

# Data Warehouse Design & Develop



---

Masoud Mirzakhani  
**Senior DW/ ETL/ BI Architect**

# Microsoft SQL Server 2019 Design & Develop



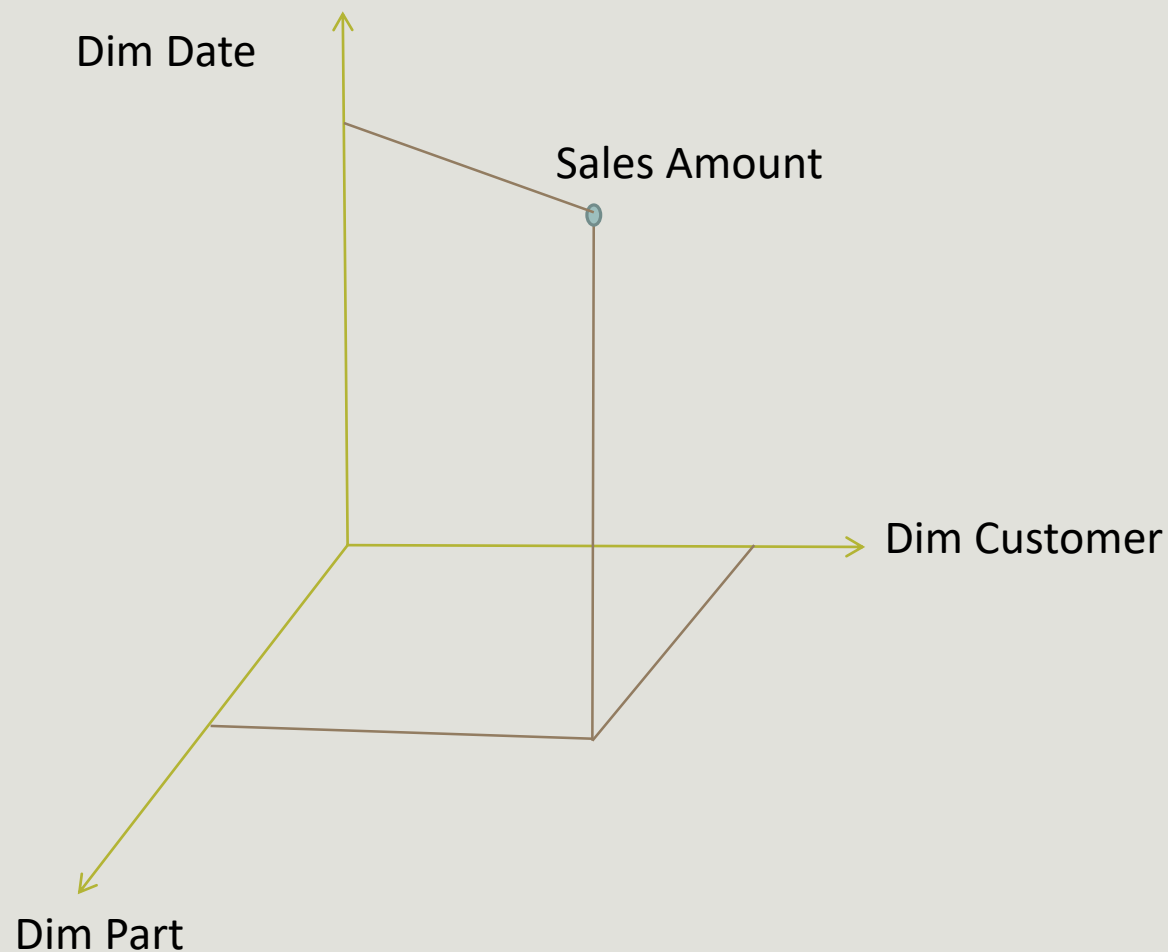
---

Masoud Mirzakhani  
**Senior DW/ ETL/ BI Architect**

- **Master of Science in Information Technology**
- **Bachelor of Science in Information Technology**
- **[md.mirzakhani@gmail.com](mailto: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

