بنام خدا

پایگاه داده ۲

Implementation of Data Warehouse

بصیری
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

ایجاد جداول در انبارداده

- بعد از اینکه طراحی انبارداده ایجاد شد، تمامی بعدها و فکت ها آماده شده اند.
- برای هر فکت یا بعد فیلدها مشخص شده است.
- نوع هر کدام از فیلدها باید مشخص شده باشد.
- لازم است بر اساس طراحی انجام شده، متادیتای جداول انبارداده ایجاد شود.
- بهتر است طول فیلدها را کمی بزرگتر از سورس در نظر بگیریم.

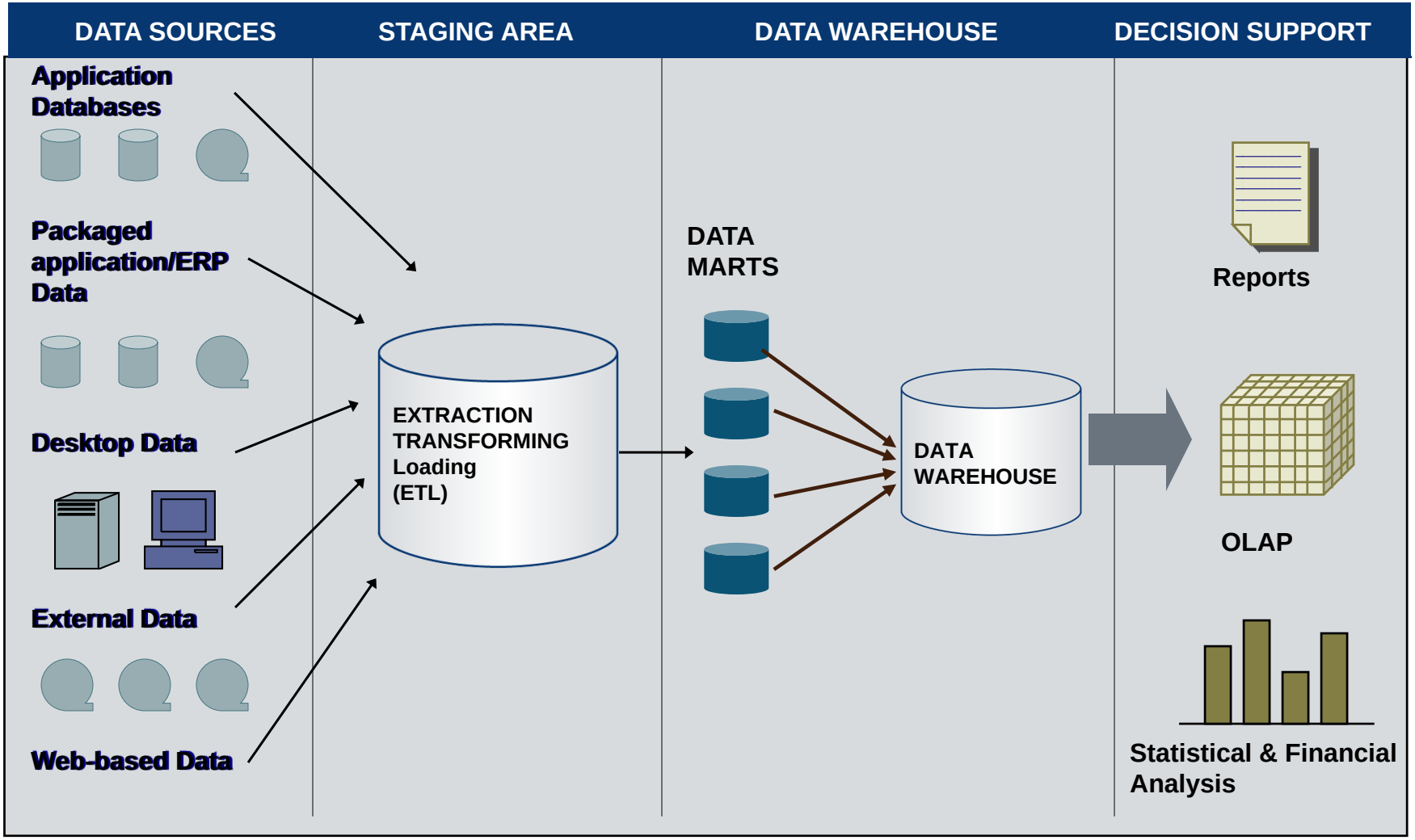
ایجاد مستند ETL

- لازم است نحوه بدست آوردن فیلدهای هر جدول انبارداده بر اساس داده های موجود در منابع داده ای، در این مستند به صورت دقیق مشخص شده باشد.
  - مثلاً فیلد نام مشتری از جدول customer بدست می آید
  - فیلد نوع مشتری، بوسیله ارتباط بین جدول customer و custype روی فیلد typ، در فیلد typedesc بدست می آید.
- این مستند مپ بین جداول سورس و جداول انبارداده را مشخص می کند.
- این مستند توسط تیم پیاده ساز ETL مورد استفاده قرار میگیرد.

انتخاب ابزار نوشتن ETL

