1、数据仓库是什么

能干什么？

1、年度销售目标的指定，需要根据以往的历史报表进行决策，不能拍脑袋。

2、如何优化业务流程

案例 1：

一个电商网站订单的完成包括：浏览、下单、支付、物流，其中物流环节可能和中通、申通、韵达等快递公司合作。快递公司每派送一个订单，都会有订单派送的确认时间，可以根据订单派送时间来分析哪个快递公司比较快捷高效，从而选择与哪些快递公司合作，剔除哪些快递公司，增加用户友好型。

*字段名称	*指标计算方法	*指标统计口径	指标计算示例	字段说明
日期		订单创建日期		
店铺编号		指定店铺的编号		
店铺名称		指定店铺的名称		
次日达完成订单		小店配送的订单中完成的次日达的订单数		
延迟订单数		配送的次日达订单，完成且延迟的订单，其中21:00之前下的订单次日送达不算延迟，21:00之后下的订单T+2日送达不算做延迟	**此处延迟单计算方式：**完成时间T+1日完成是正常的，T+2日及以上时间完成的就算延迟了	

案例 2：

互联网中国需要对 APP 进行推广，考核的主要目标是下载安装，有些第三方渠道会对这些数据造假，比如某个渠道在凌晨批量下载，点赞操作，操作步骤一致。通过数据分析，分析出应用的名称和安装时间，来判断一个渠道的是否优质、是否作假。

2、数据仓库的特点

1、数据仓库是面向主题的，比如商品主题，订单主题。（领导关注那些方面）

传统数据库面向应用，提供什么功能。数据仓库面向分析，提供那些主题的分析。

从规模来讲依次是，数据仓库、数据集市、数据报表。

2、数据仓库是集成的，数据源是分散的，来自不同的应用。数据仓库中的综合数据，不能从源数据中直接得到，一般会经过 etl 过程（数据抽取、数据转换、数据加载）。数据抽取一般会定时的进行抽取，避免对业务系统造成影响，一般叫做 T-1 抽取、T+1 抽取。

目前企业对数据的实时性要求越来越高，比如实时监控一个实时的活动效果，并根据效果进行不同策略的营销手段，保持活动的效果。

3、数据仓库是不更新的，数据仓库反应的是一段相当长的时间内的数据内容，主要的操作集中在数据查询上。 一般数据结果计算出来之后，特别是明细数据，会存放在关系数据库中，因为主流的报表工具都支持数据库。

对数据库的查询，最基本的操作是创建索引，比如 300 万的数据根据手机号查询需要十几秒，创建 btree 索引之后，需要几十毫秒。

4、数据仓库中的数据是随着时间而变化的。

3、数据仓库的发展历程

第一阶段：简单报表阶段

解决日常工作中业务人员需要的报表，为领导生成简单的汇总数据

大数据库+前段报表的形式

第二阶段：数据集市阶段

按照不同部门、不同业务人员的需要，进行一定的数据采集，整理，并进行多维度报表的展现，能够提供对特定业务指导。

业务部门对数据的口径不一致，产生的汇总数据不一致。对大领导来看，就需要一个标准的口径。

第三阶段：数据仓库阶段

对整个企业的数据进行采集，整理，并且能够按照各个业务部门的需要，提供跨部门的，完全一致的业务报表数据，能够通过数据仓库生成对业务具有指导性的数据，同为为领导决策提供全面的数据支持。

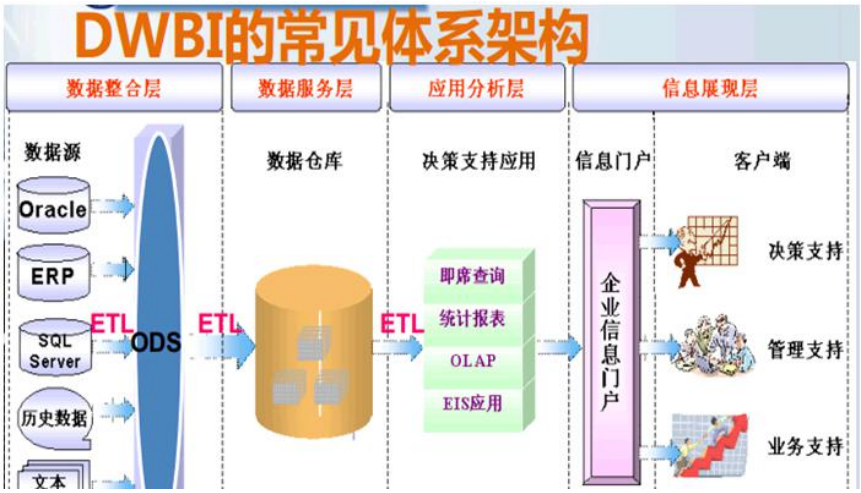
数据仓库和数据集市的区别，在于数据模型的支持。沉默用户的计算，没有沉默字段的标识，需要些复杂的 sql，有沉默字段的话，一个简单的 sql 就能搞定。

