understanding django models

How to register models against django admin interface?

```
##########################################################
# How to register models to django admin-interface?      #
##########################################################

from django.contrib import admin
from jobs.models import Portal

admin.site.register(Portal)

##########################################################
```

How to perform migrations?

`python manage.py makemigrations`

How to apply migrations?

`python manage.py migrate`

How to check migrations?

`python manage.py check`

How to get interactive shell in django?

```
python manage.py shell

# You can import models defined in your models.py here and perform CRUD ops.
```

Option 1 :: (how to create objects by importing models and save them?)

```
from jobs.models import JobDescription, JobTitle, Portal


from django.utils import timezone
```

How to manipulate data using ORM functions?

```
from jobs.models import JobDescription, JobTitle, Portal

#### To fetch all records available in database against `Portal`
portals = Portal.objects.all()


## How to create object as ORM query?
Portal.objects.create()


## How to add where clause in ORM query?
Portal.objects.get(id=1)
```

```
## How to filter data in ORM query?
Portal.objects.filter(name="naukri.com")
```

Primary key defaults by django.models?

- By default, when primary key is not specified in django model definition `id` field is used by django
- Shortcut to access primary key is `pk`
- `id` is auto-increment field (added by django itself)
-

```
"""
By default, Django gives each model an auto-incrementing primary key with the type specified per app in AppConfig.default_auto_field or glo

    id = models.BigAutoField(primary_key=True)
    If you’d like to specify a custom primary key, specify primary_key=True on one of your fields. If Django sees you’ve explicitly set Fie
"""
```

What different database engines can we use in settings.py?

- mysql :: 'django.db.backends.mysql'
- postgres :: 'django.db.backends.postgresql'
- sqlite3 :: 'django.db.backends.sqlite3'