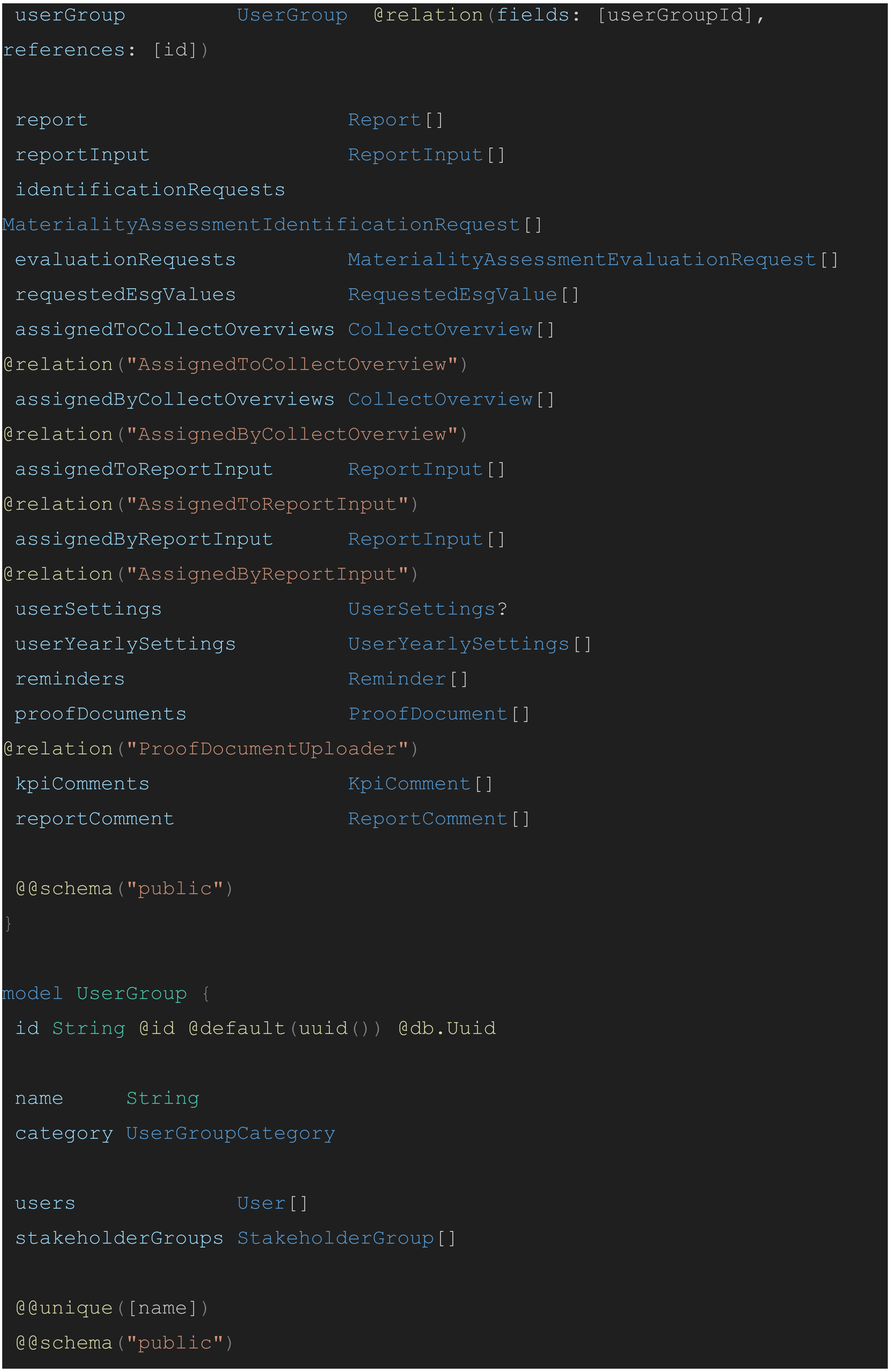
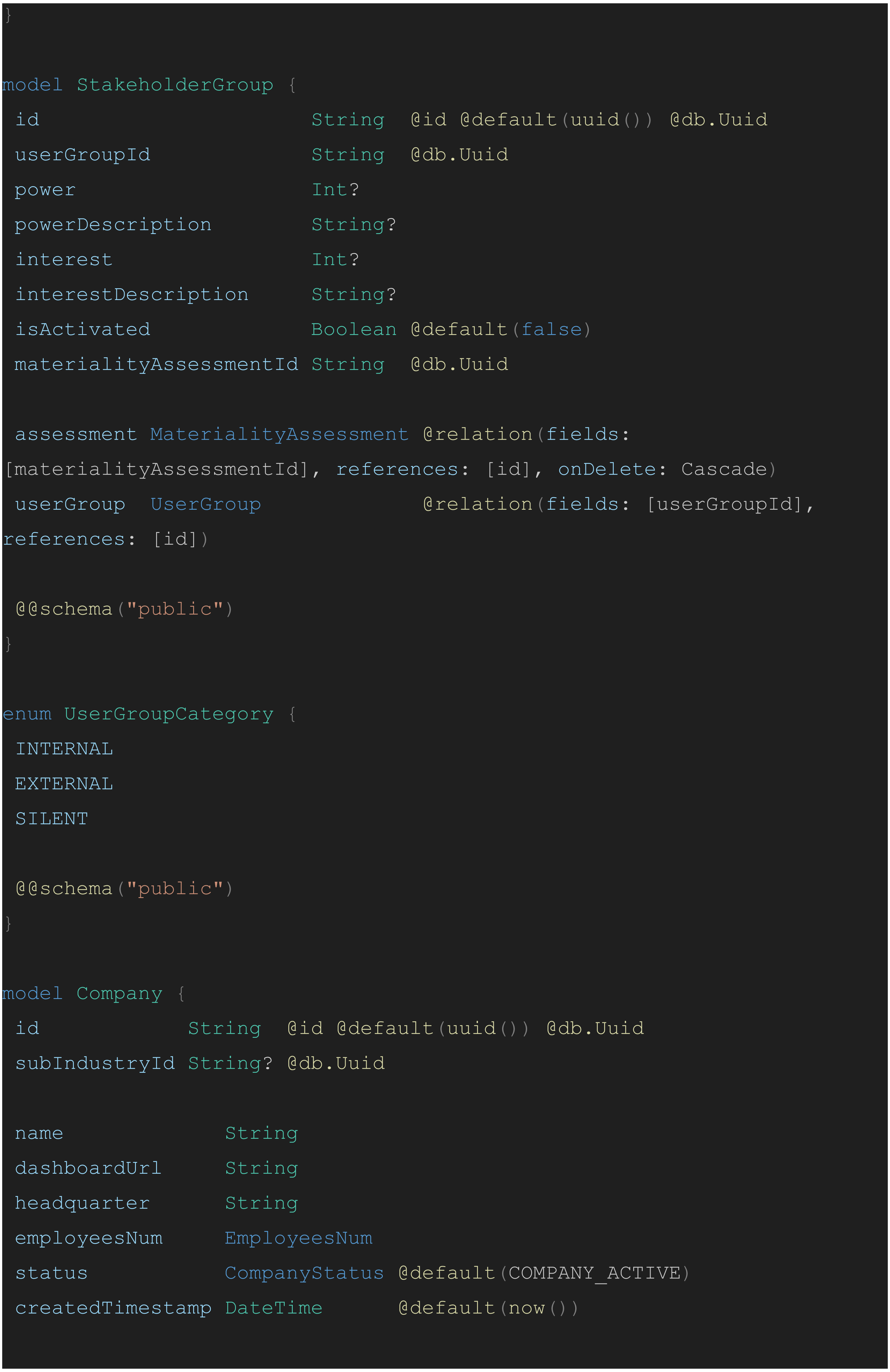
```
generator client {
 provider        = "prisma-client-js"
 output          = "../.gen/prisma"
 previewFeatures = ["multiSchema"]
}

datasource db {
 provider = "postgresql"
 url      = env("DATABASE_URL")
 schemas  = ["public", "audit"]
}

model User {
 id              String  @id @default(uuid()) @db.Uuid
 firebaseId      String? @unique
 companyId       String  @db.Uuid
 invitedByUserId String? @db.Uuid
 userGroupId     String  @db.Uuid

 email        String             @unique
 firstName    String?
 lastName     String?
 selectedYear Int?
 role         UserRole
 status       UserAccountStatus?
 department   UserDepartment     @default(OTHER)

 createdTimestamp        DateTime     @default(now())
 language                LanguageCode @default(de)
 numberFormat            NumberFormat @default(COMMA_PERIOD)
 chatbotExamplesExpanded Boolean      @default(true)
 activated               Boolean      @default(true)

 company           Company    @relation(fields: [companyId],