其他：城市商品表

4、数据库与数据仓库的区别

1、数据仓库是集成的，数据库为单一的业务提供服务。

2、BI 结构：数据整合层、数据服务层、应用分析层、信息展现层



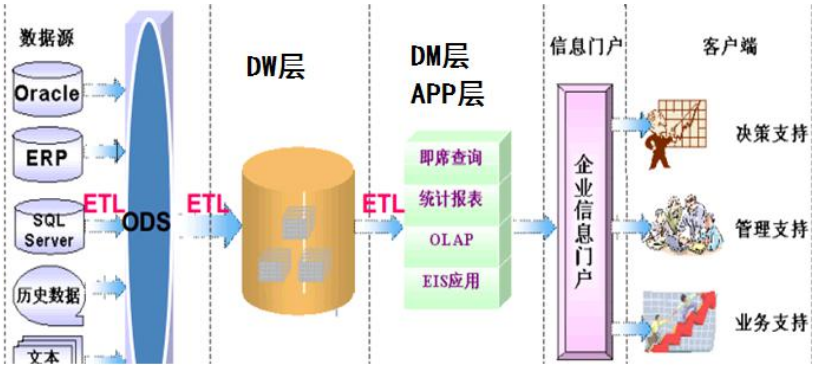
3、数据层库结构

ODS(临时存储层)，一般都是贴源设计、业务数据库是什么，ODS 层就是什么

PDW(数据仓库层)，将年月日，拆分成年、月、日字段，一般采用 Int 类型;通过 ODS 层到 DW 层的 etl 脚本对数据进行数据清洗，进行设计。分部门根据业务需求进行设计。如果没有业务需求，就根据源系统的表结构和自己建模经验去处理。

DM(数据集市层)，维度建模，星形模型，雪花模型。需要什么数据就去拉取什么数据。

APP（应用层），报表展现，需要的数据，与 DM 层处于同一级别。



4、ODS 层分为增量更新或者全量更新；PDW 层一致的、准确的、干净的数据，一般遵循数据库三范式设计。

11、为什么数据仓库需要分层？

为什么要对数据仓库分层：

1 用空间换时间，通过大量的预处理来提升应用系统的用户体验（效率），因此数据仓库会存在大量冗余的数据；

2 如果不分层的话，如果源业务系统的业务规则发生变化将会影响整个数据清洗过程，工作量巨大

3 通过数据分层管理可以简化数据清洗的过程，因为把原来一步的工作分到了多个步骤去完成，相当于把一个复杂的工作拆成了多个简单的工作，把一个大的黑盒变成了一个白盒，每一层的处理逻辑都相对简单和容易理解，这样我们比较容易保证每一个步骤的正确性，当数据发生错误的时候，往往我们只需要局部调整某个步骤即可。

