

Hibernate Fundamentals

Introducing Object-relational Mapping



Cătălin Tudose

Java Champion, PhD in Computer Science, Java and Web Technologies Expert

www.catalintudose.com | www.linkedin.com/in/catalin-tudose-847667a1



Overview

ORM (Object-relational Mapping)

JPA and Hibernate

Advantages and drawbacks of Hibernate

Object-relational Impedance Mismatch

Simple Hibernate application



What Is ORM?

Object-relational Mapping

Storing the representation of the objects



JPA and Hibernate

Jakarta Persistence
API

Hibernate

Mapping logic



Advantages of JPA and Hibernate

Write less code

Quicker development

Focusing on OOP

Consistent model to interact with the database

Independent of the database vendor



Drawbacks of JPA and Hibernate

Learning curve

Harder to debug

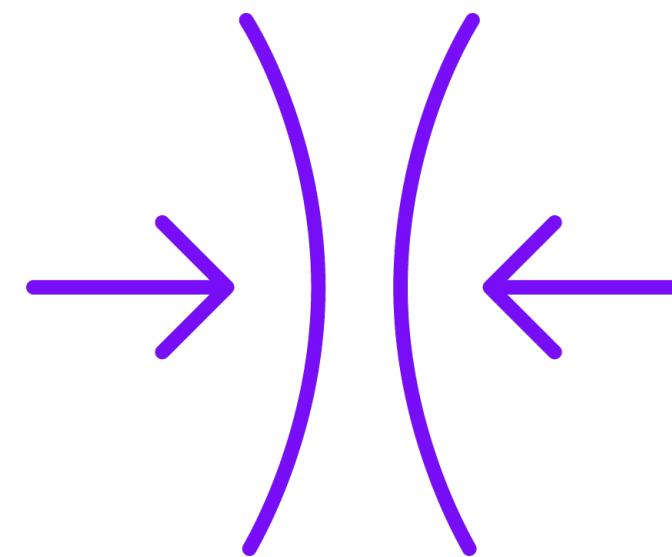
Performance may suffer

JDBC is closer to the database

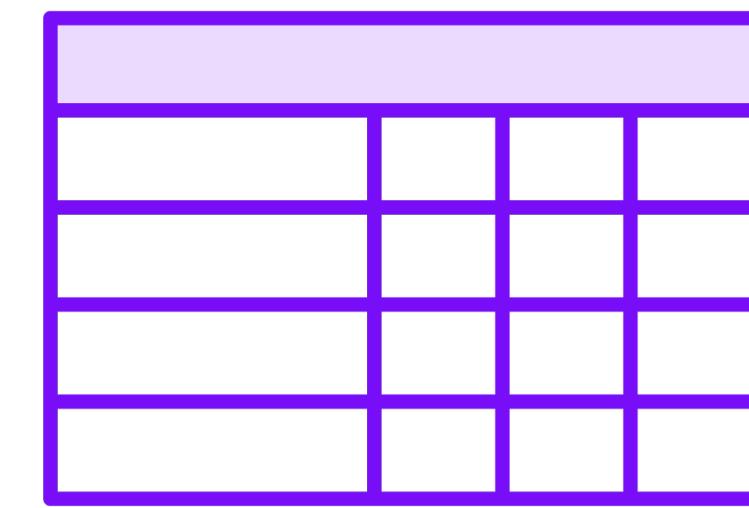
Use specific features of a vendor database



