

بنام خدا

پایگاه داده ۲

Design of Data Warehouse

بصیری

دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

مراجع

- Han, Jiawei, Micheline Kamber, and Data Mining. "Concepts and techniques." *Morgan Kaufmann* 340 (2006): 94104-3205.
- Kimball, Raiph. *The data warehouse toolkit*. John Wiley & Sons, 2006.
- Inmon, William H. *Building the data warehouse*. John wiley & sons, 2005.

Data Warehouse Design Process

- Top-down, bottom-up approaches or a combination of both
 - Top-down: Starts with overall design and planning (mature)
 - Bottom-up: Starts with experiments and prototypes (rapid)
- Typical data warehouse design process
 - Choose a **business process** to model, e.g., orders, invoices, etc.
 - Choose the **grain** (atomic level of data) of the business process
 - Choose the **dimensions** that will apply to each fact table record
 - Choose the **measure** that will populate each fact table record

grain : ریزدانگی داده

- ریزدانگی به این معنا است که تا چه سطحی لازم است ریز اطلاعات نمایش داده شود.
- با توجه به نیاز سازمان ریزدانگی باید تعیین شود.
- ریزدانگی در فکت و به تبع آن در بعدها نمایان می شود. مثلاً می خواهیم اطلاعات را به تفکیک روز ذخیره نماییم یا به تفکیک ماه.
- طبیعتاً ریزدانگی تاثیر مستقیمی روی حجم داده ذخیره شده خواهد داشت.

انواع جداول واقعیت (Fact Table)

Transaction Fact Table-

Periodic Snapshot Fact Table-

- وضعیت یک موجودیت اصلی را در دوره های زمانی مشخص ذخیره می کند.
- معمولاً لازم است فکتهای مختلفی با ریزدانگی های متفاوت زمانی از این نوع وجود داشته باشد، مثلاً روزانه و ماهانه

Accumulating Snapshot Fact Table-

Fact less Fact Table-

- برای ارتباط بین دو بعد استفاده می شود

انواع جداول واقعیت (Fact Table)

