



عنوان مقاله: انواع کلید در پایگاه داده + بررسی کاربرد آن ها در SQL Server

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: ۲۴ بهمن ۱۴۰۲

منبع: <https://nikamooz.com/types-of-keys-in-the-database/>

کلید در پایگاه داده ، یک بخش بنیادی در سیستم‌های مدیریت پایگاه داده (DBMS) به‌شمار می‌رود و انواع مختلفی دارد. شما می‌توانید کلیدها را براساس نوع آن‌ها در سناریوهای گوناگون به کار ببرید تا بدین طریق، یکپارچگی و کارایی دیتابیس تضمین شود. از میان انواع کلید در دیتابیس ، کلید اصلی (Primary Key) و خارجی (Foreign Key) اهمیت بیشتری دارند؛ با این وجود، در این مطلب به بررسی انواع آن می‌پردازیم. پیش از مطالعه این مقاله، پیشنهاد می‌شود [آموزش جامع SQL Server](#) را نیز مطالعه کنید تا با تمامی مفاهیم پیش‌نیاز برای کار با این سیستم مدیریت پایگاه داده آشنا شوید.

### مفهوم کلید در پایگاه داده

مفهوم کلید در پایگاه داده ، یک یا چند صفت (Attribute) است که به شما کمک می‌کند تا یک رکورد از جدول را به‌صورت یکتا تشخیص دهید. به بیان دیگر، کلید در دیتابیس مشابه یک [شناسه گر منحصر به فرد](#) (Unique Identifier) برای یک رکورد خاص عمل می‌کند. فرض کنید یک کتابخانه، مملو از انواع کتاب در اختیاران است؛ به طوری که هر یک از کتاب‌ها بارکد مخصوص خود را دارا باشند. در چنین شرایطی، وجود چنین بارکدهای متمایزی در یافتن کتاب مدنظر به شما کمک خواهد. استفاده از کلید نیز چنین مزیتی را برای داده‌های درون پایگاه داده به ارمغان می‌آورند. علاوه بر این، به‌واسطه تعریف مفهوم کلید در پایگاه داده ، [رابطه](#) (Relationship) میان جداول و ستون‌ها در [دیتابیس رابطه ای](#) (RDBMS) تعیین می‌شود.



## دلیل اهمیت کلید در پایگاه داده

برخی از دلایل اهمیت مفهوم کلید در پایگاه داده و استفاده از آن عبارتند از:

- **یکتایی (Uniqueness):** با تعریف کلید در پایگاه داده، تضمین می‌شود که هیچ دو رکوردی به طور یکسان در جدول وجود ندارند؛ بدین طریق، از درج داده‌های تکراری (Duplicated Data) جلوگیری می‌شود و خیالتان از بابت دقت (Accuracy) داده‌ها راحت خواهد بود.
- **استخراج کارآمد داده‌ها:** به دلیل وجود کلید در دیتابیس، می‌توان به راحتی رکوردهای خاصی را از جداول استخراج کرد و از این مزیت، در دیتابیس‌های گسترده بهره‌مند شد.
- **یکپارچگی داده‌ها:** با اعمال یکتایی از طریق کلید در پایگاه داده، از ویرایش یا حذف سهوی داده‌ها جلوگیری می‌شود و تطبیق‌پذیری داده‌ها به درستی رعایت خواهد شد.
- **روابط:** کلیدها مانند پل ارتباطی برای ارتباط میان جداول پایگاه داده عمل می‌کنند. با ارجاع دادن به **کلید اصلی (Primary Key)** جدول دیگر، یک **کلید خارجی (Foreign Key)** برای ارتباط داده‌های مرتبط تعیین می‌شود. این موضوع در تحلیل و درک صحیح ارتباط میان اطلاعات مختلف کارآمد است. به عنوان مثال، لینک کردن سفارش‌های مشتریان به محصولات خریداری شده از این طریق امکان‌پذیر است.
- **بهینه‌سازی:** کلید در پایگاه داده در بهبود کارایی دیتابیس‌ها نقش کلیدی ایفا می‌کند. آن‌ها به عنوان **نقاط ایندکس گذاری (Indexing)** عمل می‌کنند و در بهبود سرعت کوئری‌ها به شما کمک خواهند کرد.

