

关于华通项目 Nhibernate 中的 Sum 统计求和异常问题的解决

小问题，简单说下，几个月前，在做 huatongmis 个人账户余额统计的时候，碰到一个异常。当时很快解决了。不过后来项目组又有人碰到了这个异常不知道如何解决。


加上公司 Orm 框架全面替换成 Nhibernate，遂觉得有必要成文说下我的求和代码如下：

```
if (isRMB)
{
    //统计消费的人民币总额
    float? result = 0;
    try
    {
        result =
            context.ConsumerRecord.Where(p => p.Vipuser.VipId == id && p.CurrencyType == 1).Sum(
                p => p.ConsumeSum);
    }
    catch (Exception e)
    {
        LogHelper.WriteLog(
            string.Format("ConsumerRecordService.GetMemberTotalConsumerRecordByVipUserId({0},{1})", id,
                isRMB.ToString()), e);
    }
    if (result != null)
    {
        return (float)result;
    }
    else
    {
        return 0.0f;
    }
}
```

Linq 的 Sum 求和，很简单把，但是运行会报错。

错误如下。

```
//HKD充值金额不够用，则需要从港币赠送金额中扣除差额
HKDGivenTotal = givenNumTotalHKD + payRecordTotalHKD - consumerRecordTotalHKD;
}
//计算出充值余额和赠送余额，赋给bo返回
vipUserBo.TheBalanceHKD = HKDPayTotal;
vipUserBo.TheBalanceRMB = RMBPayTotal;
vipUserBo.GivenHKD = HKDGivenTotal;
vipUserBo.GivenRMB = RMBGivenTotal;
}
return vipUserBo;
}
catch (Exception e)
{
    LogHelper.WriteLog(
        return null;
    }
}
/// <summary>
/// 返回分页数据
/// add by ys on 2013-6-
/// </summary>
/// <param name="order">
/// <param name="page">
/// <param name="rows">
```



什么原因呢。从字面的意思，是不支持 Single(Float)类型，但是用使用 lamda 表达式,编译的

时候并没有报错啊

```
/// <param name="id">会员id</param>
/// <returns>返回统计计算结果</returns>
public static float GetMemberTotalConsumerRecordByVipUserId(int id, bool isRMB)
{
    if (isRMB)
    {
        //统计消费的人民币总额
        float? result = 0;
        using (var context = new MiniSysDataContext())
        {
            try
            {
                result =
                    context.ConsumerRecord.Where(p => p.Vipuser.VipId == id && p.CurrencyType == 1).Sum(p => p.ConsumeSum);
            }
            catch (Exception e)
            {
                LogHelper.WriteLog(string.Format("ConsumerRecordService.GetMemberTotalConsumerRecordByVipUserId({0},{1})", id, isRMB),
                );
            }
        }
        if (result != null)
        {
            return (float)result;
        }
    }
}
```

这些 linq 计算函数是微软的库函数。并非 Nhibernate 提供的。Nhibernate 是兼容 Linq 的，

```
/// 返回结果:
/// 投影值之和。
///
/// 异常:
/// System.ArgumentNullException:
/// source 或 selector 为 null。
public static float? Sum<TSource>(this IQueryable<TSource> source, Expression<Func<TSource, float?>> selector);
```

微软的 Linq 求和函数 Sum 是支持 float 求和的，所以编译不报错。导致这个异常产生的原因是 Nhibernate。

于是我，调整了代码，改为 long 型，编译，运行就不会报错了。

代码如下

```
///统计消费的人民币总额
float? result = 0;
try
{
    result =
        context.ConsumerRecord.Where(p => p.Vipuser.VipId == id && p.CurrencyType == 1).Sum(
            p => (long?)p.ConsumeSum);
}
catch (Exception e)
{
    LogHelper.WriteLog(
        string.Format("ConsumerRecordService.GetMemberTotalConsumerRecordByVipUserId({0},{1})", id,
            isRMB.ToString()), e);
}
if (result != null)
```

