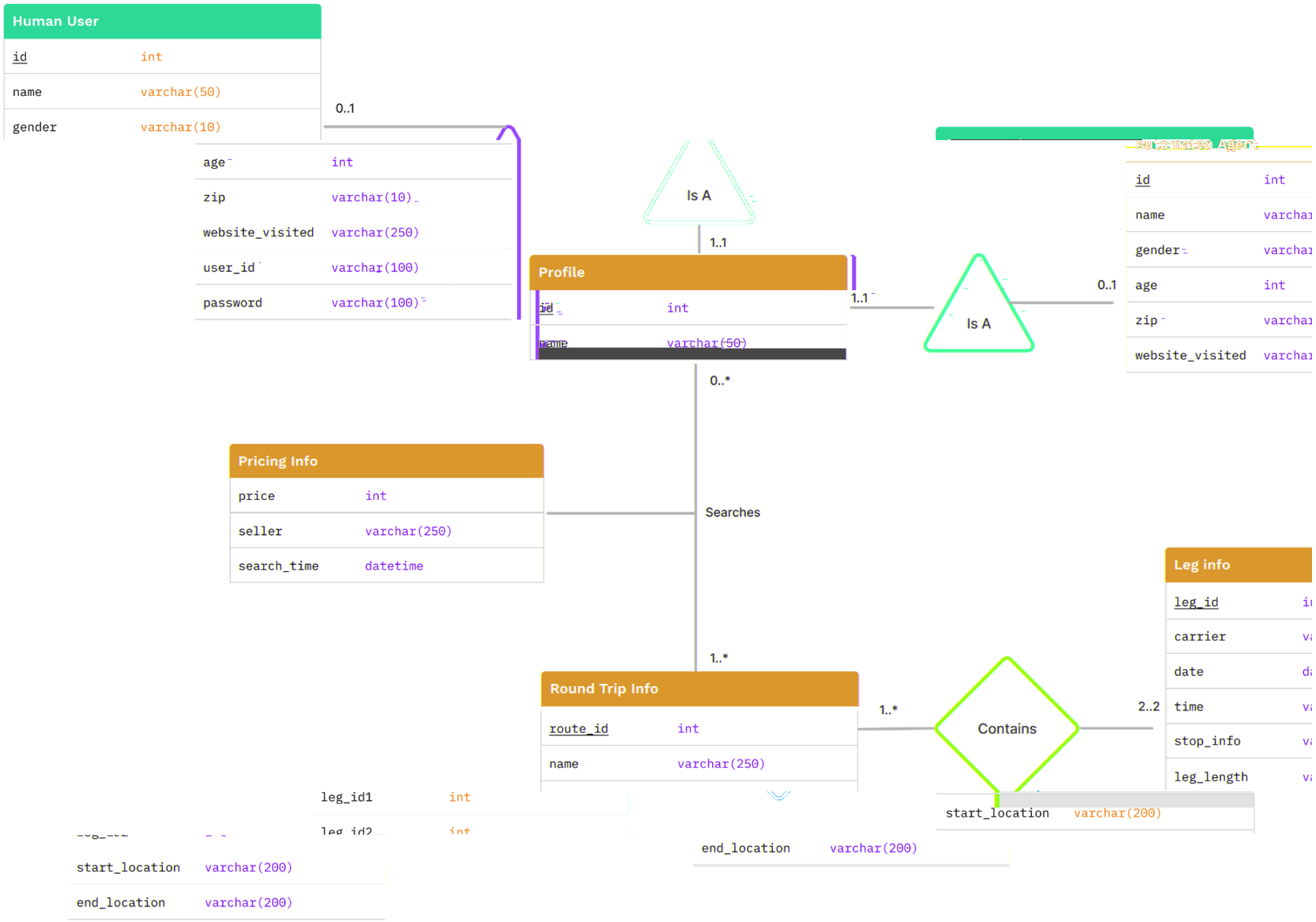
UML Diagram


Human User
id int
name varchar(50)
gender varchar(10)
0..1
age int
zip varchar(10)
website_visited varchar(250)
user_id varchar(100)
password varchar(100)
Is A
1..1
Profile
id int
name varchar(50)
1..1
Is A
0..1
Automated Agent
id int
name varchar
gender varchar
age int
zip varchar
website_visited varchar
0..*
Searches
Pricing Info
price int
seller varchar(250)
search_time datetime
1..*
Round Trip Info
route_id int
name varchar(250)
1..*
Contains
2..2
Leg info
leg_id
carrier
date
time
stop_info
leg_length
leg_id1 int
start_location varchar(200)
end_location varchar(200)
start_location varchar(200)
end_location varchar(200)

Assumptions / Description

We make the following assumptions for our UML diagram:

- Each profile must search for at least one route to exist on the profiles table.
- Each route has exactly two corresponding legs; one for the departing `leg_id` and one for the returning `leg_id`.
- Each leg is part of at least one Round Trip, and may be part of multiple Round Trips.
- There is a subclass structure between Profile and Human User / Automated User; each `id` in the profiles table is found in only one of the subclass tables: Human User or Automated User. Every `id` on the Human User or Automated User tables is found on the Profiles table.

Relational Schema

```
AutomatedAgent(
      id : INT [PK] [FK to Profile.id],
      name : VARCHAR(50),
      gender : VARCHAR(10),
      age : INT,
      zip : VARCHAR(10),
      website_visited : VARCHAR(250)
)

HumanUser(
      id : INT [PK] [FK to Profile.id],
      name : VARCHAR(50),
      gender : VARCHAR(10),
      age : INT,
      zip : VARCHAR(10),
      website_visited : VARCHAR(250),
      user_id : VARCHAR(100),
      password : VARCHAR(100)
)

Profile(
      id : INT [PK],
      name : VARCHAR(50)
)

PricingInfo(
      price : INT,
      seller : VARCHAR(250),
      search_time : DATETIME
```

```
)

RoundTripInfo(
      route_id : INT [PK],
      name : VARCHAR(250),
      leg_id1 : INT [FK references LegInfo.leg_id],
      leg_id2 : INT [FK references LegInfo.leg_id],
      start_location : VARCHAR(200),
      end_location : VARCHAR(200)
)

LegInfo(
      leg_id : INT [PK],
      carrier : VARCHAR(100),
      date : DATE,
      time : VARCHAR(100),
      stop_info : VARCHAR(200),
      leg_length : VARCHAR(200),
      start_location : VARCHAR(200),
      end_location : VARCHAR(200)
)
```