

BX代码逻辑梳理

DWD只是将ODS表经过降维形成一张宽表，逻辑略，这里主要介绍DWD综合信息表 ==》ADS数据应用表 的转换

1. ODS来源表

1.1. A单

- bosch_accessories_recycling——A单配件回收基本信息表
- bosch_maintenance_station——A维修站基本信息表
- bosch_accessories_recycling_log——A 单状态变更表
- bosch_accessories_white_list——配件回收白名单

1.2. B单

- bosch_dealer——B经销商基本信息表

2. DWD综合信息表

序号	字段名称	字段类型	描述	关联信息
0	workshop_request_id	string	维修站申请单号	-
1	workshop_id	string	维修站id	-
2	workshop_name	string	维修站名称	-
3	extra_id	string	易加网id	-
4	workshop_province	string	维修站所在省份	-
5	workshop_city	string	维修站所在城市	-
6	workshop_tag	string	维修站分类	-
7	workshop_role	string	CV, BX, CV&BX	-
8	workshop_syn_code	string	维修站编号	-
9	upload_bosch	tinyint	是否提交到博世总部	-
10	distributor_id	string	经销商id	-
11	distributor_name	string	经销商名称	-

序号	字段名称	字段类型	描述	关联信息
12	parent_distributor_id	string	父经销商id	-
13	is_dealers	tinyint	总经销商(0: 不是, 1: 是)	-
14	serial_num	string	A单单号	-
15	order_id	string	物流单号	-
16	bx_item_no	string	bx_料号	-
17	bosch_item_no	string	bosch可接受的料号	-
18	bx_core	string	bx_core号	-
19	bx_core_classification	string	bx_core分类	-
20	bx_core_type	string	配件型号	-
21	bx_core_classification_description	string	bx_core分类描述	-
22	is_arrive	tinyint	是否到达, 1到达, 0未到达	-
23	status	string	旧件回收状态	-
24	status_description	string	当前状态对应消息	-
25	order_create_time	timestamp	订单创建时间	-
26	order_occur_time	timestamp	状态发生时间	-
27	order_update_time	timestamp	订单更新时间	-
28	etl_load_time	string	数据加载时间	

3. ADS数据应用输出表

3.1. 计算到达状态十三的平均时间表

——ads_wks_bx_core_recycling_avg_time_als_df

序号	字段名称	字段类型	描述	关联信息
0	status	int	旧件回收状态	-
1	status_description	string	当前状态对应消息	-
2	avg_day_diff	float	每个状态对应平均时间	-

3.1.1. 辅助临时表

——ads_wks_bx_core_recycling_tmp01_df

序号	字段名称	字段类型	描述	关联信息
0	workshop_request_id	string	维修站申请单号	-
1	workshop_syn_code	string	维修站编号	-
2	is_arrive	tinyint	是否到达	-
3	status	int	旧件回收状态	-
4	status_description	string	当前状态对应消息	-
5	order_create_time	timestamp	订单创建时间	-
6	order_occur_time	timestamp	订单更新时间	-
7	time_union	timestamp	订单时间集合	-

3.1.1.1. 逻辑分析

- 从综合信息表里抽取
 - workshop_request_id——维修站申请单号
 - workshop_syn_code——维修站编号
 - is_arrive——是否到达，1到达，0未到达
 - STATUS——旧件回收状态
 - statue_info——根据回收状态编号，生成描述字段 ==》staus_description
 - order_create_time——订单创建时间
 - order_occur_time——状态发生时间 ==》订单更新时间
 - time_union——状态发生时间 ==》订单时间集合