5、数据质量检查

保证报表数据的正确性、稳定性，通过告警机制尽可能快的发现异常、尽可能快的解决问题。出错的次数太多之后，领导会对你失去信心，该辞职了。

检查方法：

1、 数据行数据的比较

2、 行数有变化，但是指标有变化。对领导关系的重点指标进行筛选。

一级类目	报表名称	指标	判断原则	
管理层	补贴报表	总补贴	>0	总补贴＝优惠券补贴＋商品补贴＋满减活动补贴
管理层	补贴报表	总盈亏	<>0	总盈亏＝商品铺货毛利 - 总补贴
管理层	补贴报表	商品补贴	>0	铺货到小店的鲜蜂商品：商品补贴＝（商品返款价 - 商品售价）*商品销量；非铺货到小店的
管理层	城市kpi周报	铺货率	>0and<1	铺货率为每日15点每城市的铺货率的周均值，15点每城市的铺货率＝15点所有金牌小店的
管理层	城市kpi周报	接单率	>0and<1	日接单率的周均值，日接单率＝每日创建订单在15分钟内商户接单/创建订单量
管理层	城市kpi周报	退货率	>0and<1	日退货率的周均值，日退货率＝每日所有小店的退货金额/每日所有小店的铺货金额
管理层	城市考核指标	延迟率	>=，不全为0，按行检查	当日延迟且完成的订单量/当日完成订单量，当日完成订单量，排除所有到店自提的订单。
管理层	城市考核指标	完成订单量	>=，不全为0，按行检查	当日完成的订单量
管理层	城市考核指标	下单客户数	>=，不全为0，按行检查	当日下单的用户数（完成状态）
管理层	城市考核指标	15分钟接单率	>=，不全为0，按行检查	十五分钟内成功订单量/系统派单量（系统派单量=发送订单量+拒绝订单量）
管理层	销售中心KPI周报/月报	新增金牌店数	>=0，不全为0，不为空	在T周成为金牌门店（可借鉴门店留存报表）
管理层	销售中心KPI周报/月报	城市动销商品数	>=0，不全为0，不为空	本周之内有销量的自营商品SKU数，剔除无铺货额的SKU数，只统计父商品中计算，
管理层	销售中心KPI周报/月报	总铺货额（万元	>=0，不全为0，不为空	自然周内的累计门店收货金额， 按照小店实际到货统计，按到店的铺货计算(排除次日达，
管理层	销售中心KPI周报/月报	线下补贴额（万元）	>=0，不全为0，不为空	线上订单的优惠券补贴+线上订单的满减活动补贴+线上秒杀订单补贴， 自然周内累计所有

数据行数质量检查不合格的数据表如下：

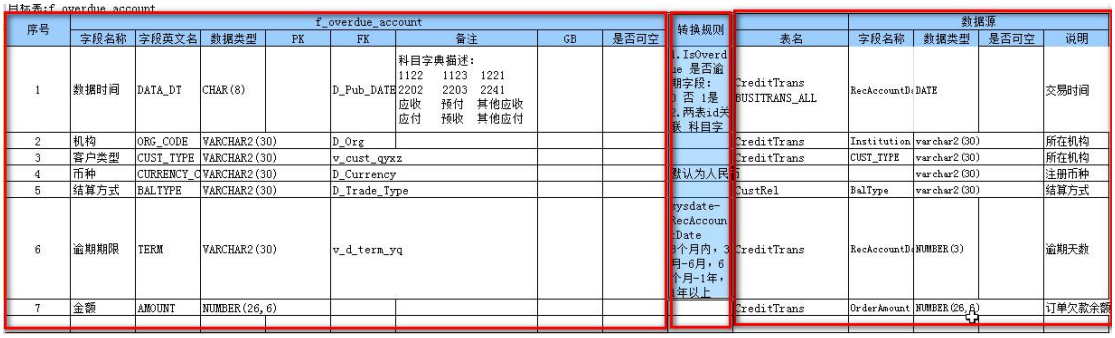
2016-01-28	t_order_goods_chk	0	2016-01-27	0	罗卫	检查订单商品表异常数据行数为0正常，大于0不正常。
2016-01-28	rpt_total_kpi	1899	2016-01-27	1899	罗卫	周报数据表，每周才会有数据变化