- نوشتن اسکریپت
  - استفاده از پراسیجر
  - مزیت: عدم محدودیت به ابزارها
  - عیب: حجم زیاد پیاده سازی
- استفاده از ابزارهای ETL
  - نظیر ORACLE ODI, SSIS
  - مزیت: پیاده سازی سریعتر
  - عیب: محدود شدن به ابزار
  - عیب: نیاز با یادگیری ابزار
- استفاده از هر دو روش فوق

BI Architecture

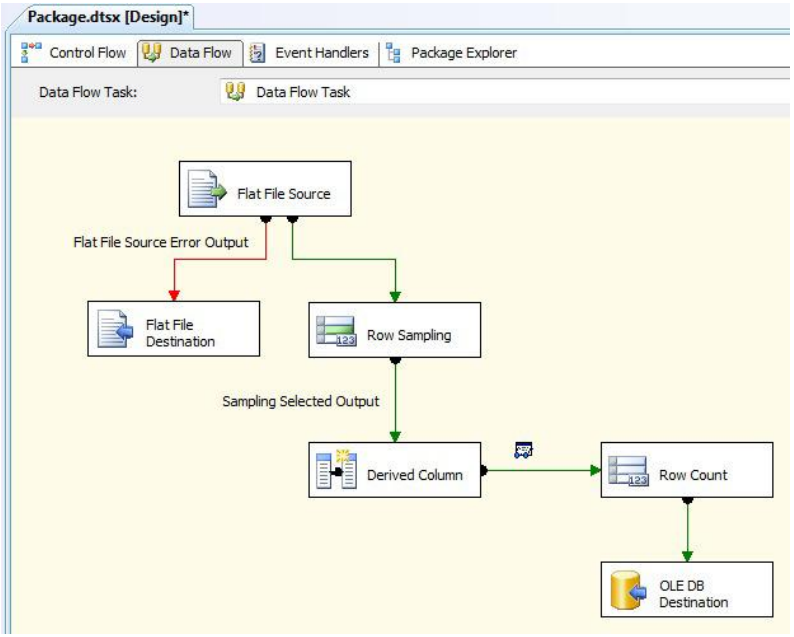


STAGING AREA

- در این قسمت می توان پردازشهای مقدماتی را در صورت لزوم انجام داد.
  - مثل حذف رکوردهای غیرضروری
  - حذف داده های تکراری
  - نرمال کردن یا هر گونه تغییر مقدار
- در محیطهای با حجم داده بزرگ، معمولاً کپی داده سورس در این محل ذخیره می شود.
  - این کار جهت عدم ایجاد بار روی پایگاه داده های سورس و یا تبدیل داده به فرمت پایگاه داده مورد نظر انجام می گیرد.
  - می توان جداول با حجم پایین و متوسط را حذف و مجدداً از داده های سورس پر کرد.
    - مانند جدول اطلاعات مشتری، جدول شعب و...
  - جداول سنگین را می توان به صورت افزایشی از منابع سورس به این محل انتقال داد.
    - مانند تراکنش های سپرده