3.1.1.2. 具体代码

```
INSERT OVERWRITE TABLE ads_wks_bx_core_recycling_tmp01_df PARTITION(ds =
${bdp.system.bizdate})
SELECT
    workshop_request_id,
    workshop_syn_code,
    is_arrive,
    STATUS,
    CASE
        STATUS
        WHEN 0 THEN '维修站申请已创建'
        WHEN 1 THEN '维修站申请已提交'
        WHEN 2 THEN '经销商审核通过'
        WHEN 3 THEN '维修站已发货'
        WHEN 4 THEN '经销商已收货'
        WHEN 5 THEN '经销商审核未通过，旧件回收自动关闭'
        WHEN 6 THEN '经销商申请已提交'
        WHEN 7 THEN '旧件在运往博世仓库途中'
        WHEN 8 THEN '博世仓库已收货'
```

```

        WHEN 9 THEN '博世质检报告已发布/已更新'
        WHEN 10 THEN '经销商已撤销'
        WHEN 11 THEN '所属提单退件处理中'
        WHEN 12 THEN '博世客服中心已退件'
        WHEN 13 THEN '旧件回收已完成或关闭'
        WHEN 20 THEN '维修站撤回申请, 旧件回收自动关闭'
        WHEN 21 THEN '订单已被关闭'
        WHEN 22 THEN '经销商终止请求'
        WHEN 23 THEN '提单中包含不合格旧件'
        WHEN 24 THEN '经销商判定不合格'
        WHEN 25 THEN '经销商判定合格'

    END AS status_info,
    order_create_time,
    order_occur_time,
    order_occur_time AS time_union
FROM
    dwd_wks_bx_core_recycling_df
WHERE
    ds = '${bdp.system.bizdate}';

```

3.1.2. 计算每个状态对应的平均时间

——ads_wks_bx_core_recycling_avg_time_als_df（状态13平均时间表）

3.1.2.1. 逻辑分析

Step1: 从ads临时表中抽取（表a）：

- workshop_request_id——维修站申请单号
- workshop_syn_code——维修站编号
- STATUS——旧件回收状态
- staus_description——当前状态对应消息
- order_create_time——订单创建时间
- time_union——状态发生时间 ==》订单时间集合
- is_arrive——是否到达，1到达，0未到达
- time_diff——根据维修站单号分组，状态升序排序后，将time_union字段上面推下来一个形成一个时间差值字段准备（状态12的更新时间）

Step2: 右连接（表b）——限制条件：已到达、分区为最新

- 从ads临时表中抽取 [维修站申请单号](#)
- 限制条件：已到达、状态发生时间 >= '2022-09-29'、旧件状态为13、分区为最新

Step3: 从上面联合的表中抽取（表c）——限制条件：维修站编号中间有“IWSD”或者“76”这样的字眼

- workshop_request_id——维修站申请单号（去重）
- workshop_syn_code——维修站编号
- STATUS——旧件回收状态
- staus_description——当前状态对应消息
- order_create_time——订单创建时间
- time_union——状态发生时间 ==》 订单时间集合
- is_arrive——是否到达，1到达，0未到达
- time_diff——计算 13状态发生的时间 & 下一个状态时间12 的差值，形成真正的time_diff

Step4: 表D，从表c中拉取所有字段——限制条件： 订单创建时间>='2022-09-28' & 旧件回收状态不为24

Step5: 表e，从表d中抽取字段——按照旧件回收状态升序排序

- STATUS——旧件回收状态
- staus_description——当前状态对应消息
- day_diff——按照 status 分组，取day_diff平均值，保留2位小数 ==》 avg_day_diff