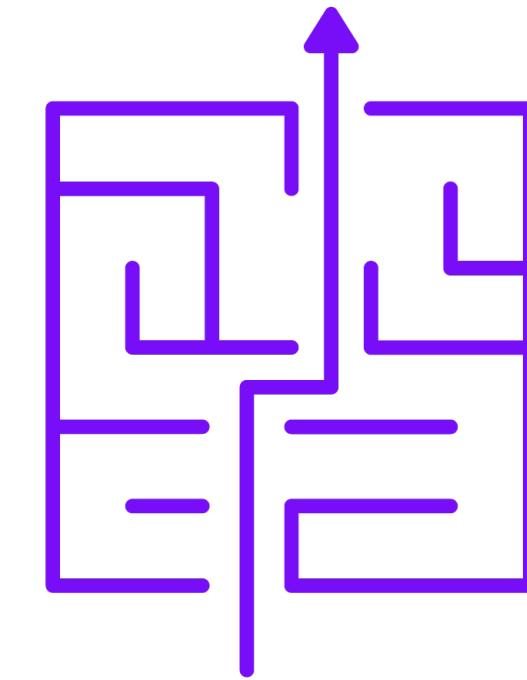
Object-relational Impedance Mismatch



**Object and relational
models do not work
fine together**

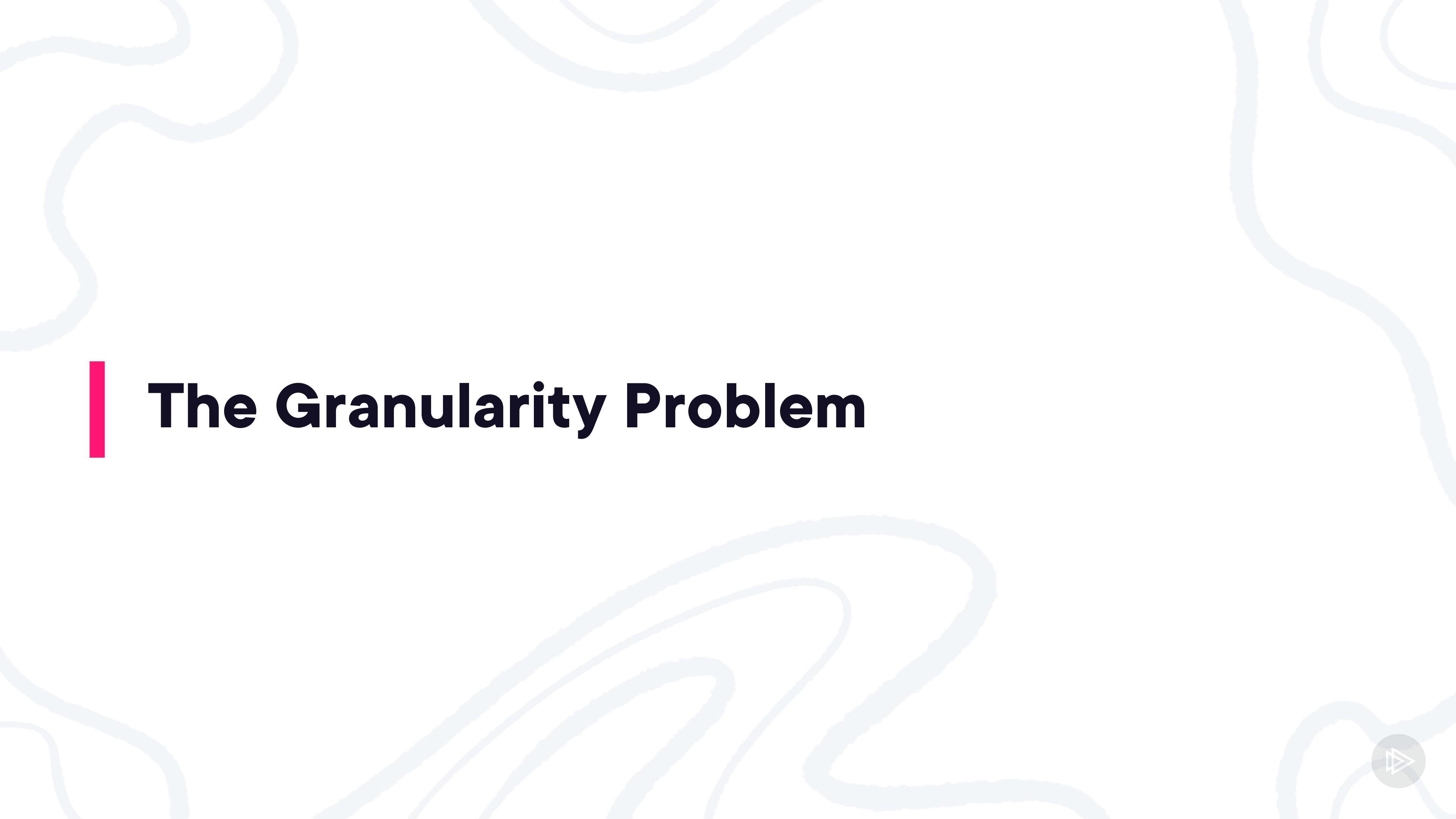


**Interconnected
objects vs. related
tables**



**Granularity,
inheritance, identity,
associations, and data
navigation**

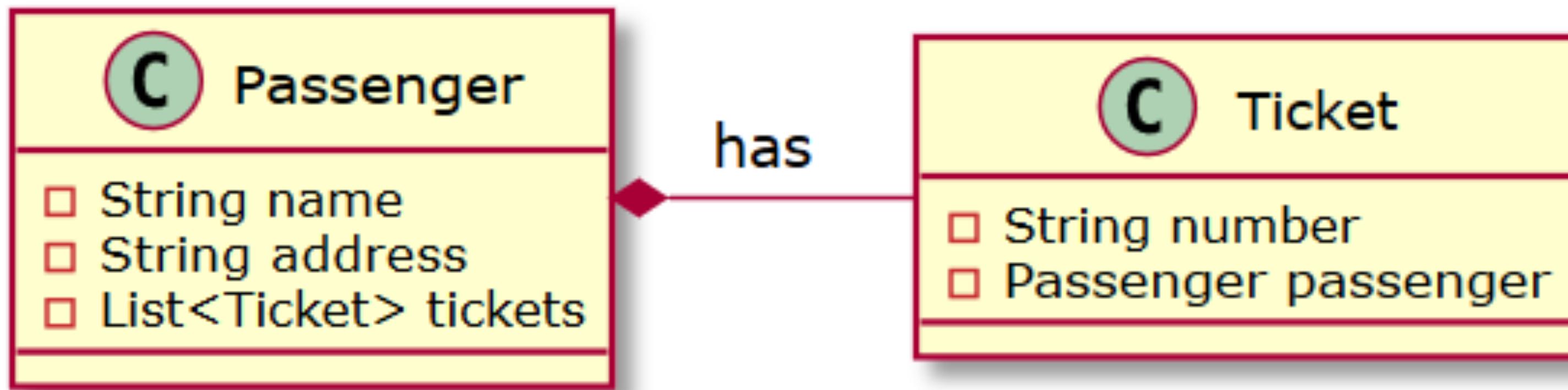




The Granularity Problem



The Flights Management Application



The Flights Management Classes

```
public class Passenger {  
    private String name;  
    private String address;  
    private List<Ticket> tickets;  
}
```

```
public class Ticket {  
    private String number;  
    private Passenger passenger;  
}
```