为什么是可空长整型，可空是因为字段设计的时候是可空类型，这里转换为可空类型，凡是 null 值都会过滤掉不计算。

长整形是避免溢出。

其实这里有一个弊端，那就是计算会直接丢掉小数。所以合理的做法是 lamda 表达式里先乘以 1000 倍，求出的结果值处以 1000.可以避免小数值被丢弃的问题。就像这样

```

if (isRMB)
{
    //统计消费的人民币总额
    float? result = 0;
    try
    {
        result =
            context.ConsumerRecord.Where(p => p.Vipuser.VipId == id && p.CurrencyType == 1).Sum(
                p => (long?)(1000 * p.ConsumeSum)) / 1000;
    }
    catch (Exception e)
    {
        LogHelper.WriteLog(
            string.Format("ConsumerRecordService.GetMemberTotalConsumerRecordByVipUserId({0}, {1})", id,
                isRMB.ToString()), e);
    }
    if (result != null)
    {

```

上面的 lamada 表达式中乘以 1000 倍，结果再处以 1000，经过我测试，抛出如下异常。

```

异常类: logerror [(null)]
ConsumerRecordService.GetMemberTotalConsumerRecordByVipUserId(44,False)
System.NotSupportedException: Expression type 'NhSumExpression' is not supported by this SelectClauseVisitor.
在 Remotion.Linq.Parsing.ExpressionTreeVisitor.VisitUnknownNonExtensionExpression(Expression expression)
在 NHibernate.Linq.Visitors.SelectClauseVisitor.VisitExpression(Expression expression)
在 NHibernate.Linq.Visitors.SelectClauseVisitor.Visit(Expression expression)
在 NHibernate.Linq.Visitors.QueryModelVisitor.VisitSelectClause(SelectClause selectClause, QueryModel queryModel)
在 Remotion.Linq.QueryModelVisitorBase.VisitQueryModel(QueryModel queryModel)
在 NHibernate.Linq.Visitors.QueryModelVisitor.Visit()
在 NHibernate.Linq.Visitors.QueryModelVisitor.GenerateHqlQuery(QueryModel queryModel, VisitorParameters parameters, Boolean root)
在 NHibernate.Linq.NhLinqExpression.Translate(ISessionFactoryImplementor sessionFactory)
在 NHibernate.Hql.Ast.ANTLR.ASTQueryTranslatorFactory.CreateQueryTranslators(String queryIdentifier, IQueryExpression queryExpression, String collectionName)
在 NHibernate.Engine.Query.HQLExpressionQueryPlan.CreateTranslators(String expressionStr, IQueryExpression queryExpression, String collectionRole, Boolean)
在 NHibernate.Engine.Query.HQLExpressionQueryPlan.ctor(String expressionStr, IQueryExpression queryExpression, String collectionRole, Boolean shallow, ID)
在 NHibernate.Engine.Query.QueryPlanCache.GetHQLQueryPlan(IQueryExpression queryExpression, Boolean shallow, IDictionary 2 enabledFilters)
在 NHibernate.Impl.AbstractSessionImpl.GetHQLQueryPlan(IQueryExpression queryExpression, Boolean shallow)
在 NHibernate.Impl.AbstractSessionImpl.CreateQuery(IQueryExpression queryExpression)
在 NHibernate.Linq.DefaultQueryProvider.PrepareQuery(Expression expression, IQuery& query, NhLinqExpression& nhQuery)
在 NHibernate.Linq.DefaultQueryProvider.Execute(Expression expression)
在 NHibernate.Linq.DefaultQueryProvider.Execute[TResult](Expression expression)
在 System.Linq.Queryable.Sum[TSource](IQueryable 1 source, Expression 1 selector)
在 Services.Admin.MemberControl.MemberBaseService.GetMemberTotalConsumerRecordByVipUserId(MiniSysDataContext context, Int32 id, Boolean isRMB) 位置 E:\Com

```

意思是说 Nhibernate 不支持此种求和表达式

主观想象真是害死人啊。此法行不通，以后想到解决办法再补充把

嗯，就这么多把，有问题的欢迎私下找我:((

2013-11-17