نوشتن ETL

- بهتر است برای هر جدول در انبارداده یک پراسیجر داشته باشیم.
- لازم است ابتدا بعدهای مورد استفاده در یک فکت بروزرسانی شوند و سپس فکت مربوطه بروزرسانی گردد.
- بهتر است یک پراسیجر اصلی داشته باشیم که سایر پراسیجرها در آن فراخوانی شوند و ترتیب آنها نیز رعایت شود.
  - فراخوانی این پراسیجر در یک JOB



نوشتن ETL (ادامه)

- بهتر است رکورهای حذف شده در سورس، در جداول بعد انبارداده باقی بمانند.
- لازم است فکت های از نوع Transaction Fact Table و Periodic Snapshot Fact Table به صورت افزایشی پر شوند.
- لازم است پراسیجر به شکلی نوشته شود که در صورتی که در هر مرحله‌ای از اجرای آن پراسیجر، توقف ایجاد شود، با اجرای مجدد پراسیجر، همه چیز به درستی انجام شود.
  - بهتر است Insert در بعد یا فکت در یک مرحله انجام شود
  - و...
- لازم است برای اجرای اولیه ETL، پراسیجرهای متفاوتی داشته باشیم. (FirstLoad)

نوشتن ETL (ادامه)

- بهتر است برای بخش‌های مختلف هر پراسیجر لاگ ثبت شود
  - شامل لحظه انجام عملیات
  - توضیحی در مورد کاری که انجام شده است
  - نام جدولی که تغییر کرده است
  - و.....
- بهتر است از * Select در ETL استفاده نشود
- بهتر است از دستور Update استفاده نشود (کار اضافه)

نوشتن ETL (ادامه)

- لازم است با توجه به حجم داده و نحوه استفاده، فکتهای تاریخچه ای پارتیشن بندی داشته باشند
- لازم است با توجه به نحوه بکارگیری و واکشی داده ها، ایندکسهای مناسب روی فیلدهای انبارداده وجود داشته باشد (کار اضافه)
  - Bitmap
  - Normal
  - Unique
  - و...

نوشتن ETL (ادامه)

- در شرایطی که زیرساخت مناسبی در اختیار باشد می توان از Parallel Hint در اوراکل استفاده کرد
  - به منظور افزایش سرعت اجرا
  - همیشه این Hint مناسب نیست (کار اضافه)

نکات ETL

- **حذف اطلاعات از جداول حجیم با دستور delete زمانبر و مشکل ساز خواهد بود.**
- **برای truncate نمودن جداول حجیم ابتدا ایندکس های آنرا حذف و سپس جدول را truncate کنید.**
- **در صورت وجود ایندکس در جدول، truncateجدول در زمان بیشتری انجام خواهد شد.**
- **به خصوص در شرایطی که حجم اطلاعات جدول زیاد باشد انجام این کار با wait ها و مشکلاتی همراه خواهد بود.**

نکات ETL

- Drop کردن جداول می‌تواند بسیار پرریکسی باشد و در صورت وقوع اشتباه در این کار، نداشتن نسخه پشتیبان بار کاری زیادی را به شما تحمیل کند.
- حتی در صورت وجود پشتیبان نیز بازگرداندن آن ممکن است زمانگیر باشد.
- توصیه می‌شود جداولی را که می‌خواهید drop کنید ابتدا rename و در فرصت مناسب drop کنید.

نکات ETL

- **joinهای بزرگ در زمان ETL بهتر است به چند join کوچکتر شکسته شوند تا مشکلاتی نظیر محدودیت سایز temp را نداشته باشیم.**
- **در صورتی که از حلقه استفاده می‌کنید از اجرای کوئری های تکراری با نتیجه یکسان در داخل حلقه خودداری و اینگونه اسکریپت ها را به بیرون از حلقه منتقل کنید.**
  - **مثلا اگر تعدادی جدول را با هم join می‌کنید که نتیجه همواره ثابت است (نتیجه اجرا در بیرون از حلقه مانند درون حلقه است)، این کار را بیرون از حلقه انجام دهید.**

نکات ETL

- **تاریخ شروع و پایان را در ای تی ال هارد کد نکنید.**
- **در ابتدای ای تی ال first load بهتر است sequence را به incremental نیز می‌توانیم مقدار اولیه و در ابتدای ای تی ال آنرا به max(مقدار استفاده شده) + interval ریست کنیم.**

