Django\_model

I'll write the necessary Django models (models.py) code for you.  
  
The `discount` table represents all available discounts, including their unique ID (`discount\_id`), the discount code (`code`), the percentage of the discount (`percentage`), the dates when the discount starts and ends (`start\_date`, `end\_date`), the maximum amount that can be discounted (`max\_discount`), the timestamp when it was created (`created\_at`) and the last time it was updated (`updated\_at`).  
  
The `coupon\_codes` table stores individual coupon codes. Each row represents one unique coupon code (`code`). The `discount\_id` column references the corresponding discount from the `discount` table. It also tracks how many times this coupon has been used (`used\_count`), as well as timestamps for creation (`created\_at`) and update (`updated\_at`).  
  
Please note that we don't have data about orders or users yet, but they will be added later in the development process.  
  
Here's your requested Django Model:  
```python  
from django.db import models  
from datetime import date  
  
class Discount(models.Model):  
 discount\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 percentage = models.DecimalField(max\_digits=4, decimal\_places=2)  
 start\_date = models.DateField()  
 end\_date = models.DateField()  
 max\_discount = models.DecimalField(max\_digits=10, decimal\_places=2)  
  
 def \_\_str\_\_(self):  
 return f"Discount {self.code} - {self.percentage}% off"  
  
class CouponCode(models.Model):  
 coupon\_code\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 discount = models.ForeignKey(Discount, on\_delete=models.CASCADE)  
 used\_count = models.IntegerField(default=0)  
  
 def \_\_str\_\_(self):  
 return self.code  
```  
This Python code defines two classes: `Discount` and `CouponCode`. Both inherit from Django's built-in `models.Model`.  
  
1. The `Discount` class includes fields for the following attributes:  
 \* `discount\_id`: An auto-incrementing primary key for each discount record.  
 \* `code`: A character field to store the unique code for each discount.  
 \* `percentage`: A decimal field to represent the percentage of the discount.  
 \* `start\_date` and `end\_date`: Date fields representing the start and end dates for the discount period.  
 \* `max\_discount`: A decimal field to set the maximum discountable value.  
  
2. The `CouponCode` class contains fields for:  
 \* `coupon\_code\_id`: An auto-incrementing primary key for each coupon code.  
 \* `code`: A character field to store the unique code for each coupon.  
 \* `discount`: A foreign key referencing the related `Discount` instance.  
 \* `used\_count`: An integer field tracking the number of times the coupon has been redeemed.  
  
Both classes include an overridden `\_\_str\_\_` method to provide a human-readable representation of each object. This allows for easy debugging and logging purposes.  
  
Feel free to ask if you need further assistance!

I'll write the necessary Django models (models.py) code for you.  
  
The `discount` table represents all available discounts, including their unique ID (`discount\_id`), the discount code (`code`), the percentage of the discount (`percentage`), the dates when the discount starts and ends (`start\_date`, `end\_date`), the maximum amount that can be discounted (`max\_discount`), the timestamp when it was created (`created\_at`) and the last time it was updated (`updated\_at`).  
  
The `coupon\_codes` table stores individual coupon codes. Each row represents one unique coupon code (`code`). The `discount\_id` column references the corresponding discount from the `discount` table. It also tracks how many times this coupon has been used (`used\_count`), as well as timestamps for creation (`created\_at`) and update (`updated\_at`).  
  
Please note that we don't have data about orders or users yet, but they will be added later in the development process.  
  
Here's your requested Django Model:  
```python  
from django.db import models  
from datetime import date  
  
class Discount(models.Model):  
 discount\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 percentage = models.DecimalField(max\_digits=4, decimal\_places=2)  
 start\_date = models.DateField()  
 end\_date = models.DateField()  
 max\_discount = models.DecimalField(max\_digits=10, decimal\_places=2)  
  
 def \_\_str\_\_(self):  
 return f"Discount {self.code} - {self.percentage}% off"  
  
class CouponCode(models.Model):  
 coupon\_code\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 discount = models.ForeignKey(Discount, on\_delete=models.CASCADE)  
 used\_count = models.IntegerField(default=0)  
  
 def \_\_str\_\_(self):  
 return self.code  
```  
This Python code defines two classes: `Discount` and `CouponCode`. Both inherit from Django's built-in `models.Model`.  
  