## انواع کلیدها در پایگاه داده

مهم‌ترین انواع کلید در پایگاه داده به شرح زیر است:



## ابر کلید (Super Key)

سوپر کلید یا ابر کلید در پایگاه داده، مجموعه‌ای از یک یا چند ستون جدول است که به واسطه آن، رکوردهای جدول از یکدیگر متمایز می‌شوند. برخلاف کلید کاندید، لازم نیست ابرکلید به صورت مینیمال یا بهینه باشد. به عنوان مثال، در جدول دانشجویان، ترکیب شماره دانشجویی و ایمیل می‌تواند یک ابرکلید باشد؛ زیرا هر دانشجو دارای یک ایمیل آدرس و شماره دانشجویی متمایز است. با این وجود، اگر شماره دانشجویی به تنهایی یک Unique Identifier برای هر دانشجو تلقی شود، می‌توان آن را یک کلید کاندید و به دنبال آن، یک ابرکلید در نظر گرفت. به همین دلیل است که سوپر کلید لزوماً مینیمال‌ترین حالت یکتایی محسوب نمی‌شود.

برای درک بهتر مفهوم کلید در پایگاه داده، آن را با کمک [دیتابیس Northwind](#) بررسی خواهیم کرد. تصویر زیر، Design مربوط به جدول Customer از این Database است.

Column Name	Data Type	Allow Nulls
CustomerID	nchar(5)	<input type="checkbox"/>
CompanyName	nvarchar(40)	<input type="checkbox"/>
ContactName	nvarchar(30)	<input checked="" type="checkbox"/>
ContactTitle	nvarchar(30)	<input checked="" type="checkbox"/>
Address	nvarchar(60)	<input checked="" type="checkbox"/>
City	nvarchar(15)	<input checked="" type="checkbox"/>
Region	nvarchar(15)	<input checked="" type="checkbox"/>
PostalCode	nvarchar(10)	<input checked="" type="checkbox"/>
Country	nvarchar(15)	<input checked="" type="checkbox"/>
Phone	nvarchar(24)	<input checked="" type="checkbox"/>
Fax	nvarchar(24)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

در این جدول، CustomerID به عنوان کلید اصلی، هر یک از مشتریان را از یکدیگر متمایز می‌کند.

ابر کلیدهای این جدول عبارتند از:

- **CustomerID**: بدیهی است که کلید اصلی یک جدول، یک Superkey باشد.
- **CustomerID و CompanyName**: ترکیب نام کارخانه و شماره مشتری یکتا خواهد بود. این موضوع حتی در شرایطی که نام‌ها مشترک باشند نیز صادق است.
- **CustomerID و ContactName**: مشابه مورد فوق، ID و Contact Name مشتریان، یکتایی را به شما تضمین می‌دهند.

به خاطر داشته باشید که ابرکلیدها می‌توانند ترکیبی از چندین **Attribute** مختلف باشند؛ به طوری که تمامی ستون‌ها را به صورت یکتا متمایز کنند. بنابراین، شما می‌توانید انواع ترکیب‌ها را براساس مشخصه‌ها و محدودیت‌های جداول خود اعمال کنید. هرچند در اغلب موارد، از کلید اصلی و کلید خارجی برای استخراج و دستکاری داده‌ها به‌طور کارآمد استفاده می‌شود.

## کلید کاندید (Candidate Key)

کلید کاندید مجموعه‌ای از یک یا چند ستون از جدول به‌شمار می‌آید که به کمک آن، یکتایی هر رکورد از سایر سطرها ممکن خواهد شد. درحقیقت، کلید کاندید نوع مینیمال و بهینه‌شده ابر کلید است و حداقل ستون‌های لازم برای رسیدن به یکتایی را دربر می‌گیرد. به‌عنوان مثال، در جدول کارمندان، هم شماره کارمندی و هم ترکیب نام و نام خانوادگی کارمندان، هر دو می‌توانند کلید کاندید در نظر گرفته شوند؛ زیرا هر کارمند دارای ID خاصی است و ترکیب نام و نام خانوادگی هر شخص نیز می‌تواند منحصر به فرد باشد. معمولاً یکی از این کلیدها به‌عنوان کلید اصلی و دیگری به‌عنوان [کلید Alternative](#) انتخاب می‌شود.

