**Create row/record**

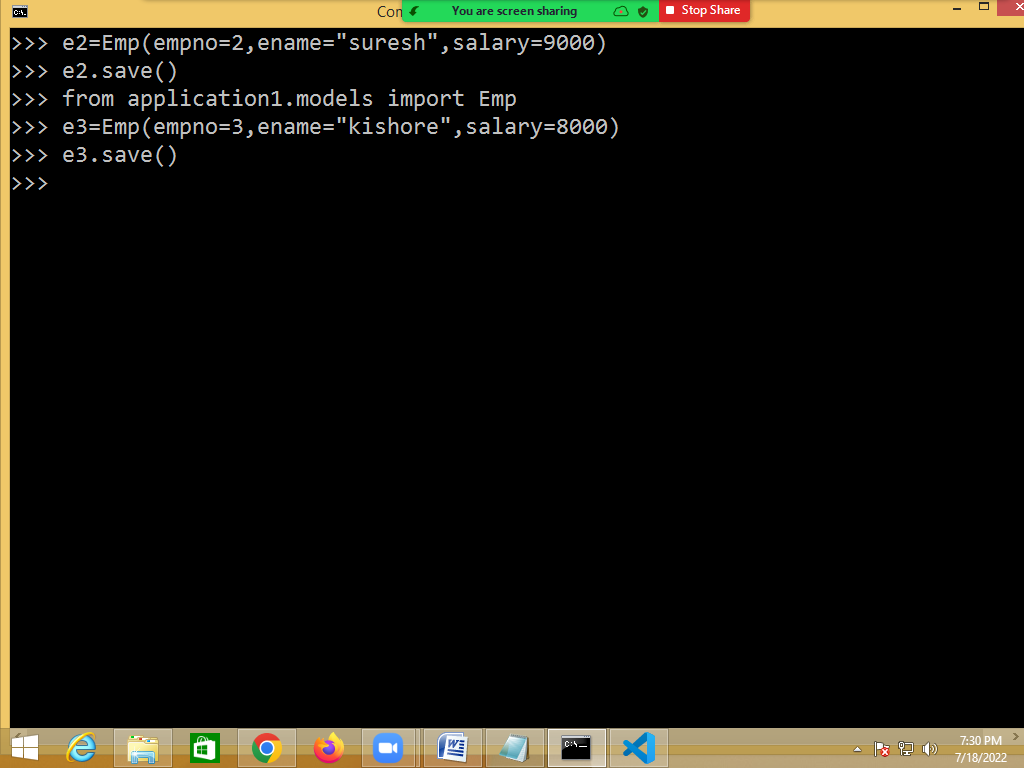
This creation for row or record is done in two ways.

1. Create method
2. Creating model object

Creating row is nothing but creating object of model class.

