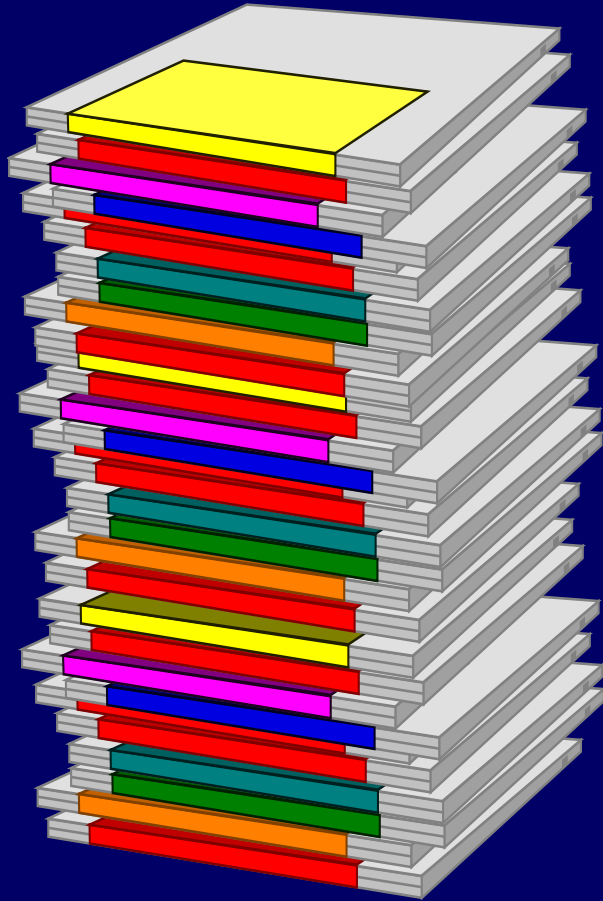


انباره داده



ارائه: امید میلانی فرد

سر فصل مطالب

⌘ مقدمه و معرفی انباره داده

⌘ معماری انباره داده

⌘ جمع آوری نیازمندی ها

⌘ طراحی انباره داده

انواع سیستمهای کامپیوتری در سازمانها

⌘ سیستمهای عملیاتی OLTP

⌘ سیستمهای اطلاعاتی DSS, DW/BI, OLAP



Watching the Wheels of Business Turns

Make the Wheels of Business Turns

OLTP: Online Transaction Processing

OLAP: Online Analytic Processing

DSS: Decision Support Systems

BI: Business Intelligence

داده ها در همه جا هستند..

⌘ نمی توانم داده های مورد نیاز را پیدا کنم

⌘ داده ها در سطح شبکه پراکنده اند

⌘ نسخه های زیادی دارند که تفاوت اندکی با هم دارند

⌘ نمی توانم داده های مورد نیاز را به دست آورم

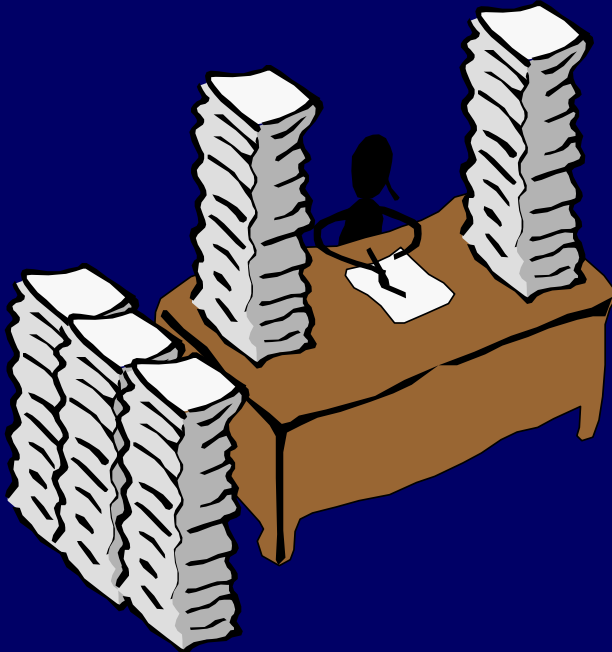
⌘ برای به دست آوردن داده ها نیازمند یک خبره هستم

⌘ نمی توانم داده هایی که پیدا کرده ام، درک کنم

⌘ مستند سازی داده های موجود بسیار ضعیف انجام شده است

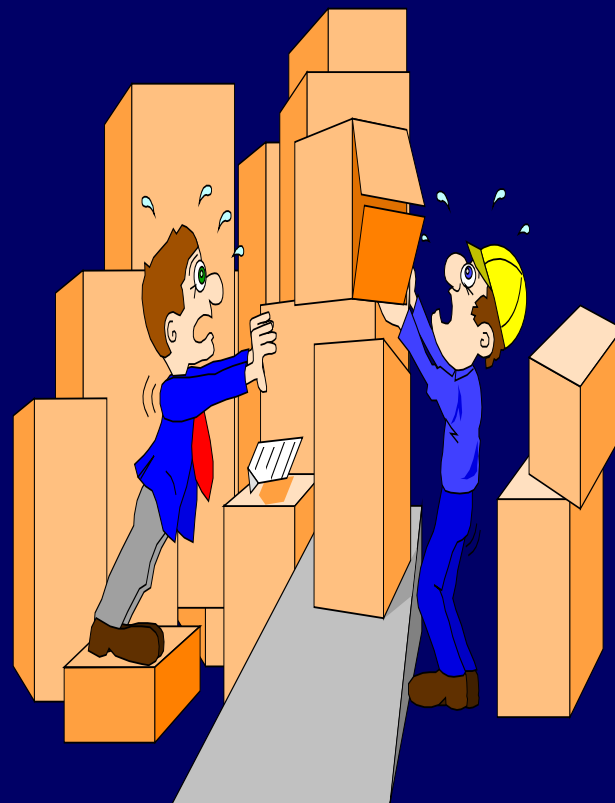
⌘ نمی توانم داده هایی که پیدا کرده ام استفاده کنم

