1. **SM收发货报表**

**1. 业务表关系：**

|  |  |  |
| --- | --- | --- |
| S/M码发货列表 | EdiAsnPackageMovingLogView | EdiAsnPackageMovingLog |
| Whouse |
| Car |
| User |
| EdiAsnPackage |
| EdiAsnTransport |
| EdiAsnDeliveryNode |
| DockPoint |
| Part |

是否有删除数据：无

是否有更新数据：无

是否可以直接增量：是

增量判断字段：createdat

月预计增量：~100w

天增量: ~3w

* 1. **枚举表**
     1. packageAction：

操作类型

public enum PackageActionType

{

/// <summary>

/// 中转入库

/// </summary>

[Description("中转入库")]

TransferEnter = 1,

/// <summary>

/// 中转发出

/// </summary>

[Description("中转发出")]

TransferSend = 2,

/// <summary>

/// 客户收货

/// </summary>

[Description("BBAC收货")]

ClientReceive = 3,

/// <summary>

/// 手动交付

/// </summary>

[Description("手动交付")]

ManualDelivery = 4,

/// <summary>

/// 司机盲扫发出

/// </summary>

[Description("前端扫描")]

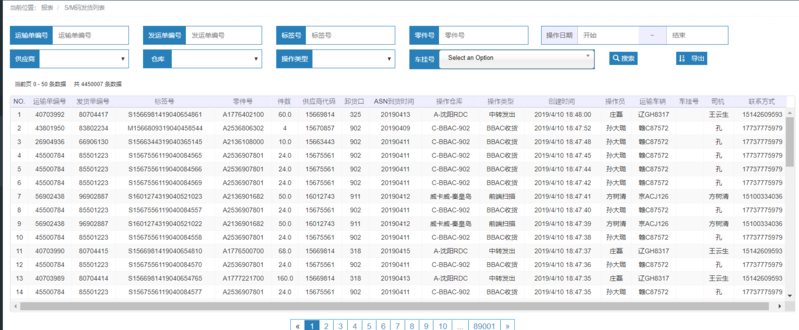
DriverSend = 9

}

**2. 报表样式：**

**2.1 报表使用频率**

**随时查询/每月做统计**



* 1. **报表内容对应**

NO.：[无]

运输单编号：transportNr

发货单编号：delnotenumber

标签号：ediAsnPackageNr

零件号：partNr

件数：quantityperpack

供应商代码：supplierCode

卸货口：dockPointCode

ASN到货时间：arrivalDate

**操作仓库：[需计算]**

如果操作类型是前端扫描，值为供应商描述(descript)；

其它全部使用视图中的whouseNr

操作类型：packageActionStr，枚举表计算

创建时间：createdAt

操作员：creatorName

运输车辆：carNr

**车挂号：carTrailerNr**

司机：driverName

联系方式：driverPhone

* 1. **报表搜索条件**

运输单编号：transportNr

发运单编号：delnotenumber

标签号：ediAsnPackageNr

零件号：partNr

操作日期：createdAt

供应商：supplierId

仓库：whouseId

操作类型：actionType

**车挂号：carTrailerId**

* 1. **报表下载**



1. **零配件报表**

**1. 业务表关系：**

|  |  |  |
| --- | --- | --- |
| 零配件报表 | PackageAccountReportView | EdiAsnPackage |
| Part |
| EdiAsnDeliveryNode |
| EdiAsnTransport |
| Supplier |
| Whouse |
| DockPoint |

是否有删除数据：无

是否有更新数据：是

* 更新字段为：state、receivedAt、isSAPReceived、receiverId、isVerify
* 更新频率为：频繁，且生命周期中会更新多次

是否可以直接增量：待定

增量判断字段：updatedAt

月预计增量：~40w

天增/更新量: ~1.3w

* 1. **枚举表**

1.1.1 DeliveryState：

/// <summary>

/// 运输状态，包装箱、托、运单

/// </summary>

public enum DeliveryState

{

/// <summary>

/// 创建

/// </summary>

[Description("创建")]

