

مراحل تبدیل یک دیتاست به پایگاه داده کامل

مرحله ۱: شناخت دیتاست

در اولین گام، باید دیتاست را به صورت مفهومی بررسی کرد.

دیتاست rainfall شامل اطلاعاتی درباره شهرهای مختلف و ویژگی‌های جغرافیایی و اقلیمی آن‌ها است. این اطلاعات شامل نام شهر، مختصات جغرافیایی، ارتفاع، نوع اقلیم و میزان بارندگی می‌باشد.

در این مرحله مشخص می‌شود که:

- کدام داده‌ها ثابت هستند (مانند مختصات و ارتفاع شهر)
- کدام داده‌ها وابسته به زمان هستند (مانند بارندگی ماهانه و سالانه)
-

مرحله ۲: استخراج موجودیت‌ها (Identifying Entities)

پس از شناخت دیتاست، مفاهیم اصلی موجود در آن استخراج می‌شوند.

در دیتاست rainfall، مفهوم «شهر» به عنوان هسته اصلی اطلاعات شناسایی می‌شود، زیرا تمام داده‌ها به یک شهر خاص وابسته هستند.

در کنار آن، اطلاعات بارندگی به دلیل تغییرپذیری در طول زمان، به عنوان مفاهیم مستقل در نظر گرفته می‌شوند. بنابراین موجودیت‌های اصلی سیستم به شکل زیر تعیین می‌شوند:

- شهر (City)
- بارندگی سالانه (Annual Rainfall)
- بارندگی ماهانه (Monthly Rainfall)

مرحله ۳: تعریف ویژگی‌های هر موجودیت (Defining Attributes)

در این مرحله، ویژگی‌های مربوط به هر موجودیت مشخص می‌شوند.

- موجودیت **City**: شامل اطلاعات پایه و ثابت هر شهر مانند نام شهر، مختصات جغرافیایی، ارتفاع و نوع اقلیم است.
- موجودیت **AnnualRainfall**: شامل سال و میزان بارندگی سالانه برای هر شهر است.
- موجودیت **MonthlyRainfall**: شامل سال، ماه و میزان بارندگی ماهانه برای هر شهر است.

این تفکیک باعث می‌شود اطلاعات به صورت منظم و قابل گسترش ذخیره شوند.

مرحله ۴: تعیین کلیدها (Primary & Foreign Keys)

برای ایجاد ارتباط بین موجودیت‌ها، کلیدها تعریف می‌شوند:

- برای هر موجودیت یک **کلید اصلی (Primary Key)** در نظر گرفته می‌شود.
- موجودیت‌های بارندگی دارای **کلید خارجی (Foreign Key)** هستند که به موجودیت City اشاره می‌کند.

این ساختار امکان اتصال داده‌ها و انجام کوئری‌های تحلیلی را فراهم می‌کند.

مرحله ۵: تعریف روابط بین موجودیت‌ها (Defining Relationships)

در این مرحله روابط منطقی بین موجودیت‌ها مشخص می‌شود:

- هر شهر می‌تواند چندین رکورد بارندگی سالانه داشته باشد → رابطه یک به چند
 - هر شهر می‌تواند چندین رکورد بارندگی ماهانه داشته باشد → رابطه یک به چند
- این روابط در قالب نمودار ERD نمایش داده می‌شوند.

مرحله ۶: نرمال‌سازی داده‌ها (Normalization)

برای جلوگیری از افزونگی داده و افزایش کیفیت طراحی، اصول نرمال‌سازی اعمال می‌شود:

- اطلاعات تکراری حذف می‌شوند
 - داده‌های ثابت و متغیر از هم جدا می‌شوند
 - ساختار پایگاه داده تا حداقل نرمال سوم (۳NF) طراحی می‌شود
- این کار باعث افزایش کارایی و سادگی در نگهداری داده‌ها می‌شود.