**Open interactive mode/shell**

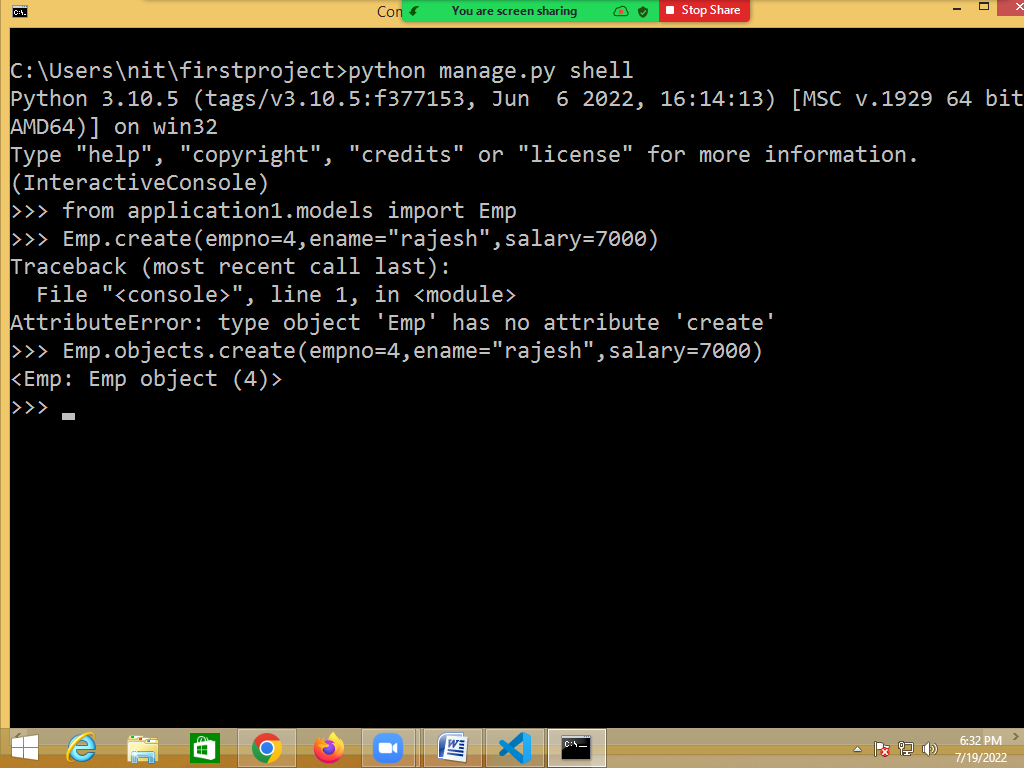
**Python manage.py shell**

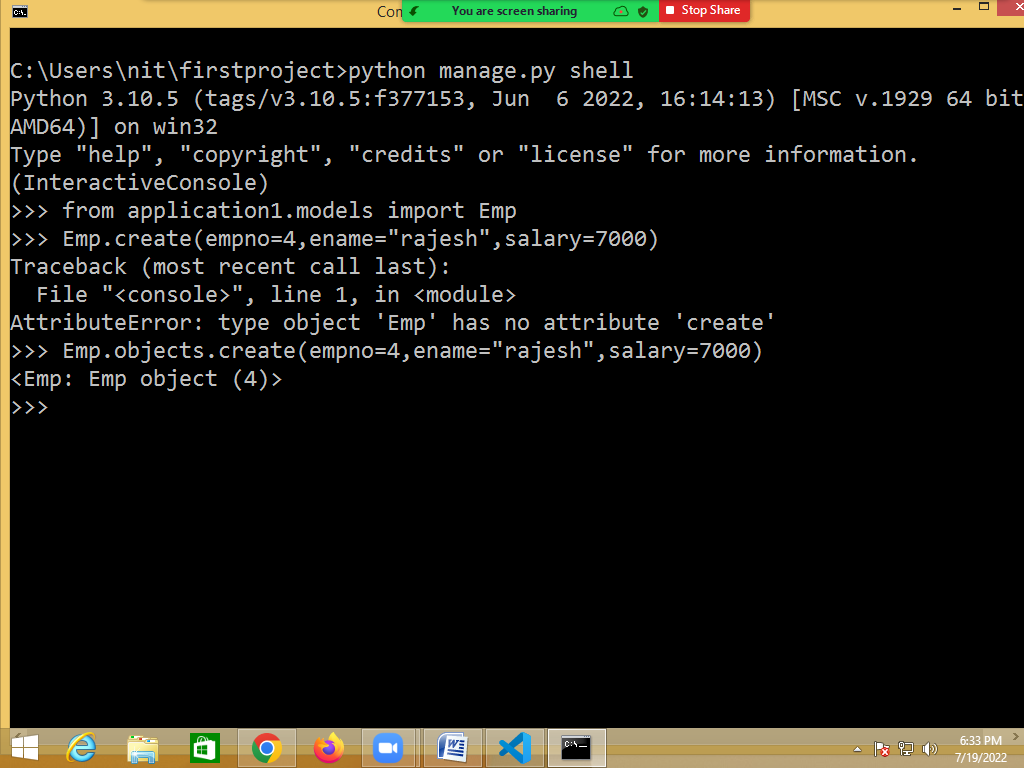
****

**Using create method**

Create method of Model is used for creating new row or object.

**Syntax: model-class-name.objects.create(field-name=value,…)**





Create method of objects create object and save. Not required to call save method explicitly.