نکات ETL

- **در صورتی که تابعی بر روی فیلدی اعمال می‌شود، ایندکس ها(هر نوع ایندکسی؟ نمره اضافه) و پارتیشن ها در خروجی آن استفاده نمی‌شوند و کارآیی کوئری تا حد بسیار زیادی کاهش می‌یابد. مثلا اگر جدولی دارای partition روی فیلد effdate یا ایندکس روی این فیلد باشد فراخوانی‌های زیر از ایندکس و پارتیشن استفاده نمی‌کند:**

- Where trunc(effdate) = to_date(20140101,'yyyymmdd')

- Where to_char(effdate,'yyyymmdd','nls_calendar=persian') = '13930203'

- **بهتر است شرط به این شکل باشد:**

- Where effdate>= to_date(20140101,'yyyymmdd') and effdate< to_date(20140102,'yyyymmdd')

- Where effdate>= to_date(13930203','yyyymmdd' ,'nls_calendar=persian') and effdate< to_date(13930204,'yyyymmdd' ,'nls_calendar=persian')

نکات ETL

- **حتما از اسکریپت‌های خود بک آپ تهیه کنید.**
- **در زمان پیاده سازی، از insert، update، delete و اعمال اینگونه دستورات در سایر اسکیماها پرهیز کنید. مثلا از اسکیمای A اطلاعات اسکیمای B را تغییر ندهید. بهتر است اسکیمای B رویه‌ای داشته باشد و اسکیمای A آنرا فراخوانی کند. (اسکیما را در SQL SERVER همان دیتابیس در نظر بگیرید)**
- **می توان برای اطلاعات پایه ای از جداولی جدا از فکتها و ابعاد استفاده کرد. مثل اینکه ۰ یعنی مرد و ۱ یعنی زن**

مراحل انجام پروژه

- شناخت داده سورس و نیازمندی‌های داده‌ای حوزه مورد نظر
- طراحی
  - بر اساس مدل STAR
  - انواع فکت
  - SCD در بعدها
- مستند ای تی ال
  - برای هر بعد و یا فکت
- ایجاد انبارداده
  - ایجاد جداول در یک دیتابیس
- پیاده‌سازی ای تی ال
  - نوشتن پراسیجرها
- تست
- زمان‌بندی اجرا

فروشگاهی را با شعب مختلف در نظر بگیرید که شعب دارای سرپرستی و استان هستند. این فروشگاه مبالغ فروش و خرید خود را در سرفصلهای مختلف مالی وارد می کند که در سه سطح تعریف شده اند. بین مشتریان این فروشگاه می تواند روابط مختلفی از جمله معرف تعریف شود. همچنین این فروشگاه اقلام خود را از تامین کنندگان مختلف تامین می کند. با در نظر گرفتن جداول سورس زیر طراحی انبارداده را انجام دهید.

جدول شعبه: کد شعبه، نام شعبه، کد سرپرستی

سرپرستی شعبه: کد سرپرستی، شرح سرپرستی، کد استان

استان: کد استان، شرح استان

مشتری: کد مشتری، نام مشتری، شعبه، نوع مشتری، کد ملی، شغل، شماره تماس

آیتم: کد آیتم، شرح آیتم

نوع مشتری: نوع مشتری، شرح نوع مشتری

رابطه مشتری با با مشتری: کد مشتری ۱، کد مشتری ۲، نوع رابطه

نوع رابطه: نوع رابطه، شرح نوع رابطه

جدول سرفصل: سرفصل سطح ۳، شرح سرفصل سطح ۳، سرفصل سطح ۲

جدول سرفصل ۲: سرفصل سطح ۲، شرح سرفصل سطح ۲، سرفصل سطح ۱

جدول سرفصل ۳: سرفصل سطح ۱، شرح سرفصل سطح ۱

تامین کننده: شماره تامین کننده، نام، محل استقرار، آدرس

جدول فروش: کد مشتری، تاریخ، کد آیتم، مبلغ فروش، تعداد، سرفصل سطح ۳

جدول خرید: کد آیتم، شماره تامین کننده، تاریخ، مبلغ خرید،تعداد، سرفصل سطح ۳