

Prisma migration guide

Scenario

We have a cohort manager app. One of the models we have in our app is "Cohort". At the moment, a Cohort object looks like so:

```
{  
  id: 1,  
  students: [] // array of students  
}
```

We would like to add a `cohortName` to the model, however, we already have quite a few cohorts in the production database.

The general steps we will need to take are as follows:

- Add the field allowing it to be null in the prisma schema
- Create an empty migration
- In our empty migration, add a `cohortName` to the existing cohorts
- Update the prisma schema to remove the nullable (remove the `?`)

Create migration with nullable field

In order to allow the existing cohorts to exist without a name, we must make the `cohortName` nullable. We can do this like so:

```
model Cohort {  
  id          Int          @id @default(autoincrement())  
  cohortName  String?      @unique  
  users       User[]  
}
```

Create an empty migration

In order to fix the data in the production database we must create a migration script. We can create an empty migration file by running the following command:

```
npx prisma migrate dev --create-only
```

The file will be added to the `prisma/migrations` folder.

Updating your data

Once we have a migration file, we can now update the data to meet the new migration. The SQL required will be specific to the problem at hand

For example, if we need to update the field `cohort.cohortName` to no longer be nullable, yet we've already have entities with `cohort.cohortName` as null - we could create a migration script like so:

```
UPDATE "Cohort"  
SET "cohortName" = id::text  
WHERE "cohortName" = NULL
```

We can then test this locally by running the command:

```
npx prisma migrate dev
```

Remove the nullable

Now that we have updated all of the cohorts to have a name we can remove the nullable rule on the `cohortName` field. So our schema will now look like this:

```
model Cohort {  
  id          Int          @id @default(autoincrement())  
  cohortName  String       @unique  
  users       User[]  
}
```