**Reading rows from database table/model**

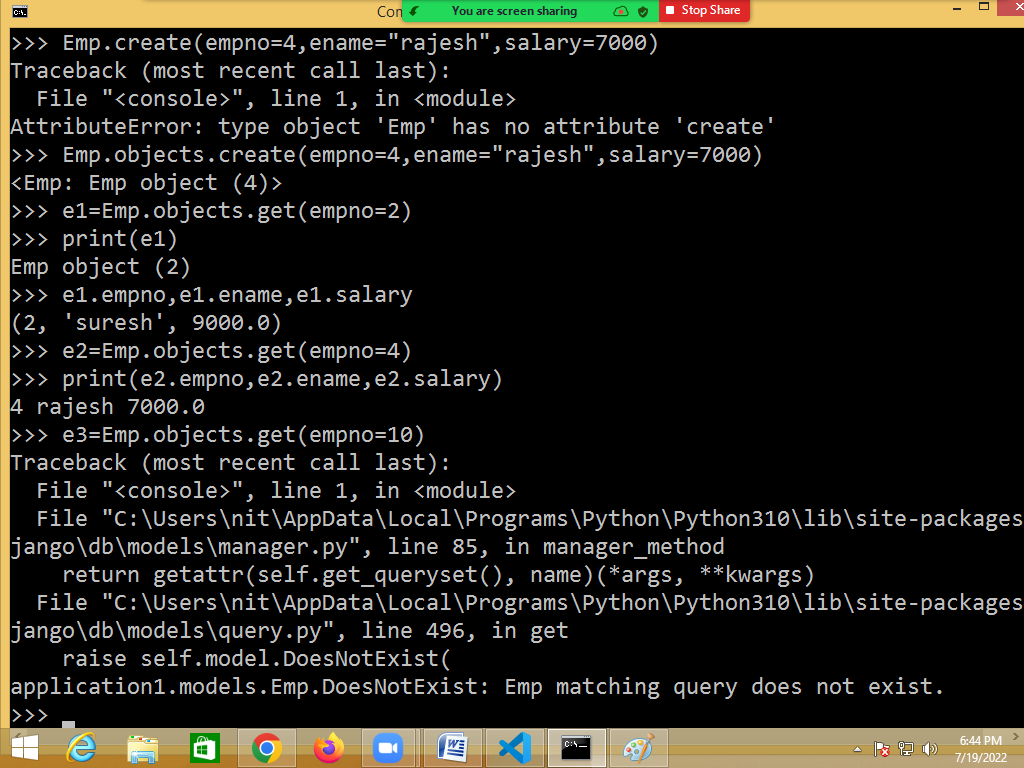
In order to read rows from database model provides various methods.

1. get()
2. filter()
3. all()

**get()**

The get() method in Django **returns a single object that matches the given lookup parameter**. While using the get(), we should always use the lookups which are unique like primary keys or fields in unique constraints.

**Syntax: model-class.objects.get(field-name=value)**



Get method raises DoesNotExist error, if there is no row match with given condition.