references: [id], onDelete: Cascade)
 invitedByUser     User?      @relation("InvitedUser", fields:
[invitedByUserId], references: [id], onDelete: SetNull)
 invitedUsers      User[]     @relation("InvitedUser")
 createdFacilities Facility[] @relation("CreatedFacilities")
```

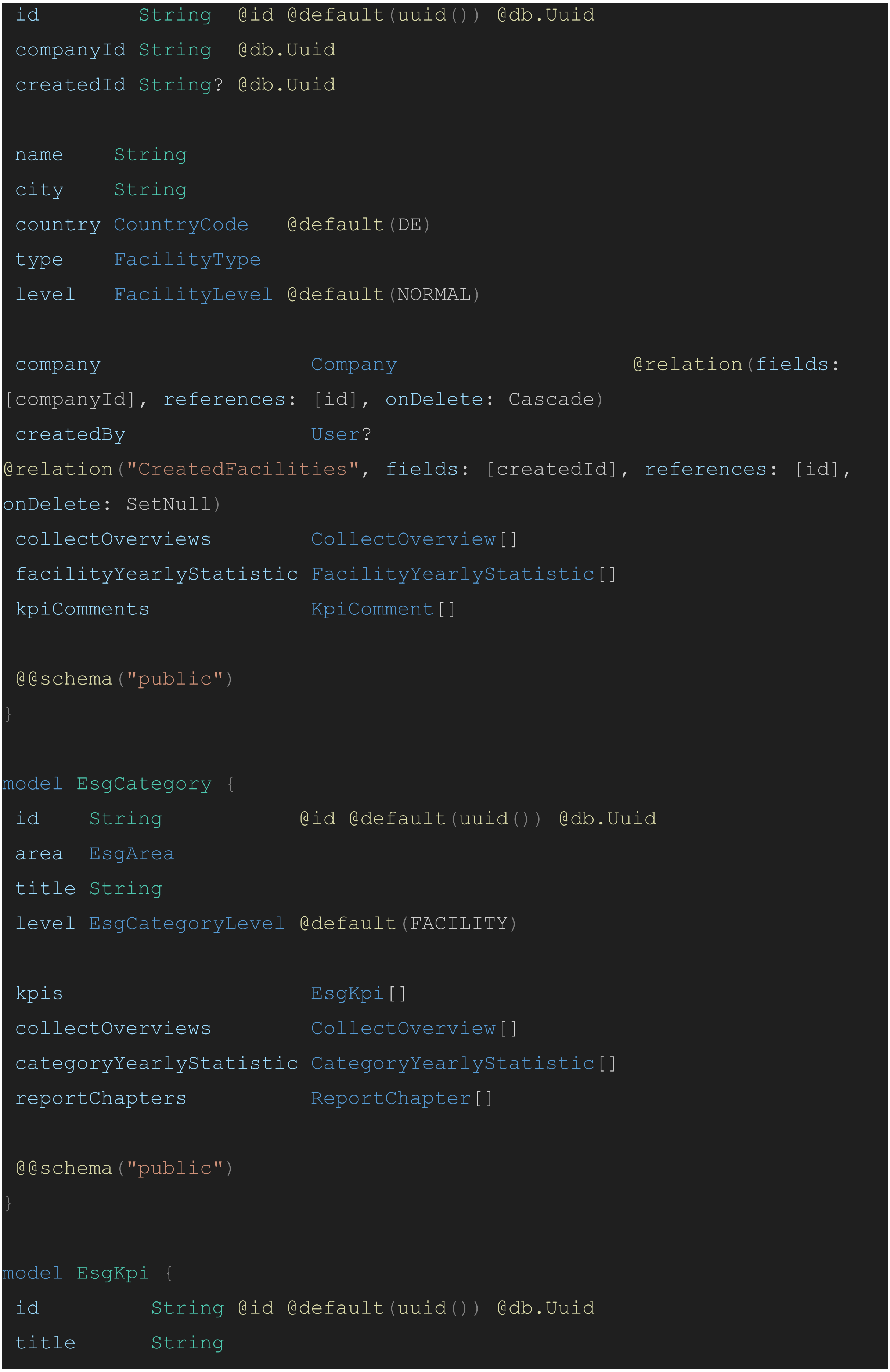


```
 userGroup         UserGroup  @relation(fields: [userGroupId],
references: [id])

 report                      Report[]
 reportInput                 ReportInput[]
 identificationRequests
MaterialityAssessmentIdentificationRequest[]
 evaluationRequests          MaterialityAssessmentEvaluationRequest[]
 requestedEsgValues          RequestedEsgValue[]
 assignedToCollectOverviews CollectOverview[]
@relation("AssignedToCollectOverview")
 assignedByCollectOverviews CollectOverview[]
@relation("AssignedByCollectOverview")
 assignedToReportInput       ReportInput[]
@relation("AssignedToReportInput")
 assignedByReportInput       ReportInput[]
@relation("AssignedByReportInput")
 userSettings                UserSettings?