在领导发现问题之前，解决问题。

告警

6、元数据管理

元数据：数据的数据，记录数据从哪里到哪里去，中间如何转化的。

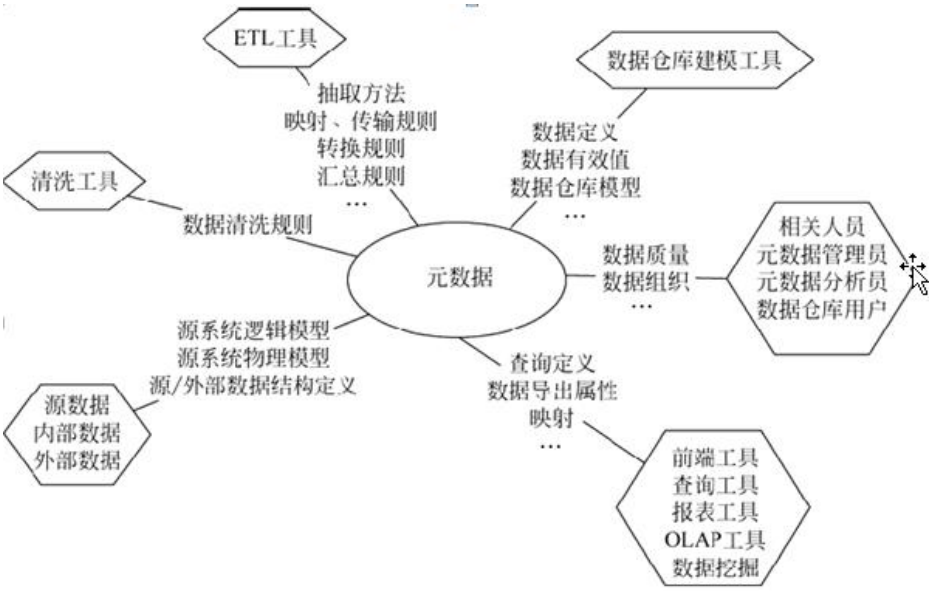


序号	f_overdue_account								转换规则	数据源				
	字段名称	字段英文名	数据类型	PK	FK	备注	GB	是否可空		表名	字段名称	数据类型	是否可空	说明
1	数据时间	DATA_DT	CHAR(8)		D_Pub_DATE	科目字典描述： 1122 1123 1221 2202 2203 2241 应收 预付 其他应收 应付 预收 其他应付			1.IsOverd e 是否逾 期字段： 0 否 1是 2.两表id关 联 科目字	CreditTrans BUSITRANS_ALL	RecAccountD	DATE		交易时间
2	机构	ORG_CODE	VARCHAR2(30)		D_Org					CreditTrans	Institution	varchar2(30)		所在机构
3	客户类型	CUST_TYPE	VARCHAR2(30)		v_cust_qyxz					CreditTrans	CUST_TYPE	varchar2(30)		所在机构
4	币种	CURRENCY_C	VARCHAR2(30)		D_Currency				默认为人民币			varchar2(30)		注册币种
5	结算方式	BALTYPE	VARCHAR2(30)		D_Trade_Type					CustRel	BalType	varchar2(30)		结算方式
6	逾期期限	TERM	VARCHAR2(30)		v_d_term_yq				ysdate- ecAccoun Date 个月内，3 月-6月，6 个月-1年， 1年以上	CreditTrans	RecAccountD	NUMBER(3)		逾期天数
7	金额	AMOUNT	NUMBER(26,6)							CreditTrans	OrderAmount	NUMBER(26,6)		订单欠款余额

元数据分为技术元数据和业务元数据

元数据可分为技术元数据和业务元数据。技术元数据为开发和管理数据仓库的 IT 人员使用，它描述了与数据仓库开发、管理和维护相关的数据，包括数据源信息、数据转换描述、数据仓库模型、数据清洗与更新规则、数据映射和访问权限等。而业务元数据为管理层和业务分析人员服务，从业务角度描述数据，包括商务术语、数据仓库中有什么数据、数据的位置和数据的可用性等，帮助业务人员更好地理解数据仓库中哪些数据是可用的以及如何使用。

元数据中的数据都有哪些？



7、数据仓库命名规范

8、缓慢变化维

维度建模的数据仓库中，有一个概念叫 Slowly Changing Dimensions，中文一般翻译成“缓慢变化维”，经常被简写为 SCD。缓慢变化维的提出是因为在现实世界中，维度的属性并不是静态的，它会随着时间的流失发生缓慢的变化。这种随时间发生变化的维度我们一般称之为缓慢变化维，并且把处理维度表的历史变化信息的问题称为处理缓慢变化维的问题，有时也简称为处理 SCD 的问题。

如何解决缓慢变化维带来的影响？

第一种方法，直接在原来维度的基础上进行更新,不会产生新的记录：

1） 更新前：

emp_rid(代理键)	emp_id(自然键)	emp_name	position
101212	12345	Jack	Developer

更新后：

emp_rid(代理键)	emp_id(自然键)	emp_name	position
101212	12345	Jack	Manager

上图中 position 有变化

第二种方法，不修改原有的数据，重新产生一条新的记录,这样就可以追溯所有的历史记录：

1） 更新前：

emp_rid(代理键)	emp_id(自然键)	emp_name	position	start_date	end_date
101212	12345	Jack	Developer	2010-2-5	2012-6-12

