

Data Warehouse Design & Develop



Masoud Mirzakhani
Senior DW/ ETL/ BI Architect

Microsoft SQL Server 2019 Design & Develop

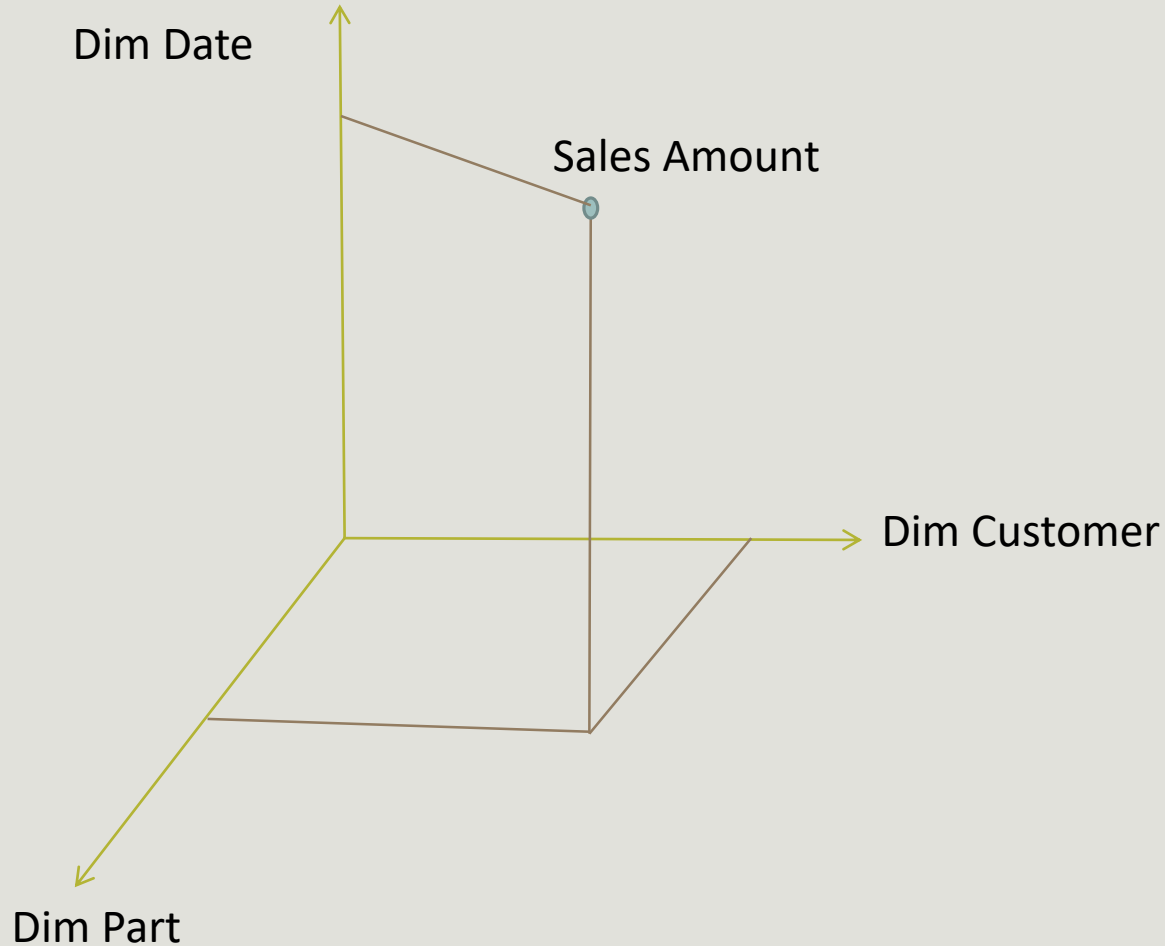


Masoud Mirzakhani
Senior DW/ ETL/ BI Architect

- **Master of Science in Information Technology**
- **Bachelor of Science in Information Technology**
- **md.mirzakhani@gmail.com**
- **[@MasoudMirzakhani](#)**
- **[linkedin.com/in/masoudmirzakhani](https://www.linkedin.com/in/masoudmirzakhani)**



معماری سیستم انبار داده



Fact چیست؟

Fact (چه اتفاقی افتاد)

- فروش محصولات
- خرید مشتریان
- ورود کالا به انبار
- و غیره.