 userYearlySettings          UserYearlySettings[]
 reminders                   Reminder[]
 proofDocuments              ProofDocument[]
@relation("ProofDocumentUploader")
 kpiComments                 KpiComment[]
 reportComment               ReportComment[]

 @@schema("public")
}

model UserGroup {
 id String @id @default(uuid()) @db.Uuid

 name     String
 category UserGroupCategory

 users             User[]
 stakeholderGroups StakeholderGroup[]

 @@unique([name])
 @@schema("public")
```



```
}

model StakeholderGroup {
  id                        String  @id @default(uuid()) @db.Uuid
  userGroupId               String  @db.Uuid
  power                     Int?
  powerDescription          String?
  interest                  Int?
  interestDescription       String?
  isActivated               Boolean @default(false)
  materialityAssessmentId String  @db.Uuid

  assessment MaterialityAssessment @relation(fields: [materialityAssessmentId], references: [id], onDelete: Cascade)
  userGroup  UserGroup             @relation(fields: [userGroupId], references: [id])

  @@schema("public")
}

enum UserGroupCategory {
  INTERNAL
  EXTERNAL
  SILENT

  @@schema("public")
}

model Company {
  id            String  @id @default(uuid()) @db.Uuid
  subIndustryId String? @db.Uuid

  name             String
  dashboardUrl     String
  headquarter      String
  employeesNum     EmployeesNum
  status           CompanyStatus @default(COMPANY_ACTIVE)
  createdTimestamp DateTime      @default(now())
```

```
 subIndustry                     SubIndustry?
@relation(fields: [subIndustryId], references: [id], onDelete:
Restrict)
 esgScore                        EsgScore?
 users                           User[]
 facilities                      Facility[]
 Report                          Report[]
 yearlystatistics                CompanyYearlyStatistic[]
 categoryYearlyStatistic         CategoryYearlyStatistic[]
 companyEsgKpiYearlyStatistic CompanyEsgKpiYearlyStatistic[]
 materialityAssessments          MaterialityAssessment[]

 @@schema("public")
}

model Industry {
 id String @id @default(uuid()) @db.Uuid

 name String

 subIndustries SubIndustry[]

 @@schema("public")
}

model SubIndustry {
 id          String @id @default(uuid()) @db.Uuid
 industryId String @db.Uuid

 name String

 industry  Industry  @relation(fields: [industryId], references: [id],
onDelete: Cascade)
 companies Company[]

 @@schema("public")
}

model Facility {
```



```
 id         String  @id @default(uuid()) @db.Uuid
 companyId String  @db.Uuid
 createdId String? @db.Uuid

 name    String
 city    String
 country CountryCode   @default(DE)
 type    FacilityType
 level   FacilityLevel @default(NORMAL)

 company                 Company                   @relation(fields:
[companyId], references: [id], onDelete: Cascade)
 createdBy               User?
@relation("CreatedFacilities", fields: [createdId], references: [id],
onDelete: SetNull)
 collectOverviews        CollectOverview[]
 facilityYearlyStatistic FacilityYearlyStatistic[]
 kpiComments             KpiComment[]

 @@schema("public")
}

model EsgCategory {
 id    String           @id @default(uuid()) @db.Uuid
 area  EsgArea
 title String
 level EsgCategoryLevel @default(FACILITY)

 kpis                    EsgKpi[]
 collectOverviews        CollectOverview[]
 categoryYearlyStatistic CategoryYearlyStatistic[]
 reportChapters          ReportChapter[]

 @@schema("public")
}

model EsgKpi {
 id          String @id @default(uuid()) @db.Uuid
 title       String
 categoryId String @db.Uuid

 description String?
 source      String?
 units       Unit[]
 isMandatory Boolean @default(false)
 type        KpiType @default(QUANTITATIVE)

 category                      EsgCategory
@relation(fields: [categoryId], references: [id])
 esgValues                     EsgValue[]
 CompanyEsgKpiYearlyStatistic CompanyEsgKpiYearlyStatistic[]
 frameworkNumbers              FrameworkNumber[]
 kpiComments                   KpiComment[]
 emissionFactor                EmissionFactor[]

 @@schema("public")
}

model EmissionFactor {
 id          String  @id @default(uuid()) @db.Uuid
 esgKpiId    String  @db.Uuid
 climatiqId String? @unique @db.Uuid
 activityId String?
 factor      Float
 unit        Unit
 source      String
 region      String
 year        Int

 esgValues EmissionFactorEsgValues[]
 esgKpi    EsgKpi                    @relation(fields: [esgKpiId],
references: [id])

 @@unique([esgKpiId, region, year])
 @@schema("public")
}

model EmissionFactorEsgValues {
 emissionFactorId String @db.Uuid
 esgValueId       String @unique @db.Uuid

 emissionFactor EmissionFactor @relation(fields: [emissionFactorId],
references: [id])
 esgValue       EsgValue       @relation(fields: [esgValueId],
references: [id], onDelete: Cascade)

 @@id([emissionFactorId, esgValueId])
 @@schema("public")
}

model FrameworkNumber {
 id               String  @id @default(uuid()) @db.Uuid
 reportSectionId String? @db.Uuid

 title     String              @unique
 framework ReportingFramework

 esgKpis       EsgKpi[]
 reportSection ReportSection? @relation(fields: [reportSectionId],
references: [id])

 @@schema("public")
}

model CollectOverview {
 id          String @id @default(uuid()) @db.Uuid
 facilityId String @db.Uuid
 categoryId String @db.Uuid
 year       Int    @default(2022)

 assignedToId      String?              @db.Uuid
 assignedById      String?              @db.Uuid
 progress          Int                  @default(0)
 requestedProgress Int?                 @default(0)
 dueDate           DateTime?