Init = 50,

/// <summary>

/// 已配载

/// </summary>

[Description("已配载")]

DeliverDispatched =90,

/// <summary>

/// 供应商已备货

/// </summary>

[Description("已备货")]

SupplierReady = 100,

/// <summary>

/// 已验收

/// </summary>

[Description("已验收")]

Checked = 400,

/// <summary>

/// 部分验收

/// </summary>

[Description("部分验收")]

PartialChecked = 401,

/// <summary>

/// 送货中

/// </summary>

[Description("送货中")]

OnDelivering = 500,

///// <summary>

///// 部分送货途中

///// </summary>

//[Description("部分送货途中")]

//PartialOnDelivering = 501,

/// <summary>

/// 已送达

/// </summary>

[Description("已送达")]

SendArrival = 600,

///// <summary>

///// 部分送达

///// </summary>

//[Description("部分送达")]

//PartialSendArrival =601,

/// <summary>

/// 已接收

/// </summary>

[Description("已接收")]

Received = 700,

/// <summary>

/// 部分接收

/// </summary>

[Description("部分接收")]

PartialReceived = 701,

/// <summary>

/// 已交付

/// </summary>

[Description("已交付")]

InStocked = 800,

/// <summary>

/// 部分交付

/// </summary>

[Description("部分交付")]

PartialInStocked = 801

}

1.1.2 PackageType

public enum PackageType

{

/// <summary>

/// 托

/// </summary>

[Description("托")]

M = 1,

/// <summary>

/// 混合包装

/// </summary>

[Description("混合")]

G = 2,

/// <summary>

/// 箱子

/// </summary>

[Description("箱")]

S = 3,

/// <summary>

/// 未知

/// </summary>

[Description("未知")]

NA = 999

}

**2. 报表样式：**

**2.1 报表使用频率**

**随时查询/每月做统计**



**2.2 报表内容对应**

NO.：[无]

运输单编号：asnTransportNr

发货单编号：asnDeliveryNodeNr

标签号：nr

状态：state通过枚举DeliveryState计算

是否验单：bool isVerify计算为： 是/否

SAP状态：bool isSAPReceived计算为：已交付/未交付

零件号：partNr

件数 ：totalQty

体积 ：通过包装类型计算：

public double valumeM

{

get

{

double v = 0.0;

if (this.packagetype == (int)PackageType.M)

{

if (this.luLength.HasValue && this.luWidth.HasValue && this.luHeight.HasValue)

{

var l = this.luLength.Value / 1000.0;

var w = this.luWidth.Value / 1000.0;

var h = this.luHeight.Value / 1000.0;

v = l \* w \* h;

}

}

else if (this.packagetype == (int)PackageType.S)

{

if (this.kltLength.HasValue && this.kltWidth.HasValue && this.kltHeight.HasValue)

{

var l = this.kltLength.Value / 1000.0;

var w = this.kltWidth.Value / 1000.0;

var h = this.kltHeight.Value / 1000.0;

v = l \* w \* h;

}

}

return Math.Round(v,1);

}

}

重量 ：通过包装类型计算：

public double weightT

{

get

{

double d = 0;

if (this.packagetype == (int)PackageType.M)

{

double.TryParse(this.grossWeight, out d);

}

else if (this.packagetype == (int)PackageType.S)

{

double.TryParse(this.tareWeight, out d);

}

return Math.Round(d, 1);