تصویر زیر، نمای جدول Orders از دیتابیس Northwind است.

Column Name	Data Type	Allow Nulls
OrderID	int	<input type="checkbox"/>
CustomerID	nchar(5)	<input checked="" type="checkbox"/>
EmployeeID	int	<input checked="" type="checkbox"/>
OrderDate	datetime	<input checked="" type="checkbox"/>
RequiredDate	datetime	<input checked="" type="checkbox"/>
ShippedDate	datetime	<input checked="" type="checkbox"/>
ShipVia	int	<input checked="" type="checkbox"/>
Freight	money	<input checked="" type="checkbox"/>
ShipName	nvarchar(40)	<input checked="" type="checkbox"/>
ShipAddress	nvarchar(60)	<input checked="" type="checkbox"/>
ShipCity	nvarchar(15)	<input checked="" type="checkbox"/>
ShipRegion	nvarchar(15)	<input checked="" type="checkbox"/>
ShipPostalCode	nvarchar(10)	<input checked="" type="checkbox"/>
ShipCountry	nvarchar(15)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

برای درک بهتر، قصد داریم به بررسی جدول Orders (سفارش‌ها) از دیتابیس Northwind بپردازیم.

- **OrderID**: این ستون، کلید اصلی جدول Orders است و به‌طور خودکار، یک کلید کاندید نیز محسوب می‌شود. این یعنی، کلید اصلی خود یک کلید کاندید به‌حساب می‌آید.
- **OrderDate و CustomerID**: به‌واسطه این ترکیب، احتمالاً امکان تعیین هر سفارش مشتری در تاریخ خاصی وجود دارد. به همین دلیل، می‌توان آن را کلید کاندید بالقوه در نظر داشت.
- **ProductID و CustomerID**: در شرایطی که مشتری یک محصول خاص را فقط یک بار سفارش دهد، می‌توان اذعان داشت که این ترکیب، پتانسیل یکتا بودن را دارد.

### کلید اصلی (Primary Key)

کلید اصلی در پایگاه داده، یک یا چند ستون جدول است که هر رکورد را از سایر آن‌ها متمایز می‌کند. به واسطه استفاده از کلید اصلی در دیتابیس، این تضمین به وجود می‌آید که رکوردهای تکراری در جدول وجود ندارند. به‌طور پیش‌فرض، کلید اصلی نمی‌تواند مقدار Null بگیرد و لازم است مقادیر مربوط به آن Unique باشند. برای آشنایی با انواع Constraint، پیشنهاد می‌شود [مقاله تعریف محدودیت \(Constraint\) در SQL Server + بررسی انواع و کاربردهای آن‌ها](#) را نیز مطالعه کنید.