 status            CollectOverviewStatus @default(NOT_STARTED)
 facility   Facility    @relation(fields: [facilityId], references: [id], onDelete: Cascade)
 category   EsgCategory @relation(fields: [categoryId], references: [id])
 assignedTo User?       @relation("AssignedToCollectOverview", fields: [assignedToId], references: [id], onDelete: SetNull)
 assignedBy User?       @relation("AssignedByCollectOverview", fields: [assignedById], references: [id], onDelete: SetNull)
 esgValues  EsgValue[]
 reminders  Reminder[]

 @@unique([facilityId, categoryId, year])
 @@schema("public")
}

model EsgValue {
 id                String @id @default(uuid()) @db.Uuid
 esgKpiId          String @db.Uuid
 collectOverviewId String @db.Uuid

 value            String?
 selectedUnit     Unit?
 createdTimestamp DateTime @default(now())

 kpi                EsgKpi                    @relation(fields: [esgKpiId], references: [id])
 collectOverview    CollectOverview           @relation(fields: [collectOverviewId], references: [id], onDelete: Cascade)
 requestedEsgValues RequestedEsgValue[]
 emissionFactors    EmissionFactorEsgValues[]
 proofDocuments     ProofDocument[]

 @@unique([collectOverviewId, esgKpiId])
 @@schema("public")
}

model RequestedEsgValue {
 id         String  @id @default(uuid()) @db.Uuid
 esgValueId String  @db.Uuid
 userId     String? @db.Uuid

 value          String?
 selectedUnit   Unit?
 approvalStatus ApprovalStatus @default(OPEN)

 esgValue EsgValue @relation(fields: [esgValueId], references: [id],
onDelete: Cascade)
 user     User?    @relation(fields: [userId], references: [id],
onDelete: SetNull)

 @@schema("public")
}

model UserSettings {
 id                                 String       @id @default(uuid())
@db.Uuid
 userId                             String       @unique @db.Uuid
 statuses                           String?
 facilities                         String?
 categories                         String?
 assignedTo                         String?
 rowsPerPageCount                   Int?         @default(5)
 orderField                         String?
@default("facilityName")
 orderDirection                     SortingOrder? @default(ASC)
 global_statuses                    String?
 global_categories                  String?
 global_assignedTo                  String?
 global_rowsPerPageCount            Int?
 global_orderField                  String?
 global_orderDirection              SortingOrder?
 materialTopicsOrderField           String?      @default("status")
 stakeholderSelectionOrderField     String?
@default("categoryLabel")
 materialTopicsOrderDirection       SortingOrder? @default(ASC)
 stakeholderSelectionOrderDirection SortingOrder? @default(ASC)
 user User @relation(fields: [userId], references: [id], onDelete:
Cascade)

 @@schema("public")
}

model UserYearlySettings {
 userId             String  @unique @db.Uuid
 year               Int
 manageKpiFrameworks String?

 user User @relation(fields: [userId], references: [id], onDelete:
Cascade)

 @@schema("public")
}

model Reminder {
 id            String         @id @default(uuid()) @db.Uuid
 createDate    DateTime       @default(now())
 executionDate DateTime
 status        ReminderStatus @default(EXECUTED)
 recipientId   String         @db.Uuid

 recipient        User             @relation(fields: [recipientId],
references: [id], onDelete: Cascade)
 collectOverviews CollectOverview[]

 @@unique([recipientId, executionDate])
 @@schema("public")
}

model EsgScore {
 id        String @id @default(uuid()) @db.Uuid
 companyId String @unique @db.Uuid

 environmentScore Int?
 socialScore      Int?
 governanceScore  Int?
```

```
  company Company @relation(fields: [companyId], references: [id],
onDelete: Cascade)

  @@schema("public")
}

model MaterialTopic {
  id                String              @id @default(uuid()) @db.Uuid
  reportChapterId String?             @db.Uuid
  title             String
  area              EsgArea
  type              MaterialTopicType @default(OPTIONAL)

  materialityAssessmentTopics MaterialityAssessmentTopic[]
  reportChapter                ReportChapter?
@relation(fields: [reportChapterId], references: [id])

  @@schema("public")
}

model MaterialityAssessment {
  id                                String  @id @default(uuid()) @db.Uuid
  companyId                         String  @db.Uuid
  year                              Int
  identificationTypeformUrl         String?
  evaluationTypeformUrl             String?
  identificationTypeformUrlEnglish String?

  company                   Company
@relation(fields: [companyId], references: [id], onDelete: Cascade)
  topics                    MaterialityAssessmentTopic[]
  identificationrequests MaterialityAssessmentIdentificationRequest[]
  evaluationrequests      MaterialityAssessmentEvaluationRequest[]
  stakeholderGroups       StakeholderGroup[]

  @@unique([companyId, year])
  @@schema("public")
}
```

```
model MaterialityAssessmentTopic {
 id                      String   @id @default(uuid()) @db.Uuid
 materialityAssessmentId String   @db.Uuid
 materialTopicId         String   @db.Uuid
 createdDate             DateTime
 impactDescription       String?
 financialDescription    String?
 impactScale             Int?
 impactScope             Int?
 impactIrreversibility   Int?
 financialSignificance   Int?
 financialProbability    Int?

