| **Field Name** | **Description** |
| --- | --- |
| [AutoField](https://www.geeksforgeeks.org/autofield-django-models/) | It is an IntegerField that automatically increments. |
| [BigAutoField](https://www.geeksforgeeks.org/bigautofield-django-models/) | It is a 64-bit integer, much like an AutoField except that it is guaranteed to fit numbers from 1 to 9223372036854775807. |
| [BigIntegerField](https://www.geeksforgeeks.org/bigintegerfield-django-models/) | It is a 64-bit integer, much like an IntegerField except that it is guaranteed to fit numbers from -9223372036854775808 to 9223372036854775807. |
| [BinaryField](https://www.geeksforgeeks.org/binaryfield-django-models/) | A field to store raw binary data. |
| [BooleanField](https://www.geeksforgeeks.org/booleanfield-django-models/) | A true/false field.  The default form widget for this field is a CheckboxInput. |
| [CharField](https://www.geeksforgeeks.org/charfield-django-models/) | A field to store text-based values. |
| [DateField](https://www.geeksforgeeks.org/datefield-django-models/) | A date, represented in Python by a datetime.date instance |
| [DateTimeField](https://www.geeksforgeeks.org/datetimefield-django-models/) | It is used for date and time, represented in Python by a datetime.datetime instance. |
| [DecimalField](https://www.geeksforgeeks.org/decimalfield-django-models/) | It is a fixed-precision decimal number, represented in Python by a Decimal instance. |
| [DurationField](https://www.geeksforgeeks.org/durationfield-django-models/) | A field for storing periods of time. |
| [EmailField](https://www.geeksforgeeks.org/emailfield-django-models/) | It is a CharField that checks that the value is a valid email address. |
| [FileField](https://www.geeksforgeeks.org/filefield-django-models/) | It is a file-upload field. |
| [FloatField](https://www.geeksforgeeks.org/floatfield-django-models/) | It is a floating-point number represented in Python by a float instance. |
| [ImageField](https://www.geeksforgeeks.org/imagefield-django-models/) | It inherits all attributes and methods from FileField, but also validates that the uploaded object is a valid image. |
| [IntegerField](https://www.geeksforgeeks.org/integerfield-django-models/) | It is an integer field. Values from -2147483648 to 2147483647 are safe in all databases supported by Django. |
| [GenericIPAddressField](https://www.geeksforgeeks.org/genericipaddressfield-django-models/) | An IPv4 or IPv6 address, in string format (e.g. 192.0.2.30 or 2a02:42fe::4). |
| [NullBooleanField](https://www.geeksforgeeks.org/nullbooleanfield-django-forms/) | Like a BooleanField, but allows NULL as one of the options. |
| [PositiveIntegerField](https://www.geeksforgeeks.org/positiveintegerfield-django-models/) | Like an IntegerField, but must be either positive or zero (0). |
| [PositiveSmallIntegerField](https://www.geeksforgeeks.org/positivesmallintegerfield-django-models/) | Like a PositiveIntegerField, but only allows values under a certain (database-dependent) point. |
| [SlugField](https://www.geeksforgeeks.org/slugfield-django-models/) | Slug is a newspaper term. A slug is a short label for something, containing only letters, numbers, underscores or hyphens. They’re generally used in URLs. |
| [SmallIntegerField](https://www.geeksforgeeks.org/smallintegerfield-django-models/) | It is like an IntegerField, but only allows values under a certain (database-dependent) point. |
| [TextField](https://www.geeksforgeeks.org/textfield-django-models/) | A large text field. The default form widget for this field is a Textarea. |
| [TimeField](https://www.geeksforgeeks.org/timefield-django-models/) | A time, represented in Python by a datetime.time instance. |
| [URLField](https://www.geeksforgeeks.org/urlfield-django-models/) | A CharField for a URL, validated by URLValidator. |
| [UUIDField](https://www.geeksforgeeks.org/uuidfield-django-models/) | A field for storing universally unique identifiers. Uses Python’s UUID class. When used on PostgreSQL, this stores in a uuid datatype, otherwise in a char(32). |

### Relationship Fields

Django also defines a set of fields that represent relations.

| **Field Name** | **Description** |
| --- | --- |
| [ForeignKey](https://www.geeksforgeeks.org/python-relational-fields-in-django-models/) | A many-to-one relationship. Requires two positional arguments: the class to which the model is related and the on\_delete option. |
| [ManyToManyField](https://www.geeksforgeeks.org/python-relational-fields-in-django-models/) | A many-to-many relationship. Requires a positional argument: the class to which the model is related, which works exactly the same as it does for ForeignKey, including recursive and lazy relationships. |
| [OneToOneField](https://www.geeksforgeeks.org/python-relational-fields-in-django-models/) | A one-to-one relationship. Conceptually, this is similar to a ForeignKey with unique=True, but the “reverse” side of the relation will directly return a single object. |