PRISMA

- Prisma is an open source next-generation ORM.
- Works with MySQL, SQLite, MSSQL, PostgreSQL and also with mongoDB.

PRISMA

Prisma Client

Auto-generated and type-safe query builder for Node.js & TypeScript

Prisma Migrate

Migration tool to easily evolve your database schema from prototyping to production

Prisma Studio

GUI to view and edit data in your database

PRISMA

- Prisma with Next.js
- Prisma with Express
- Prisma with Apollo
- Prisma with Nestjs
- Prisma with GraphQL

PRISMA

MySQL®

PostgreSQL

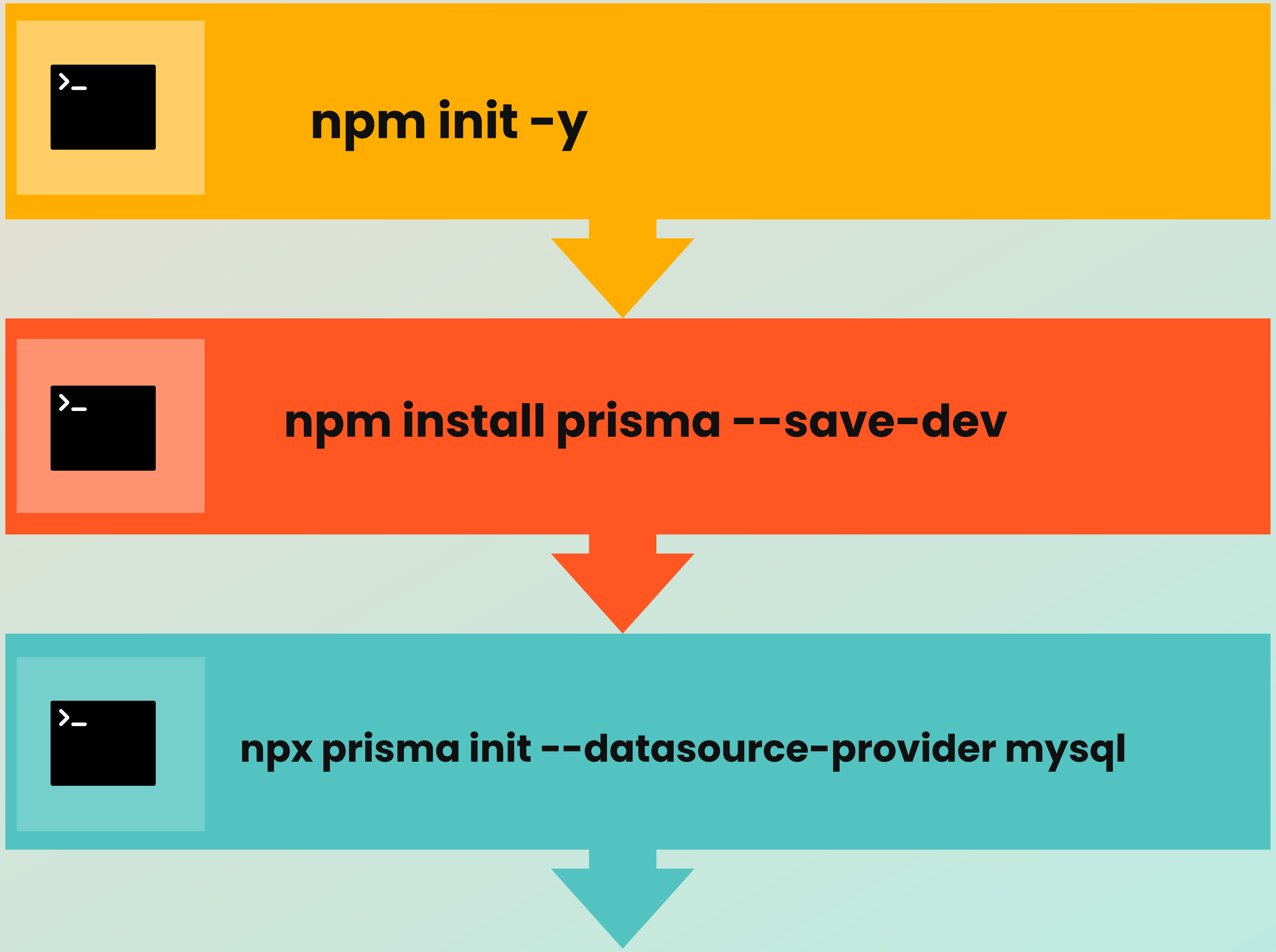
mongoDB

Cockroach DB

Microsoft® SQL Server®

SQLite