}

}

供应商简称：supplierDesc

供应商代码：supplierCode

ASN到货时间：asnArrivalDate

ASN发货时间：milkDate格式化为：yyyy-MM-dd

ASN同步时间：createdAt

收货时间：receivedAt格式化为：yyyy/MM/dd HH:mm

收货扫描人：同过receiverId到用户表中查询，使用用户信息拼接name+phone，例如：李丹丹13251650234

【待定：这个字段可再处理，不一定用旧的模式，直接分成两列】

仓库：whouseNr

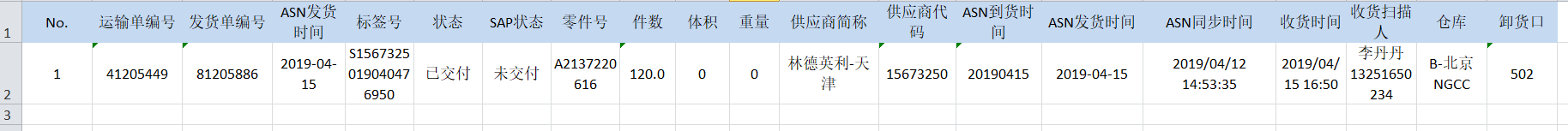
卸货口：dockPointCode

创建时间：createdAt

* 1. **报表搜索条件**

【见2.2，注意nr/name使用的是对应字段的id搜索】

* 1. **报表下载**



1. **驾驶员提货记录**

**1. 业务表关系：**

|  |  |  |
| --- | --- | --- |
| 驾驶员提货记录 | DriverCheckedRecordDetailView | DriverCheckedRecord |
| DeliveryNode |
| DeliveryNodeCar |
| Car |
| User |
| CarTroop |
| Whouse |