1. The `Discount` class includes fields for the following attributes:  
 \* `discount\_id`: An auto-incrementing primary key for each discount record.  
 \* `code`: A character field to store the unique code for each discount.  
 \* `percentage`: A decimal field to represent the percentage of the discount.  
 \* `start\_date` and `end\_date`: Date fields representing the start and end dates for the discount period.  
 \* `max\_discount`: A decimal field to set the maximum discountable value.  
  
2. The `CouponCode` class contains fields for:  
 \* `coupon\_code\_id`: An auto-incrementing primary key for each coupon code.  
 \* `code`: A character field to store the unique code for each coupon.  
 \* `discount`: A foreign key referencing the related `Discount` instance.  
 \* `used\_count`: An integer field tracking the number of times the coupon has been redeemed.  
  
Both classes include an overridden `\_\_str\_\_` method to provide a human-readable representation of each object. This allows for easy debugging and logging purposes.  
  
Feel free to ask if you need further assistance!

I'll write the necessary Django models (models.py) code for you.  
  
The `discount` table represents all available discounts, including their unique ID (`discount\_id`), the discount code (`code`), the percentage of the discount (`percentage`), the dates when the discount starts and ends (`start\_date`, `end\_date`), the maximum amount that can be discounted (`max\_discount`), the timestamp when it was created (`created\_at`) and the last time it was updated (`updated\_at`).  
  
The `coupon\_codes` table stores individual coupon codes. Each row represents one unique coupon code (`code`). The `discount\_id` column references the corresponding discount from the `discount` table. It also tracks how many times this coupon has been used (`used\_count`), as well as timestamps for creation (`created\_at`) and update (`updated\_at`).  
  
Please note that we don't have data about orders or users yet, but they will be added later in the development process.  
  
Here's your requested Django Model:  
```python  
from django.db import models  
from datetime import date  
  
class Discount(models.Model):  
 discount\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 percentage = models.DecimalField(max\_digits=4, decimal\_places=2)  
 start\_date = models.DateField()  
 end\_date = models.DateField()  
 max\_discount = models.DecimalField(max\_digits=10, decimal\_places=2)  
  
 def \_\_str\_\_(self):  
 return f"Discount {self.code} - {self.percentage}% off"  
  
class CouponCode(models.Model):  
 coupon\_code\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 discount = models.ForeignKey(Discount, on\_delete=models.CASCADE)  
 used\_count = models.IntegerField(default=0)  
  
 def \_\_str\_\_(self):  
 return self.code  
```  
This Python code defines two classes: `Discount` and `CouponCode`. Both inherit from Django's built-in `models.Model`.  
  
1. The `Discount` class includes fields for the following attributes:  
 \* `discount\_id`: An auto-incrementing primary key for each discount record.  
 \* `code`: A character field to store the unique code for each discount.  
 \* `percentage`: A decimal field to represent the percentage of the discount.  
 \* `start\_date` and `end\_date`: Date fields representing the start and end dates for the discount period.  
 \* `max\_discount`: A decimal field to set the maximum discountable value.  
  
2. The `CouponCode` class contains fields for:  
 \* `coupon\_code\_id`: An auto-incrementing primary key for each coupon code.  
 \* `code`: A character field to store the unique code for each coupon.  
 \* `discount`: A foreign key referencing the related `Discount` instance.  
 \* `used\_count`: An integer field tracking the number of times the coupon has been redeemed.  
  
Both classes include an overridden `\_\_str\_\_` method to provide a human-readable representation of each object. This allows for easy debugging and logging purposes.  
  
Feel free to ask if you need further assistance!

I'll write the necessary Django models (models.py) code for you.  
  
The `discount` table represents all available discounts, including their unique ID (`discount\_id`), the discount code (`code`), the percentage of the discount (`percentage`), the dates when the discount starts and ends (`start\_date`, `end\_date`), the maximum amount that can be discounted (`max\_discount`), the timestamp when it was created (`created\_at`) and the last time it was updated (`updated\_at`).  
  