 assessment        MaterialityAssessment
@relation(fields: [materialityAssessmentId], references: [id],
onDelete: Cascade)
 topic             MaterialTopic
@relation(fields: [materialTopicId], references: [id])
 evaluationRequest MaterialityAssessmentEvaluationRequest[]
 impact            Impact[]
 riskOpportunity   RiskOpportunity[]

 @@schema("public")
}

model MaterialityAssessmentIdentificationRequest {
 id                      String   @id @default(uuid()) @db.Uuid
 stakeholderId           String   @db.Uuid
 materialityAssessmentId String   @db.Uuid
 dueDate                 DateTime

 stakeholder User                  @relation(fields: [stakeholderId],
references: [id], onDelete: Cascade)
 assessment  MaterialityAssessment @relation(fields:
[materialityAssessmentId], references: [id], onDelete: Cascade)

 @@unique([stakeholderId, materialityAssessmentId])
 @@schema("public")
```

```
}

model MaterialityAssessmentEvaluationRequest {
 id                        String
@id @default(uuid()) @db.Uuid
 stakeholderId             String
@db.Uuid
 materialityAssessmentId String
@db.Uuid
 dueDate                   DateTime
 status                    MaterialityAssessmentEvaluationRequestStatus
@default(NOT_STARTED)

 stakeholder                   User
@relation(fields: [stakeholderId], references: [id], onDelete: Cascade)
 assessment                    MaterialityAssessment
@relation(fields: [materialityAssessmentId], references: [id],
onDelete: Cascade)
 materialityAssessmentTopics MaterialityAssessmentTopic[]
 impact                        Impact[]
 riskOpportunity               RiskOpportunity[]

 @@unique([stakeholderId, materialityAssessmentId])
 @@schema("public")
}

model Impact {
 id                                        String @id @default(uuid())
@db.Uuid
 materialityAssessmentTopicId              String @db.Uuid
 materialityAssessmentEvaluationRequestId String @db.Uuid

 createdDate          DateTime
 title                String
 description          String?
 type                 ImpactType
 occurenceType        OccurenceType
 uncontrolledDuration UncontrolledDurationType
 extentOfImpact       Int                          @default(0)
 degreeOfCoverage     Int                      @default(0)
 irreversibility      Int                      @default(0)
 occurenceProbability Int                      @default(0)
 isMaterial           Boolean                  @default(false)

 evaluationRequest          MaterialityAssessmentEvaluationRequest
@relation(fields: [materialityAssessmentEvaluationRequestId],
references: [id], onDelete: Cascade)
 materialityAssessmentTopic MaterialityAssessmentTopic
@relation(fields: [materialityAssessmentTopicId], references: [id])

 @@schema("public")
}

model RiskOpportunity {
 id                                       String @id @default(uuid())
@db.Uuid
 materialityAssessmentTopicId             String @db.Uuid
 materialityAssessmentEvaluationRequestId String @db.Uuid

 createdDate          DateTime
 title                String
 description          String?
 type                 RiskOpportunityType
 uncontrolledDuration UncontrolledDurationType
 significance         Int                      @default(0)
 probability          Int                      @default(0)
 isMaterial           Boolean                  @default(false)

 evaluationRequest          MaterialityAssessmentEvaluationRequest
@relation(fields: [materialityAssessmentEvaluationRequestId],
references: [id], onDelete: Cascade)
 materialityAssessmentTopic MaterialityAssessmentTopic
@relation(fields: [materialityAssessmentTopicId], references: [id])

 @@schema("public")
}

model Report {
 id          String  @id @default(uuid()) @db.Uuid
 companyId   String  @db.Uuid
 createdById String? @db.Uuid

 title             String
 year              Int
 framework         ReportingFramework
 createdTimestamp DateTime           @default(now())
 progress          Int                @default(0)

 company             Company             @relation(fields: [companyId],
references: [id], onDelete: Cascade)
 createdBy           User?               @relation(fields:
[createdById], references: [id], onDelete: SetNull)
 reportInput         ReportInput[]
 aIGeneratedReports AIGeneratedReport[]
 reportComment       ReportComment[]

 @@schema("public")
}

model ReportInput {
 id               String  @id @default(uuid()) @db.Uuid
 reportId         String  @db.Uuid
 createdById      String? @db.Uuid
 reportSectionId String  @db.Uuid

 input              String?
 createdTimestamp  DateTime @default(now())
 isGenerating       Boolean? @default(false)
 isMarkedCompleted Boolean  @default(false)
 assignedToId       String?  @db.Uuid
 assignedById       String?  @db.Uuid

 createdBy      User?          @relation(fields: [createdById],
references: [id], onDelete: SetNull)
 report         Report         @relation(fields: [reportId], references:
[id], onDelete: Cascade)
```

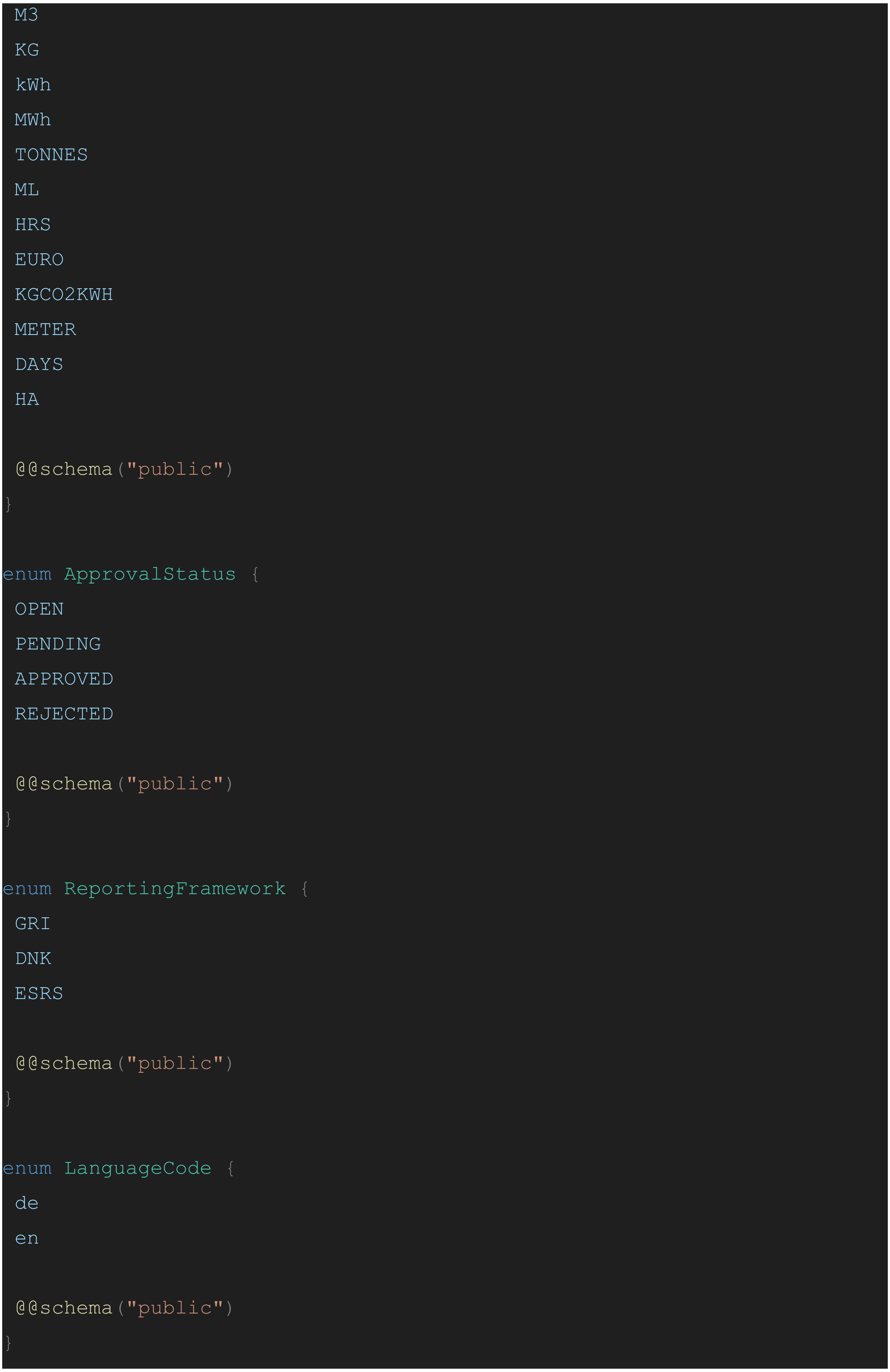
```
