|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(1,2,3,,4,5) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Authentication Group. Ex: 1 |
| Name(students,library staff,faculties) | Text | 300 | Name of the Authentication Group.  Ex: Students |
| Message\_board\_notice |  |  | Contains a notice as a Foreign Key. |
| Auth\_user\_groups |  |  | Contains a Auth User Group as a Foreign Key. |
| Auth\_group\_permissions |  |  | This contains the Auth Group Permissions as a Foreign Key. |

auth\_group

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(1,2,3,4,121) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Authentication Group Permissions.  Ex: 1 |
| group\_id(2,2,3,3,5) | Number | Any positive number | Group ID from the Group Table as a Foreign Key.  Ex: 1 |
| permission\_id(1,2,3,4,45) | Number | Any positive number | Permission ID from the Permission Table as a Foreign Key.  Ex: 1 |

Auth\_group\_permissions

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Content\_type\_id (1,1,2,2,3) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Authentication Permissions. Ex: 1 |
| Codename(add\_logentry,  Change\_logentry,add\_book) |  |  |  |
| Name(can add log entry,can change log entry,can add group) | EI TABLE E PROBLEM ASE  CAUSE EIKHANE USER MODEL TA DATA TYPE  SO KI DIBO? |  |  |
| Auth\_group\_permissions |  |  | This contains the Auth Group Permissions as a Foreign Key. |

Auth\_permission

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(1,2,3,4,5) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Authenticated User. Ex: 1 |
| Last\_login( 2018-06-13 13:29:33.005402,null) | Date-Time |  | This contains the Date and Time of the Last Login.  Ex: 2018-06-21 13:29:33.005402 |
| Is\_superuser(1,0,0,0) | Boolean | Yes/No | This contains the Boolean information if the User is Superuser or not.  Ex: Yes,  No |
| Username(john,leon,shamu) | Text | 100 | This contains the Account’s User Name.  Ex: Mdadul Hakqe |
| First\_name(ratul,leon,john | Text | 100 | This contains the User’s First Name.  Ex: Charles |
| Email(ratul\_shams@yahoo.com, kazi\_mdadul@yahoo.com) | Email | 100 | This contains the User’s primary email.  Ex: Ratul\_Shams@yahoo.com |
| Is\_staff(1,0,0,0) | Boolean | Yes/No | This contains the Boolean information if the User is staff or not.  Ex: Yes,  No |
| Is\_active(1,1,1,1) | Boolean | Yes/No | This contains the Boolean information if the User Account is active or not.  Ex: Yes,  No |
| Data\_joined | Date | “YYYY-MM-DD HH:MM:SS, YYYY-MM-DD HH:MM:SS, “ | This contains the Date and Time of when the User Account was created.  Ex: 2018-06-21 13:29:33.005402 |
| Last\_name(hasan,sutapa) | Text | 100 | This contains the User’s Last Name.  Ex: Dickens |
| Account\_profile |  |  | This contains a Account Profile as a Foreign Key. |
| Django\_admin\_log |  |  | This is the admin log as a foreign key. |
| Catalogue\_Book\_instance |  |  | This contains the book instance as the Foreign Key. |
| Auth\_user\_groups |  |  | Contains a Auth User Group as a Foreign Key. |

Auth\_user

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(1,2,3,4) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Authenticated User group. Ex: 1 |
| User\_id(2,3,4) | Number | Any positive number | This contains the ID from the User Table as a Foreign Key.  Ex: 1 |
| Group\_id(1,2,3,4 | Number | Any positive number | This contains the ID from the Group Table as a Foreign Key.  Ex: 1 |

Auth\_user\_groups

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
|  |  |  | SELECT \* FROM 'auth\_user\_user\_permissions' LIMIT 0,30 |

Auth\_user\_user\_permissions

Catalog\_author

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(1,2) |  | Any positive number | This is the primary key of this relation. This contain the ID of the Author. Ex: 1 |
| First\_name(john,ratul) | Text | 100 | This contains the First Name of the Author.  Ex: Charles |
| Last\_name(kumar,khan) | Text | 100 | This contains the Last Name of the Author.  Ex: Dickens |
| Data\_of\_birth(2018-06-01,2018-03-13) | Date | “YYYY-MM-DD” | This contains the Date of Birth of the Author.  Ex: 2018-06-21 |
| Date\_of\_death(2018-11-16,null) | Date | “YYYY-MM-DD” | This contains the Date of Death of the Author.  Ex: 2018-06-21 |
| Catalog\_book |  |  | This contains the Book from the Book Table as a Foreign Key. |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(First Book, Second Book) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Book.  Ex: 1 |
| Summary(this is the first book..!,thi is the second book..!) | Text | 200 | This contains the text summary of the book.  Ex: ADD SUMMARY HERE |
| Isbn(4356457567523, 089032341429) | Number | Standard ISBN number | This contains a standard ISBN number of the Book.  Ex: 4356457567523 |
| Author\_id(1,2) | Number | Any positive number | This contains the ID from the Author Table as a Foreign Key.  Ex: 1 |
| Catalog\_bookinstance |  |  | This contains the Book Instance as the Foreign Key. |
| Catalog\_book\_genre |  |  | This contains the Book Genre as the Foreign Key. |

Catalog\_book

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(1,2) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Book Genre. Ex: 1 |
| Book\_id(1,2) | Number | Any positive number | This contains the ID from the Book Table as a Foreign Key.  Ex: 1 |
| Genre\_id(2,1) | Number | Any positive number | This contains the ID from the Genre Table as a Foreign Key.  Ex: 1 |
| Catalog\_book\_genre |  |  | This contains the Book Genre as the Foreign Key. |

