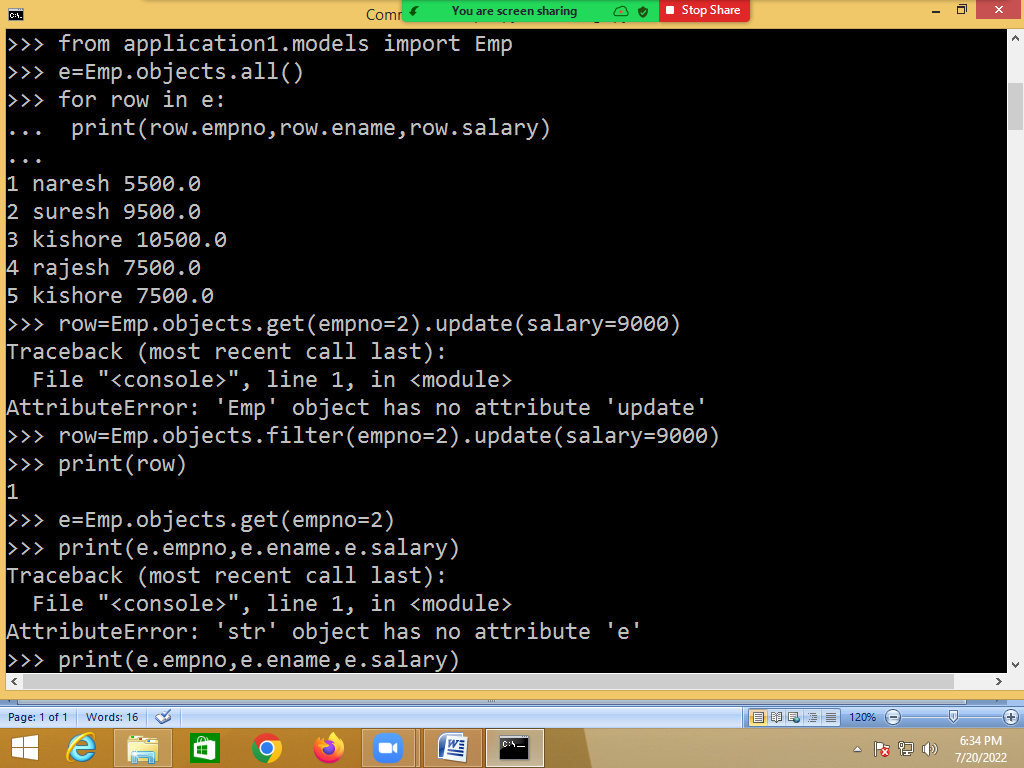
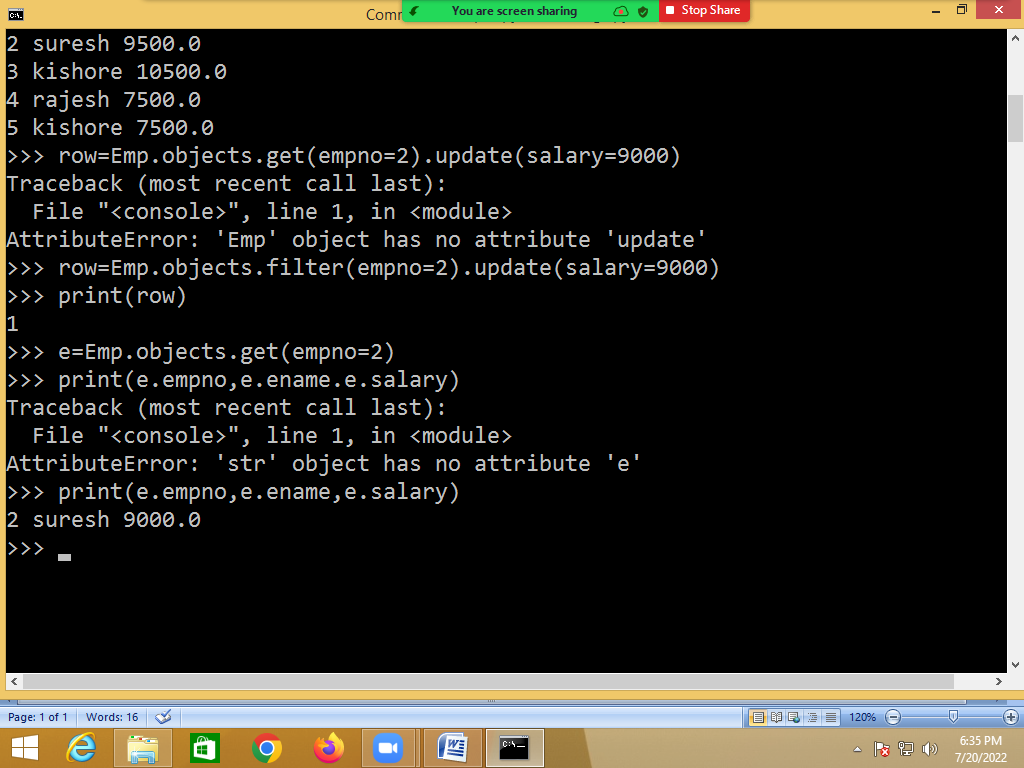
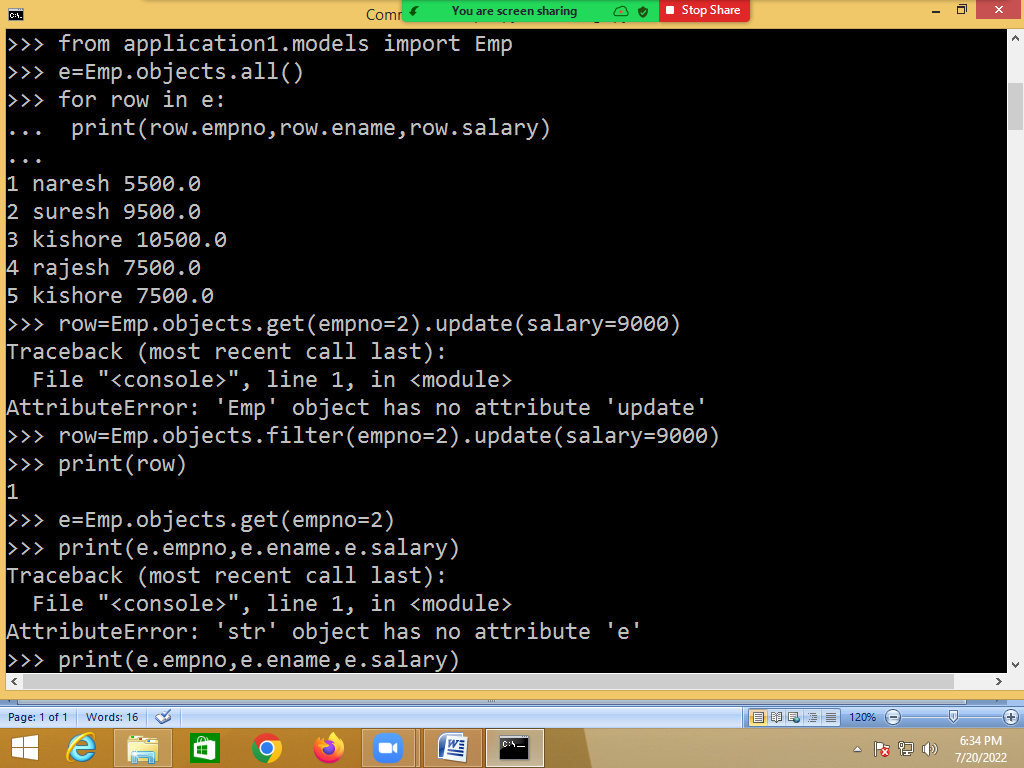
**Using update method of model**

