**Write a notes for each field in the django.db.models.fields\_all, default value. default validator**

1. **AutoField**:
   * Default Value: An **AutoField** automatically generates a unique integer primary key for each new object added to the database. The default value is usually an incrementing integer starting from 1.
   * **default: int | Callable[[], int] | None = ...**
   * Default Validator: No default validator is set for **AutoField**. **validators: Iterable[\_ValidatorCallable] = ...,**
2. **BigAutoField**:
   * Default Value: Similar to **AutoField**, but it's a 64-bit integer field suitable for larger numeric ranges.
   * **default: int | Callable[[], int] | None = ...,**
   * Default Validator: No default validator is set for **BigAutoField**.
   * **validators: Iterable[\_ValidatorCallable] = ...,**
3. **BigIntegerField**:
   * Default Value: Allows storing large integers.
   * **default: \_I | Callable[[], \_I] | None = ...**
   * Default Validator: Validates that the value is a valid integer within the 64-bit range.
   * **validators: Iterable[\_ValidatorCallable] = ...**
4. **BinaryField**:
   * Default Value: Usually **None** if no data is provided. Stores binary data.
   * **default: \_BIN | Callable[[], \_BIN] | None = ...**
   * Default Validator: Validates that the value is a valid binary data.
   * **validators: Iterable[\_ValidatorCallable] = ...**
5. **BooleanField**:
   * Default Value: Defaults to **False** if not specified.
   * **default: \_B | Callable[[], \_B] | None = ...**
   * Default Validator: Ensures that the value is either **True** or **False**.
   * **validators: Iterable[\_ValidatorCallable] = ...**
6. **CharField**:
   * Default Value: An empty string **''** is the default value if not specified.
   * **default: \_C | Callable[[], \_C] | None = ...**
   * Default Validator: Validates that the value is a string with a maximum length.
   * **validators: Iterable[\_ValidatorCallable] = ...**
7. **DateField**:
   * Default Value: Defaults to **None** if not specified.
   * **default: \_DD | Callable[[], \_DD] | None = ...**
   * Default Validator: Ensures that the value is a valid date.
   * **validators: Iterable[\_ValidatorCallable] = ...**
8. **DateTimeField**:
   * Default Value: Defaults to **None** if not specified.
   * **default: \_DT | Callable[[], \_DT] | None = ...**
   * Default Validator: Validates that the value is a valid date and time.
   * **validators: Iterable[\_ValidatorCallable] = ...**
9. **DecimalField**:
   * Default Value: Defaults to **None** if not specified.
   * **default: \_DEC | Callable[[], \_DEC] | None = ...**
   * Default Validator: Validates that the value is a decimal number within specified precision and scale.
   * **validators: Iterable[\_ValidatorCallable] = ...**
10. **DurationField**:
    * Default Value: Defaults to **None** if not specified.
    * **default: \_TD | Callable[[], \_TD] | None = ...**
    * Default Validator: Validates that the value is a valid duration or time span.
    * **validators: Iterable[\_ValidatorCallable] = ...**
11. **EmailField**:
    * Default Value: An empty string **''** is the default value if not specified.
    * **default: \_C | Callable[[], \_C] | None = ...**
    * Default Validator: Validates that the value is a valid email address.
    * **validators: Iterable[\_ValidatorCallable] = ...**
12. **FilePathField**:
    * Default Value: Defaults to **None** if not specified.
    * **default: \_C | Callable[[], \_C] | None = ...**
    * Default Validator: Validates that the value is a file path and the file exists.
    * **validators: Iterable[\_ValidatorCallable] = ...**
13. **FloatField**:
    * Default Value: Defaults to **None** if not specified.
    * **default: \_F | Callable[[], \_F] | None = ...**
    * Default Validator: Validates that the value is a floating-point number.
    * **validators: Iterable[\_ValidatorCallable] = ...**
14. **IntegerField**:
    * Default Value: Defaults to **None** if not specified.
    * **default: \_I | Callable[[], \_I] | None = ...**
    * Default Validator: Validates that the value is a valid integer within a specific range.
    * **validators: Iterable[\_ValidatorCallable] = ...**
15. **PositiveIntegerField**:
    * Default Value: Defaults to **None** if not specified.
    * Default Validator: Validates that the value is a positive integer.
16. **PositiveSmallIntegerField**:
    * Default Value: Defaults to **None** if not specified.
    * **default: \_I | Callable[[], \_I] | None = ...**
    * Default Validator: Validates that the value is a positive small integer within a specific range.
    * **validators: Iterable[\_ValidatorCallable] = ...**
17. **SlugField**:
    * Default Value: An empty string **''** is the default value if not specified.
    * **default: \_C | Callable[[], \_C] | None = ...**
    * Default Validator: Validates that the value is a valid slug.
    * **validators: Iterable[\_ValidatorCallable] = ...**
18. **SmallAutoField**:
    * Default Value: Similar to **AutoField**, but limited to a smaller numeric range.
    * **default: int | Callable[[], int] | None = ...**
    * Default Validator: No default validator is set for **SmallAutoField**.
    * **validators: Iterable[\_ValidatorCallable] = ...**
19. **SmallIntegerField**:
    * Default Value: Defaults to **None** if not specified.
    * **default: \_I | Callable[[], \_I] | None = ...**
    * Default Validator: Validates that the value is a small integer within a specific range.
    * **validators: Iterable[\_ValidatorCallable] = ...**
20. **TextField**:
    * Default Value: An empty string **''** is the default value if not specified.
    * **default: \_C | Callable[[], \_C] | None = ...**
    * Default Validator: No specific default validator is set for **TextField**.
    * **validators: Iterable[\_ValidatorCallable] = ...**
21. **TimeField**:
    * Default Value: Defaults to **None** if not specified.
    * **default: \_TM | Callable[[], \_TM] | None = ...**
    * Default Validator: Validates that the value is a valid time.
    * **validators: Iterable[\_ValidatorCallable] = ...**