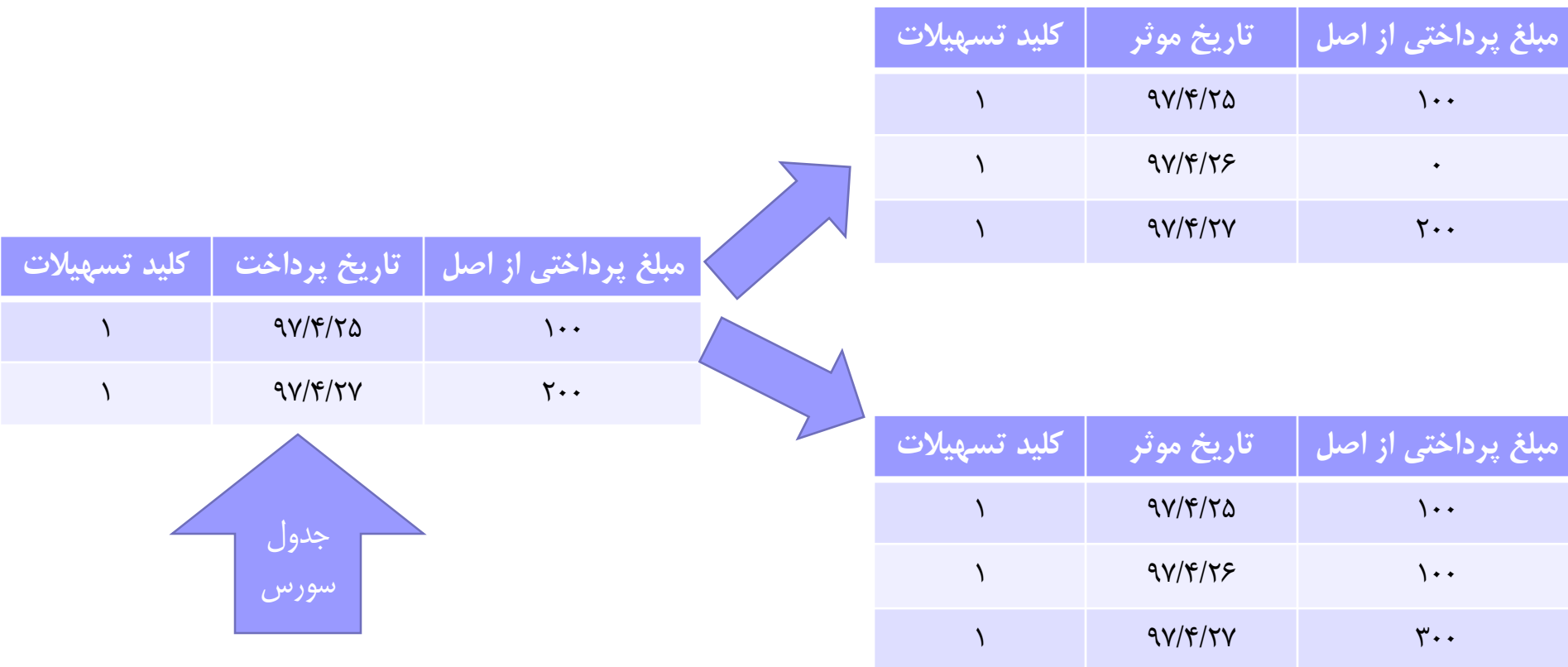
Temporal Snapshot Fact Table-

[https://sqlbits.com/\(X\(1\)S\(xkq3bqj54ulh1rmknteod-x3p\)\)/Sessions/Event10/Temporal_Snapshot_Fact_Table](https://sqlbits.com/(X(1)S(xkq3bqj54ulh1rmknteod-x3p))/Sessions/Event10/Temporal_Snapshot_Fact_Table)

Periodic Snapshot Fact Table

Periodic Snapshot Fact Table-

- برای مژره‌های جمع‌پذیر لازم است تصمیم‌گیری شود که به صورت جمع شده یا تراکنشی وارد شوند. مثل مبلغ پرداختی مشتری از اصل تسهیلات



FACT TABLE

- ❖ وجود کلید نال مربوط به یک بعد در فکت منطقی نیست
- ❖ معمولاً در صورت وجود یک رکورد با مقدار کلید ۱- در بعد وجود دارد که در فکت نیز ۱- برای آن بعد ثبت می شود. این رکورد نماینده مقدار نامشخص است.
- ❖ وجود مقادیر نال برای مژرها ممکن است.

Dimension

- وجود یک بعد به کاربرد آن بستگی دارد.
- ممکن است یک بعد را بتوان به عنوان ویژگی های یک بعد دیگر یا به عنوان یک بعد مستقل در نظر گرفت
 - ❖ این مساله به میزان استفاده از آن بعد بستگی دارد
 - ❖ مثلاً نوع سپرده را می توان یک بعد جداگانه در نظر گرفت یا به عنوان یک ویژگی در بعد سپرده

کلید Dimension

❖ Natural Key

- ❖ در صورتی یک کلید طبیعی ساده برای موجودیت مربوطه در بیزینس وجود دارد می توان از آن برای کلید بعد استفاده کرد.
- ❖ مثلاً برای بعد شعبه، می توان کد شعبه را در نظر گرفت.
- ❖ بهتر است کلید را از نوع عددی ایجاد نماییم.

❖ Surrogate Key

- ❖ معمولاً بهتر است برای هر بعد یک Surrogate Key در نظر گرفت.