Create **New Project** & Explore



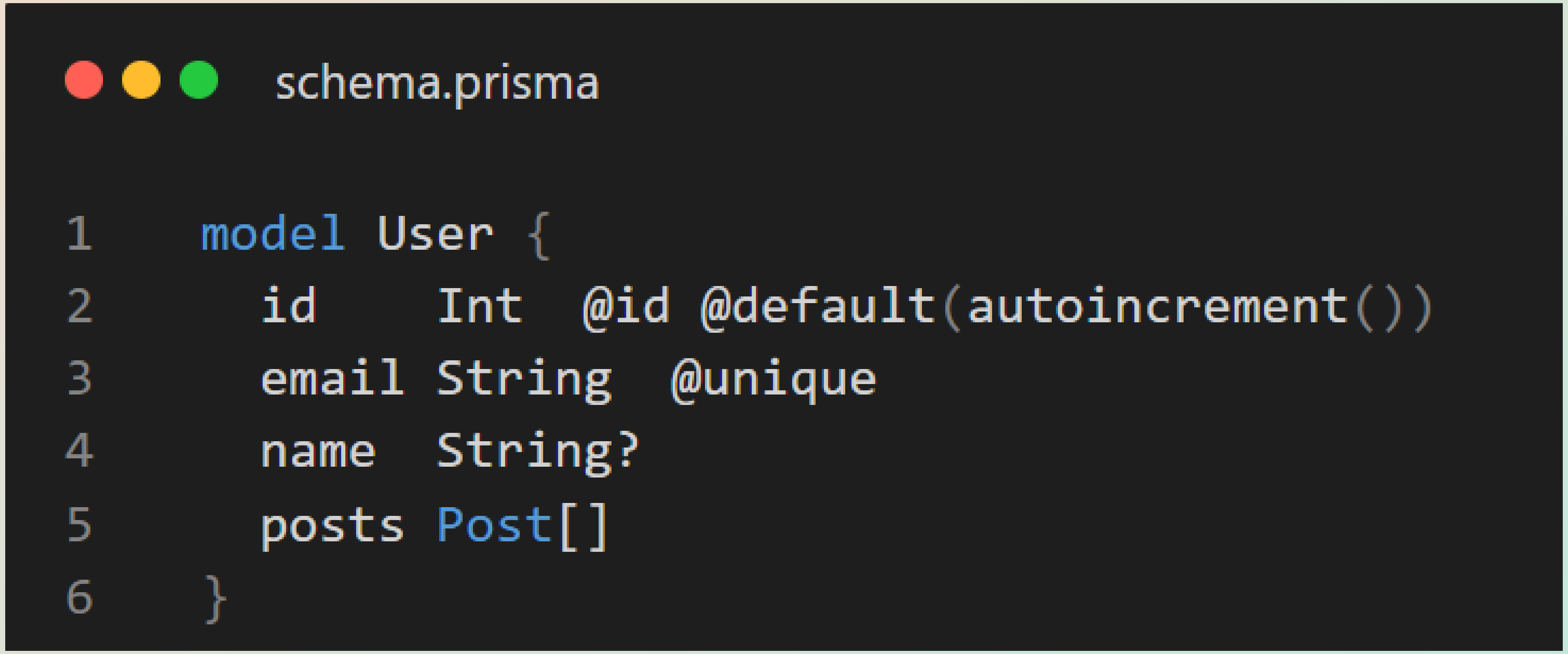
```
npm init -y
```

```
npm install prisma --save-dev
```

```
npx prisma init --datasource-provider mysql
```

Create **Your First** Migration

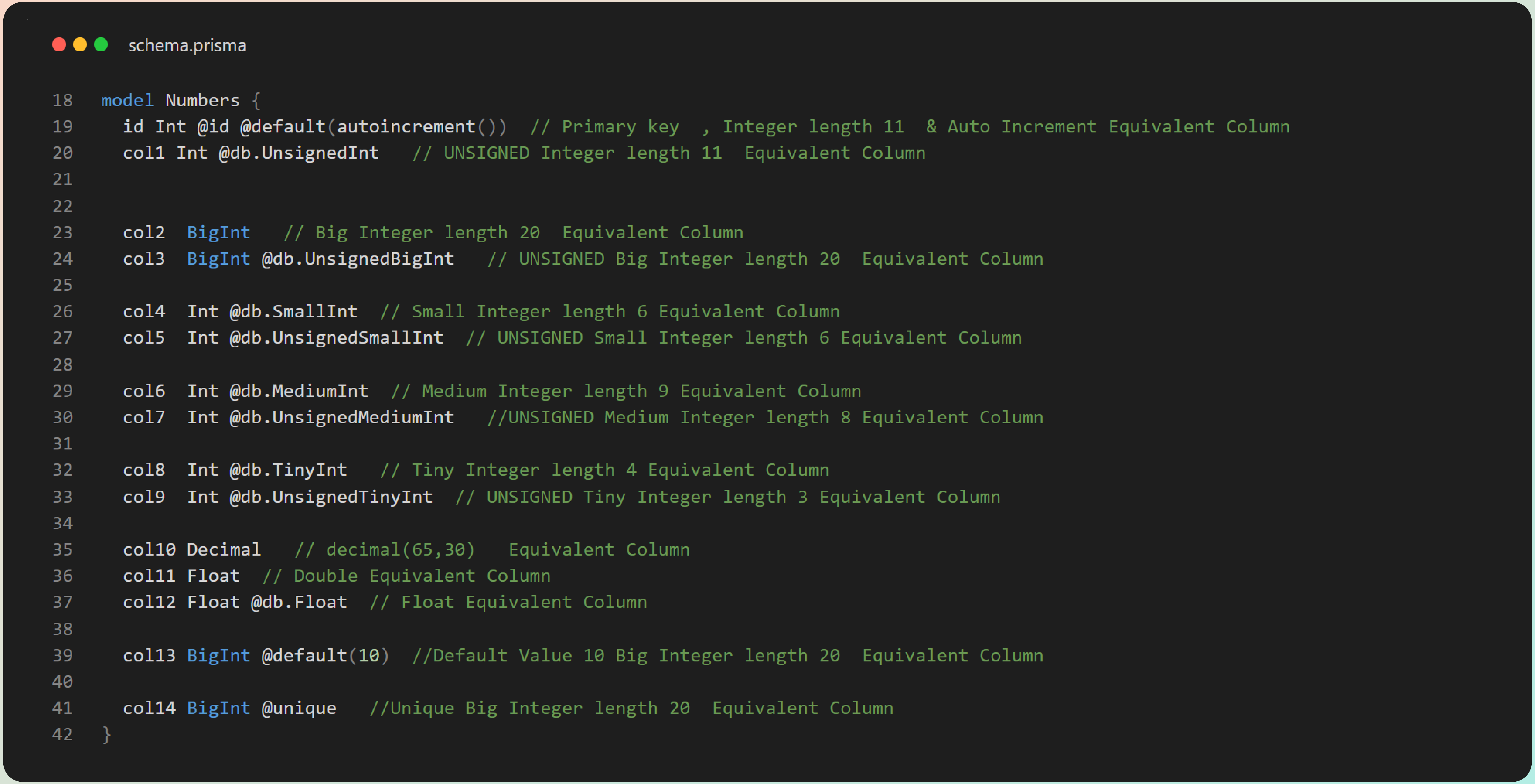
- Every record of a model must be uniquely identifiable.
- Model names must start with a letter and are typically spelled in PascalCase
- Model names should use the singular form (for example, User instead of user, users or Users)



```
model User {
  id    Int  @id @default(autoincrement())
  email String  @unique
  name  String?
  posts Post[]
}
```