是否有删除数据：无

是否有更新数据：无

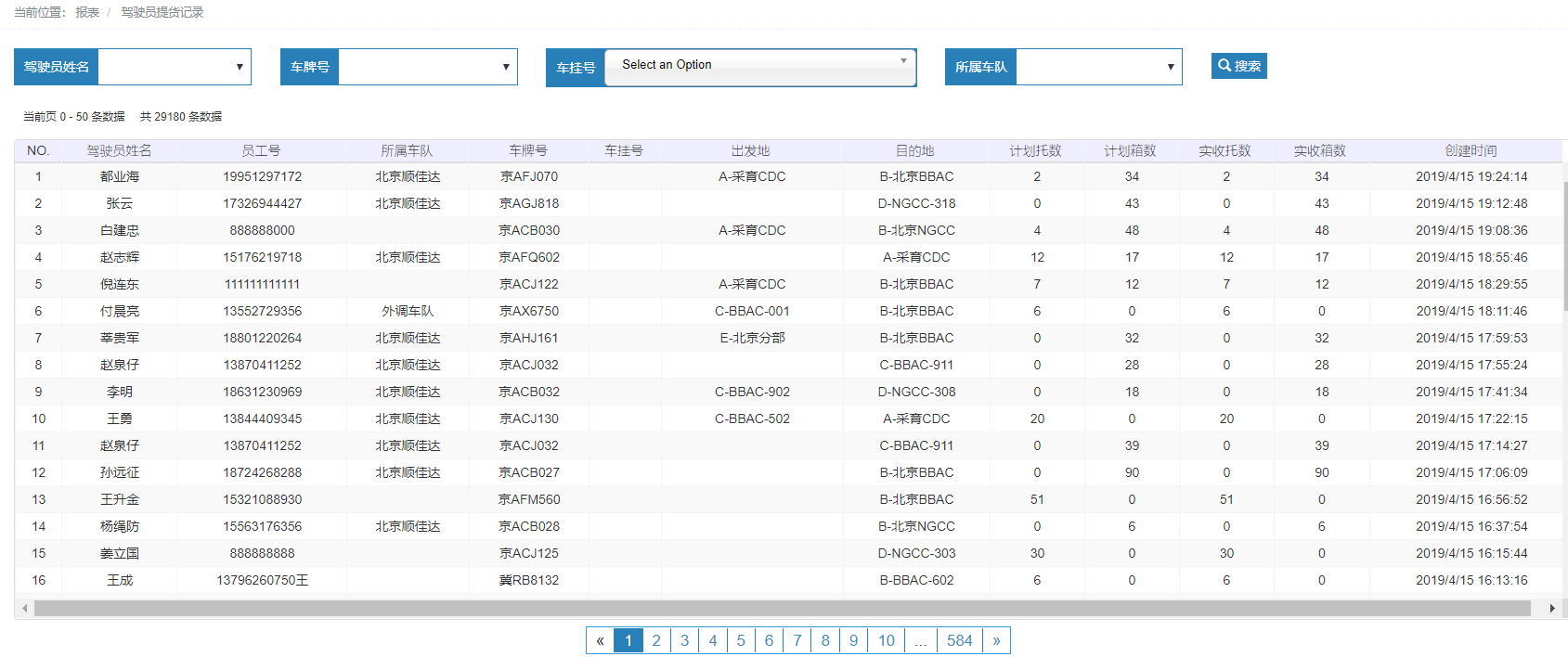
是否可以直接增量：是

增量判断字段：createdat

月预计增量：~2000

天增量: ~100

1. **报表样式：**



**2.1 报表使用频率**

**随时查询**

* 1. **报表内容对应**

NO.：[无]

驾驶员姓名：driverName

员工号：driverNr

所属车队：carTroopName

车牌号：carNr

车挂号：通过carTrailerId在carTrailer表中查询nr

出发地：sourceWhouseName

目的地：destinationWhouseName

计划托数：planTrayCount

计划箱数：planPackageCount

实收托数：realTrayCount

实收箱数：realPackageCount

创建时间：createdAt

* 1. **报表搜索条件**

【见2.2，注意nr/name使用的是对应字段的id搜索】

* 1. **报表下载**

【无】

1. **收货单列表**

**1. 业务表关系：**

|  |  |  |
| --- | --- | --- |
| 收货单列表 |  | DeliveryReceivedRecord |

【这个表没有使用视图，是在显示时动态查询数据】

是否有删除数据：是

是否有更新数据：是，主要更新状态

是否可以直接增量：是

增量判断字段：receivedAt，无updatedAt，可以考虑使用触发器生成

月预计增量：~4000

天增量: ~1300

**1.1枚举表**

* + 1. DeliveryReceivedType：

运单接收类型

public enum DeliveryReceivedType

{

/// <summary>

/// 自有仓库接收

/// </summary>

[Description("自有仓库接收")]

OwnWhouseReceived = 1,

/// <summary>

/// 客户仓库接收

/// </summary>

[Description("客户仓库接收")]

ClientWhouseReceived = 2,

}

* + 1. DeliveryReceivedStatus：

运单接收状态

public enum DeliveryReceivedStatus

{

/// <summary>

/// 新建

/// </summary>

[Description("新建")]

Init = 100,

/// <summary>

/// 收货中

/// </summary>

[Description("收货中")]

OnReceiveing = 200,

/// <summary>

/// 收货完成

/// </summary>

[Description("收货完成")]