⌘ داده ها نیازمند تبدیل از شکلی به شکل دیگر هستند



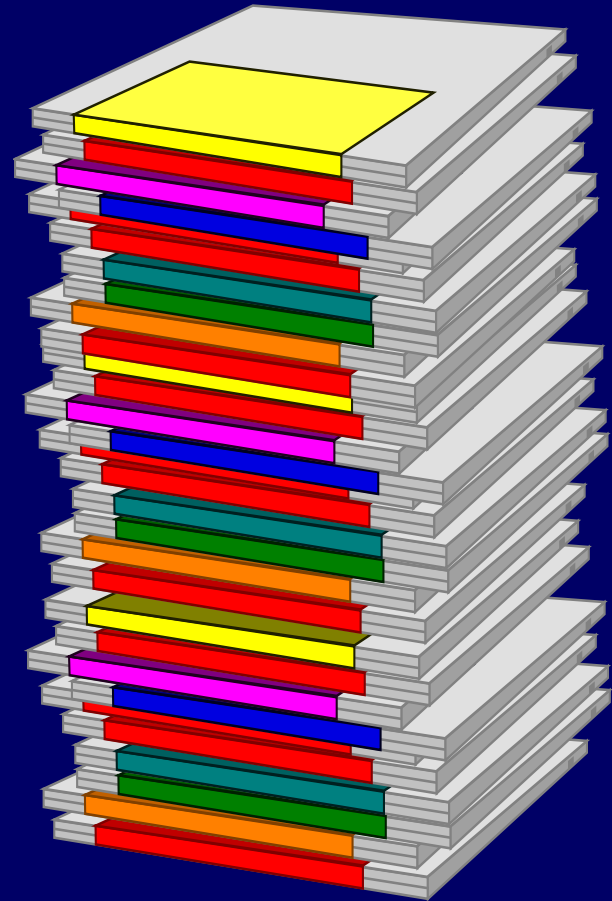
از دید کاربران...

- ❧ داده باید در سراسر سازمان یکپارچه باشد
- ❧ داده خلاصه سازی شده ارزشی واقعی در سازمان دارد
- ❧ داده های تاریخی کلید فهم داده در طی زمان را در خود دارند



انبار داده چیست؟

یک مخزن یکتا، کامل و یکپارچه از داده هایی که از منابع گوناگون گرفته شده و به شکلی در دسترس کاربر نهایی قرار گرفته اند که بتواند آنها را در یک زمینه ی تجاری، درک کرده و مورد استفاده قرار دهد.

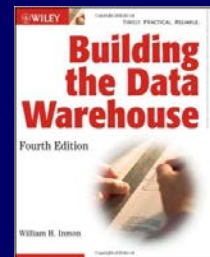


پیدایش

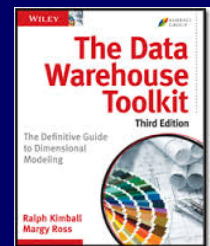
اواسط دهه ۸۰، توسط IBM تحقیقاتی انجام و Information Warehouse نامیده شد.



اواخر دهه ۸۰، Inmon (پدر DW)، کتاب Building the Datawarehouse را منتشر ساخت. (دیدگاه Top-Down)



Kimball در سال ۹۶ کتاب The Data Warehouse Toolkit را منتشر کرد. او یکی از اساسی ترین مفاهیم طراحی یعنی Dimensional Modeling را ارائه داد که اساس چیزی است که امروزه Cube نامیده می شود. (دیدگاه Bottom-Up)



انبار داده

⌘ خصوصیات اصلی:

☒ موضوع گرا

☒ جامع، یکپارچه

☒ زمان گرا time-varying

☒ فقط خواندنی non-volatile

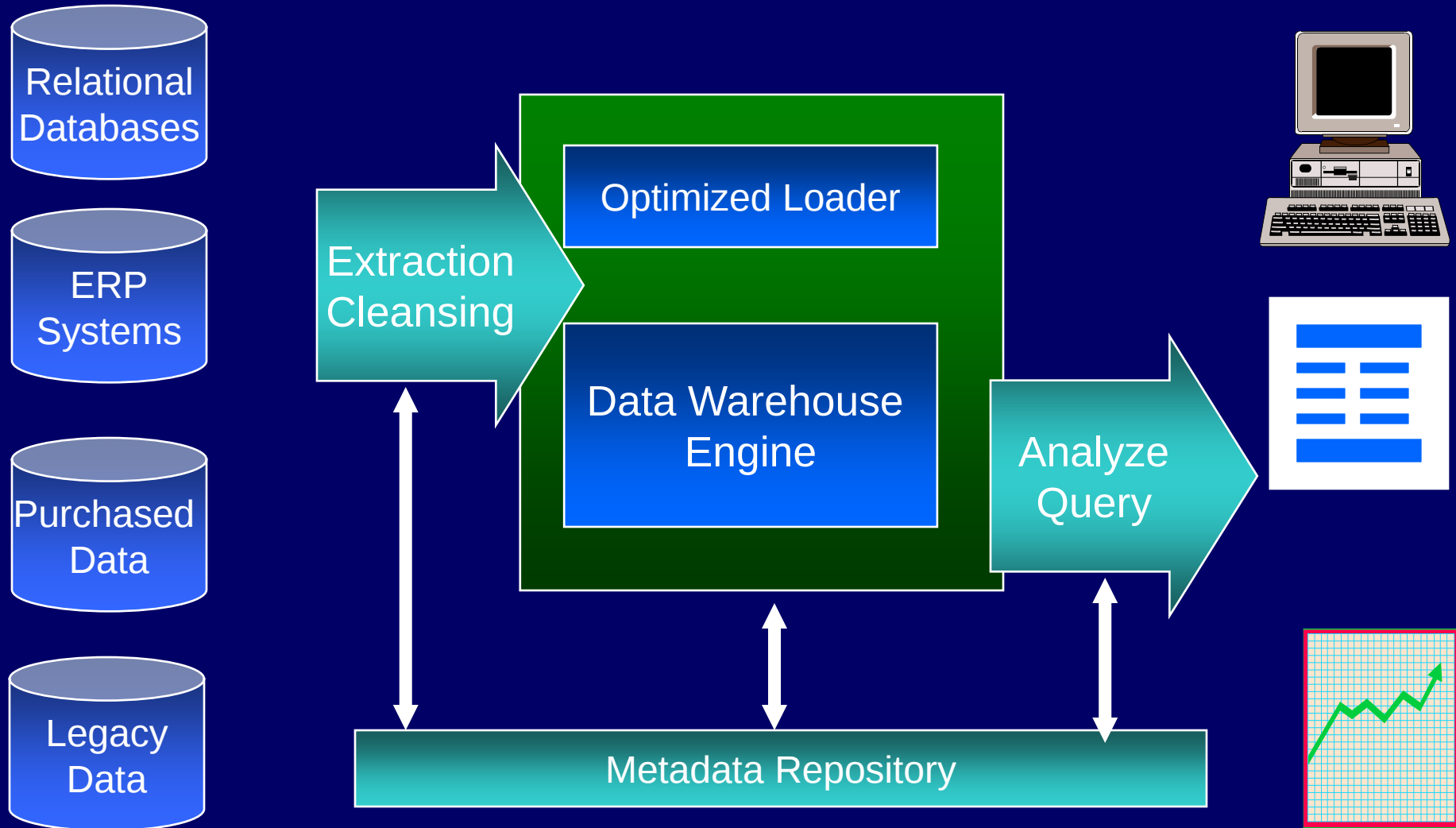
⌘ و بیش از اینها:

⌘ جمع آوری داده از منابع مختلف (منابع داخلی و خارجی با سیستمهای عملیاتی غیر یکپارچه)

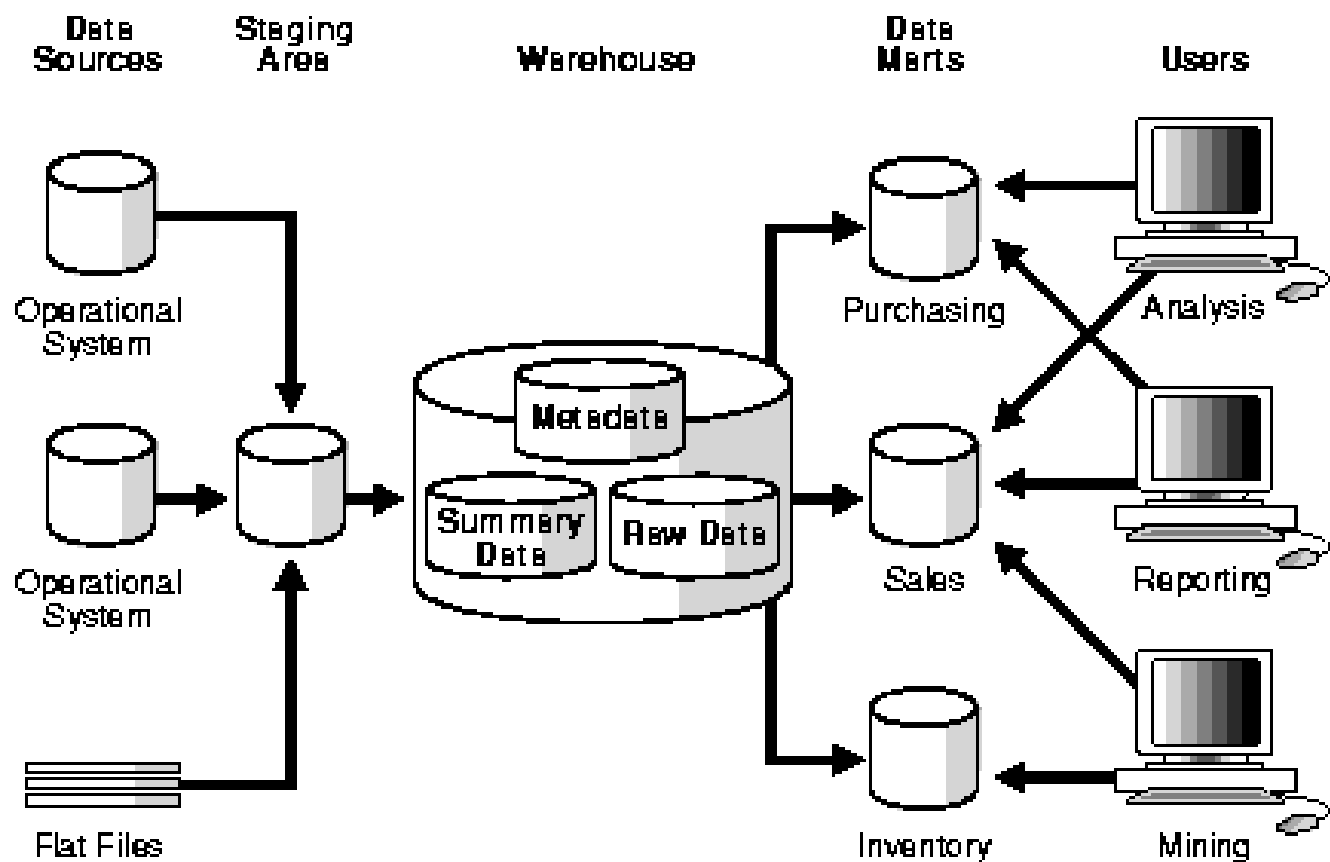
⌘ مجزا از تمام سامانه های عملیاتی (چرا؟)