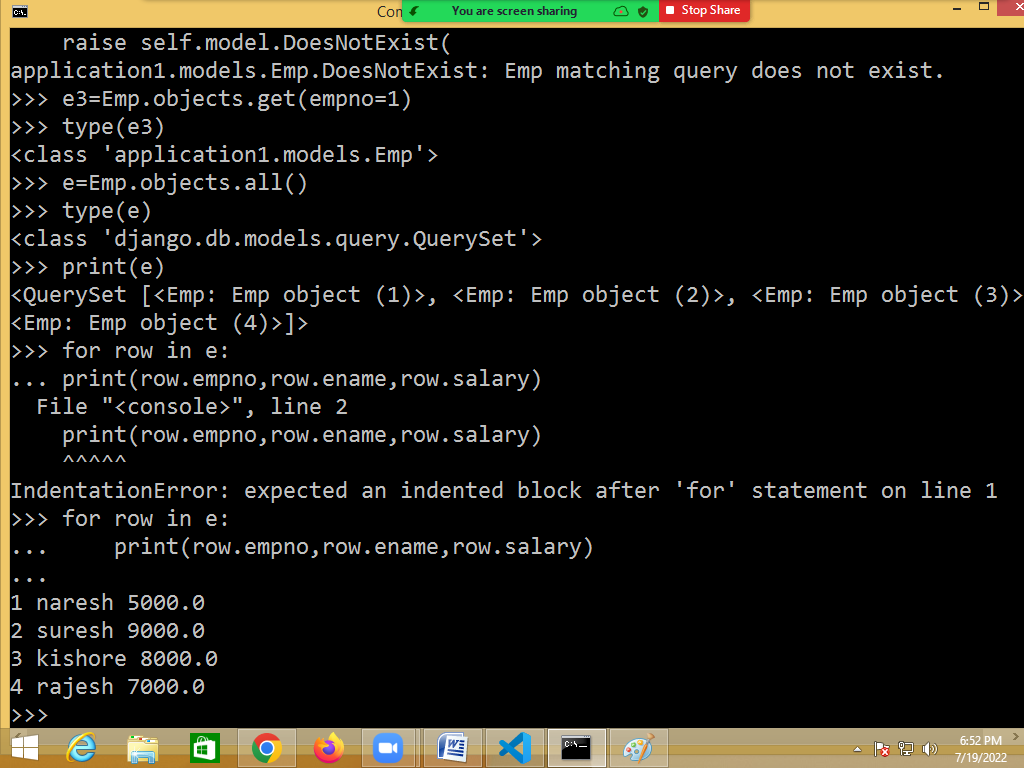
**all()**

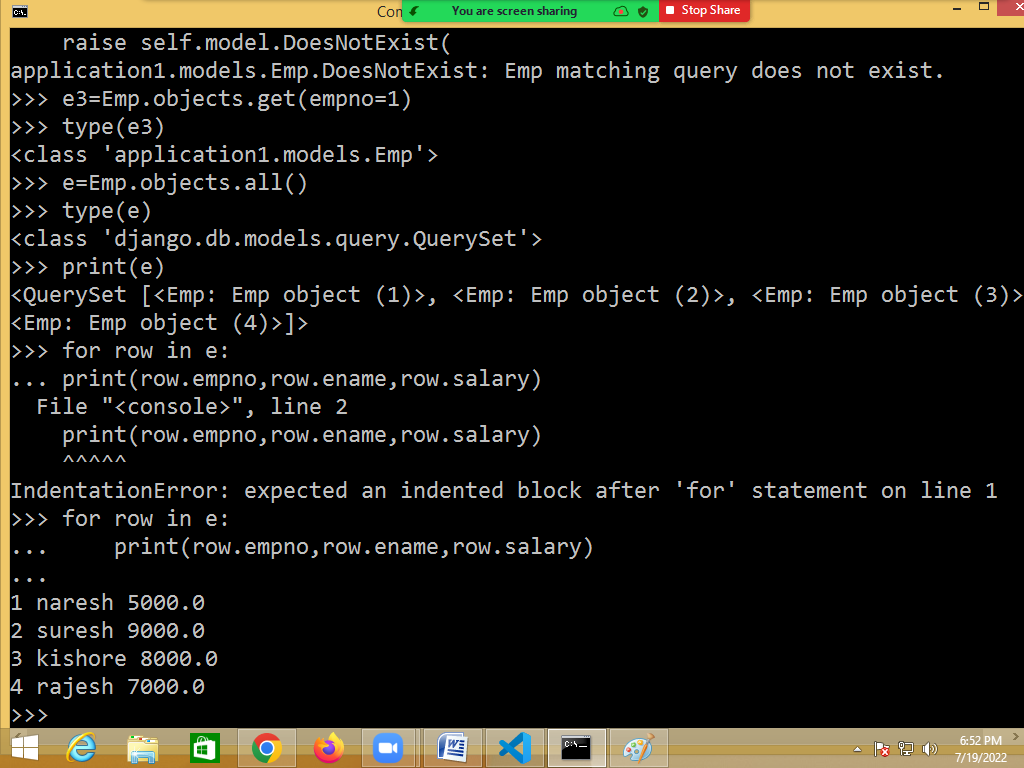
This method read all the rows from database table or it read all the objects from model.

all() method returns QuerySet.