Close = 300,

/// <summary>

/// 部分收货完成

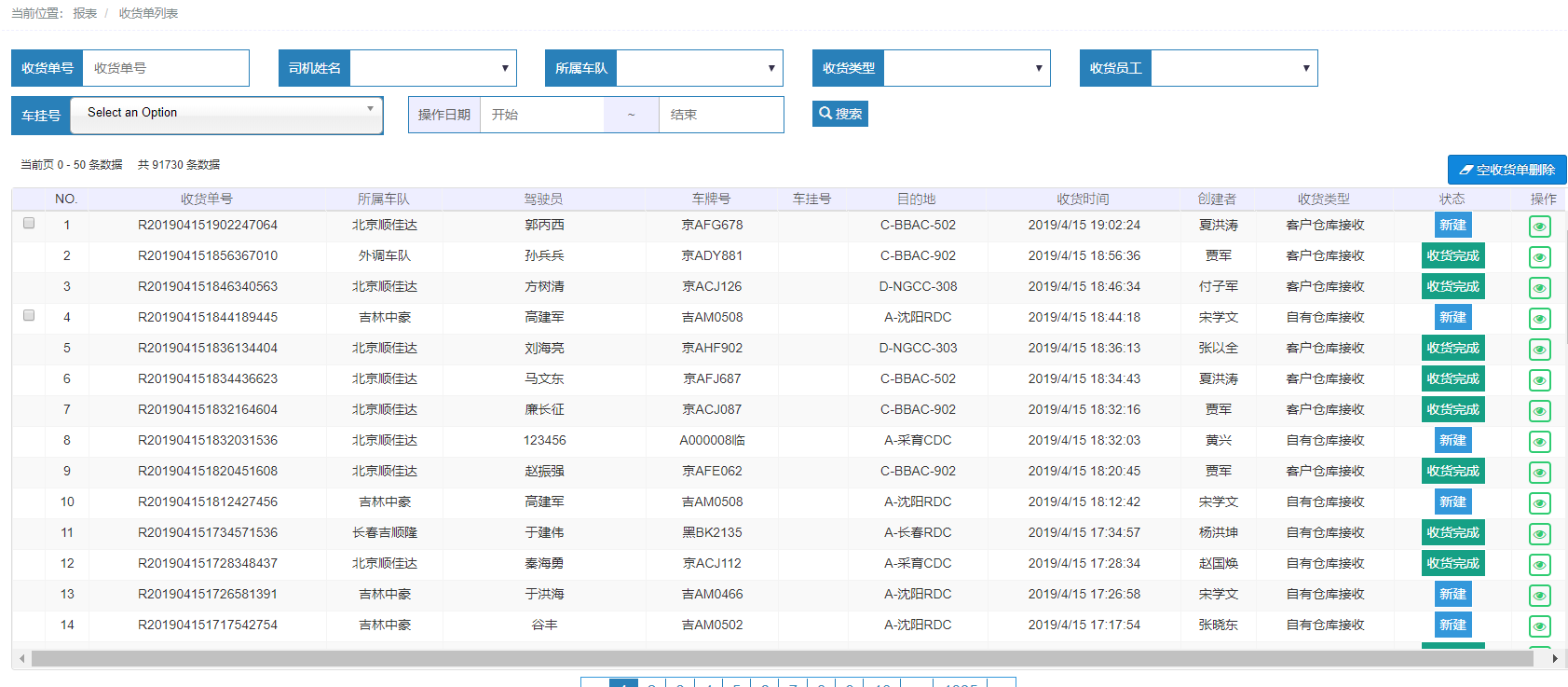
/// </summary>

[Description("部分收货")]

PartClose = 400,

}

**2. 报表样式：**



**2.1 报表使用频率**

**随时查询**

* 1. **报表内容对应**

NO.：[无]

收货单号：nr

所属车队：carTroopId在carTroop表查出nr

驾驶员：driverId在driver表查出name

车牌号：carId在car表查出nr

车挂号：carTrailerId在driver表查出nr

目的地：toId在whouse表查出nr

收货时间：receivedAt

创建者：creatorId在user表查出name

收货类型：receiveTypeStr, receiveType对应枚举表，DeliveryReceivedType，枚举表计算

状态：statusStr , status 对应枚举表，DeliveryReceivedStatus，枚举表计算

* 1. **报表搜索条件**

【见2.2，注意nr/name使用的是对应字段的id搜索】

* 1. **报表下载**

【无】

1. **运输报表**

**1. 业务表关系：**

|  |  |  |
| --- | --- | --- |
| 运输报表 | TransportReportView | DeliveryReceivedEdiPackageRelation |
| EdiAsnPackage |
| EdiAsnTransport |
| Part |
| Supplier |
| DockPoint |
| DeliveryReceivedRecord |
| User |
| Whouse |
| Car |
| CarType |

是否有删除数据：是

* 删除DeliveryReceivedEdiPackageRelation

是否有更新数据：是

* 更新DeliveryReceivedEdiPackageRelation的status；
* 更新EdiAsnPackage updatedAt、status等；

是否可以直接增量：是

增量判断字段：【无】

【建议在数据库表中DeliveryReceivedEdiPackageRelation加入createdAt/updatedAt字段，并使用数据库触发器创建/更新它】