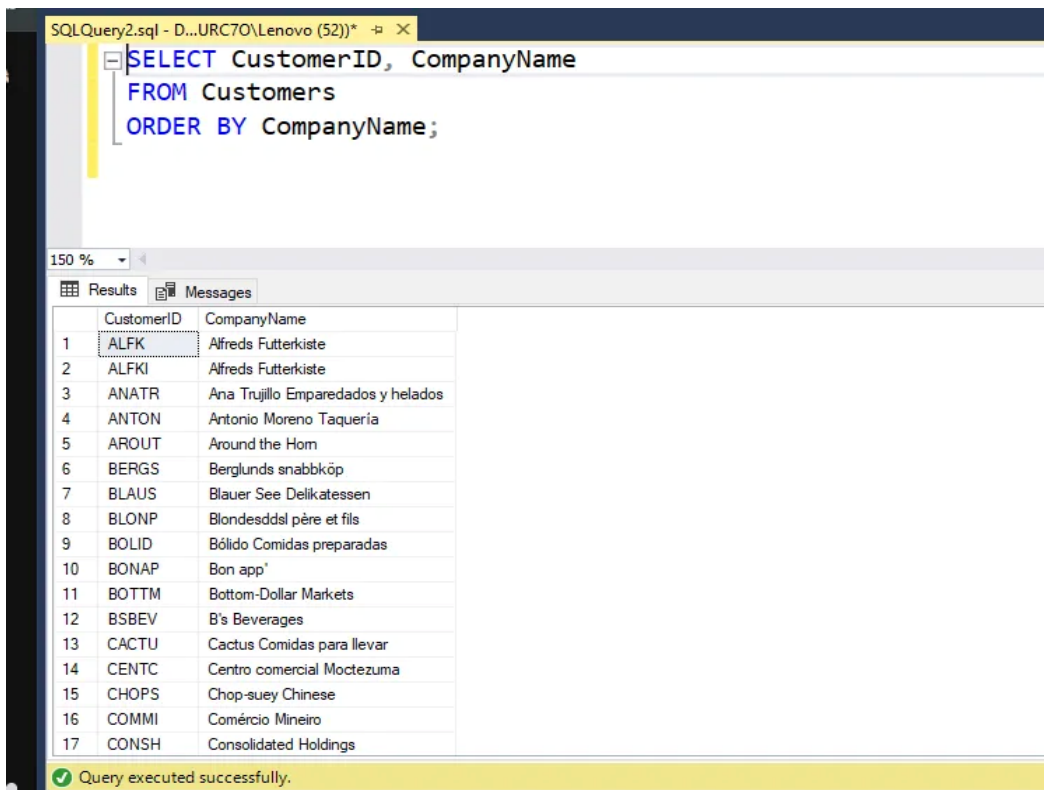
کلید اصلی، اغلب به‌طور خودکار توسط سیستم مدیریت پایگاه داده (DBMS) ایندکس‌گذاری می‌شود. بدین طریق، سرعت عملیات استخراج داده‌ها بهبود می‌یابد و اموری همچون جستجو (Search) و مرتب‌سازی (Sorting) تسهیل پیدا خواهند کرد. هنگامی که مقادیر کلید اصلی تعیین می‌شوند، دیگر امکان تغییر آن‌ها وجود نخواهد داشت؛ به همین دلیل است که می‌توان کلید اصلی را به‌عنوان یک Unique Identifier پایدار در نظر داشت. برای مثال، ستون Customer ID در جدول مشتریان، Product ID در جدول محصولات یا شماره سفارش در جدول سفارشات، همگی کلید اصلی برای چنین جدول‌هایی محسوب شوند.

برای درک بهتر اهمیت کلید اصلی در پایگاه داده و کاربردهای آن، به مثال زیر توجه کنید:

فرض کنید قصد داریم نام مشتریان و نام شرکت مربوط به آن‌ها را لیست کنیم.

```
SELECT CustomerID, CompanyName
FROM Customers
ORDER BY CompanyName;
```

این کوئری به‌سادگی CustomerID و CompanyName را از جدول مشتریان استخراج می‌کند. در این مثال ساده، فیلترینگ داده‌ها وجود ندارد و می‌توان آن را یک نمونه پایه برای نحوه استفاده از کلید اصلی در پایگاه داده در نظر گرفت. توجه کنید که **کلز Order By**، نتایج حاصل را براساس نام شرکت‌ها مرتب‌سازی خواهد کرد.



## کلید خارجی (Foreign Key)

کلید خارجی (FK | Foreign Key)، آن ستون از جدول است که به کلید اصلی یا **Unique Key** یک جدول دیگر ارجاع داده می‌شود. این نوع از کلید در پایگاه داده، به عنوان یک لینک و به منظور برقراری ارتباط میان داده‌های مرتبط در جداول گوناگون مورد استفاده قرار می‌گیرد. وجود کلید خارجی به دلایلی همچون، اعمال **یکپارچگی ارجاعی** (Referential Integrity)، تعیین روابط صریح میان جداول و امکان استفاده از Join های کارآمد میان آن‌ها، نقش بنیادی در RDBMS ها ایفا می‌کنند.