The Flights Management Tables

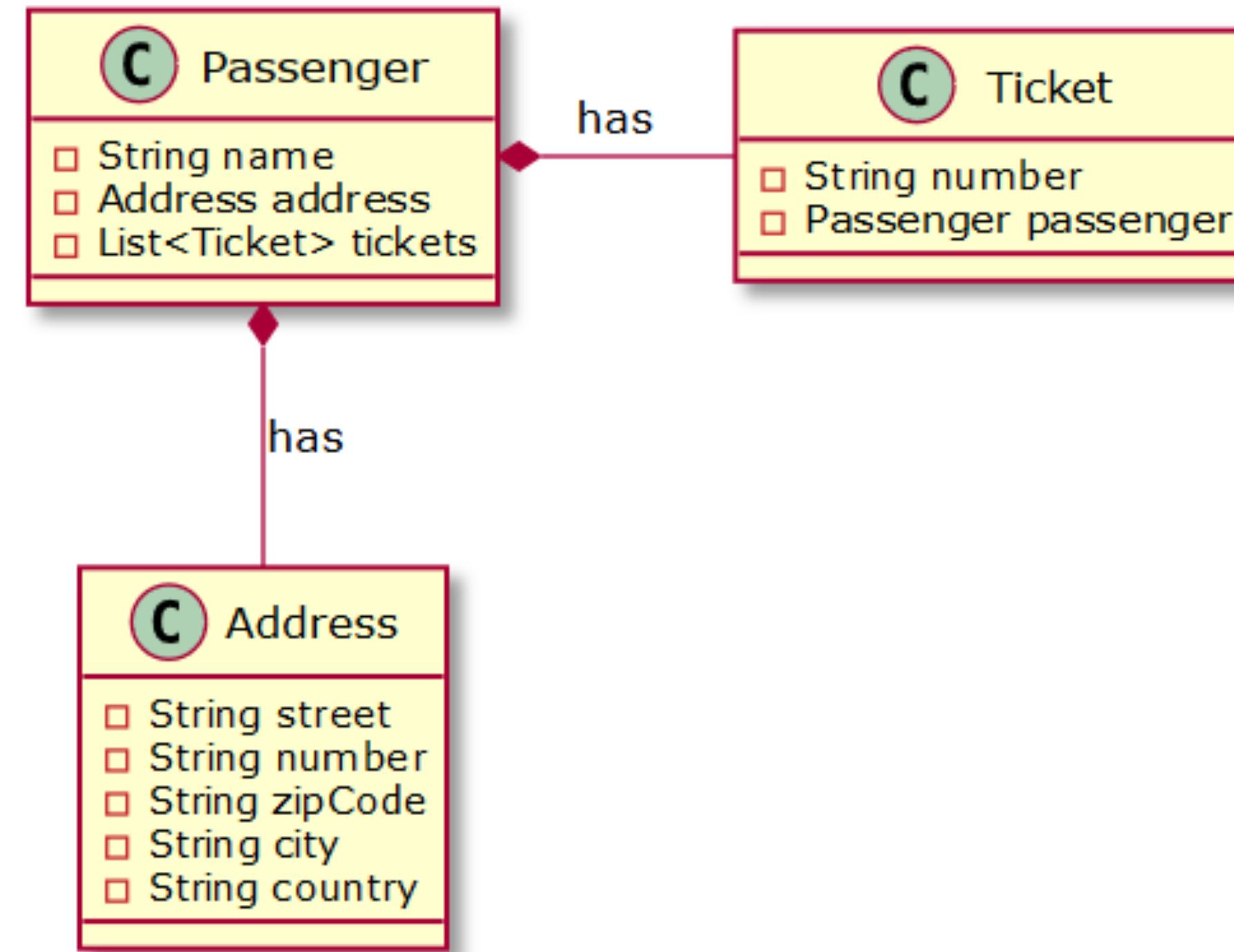
```
create table PASSENGERS (
    NAME varchar(255),
    ADDRESS varchar(255),
    primary key (NAME)
)
```

```
create table TICKETS (
    NUMBER varchar(255),
    PASSENGER_NAME varchar(255),
    primary key (NUMBER)
)
```

```
alter table TICKETS
    add constraint FK_PASSENGERS
        foreign key (PASSENGER_NAME)
        references PASSENGERS (NAME)
```