月预计增量：~60w

天增量: ~2w

**1.1报表逻辑**

基本视图在显示时需要进行动态计算进行展示

【建议在实例化视图时，就将报表计算出来】

* 先对数据进行按照DeliveryReceivedRecord-nr+ EdiAsnTransport- transportnumber进行分组
* 再将每个分组转化为显示视图
* 通过来源类型计算来源编号

=🡺

* 计算的数据如：托数量、箱数量直接写入示例表；
* 如实体化报表时：DeliveryReceivedRecordReport，遍历改动的DeliveryReceivedRecord表，将统计数据写入DeliveryReceivedRecord

**逻辑代码1：**

from t in list

group t by new { t.recordNr, t.transportNr }

into tg

select new TransportReportViewModel()

{

# DeliveryReceivedRecord可以确认的

recordNr = tg.Key.recordNr,

driverId = tg.FirstOrDefault().driverId,

driverNr = tg.FirstOrDefault().driverNr,

driverName = tg.FirstOrDefault().driverName,

driverPhone = tg.FirstOrDefault().driverPhone,

carTroopId = tg.FirstOrDefault().carTroopId,

carTrailerId = tg.FirstOrDefault().carTrailerId,

carId = tg.FirstOrDefault().carId,

carNr = tg.FirstOrDefault().carNr,

carTypeId = tg.FirstOrDefault().carTypeId,

carTypeNr = tg.FirstOrDefault().carTypeNr,

carTypeName = tg.FirstOrDefault().carTypeName,

fromId = tg.FirstOrDefault().fromId,

fromType = tg.FirstOrDefault().fromType,

toId = tg.FirstOrDefault().toId,

toWhouseNr = tg.FirstOrDefault().toWhouseNr,

toWhouseAddress = tg.FirstOrDefault().toWhouseAddress,

receivedAt = tg.FirstOrDefault().receivedAt,

# EdiAsnTransport可以确认的

transportNr = tg.FirstOrDefault().transportNr,

supplierId = tg.FirstOrDefault().supplierId,

supplierName = tg.FirstOrDefault().supplierName,

supplierCode = tg.FirstOrDefault().supplierCode,

supplierDesc = tg.FirstOrDefault().supplierDesc,

dockPointId = tg.FirstOrDefault().dockPointId,

dockPointCode = tg.FirstOrDefault().dockPointCode,

milkDate = tg.FirstOrDefault().milkDate,

delivDate = tg.FirstOrDefault().delivDate,

#可计算的

trayCount = tg.Count(s => s.packagetype == (int)PackageType.M || s.packagetype == (int)PackageType.G),

packageCount = tg.Count(s => s.packagetype == (int)PackageType.S),

singlePackageCount = tg.Count(s => s.packagetype == (int)PackageType.S && s.parentPackageId == null),

totalValumeM = tg.Sum(s => s.valumeM) # 见**逻辑代码3**

};

**逻辑代码2：**

foreach (var model in qs)

{

if (model.fromType.Value == (int)ReceivedRecordSourceType.OwnWhouse)

{

var from = whouseDic[model.fromId.Value];

model.fromNr = from.nr;

}

else if (model.fromType.Value == (int)ReceivedRecordSourceType.SupplierWhouse)

{

var from = supplierDic[model.fromId.Value];

model.fromNr = from.code;

}

models.Add(model);

}

**逻辑代码3：**

public partial class TransportReportView

{

public double valumeM

{

get

{

if (this.packagetype == (int)PackageType.M)

{

if (this.luLength.HasValue && this.luWidth.HasValue && this.luHeight.HasValue)

{

var l = this.luLength.Value / 1000.0;

var w = this.luWidth.Value / 1000.0;

var h = this.luHeight.Value / 1000.0;

var v = l \* w \* h;

return v;

}

}

else if (this.packagetype == (int)PackageType.S)

{

if (this.kltLength.HasValue && this.kltWidth.HasValue && this.kltHeight.HasValue)

{

var l = this.kltLength.Value / 1000.0;

var w = this.kltWidth.Value / 1000.0;

var h = this.kltHeight.Value / 1000.0;

var v = l \* w \* h;

return v;