Dim (ویژگی هایی که وقایع را توصیف می کند)

- زمان فروش محصول
- محصول فروخته شده

چالش های دیتابیس نرمال

- برای نگهداری اطلاعات ریز و جزئی طراحی شده است
- کارایی مناسبی برای ذخیره سازی داده های تراکنشی دارد.
- از یکپارچگی داده ها به خوبی پشتیبانی می کند.
- در نتیجه این طراحی تعداد زیادی جدول ایجاد می شود.
- راه حل مناسبی برای مدیریت داده های روزانه است.

Normalized Structure Challenges

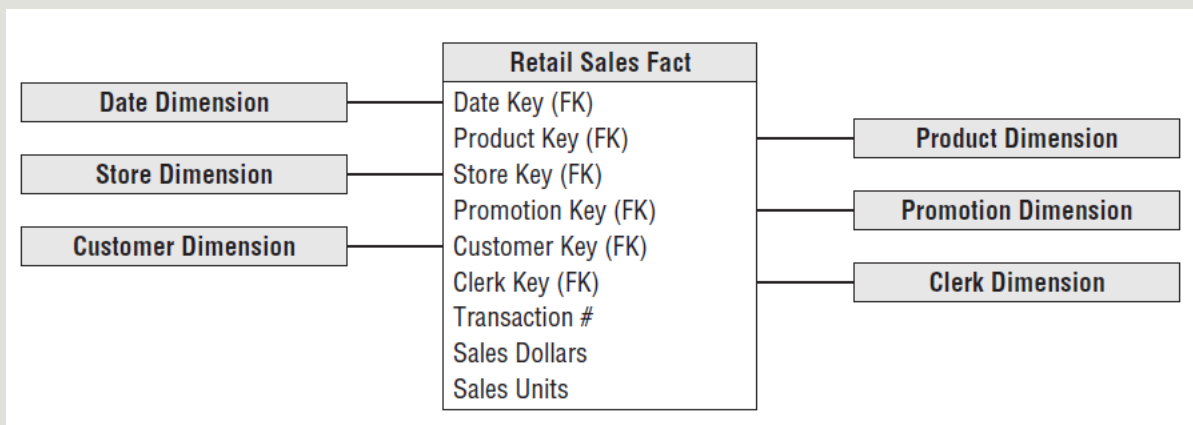


-
- معمولا کارایی مناسبی در خواندن اطلاعات ندارد.
 - نیاز به JOIN کردن تعداد زیادی جدول دارد.
 - معماری مناسبی برای تحلیل داده ها ارائه نمی دهد.
 - داده ها در ریزترین حالت خود ذخیره می شوند.
 - با زیاد شدن حجم داده، کارایی Query ها کاهش پیدا می کند.

معماری سیستم انبار داده

Star Schema

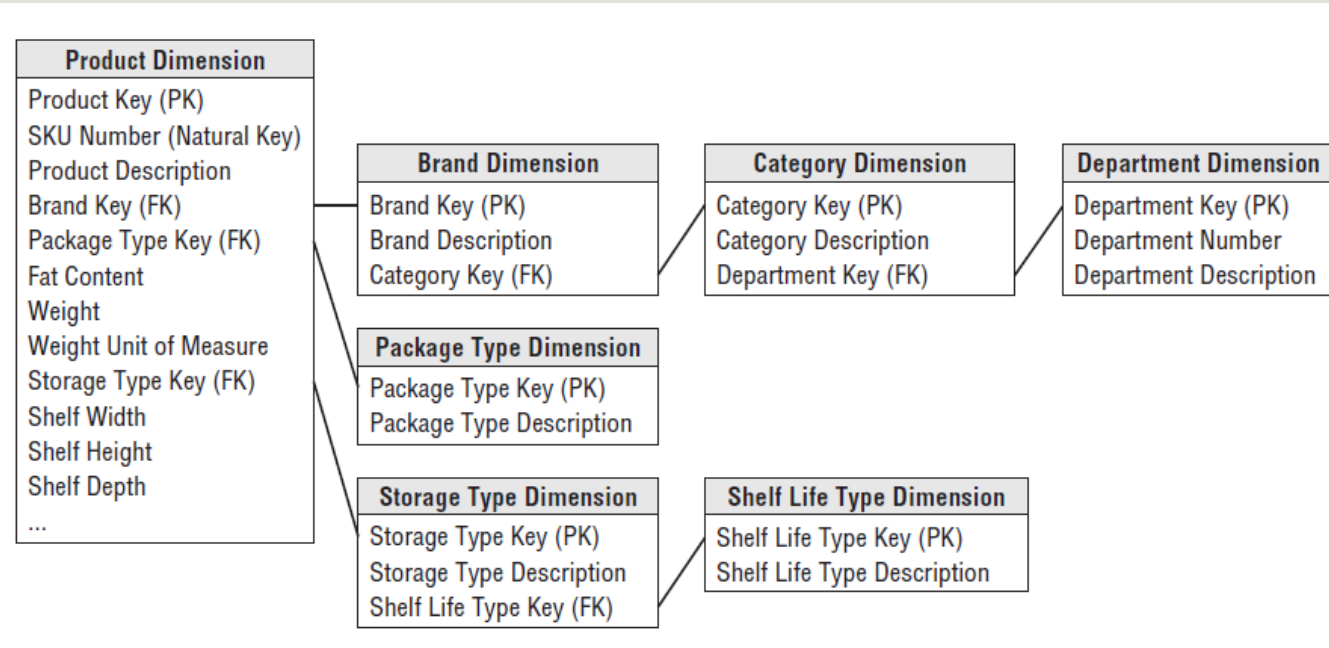
- ساده ترین معماری انبار داده.
- بهترین کارایی را دارد.
- ارتباط تنها بین جداول Fact و Dimension برقرار می شود.