The Extended Flights Management Application



The Extended PASSENGERS Table

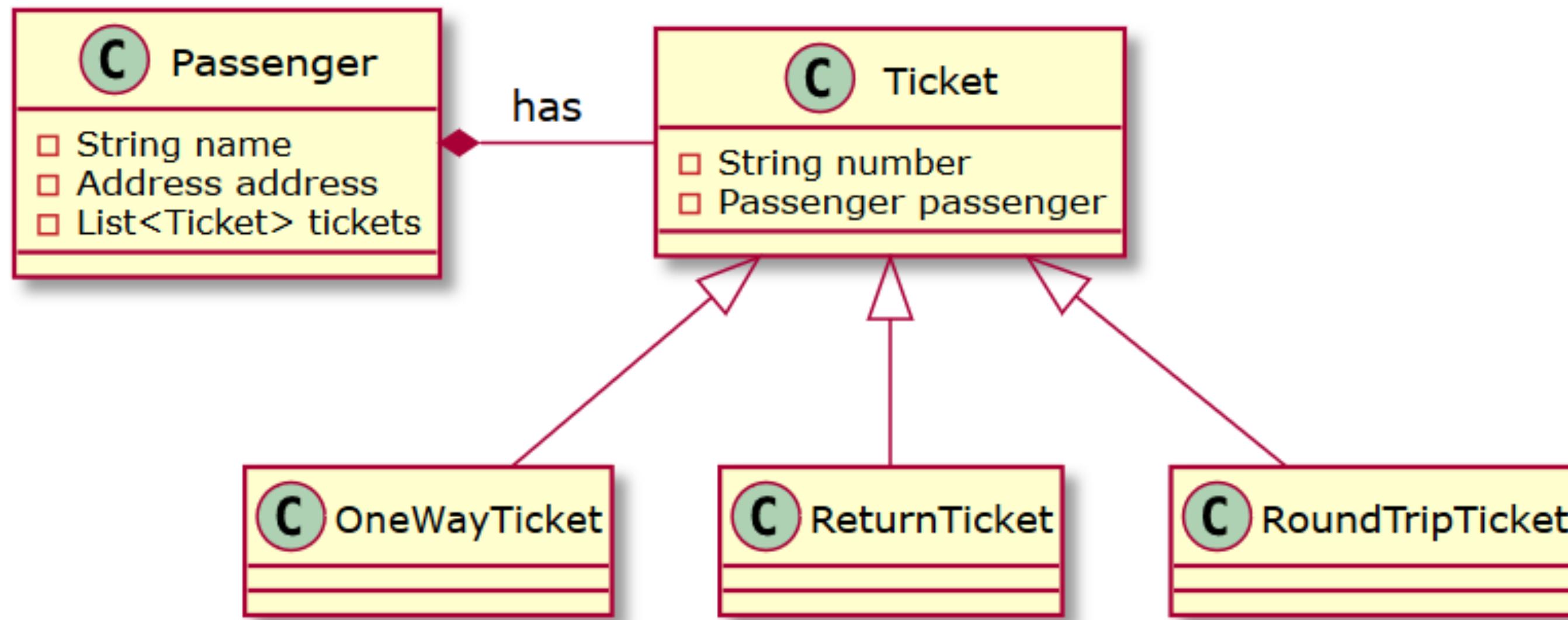
```
create table PASSENGERS (
    NAME varchar(255),
    ADDRESS_STREET varchar(30),
    ADDRESS_NUMBER varchar(6),
    ADDRESS_ZIPCODE varchar(10),
    ADDRESS_CITY varchar(25),
    ADDRESS_COUNTRY varchar(25),
    primary key (NAME)
)
```



The Inheritance Problem



Using Inheritance



The Identity Problem



The PK in the PASSENGERS Table

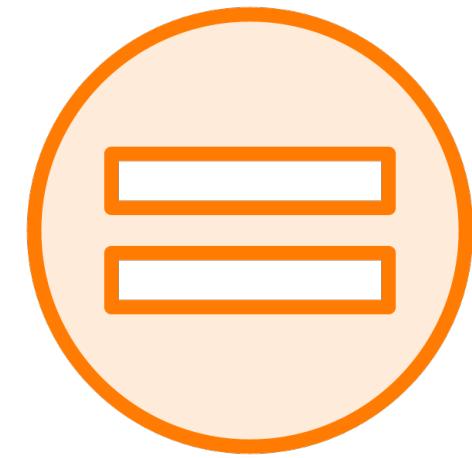
```
create table PASSENGERS (
    NAME varchar(255),
    ADDRESS_STREET varchar(30),
    ADDRESS_NUMBER varchar(6),
    ADDRESS_ZIPCODE varchar(10),
    ADDRESS_CITY varchar(25),
    ADDRESS_COUNTRY varchar(25),
    primary key (NAME)
)
```