Catalog\_book\_genre

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(e3233df896b84bfd89e55637212f018f,  235452acaaea4632ba4749f3cb401032) | Text | 100 | This is the primary key of this relation. This contain the ID of the Book Instance.  Ex: 1 |
| Imprint(yes,no) | Boolean | Yes/No | This is a Boolean data if the book is imprint or not.  Ex: Yes,  No |
| Due\_back( 2018-06-01, null) | Date | YYY-MM-DD | This contains the Due Back Date Time.  Ex: 2018-06-21 |
| Status(a,a) | Text | A - Available,  W - Withdrawn,  M - Maintenance,  R - Reserved,  etc | This contains the Availability Status of a Book.  Ex: Available |
| Book\_id(1,2) | Number | Any positive number | This contains the ID from the Book Table as a Foreign Key.  Ex: 1 |
| Borrower\_id(null,null) | Number | Any positive number | This contains the ID from the User Table as a Foreign Key.  Ex: 1 |

Catalog\_bookinstance

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(1,2) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Genre. Ex: 1 |
| Name(horror,comedy) | Text | 50 | This is the Name of the Genre.  Ex: Horror,  Comedy |

Catalog\_genre

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(1,2,3,4) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Admin Log. Ex: 1 |
| Object\_id(1,2,1,2) | Number | Any positive number | This contains the ID from the Object Table as a Foreign Key.  Ex: 1 |
| Object\_repr(kumar,Anil,  Comedy,horror) |  |  |  |
| Action\_flag(1,1,1,2) |  |  |  |
| Change\_message([{"added": {}}], [{"added": {}}]) |  |  |  |
| Content\_type\_id(10,10,9) | Number | Any positive number | This contains the ID from the Content Type Table as a Foreign Key.  Ex: 1 |
| User\_id(1,1,1,1) | Number | Any positive number | This contains the ID from the User Table as a Foreign Key.  Ex: 1 |
| Action\_time( 2018-06-01 20:30:18.107052, 2018-06-01 20:30:42.179193) | Date-Time | “YYYY-MM-DD HH:MM:SS, YYYY-MM-DD HH:MM:SS, “ | This contains the Action Date Time.  Ex: 2018-06-21 19:07:13.182173,  2018-06-26 13:22:36.450456 |

Django\_admin\_log

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(1,2,3,4) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Content Type. Ex: 1 |
| App\_label(admin,auth) |  |  |  |
| Model(logentry,group) |  |  |  |

Django\_content\_type

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id(1,2,3,4) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Migration.  Ex: 1 |
| App(contenttypes,auth,admin) |  |  |  |
| Name(0001\_initial,0001\_initial) | Text | “NNNN\_MIGRATION-NAME” | This contains the autogenerated Name of the Migration.  Ex: 0001\_initial,  0001\_initial |
| Applied(2018-06-01 17:41:39.937277,2018-06-01 17:41:40.022913) | Date-Time | “YYYY-MM-DD HH:MM:SS, YYYY-MM-DD HH:MM:SS, “ | This contains the Migration Applied Date.  Ex: 2018-06-21 19:07:13.182173,  2018-06-26 13:22:36.450456 |

Django\_migrations

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Session\_key(m69agka8ui8gyl0wrnqus8fa3oz2mhp5,  seawglfxjmaggb7c6v2iel6hzchj3jpp) | Text | 300 | This contains the Session Key. |
| Session\_data( |  |  | This contains the Session Data. |
| Expire\_data(2018-06-21 19:07:13.182173,2018-06-26 13:22:36.450456) | Date-Time | “YYYY-MM-DD HH:MM:SS, YYYY-MM-DD HH:MM:SS, “ | This contains the Expire Data.  Ex: 2018-06-21 19:07:13.182173,  2018-06-26 13:22:36.450456 |

Django\_session

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id (1,2,3) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Notice. Ex: 1 |
| Notice\_title (Make Up Class, Mid Term Examination) | Text | 300 | This contains the Title of the Notice. |
| Notice\_text (Due to Durga Puga, makeup class for 15th february will be held on 20th February) | Text | 1000 | This contains the Description of the Notice. |
| Published\_date(2018-06-05) | Date | “DD/MM/YYYY” | This contains the Published Date of the Notice. |
| Event\_date (2018-06-15,2018-07-05) | Date | “DD/MM/YYYY” | This contains the Event Date of the Notice. |
| Event\_kind (n,m,h) | Text | 1 | This contains the Event Type of the Notice.  Ex: n - Notice, m - Month, h - Holiday |
| Posted\_by\_id(null,null) | Number | Any positive number | This contains the Group ID - from the Group Table as a Foreign Key - that posted the Notice.  Ex: 1 |

Message\_board\_notice

Sqlite\_sequence

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Name(django\_migrations,  Django\_admin\_log) |  |  |  |
| Seq (22,30) |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| Id (1) | Number | Any positive number | This is the primary key of this relation. This contain the ID of the Profile.  Ex: 1 |
| Date\_of\_birth (2018-06-21) | Date | ‘’DD-MM-YYYY” | This contains date of birth of the User.  Ex: 29/01/81 |
| photo | Image | .jpg, or any well known image format | This contains the User Account’s Profile Picture. |
| User\_id (1) | Number | Any positive number | This contains the User ID from Users Table as a Foreign Key.  Ex: 1 |

Account\_profile