QuerySet object is having result of executed Query. This QuerySet object is having more than one row/object.

all() method is equal to select \* from <table-name>





**filter()**

filter() read selected rows from database table.

Filter is like using where condition with select statement.

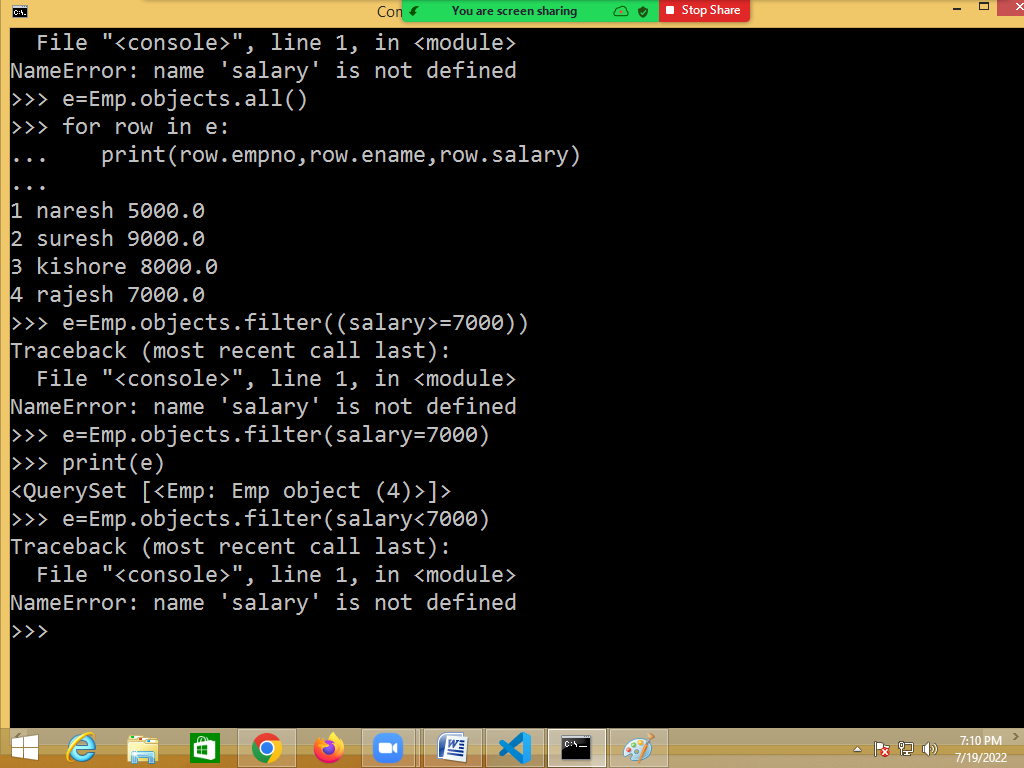
Select \* from <table-name> where condition