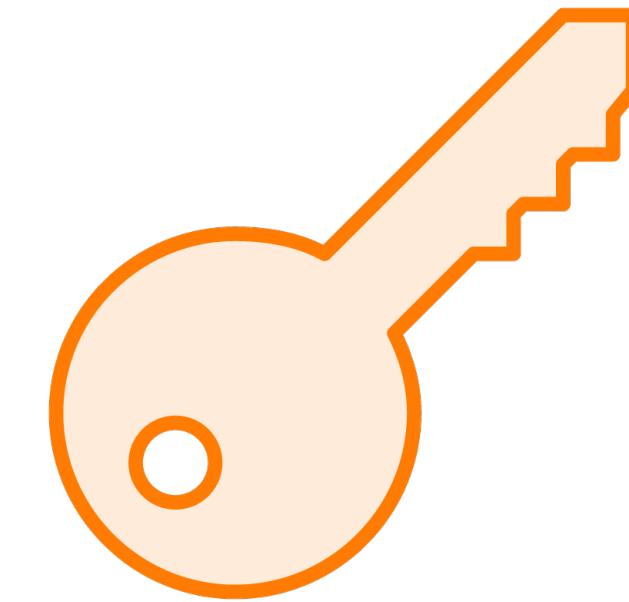
Defining Uniqueness



**Objects identity
(the `==` operator)**



**Logical equality
(the `equals` method)**



Primary Keys



Tables with Surrogate Keys

```
create table PASSENGERS (
    ID integer not null,
    NAME varchar(255),
    primary key (ID)
)

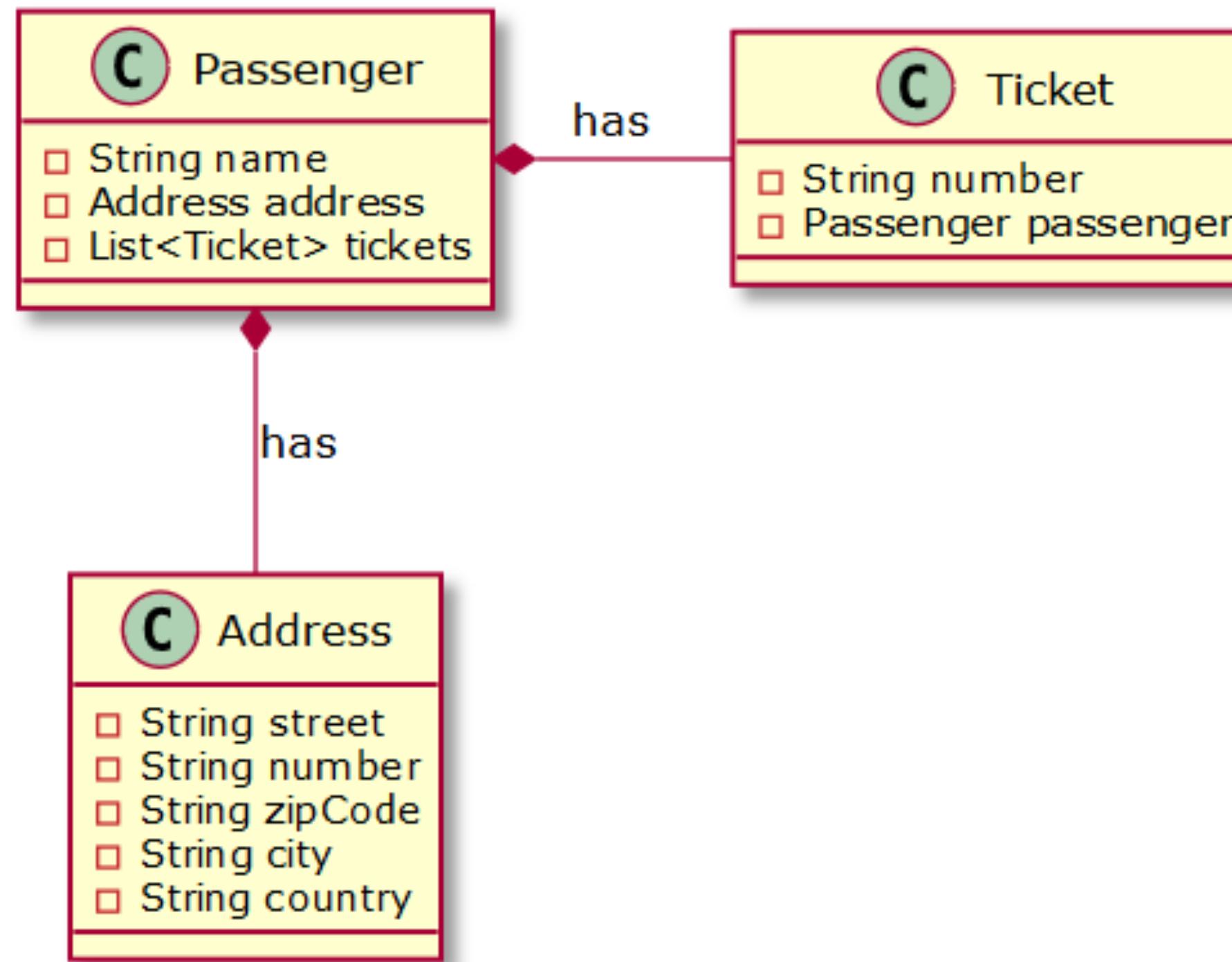
create table TICKETS (
    ID integer not null,
    NUMBER varchar(255),
    PASSENGER_ID integer,
    primary key (ID)
)

alter table TICKETS
    add constraint FK_PASSENGERS
        foreign key (PASSENGER_ID)
        references PASSENGERS (ID)
```



The Associations Problem

Associations in the Object-oriented Model



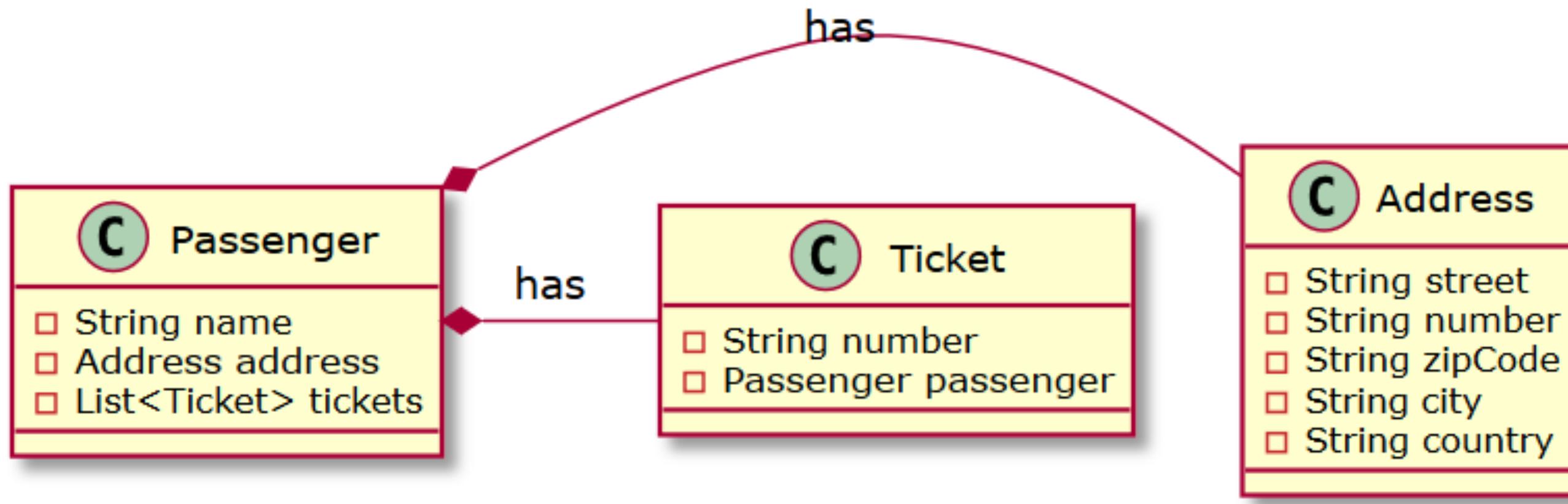
Associations in the Relational Model

```
create table PASSENGERS (
    ID integer not null,
    NAME varchar(255),
    ADDRESS_STREET varchar(30),
    ADDRESS_NUMBER varchar(6),
    ADDRESS_ZIPCODE varchar(10),
    ADDRESS_CITY varchar(25),
    ADDRESS_COUNTRY varchar(25),
    primary key (ID)
)
```

```
create table TICKETS (
    ID integer not null,
    NUMBER varchar(255),
    PASSENGER_ID integer,
    primary key (ID)
)
```



