

# Translation

To support multiple languages (e.g., Arabic and English) for the Article and Subtitle tables in your Prisma schema, you can create a separate Translation table for storing translations. This approach normalizes the schema and allows you to add translations for any language without modifying the schema.

Updated Prisma Schema Here's how you can modify your schema to support multi-language translations:

```

model Article {
  id          String          @id @default(cuid())
  createdAt   DateTime         @default(now())
  updatedAt   DateTime         @updatedAt
  translations ArticleTranslation[]
  subtitles   Subtitle[]
}

model ArticleTranslation {
  id          String  @id @default(cuid())
  articleId   String
  language    String  /// e.g., 'en', 'ar'
  title       String  ///@zod.string.min(3).max(60)
  mainParagraph String  ///@zod.string.min(60).max(1000)

  Article     Article  @relation(fields: [articleId], references: [id], onDelete: Cascade)

  @@unique([articleId, language]) // Ensure no duplicate translations for the same article and :
}

model Subtitle {
  id          String          @id @default(cuid())
  createdAt   DateTime         @default(now())
  updatedAt   DateTime         @updatedAt
  translations SubtitleTranslation[]
  articleId   String
  Article     Article          @relation(fields: [articleId], references: [id])
}

model SubtitleTranslation {
  id          String  @id @default(cuid())
  subtitleId  String
  language    String  /// e.g., 'en', 'ar'
  title       String  ///@zod.string.min(3).max(60)
  paragraph   String  ///@zod.string.min(60).max(500)

  Subtitle    Subtitle @relation(fields: [subtitleId], references: [id], onDelete: Cascade)

  @@unique([subtitleId, language]) // Ensure no duplicate translations for the same subtitle and :
}

```

# Explanation of Changes

## 1. ArticleTranslation Table:

- Stores translations for the Article table.
- Includes fields for language, title, and mainParagraph.
- Uses a foreign key (articleId) to link to the Article table.
- Ensures unique translations per language using `@@unique([articleId, language])`.

## 2. SubtitleTranslation Table:

- Stores translations for the Subtitle table.
- Includes fields for language, title, and paragraph.
- Uses a foreign key (subtitleId) to link to the Subtitle table.
- Ensures unique translations per language using `@@unique([subtitleId, language])`.

# Querying Data

To fetch translations for a specific language:

```
const articleWithTranslations = await prisma.article.findMany({
  include: {
    translations: {
      where: { language: 'ar' }, // Fetch Arabic translations
    },
    subtitles: {
      include: {
        translations: {
          where: { language: 'ar' }, // Fetch Arabic translations
        },
      },
    },
  },
})

console.log(articleWithTranslations)
```

# Benefits of This Approach

1. Scalability: Easily add support for more languages without modifying the schema.
2. Normalization: Avoids duplicating data by separating translations into their own tables.

3. Flexibility: Allows querying specific languages or all translations as needed.