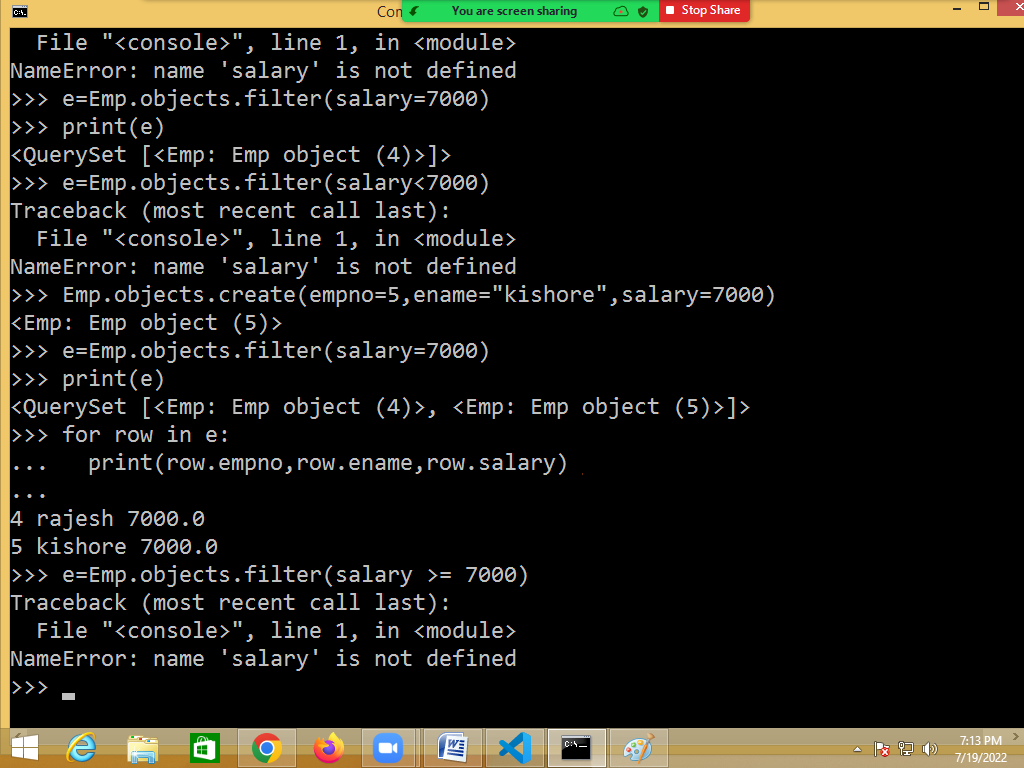
Returns a new **[QuerySet](https://docs.djangoproject.com/en/4.0/ref/models/querysets/" \l "django.db.models.query.QuerySet" \o "django.db.models.query.QuerySet)** containing objects that match the given lookup parameters.

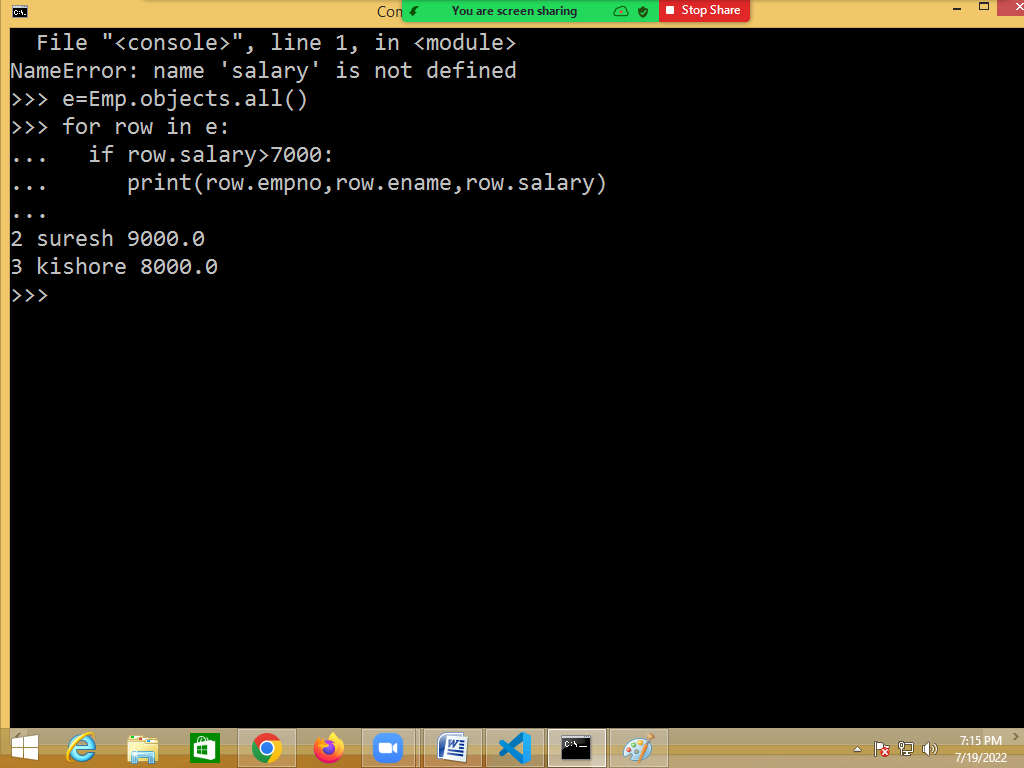
Syntax-1: model-name.objects.filter(condition)

Syntax-2: model-name.objects.all().filter(condition)

filter() method returns QuerySet.