⌘ خلاصه سازی شده: داده های عملیاتی به قالب قابل استفاده در تصمیم گیری نگاشت می شوند

معماری انبار داده

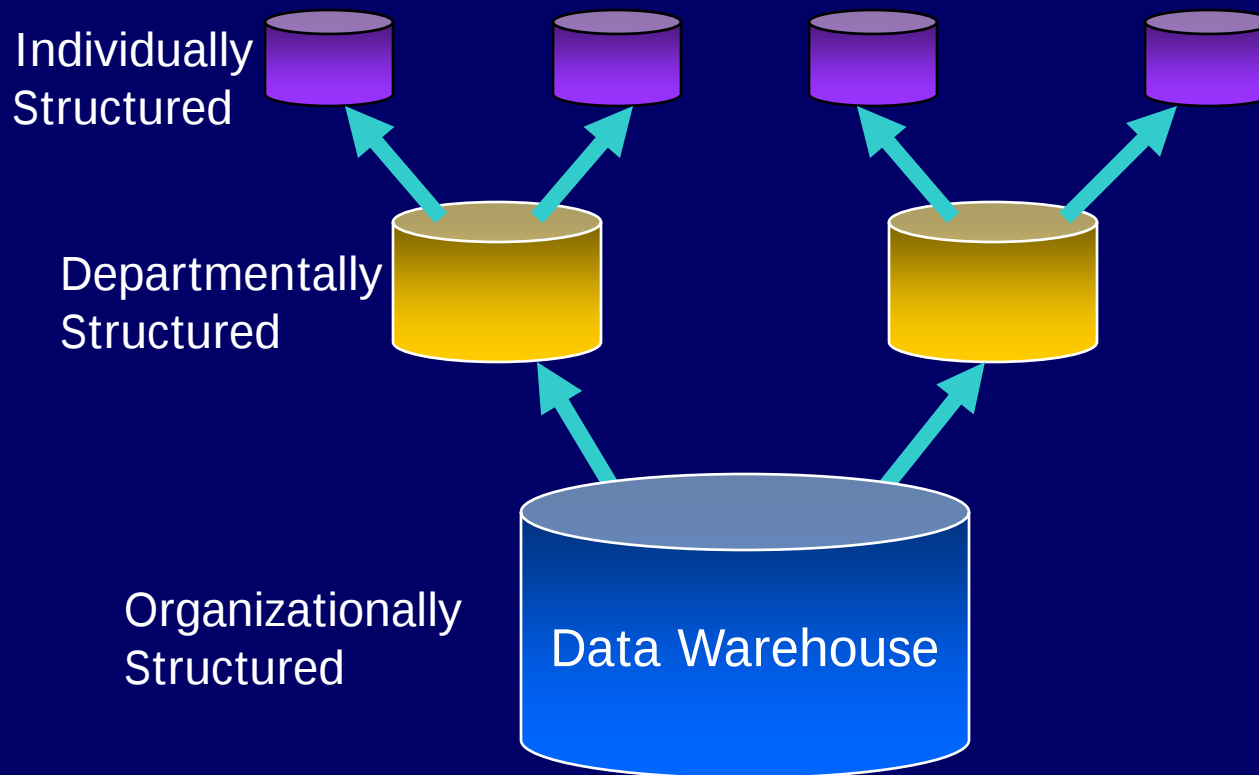


معماری انبار داده

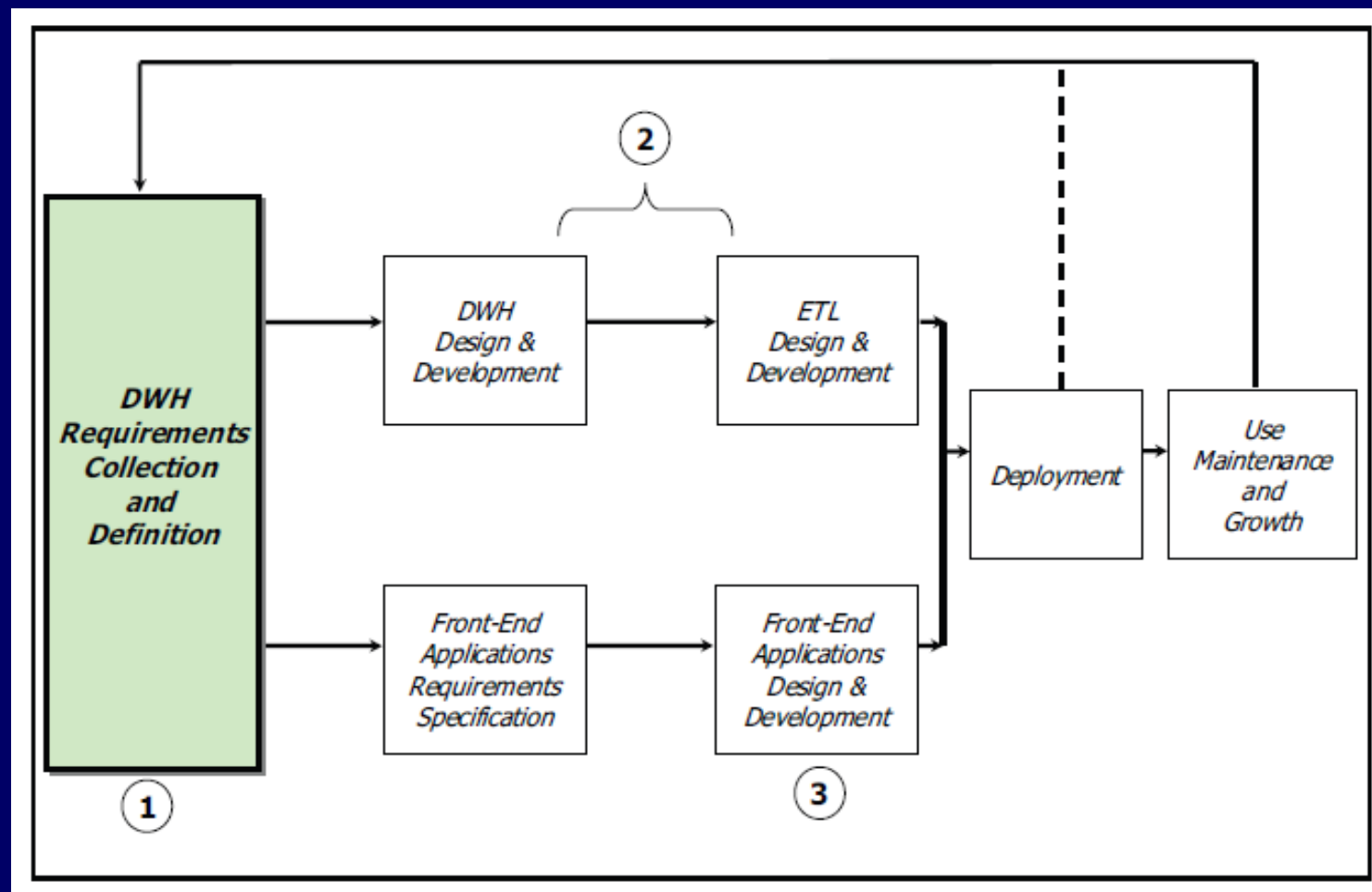


Data Mart & Data Warehouse

کاربران دپارتمانها انتظارات متفاوتی از DW دارند. داده ها از DW به سوی دپارتمان ها جریان می یابد تا نیازهای آنها را برطرف سازد. Data Mart در واقع DW کوچکتري است که مربوط به یک دپارتمان می باشد