سلسله مراتب در Dimension

- لزومی ندارد که تمام ویژگی های در سلسله مراتب باشند
- می توان چندین سلسله مراتب در یک بعد متصور شد.
 - شعبه، شهر، استان
 - شهر، استان
 - شعبه، استان

سلسله مراتب در Dimension

- سلسله مراتب نوع ۱ : برای هر سطح یک ویژگی در بعد داریم:
 - شعبه، شهر، استان
- سلسله مراتب نوع ۲: دو ویژگی داریم که رابطه پدر فرزندی دارند.
 - سرفصل حسابداری

کد سرفصل	کد سرفصل لایه بالاتر
10000	1000
1000	100
100	10
10	1
20000	2000
300	30

نگهداری تغییرات یک ویژگی در Dimension

Slowly Changing Dimension (SCD)

- ممکن است لازم باشد تغییرات یک یا چند ویژگی را در یک بعد نگهداری نماییم.
- مثلاً در بعد مشتری، می خواهیم در صورت تغییر شماره تلفن همراه شماره جدید را (نیز) داشته باشیم.
- برای این کار روشهای مختلفی وجود دارد
 - بروزرسانی
 - ایجاد رکورد جدید
 - ...

SCD نوع ۱

➤ در این روش در صورت مشاهده تغییر در جداول سورس، مقدار قبلی موجود در بعد را با مقدار جدید جایگزین می نماییم.

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	CA



Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	IL

SCD نوع ۲

- در این روش در صورت مشاهده تغییر در جداول سورس، رکورد جدیدی ایجاد می نماییم.
- در این روش وجود Surrogate Key در بعد الزامی است.
- سه فیلد برای تاریخ شروع، تاریخ پایان و مشخص کردن فعال بودن رکورد جاری، در بعد در نظر گرفته می شود.

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	CA



Supplier_Ke y	Supplier_Co de	Supplier_Na me	Supplier_St ate	Start_Date	End_Date	Current_Fla g
123	ABC	Acme Supply Co	CA	01-Jan- 2000	21-Dec- 2004	0
124	ABC	Acme Supply Co	IL	22-Dec- 2004		1

SCD نوع 3

- در این روش **فیلدهای اضافی** برای تغییرات احتمالی در مقادیر فیلد مربوطه در نظر گرفته می شود. مثلاً برای شماره تلفن **۳ ستون مختلف** در نظر گرفته می شود. در صورت وجود شماره جدید در ستون بعدی ثبت می شود.
- تاریخچه تغییرات را به **صورت محدود** ذخیره می کند.