Associations in the Object-oriented Model



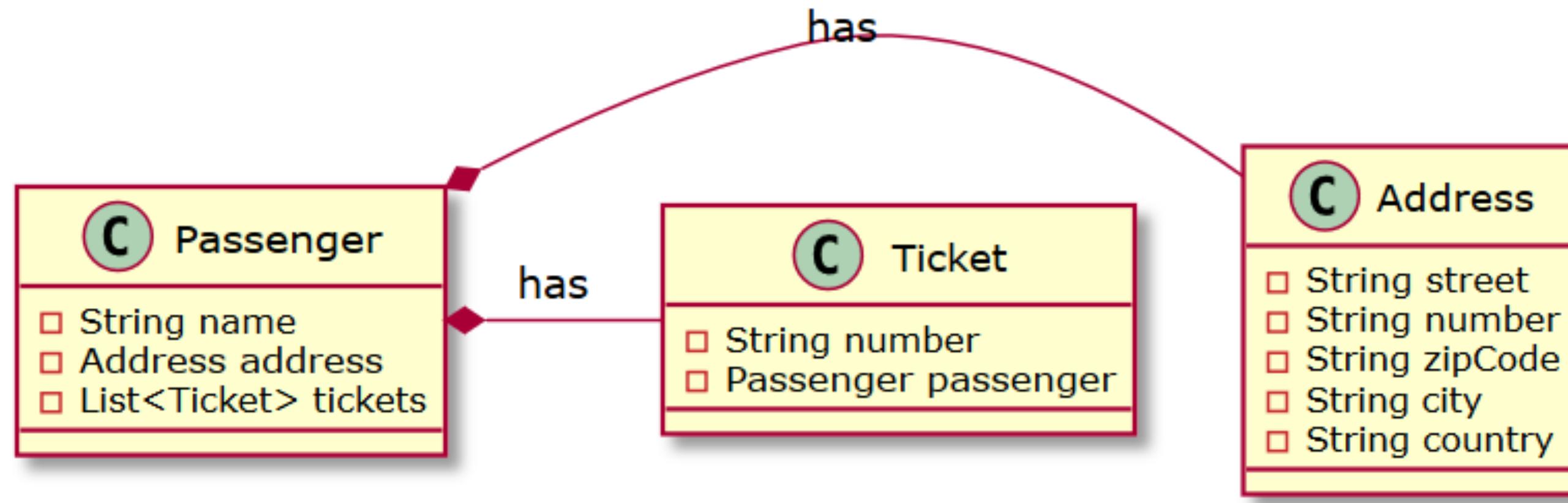
Associations in the Relational Model

```
create table PASSENGERS (
    ID integer not null,
    NAME varchar(255),
    ADDRESS_STREET varchar(30),
    ADDRESS_NUMBER varchar(6),
    ADDRESS_ZIPCODE varchar(10),
    ADDRESS_CITY varchar(25),
    ADDRESS_COUNTRY varchar(25),
    primary key (ID)
)
```