چرخه عمر توسعه ی سیستم انبار داده



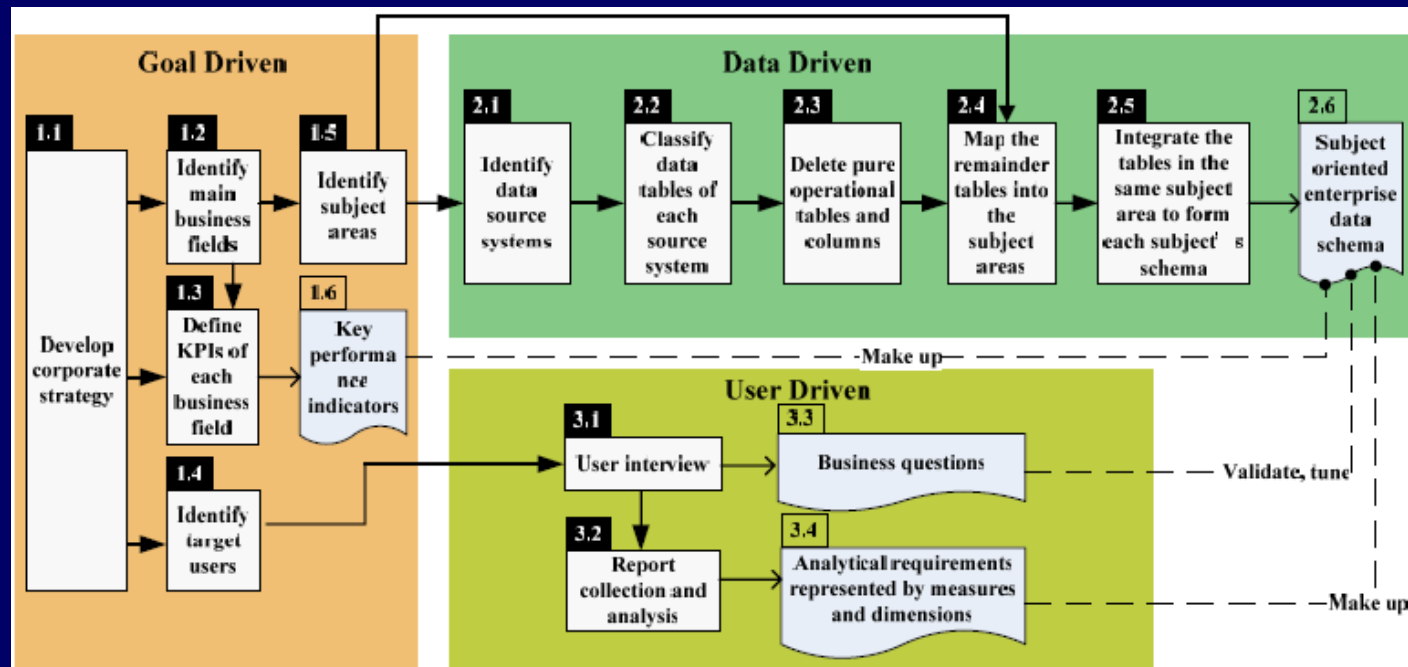
رویکردهای تحلیل نیازمندی ها

- ⌘ مستخرج از داده (Data Driven)
- ⌘ مستخرج از کاربر (User Driven)
- ⌘ مستخرج از هدف (Goal Driven)
- ⌘ ترکیبی

رویکردهای ترکیبی

رویکرد استخراج از سه گانه (triple-driven)

استفاده از goal-driven به عنوان راهنمای مدیر پروژه برای تعیین اهداف تجاری توسعه‌ی استراتژی سازمان و شناسایی زمینه‌ی اصلی کسب و کار شناسایی شاخصهای کلیدی کارایی (KPI) توسط مدیران ارشد (جهت رسیدن به اهداف) همزمان، شناسایی داده‌های مرتبط با DW و تحلیل آنها برای ساخت شمای داده موضوع گرا توسط مدیر پایگاه داده استخراج نیازمندی‌ها با مصاحبه با کاربران نهایی و تحلیل گزارشات تجاری ادغام اسناد حاصل از تمام این مراحل برای رسیدن به مدل منطقی



رویکردهای طراحی / مدلسازی

⌘ نرمال سازی

⌘ چند بعدی

رویکرد نرمال سازی

- ارائه شده توسط Bill Inmon (اوایل دهه ۹۰)، با روش **Top-Down** پیاده سازی می شود.
- داده های ذخیره شده در قالب نرمال نوع سوم 3N (اطلاعات کلیدی در جداول جداگانه)
- شدیداً قابل گسترش است (هر قطعه کلیدی از اطلاعات جداگانه نگهداری می شود)
- اندازه ی DW به صورت قابل ملاحظه ای بزرگ شده و کوئری زدن در آن مستلزم JOIN کردن جداول زیادی خواهد بود.
- برای دسترسی به داده ها در DW نرمال شده باید شمای DW، کلیدهای آن، روابط و جداول موجود را شناخت.

جنبه های مثبت و منفی رویکرد نرمال سازی

جنبه های مثبت:

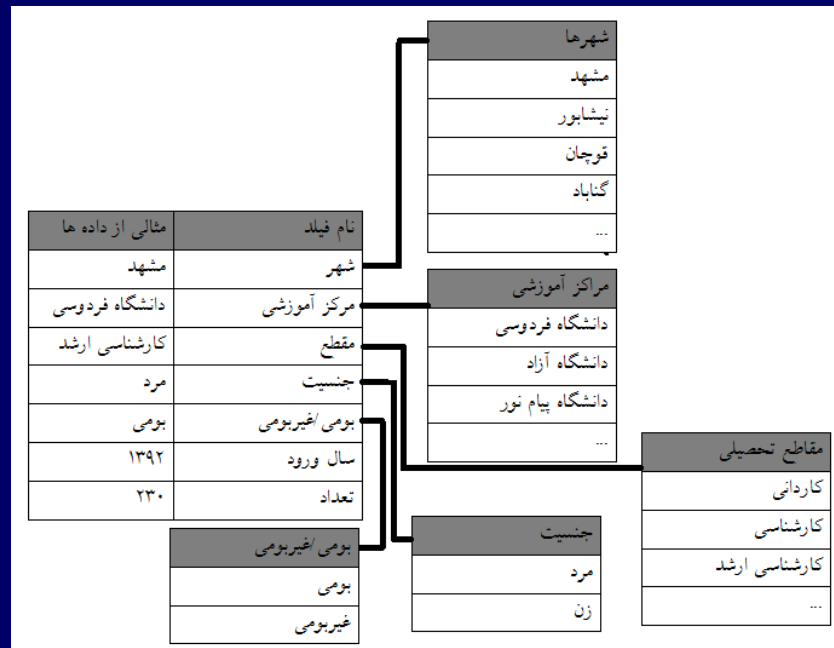
- داده در قالب اتمیک رابطه ای 3N است
- به راحتی با تغییرات منطبق می شود
- قالب 3N افزونگی داده را از بین می برد
- مستقل از برنامه کاربردی است

جنبه های منفی:

- انباره داده های 3N نمی توانند کوئری های کارا تولید کنند، برخی اوقات کند بوده و باید توسط چندین کوئری اجرا شوند
- ابزار OLAP بر خلاف OLTP برای بالا بردن کارایی ترجیح می دهند با داده های غیر نرمال کار کنند.
- از آنجا که به دلیل نرمال بودن داده ها و نداشتن افزونگی تعداد بسیار زیادی جدول و رابطه ایجاد می شود، ممکن است این اطلاعات روی سرورهای مجزا پخش شوند که تاثیر منفی در کارایی کوئری ها خواهد گذاشت.
- بیشتر برای پایگاه داده های غیر DW مناسب است.