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	CA

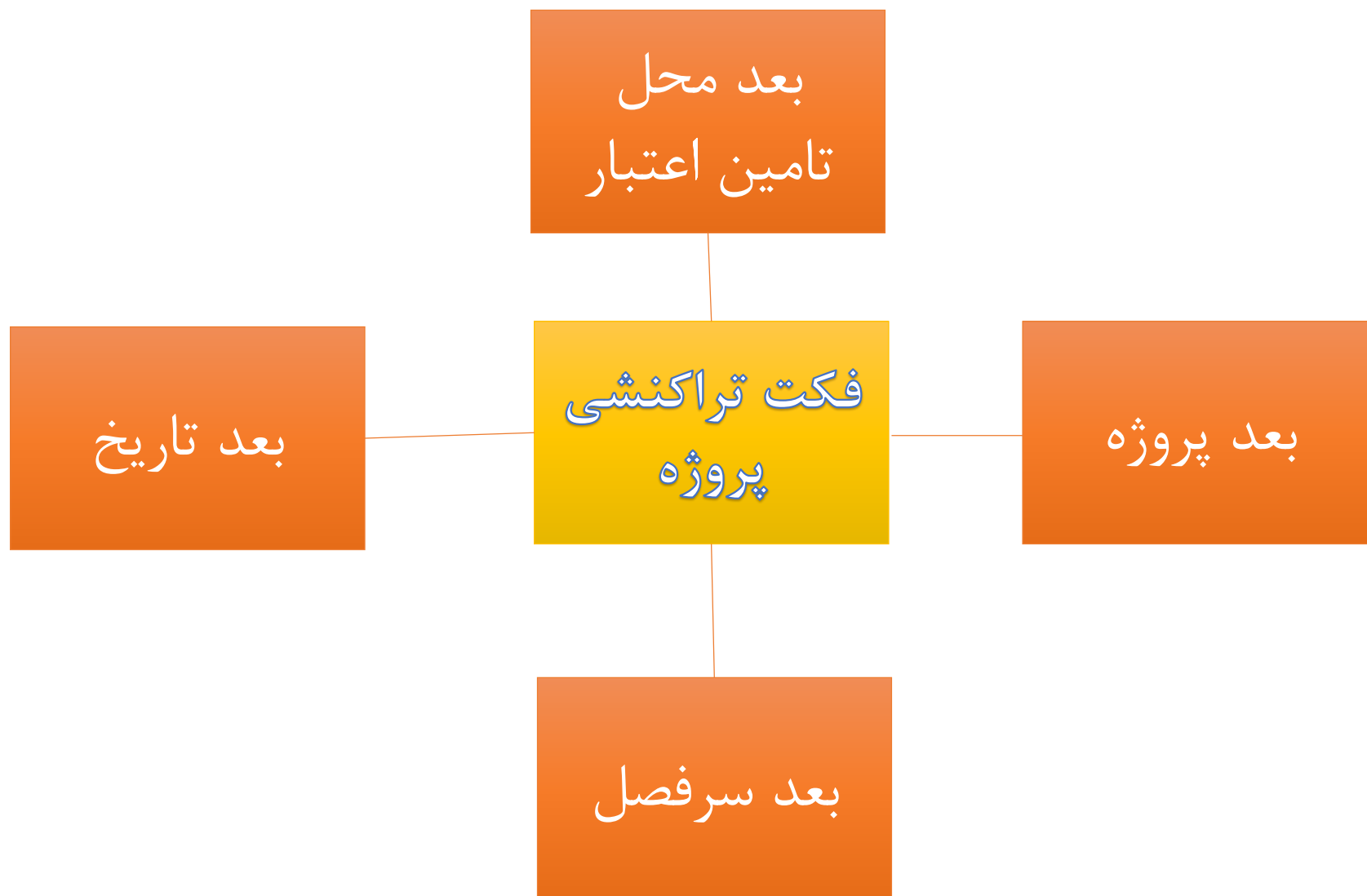


Supplier_Key	Supplier_Code	Supplier_Name	Original_Supplier_State	Effective_Date	Current_Supplier_State
123	ABC	Acme Supply Co	CA	22-Dec-2004	IL

سامانه مدیریت پروژه

- سامانه ای جهت کنترل و مدیریت پروژه‌های یک سازمان پروژه محور
- ثبت اطلاعات اولیه هر پروژه
- مانند نام پروژه، ناظر پروژه، تاریخ شروع و....
- ثبت مبالغ ورودی پروژه‌ها
- ثبت مبالغ هزینه شده برای هر پروژه با مشخص بودن نوع هزینه

مدل طراحی شده برای انبارداده



بعد پروژه

نام فارسی فیلد	نام لاتین فیلد
کد پروژه	Project_Code
عنوان	Project_Title
مرکز	Center
مدیر پروژه	Project_Manager
نوع پروژه	Project_Type
تاریخ شروع	Start_Date
طول پروژه-روز	Day Duration
طول پروژه-ماه	Month Duration
طول پروژه-سال	Year Duration
جریمه هر ماه	Penalty Each Month
مبلغ پروژه	Total Project Amount
طبقه بندی	Classification

بعد تاریخ

نام فارسی فیلد	نام لاتین فیلد
کلید تاریخ	Date_Key
کلید تاریخ شمسی	Shamsi_Date_Key
سال میلادی	Year
سال شمسی	Shamsi_Year
فصل میلادی	Quarter
فصل شمسی	Shamsi_Quarter
ماه میلادی	Month
ماه شمسی	Shamsi_Month
روز هفته میلادی	Weak_Day
روز هفته شمسی	Shamsi_Weak_Day