更新后：

emp_rid(代理键)	emp_id(自然键)	emp_name	position	start_date	end_date
201245	12345	Jack	Manager	2012-6-12	

上图中多了一条记录

第三种方法，直接在原来维度的基础上进行更新,不会产生新的记录但是只会记录上一次的历史记录：

1） 更新前：

emp_rid(代理键)	emp_id(自然键)	emp_name	position	old_position
101212	12345	Jack	Developer	null

更新后：

emp_rid(代理键)	emp_id(自然键)	emp_name	position	old_position
101212	12345	Jack	Manager	Developer

多一个字段，用来存放以前的 position

9、数据仓库建模

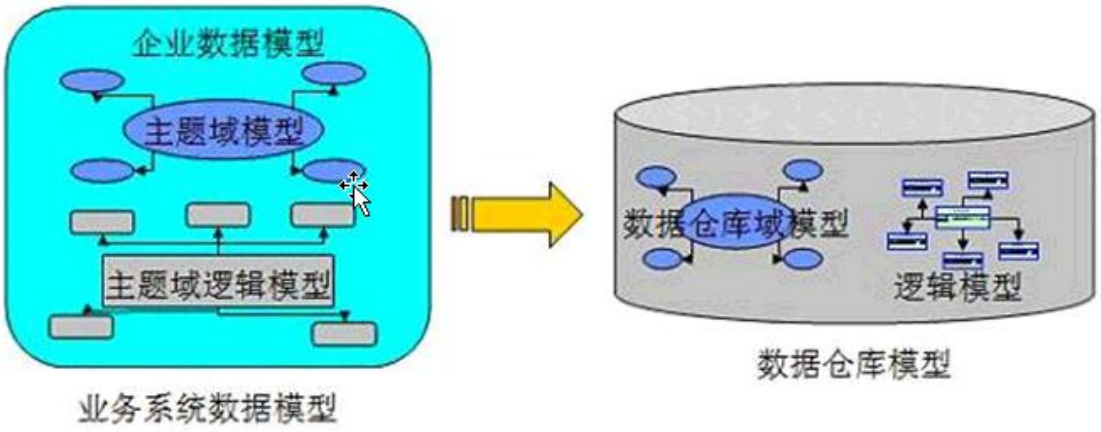
数据仓库模型的层次划分

业务建模，生成业务模型，主要解决业务层面的分解和程序化。

领域建模，生成领域模型，主要是对业务模型进行抽象处理，生成领域概念模型。

逻辑建模，生成逻辑模型，主要是将领域模型的概念实体以及实体之间的关系进行数据库层次的逻辑化。

物理建模，生成物理模型，主要解决，逻辑模型针对不同关系型数据库的物理化以及性能等一些具体的技术问题。



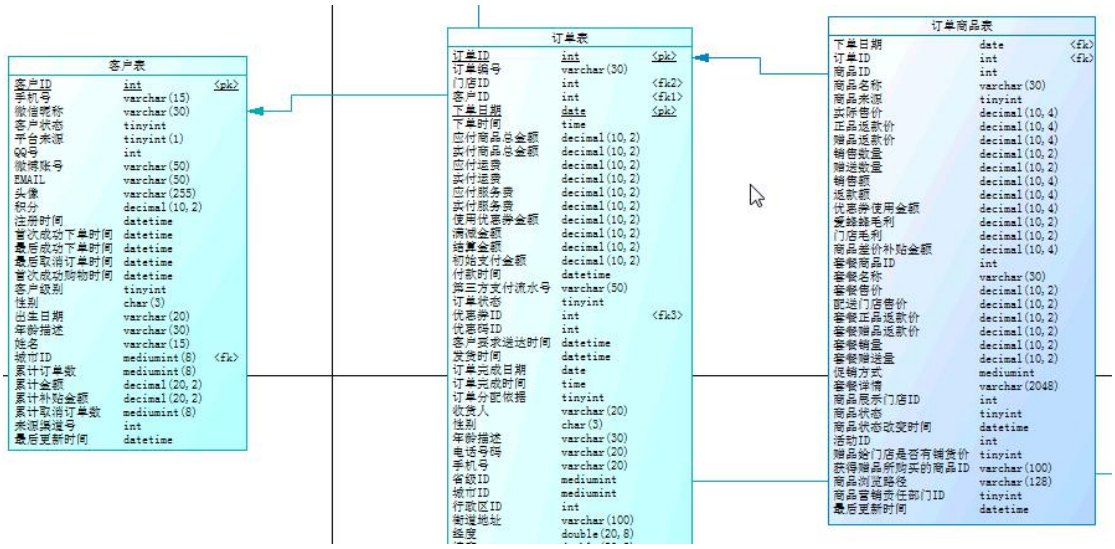
因此，**在整个数据仓库的模型的设计和架构中，既涉及到业务知识，也涉及到了具体的技术，我们既需要了解丰富的行业经验，同时，也需要一定的信息技术来帮助我们实现我们的数据模型，最重要的是，我们还需要一个非常适用的方法论，来指导我们自己针对我们的业务进行抽象，处理，生成各个阶段的模型。**

