Trainee Assessment Tools: Web Application Development with Python (Level-4)

Total points 4/20

Time 10min

Email *

lnrs.nayem@gmail.com

✗ You want to retrieve the sum of all ratings for books in the database. Which method would you use? *0/1

- (●) Book.objects.sum('rating') ✗
- ( ) Book.objects.aggregate(Sum('rating'))
- ( ) Book.objects.rating__sum()
- ( ) Book.objects.aggregate(Total('rating'))

Correct answer

- (●) Book.objects.aggregate(Sum('rating'))

✗ You want to retrieve all books whose title contains the word "Python". Which query would you use? *0/1

- (●) Book.objects.filter(title___contains="Python") ✗
- ( ) Book.objects.filter(title__like="Python")
- ( ) Book.objects.filter(title__icontains="Python")
- ( ) Book.objects.filter(title__ilike="Python")

Correct answer

- (●) Book.objects.filter(title__icontains="Python")

✗ In Django, what is the purpose of the values() method in a queryset? * 0/1

- ( ) To return a list of field values for each object in the queryset.
- (●) To return a single value from the queryset. ✗
- ( ) To filter the queryset based on specific values.
- ( ) To sort the queryset in ascending order.

Correct answer

- (●) To return a list of field values for each object in the queryset.

✗ Which method would you use to retrieve the last ten objects from a queryset in Django? *0/1

- ○ last(10)
- ○ tail(10)
- ◉ reverse()[:10] ✗
- ○ order_by('-id')[:10]

Correct answer

- ◉ order_by('-id')[:10]

✓ You want to retrieve the average rating of all books in the database. Which method would you use? *1/1

- ○ Book.objects.average('rating')
- ◉ Book.objects.aggregate(Avg('rating')) ✓
- ○ Book.objects.avg('rating')
- ○ Book.objects.rating__avg()

Feedback

Calculating Aggregated Values: When you need to calculate summary statistics, such as the average, count, sum, minimum, or maximum values of a specific field across a set of objects in a queryset.

average_rating=Book.objects.aggregate() allows you to perform aggregate calculations on the queryset.
Avg('rating') calculates the average rating of all books based on the rating field.
['average_rating'] accesses the result of the aggregation operation, giving you the average rating value.
This query will return the average rating of all books in the database. Make sure to replace 'rating' with the actual field name representing the rating in your Book model. If your model doesn't have a rating field or if you're using a different field name, adjust it accordingly.

✗ You want to retrieve all books that have been published in the years 2018, 2019, or 2020. Which query would you use? *0/1

- (●) Book.objects.filter(published_year=2018, published_year=2019, published_year=2020) ✗
- ( ) Book.objects.filter(published_year__in=[2018, 2019, 2020])
- ( ) Book.objects.filter(published_year=[2018, 2019, 2020])
- ( ) Book.objects.filter(published_year__range=(2018, 2020))

Correct answer

- (●) Book.objects.filter(published_year__in=[2018, 2019, 2020])

✗ You want to retrieve the first ten books from a queryset in Django. Which method would you use? *0/1

- (●) first(10) ✗
- ( ) limit(10)
- ( ) slice(10)
- ( ) [:10]

Correct answer

- (●) [:10]

✗ Which method is used to count the number of objects in a queryset in Django? *0/1

- ○ size()
- ◉ length() ✗
- ○ count()
- ○ total()

Correct answer

- ◉ count()

✗ Which method is used to perform an OR query in Django? * 0/1

- ○ or_filter()
- ○ Q()
- ○ or()
- ◉ filter_or() ✗

Correct answer

- ◉ Q()

✗ Which method would you use to retrieve a random object from a Django model queryset? *0/1

- (●) random() ✗
- ( ) get_random()
- ( ) order_by('?').first()
- ( ) random_object()

Correct answer

- (●) order_by('?').first()

✗ What does the exists() method do in Django? * 0/1

- ( ) Checks if a queryset is empty.
- ( ) Checks if a queryset exists in the database.
- ( ) Checks if a specific object exists in a queryset.
- (●) Checks if a specific field exists in a model. ✗

Correct answer

- (●) Checks if a queryset is empty.