بعد محل تامین اعتبار

نام فارسی فیلد	نام لاتین فیلد
کد محل اعتبار	Resource_Code
نام محل اعتبار	Resource_Name

بعد سرفصل

نام فارسی فیلد	نام لاتین فیلد
سرفصل سطح ۱	GLlevel1
سرفصل سطح ۲	GLlevel2
شرح سرفصل سطح ۱	GLlevel1_Desc
شرح سرفصل سطح ۲	GLlevel2_Desc

فکت پروژه (از نوع تراکنشی)

نام فارسی فیلد	نام لاتین فیلد
کد پروژه	Project_Code
کلید تاریخ	Date_Key
کد محل تامین اعتبار	Resource_Code
سرفصل سطح ۲	GLlevel2
هزینه کرد	Turn_Over_Bed
دریافتی	Turn_Over_Bes
پیشرفت برنامه ای	Progress_Programmatic
پیشرفت واقعی	Progress_Supervisor
مبلغ جریمه	Penalty

مثال

سپرده‌های یک بانک در نظر بگیرید که به صورت روزانه روی آنها تراکنش اعمال می‌شود. فرض کنید که جدول تراکنش سپرده به صورت زیر باشد:

کلید سپرده	لحظه انجام تراکنش	گردش بدهکار	گردش بستانکار	Id تراکنش در روز
۱	10/20/2016 2:19:27	0	0	1
۱	10/20/2016 2:19:27	0	20	2
۱	10/21/2016 2:19:27	15	0	1
۲	10/25/2016 2:19:29	0	200	1
۲	10/25/2016 2:19:29	0	100	2
۳	10/26/2016 2:19:29	0	20	1
۱	10/26/2016 1:19:29	5	0	1

سپرده‌ها قبل از ورود به جدول تراکنش، هیچ مانده ای ندارند، (یعنی اولین تراکنش یک سپرده، مانده آن را تعیین می کند. مثلاً در مثال بالا، اولین مانده سپرده ۱ برابر ۰ و اولین مانده سپرده ۲ برابر ۲۰۰ می باشد).

یک فکت روزانه با ابعاد سپرده و تاریخ ایجاد کنید که در آن مقدار حداقل، حداکثر، میانگین (بر حسب تعداد تراکنش روز)، آخرین مانده و تعداد روزهای بدون عملکرد هر سپرده را در هر روز ثبت شده باشد. مثالی از خروجی مورد نظر که لازم است در یک جدول ذخیره شده باشد در زیر آمده است.

تعداد روزهای بدون عملکرد	آخرین مانده	میانگین مانده	حداکثر مانده	حداقل مانده	تاریخ	کلید سپرده
0	۲۰	10	20	0	10/2۰/2016	1
0	۵	۱۲.۵	۲۰	۵	10/2۱/2016	۱
1	۵	5	5	۵	10/22/2016	۱
2	۵	5	5	۵	10/23/2016	۱
3	۵	5	5	۵	10/24/2016	۱
4	۵	5	5	۵	10/25/2016	۱
0	۳۰۰	۲۵۰	۳۰۰	۲۰۰	10/25/2016	۲
0	0	2.5	5	0	10/26/2016	1
1	۳۰۰	300	۳۰۰	300	10/26/2016	۲
0	20	20	20	20	10/26/2016	3