10、数据仓库五大核心模块

- **系统记录域（System of Record）**：这部分是主要的数据仓库业务数据存储区，数据模型在这里保证了数据的一致性。
- **内部管理域（Housekeeping）**：这部分主要存储数据仓库用于内部管理的元数据，数据模型在这里能够帮助进行统一的元数据的管理。
- **汇总域（Summary of Area）**：这部分数据来自于系统记录域的汇总，数据模型在这里保证了分析域的主题分析的性能，满足了部分的报表查询。
- **分析域（Analysis Area）**：这部分数据模型主要用于各个业务部分的具体的主题业务分析。这部分数据模型可以单独存储在相应的数据集市中。
- **反馈域(Feedback Area)**：可选项，这部分数据模型主要用于相应前端的反馈数据，数据仓库可以视业务的需要设置这一区域。

11、实体建模和维度建模

12、O2O 业务建模案例




```
ORDER_ID              int comment '订单ID',
ORDER_NO              varchar(30)  comment
'订单编号（唯一字段），前缀字符表示订单来源：a，Andriod；b，微博；c，WEB；e，饿了么；i，Iphone；m，Mobile；x，微信；
z，中粮我买网；1，其它。 接着3位数字代表订单城市编号；接着字符z与后面的真正订单编号分隔。这套机制从2014年12月开始实施。',
DEALER_ID             int comment '门店ID',
CUST_ID               int comment '客户ID',
ORDER_DATE            string  comment '下单日期',
ORDER_TIME            string  comment '下单时间',
PAYABLE_AMOUNT        decimal(10,2) comment '应付商品总金额',
REAL_AMOUNT           decimal(10,2) comment '实付商品总金额：应付商品总金额 - 促销优惠金额',
PAYABLE_FREIGHT       decimal(10,2) comment '应付运费',
REAL_FREIGHT          decimal(10,2) comment '实付运费',
EXPECT_SERVICE_FEE    decimal(10,2) comment '应付服务费，如果  EXPECT_SERVICE_FEE > SERVICE_FEE ，则要给门店补贴服务费差
(EXPECT_SERVICE_FEE - SERVICE_FEE)',
SERVICE_FEE           decimal(10,2) comment '实付服务费',
PROM_VALUE            decimal(10,2) comment '促销优惠金额，主要是优惠券抵扣金额',
FULLCUT               decimal(10,2) comment '满减金额，如满59元减5元',
SETTLEMENT_AMOUNT     decimal(10,2) comment '结算金额，客户最终或实际支付的金额。对应hs_order表order_amount',
INITIAL_PAY_AMOUNT    decimal(10,2) comment
'初始支付金额：对应hs_order表user_pay_amount。记录订单创建时的order_amount，如果客服改单后order_amount可能会发生变化，但user_
pay_amount不变',
PAY_TIME              string comment '付款时间',
THIRD_PAY_SEQUENCE    varchar(50) comment '第三方支付流水号：空值NULL表示现金支付、 前缀A-表示支付宝支付、T-表示财付通支付；',
ORDER_STATUS          tinyint comment '订单状态：  1，生成订单；   2，确认订单；3，取消订单；4，作废订单；5，完成订单；
6，无法配送。',
COUPON_ID             int comment '优惠券ID',
PROM_CODE_ID          int comment '优惠码ID',
SPECIFY_TIME          string comment '客户要求送达时间：对应hs_order表accept_time 预订单为客户设定时段的结束时间。',
```