npx prisma migrate dev

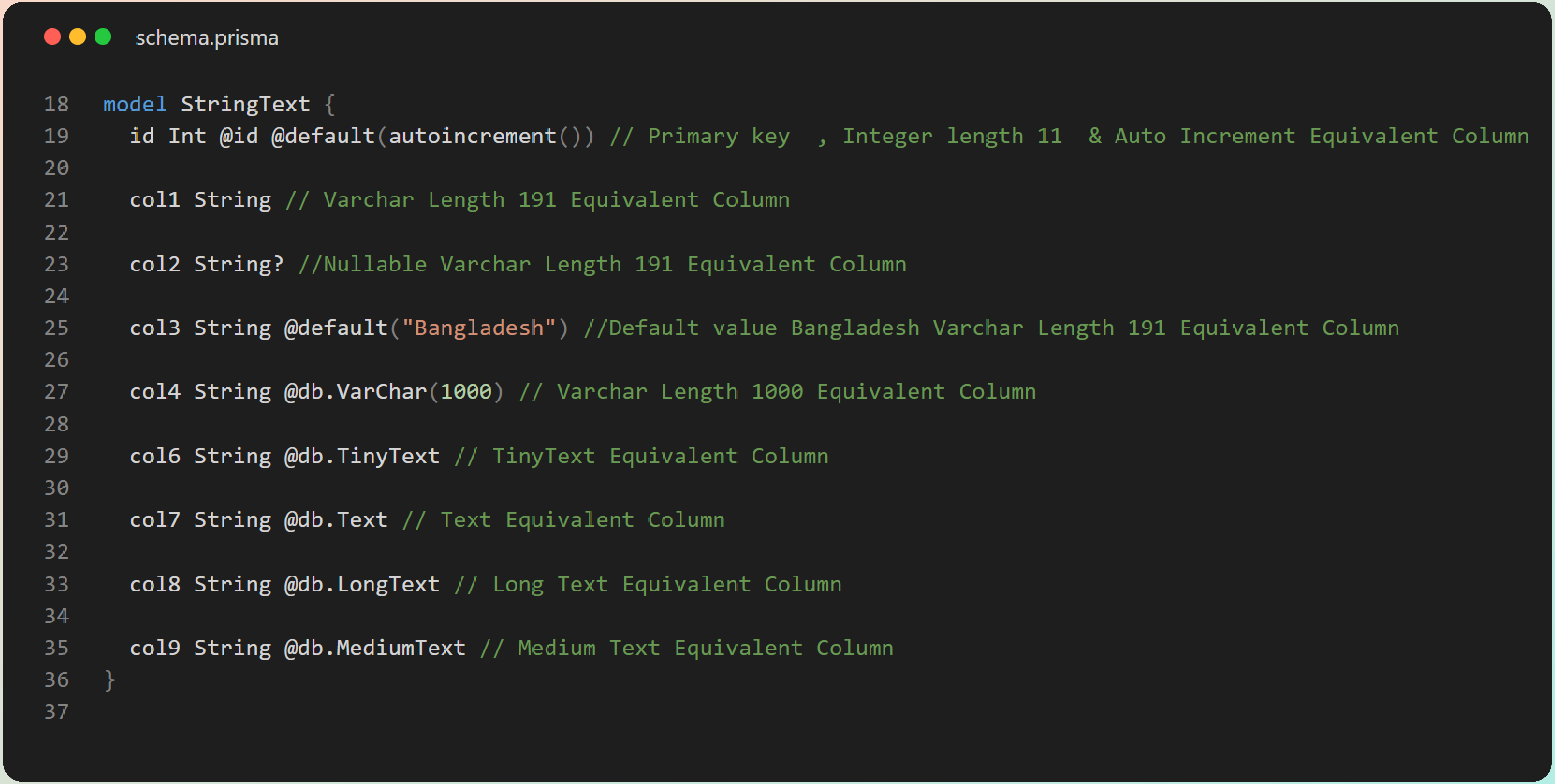
Migration Properties (Number Type)