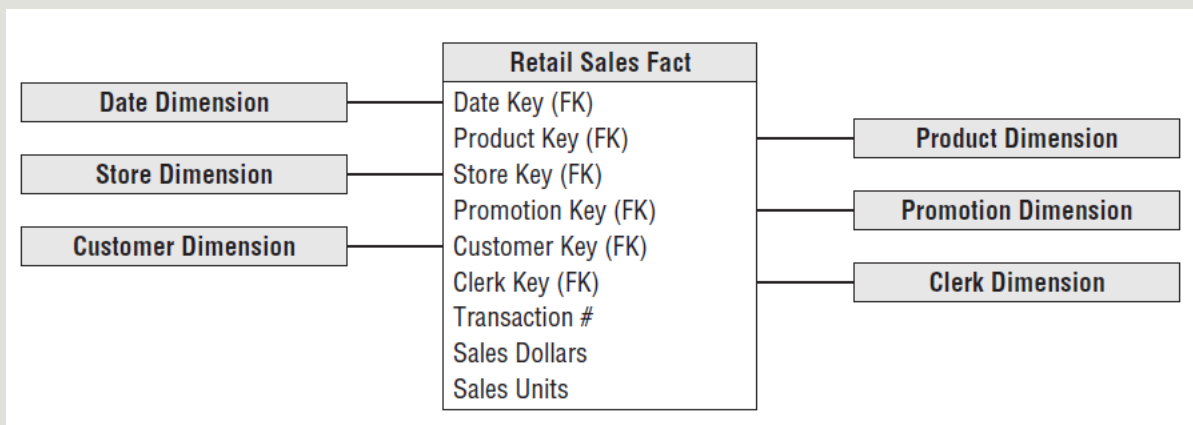


- 
- معمولا کارایی مناسبی در خواندن اطلاعات ندارد.
  - نیاز به JOIN کردن تعداد زیادی جدول دارد.
  - معماری مناسبی برای تحلیل داده ها ارائه نمی دهد.
  - داده ها در ریزترین حالت خود ذخیره می شوند.
  - با زیاد شدن حجم داده، کارایی Query ها کاهش پیدا می کند.

# 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

## Star Schema – طراحی اشتباه

FactSales				
DateID	ProductID	CustomerID	Quantity	Price
13990701	45	10	5	1000
13997702	78	14	9	7890

DimCustomer			
ID	FirstName	LastName	LocationTitle
10	Ali	Hasani	Tehran
14	Omid	Rezaee	Shiraz

FactEmployeeCount			
DateID	EmployeeID	DepartmentID	Count
13990802	78	3	1
13980802	79	3	1

DimEmployee			
ID	FirstName	LastName	LocationTitle
78	Behzad	Babaei	Tehran
79	Farid	Lari	Shiraz

# معماری سیستم انبار داده

## Star Schema – طراحی صحیح

FactSales					
DateID	ProductID	CustomerID	LocationID	Quantity	Price
13990701	45	10	11	5	1000
13997702	78	14	12	9	7890

FactEmployeeCount				
DateID	EmployeeID	LocationID	DepartmentID	Count
13990802	78	11	3	1
13980802	79	12	3	1