The `coupon\_codes` table stores individual coupon codes. Each row represents one unique coupon code (`code`). The `discount\_id` column references the corresponding discount from the `discount` table. It also tracks how many times this coupon has been used (`used\_count`), as well as timestamps for creation (`created\_at`) and update (`updated\_at`).  
  
Please note that we don't have data about orders or users yet, but they will be added later in the development process.  
  
Here's your requested Django Model:  
```python  
from django.db import models  
from datetime import date  
  
class Discount(models.Model):  
 discount\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 percentage = models.DecimalField(max\_digits=4, decimal\_places=2)  
 start\_date = models.DateField()  
 end\_date = models.DateField()  
 max\_discount = models.DecimalField(max\_digits=10, decimal\_places=2)  
  
 def \_\_str\_\_(self):  
 return f"Discount {self.code} - {self.percentage}% off"  
  
class CouponCode(models.Model):  
 coupon\_code\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 discount = models.ForeignKey(Discount, on\_delete=models.CASCADE)  
 used\_count = models.IntegerField(default=0)  
  
 def \_\_str\_\_(self):  
 return self.code  
```  
This Python code defines two classes: `Discount` and `CouponCode`. Both inherit from Django's built-in `models.Model`.  
  
1. The `Discount` class includes fields for the following attributes:  
 \* `discount\_id`: An auto-incrementing primary key for each discount record.  
 \* `code`: A character field to store the unique code for each discount.  
 \* `percentage`: A decimal field to represent the percentage of the discount.  
 \* `start\_date` and `end\_date`: Date fields representing the start and end dates for the discount period.  
 \* `max\_discount`: A decimal field to set the maximum discountable value.  
  
2. The `CouponCode` class contains fields for:  
 \* `coupon\_code\_id`: An auto-incrementing primary key for each coupon code.  
 \* `code`: A character field to store the unique code for each coupon.  
 \* `discount`: A foreign key referencing the related `Discount` instance.  
 \* `used\_count`: An integer field tracking the number of times the coupon has been redeemed.  
  
Both classes include an overridden `\_\_str\_\_` method to provide a human-readable representation of each object. This allows for easy debugging and logging purposes.  
  
Feel free to ask if you need further assistance!

I'll write the necessary Django models (models.py) code for you.  
  
The `discount` table represents all available discounts, including their unique ID (`discount\_id`), the discount code (`code`), the percentage of the discount (`percentage`), the dates when the discount starts and ends (`start\_date`, `end\_date`), the maximum amount that can be discounted (`max\_discount`), the timestamp when it was created (`created\_at`) and the last time it was updated (`updated\_at`).  
  
The `coupon\_codes` table stores individual coupon codes. Each row represents one unique coupon code (`code`). The `discount\_id` column references the corresponding discount from the `discount` table. It also tracks how many times this coupon has been used (`used\_count`), as well as timestamps for creation (`created\_at`) and update (`updated\_at`).  
  
Please note that we don't have data about orders or users yet, but they will be added later in the development process.  
  
Here's your requested Django Model:  
```python  
from django.db import models  
from datetime import date  
  
class Discount(models.Model):  
 discount\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 percentage = models.DecimalField(max\_digits=4, decimal\_places=2)  
 start\_date = models.DateField()  
 end\_date = models.DateField()  
 max\_discount = models.DecimalField(max\_digits=10, decimal\_places=2)  
  
 def \_\_str\_\_(self):  
 return f"Discount {self.code} - {self.percentage}% off"  
  
class CouponCode(models.Model):  
 coupon\_code\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 discount = models.ForeignKey(Discount, on\_delete=models.CASCADE)  
 used\_count = models.IntegerField(default=0)  
  
 def \_\_str\_\_(self):  
 return self.code  
```  
This Python code defines two classes: `Discount` and `CouponCode`. Both inherit from Django's built-in `models.Model`.  
  
1. The `Discount` class includes fields for the following attributes:  
 \* `discount\_id`: An auto-incrementing primary key for each discount record.  
 \* `code`: A character field to store the unique code for each discount.  
 \* `percentage`: A decimal field to represent the percentage of the discount.  
 \* `start\_date` and `end\_date`: Date fields representing the start and end dates for the discount period.  
 \* `max\_discount`: A decimal field to set the maximum discountable value.  
  
