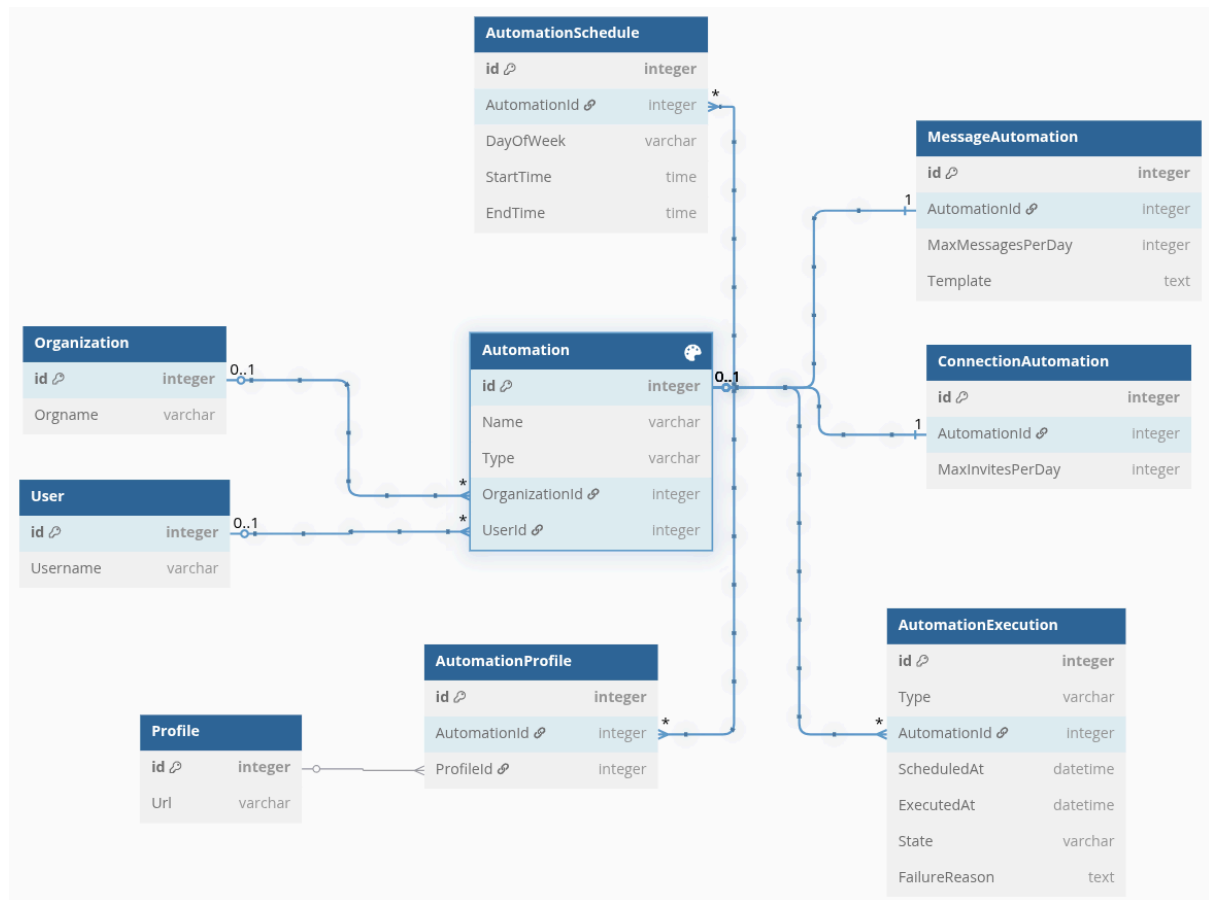


Diseño de base de datos. Ejercicio Squads Ventures



CÓDIGO dbml:

```

Table Automation {
    id integer [primary key]
    Name varchar
    Type varchar // "message" o "connection"
    OrganizationId integer [ref: > Organization.id]//Foreign key
    UserId integer [ref: > User.id]// Foreign key
}
    
```

```

Table Organization {
    id integer [primary key]
    Orgname varchar
}
    
```

```

Table User {
    id integer [primary key]
    Username varchar
}
    
```

```

Table AutomationSchedule {
  id integer [primary key]
  AutomationId integer [ref: > Automation.id]
  DayOfWeek varchar
  StartTime time
  EndTime time
}

```

```

Table MessageAutomation {
  id integer [primary key]
  AutomationId integer [unique, ref: - Automation.id]
  MaxMessagesPerDay integer
  Template text
}

```

```

Table ConnectionAutomation {
  id integer [primary key]
  AutomationId integer [unique, ref: - Automation.id]
  MaxInvitesPerDay integer
}

```

```

Table Profile {
  id integer [primary key]
  Url varchar [unique] // Para no repetir perfiles
}

```

```

Table AutomationProfile {
  id integer [primary key]
  AutomationId integer [ref: > Automation.id]
  ProfileId integer [ref: > Profile.id]
}

```

```

Table AutomationExecution {
  id integer [primary key]
  Type varchar //"message" o "connection"
  AutomationId integer [ref: > Automation.id]
  ScheduledAt datetime
  ExecutedAt datetime
  State varchar // SCHEDULED, SENT, FAILED, RECEIVED, ACCEPTED, REJECTED
  FailureReason text
}

```

Explicación:

Para realizar este trabajo plantee un diseño simple y bien normalizado que refleja los distintos componentes: automatizaciones, perfiles, configuraciones horarias y ejecuciones. La tabla **Automation** funciona como entidad principal, con referencias a **User** y

Organization. A partir de ella, se definen dos tipos específicos: **MessageAutomation** y **ConnectionAutomation**, permitiendo separar propiedades particulares sin mezclar responsabilidades.

Cada automatización puede tener múltiples perfiles asociados. Para esto, se creó una tabla de **Profile**, que contiene URLs únicas de perfiles de LinkedIn, y una tabla intermedia **AutomationProfile** que permite relacionar cada automatización con varios perfiles. Esta decisión permite reutilizar un mismo perfil en diferentes automatizaciones sin duplicar información, y facilita futuras operaciones como analizar qué automatizaciones comparten un mismo perfil.

Para la configuración horaria, **AutomationSchedule** guarda un día por fila, lo cual simplifica las consultas por día de la semana y evita estructuras difíciles de manipular.

Las ejecuciones se registran en **AutomationExecution**, incluyendo estados y posibles errores, lo que permite hacer seguimiento detallado del sistema.

Este enfoque busca simplicidad, extensibilidad y consultas eficientes, dejando la lógica dinámica al código para mantener la base limpia y bien estructurada.