Query ORM table untuk database miniproject

enum UserRole {

  CUSTOMER

  ORGANIZER

  ADMIN

}

enum TransactionStatus {

  waiting\_payment

  waiting\_confirm

  done

  rejected

  expired

  canceled

}

enum PointSource {

  referral

  refund

  redeem

}

model User {

  id              String   @id @default(uuid())

  full\_name       String

  email           String   @unique

  password        String

  role            UserRole

  referral\_code   String   @unique

  referred\_by     String? // simpan kode referral yang digunakan

  profile\_picture String?

  created\_at      DateTime @default(now())

  // relasi

  referralLogsSent     ReferralLog[] @relation("referrer")

  referralLogsReceived ReferralLog[] @relation("referee")

  events               Event[]

  transactions         Transaction[]

  points               Point[]

  coupons              Coupon[]

  reviews              Review[]

}

model ReferralLog {

  id          String   @id @default(uuid())

  referrer    User     @relation("referrer", fields: [referrer\_id], references: [id])

  referrer\_id String

  referee     User     @relation("referee", fields: [referee\_id], references: [id])

  referee\_id  String

  created\_at  DateTime @default(now())

}

model Event {

  id              String   @id @default(uuid())

  organizer       User     @relation(fields: [organizer\_id], references: [id])

  organizer\_id    String

  name            String

  description     String

  category        String

  location        String

  price           Int // 0 = gratis

  start\_date      DateTime

  end\_date        DateTime

  total\_seats     Int

  remaining\_seats Int

  created\_at      DateTime @default(now())

  vouchers     Voucher[]

  transactions Transaction[]

  reviews      Review[]

}

model Voucher {

  id              String   @id @default(uuid())

  event           Event    @relation(fields: [event\_id], references: [id])

  event\_id        String

  code            String   @unique

  discount\_amount Int

  start\_date      DateTime

  end\_date        DateTime

  created\_at      DateTime @default(now())

}

model Transaction {

  id              String            @id @default(uuid())

  user            User              @relation(fields: [user\_id], references: [id])

  user\_id         String

  event           Event             @relation(fields: [event\_id], references: [id])

  event\_id        String

  ticket\_quantity Int

  locked\_price    Int // harga saat checkout

  total\_price     Int // locked\_price \* ticket\_quantity

  status          TransactionStatus

  payment\_proof   String?

  created\_at      DateTime          @default(now())

  updated\_at      DateTime          @updatedAt

}

model Point {

  id         String      @id @default(uuid())

  user       User        @relation(fields: [user\_id], references: [id])

  user\_id    String

  amount     Int

  source     PointSource

  expired\_at DateTime

  created\_at DateTime    @default(now())

}

model Coupon {

  id              String   @id @default(uuid())

  user            User     @relation(fields: [user\_id], references: [id])

  user\_id         String

  code            String   @unique

  discount\_amount Int

  expired\_at      DateTime

  used            Boolean  @default(false)

  created\_at      DateTime @default(now())

}

model Review {

  id         String   @id @default(uuid())

  user       User     @relation(fields: [user\_id], references: [id])

  user\_id    String

  event      Event    @relation(fields: [event\_id], references: [id])

  event\_id   String

  rating     Int

  comment    String?

  created\_at DateTime @default(now())

}