
Subject: Re: How to display fields from different tables in one SQLArray

Posted by [mirek](#) on Wed, 12 Nov 2008 21:23:38 GMT

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

sergeynikitin wrote on Sun, 02 November 2008 16:09: By the way, I have decided to how to display related fields from other tables by using SqlArray.

[code]

```
struct ConvCompany : Convert
{
    Value Format(const Value &q) const
    {
        SQL * Select(COM_NAME).From(COMPANY).Where(COM_ID == q);
        String company;
        if(SQL.Fetch())
            company = SQL[COM_NAME];
        else
            company = "";
        return q.IsNull() ? Null : company;
    }
};
```

WARNING: Fetching from SQL ("SQL.Fetch()") is NOT RECOMMENDED (based on experiences:)

It is way too simple to call something in the fetch loop which will issue different SQL statment through SQL.

Use "Sql sql; sql * Select....; sql.Fetch()" instead.

Considering efficiency, IME, this method is usually still passable - one reason is that ArrayCtrl converts only if it needs to Paint it, so if the table has 10000 rows, for the first user action, only about 30 selects are performed.

You can also help it a bit by using "Cache" column modified - in that case, for particular ArrayCtrl cell, Convert is only invoked once, then the converted value is cached. That, obviously, has disadvantages of its own, but usually is harmless.

If you need to be more efficient, you can also use WhenPostQuery or something like that to prefetch all values you need for conversion. E.g. this is direct excerpt from my current job, where I have to solve quite similar problem:

```
struct NameConvert : Convert {
    VectorMap<int, String> name;
    VectorMap<int, String> surname;
    virtual Value Format(const Value& q) const;
    void LoadAttrs();
};
```

```
};

Value NameConvert::Format(const Value& q) const
{
    return surname.Get(q, String()) + ' ' + name.Get(q, String());
}

void NameConvert::LoadAttrs()
{
    name.Clear();
    surname.Clear();
    Sql sql;
    sql * Select(SUBJECT_ID, ATTR, TEXT).From(SUBJECT_ATTR)
        .OrderBy(Descending(Nvl(TILL, Date::High())));
    while(sql.Fetch()) {
        int sid = sql[SUBJECT_ID];
        String id = sql[ATTR];
        if(id == "NAME" && name.Find(sid) < 0)
            name.Add(sid, sql[TEXT]);
        if(id == "LASTNAME" && surname.Find(sid) < 0)
            surname.Add(sid, sql[TEXT]);
    }
}
```

Quote:

Maybe this way: [code]

```
projects.SetTable(PROJECTV),Leftjoined(COMPANY,VPR_COMPANYID
),AS_PREFIX(VCOM_);
projects.AddColumn(VCOM_COMPANYNAME, t_("Customer"));
```

This is the ideal, or something similar.

I will think about it.

However, 85% of SqlArray code is dealing with insert/update/edit, which would not apply here (IMO). You can easily reimplement those 15% as special case quite easily IMO.

Mirek