DimCustomer		
ID	FirstName	LastName
10	Ali	Hasani
14	Omid	Rezaee

DimLocation	
ID	Name
11	Tehran
12	Shiraz

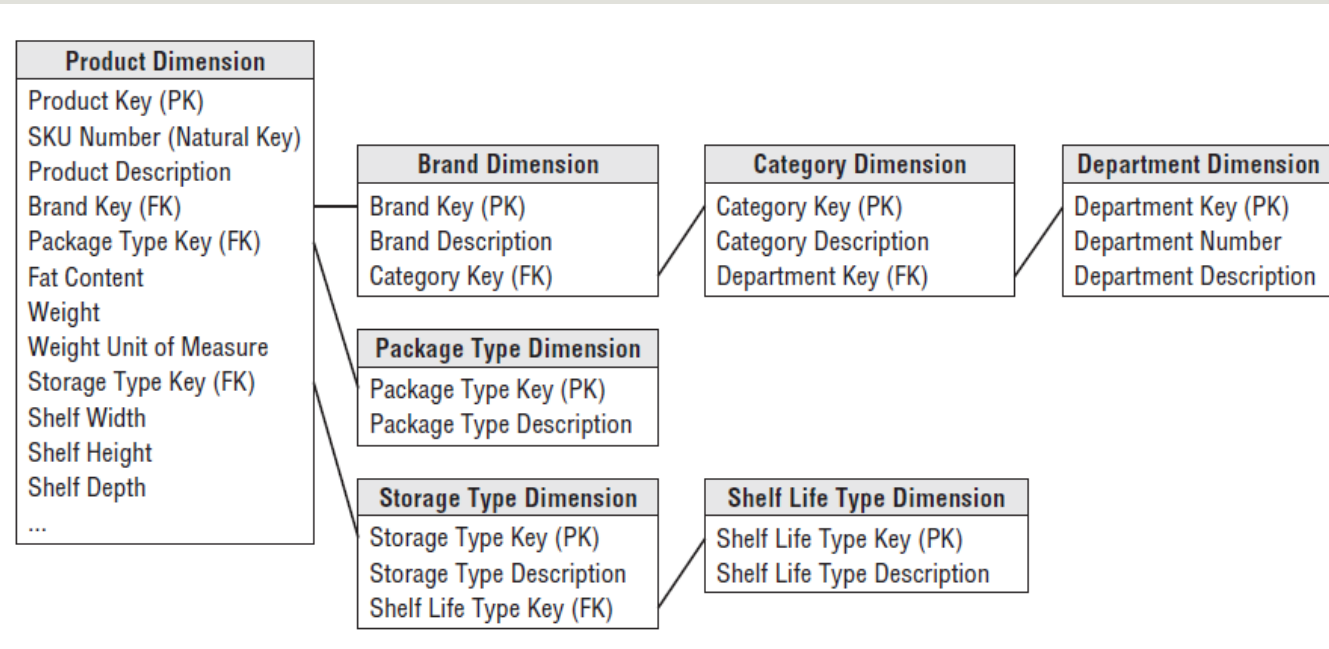
DimEmployee		
ID	FirstName	LastName
78	Behzad	Babaei
79	Farid	Lari



# 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.



# معماری سیستم انبار داده

## Snowflake Schema – طراحی صحیح (توصیه نمی کنم)

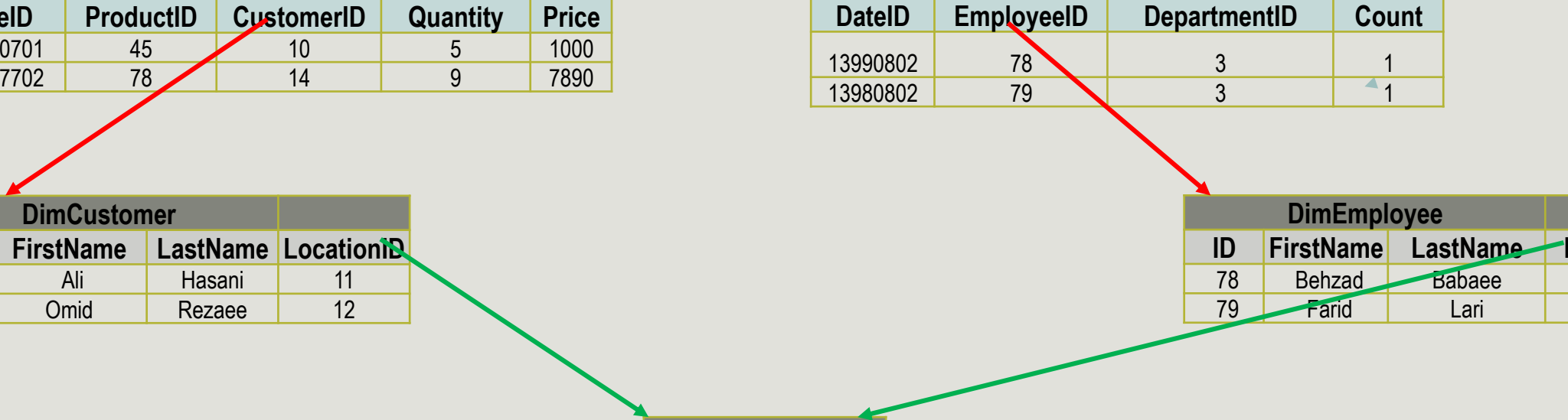
FactSales				
DateID	ProductID	CustomerID	Quantity	Price
13990701	45	10	5	1000
13997702	78	14	9	7890

FactEmployeeCount			
DateID	EmployeeID	DepartmentID	Count
13990802	78	3	1
13980802	79	3	1

DimCustomer			
ID	FirstName	LastName	LocationID
10	Ali	Hasani	11
14	Omid	Rezaee	12

DimEmployee			
ID	FirstName	LastName	LocationID
78	Behzad	Babaei	11
79	Farid	Lari	12