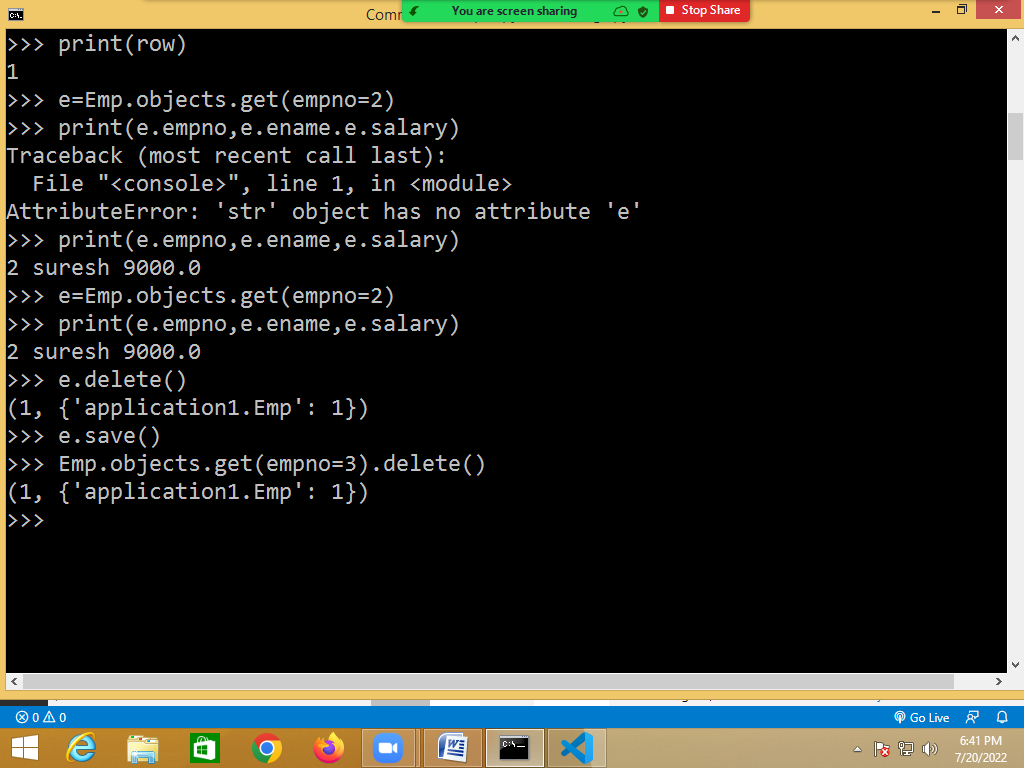
This method is used to update the value a given column.