✗ You want to retrieve all books that have either been published by "Publisher A" or "Publisher B". Which query would you use? *0/1

- (●) Book.objects.filter(publisher="Publisher A" | "Publisher B") ✗
- ( ) Book.objects.filter(publisher="Publisher A").filter(publisher="Publisher B")
- ( ) Book.objects.filter(Q(publisher="Publisher A") | Q(publisher="Publisher B"))
- ( ) Book.objects.filter(publisher="Publisher A" || publisher="Publisher B")

Correct answer

- (●) Book.objects.filter(Q(publisher="Publisher A") | Q(publisher="Publisher B"))

✓ You want to retrieve all books that have been published between the years 2010 and 2020. Which query would you use? *1/1

- ( ) Book.objects.filter(published_year__gte=2010).filter(published_year__lte=2020)
- ( ) Book.objects.filter(published_year__gte=2010, published_year__lte=2020)
- (●) Book.objects.filter(published_year__range=(2010, 2020)) ✓
- ( ) Book.objects.filter(published_year__gt=2010, published_year__lt=2020)

Feedback

.filter(publication_date__year__range=(2010, 2020)) filters the queryset to include only objects where the publication_date falls within the range of years 2010 to 2020.

books_published_between_2010_and_2020 = Book.objects.filter(publication_date__year__range=(2010, 2020))

✗ Which method is used to retrieve the last object from a queryset in Django? *0/1

- (●) last_get() ✗
- ( ) get_last()
- ( ) reverse().first()
- ( ) order_by('-id').first()

Correct answer

- (●) order_by('-id').first()

✗ What is the purpose of the distinct() method in Django? * 0/1

- ( ) To remove duplicate objects from a queryset.
- (●) To select distinct fields from a queryset. ✗
- ( ) To ensure that a queryset is not empty.
- ( ) To order the queryset in descending order.

Correct answer

- (●) To remove duplicate objects from a queryset.

✗ You want to retrieve all books that are authored by "John Doe" and published after the year 2015. Which query would you use? *0/1

- ○ Book.objects.filter(author="John Doe").filter(published_year__gt=2015)
- ◉ Book.objects.filter(author="John Doe" && published_year__gt=2015) ✗
- ○ Book.objects.filter(author="John Doe" & published_year__gt=2015)
- ○ Book.objects.filter(Q(author="John Doe") & Q(published_year__gt=2015))

Correct answer

- ◉ Book.objects.filter(Q(author="John Doe") & Q(published_year__gt=2015))

✓ You want to retrieve all books that are either available for purchase or have a rating of more than 4.5. Which query would you use? *1/1

- ○ Book.objects.filter(available=True || rating__gt=4.5)
- ◉ Book.objects.filter(Q(available=True) | Q(rating__gt=4.5)) ✓
- ○ Book.objects.filter(available=True).filter(rating__gt=4.5)
- ○ Book.objects.filter(available=True | rating__gt=4.5)

Feedback

books_query = Book.objects.filter(Q(available_for_purchase=True) | Q(rating__gt=4.5))
** Q(available_for_purchase=True) creates a condition to filter books that are available for purchase.*
** Q(rating__gt=4.5) creates a condition to filter books with a rating greater than 4.5.*
** '|' is the bitwise OR operator, which combines the two conditions using OR logic.*

✗ You need to retrieve all books published after the year 2000. Which query would you use? *0/1

- ○ Book.objects.filter(published_year__gt=2000)
- ○ Book.objects.filter(published_year__gte=2000)
- ◉ Book.objects.filter(published_year__lt=2000) ✗
- ○ Book.objects.filter(published_year__lte=2000)

Correct answer

- ◉ Book.objects.filter(published_year__gt=2000)

✓ Suppose you have a Django model **Book** with fields **title and author.** What is the correct way to retrieve all books written by **"John Doe"**? *1/1

- ◉ Book.objects.filter(author="John Doe") ✓
- ○ Book.objects.get(author="John Doe")
- ○ Book.objects.all(author="John Doe")
- ○ Book.objects.find(author="John Doe")

Feedback

filter(author="John Doe") filters the queryset to include only objects where the author field is equal to "John Doe".

✗ You want to retrieve all books that are available for purchase. Which query would you use? *0/1

- ○ Book.objects.filter(available=True)
- ◉ Book.objects.filter(availability=True) ✗
- ○ Book.objects.filter(available="Yes")
- ○ Book.objects.filter(status="Available")

Correct answer

- ◉ Book.objects.filter(available=True)

Name *

MD. NAYEM

This form was created inside of DIPTI.

Google Forms