**Write a parent class to each model field**

1. Field
2. DateTimeCheckMixin
3. PositiveIntegerRelDbTypeMixin
4. AutoFieldMixin
5. **BooleanField**(Field) -->A true/false field.   
   The default form widget for this field is a CheckboxInput.
6. **CharField**(Field) -->A field to store text-based values.
7. **CommaSeparatedIntegerField**(CharField)
8. **DateField**(DateTimeCheckMixin, Field) -->A date, represented in Python by a datetime.date instance
9. **DateTimeField**(DateField) -->It is used for date and time, represented in Python by a datetime.datetime instance.
10. **DecimalField**(Field) -->It is a fixed-precision decimal number, represented in Python by a Decimal instance.
11. **DurationField**(Field) -->A field for storing periods of time.
12. **EmailField**(CharField) -->It is a CharField that checks that the value is a valid email address.
13. **FilePathField**(Field) -->It is a file-upload field.
14. **FloatField**(Field) -->It is a floating-point number represented in Python by a float instance.
15. **IntegerField**(Field) -->It is an integer field. Values from -2147483648 to 2147483647 are safe in all databases supported by Django.
16. **BigIntegerField**(IntegerField) -->It is a 64-bit integer, much like an IntegerField except that it is guaranteed to fit numbers from -9223372036854775808 to 9223372036854775807.
17. **SmallIntegerField**(IntegerField) -->It is like an IntegerField, but only allows values under a certain (database-dependent) point.
18. **IPAddressField**(Field)
19. **NullBooleanField**(BooleanField) -->Like a BooleanField, but allows NULL as one of the options.
20. **PositiveBigIntegerField**(PositiveIntegerRelDbTypeMixin, BigIntegerField)
21. **PositiveIntegerField**(PositiveIntegerRelDbTypeMixin, IntegerField) -->Like an IntegerField, but must be either positive or zero (0).
22. **PositiveSmallIntegerField**(PositiveIntegerRelDbTypeMixin, SmallIntegerField) -->Like a PositiveIntegerField, but only allows values under a certain (database-dependent) point.
23. **SlugField**(CharField) -->Slug is a newspaper term. A slug is a short label for something, containing only letters, numbers, underscores or hyphens. They’re generally used in URLs.
24. **TextField**(Field) -->A large text field. The default form widget for this field is a Textarea.
25. **TimeField**(DateTimeCheckMixin, Field) -->A time, represented in Python by a datetime.time instance.
26. **URLField**(CharField) -->A CharField for a URL, validated by URLValidator.
27. **BinaryField**(Field) -->A field to store raw binary data.
28. **UUIDField**(Field) -->A field for storing universally unique identifiers. Uses Python’s UUID class. When used on PostgreSQL, this stores in a uuid datatype, otherwise in a char(32).
29. **AutoFieldMeta**(type) -->It is an IntegerField that automatically increments.
30. **AutoField**(AutoFieldMixin, IntegerField, metaclass=AutoFieldMeta)
31. **SmallAutoField**(AutoFieldMixin, SmallIntegerField)

**Prepare some shell notes, to insert some rows in to database tables. call full\_clean method to show the validations.**

* First we have to import the model from where we have created model classes. Like **from app\_name.models import class\_names**
* Then create instance for class in the instance we can pass keyword arguments as column name as key and value for what required for that column
* If we miss any fields in the instance while inserting data. It will not validate.
* To validate whether we have entered all field we have to check with **object.full\_clean()**
* And finally to push the data into database we use **object.save()**