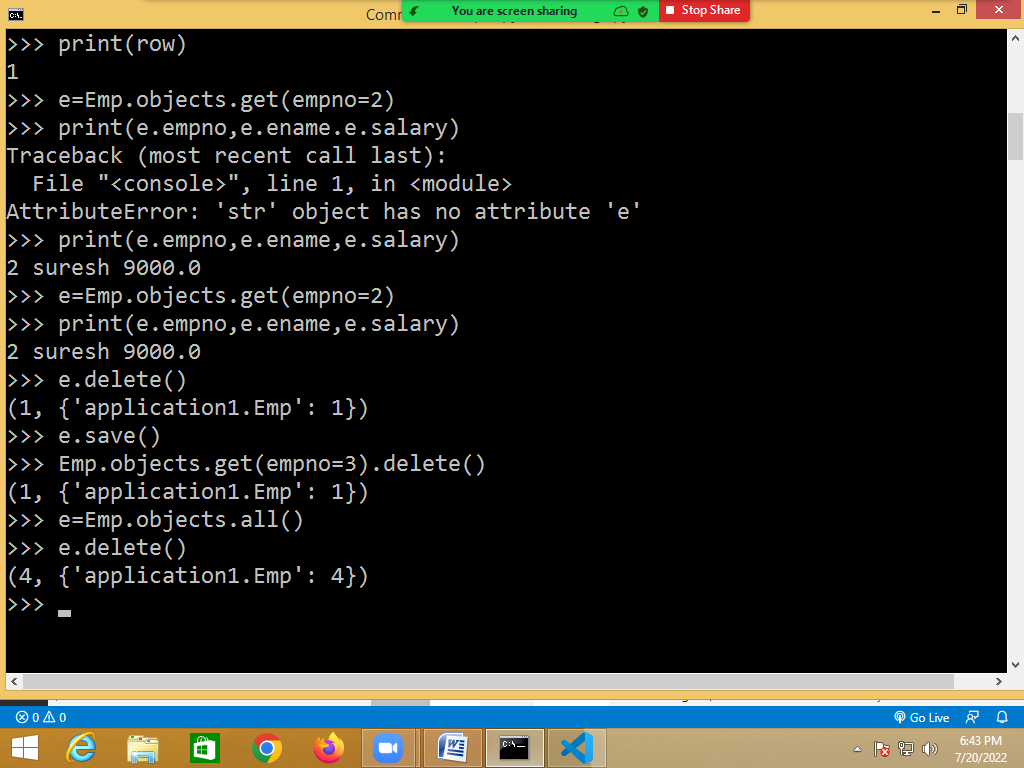




**Deleting row from database table**

Model class provides delete() method, which delete one object or row.





**Relationship between models**

Designing the database is one of the most important parts of building a web application. If you’re using a relational database, maintaining proper relationships between the tables is a must.

Each [model](https://docs.djangoproject.com/en/3.2/topics/db/models/) in a [Django](https://www.djangoproject.com/" \t "_blank) application represents a database table. By default, Django models operate on Relational Database Management System (RDBMS). So you need to design your Django models maintaining proper relationships between them.

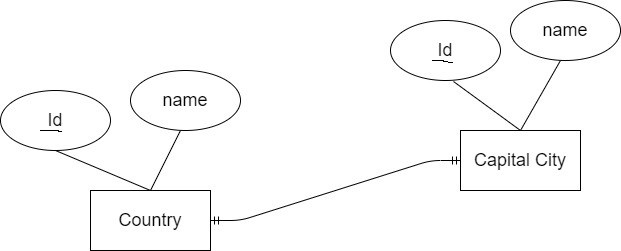
**In a relational database system, there are three types of relationships:**

1. One-to-One relationship
2. One-to-Many relationship (or Many-to-One)
3. Many-to-Many relationship

**One-to-One Relationship**

*“A one-to-one relationship means one record of a table is associated with exactly one record of another table.”*

A real-life example of a one-to-one relationship can be a country and its capital city. Each country has only one capital city. And each capital city is the capital city of only one country.



To establish a one-to-one relationship in Django, the keyword OneToOneField is used.

**models.py**

class Country(models.Model):

name=models.CharField(max\_length=20)

class Capital(models.Model):

name=models.CharField(max\_length=20)

country=models.OneToOneField

(Country,primary\_key=True,on\_delete=models.CASCADE)

