Subject: S_type and ValueMap
Posted by Giorgio on Tue, 15 Oct 2019 09:40:27 GMT
View Forum Message <> Reply to Message

Hi there,
I have the following code: the idea is to get data from db, put them in a vector of ValueMap and then put them in a S_type structure.

```
std::vector<ValueMap> * results=this->dao->QueryTable(table, vmfields); //ValueMap(s) here represent lines from db
for(auto it = results->begin(); it != results->end(); ++it)

  S_FILLER s(*it); //S_FILLER is defined in a .sch file
  //Do something with s
}
```

S_FILLER is defined as follows:

```
TABLE_ (FILLER)
  STRING   (KMP,8) PRIMARY_KEY REFERENCES(ITEM.KMP)
  STRING_  (DESCRIPTION,50)
  STRING   (STATUS,8)
  INT_     (NOELEMENTS)
  DOUBLE_  (DENSITY)
END_TABLE
```

The problem is that the code crashes if in the S_type structure there are numerical types, if they are just strings the code runs perfectly.
Is there a way to force some kind of "automatic" conversion from String to the correct type?
Thanks,
gio

Subject: Re: S_type and ValueMap
Posted by mirek on Tue, 15 Oct 2019 10:00:11 GMT
View Forum Message <> Reply to Message

Giorgio wrote on Tue, 15 October 2019 11:40Hi there,
I have the following code: the idea is to get data from db, put them in a vector of ValueMap and then put them in a S_type structure.

```
std::vector<ValueMap> * results=this->dao->QueryTable(table, vmfields); //ValueMap(s) here
```

```
represent lines from db
for(auto it = results->begin(); it != results->end(); ++it)

  S_FILLER s(*it); //S_FILLER is defined in a .sch file
  //Do something with s
}
```

S_FILLER is defined as follows:

```
TABLE_ (FILLER)
  STRING   (KMP,8) PRIMARY_KEY REFERENCES(ITEM.KMP)
  STRING_  (DESCRIPTION,50)
  STRING   (STATUS,8)
  INT_     (NOELEMENTS)
  DOUBLE_  (DENSITY)
END_TABLE
```

The problem is that the code crashes if in the S_type structure there are numerical types, if they are just strings the code runs perfectly.
Is there a way to force some kind of "automatic" conversion from String to the correct type?
Thanks,
gio

No. I believe that would be dangerous.

I think you can do this by using S_type introspection and doing the conversion yourself. E.g. use GetRef to get a reference to column attribute, then you can use 'Is' to check for its type. Create a template function to do the conversion to accomodate various S_tables.

Mirek

Subject: Re: S_type and ValueMap
Posted by Giorgio on Tue, 15 Oct 2019 11:54:52 GMT

View Forum Message <> Reply to Message

Hi,
thank you for your suggestion.

This is what I came up with, I post it here for reference and improvements.

[...]

```
  //S_FILLER s(*vm); Crashes if it contains int or double
  S_FILLER s;
  S_convert(&s,vm);

[...]

 template <class T>
 void S_convert(T * s, ValueMap * vm){

  for(int i=0;i<vm->GetCount();i++){

   Ref r=s->GetRef(SqlId(vm->GetKey(i)));
   if(r.Is<double>())
    s->Set(SqlId(vm->GetKey(i)),StrDbl(AsString(vm->GetValue(i))));

   if(r.Is<int>())
    s->Set(SqlId(vm->GetKey(i)),StrInt(AsString(vm->GetValue(i))));

   if(r.Is<String>())
    s->Set(SqlId(vm->GetKey(i)),AsString(vm->GetValue(i)));
  }

 }
```

Subject: Re: S_type and ValueMap
Posted by mirek on Tue, 15 Oct 2019 12:39:41 GMT

View Forum Message <> Reply to Message

Giorgio wrote on Tue, 15 October 2019 13:54Hi,
thank you for your suggestion.

This is what I came up with, I post it here for reference and improvements.

```
[...]
  //S_FILLER s(*vm); Crashes if it contains int or double
  S_FILLER s;
  S_convert(&s,vm);

[...]

 template <class T>
 void S_convert(T * s, ValueMap * vm){
  for(int i=0;i<vm->GetCount();i++){

   Ref r=s->GetRef(SqlId(vm->GetKey(i)));
   if(r.Is<double>())
    s->Set(SqlId(vm->GetKey(i)),StrDbl(AsString(vm->GetValue(i))));

   if(r.Is<int>())
    s->Set(SqlId(vm->GetKey(i)),StrInt(AsString(vm->GetValue(i))));

   if(r.Is<String>())
    s->Set(SqlId(vm->GetKey(i)),AsString(vm->GetValue(i)));
  }

 }
```

Why not

```
 template <class T>
 void S_convert(T& s, const ValueMap& vm){
```

?

Mirek

Subject: Re: S_type and ValueMap
Posted by Giorgio on Mon, 21 Oct 2019 07:31:57 GMT
View Forum Message <> Reply to Message

You are right, in my specific case I already have the ValueMap as a pointer, but in general your solution is better.

Also, I added a continue instruction after evry set (all in brackets of course) to improve a little the execution time.