schema.prisma

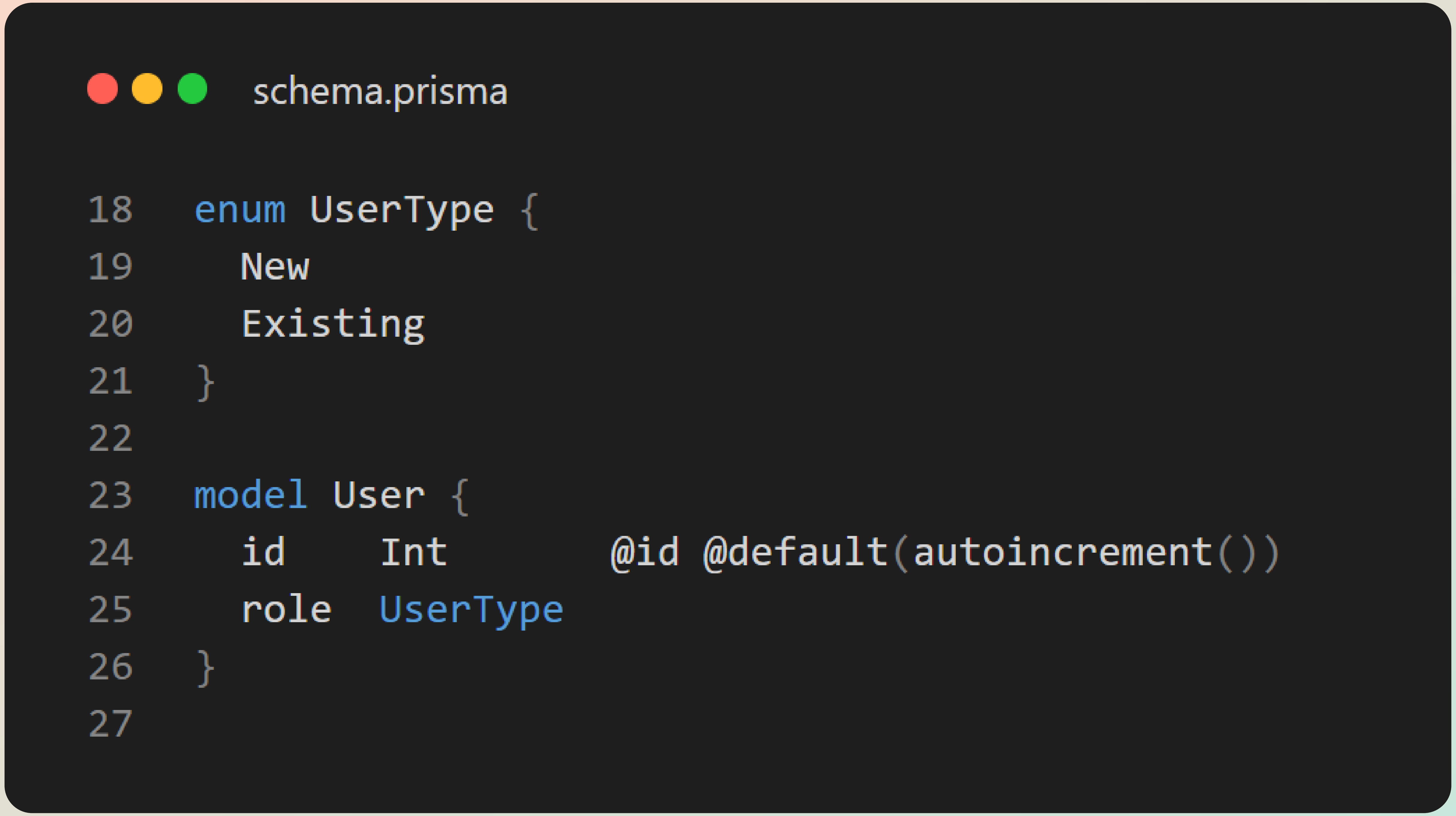
```
model Numbers {
  id Int @id @default(autoincrement())  // Primary key  , Integer length 11  & Auto Increment Equivalent Column
  col1 Int @db.UnsignedInt   // UNSIGNED Integer length 11  Equivalent Column


  col2  BigInt   // Big Integer length 20  Equivalent Column
  col3  BigInt @db.UnsignedBigInt   // UNSIGNED Big Integer length 20  Equivalent Column

  col4  Int @db.SmallInt  // Small Integer length 6 Equivalent Column
  col5  Int @db.UnsignedSmallInt  // UNSIGNED Small Integer length 6 Equivalent Column

  col6  Int @db.MediumInt  // Medium Integer length 9 Equivalent Column
  col7  Int @db.UnsignedMediumInt   //UNSIGNED Medium Integer length 8 Equivalent Column

  col8  Int @db.TinyInt   // Tiny Integer length 4 Equivalent Column
  col9  Int @db.UnsignedTinyInt  // UNSIGNED Tiny Integer length 3 Equivalent Column

  col10 Decimal   // decimal(65,30)   Equivalent Column
  col11 Float  // Double Equivalent Column
  col12 Float @db.Float  // Float Equivalent Column

  col13 BigInt @default(10)  //Default Value 10 Big Integer length 20  Equivalent Column

  col14 BigInt @unique   //Unique Big Integer length 20  Equivalent Column
}
```

Migration Properties (String Type)



```
model StringText {
  id Int @id @default(autoincrement()) // Primary key  , Integer length 11  & Auto Increment Equivalent Column

  col1 String // Varchar Length 191 Equivalent Column

  col2 String? //Nullable Varchar Length 191 Equivalent Column

  col3 String @default("Bangladesh") //Default value Bangladesh Varchar Length 191 Equivalent Column

  col4 String @db.VarChar(1000) // Varchar Length 1000 Equivalent Column

  col6 String @db.TinyText // TinyText Equivalent Column

  col7 String @db.Text // Text Equivalent Column

  col8 String @db.LongText // Long Text Equivalent Column

  col9 String @db.MediumText // Medium Text Equivalent Column
}

```

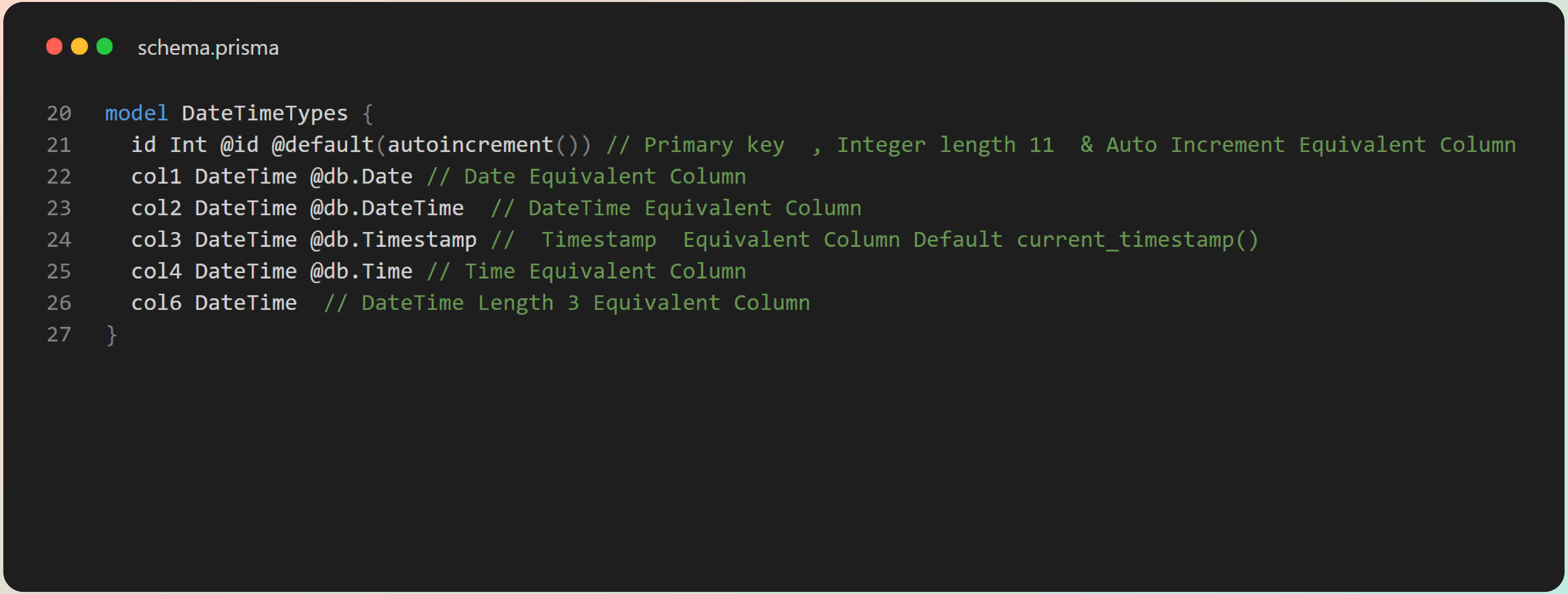
Migration Properties (Enum Type)



schema.prisma

```
enum UserType {
  New
  Existing
}

model User {
  id    Int       @id @default(autoincrement())
  role  UserType
}