 reportSection ReportSection @relation(fields: [reportSectionId], references: [id])

 assignedTo User? @relation("AssignedToReportInput", fields: [assignedToId], references: [id], onDelete: SetNull)
 assignedBy User? @relation("AssignedByReportInput", fields: [assignedById], references: [id], onDelete: SetNull)

 @@schema("public")
}

model AIGeneratedReport {
 id              String       @id @default(uuid()) @db.Uuid
 reportId        String       @db.Uuid
 reportSectionId String       @db.Uuid
 prompt          String
 response        String
 modelOptions    String
 language        LanguageCode @default(de)

 createdTimestamp DateTime  @default(now())
 updatedTimestamp DateTime? @default(now())

 report        Report        @relation(fields: [reportId], references: [id], onDelete: Cascade)
 reportSection ReportSection @relation(fields: [reportSectionId], references: [id], onDelete: Cascade)

 @@unique([reportId, reportSectionId, language])
 @@schema("public")
}

model ReportSection {
 id String @id @default(uuid()) @db.Uuid

 reportChapterId       String @db.Uuid
 title                 String
 mandatoryRequirements Int    @default(0)
 reportInput        ReportInput[]
 frameworkNumbers   FrameworkNumber[]
 aIGeneratedReport  AIGeneratedReport[]
 reportComment      ReportComment[]
 reportChapter      ReportChapter     @relation(fields:
[reportChapterId], references: [id])

 @@schema("public")
}

model ReportChapter {
 id             String              @id @default(uuid()) @db.Uuid
 title          String
 framework      ReportingFramework
 label          String?
  materialTopics MaterialTopic[]
 reportSections ReportSection[]
 esgCategories EsgCategory[]

 @@schema("public")
}

model ReportComment {
 id               String  @id @default(uuid()) @db.Uuid
 reportSectionId String  @db.Uuid
 reportId         String  @db.Uuid
 comment          String
 userId           String? @db.Uuid

 createdTimestamp DateTime @default(now())

 user           User?         @relation(fields: [userId], references:
[id], onDelete: SetNull)
 reportSection ReportSection @relation(fields: [reportSectionId],
references: [id], onDelete: Cascade)
 report         Report        @relation(fields: [reportId], references:
[id], onDelete: Cascade)

 @@schema("public")
```

```
}

model CompanyYearlyStatistic {
 id                             String @id @default(uuid()) @db.Uuid
 companyId                      String @db.Uuid
 year                           Int
 directEmissions                Float?
 emissionIntensity              Float?
 environmentalEsgProgress       Float?
 indirectEmissions              Float?
 totalNonRenewableEnergy        Float?
 totalRenewableEnergy           Float?
 employeeTurnoverRate           Float?
 totalFemaleManagerEmployees    Int?
 totalFemaleNonManagerEmployees Int?
 totalMaleManagerEmployees      Int?
 totalMaleNonManagerEmployees   Int?
 numberOccupationalAccidents    Int?
 percentageFemaleEmployees      Float?
 trainingHoursPerEmployee       Float?
 overallEsgProgress             Int?
 completedFacilities            Int?
 totalFacilities                Int?
 numberOfCompletedEsgValues     Int?
 totalEnergyConsumption         Float?
 totalGeneratedRevenue          Float?

 company Company @relation(fields: [companyId], references: [id],
onDelete: Cascade)

 @@unique([companyId, year])
 @@schema("public")
}

model CompanyEsgKpiYearlyStatistic {
 id            String @id @default(uuid()) @db.Uuid
 companyId     String @db.Uuid
 esgKpiId      String @db.Uuid
 year          Int
 value        Float?
 selectedUnit Unit?

 company Company @relation(fields: [companyId], references: [id],
onDelete: Cascade)
 esgKpi  EsgKpi  @relation(fields: [esgKpiId], references: [id],
onDelete: Cascade)

 @@unique([companyId, year, esgKpiId])
 @@schema("public")
}

model FacilityYearlyStatistic {
 id                                   String @id @default(uuid())
@db.Uuid
 facilityId                           String @db.Uuid
 year                                 Int
 environmentProgress                  Int?
 socialProgress                       Int?
 governanceProgress                   Int?
 gasConsumption                       Float?
 energyConsumption                    Float?
 waterConsumption                     Float?
 totalFemaleEmployees                 Int?
 totalMaleEmployees                   Int?
 electricityEmissions                 Float?
 totalGeneratedRevenue                Float?
 totalWasteGeneration                 Float?
 totalPurchasedElectricityConsumption Float?
 totalPurchasedHeatingConsumption     Float?
 totalHeatingOilConsumption           Float?
 totalBiomassConsumption              Float?
 totalBiogasConsumption               Float?
 totalCoalConsumption                 Float?
 totalGeothermalConsumption           Float?
 totalHydroelectricConsumption        Float?
 totalHydrogenConsumption             Float?
 totalSolarConsumption                Float?
 totalWasteHeatRecoveryConsumption    Float?