}

}

return 0;

}

}

public double weightT

{

get

{

if (this.packagetype == (int)PackageType.M)

{

double d = 0;

double.TryParse(this.grossWeight, out d);

return d;

}

else if (this.packagetype == (int)PackageType.S)

{

double d = 0;

double.TryParse(this.tareWeight, out d);

return d;

}

return 0;

}

}

}

**1.2枚举表**

* + 1. ReceivedRecordSourceType：

接收记录来源类型

public enum ReceivedRecordSourceType

{

/// <summary>

/// 供应商

/// </summary>

[Description("供应商")]

SupplierWhouse = 1,

/// <summary>

/// 自有仓库

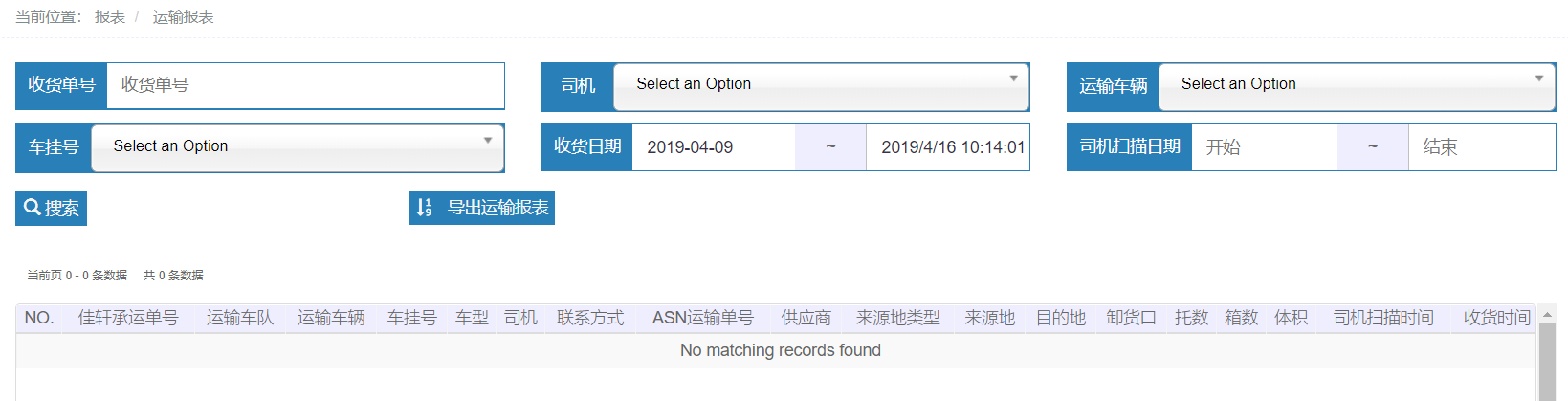
/// </summary>

[Description("自有仓库")]

OwnWhouse = 2,

}

**2. 报表样式：**



**2.1 报表使用频率**

**随时查询/每月做统计**

* 1. **报表内容对应**

NO.：[无]

佳轩承运单号：recordNr

运输车队：通过carTroopId到carTroop表中查询name

运输车辆：carNr

车挂号：通过carTrailerId到carTrailer表中查询nr

车型：carTypeNr

司机：driverName

联系方式：driverPhone

ASN运输单号：transportNr

供应商：supplierDesc

来源地类型：fromTypeStr, fromType对应枚举表，ReceivedRecordSourceType，枚举表计算

来源地：fromNr，见**逻辑代码2**

目的地：toWhouseNr

卸货口：dockPointCode

托数：trayCount，计算值，见**逻辑代码1**

箱数：singlePackageCount，计算值，见**逻辑代码1**

体积：totalValumeMRound，计算值，见**逻辑代码1**

司机扫描时间：checkedAt

收货时间：receivedAt

* 1. **报表搜索条件**

【见2.2，注意nr/name使用的是对应字段的id搜索】

* 1. **报表下载**

