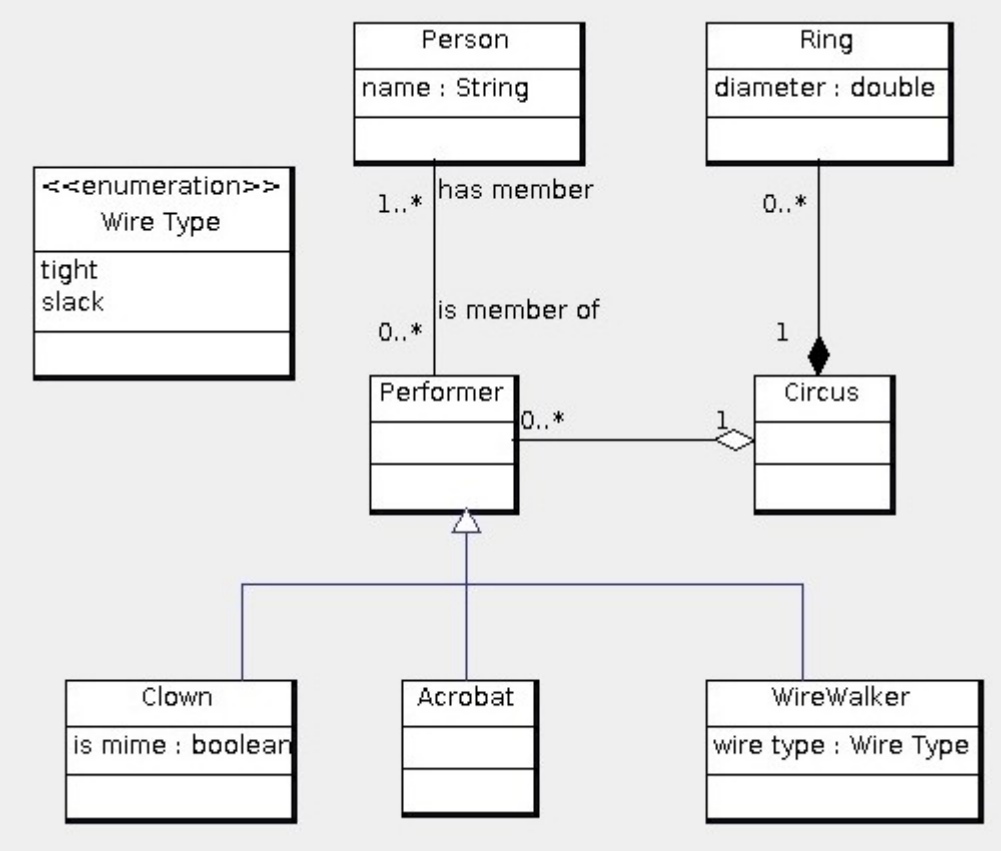
Circus Performer Translation Problem

Translate the following UML diagram to an SQL schema:



Translation to SQL

```
create table Person(
  id int primary key,
  name varchar(200) not null
);
create table Circus(
  id int primary key
);
create table Performer(
  id int primary key,
  partOf int not null,
  foreign key(partOf) references Circus(id)
    on update cascade on delete no action
);
create table Ring(
  id int primary key,
  diameter double not null,
  partOf int not null,
  foreign key(partOf) references Circus(id)
    on update cascade on delete cascade
);
```

Translation to SQL

```
create table Clown(
  id int primary key,
  foreign key(id) references Performer(id)
    on update cascade on delete cascade,
  -- Inheritance strategy: Joined strategy
  isMime boolean not null
);
create table Acrobat(
  id int primary key,
  foreign key(id) references Performer(id)
    on update cascade on delete cascade
  -- Inheritance strategy: Joined strategy
);
create table WireWalker(
  id int primary key,
  foreign key(id) references Performer(id)
    on update cascade on delete cascade,
  -- Inheritance strategy: Joined strategy
  wireType enum('tight', 'slack') not null
  -- Enumeration strategy: Column type
);
```

Translation to SQL

```
create table PerformPerson(
  hasMember int not null,
  foreign key(hasMember) references Person(id)
    on update cascade on delete cascade,
  isMemberOf int not null,
  foreign key(isMemberOf) references Performer(id)
    on update cascade on delete cascade,
  primary key(hasMember, isMemberOf)
);
alter table Performer
  add foreign key(id) references PerformPerson(isMemberOf)
    on update no action on delete no action;
```

Alternative Translation to SQL

```
create table Person(
  id int primary key,
  name varchar(200) not null
);
create table Circus(
  id int primary key
);
create table Performer(
  id int primary key,
  partOf int not null,
  foreign key(partOf) references Circus(id)
    on update cascade on delete no action
);
create table Ring(
  id int primary key,
  diameter double not null,
  partOf int not null,
  foreign key(partOf) references Circus(id)
    on update cascade on delete cascade
);
```

Alternative Translation to SQL

```
create table Clown(
  id int primary key,
  foreign key(id) references Performer(id)
    on update cascade on delete cascade,
  -- Inheritance strategy: Joined strategy
  isMime boolean not null
);
create table Acrobat(
  id int primary key,
  foreign key(id) references Performer(id)
    on update cascade on delete cascade
  -- Inheritance strategy: Joined strategy
);
create table WireType(
  id int primary key,
  type varchar(200) not null unique
);
insert into WireType(id, type) values(1,'tight');
insert into WireType(id, type) values(2,'slack');
```

Alternative Translation to SQL

```
create table WireWalker(
  id int primary key,
  foreign key(id) references Performer(id)
    on update cascade on delete cascade,
  -- Inheritance strategy: Joined strategy
  wireType int not null,
  foreign key(wireType) references WireType(id)
    on update cascade on delete no action
  -- Enumeration strategy: Object Enumeration strategy
);
create table PerformPerson(
  hasMember int not null,
  foreign key(hasMember) references Person(id)
    on update cascade on delete cascade,
  isMemberOf int not null,
  foreign key(isMemberOf) references Performer(id)
    on update cascade on delete cascade,
  primary key(hasMember, isMemberOf)
);
alter table Performer
  add foreign key(id) references PerformPerson(isMemberOf)
    on update no action on delete no action;
```