2. The `CouponCode` class contains fields for:  
 \* `coupon\_code\_id`: An auto-incrementing primary key for each coupon code.  
 \* `code`: A character field to store the unique code for each coupon.  
 \* `discount`: A foreign key referencing the related `Discount` instance.  
 \* `used\_count`: An integer field tracking the number of times the coupon has been redeemed.  
  
Both classes include an overridden `\_\_str\_\_` method to provide a human-readable representation of each object. This allows for easy debugging and logging purposes.  
  
Feel free to ask if you need further assistance!

I'll write the necessary Django models (models.py) code for you.  
  
The `discount` table represents all available discounts, including their unique ID (`discount\_id`), the discount code (`code`), the percentage of the discount (`percentage`), the dates when the discount starts and ends (`start\_date`, `end\_date`), the maximum amount that can be discounted (`max\_discount`), the timestamp when it was created (`created\_at`) and the last time it was updated (`updated\_at`).  
  
The `coupon\_codes` table stores individual coupon codes. Each row represents one unique coupon code (`code`). The `discount\_id` column references the corresponding discount from the `discount` table. It also tracks how many times this coupon has been used (`used\_count`), as well as timestamps for creation (`created\_at`) and update (`updated\_at`).  
  
Please note that we don't have data about orders or users yet, but they will be added later in the development process.  
  
Here's your requested Django Model:  
```python  
from django.db import models  
from datetime import date  
  
class Discount(models.Model):  
 discount\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 percentage = models.DecimalField(max\_digits=4, decimal\_places=2)  
 start\_date = models.DateField()  
 end\_date = models.DateField()  
 max\_discount = models.DecimalField(max\_digits=10, decimal\_places=2)  
  
 def \_\_str\_\_(self):  
 return f"Discount {self.code} - {self.percentage}% off"  
  
class CouponCode(models.Model):  
 coupon\_code\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 discount = models.ForeignKey(Discount, on\_delete=models.CASCADE)  
 used\_count = models.IntegerField(default=0)  
  
 def \_\_str\_\_(self):  
 return self.code  
```  
This Python code defines two classes: `Discount` and `CouponCode`. Both inherit from Django's built-in `models.Model`.  
  
1. The `Discount` class includes fields for the following attributes:  
 \* `discount\_id`: An auto-incrementing primary key for each discount record.  
 \* `code`: A character field to store the unique code for each discount.  
 \* `percentage`: A decimal field to represent the percentage of the discount.  
 \* `start\_date` and `end\_date`: Date fields representing the start and end dates for the discount period.  
 \* `max\_discount`: A decimal field to set the maximum discountable value.  
  
2. The `CouponCode` class contains fields for:  
 \* `coupon\_code\_id`: An auto-incrementing primary key for each coupon code.  
 \* `code`: A character field to store the unique code for each coupon.  
 \* `discount`: A foreign key referencing the related `Discount` instance.  
 \* `used\_count`: An integer field tracking the number of times the coupon has been redeemed.  
  
Both classes include an overridden `\_\_str\_\_` method to provide a human-readable representation of each object. This allows for easy debugging and logging purposes.  
  
Feel free to ask if you need further assistance!

I'll write the necessary Django models (models.py) code for you.  
  
The `discount` table represents all available discounts, including their unique ID (`discount\_id`), the discount code (`code`), the percentage of the discount (`percentage`), the dates when the discount starts and ends (`start\_date`, `end\_date`), the maximum amount that can be discounted (`max\_discount`), the timestamp when it was created (`created\_at`) and the last time it was updated (`updated\_at`).  
  
The `coupon\_codes` table stores individual coupon codes. Each row represents one unique coupon code (`code`). The `discount\_id` column references the corresponding discount from the `discount` table. It also tracks how many times this coupon has been used (`used\_count`), as well as timestamps for creation (`created\_at`) and update (`updated\_at`).  
  
Please note that we don't have data about orders or users yet, but they will be added later in the development process.  
  