معماری سیستم انبار داده

Snowflake Schema

- ارتباط بین جداول Dimension هم وجود دارد.
- همان Star Schema است که جداول Dimension کمی نرمال شده اند.
- برای زمانی که بخواهیم فرایند ETL را کمی ساده تر کنیم.
- برای سبک تر کردن Dimension های بزرگ
- تنها در موارد خاص استفاده شود.



ساختار جداول Fact



• Additive, Semi-Additive, Non-Additive Facts

• **Additive** : اندازه هایی که در مقابل تمامی ابعاد، جمع پذیر باشند.

• تعداد فروش

• **Semi-Additive**: اندازه هایی که در مقابل برخی ابعاد، جمع پذیر بوده و در مقابل دیگر ابعاد جمع پذیر نمی باشند..

• مانده بانک

• مانده انبار

• **Non-Additive**: اندازه هایی که در مقابل هیچ کدام از ابعاد، جمع پذیر نمی باشند.

• نسبت ها

Fact ساختار جداول



NULL Values

- توابع SUM, AVG, MIN, MAX, COUNT نسبت به مقادیر NULL درست رفتار می کنند.
- کلید های خارجی نبایستی حاوی مقادیر NULL باشند.

ساختار جداول Fact



Transaction Fact Tables •

- هر سطر نماینده یک اتفاق در فرایند متناظر است.

Periodic Snapshot Fact Tables •

- هر سطر نماینده اتفاق های فرایند در یک بازه زمانی خاص (روز، ماه، سال ...) است.
- هر سطر در جداول اسنپ شات می تواند معادل یک روز، ماه یا سال باشد.
- حتی اگر در بازه زمانی مورد نظر اتفاقی نیافتاده باشد، یک رکورد با اندازه های * بایستی ثبت شود.

Accumulating Snapshot Fact Tables •

- هر سطر نماینده اتفاق های فرایند از ابتدای زمان (مبدا) تا یک لحظه خاص است.

Fact جداول ساختار



Conformed Facts •

- جداول فکتی که ابعاد و اندازه های شبیه به هم داشته باشند.

Factless Fact Tables •

- مثال: حضور دانشجو با کلاس و تاریخ ارتباط دارد. ولی هیچ اندازه ای ندارد.

Consolidated Fact Tables •

- ترکیب کردن فکت های فرایند های مختلف در یک فکت خاص
- مثال: فروش و پیش بینی فروش

Fact جداول ساختار



Surrogate Keys •

- اجباری نیست.
- توصیه میشود که جداول فکت، دارای کلید های با مقادیر خودکار باشند.

Centipede Fact Tables •

- هنگامی که جدول تعداد خیلی زیادی ارتباط با ابعاد داشته باشد.
- وقتی که سلسله مراتب یک بعد، به صورت ابعاد جداگانه طراحی شده باشد و هر یک جداگانه با جدول فکت ارتباط داشته باشند.
- مثال: سال، فصل؛ ماه؛ روز
- در کل توصیه نمی شود.

ساختار جداول Dimension



- ساختار کلی:

- تمامی جداول Dimension، دارای یک کلید اصلی هستند.
- جداول فکت از طریق FK با کلید اصلی فوق در ارتباط هستند.
- اصولاً دینرمال بوده و دارای ستون های زیادی هستند.