رویکرد بُعدی (Dimensional)

- توسط Kimball ارائه شده و بیشتر با روش **Bottom-Up** پیاده سازی می شود.
- ابعاد، اطلاعات مرجع برای واقعیتها هستند، اعداد موجود در داده های تراکنشی هستند.
- اطلاعات ذخیره شده در یک DW چند بعدی، می تواند شامل یک جدول حاوی واقعیتها و تعداد زیادی بُعد متصل باشد. (به عنوان شمای ستاره ای نیز مورد اشاره قرار می گیرد.)
- از آنجا که واقعیتها و ابعاد از نظر درک کاربری آسان هستند، مدیران و متخصصان کسب و کار بدون دانش زیاد کامپیوتری هم می توانند آنها را به کار برند.



طراحی Schema در رویکرد چند بُعدی

انواع Schema

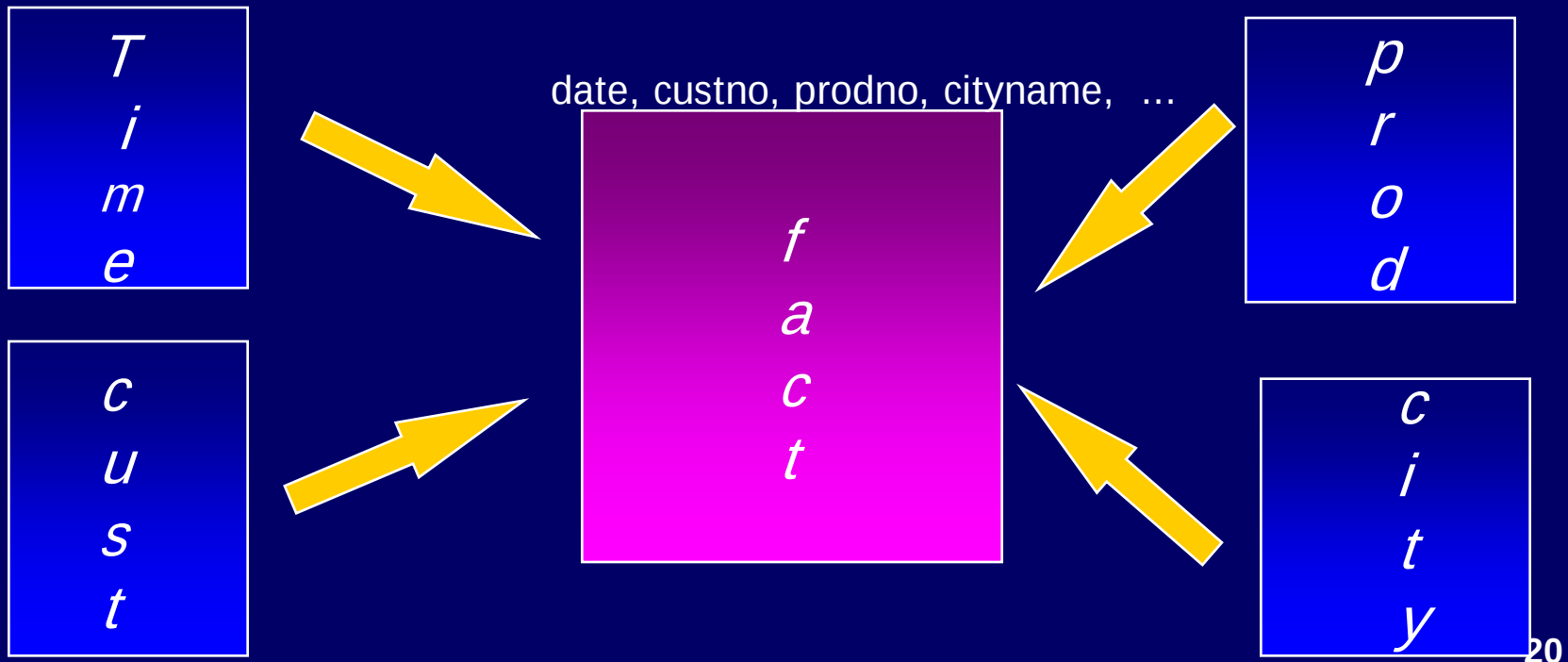