3.1.2.2. 具体代码

```
-- *****
-- 所属主题:维修站域
-- 功能描述: 自然日平均时间计算
-- 创建者:Fixed-term Shi Yue
-- 创建日期:2022-10-24
-- 修改日期  修改人  修改内容
-- yyyymmdd  name  comment
-- *****

-- 状态已经到达13的订单，自然日平均时间计算，2022年9月29日之后完成的订单
INSERT OVERWRITE TABLE ads_wks_bx_core_recycling_avg_time_als_df PARTITION(ds =
${bdp.system.bizdate})
SELECT
    DISTINCT STATUS,
    status_description,
    round(avg(day_diff) OVER(PARTITION by STATUS), 2) AS avg_day_diff
FROM
    (
        SELECT
            workshop_request_id,
            workshop_syn_code,
            STATUS,
            status_description,
            order_create_time,
```

```

time_union,
time_diff,
is_arrive,
day_diff
FROM
(
    SELECT
        distinct workshop_request_id,
        workshop_syn_code,
        STATUS,
        status_description,
        order_create_time,
        time_union,
        time_diff,
        is_arrive,
        datediff(time_union, time_diff) AS day_diff
    FROM
        (
            SELECT
                a.workshop_request_id,
                a.workshop_syn_code,
                a.STATUS,
                a.status_description,
                a.order_create_time,
                a.time_union,
                a.is_arrive,
                lag(a.time_union, 1, a.time_union) OVER(
                    PARTITION by a.workshop_request_id
                    ORDER BY
                        STATUS ASC
                ) AS time_diff
            FROM
                ads_wks_bx_core_recycling_tmp01_df AS a
            RIGHT JOIN (
                SELECT
                    workshop_request_id
                FROM
                    ads_wks_bx_core_recycling_tmp01_df
                WHERE
                    is_arrive = 1 and order_occur_time >= '2022-09-29'
                AND STATUS = 13 AND ds = '${bdp.system.bizdate}'
                GROUP BY workshop_request_id
            ) b ON a.workshop_request_id = b.workshop_request_id
            WHERE
                is_arrive = 1 AND ds = '${bdp.system.bizdate}'

        ) c
        WHERE
            workshop_syn_code LIKE '%IWSD%' OR workshop_syn_code LIKE '%76%'
        ) d
    where order_create_time >= '2022-09-28' and STATUS != 24

```

```
) e
ORDER BY
STATUS ASC;
```

3.2. 旧件每天订单创建统计表

——ads_wks_bx_core_recycling_order_create_num_count_day_df

序号	字段名称	字段类型	描述
0	day	date	订单创建日期
1	order_create_num	int	订单创建数量

3.2.1. 逻辑分析

Step1: （表a）从ads_wks_bx_core_recycling_tmp01_df抽取字段——限制条件：已到达、旧件回收状态为0，分区取最新

ads_wks_bx_core_recycling_tmp01_df结构为：

序号	字段名称	字段类型	描述	关联信息
0	workshop_request_id	string	维修站申请单号	-
1	workshop_syn_code	string	维修站编号	-
2	is_arrive	tinyint	是否到达	-
3	status	int	旧件回收状态	-
4	status_description	string	当前状态对应消息	-
5	order_create_time	timestamp	订单创建时间	-
6	order_occur_time	timestamp	订单更新时间	-
7	time_union	timestamp	订单时间集合	-

- order_create_time——订单创建时间，单取日 ==》 day
- status——旧件回收状态
- status_description——状态描述

Step2: 从表a中取出字段——限制条件：day>='2022-01-01'、按照day升序排序

- day——日
- status——按照day分组，统计每个day的status次数为订单量 ==》 order_create_num

3.2.2. 具体代码

```
--统计每天创建订单量
INSERT OVERWRITE TABLE ads_wks_bx_core_recycling_order_create_num_count_day_df
PARTITION(ds = ${bdp.system.bizdate})
SELECT
    DAY,
    count(STATUS) as order_create_num
FROM
    (
        SELECT
            DATE(order_create_time) AS DAY,
            STATUS,
            status_description
        FROM
            ads_wks_bx_core_recycling_tmp01_df
        WHERE
            is_arrive = 1 AND STATUS = 0 AND ds = '${bdp.system.bizdate}'
    ) a
WHERE
    DAY >= '2022-01-01'
GROUP BY
    DAY
ORDER BY
    DAY ASC
```

3.3. 旧件订单城市维度创建数量统计表

——ads_wks_bx_core_recycling_city_count_df （城市，维修站热度图）

序号	字段名称	字段类型	描述
0	workshop_province	string	维修站所在省份
1	workshop_city	string	维修站所在城市
2	city_count	int	次数统计