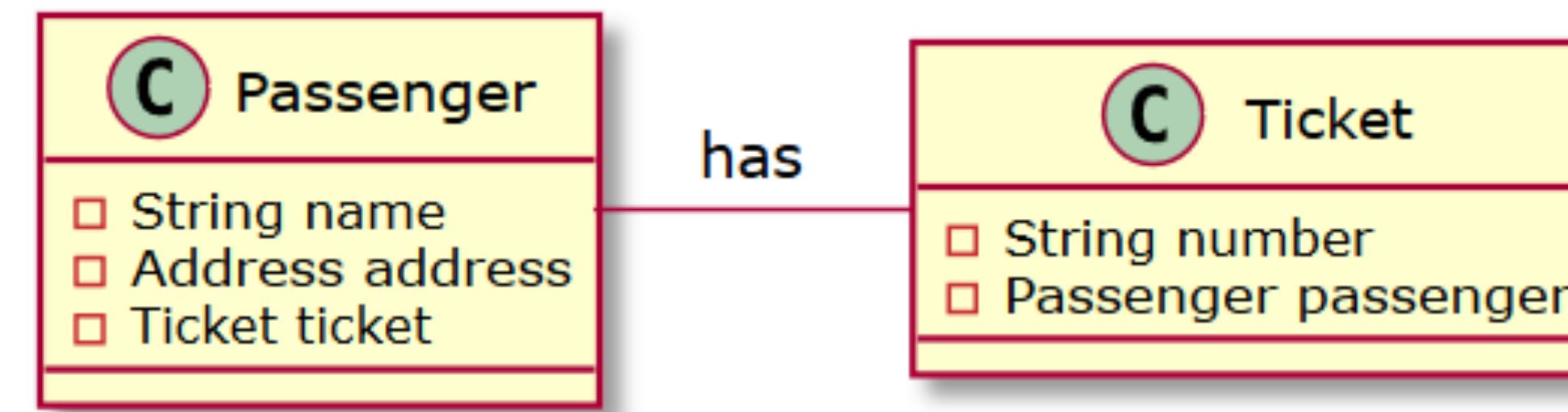
```
create table TICKETS (
    ID integer not null,
    NUMBER varchar(255),
    PASSENGER_ID integer,
    primary key (ID)
)
```



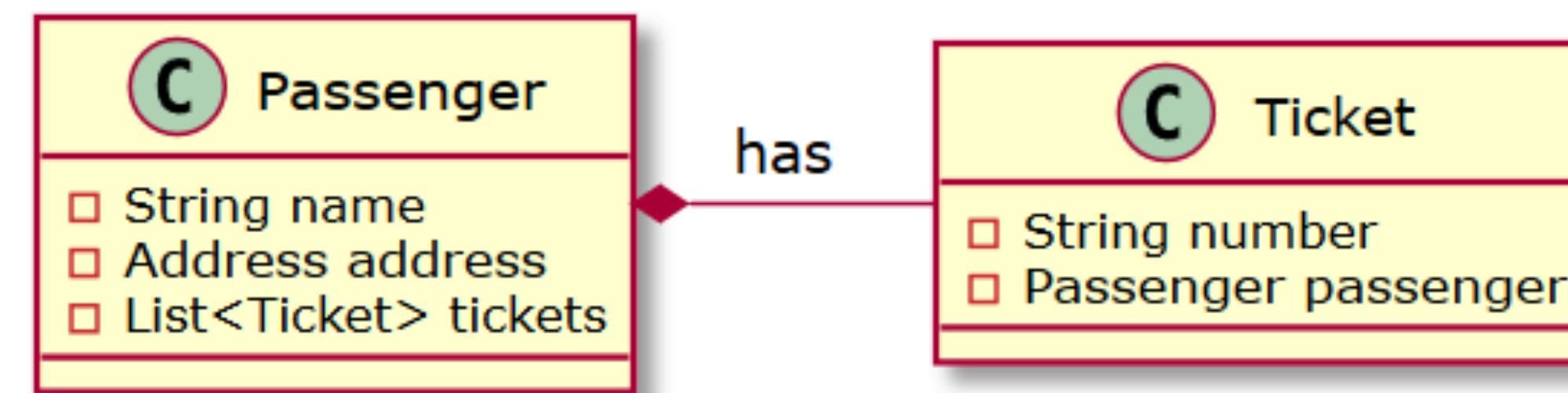
Associations in the Object-oriented Model



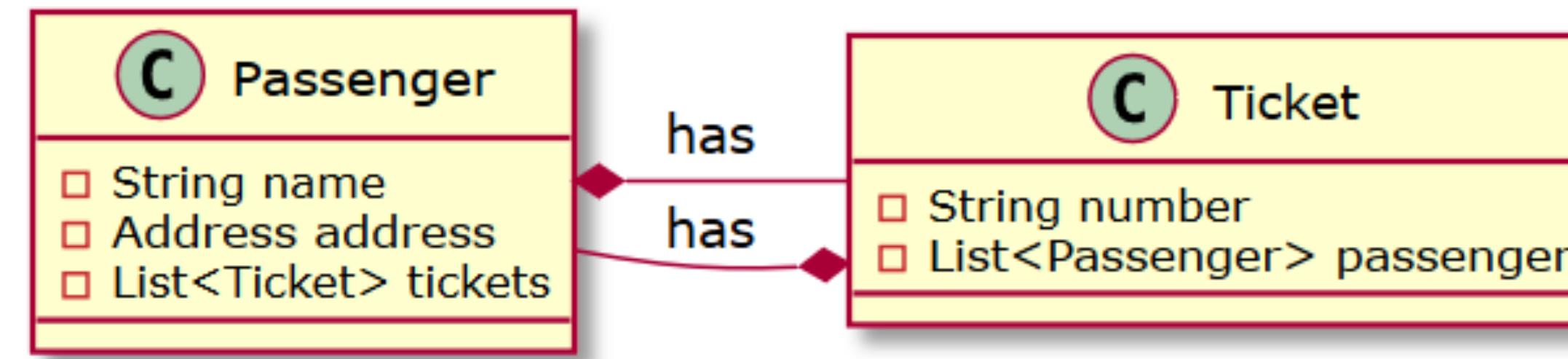
One-to-One Association



One-to-Many Association



Many-to-Many Association



Associations in the Relational Model

```
create table PASSENGERS (
    ID integer not null,
    NAME varchar(255),
    ADDRESS_STREET varchar(30),
    ADDRESS_NUMBER varchar(6),
    ADDRESS_ZIPCODE varchar(10),
    ADDRESS_CITY varchar(25),
    ADDRESS_COUNTRY varchar(25),
    primary key (ID)
)
```

```
create table TICKETS (
    ID integer not null,
    NUMBER varchar(255),
    PASSENGER_ID integer,
    primary key (ID)
)
```