```

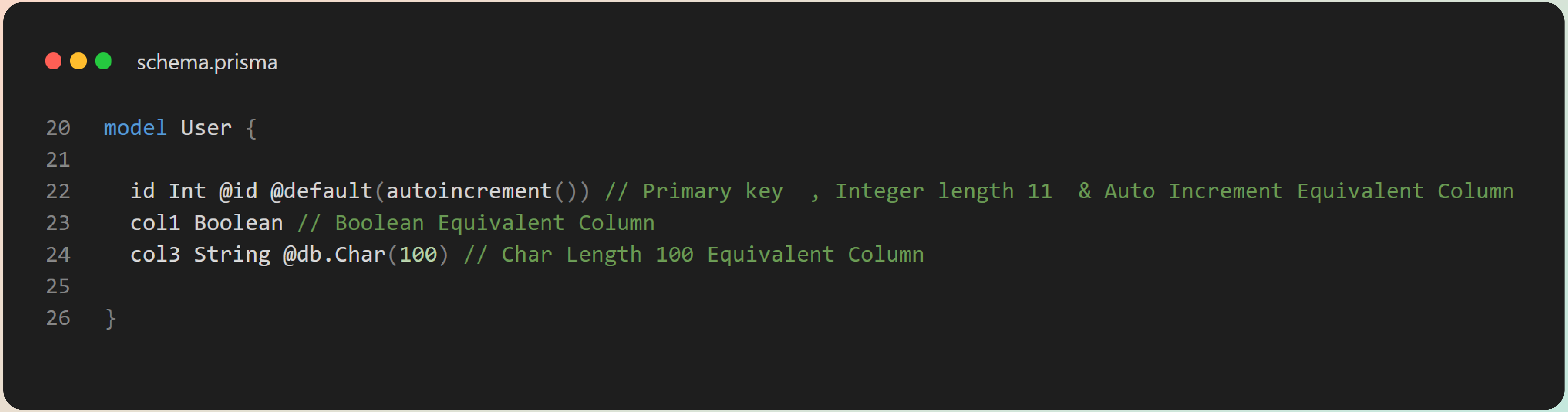
Migration Properties (Date Time Type)



schema.prisma

```
model DateTimeTypes {
  id Int @id @default(autoincrement()) // Primary key  , Integer length 11  & Auto Increment Equivalent Column
  col1 DateTime @db.Date // Date Equivalent Column
  col2 DateTime @db.DateTime  // DateTime Equivalent Column
  col3 DateTime @db.Timestamp //  Timestamp  Equivalent Column Default current_timestamp()
  col4 DateTime @db.Time // Time Equivalent Column
  col6 DateTime  // DateTime Length 3 Equivalent Column
}
```

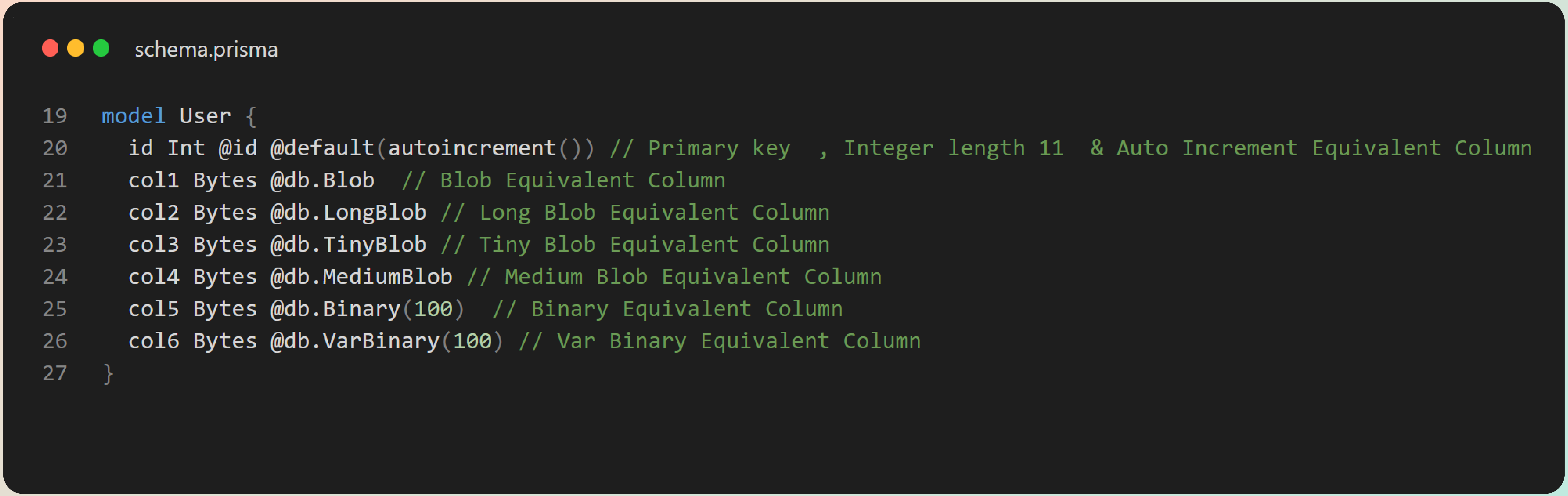
Migration Properties (Boolean & Char Type)



schema.prisma

```
model User {

  id Int @id @default(autoincrement()) // Primary key  , Integer length 11  & Auto Increment Equivalent Column
  col1 Boolean // Boolean Equivalent Column
  col3 String @db.Char(100) // Char Length 100 Equivalent Column

}
```

