TABLE_B) == xxxxxx).Where(A_REF_C.Of(TABLE_A) == yyyyyy);
```

Here I expect yyyyyy to be single value, not to represent another "in"...

Subject: Re: Sql: *JoinRef
Posted by mirek on Fri, 23 Dec 2011 07:48:55 GMT
View Forum Message <> Reply to Message

(edited: Forgot to put there that "Ref"

Subject: Re: Sql: *JoinRef
Posted by Didier on Fri, 23 Dec 2011 16:24:44 GMT
View Forum Message <> Reply to Message

I tried it and it works in one case and not in another case

The only difference between the two cases which have exactly the same select line, is that:

case 1: TABLE_A has 1 ref ==> OK
case 2: TABLE_A has 2 refs ==> Does not work

mirek wrote on Fri, 23 December 2011 08:40
Here I expect yyyyyy to be single value, not to represent another "in"...
==> Yes

Subject: Re: Sql: *JoinRef
Posted by mirek on Fri, 23 Dec 2011 16:26:59 GMT
View Forum Message <> Reply to Message

Didier wrote on Fri, 23 December 2011 11:24I tried it and it works in one case and not in another case

The only difference between the two cases is that:

case 1: TABLE_A has 1 ref ==> OK
case 2: TABLE_A has 2 refs ==> Does not work

Well, it all depends on what SQL you want to emit....

I guess, in case 2, it as to return TABLE_A record twice, but that is exactly what join is supposed to do...

Mirek

Subject: Re: Sql: *JoinRef
Posted by mirek on Fri, 10 Feb 2012 18:12:36 GMT
View Forum Message <> Reply to Message

After some experiences I have to say that this feature is a little bit dangerous, as it can eventually pick wrong 'connection' between tables. Worse, it can in some cases even break existing code by adding columns to tables. (Google "natural join sql" which is quite similar feature for explanation).

However, it still seems very useful and works just fine in most cases. To fix "wrong connection" problem, I have introduced heuristics that it first tries to pick foreign key with the same name as primary or foreign key key that has name table_pk (if FOO has primary key ID, then foreign key has to be FOO_ID) - this covers two most common naming conventions. This has fixed "picked wrong connection" problem for me.

As for other problem (instability in SQL issues), I guess that given this heuristics, it is something that can be avoided when designing and upgrading DB schema. I mean, if DB schema makes any sense, things should work.

Subject: Re: Sql: *JoinRef
Posted by Didier on Sat, 11 Feb 2012 08:31:51 GMT
View Forum Message <> Reply to Message

OK Thank's,

I will try it in a week after my vacations