فرض کنید می‌خواهیم تمامی سفارش‌هایی که توسط مشتریان از آلمان ثبت شده را به همراه اطلاعات آن‌ها نمایش دهیم. مشابه کوئری زیر، ابتدا باید جداول Orders و Customers را در نظر داشته باشیم و فیلترینگ داده‌ها را براساس کشور مشتری انجام دهیم. این فرآیند، از طریق رابطه کلید خارجی و اعمال **Join** امکان‌پذیر است.

```
SELECT o.OrderID, o.OrderDate, c.CompanyName, c.ContactName, c.Country
FROM Orders o
INNER JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE c.Country = 'Germany';
```

در این کوئری، از یک **Inner Join** میان دو جدول Orders و Customers استفاده شده؛ این عمل، بر مبنای کلید خارجی CustomerID انجام می‌شود. درحقیقت، کلید فرعی در جدول Order به کلید اصلی در جدول Customers ارجاع داده شده و یک رابطه والد - فرزندی میان آن‌ها صورت گرفته است.

The screenshot shows a SQL query window with the following text:

```
SELECT o.OrderID, o.OrderDate, c.CompanyName, c.ContactName, c.Country
FROM Orders o
INNER JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE c.Country = 'Germany';
```

Below the query window, a results grid is displayed with the following data:

OrderID	OrderDate	CompanyName	ContactName	Country
10249	1996-07-05 00:00:00.000	Toms Spezialitäten	Karin Josephs	Germany
10260	1996-07-19 00:00:00.000	Ottilies Käseladen	Henriette Pfalzheim	Germany
10267	1996-07-29 00:00:00.000	Frankenversand	Peter Franken	Germany
10273	1996-08-05 00:00:00.000	QUICK-Stop	Horst Kloss	Germany
10277	1996-08-09 00:00:00.000	Morgenstern Gesundkost	Alexander Feuer	Germany
10279	1996-08-13 00:00:00.000	Lehmanns Marktstand	Renate Messner	Germany
10284	1996-08-19 00:00:00.000	Lehmanns Marktstand	Renate Messner	Germany
10285	1996-08-20 00:00:00.000	QUICK-Stop	Horst Kloss	Germany
10286	1996-08-21 00:00:00.000	QUICK-Stop	Horst Kloss	Germany
10301	1996-09-09 00:00:00.000	Die Wandemde Kuh	Rita Müller	Germany
10312	1996-09-23 00:00:00.000	Die Wandemde Kuh	Rita Müller	Germany
10313	1996-09-24 00:00:00.000	QUICK-Stop	Horst Kloss	Germany
10323	1996-10-07 00:00:00.000	Königlich Essen	Philip Cramer	Germany
10325	1996-10-09 00:00:00.000	Königlich Essen	Philip Cramer	Germany
10337	1996-10-24 00:00:00.000	Frankenversand	Peter Franken	Germany
10342	1996-10-30 00:00:00.000	Frankenversand	Peter Franken	Germany
10343	1996-10-31 00:00:00.000	Lehmanns Marktstand	Renate Messner	Germany

به طور کلی، کلید فرعی به منظور اتصال جداول، یکپارچگی ارجاعی و استخراج کارآمد داده‌ها کاربرد دارد و شما می‌توانید آن را در فیلترینگ و جوین روی جداول استفاده کنید.

### کلید فرعی (Alternative Key)

کلید فرعی (Alternative Key) همان کلید کاندیدی است که به عنوان Primary Key یا همان کلید اصلی انتخاب نشده باشد. یک جدول می‌تواند فقط یک کلید اصلی داشته باشد، اما شما می‌توانید برای همان جدول چند کلید فرعی بالقوه داشته باشید. مشابه PK، هر یک از کلیدهای فرعی، یکتایی هر رکورد از جدول را تضمین می‌کنند. کلید فرعی در پایگاه داده به هدف بهینه‌سازی کارایی، ایندکس‌گذاری، Identifier های پشتیبان و اعمال یکتایی روی جدول به کار می‌روند.

کلیدهای فرعی به طور صریح در دیتابیس Northwind تعیین نشده‌اند، اما می‌توان به بررسی مواردی که احتمالاً کلید کاندید هستند پرداخت. جدول Customers از این دیتابیس را در نظر بگیرید؛ پیش‌تر اشاره کردیم که در این جدول، CustomerID کلید اصلی است. اگر فرض کنیم نام شرکت‌ها یکتا و منحصر به فرد هستند، می‌توان CompanyName را یک کلید فرعی یا همان Alternative تلقی کرد. در این سناریو، با استفاده از CompanyName، امکان تشخیص هر مشتری از سایرین فراهم می‌شود.



