

Practice Questions

Models

INT 253

Questions

Models

ForeignKey and Relationships:

- **Question:** How would you design a model structure for a library system where each book can have multiple authors and each author can write multiple books? Create the necessary models and show how you would set up the relationships.
- **Practical Task:** Implement the models for `Book` and `Author` in Django and demonstrate how to query all books written by a specific author and all authors of a specific book.

Custom Model Methods:

- **Question:** How can you add a custom method to a Django model to calculate the total number of comments on a blog post?
- **Practical Task:** Extend the `Post` model from the previous example to include a method `total_comments` that returns the number of comments associated with a post. Write a view and template code to display this information.
-

Model Inheritance:

- **Question:** How would you design a system where you have a common set of fields for all types of products, but some products have additional fields specific to their type (e.g., books have an ISBN, electronics have a warranty period)?
- **Practical Task:** Create a base `Product` model and two derived models `BookProduct` and `ElectronicsProduct` using model inheritance. Demonstrate how to create and query these models.

Many-to-Many Relationships:

- **Question:** In a course management system, how would you model the relationship between students and courses where each student can enroll in multiple courses and each course can have multiple students?
- **Practical Task:** Design the `Student` and `Course` models with a many-to-many relationship. Show how to add a student to a course and list all students in a particular course.
-

Django Admin Customization:

- **Question:** How can you customize the Django admin interface to show additional information and improve the usability for managing blog posts and comments?
- **Practical Task:** Customize the Django admin for the `Post` and `Comment` models to include search fields, list filters, and inline editing of comments within the post admin. Provide the necessary code for `admin.py`.
-