```

```
  totalWindConsumption                 Float?
  totalNaturalGasConsumption           Float?
  totalWoodChipsConsumption            Float?
  totalWoodPelletsConsumption          Float?
  directEmissions                      Float?
  indirectEmissions                    Float?

  facility Facility @relation(fields: [facilityId], references: [id], onDelete: Cascade)

  @@unique([facilityId, year])
  @@schema("public")
}

model CategoryYearlyStatistic {
  companyId  String @db.Uuid
  categoryId String @db.Uuid

  year     Int
  progress Int @default(0)

  company  Company     @relation(fields: [companyId], references: [id], onDelete: Cascade)
  category EsgCategory @relation(fields: [categoryId], references: [id], onDelete: Cascade)

  @@id([companyId, categoryId, year])
  @@schema("public")
}

model CompanyYear {
  id        String @id @default(uuid()) @db.Uuid
  companyId String @db.Uuid
  year      Int

  @@unique([companyId, year])
  @@schema("public")
}
```

```
model ProofDocument {
 id             String   @id @default(uuid()) @db.Uuid
 name           String
 type           String
 encoding       String
 uploadDate     DateTime
 uploaderId     String?  @db.Uuid
 url            String
 bucketFileName String

 uploader  User?      @relation("ProofDocumentUploader", fields:
[uploaderId], references: [id], onDelete: SetNull)
 esgValues EsgValue[]

 @@schema("public")
}

model KpiComment {
 id          String  @id @default(uuid()) @db.Uuid
 esgKpiId    String  @db.Uuid
 facilityId  String  @db.Uuid
 comment     String
 userId      String? @db.Uuid

 year             Int
 createdTimestamp DateTime @default(now())

 esgKpi   EsgKpi   @relation(fields: [esgKpiId], references: [id],
onDelete: Cascade)
 user     User?    @relation(fields: [userId], references: [id],
onDelete: SetNull)
 facility Facility @relation(fields: [facilityId], references: [id],
onDelete: Cascade)

 @@schema("public")
}

model AuditLog {
 id               String   @id @default(uuid()) @db.Uuid
 model            String
 operation        String
 recordId         String
 editorId         String
 companyId        String
 changedValue     Json
 args             Json?
 changeTimestamp DateTime
 persistedIn      String

 @@index(fields: [companyId, persistedIn], name:
"companyId_persistedIn_index")
 @@schema("audit")
}

enum CompanyStatus {
 COMPANY_CREATED
 COMPANY_INFO_COMPLETED
 COMPANY_ACTIVE
 COMPANY_DEACTIVATED

 @@schema("public")
}

enum UserAccountStatus {
 ACCOUNT_CREATED
 PENDING_INVITES
 SEND_REMINDER
 NEW_PASSWORD_SET
 USER_INFO_COMPLETED
 CONTACT_STORED

 @@schema("public")
}

enum UserRole {
 SUPER
 ADMIN
 LEAD
 EMPLOYEE
 EXTERNAL
 STAKEHOLDER

 @@schema("public")
}

enum UserDepartment {
 HUMAN_RESOURCE
 PRODUCTION
 LOGISTICS
 CUSTOMER_SERVICE
 RESEARCH_DEVELOPMENT
 SUSTAINABILITY
 FINANCE_CONTROLLING
 EXECUTIVE_MANAGEMENT
 SALES
 OPERATIONS
 QUALITY_MANAGEMENT
 PURCHASING
 MARKETING
 INFORMATION_TECHNOLOGY
