

第10期-数据库索引与备用键约束

2019年4月23日 09:24

索引原理与配置

索引是许多数据常见概念，数据存储区中的实现可能有所不同，索引的作用是让基于列（或一组列）的查找更高效。

[数据库索引是什么？](#)

[数据库索引融会贯通](#)

[20分钟数据库索引设计实战](#)

[数据库索引为什么用B+树实现？](#)

[深入浅出数据库索引原理](#)

```
modelBuilder.Entity<Blog>().HasIndex(b => b.Url);
```

```
modelBuilder.Entity<Blog>().HasIndex(b => b.Url).IsUnique();
```

```
modelBuilder.Entity<Person>().HasIndex(p => new { p.FirstName, p.LastName });
```

按照约定，EF Core 生成的索引名为 IX_<type name>_<property name> 格式。对于复合索引 <property name> 使用下划线分隔属性名称，可手动指定索引名称。

```
modelBuilder.Entity<Blog>().HasIndex(b => b.Url).HasName("Index_Url");
```

```
modelBuilder.Entity<Blog>().HasIndex(b => b.Url).HasFilter("[Url] IS NOT NULL");
```

```
modelBuilder.Entity<Blog>().HasIndex(b => b.Url).IsUnique().HasFilter(null);
```

备用键

除主键之外，备用键也能唯一标识一条数据（跟主键一样具有唯一约束）。备用键可以用作外键关系的目标。当使用关系数据库时，系统通常会在需要时默认你引入备用键，你无需手动配置它们，当然也可以手动配置。

按照约定，系统将在识别外键属性目标（不是主键）时为你引入备用键，充当关系的目标。

```
modelBuilder.Entity<Post>().HasOne(p => p.Blog).WithMany(b => b.Posts).HasForeignKey(p => p.BlogUrl).HasPrincipalKey(b => b.Url);
```

Fluent API 可用于手动配置要作为备用键的单个属性。

```
modelBuilder.Entity<Car>().HasAlternateKey(c => c.LicensePlate);
```

```
modelBuilder.Entity<Car>().HasAlternateKey(c => new { c.State, c.LicensePlate });
```

按照约定，备用键的索引和约束被命名为 AK_<type name>_<property name> 格式，备用组合键 <property name> 为

下划线分隔属性名称的列表。

可以使用 Fluent API 配置备用键的索引和约束名称。

```
modelBuilder.Entity<Car>().HasAlternateKey(c => c.LicensePlate).HasName("AlternateKey_LicensePlate");
```