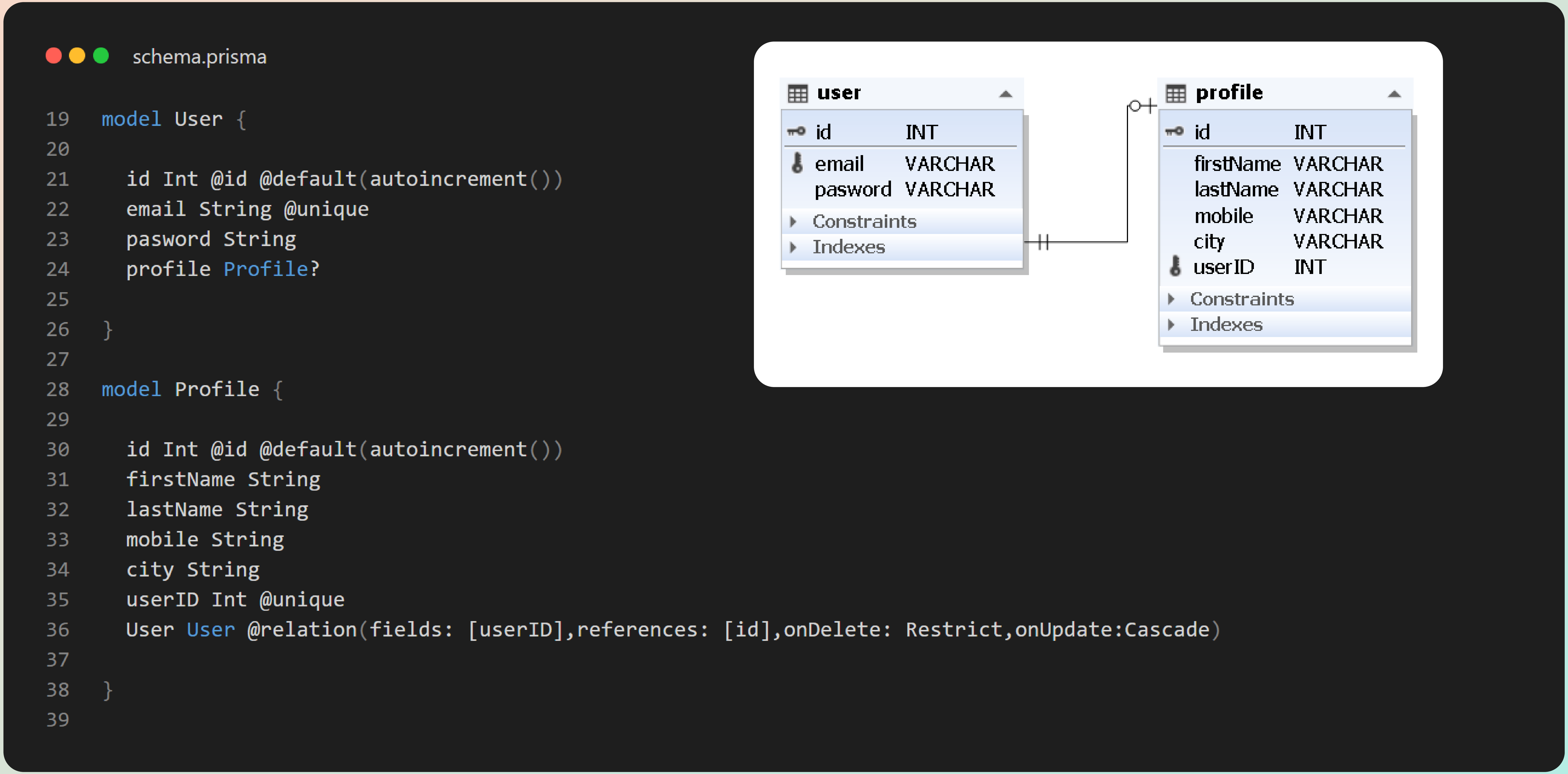
Migration Properties (Blob Type)



schema.prisma

```prisma
model User {
  id Int @id @default(autoincrement()) // Primary key  , Integer length 11  & Auto Increment Equivalent Column
  col1 Bytes @db.Blob  // Blob Equivalent Column
  col2 Bytes @db.LongBlob // Long Blob Equivalent Column
  col3 Bytes @db.TinyBlob // Tiny Blob Equivalent Column
  col4 Bytes @db.MediumBlob // Medium Blob Equivalent Column
  col5 Bytes @db.Binary(100)  // Binary Equivalent Column
  col6 Bytes @db.VarBinary(100) // Var Binary Equivalent Column
}
```

Migration One to One



schema.prisma

```
model User {

  id Int @id @default(autoincrement())
  email String @unique
  pasword String
  profile Profile?

}

model Profile {

  id Int @id @default(autoincrement())
  firstName String
  lastName String
  mobile String
  city String
  userID Int @unique
  User User @relation(fields: [userID],references: [id],onDelete: Restrict,onUpdate:Cascade)

}
```