مرحله ۷: طراحی مفهومی با (ERD (Conceptual Design)

در این مرحله، ساختار کلی پایگاه داده به صورت یک نمودار ERD نمایش داده می‌شود. این نمودار شامل موجودیت‌ها، ویژگی‌ها و روابط بین آن‌ها است و به عنوان نقشه راه پیاده‌سازی در SQL Server استفاده می‌شود. ERD باعث می‌شود:

- ساختار سیستم قبل از پیاده‌سازی مشخص شود
- خطاهای طراحی کاهش یابد
- توسعه پایگاه داده در آینده ساده‌تر شود

مرحله ۸: آماده‌سازی برای پیاده‌سازی (Ready for Implementation)

پس از تکمیل طراحی مفهومی:

- پایگاه داده آماده پیاده‌سازی در SQL Server است
- اسکرپت‌های SQL بر اساس ERD نوشته می‌شوند
- دیتاست اولیه به جداول طراحی شده منتقل می‌شود

در این مرحله، دیتاست خام به یک پایگاه داده ساخت‌یافته و استاندارد تبدیل شده است.

مراحل تبدیل به یک دیتابیس با چندین جدول در SQL Server

۱. تحلیل داده‌ها و طراحی جدول‌ها (ERD)

جدول ما دارای attribute های زیر می باشد:

city	latitude	longitude	month	year	rainfall	elevation	Climate_Type
nvarchar(۵۰)	float	float	float	float	float	float	nvarchar(۵۰)

rainfall data			
	Column Name	Data Type	Allow Nulls
	city	nvarchar(50)	<input type="checkbox"/>
	latitude	float	<input type="checkbox"/>
	longitude	float	<input type="checkbox"/>
	month	float	<input type="checkbox"/>
	year	float	<input type="checkbox"/>
	rainfall	float	<input type="checkbox"/>
	elevation	float	<input type="checkbox"/>
	Climate_Type	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

که قابل تبدیل به جدول های زیر می باشند:

۱. **City**: اطلاعات ثابت شهرها
 - CityID (Primary Key), CityName, Latitude, Longitude, ClimateType
۲. **Climate**: اطلاعات آب و هوایی (اختیاری، اگر خیلی پیچیده باشد)
 - ClimateID (Primary Key), ClimateType
۳. **Rainfall**: اطلاعات ماهانه و سالانه
 - RainfallID (PK), CityID (FK), Year, Month, Rainfall, Elevation

۲. ساخت جدول ها در SQL Server :

```
CREATE TABLE City (  
    CityID INT IDENTITY(۱,۱) PRIMARY KEY,  
    CityName NVARCHAR(۱۰۰),  
    Latitude FLOAT,  
    Longitude FLOAT,  
    ClimateType NVARCHAR(۵۰)  
);
```

توضیحات **:CityID INT IDENTITY(۱,۱) PRIMARY KEY**

:IDENTITY(۱,۱)

- باعث می‌شود SQL Server خودکار عدد منحصر به فرد به این ستون اختصاص دهد.
- (۱,۱) یعنی شروع از ۱ و افزایش به صورت ۱ واحدی برای هر ردیف جدید.
- مثال: اولین شهر ۱، دومین شهر ۲، سومین ۳ و...

PRIMARY KEY برای CityID:

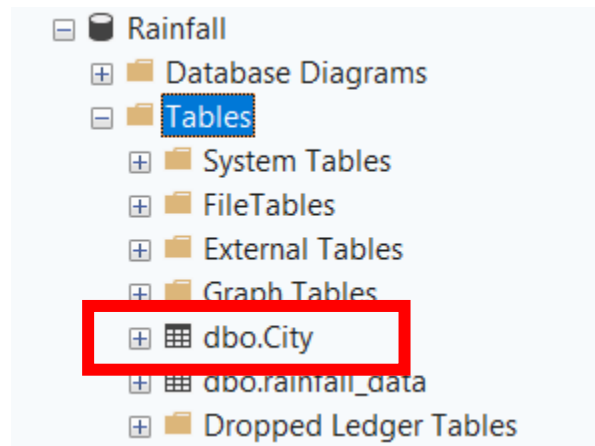
- مشخص می‌کند که این ستون **شناسه یکتا برای هر ردیف** است که دو ویژگی مهم دارد:
 ۱. **یکتا بودن**: هیچ دو ردیفی نمی‌تواند CityID یکسان داشته باشد.
 ۲. **مقدار Null ندارد**: همیشه باید مقدار داشته باشد.