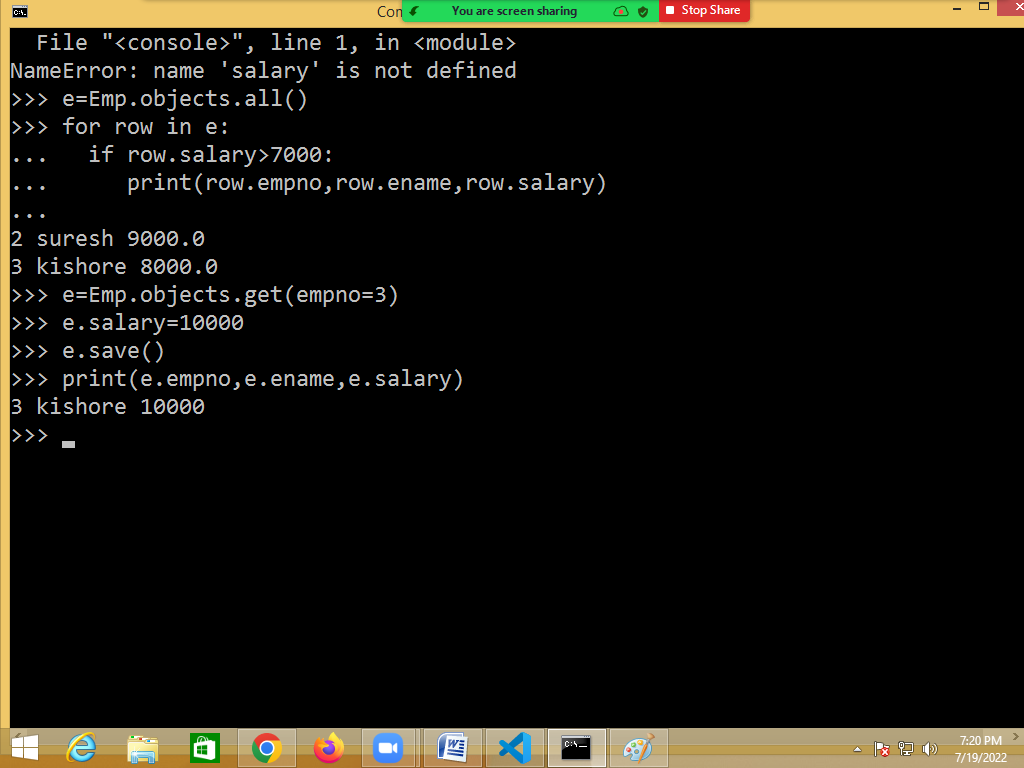
**Update row or replacing values**

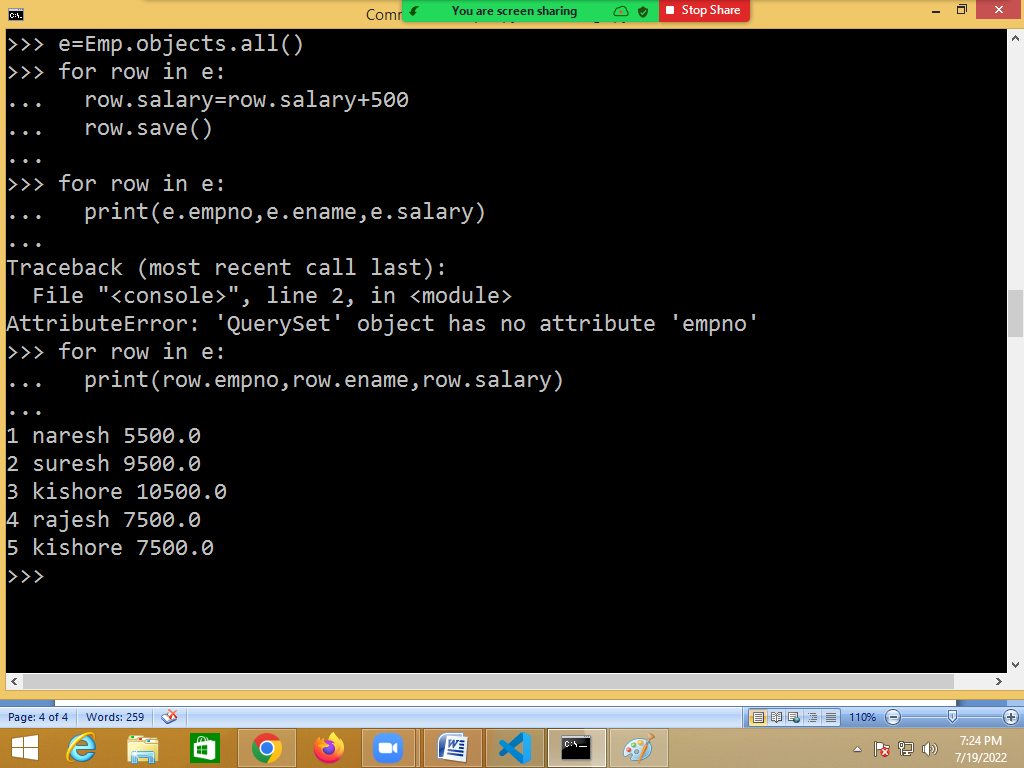
**This approach is used for updating one row**