## کلید مرکب (Compound Key)

کلید مرکب در DBMS، نوعی Identifier یکتا به حساب می‌آید که در قالب ترکیبی از ستون‌ها ایجاد شده است. این یعنی، برخلاف یک کلید ساده که در آن یک ستون برای اعمال Uniqueness به کار می‌رود، کلید مرکب به ترکیبی از ستون‌های یکتا برای هر رکورد از جدول اکتفا خواهد کرد. هرچند این نوع از کلید در پایگاه داده برای برخی از سناریوها مناسب است، اما در اغلب موارد افزایش پیچیدگی را به همراه دارد.

## کلید ترکیبی (Composite Key)

کلید ترکیبی یک نوع خاصی از کلید کاندید است که یک یا چند Attribute (ستون) را شامل می‌شود. به بیان ساده، Composite Key نوعی Unique Identifier است که از ترکیب چندین ستون به وجود آمده است. تفاوت کلید مرکب و ترکیبی این است که کلید ترکیبی به عنوان کلید اصلی جدول در نظر گرفته می‌شود؛ در حالی که کلیدهای مرکب می‌توانند هر شناسه منحصر به فردی از جمله کلیدهای اصلی، کلیدهای جایگزین یا کلیدهای خارجی باشند.

کلید ترکیبی مزیت‌های مانند تضمین یکپارچگی داده‌ها، انعطاف‌پذیری و امکان استفاده از ایندکس‌ها روی آن‌ها را به همراه دارند؛ با این وجود، ممکن است به افزایش پیچیدگی کوئری‌ها و تأثیر منفی روی کارایی منجر شوند. بنابراین پیش از استفاده، بهتر است بررسی کنید که آیا استفاده از یک کلید اصلی یک ستونه یا کلید جایگزین برای سناریو مدنظر کافی است یا خیر.

## مقایسه کلید اصلی و کلید خارجی

کلید اصلی در پایگاه داده، همانند یک Fingerprint برای هر رکورد از جدول عمل می‌کند و به واسطه آن، از تکرار داده‌ها جلوگیری می‌شود. استفاده از کلید اصلی مزایایی مانند استخراج سریع داده‌ها را به همراه دارد و به عنوان یک نقطه ارجاعی برای اتصال به سایر جداول تلقی می‌شود. از سوی دیگر، کلید خارجی مانند یک پاسپورت عمل می‌کنند که شما می‌توانید برای اتصال داده‌ها روی جداول مختلف استفاده کنید. این فرآیند، با ارجاع دادن به کلید اصلی یک جدول دیگر امکان‌پذیر است و مزایایی همچون سازگاری داده‌ها را برای شما به ارمغان می‌آورد. از میان انواع کلید در پایگاه داده، این دو نوع به وفور و به عنوان بخش اساسی از DBMS ها مورد استفاده قرار می‌گیرند.

## عوامل موثر در انتخاب نوع کلید

شما باید بتوانید کلید در پایگاه داده را به گونه‌ای انتخاب کنید که دقیقاً مناسب دیتابیس باشد. درحقیقت، این انتخاب اگر به شکل صحیح انجام شود، یکپارچگی داده‌ها، بهبود کارایی و انعطاف‌پذیری را در اختیارتان قرار می‌دهد.



## انتخاب Primary Key در پایگاه داده

موارد زیر در انتخاب کلید اصلی نقش بسزایی دارند:

- **یکتایی:** انتخاب ستون‌هایی که یکتایی هر رکورد را تضمین کنند.
- **استخراج کارآمد داده‌ها:** توجه کنید که به چه میزان یک کلید می‌تواند برای استخراج داده‌ها و جستجوی سریع به کار برده شود.
- **پایداری:** از انتخاب ستون‌هایی پرهیز شود که به وفور تغییر می‌کنند؛ زیرا این عمل، نیازمند آپدیت ارجاعات در تمامی دیتابیس است و می‌تواند روی یکپارچگی دیتا تأثیر نامناسبی ایجاد کند.
- **مقادیر Null حداقلی:** به طور ایده‌آل، کلید اصلی نباید مقدار Null داشته باشد تا از ابهام اجتناب شود.