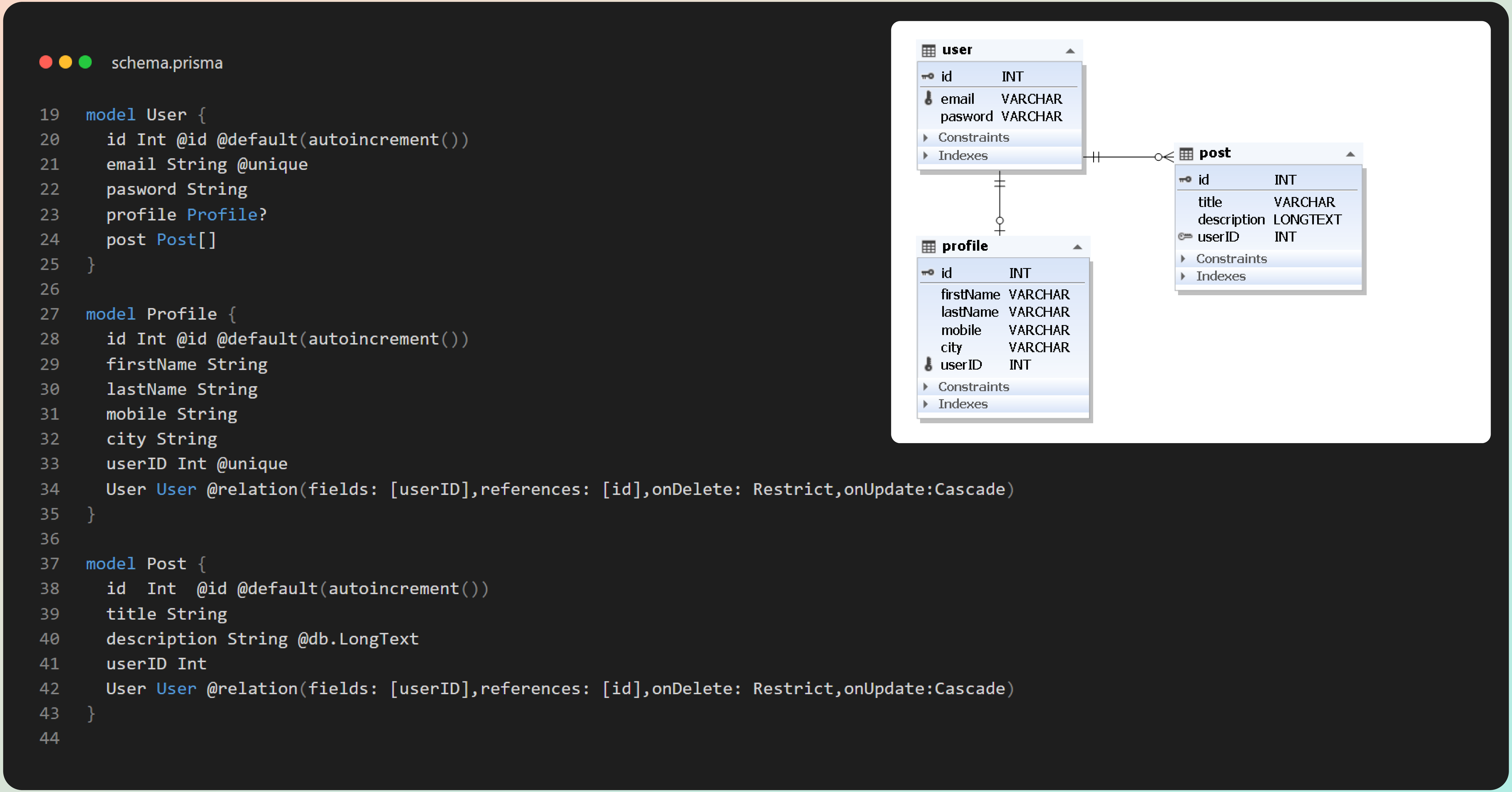
user

id	INT
email	VARCHAR
pasword	VARCHAR

profile

id	INT
firstName	VARCHAR
lastName	VARCHAR
mobile	VARCHAR
city	VARCHAR
userID	INT

Migration One to Many



schema.prisma

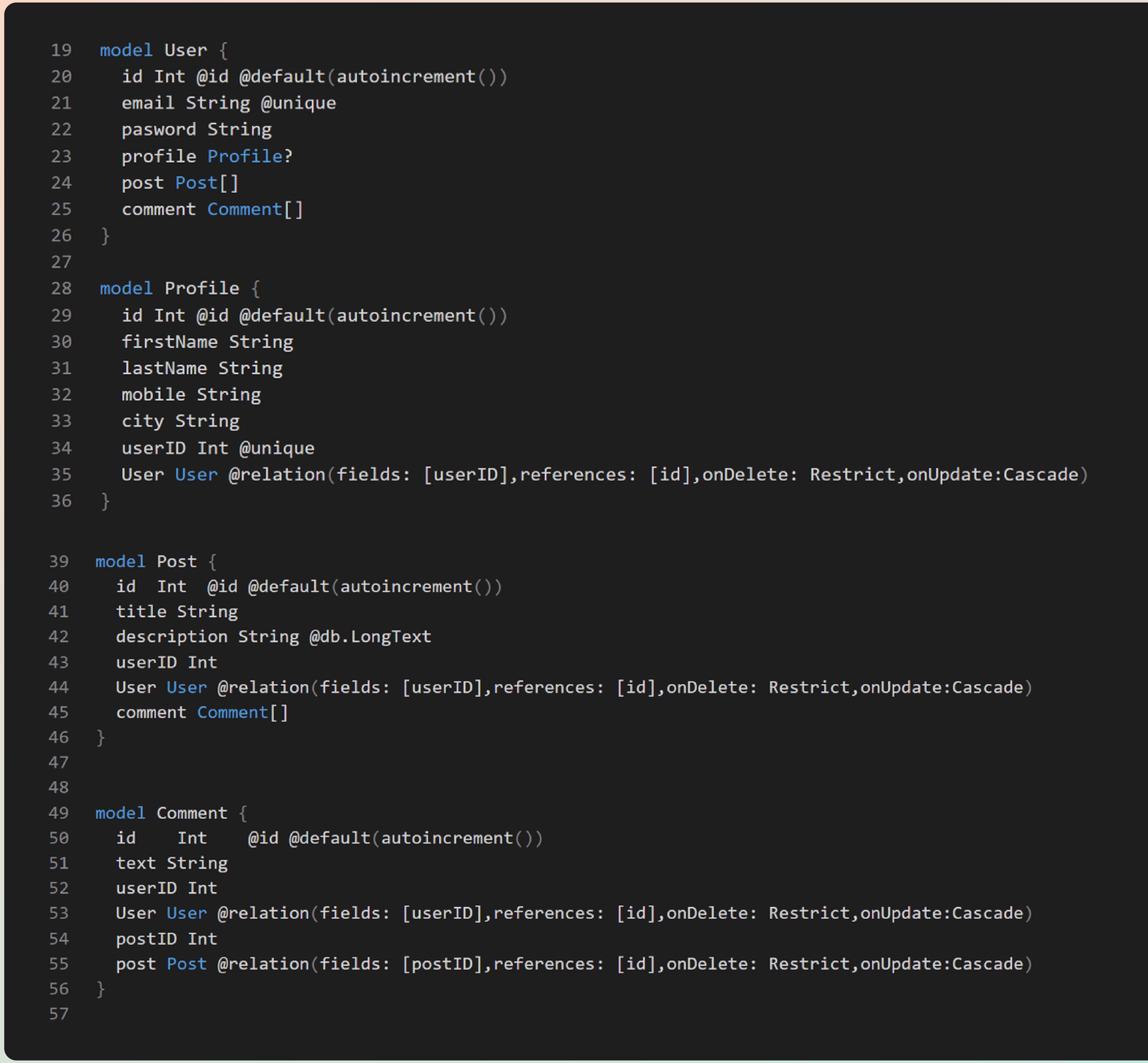
```
model User {
  id Int @id @default(autoincrement())
  email String @unique
  pasword String
  profile Profile?
  post Post[]
}

model Profile {
  id Int @id @default(autoincrement())
  firstName String
  lastName String
  mobile String
  city String
  userID Int @unique
  User User @relation(fields: [userID],references: [id],onDelete: Restrict,onUpdate:Cascade)
}

model Post {
  id  Int  @id @default(autoincrement())
  title String
  description String @db.LongText
  userID Int
  User User @relation(fields: [userID],references: [id],onDelete: Restrict,onUpdate:Cascade)
}
```