ستاره ای

Constellation صورت فلکی

Snowflake دانه برفی

Star Schema

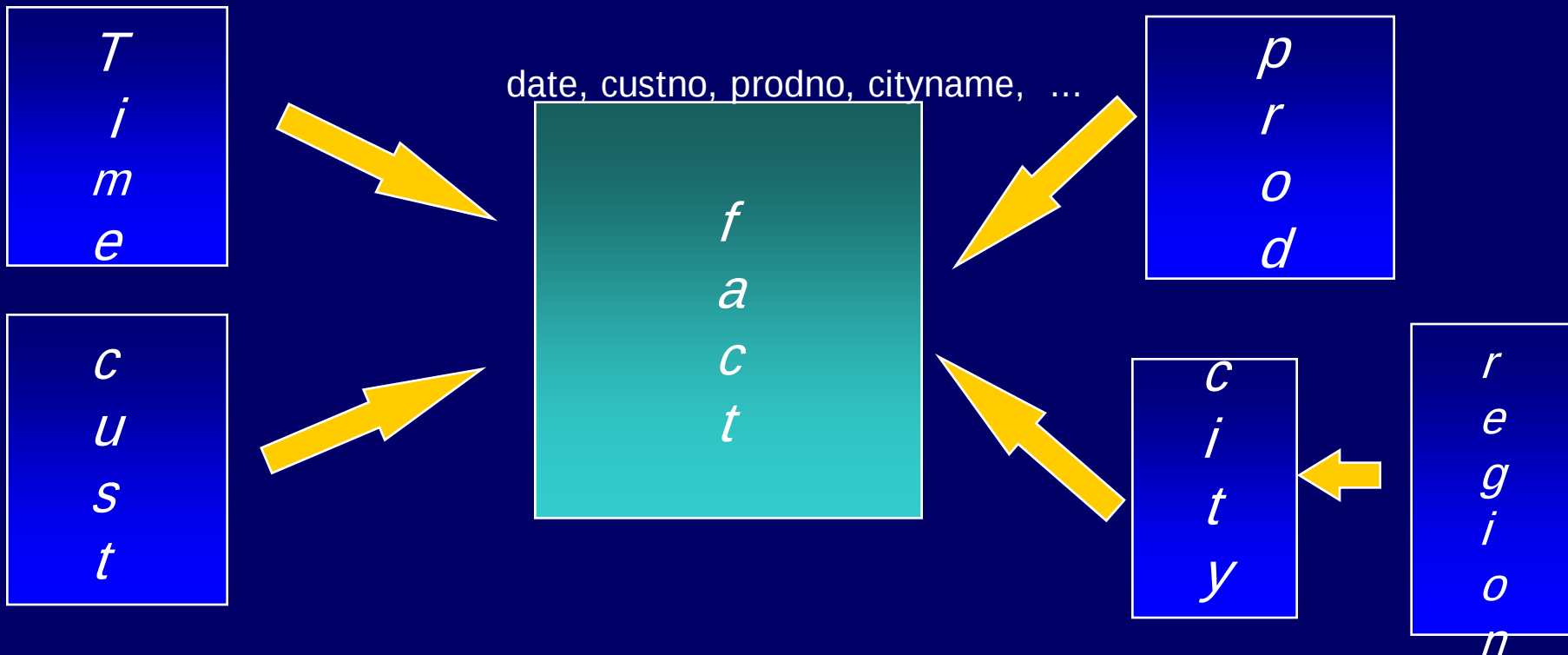
یک fact table و یک dimension
table



Snowflake schema

⌘ نشان دادن سلسله مراتب بعدی بصورت مستقیم به وسیله جداول نرمال شده .

⌘ نگهداری و ذخیره سازی انبار آسان است.

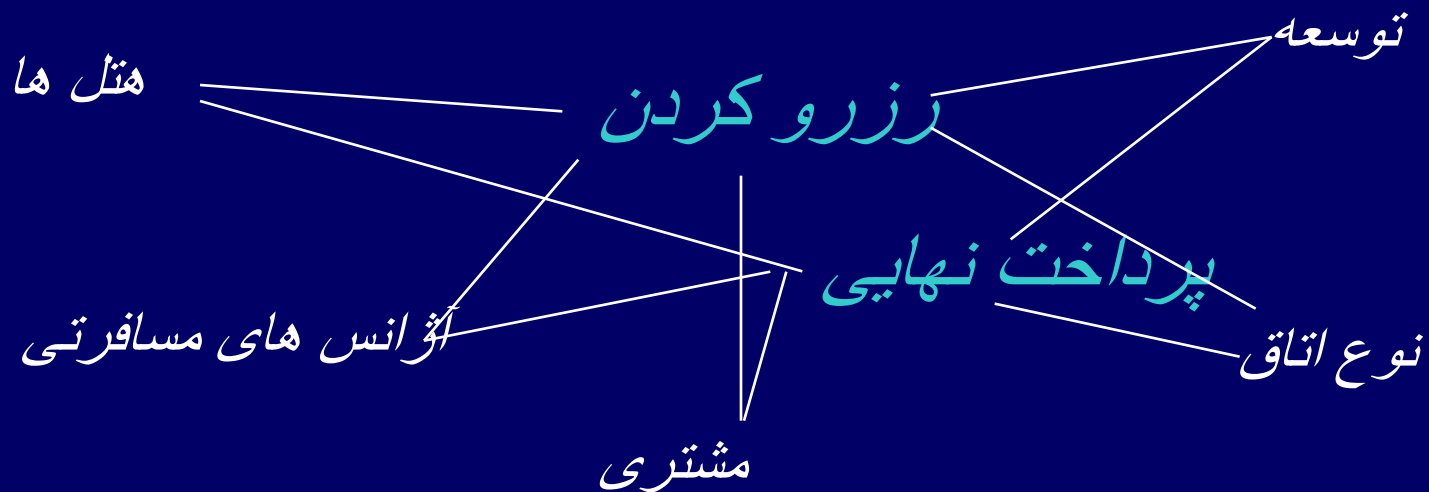


Fact Constellation

Fact Constellation⌘

چندین fact table که تعداد زیادی dimension table را به اشتراک می گذارند.

رزرو کردن و پرداخت نهایی ممکن است تعداد زیادی dimension tables را در صنعت هتل به اشتراک گذارد



سپاس از همراهی شما..