پاسخ

```
procedure Ins_factDeposit(fromdate date, todate date) is
    currDate date;

begin
    currDate := fromdate;
    while currDate <= todate loop
        truncate table tmp_all_trn;
        truncate table tmp_dpst_sum_trnover;
        truncate table tmp_all_remain;
        truncate table tmp_last_remain;
        truncate table tmp_last_remain2;

        insert into tmp_last_remain
            select /*+parallel(8)*/
                f.dpst_num, f.effdate, f.bal, f.passivedays
            from factdeposit f
            and f.effdate = currDate - 1;
    commit;
```

```
insert into tmp_all_trn
  select /*+parallel(8)*/
    t.dpst_num, t.trns_time, t.dpst_trnover_bes, t.dpst_trnover_bed, t.trun_id
  from SA.dpst_trn t
  where t.trns_time >= currDate
    and t.trns_time < currDate + 1
commit;

insert into tmp_last_remain2
  select /*+parallel(8)*/
    t.dpst_num, t.trns_time effdate, sum(t2.dpst_trnover_bes - t2.dpst_trnover_bed)
dpst_bal
  from tmp_all_trn t
 inner join tmp_all_trn t2
    on t.dpst_num = t2.dpst_num
    and t.trns_time >= t2.trns_time
    and t.trun_id >= t2.trun_id
group by t.dpst_num, t.trns_time, t.trun_id;
commit;
```

-- mande dirooz sepordehaee ke tarakonesh dashteand ra dar tmp_all_remain mirizad

```
insert into tmp_all_remain
select /*+parallel(8)*/
t.dpst_num, currddate, min(t.dpst_bal) dpst_bal
from tmp_last_remain t
inner join tmp_all_trn t2
on t.dpst_num = t2.dpst_num
group by t.dpst_num;
```

commit;

-- Mandehaye jadid ra dar tmp_all_remain mirizad, bad az anjame in marhale agar sepordehee n tarakonesh dashte bashad, n+1 mande barayash darim

```
insert into tmp_all_remain
select /*+parallel(8)*/
nvl(t.dpst_num, t2.dpst_num),
nvl(t2.effdate, t.effdate + 1),
nvl(t.dpst_bal, 0) + nvl(t2.dpst_bal, 0) dpst_bal
from tmp_last_remain t
full outer join tmp_last_remain2 t2
on t.dpst_num = t2.dpst_num;
```

commit;

```
truncate table tmp_min_max_avg;
insert into tmp_min_max_avg
select /*+parallel(8)*/
    t.dpst_num, min(t.dpst_bal), max(t.dpst_bal), avg(t.dpst_bal)
    from tmp_all_remain t
    group by t.dpst_num;
commit;
```

```
insert into tmp_dpst_sum_trnover
select /*+parallel(8)*/
    t.dpst_num,
    SUM(t.dpst_trnover_bes),
    SUM(t.dpst_trnover_bed)
    from SA.dpst_trn t
    where t.trns_time >= currDate
        and t.trns_time < currDate + 1
    group by t.dpst_num;
commit;
```

```
insert into factdeposit
select /*+parallel(8)*/
  nvl(tmp.dpst_num,tr.dpst_num)dpst_num,
  currDate effdate,
  nvl(tmp.dpst_bal, 0) + nvl(tr.trnoverbes, 0) - nvl(tr.trnoverbed, 0) bal,
  nvl(tr.trnoverbes, 0) trnoverbes,
  nvl(tr.trnoverbed, 0)trnoverbed,
  nvl(tmp_bal.min_dpst_bal, 0) endayminbal,
  nvl(tmp_bal.max_dpst_bal, 0) endaymaxbal,
  nvl(tmp_bal.avg_dpst_bal, 0) endayavgbal,
  case
    when tr.dpst_num is null then
      nvl(tmp.passivedays + 1, 0)
    else
      0
  end passivedays
from tmp_last_remain tmp
  on tmp.dpst_num = f.dpst_num
 full outer join tmp_dpst_sum_trnover tr
  on tr.dpst_num = f.dpst_num
 full outer join tmp_min_max_avg tmp_bal
  on tmp_bal.dpst_num = f.dpst_num;
commit;

currDate := currDate + 1;
end loop;
end;
```