DimLocation	
ID	Name
11	Tehran
12	Shiraz



# 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.

# ساختار جداول Fact

## Conformed Facts •

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

FactSales		
DateID	ProductID	Quantity
۱۳۹۹.۱۰.۱	۱	۱۰
۱۳۹۹.۱۰.۱	۲	۲۰
۱۳۹۹.۱۰.۱	۳	۳۰

FactSalesBudget		
DateID	ProductID	Quantity
۱۳۹۹.۱۰.۱	۱	۱۵
۱۳۹۹.۱۰.۱	۲	۱۵
۱۳۹۹.۱۰.۱	۳	۲۰

# ساختار جداول Fact

FactSalesAndBudget (Design 1)			
DateID	ProductID	Quantity	QuantityBudget
۱۳۹۹.۱۰.۱	۱	۱۰	۱۵
۱۳۹۹.۱۰.۱	۲	۲۰	۱۵
۱۳۹۹.۱۰.۱	۳	۳۰	۲۰

FactSalesAndBudget (Design 2)			
DateID	ProductID	ValueType	Quantity
۱۳۹۹.۱۰.۱	۱	Sales	۱۰
۱۳۹۹.۱۰.۱	۲	Sales	۲۰
۱۳۹۹.۱۰.۱	۳	Sales	۳۰
۱۳۹۹.۱۰.۱	۱	Budget	۱۵
۱۳۹۹.۱۰.۱	۲	budget	۱۵
۱۳۹۹.۱۰.۱	۳	budget	۲۰

## Consolidated Fact Tables

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

# Fact ساختار جداول



---

## Factless Fact Tables

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



# ساختار جداول Fact



## 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

Product (Source1)	
ID	Title
۱	Chips
۲	Coca
۳	Gum

Product (Source2)	
ID	Title
۱	Fanta
۲	Pepsi
۳	7up

DimProduct (Design 1)	
ID	Title
۱۰۰۰۰۱	Chips
۱۰۰۰۰۲	Coca
۱۰۰۰۰۳	Gum
۲۰۰۰۰۱	Fanta
۲۰۰۰۰۲	Pepsi
۲۰۰۰۰۳	7up

DimProduct (Design 2)			
ID	SourceID	BaseID	Title
۱	۱	۱	Chips
۲	۱	۲	Coca
۳	۱	۳	Gum
۴	۲	۱	Fanta
۵	۲	۲	Pepsi
۶	۲	۳	7up

# 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.

DimSex	
۱	Female
۲	Male
۳	Unknown

# Slowly Changing Dimension

DimLocation		
ID	ParentID	Title
۱		Tehran
۲	۱	Rey
۳	۱	Karaj

DimLocation		
ID	ParentID	Title
۱		Tehran
۲	۱	Rey
۳	۴	Karaj
۴		Alborz

## Type 1: Overwrite

- مقادیر جدید؛ جاگزین مقادیر قبل می شوند.
- مثال: جابجایی **کرج** در تقسیمات کشوری.

FactEmployeeCount			
DateID	EmployeeID	LocationID	Count
۱۳۹۰.۰۱.۰۱	۱۲	۳	۱
۱۳۹۱.۰۱.۰۱	۱۲	۳	۱
۱۳۹۲.۰۱.۰۱	۱۲	۳	۱
۱۳۹۳.۰۱.۰۱	۱۲	۳	۱



# Slowly Changing Dimension



DataToInsight.ir

## 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

DimLocation						
ID	SNID	ParentID	Title	FromDate	ToDate	IsLast?
۱	۱		Tehran	NULL	NULL	TRUE
۲	۲	۱	Rey	NULL	NULL	TRUE
۳	۳	۱	Karaj	NULL	۱۳۹۰.۱۲۳۰	FALSE
۴	۴		Alborz	۱۳۹۱.۱.۱	NULL	TRUE
۵	۳	۴	Karaj	۱۳۹۱.۱.۱	NULL	TRUE

Type 2: Add New Row●

FactEmployeeCount			
DateID	EmployeeID	LocationID	Count
۱۳۸۹.۱.۱	۱۲	۳	۱
۱۳۹۰.۱.۱	۱۲	۳	۱
۱۳۹۱.۱.۱	۱۲	۵	۱
۱۳۹۲.۱.۱	۱۲	۵	۱