## انتخاب Foreign Key در پایگاه داده

عوامل مؤثر در انتخاب کلید خارجی عبارتند از:

- **یکپارچگی ارجاعی:** مطمئن شوید که کلید خارجی انتخابی به یک کلید اصلی قابل قبول در جدول مرتبط ارجاع داده می‌شود. علاوه بر این، از ارجاعات Circular میان جداول مختلف اجتناب کنید.
- **کاردینالیتی روابط:** نوع ارتباط میان جداول را در نظر بگیرید؛ این موضوع در تعیین اینکه آیا یک کلید خارجی واحد یا ترکیبی نیاز است به شما کمک خواهد کرد. منظور از کاردینالیتی، یک به یک بودن (One-To-One)، یک به چند بودن (One-To-Many) یا چند به چند بودن (Many-To-Many) روابط است.
- **کارایی:** شما می‌توانید هنگام اعمال جویین روی این نوع روابط، از ایندکس‌گذاری کلیدهای خارجی استفاده کنید تا بدین طریق، کارایی کوئری‌ها بهبود بیابد.
- **انعطاف‌پذیری مدل‌سازی داده‌ها:** در صورتی که مدل داده شما در طول زمان تکامل یابد، می‌توانید از ساختار منعطف کلید خارجی برای تطبیق‌پذیری نسبت به تغییرات بهره‌مند شوید.

## موارد جانبی مؤثر در انتخاب کلید

انتخاب کلید در پایگاه داده می‌تواند تحت تأثیر موارد زیر نیز باشد:

- **نرمال‌سازی پایگاه داده:** با استفاده از **نرمال‌سازی** (Normalization) پایگاه داده می‌توان به کمینه‌سازی افزونگی (Redundancy) و بهینه‌سازی ساختار داده‌ها پرداخت کرد.
- **نیازمندی‌های کارایی:** در شرایطی که کوئری‌های خاصی برای اپلیکیشن شما نیاز باشند، بهتر است بررسی کنید که چه کلیدهایی روی کارایی مؤثر خواهند بود.
- **نوع‌های داده:** انتخاب **نوع‌های داده** که برای کلید مناسب باشد، حائز اهمیت فراوانی است. عملیات دیتابیس مانند مقایسه و فیلترینگ داده‌ها بر کلیدها متکی هستند. این یعنی، نوع‌های داده مختلف از لحاظ کارایی می‌توانند متفاوت عمل کنند.

## بهینه‌سازی عملکرد با استفاده از کلیدها

به دلیل اهمیت انواع کلید در پایگاه داده، به‌ویژه کلید اصلی و خارجی، می‌توان آن‌ها به روش‌های زیر و به‌منظور بهینه‌سازی عملکرد دیتابیس استفاده کرد.

- پیاده‌سازی ایندکس‌گذاری به‌صورت کارآمد
- انتخاب نوع داده مناسب
- اعمال یکپارچگی ارجاعی
- دقت در تعیین **قوانین Cascading**
- اجتناب از ارجاعات دایره‌وار یا همان Circular
- نرمال‌سازی دیتابیس و کاهش افزونگی‌های آن
- پیگیری الگوها و معیارهای استفاده از کلید و بهینه‌سازی هرچه بیشتر
- تطبیق رویکرد براساس الگوهای کوئری‌نویسی و نیازمندی‌های عملکردی



### سخن پایانی: معرفی انواع کلید در پایگاه داده

انتخاب کلید در پایگاه داده و نحوه استفاده از آن، به نیازمندی‌ها و مدل داده بستگی دارد. در این مقاله به بررسی کارکرد انواع کلید در پایگاه داده پرداخته شد؛ اکنون می‌توانید با درک دقیق کاربردهای هر یک، مواردی مانند ذخیره‌سازی، مدیریت و استخراج داده‌ها از دیتابیس را به صورت کارآمد و دقیق انجام دهید. علاوه بر این، افراد علاقه‌مند به توسعه دانش در زمینه کوئری‌نویسی و کار با دیتابیس، می‌توانند [مقاله پرکاربردترین دستورات SQL Server و اسکرپت های SQL](#) را نیز مطالعه کنند.