- Denormalized Flattened Dimensions

- دینرمال و مسطح بودن ابعاد توصیه می شود.
- این ویژگی تاثیر مستقیم بر روی سرعت و سادگی ابعاد می گذارد.
- از نرمال سازی ابعاد پرهیز کنید.
- کد ها و شناسه های عملیاتی را می توان به صورت ستون (ویژگی) در این جداول مشاهده کرد.
- اکثر ستون ها از طریق کلمات و عبارات کوتاه پر شده اند.
- از ستون های ابعاد به منظور گروه بندی و فیلتر در گزارش ها استفاده می شود.

ساختار جداول Dimension



• Dimension Surrogate Keys

- کلید اصلی ابعاد، معمولاً با کلید اصلی در سیستم اطلاعاتی متفاوت است.
- در جاهایی که اطلاعات از بیش از یک منبع جمع آوری شده است
- در جاهایی که اطلاعات ابعاد در طول زمان تغییر می کند.
- خوب است که مقادیر عددی ترتیبی جانشین کلید اصلی که از سیستم عملیاتی می آید بشود.

• Natural, Durable, and Supernatural Keys

- کلید های طبیعی توسط سیستم عملیاتی تولید می شوند.
- مثال: اگر کارمندی از شرکت برود و مجدد استخدام بشود؛ کلید جدید در سیستم عملیاتی می گیرد.
- کلید های مانا و فراطبیعی، در طول زمان تغییر نمی کنند.
- مثال: اگر کارمندی از شرکت برود و مجدد استخدام بشود؛ کلید مانا یا فراطبیعی آن کارمند در انبار داده نباید تغییر کند.

ساختار جداول Dimension



• Degenerate Dimensions

- ابعادی که هیچ محتوایی به غیر از کلید اصلی خود ندارند.
- این ابعاد، در همان جداول فکت باقی می ماند؛ بدون این که ارتباطی با جدول بعدی داشته باشند.
- مثال: شماره فاکتور

• Flags and Indicators

- مخفف ها و نشانگرهای صحیح / نادرست را با کلمات متن کامل در جداول بعد قرار دهید.

• Null Attributes

- توصیه می شود مقادیر گم شده را با عناوین توصیفی همچون "نا مشخص" پر کنید.

ساختار جداول Dimension



Conformed Dimensions

- ابعادی که ساختار یکسانی دارند.

Role-Playing Dimensions

- جدول بعدی که بیش از یک بار با جدول فکت ارتباط دارد.
- توصیه می شود که به ازای هر رابطه مضاعف یک ویو از جدول بعد اصلی ایجاد شود و رابطه فوق با ویوی ایجاد شده برقرار شود.
- به ویوهای فوق Role گفته می شود.

Junk Dimensions

- بعد مستقلی است که برای نشان دادن مخفف ها و نشانگرهای صحیح / نادرست استفاده می شود.

Snowflaked Dimensions

- هنگامی که جدول بعد، نرمال شود.

Outrigger Dimensions

- در حالت دانه برفی، به بعدی گفته می شود که به آن اشاره می شود.

Slowly Changing Dimension



• Type 0: Retain Original

- مقادیر ستون ها ثابت است و تغییر نمی کند.

• Type 1: Overwrite

- مقادیر جدید؛ جاگزین مقادیر قبل می شوند.

• Type 2: Add New Row

- با تغییر مقادیر، یک رکورد جدید به بعد اضافه می شود.
- از کلید های مانا و فراطبیعی به منظور نگهداری ارتباط بین مقادیر قدیم و جدید استفاده می شود.
- برای پشتیبانی از این حالت حداقل نیاز است که ستون های زیر به جدول بعد اضافه شود:
 - از تاریخ
 - تا تاریخ
 - آخرین مقدار؟

Slowly Changing Dimension



Type 3: Add New Attribute •

- به ازای هر تغییر، یک ستون به بعد اضافه می شود تا مقادیر پیشین را نگهداری کند.
- این حالت به ندرت استفاده می شود.
- از این حالت به نام alternate reality نیز معروف است.

Dimension Hierarchies



- ابعاد سلسله مراتبی با عمق ثابت:
 - بهتر است که سطوح سلسله مراتب تبدیل به ستون های بعد شود.
 - مثال: ساختار حساب در حسابداری
- ابعاد سلسله مراتبی با عمق متغیر:
 - اگر سطوح سلسله مراتب متغیر است ولی دارای محدودیت می باشد؛ می توان همانند روش قبل آن ها را مسطح کرد.
 - مثال: مناطق جغرافیایی