3.3.1. 逻辑分析

Step1: 从dwd综合信息表中抽取字段（表a）——限制条件：省份不为空、回收状态为0，分区取最新

- workshop_request_id——维修站申请单号，按此去重
- workshop_id——维修站id
- workshop_name——维修站名称
- workshop_syn_code——维修站编号
- workshop_province——所在省份
- workshop_city——所在城市
- STATUS——旧件回收状态（订单记录）

Step2: 从表a中抽取维修站 **省份** 和 **城市** ——限制条件：维修站编号中出现“IWSD”、“76”、“YNBX”等字眼、按照城市数量降序排序

- workshop_city——按照省份和城市分组，计算每个城市出现的次数 ==> city_count

3.3.2. 具体代码

```
INSERT OVERWRITE TABLE ads_wks_bx_core_recycling_city_count_df PARTITION(ds =
${bdp.system.bizdate})
SELECT
    workshop_province,
    workshop_city,
    count(workshop_city) AS city_count
FROM
    (
        SELECT
            DISTINCT workshop_request_id,
            workshop_id,
            workshop_name,
            workshop_syn_code,
            workshop_province,
            workshop_city,
            STATUS
        FROM
            dwd_wks_bx_core_recycling_df
        WHERE
            workshop_province IS NOT NULL AND STATUS = 0 AND ds =
            '${bdp.system.bizdate}'
    ) a
WHERE
    workshop_syn_code LIKE '%IWSD%' OR workshop_syn_code LIKE '%76%' OR
    workshop_syn_code LIKE '%YNBX%'
GROUP BY
```

```
workshop_province,  
workshop_city  
ORDER BY  
city_count DESC
```

3.4. 旧件订单维修站维度创建数量统计表

——ads_wks_bx_core_recycling_workshop_count_df（城市、维修站热度图）

序号	字段名称	字段类型	描述	关联信息
0	workshop_name	string	维修站名称	-
1	workshop_syn_code	string	维修站编号	-
2	workshop_province	string	维修站所在省份	-
3	workshop_city	string	维修站所在城市	-
4	workshop_count	int	次数统计	-

3.4.1. 逻辑分析

和上面ads_wks_bx_core_recycling_city_count_df 代码结构相同

比之多抽取维修站名称、维修站编号，按照省份、城市、维修站分组，以维修站名计算的维修站次数

3.4.2. 具体代码

```
INSERT OVERWRITE TABLE ads_wks_bx_core_recycling_workshop_count_df PARTITION(ds =  
${bdp.system.bizdate})  
SELECT  
    workshop_name,  
    workshop_syn_code,  
    workshop_province,  
    workshop_city,  
    count(workshop_name) AS workshop_count  
FROM  
    (  
        SELECT  
            DISTINCT workshop_request_id,  
            workshop_id,  
            workshop_name,  
            workshop_syn_code,  
            workshop_province,  
            workshop_city,  
            STATUS  
        FROM  
            dwd_wks_bx_core_recycling_df  
        WHERE
```

```

        workshop_province IS NOT NULL AND STATUS = 0 AND ds =
        '${bdp.system.bizdate}'
    ) a
WHERE
    workshop_syn_code LIKE '%IWSD%' OR workshop_syn_code LIKE '%76%' OR
    workshop_syn_code LIKE '%YNBX%'
GROUP BY
    workshop_name,
    workshop_syn_code,
    workshop_province,
    workshop_city
ORDER BY
    workshop_count DESC

```

3.5. 旧件订单最新状态统计表

——ads_wks_bx_core_recycling_latest_status_df

序号	字段名称	字段类型	描述	关联信息
0	latest_status	string	状态	-
1	status_description	string	状态描述	-
2	cu_status_sta	string	状态统计	-

3.5.1. 逻辑分析

Step1: （表a）从dwd综合信息表中抽取字段——限制条件：已到达、分区取最新

- workshop_request_id——维修站申请单号
- workshop_syn_code——维修站编号
- status——旧件回收状态，转int类型
- status_description——当前的状态对应的信息
- is_arrive——是否到达