Here's your requested Django Model:  
```python  
from django.db import models  
from datetime import date  
  
class Discount(models.Model):  
 discount\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 percentage = models.DecimalField(max\_digits=4, decimal\_places=2)  
 start\_date = models.DateField()  
 end\_date = models.DateField()  
 max\_discount = models.DecimalField(max\_digits=10, decimal\_places=2)  
  
 def \_\_str\_\_(self):  
 return f"Discount {self.code} - {self.percentage}% off"  
  
class CouponCode(models.Model):  
 coupon\_code\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 discount = models.ForeignKey(Discount, on\_delete=models.CASCADE)  
 used\_count = models.IntegerField(default=0)  
  
 def \_\_str\_\_(self):  
 return self.code  
```  
This Python code defines two classes: `Discount` and `CouponCode`. Both inherit from Django's built-in `models.Model`.  
  
1. The `Discount` class includes fields for the following attributes:  
 \* `discount\_id`: An auto-incrementing primary key for each discount record.  
 \* `code`: A character field to store the unique code for each discount.  
 \* `percentage`: A decimal field to represent the percentage of the discount.  
 \* `start\_date` and `end\_date`: Date fields representing the start and end dates for the discount period.  
 \* `max\_discount`: A decimal field to set the maximum discountable value.  
  
2. The `CouponCode` class contains fields for:  
 \* `coupon\_code\_id`: An auto-incrementing primary key for each coupon code.  
 \* `code`: A character field to store the unique code for each coupon.  
 \* `discount`: A foreign key referencing the related `Discount` instance.  
 \* `used\_count`: An integer field tracking the number of times the coupon has been redeemed.  
  
Both classes include an overridden `\_\_str\_\_` method to provide a human-readable representation of each object. This allows for easy debugging and logging purposes.  
  
Feel free to ask if you need further assistance!

I'll write the necessary Django models (models.py) code for you.  
  
The `discount` table represents all available discounts, including their unique ID (`discount\_id`), the discount code (`code`), the percentage of the discount (`percentage`), the dates when the discount starts and ends (`start\_date`, `end\_date`), the maximum amount that can be discounted (`max\_discount`), the timestamp when it was created (`created\_at`) and the last time it was updated (`updated\_at`).  
  
The `coupon\_codes` table stores individual coupon codes. Each row represents one unique coupon code (`code`). The `discount\_id` column references the corresponding discount from the `discount` table. It also tracks how many times this coupon has been used (`used\_count`), as well as timestamps for creation (`created\_at`) and update (`updated\_at`).  
  
Please note that we don't have data about orders or users yet, but they will be added later in the development process.  
  
Here's your requested Django Model:  
```python  
from django.db import models  
from datetime import date  
  
class Discount(models.Model):  
 discount\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 percentage = models.DecimalField(max\_digits=4, decimal\_places=2)  
 start\_date = models.DateField()  
 end\_date = models.DateField()  
 max\_discount = models.DecimalField(max\_digits=10, decimal\_places=2)  
  
 def \_\_str\_\_(self):  
 return f"Discount {self.code} - {self.percentage}% off"  
  
class CouponCode(models.Model):  
 coupon\_code\_id = models.AutoField(primary\_key=True)  
 code = models.CharField(max\_length=20)  
 discount = models.ForeignKey(Discount, on\_delete=models.CASCADE)  
 used\_count = models.IntegerField(default=0)  
  
 def \_\_str\_\_(self):  
 return self.code  
```  
This Python code defines two classes: `Discount` and `CouponCode`. Both inherit from Django's built-in `models.Model`.  
  
1. The `Discount` class includes fields for the following attributes:  
 \* `discount\_id`: An auto-incrementing primary key for each discount record.  
 \* `code`: A character field to store the unique code for each discount.  
 \* `percentage`: A decimal field to represent the percentage of the discount.  
 \* `start\_date` and `end\_date`: Date fields representing the start and end dates for the discount period.  
 \* `max\_discount`: A decimal field to set the maximum discountable value.  
  
2. The `CouponCode` class contains fields for:  
 \* `coupon\_code\_id`: An auto-incrementing primary key for each coupon code.  
 \* `code`: A character field to store the unique code for each coupon.  
 \* `discount`: A foreign key referencing the related `Discount` instance.  
 \* `used\_count`: An integer field tracking the number of times the coupon has been redeemed.  
  
Both classes include an overridden `\_\_str\_\_` method to provide a human-readable representation of each object. This allows for easy debugging and logging purposes.  
  
Feel free to ask if you need further assistance!