Many-to-Many Associations in Relational Model

```
create table PASSENGERS_TICKETS (
    PASSENGER_ID integer not null,
    TICKET_ID integer not null,
    primary key (PASSENGER_ID, TICKET_ID)
)
```

```
alter table PASSENGERS_TICKETS
    add constraint FK_PASSENGERS
        foreign key (PASSENGER_ID)
        references PASSENGERS (ID)
```

```
alter table PASSENGERS_TICKETS
    add constraint FK_TICKETS
        foreign key (TICKET_ID)
        references TICKETS (ID)
```

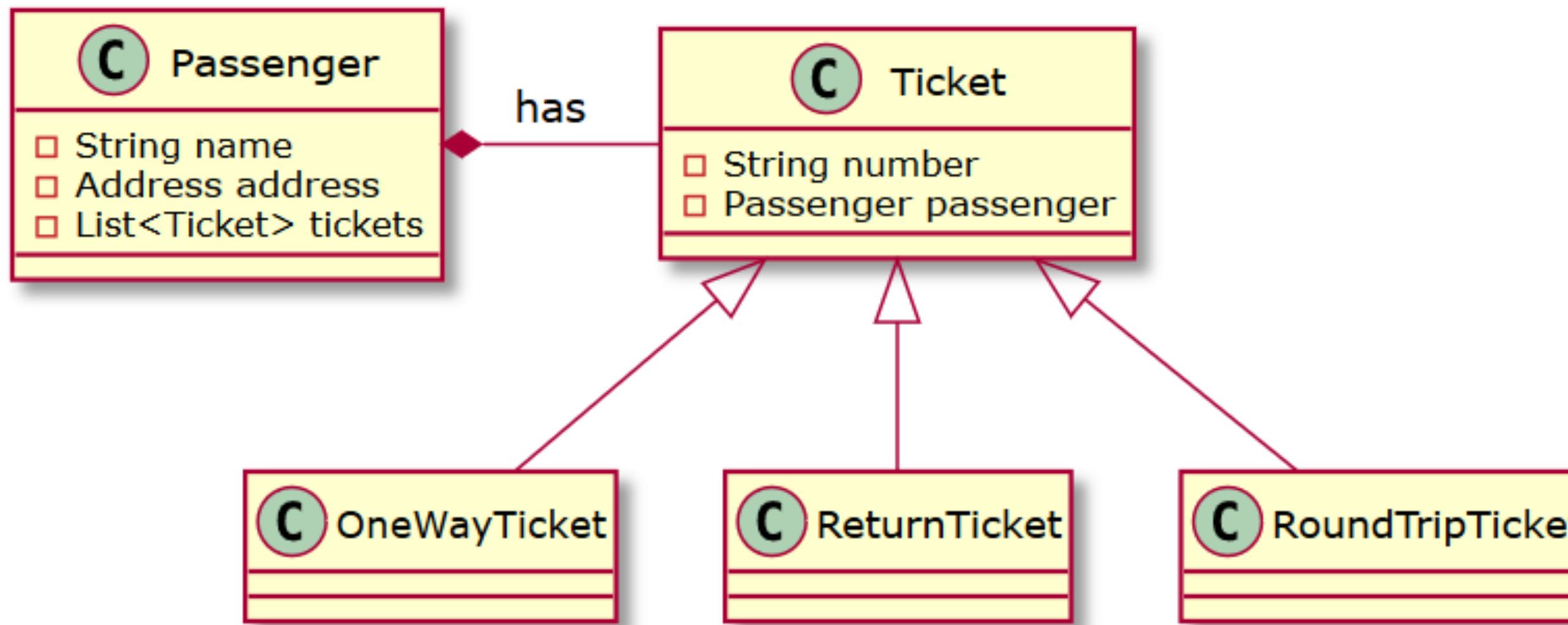




The Data Navigation Problem



Data Navigation in the Object-Oriented Model



Data Navigation in the Relational Model

```
create table PASSENGERS (
    ID integer not null,
    NAME varchar(255),
    ADDRESS_STREET varchar(30),
    ADDRESS_NUMBER varchar(6),
    ADDRESS_ZIPCODE varchar(10),
    ADDRESS_CITY varchar(25),
    ADDRESS_COUNTRY varchar(25),
    primary key (ID)
)
```

```
create table TICKETS (
    ID integer not null,
    NUMBER varchar(255),
    PASSENGER_ID integer,
    primary key (ID)
)
```



Data Navigation Approaches

```
for(Ticket ticket: passenger.getTickets())
```

```
SELECT * FROM PASSENGERS WHERE ID = 727423
```

```
SELECT * FROM PASSENGERS, TICKETS  
WHERE PASSENGERS.ID = 727423 AND  
PASSENGERS.ID = TICKETS.PASSENGER_ID
```



Demo

Create a Jakarta EE 10 with Hibernate project

Create the entity classes

Persist objects to the database



Summary

Object-relational Mapping (ORM)

JPA and Hibernate

Hibernate:

- Advantages
- Drawbacks

Problems of Object-relational Impedance Mismatch

Simple Hibernate application