Step2: （表b）从表a中抽取 维修站申请单号、维修站编号、旧件回收状态、当前状态对应的信息

限制条件：维修站编号中有“IWSD”、“76”、“YNBX”字样

Step3: （表c）从表b中抽取

- workshop_request_id——维修站申请单号
- Status——旧件回收状态，按照申请单号分组，取最新状态 ==》latest_status

Step4: 从表c中抽取生成字段

- latest_status——最新状态，也按此排序
- cu_status_sta——由latest_status计算，按此分组，达到最新状态的申请单号个数
- status_description——由latest_status进行判断，展现具体描述信息

3.5.2. 具体代码

```
INSERT OVERWRITE TABLE ads_wks_bx_core_recycling_latest_status_df PARTITION(ds =
${bdp.system.bizdate})
SELECT
    latest_status,
    count(latest_status) AS cu_status_sta,
    CASE
        latest_status
        WHEN 0 THEN '维修站申请已创建'
        WHEN 1 THEN '维修站申请已提交'
        WHEN 2 THEN '经销商审核通过'
        WHEN 3 THEN '维修站(顺丰)已发货'
        WHEN 4 THEN '经销商已收货'
        WHEN 5 THEN '中心库审核未通过, 旧件回收自动关闭'
        WHEN 6 THEN '经销商申请已提交'
        WHEN 7 THEN '旧件已到达博世仓库'
        WHEN 8 THEN '博世仓库已验收'
        WHEN 9 THEN '博世质检报告已发布/已更新'
        WHEN 10 THEN '经销商已撤销'
        WHEN 11 THEN '所属提单退件处理中'
        WHEN 12 THEN '博世客服中心已退件'
        WHEN 13 THEN '旧件回收已完成或关闭'
        WHEN 20 THEN '维修站撤回申请, 旧件回收自动关闭'
        WHEN 21 THEN '订单已被关闭'
        WHEN 22 THEN '经销商终止请求'
        WHEN 23 THEN '提单中包含不合格旧件'
        WHEN 24 THEN '经销商判定合格'
        WHEN 25 THEN '经销商判定不合格'
        WHEN 101 THEN '主机厂审核通过'
        when 102 THEN '主机厂审核不通过, 旧件回收自动关闭'
    END AS status_description
FROM
    (
        SELECT
            workshop_request_id,
            max(STATUS) AS latest_status
        FROM
            (
                SELECT
```

```

        workshop_request_id,
        workshop_syn_code,
        STATUS,
        status_description
    FROM
    (
        SELECT
            workshop_request_id,
            workshop_syn_code,
            CAST(STATUS AS INT),
            status_description,
            is_arrive
        FROM
            dwd_wks_bx_core_recycling_df
        WHERE
            is_arrive = 1 AND ds = '${bdp.system.bizdate}'
    ) a
    WHERE
        workshop_syn_code LIKE '%IWSO%' OR workshop_syn_code LIKE '%76%' OR
        workshop_syn_code LIKE '%YNBX%'
    ) b
    GROUP BY
        workshop_request_id
    ) c
GROUP BY
    latest_status
ORDER BY
    latest_status

```

3.6. 旧件订单最新状态信息表

——ads_wks_bx_core_recycling_last_status_info_df

序号	字段名称	字段类型	描述	关联信息
0	workshop_request_id	string	维修站单号	-
1	workshop_id	string	维修站id	-
2	workshop_name	string	维修站名称	-
3	workshop_syn_code	string	维修站代码	-
4	serial_num	string	serial_num	-
5	bx_item_no	string	bx料号	-
6	bx_core	string	bx物料名	-
7	bx_core_classification	string	bx物料分类	-
8	bx_core_type	string	bx物料类型	-
9	bx_core_classification_description	string	bx物料类型描述	-
10	latest_status	string	最新订单状态	-

3.6.1. 逻辑分析

Step1: （表a）从dwd信息表中抽取字段——限制条件：已到达、分区取最新

- workshop_request_id——维修站申请单号
- workshop_id——维修站id
- workshop_name——维修站名称
- workshop_syn_code——维修站编号
- serial_num——A单单号
- bx_item_no——bosch可以接受的料号
- bx_core——bx_core号
- bx_core_classification——bx_core分类
- bx_core_type——配件类型
- bx_core_classification_description——bx_core分类描述
- is_arrive——是否到达