user	
id	INT
email	VARCHAR
pasword	VARCHAR

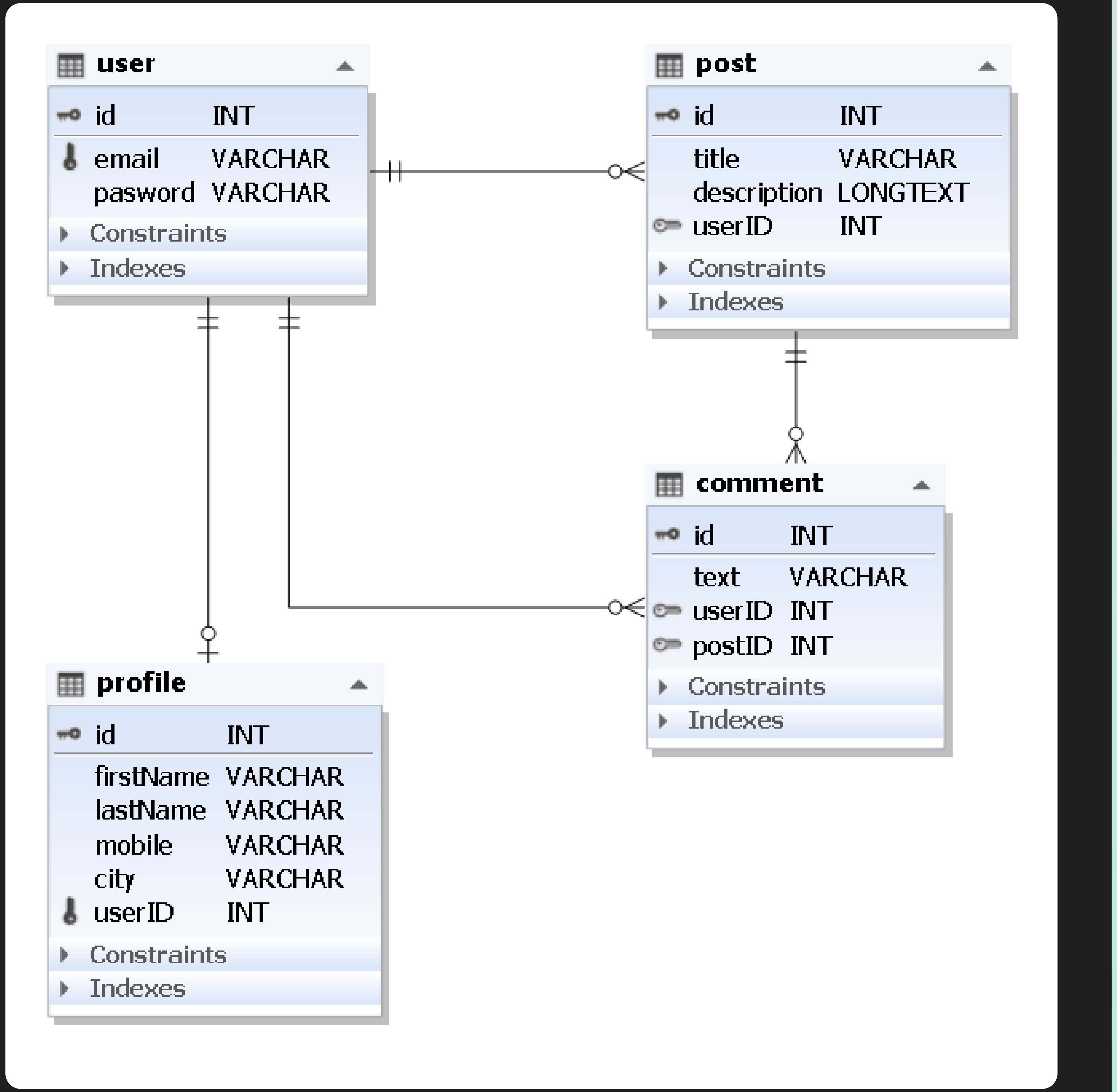
profile	
id	INT
firstName	VARCHAR
lastName	VARCHAR
mobile	VARCHAR
city	VARCHAR
userID	INT

post	
id	INT
title	VARCHAR
description	LONGTEXT
userID	INT

Migration Many to Many



```
model User {
  id Int @id @default(autoincrement())
  email String @unique
  pasword String
  profile Profile?
  post Post[]
  comment Comment[]
}

model Profile {
  id Int @id @default(autoincrement())
  firstName String
  lastName String
  mobile String
  city String
  userID Int @unique
  User User @relation(fields: [userID],references: [id],onDelete: Restrict,onUpdate:Cascade)
}

model Post {
  id  Int  @id @default(autoincrement())
  title String
  description String @db.LongText
  userID Int
  User User @relation(fields: [userID],references: [id],onDelete: Restrict,onUpdate:Cascade)
  comment Comment[]
}


model Comment {
  id    Int    @id @default(autoincrement())
  text String
  userID Int
  User User @relation(fields: [userID],references: [id],onDelete: Restrict,onUpdate:Cascade)
  postID Int
  post Post @relation(fields: [postID],references: [id],onDelete: Restrict,onUpdate:Cascade)
}
```



user	
id	INT
email	VARCHAR
pasword	VARCHAR
Constraints	
Indexes	

post	
id	INT
title	VARCHAR
description	LONGTEXT
userID	INT
Constraints	
Indexes	

comment	
id	INT
text	VARCHAR
userID	INT
postID	INT
Constraints	
Indexes	

profile	
id	INT
firstName	VARCHAR
lastName	VARCHAR
mobile	VARCHAR
city	VARCHAR
userID	INT
Constraints	
Indexes	