اجرای کوئری:

- ابتدا یک کوئری جدید بر روی دیتابیس Rainfall ایجاد می‌کنیم و سپس کوئری بالا را اجرا می‌کنیم:

```
SQLQuery2.s...ahtab (59)* X MahtabPC.Rain...rainfall_data SQLQuery1.s...Mahta
1 CREATE TABLE City (
2     CityID INT IDENTITY(1,1) PRIMARY KEY,
3     CityName NVARCHAR(100),
4     Latitude FLOAT,
5     Longitude FLOAT,
6     ClimateType NVARCHAR(50)
7 );
```

وقتی اجرا کنیم، پیام موفقیت آمیز بودن را نشان می‌دهد و به این معنی است جدول جدید ساخته شد. سپس با یک Refresh کردن بر روی فولدر جدول‌ها، جدول جدید را مشاهده می‌کنیم:



به این ترتیب این مراحل را برای ایجاد بقیه جدول‌ها دنبال می‌کنیم.

```
CREATE TABLE Rainfall (
    RainfallID INT IDENTITY(1,1) PRIMARY KEY,
    CityID INT FOREIGN KEY REFERENCES City(CityID),
    year INT,
    month INT,
    rainfall FLOAT,
    elevation FLOAT
);
```

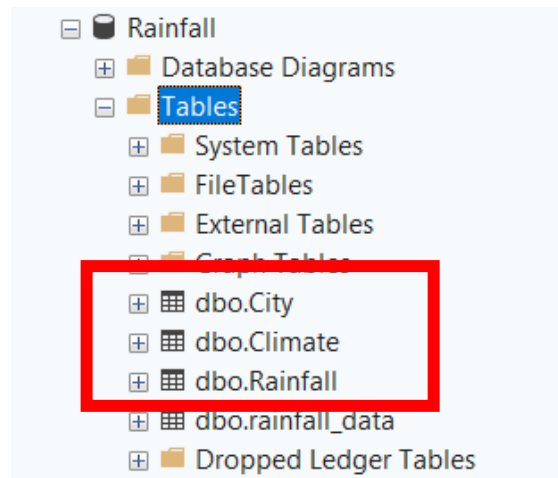
توضیحات CityID INT FOREIGN KEY REFERENCES City(CityID):

- CityID INT: ستونی در جدول Rainfall که مقدارش عدد صحیح است و همان شناسه شهر است.
- FOREIGN KEY:

- این ستون یک کلید خارجی است.
 - یعنی مقدارش باید در جدول دیگر وجود داشته باشد (در اینجا جدول City).
 - REFERENCES City(CityID):
 - مشخص می‌کند که این کلید خارجی به ستون CityID در جدول City اشاره دارد.
 - بنابراین هر ردیف بارندگی به یک شهر معتبر لینک می‌شود و نمی‌توان برای ردیف بارندگی، شهری که در جدول City نیست، ثبت کرد.
- نتیجه: این کار باعث می‌شود یکپارچگی داده‌ها (Data Integrity) حفظ شود و جدول Rainfall فقط شهرهایی که در جدول City وجود دارند، داشته باشد.

```
CREATE TABLE Climate (
    ClimateID INT IDENTITY(1,1) PRIMARY KEY,
    Climate_Type NVARCHAR(50)
);
```

جدول های ایجاد شده:



۳. پر کردن جدول‌ها از دیتاست اصلی

(۱) پر کردن جدول City:

```
INSERT INTO City (CityName, Latitude, Longitude, ClimateType)
SELECT DISTINCT City, Latitude, Longitude, Climate_Type
FROM rainfall_data;
```

توضیحات:

این کوئری تمام شهرهای یکتا را وارد جدول City می‌کند. دستور DISTINCT باعث می‌شود فقط یکبار برای هر شهر یکتا اطلاعات درج شود و داده تکراری وارد جدول City نشود. FROM Rainfall: مشخص می‌کند داده‌ها از کدام جدول بزرگ منبع گرفته شوند.

۲) پر کردن جدول Rainfall:

```
INSERT INTO Rainfall (CityID, Year, Month, Rainfall, elevation)
SELECT c.CityID, R.Year, R.Month, R.Rainfall, R. elevation
FROM rainfall_data R
JOIN City c ON R.City = c.CityName;
```

توضیحات:

داده‌های ماهانه و سالانه هر شهر به جدول Rainfall منتقل می‌شوند و به جدول City لینک شده‌اند. از JOIN استفاده شده تا داده‌های بارندگی به شناسه شهر (CityID) مرتبط شوند. این کار تضمین می‌کند که جدول Rainfall فقط شناسه شهر را ذخیره کند، نه نام شهر، که به نرمال‌سازی پایگاه داده کمک می‌کند.

۳) پر کردن جدول Climate:

```
INSERT INTO Climate (Climate_Type)
SELECT DISTINCT Climate_Type
FROM rainfall_data;
```

با اجرای `SELECT *` دیتاهای هر جدول را مشاهده می‌کنیم که برای هر ۳ جدول با موفقیت انجام شده است:

MahtabPC.Rainfall - dbo.City

SQLQuery1.s...ahtab (63))*

1

2

3

4

5

6

SELECT * FROM CITY;

SELECT * FROM Climate;

SELECT * FROM Rainfall;

108 %

No issues found

Results

Messages

	CityID	CityName	Latitude	Longitude	ClimateType
1	1	Ahvaz	31.318000793457	48.6689987182617	Desert Climate (BWh)
2	2	Arak	34.0909996032715	49.6889991760254	Cold Semi-Arid Climate (BSk)
3	3	Ardabil	38.2490005493164	48.2949981689453	Humid Continental Climate (Dsb)
4	4	Bandar Abbas	27.1830005645752	56.265998840332	Hot Desert Climate (BWh)
5	5	Bandar-e Bushehr	28.9230003356934	50.8199996948242	Hot Semi-Arid Climate (BSH)
6	6	Birjand	32.8660011291504	59.2169990539551	Cold Semi-Arid Climate (BSk)
7	7	Bojnurd	37.4739990234375	57.3289985656738	NULL
8	8	Esfahan	32.6529998779297	51.6669998168945	Cold Semi-Arid Climate (BSk)

	ClimateID	Climate_Type
1	1	Mediterranean Climate (Csa)
2	2	NULL
3	3	Cold Desert Climate (BWk)
4	4	Humid Continental Climate (Dsb)
5	5	Cold Semi-Arid Climate (BSk)
6	6	Hot Desert Climate (BWh)
7	7	Hot Semi-Arid Climate (BSH)
8	8	Humid Subtropical Climate (Cfa)

	RainfallID	CityID	year	month	rainfall	elevation
1	1	1	2000	1	68.4557037353516	20
2	2	2	2000	1	43.0526008605957	1708
3	3	3	2000	1	44.9910011291504	1350
4	4	4	2000	1	94.6748962402344	9
5	5	5	2000	1	99.7760009765625	18
6	6	6	2000	1	34.5848999023438	1491
7	7	7	2000	1	35.9112014770508	1070
8	8	8	2000	1	18.1595993041992	1574
9	9	9	2000	1	47.847599029541	155
10	10	10	2000	1	52.6425018310547	1741