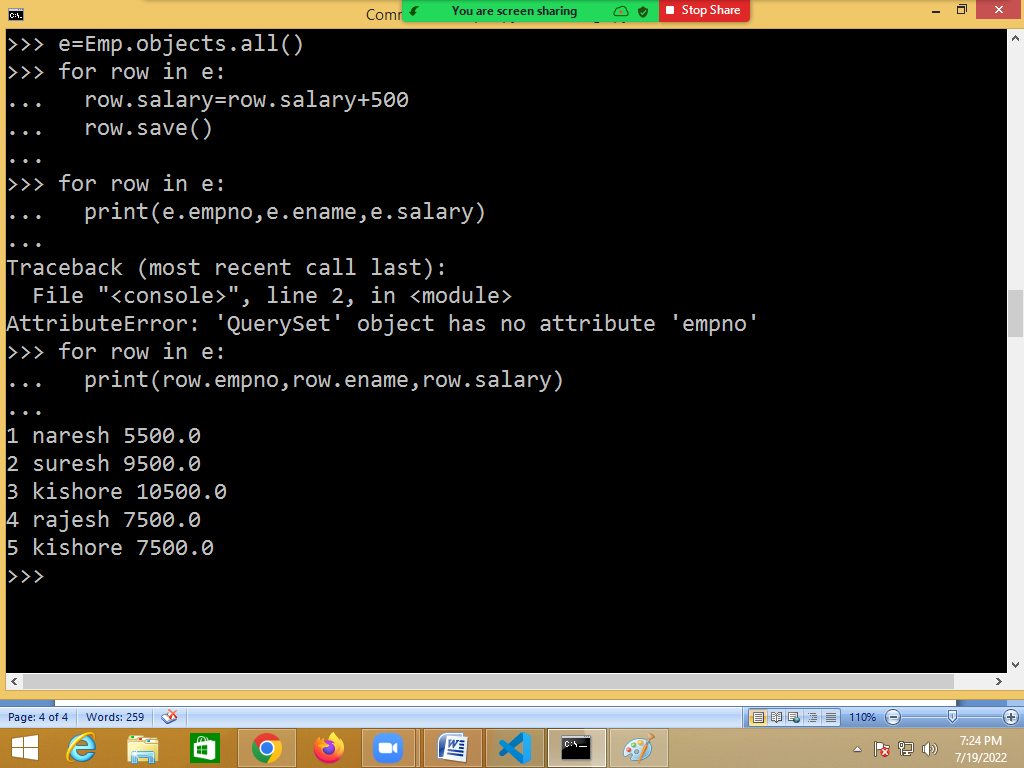
Read the object/row for which update has to be done

<object-name>.<col-name>=<value>

<object-name>.save()







Using update method of model