Step2: （表b）从表a中抽取 除 is_arrive 之外的所有字段

限制条件：维修站编号中有“IWSD”、“76”、“YNBX”字样

Step3: （表a1）从dwd综合信息表中抽取字段——限制条件：已到达、分区取最新

- workshop_request_id——维修站申请单号
- workshop_syn_code——维修站编号
- status——旧件回收状态，转int类型
- status_description——当前的状态对应的信息
- is_arrive——是否到达

Step4: （表b1）从表a1中抽取 维修站申请单号、维修站编号、旧件回收状态、当前状态对应的信息

限制条件：维修站编号中有“IWSD”、“76”、“YNBX”字样

Step5: （表c）从表b1中抽取

- workshop_request_id——维修站申请单号
- Status——旧件回收状态，按照申请单号分组，取最新状态 ==> latest_status

Step6: 将表b与表c进行左连接，取出表b里面的所有字段与表c里面的每个维修站申请单号的最新状态

- workshop_request_id——维修站申请单号

- workshop_id——维修站id
- workshop_name——维修站名称
- workshop_syn_code——维修站编号
- serial_num——A单单号
- bx_item_no——bosch可以接受的料号
- bx_core——bx_core号
- bx_core_classification——bx_core分类
- bx_core_type——配件类型
- bx_core_classification_description——bx_core分类描述
- latest_status——最新状态

3.6.2. 具体代码

```
INSERT OVERWRITE TABLE ads_wks_bx_core_recycling_last_status_info_df PARTITION(ds =  
${bdp.system.bizdate})  
SELECT  
    b.workshop_request_id,  
    b.workshop_id,  
    b.workshop_name,  
    b.workshop_syn_code,  
    b.serial_num,  
    b.bx_item_no,  
    b.bx_core,  
    b.bx_core_classification,  
    b.bx_core_type,  
    b.bx_core_classification_description,  
    c.latest_status  
FROM  
    (  
        SELECT  
            DISTINCT workshop_request_id,  
            workshop_id,  
            workshop_name,  
            workshop_syn_code,  
            serial_num,  
            bx_item_no,  
            bx_core,  
            bx_core_classification,  
            bx_core_type,  
            bx_core_classification_description  
        FROM  
            (  
                SELECT  
                    workshop_request_id,  
                    workshop_id,
```

```

        workshop_name,
        workshop_syn_code,
        serial_num,
        bx_item_no,
        bx_core,
        bx_core_classification,
        bx_core_type,
        bx_core_classification_description,
        is_arrive
    FROM
        dwd_wks_bx_core_recycling_df
    WHERE
        is_arrive = 1 AND ds = '${bdp.system.bizdate}'
    ) a
    WHERE
        workshop_syn_code LIKE '%IWSD%' OR workshop_syn_code LIKE '%76%' OR
workshop_syn_code LIKE '%YNBX%'
    ) b
LEFT JOIN
    (
        SELECT
            workshop_request_id,
            max(STATUS) AS latest_status
        FROM
            (
                SELECT
                    workshop_request_id,
                    workshop_syn_code,
                    STATUS,
                    status_description
                FROM
                    (
                        SELECT
                            workshop_request_id,
                            workshop_syn_code,
                            CAST(STATUS AS INT),
                            status_description,
                            is_arrive
                        FROM
                            dwd_wks_bx_core_recycling_df
                        WHERE
                            is_arrive = 1 AND ds = '${bdp.system.bizdate}'
                        ) a
                    ) b
                WHERE
                    workshop_syn_code LIKE '%IWSD%' OR workshop_syn_code LIKE '%76%' OR
workshop_syn_code LIKE '%YNBX%'
                ) b
            GROUP BY
                workshop_request_id
        ) c ON b.workshop_request_id = c.workshop_request_id

```


