

# Django ORM Model Relationships

## العلاقات في قواعد البيانات ذات العلاقة

أحدى أهم مزايا قواعد البيانات ذات العلاقة (Relational Database) هي العلاقات التي بالإمكان استخدامها بين جدول (مودل) و جدول آخر بناء على ما يربط بينها . فيتم تصميم قاعدة البيانات بناء على جداول مقتضبة بينها علاقات . و هنالك عدة أنواع من العلاقة :

- واحد إلى واحد One-To-One
- كثير إلى واحد Many-To-One
- كثير إلى كثير Many-To-Many

سنتعلم ان شاء الله كيف نستخدمها من خلال ربط المودل في مشروعنا من خلال الـ Django ORM.

## واحد لـ واحد One-To-One

علاقة الواحد إلى واحد في قاعدة البيانات تعني أن كل سجل (مُدخل، كائن) في المودل A تربطه علاقة واحدة فقط مع سجل واحد في المودل B . و كل سجل في المودل B تربطه علاقة واحدة فقط مع سجل في المودل A .

مثال : موديل شخص ، و موديل هوية وطنية . فيصبح كل شخص لديه فقط هوية وطنية واحدة . و كل هوية وطنية لديها شخص واحد فقط .

للتعبير عن علاقة One-To-One في الـ Django ORM نستخدم حقل الـ OneToOneField في المودل . يتم استخدامه في المودل التابع ، في مثالنا السابق سنستخدم هذا الحقل داخل موديل الهوية الوطنية لأنها تابعة للشخص حسب العلاقة (مع العلم بالإمكان عكسها).

## كثير إلى واحد Many-To-One

علاقة الكثير الى واحد في قاعدة البيانات تعني أن كل سجل (مدخل، كائن) في المودل A قد يكون له علاقة مع 0 سجل ، سجل واحد أو عدة سجلات في المودل B . بينما السجل في المودل B له علاقة مع حقل واحد فقط في المودل A .

مثال : موديل زبون و موديل طلب . فكل زبون قد يكون ليس لديه أي طلب ، او لديه طلب واحد أو عدة طلبات . لكن كل طلب لديه فقط زبون واحد .

للتعبير عن علاقة الـ Many-To-One في الـ Django ORM نستخدم حقل الـ ForeignKey . يتم استخدامه في الموديل التابع . ففي مثالنا السابق سنستخدم الحقل في موديل الطلب لأنه موديل تابع للزبون حسب العلاقة .

## كثير إلى كثير Many-To-Many

علاقة الكثير الى كثير في قاعدة البيانات تعني أن كل سجل (مُدخل، كائن) في الموديل A قد يكون له علاقة مع 0 سجل، سجل واحد أو عدة سجلات في الموديل B . والعكس صحيح ، بحيث أن كل سجل في الموديل B قد يكون له علاقة مع 0 حقل ، واحد أو عدة حقول في الموديل A .

مثال: موديل موظف و موديل مشروع . فكل موظف قد يعمل على مشروع واحد أو عدة مشاريع في نفس الوقت . و ايضا كل مشروع قد يكون يعمل عليه موظف واحد او موظفين عدة.

للتعبير عن علاقة الـ Many-To-Many في الـ Django ORM نستخدم حقل الـ ManyToManyField . ففي مثالنا السابق سنستخدم الحقل في موديل المشروع . بحيث كل مشروع سيكون لديه امكانية أن يحتوي على موظف يعمل عليه ، ولا موظف أو موظفين عدة .

## مرجع علاقات الموديل في Django ORM

لأمثلة و مرجع لكل علاقة في Django ORM ، قم بزيارة الرابط التالي :

<https://docs.djangoproject.com/en/4.0/topics/db/examples/>

## عند الحذف on\_delete و علاقات المودل

عند تعريف علاقات من نوع كثير الى واحد و واحد الى واحد ، يجب توفير المعامل on\_delete و التي تخبر الـ Django ORM ما الذي يجب عمله في حال تم حذف المودل الذي تم استخدامه في علاقة في مودل آخر .

أهم الخيارات :

- خيار CASCADE

هذا الخيار يعني أنه عند حذف المودل الرئيسي (الذي تم استخدامه كعلاقة في مودل آخر) ، تلقائيا سيتم حذف جميع السجلات التي لها علاقة بذلك المودل .

- **PROTECT** خيار

هذا الخيار يعني أنه عند حذف المودل الرئيسي سيتم التحقق اذا ما كانت هنالك سجلات تستخدمه كعلاقة ، اذا كانت هنالك سجلات في قاعدة البيانات لها علاقة معه ، فسيتم ايقاف العملية و منع عملية الحذف مع رفع خطأ من نوع `ProtectError` .

- **SET\_NULL** خيار

يتم تعيين جميع الحقل صاحب العلاقة في جميع السجلات الى `NULL` ، هذا في حال تم حذف المودل الاساسي . يمكن استخدام هذا الخيار فقط في حال تم تعيين المعامل `null=True` .

- **DO\_NOTHING** خيار

هذا الخيار يخبر الـ `Django ORM` بعدم اتخاذ اي اجراء . اي سيتم التعامل مع هذه الحالات من قبل المطور سواء برمجيا او على مستوى قاعدة البيانات .