 TECHNICAL_MAINTENANCE
 ENVIRONMENTAL_PROTECTION_OCCUPATIONAL_SAFETY
 PROJECT_MANAGEMENT
 OTHER

 @@schema("public")
}

enum EmployeesNum {
 N_1_10       @map("1-10")
 N_11_50      @map("11-50")
 N_51_250     @map("51-250")
 N_251_500    @map("251-500")
 N_501_750    @map("501-750")
 N_751_1000   @map("751-1000")
 N_1001_10000 @map("1001-10000")
```

```
  @@schema("public")
}

enum SortingOrder {
  ASC
  DESC

  @@schema("public")
}

enum FacilityType {
  MANUFACTURING
  WAREHOUSE
  OFFICE
  SERVICE
  LOGISTICS

  @@schema("public")
}

// TODO remove FacilityType and rename FacilityLevel to FacilityType
enum FacilityLevel {
  NORMAL
  GLOBAL

  @@schema("public")
}

enum EsgArea {
  ENVIRONMENT
  SOCIAL
  GOVERNANCE

  @@schema("public")
}

enum Unit {
  LITRE
  PERCENTAGE
```



```
  M3
  KG
  kWh
  MWh
  TONNES
  ML
  HRS
  EURO
  KGCO2KWH
  METER
  DAYS
  HA

  @@schema("public")
}

enum ApprovalStatus {
  OPEN
  PENDING
  APPROVED
  REJECTED

  @@schema("public")
}

enum ReportingFramework {
  GRI
  DNK
  ESRS

  @@schema("public")
}

enum LanguageCode {
  de
  en

  @@schema("public")
}
```

```
enum NumberFormat {
 COMMA_PERIOD
 PERIOD_COMMA

 @@schema("public")
}

enum CollectOverviewStatus {
 NOT_STARTED
 IN_PROGRESS
 COMPLETED
 OVERDUE
 INACTIVE

 @@schema("public")
}

enum CountryCode {
 AD
 AE
 AF
 AG
 AI
 AL
 AM
 AO
 AQ
 AR
 AS
 AT
 AU
 AW
 AX
 AZ
 BA
 BB
 BD
 BE
```

```
BF
BG
BH
BI
BJ
BL
BM
BN
BO
BR
BS
BT
BV
BW
BY
BZ
CA
CC
CD
CF
CG
CH
CI
CK
CL
CM
CN
CO
CR
CU
CV
CW
CX
CY
CZ
DE
DJ
DK
DM
```

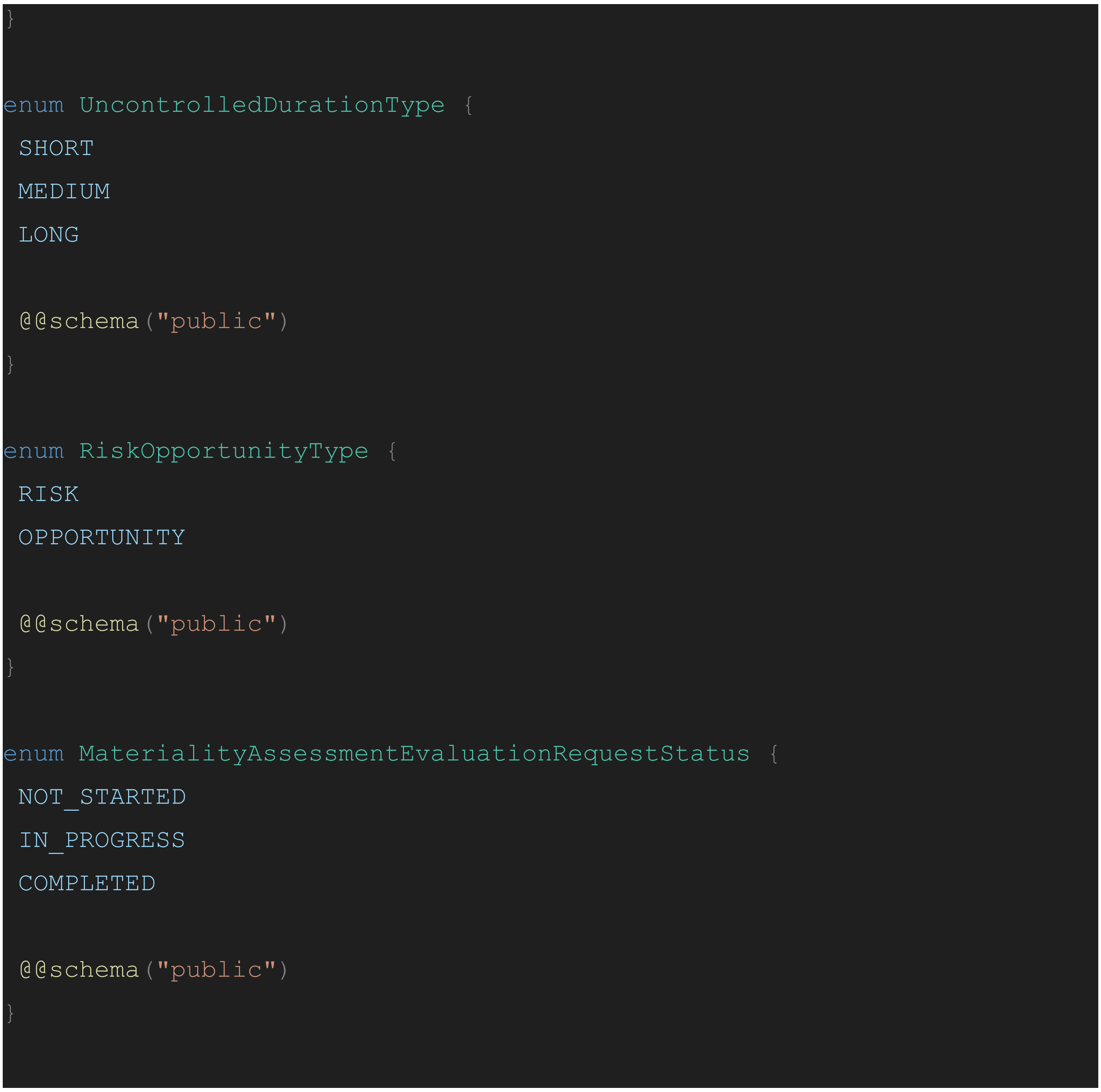
```
DO
DZ
EC
EE
EG
EH
ER
ES
ET
FI
FJ
FK
FM
FO
FR
GA
GB
GD
GE
GF
GG
GH
GI
GL
GM
GN
GP
GQ
GR
GS
GT
GU
GW
GY
HK
HM
HN
HR
HT
```

```
HU
ID
IE
IL
IM
IN
IO
IQ
IR
IS
IT
JE
JM
JO
JP
KE
KG
KH
KI
KM
KN
KP
KR
KW
KY
KZ
LA
LB
LC
LI
LK
LR
LS
LT
LU
LV
LY
MA
MC
```

```
MD
ME
MF
MG
MH
MK
ML
MM
MN
MO
MP
MQ
MR
MS
MT
MU
MV
MW
MX
MY
MZ
NA
NC
NE
NF
NG
NI
NL
NO
NP
NR
NU
NZ
OM
PA
PE
PF
PG
PH
```

```
PK
PL
PM
PN
PR
PS
PT
PW
PY
QA
RE
RO
RS
RU
RW
SA
SB
SC
SD
SE
SG
SH
SI
SJ
SK
SL
SM
SN
SO
SR
SS
ST
SV
SX
SY
SZ
TC
TD
TF
```

```
  TG
  TH
  TJ
  TK
  TL
  TM
  TN
  TO
  TR
  TT
  TV
  TW
  TZ
  UA
  UG
  US
  UY
  UZ
  VA
  VC
  VE
  VG
  VI
  VN
  VU
  WF
  WS
  XK
  YE
  YT
  ZA
  ZM
  ZW

  @@schema("public")
}

enum KpiType {
  NARRATIVE
  QUANTITATIVE

  @@schema("public")
}

enum MaterialTopicType {
  RECOMMENDED
  OPTIONAL

  @@schema("public")
}

enum ReminderStatus {
  CREATED
  EXECUTED
  SCHEDULED

  @@schema("public")
}

enum EsgCategoryLevel {
  FACILITY
  COMPANY

  @@schema("public")
}

enum ImpactType {
  POSITIVE
  NEGATIVE

  @@schema("public")
}

enum OccurenceType {
  ACTUAL
  POTENTIAL

  @@schema("public")
```



```
}

enum UncontrolledDurationType {
 SHORT
 MEDIUM
 LONG

 @@schema("public")
}

enum RiskOpportunityType {
 RISK
 OPPORTUNITY

 @@schema("public")
}

enum MaterialityAssessmentEvaluationRequestStatus {
 NOT_STARTED
 IN_PROGRESS
 COMPLETED

 @@schema("public")
}
```