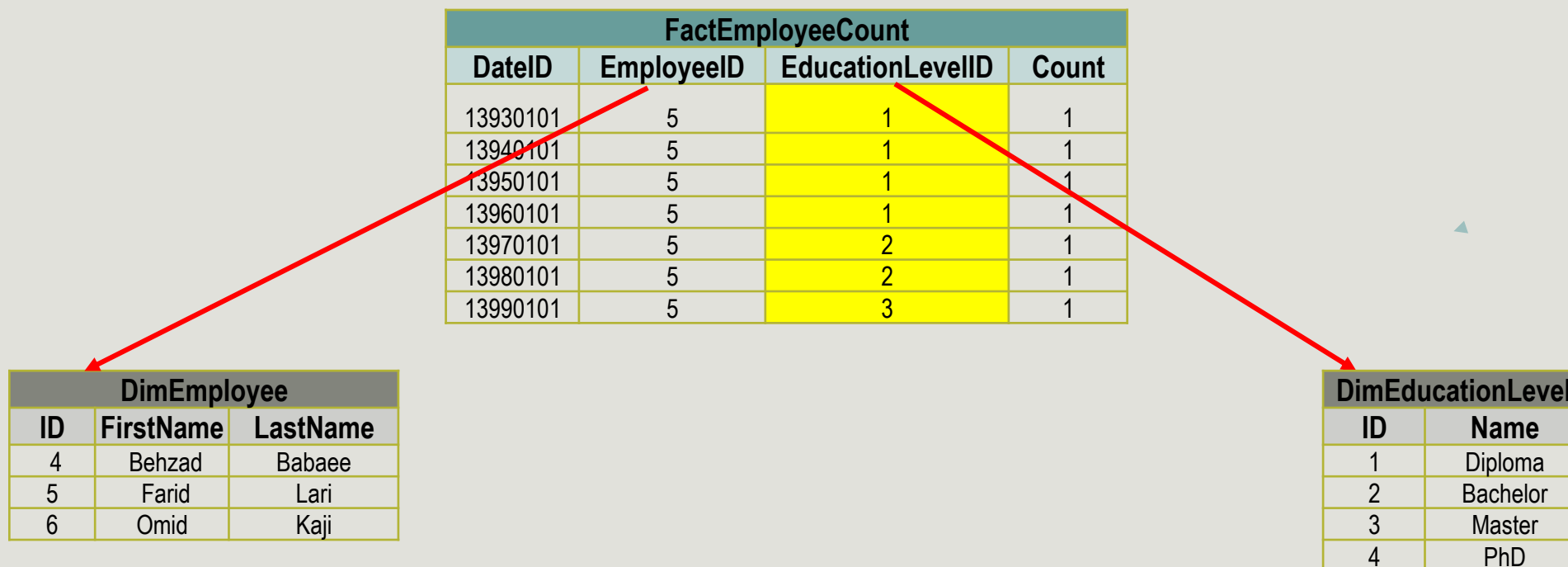
# Slowly Changing Dimension

## 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.

DimLocation			
ID	ParentID1	ParentID2	Title
۱			Tehran
۲	۱	۱	Rey
۳	۱	۴	Karaj
۴			Alborz

# SCD & Star Schema



# SCD (Type1) & Snow Flake Schema

FactEmployeeCount		
DateID	EmployeeID	Count
13930101	5	1
13940101	5	1
13950101	5	1
13960101	5	1
13970101	5	1
13980101	5	1
13990101	5	1

DimEmployee (Type 1)			
ID	FirstName	LastName	EducationLevelID
4	Behzad	Babaei	2
5	Farid	Lari	3
6	Omid	Kaji	2

DimEducationLevel	
ID	Name
1	Diploma
2	Bachelor
3	Master
4	PhD

# SCD (Type2) & Snow Flake Schema

FactEmployeeCount		
DateID	EmployeeID	Count
13930101	5	1
13940101	5	1
13950101	5	1
13960101	5	1
13970101	100	1
13980101	100	1
13990101	200	1

DimEmployee (Type 2)							
ID	SNID	FirstName	LastName	EducationLevelID	FromDate	ToDate	IsLast
4	4	Behzad	Babaei	2			
5	5	Farid	Lari	1	NULL	13961230	False
6	6	Omid	Kaji	2			
100	5	Farid	Lari	2	13970101	13981230	False
200	5	Farid	Lari	3	13990101	NULL	True

DimEducationLevel	
ID	Name
1	Diploma
2	Bachelor
3	Master
4	PhD

# 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