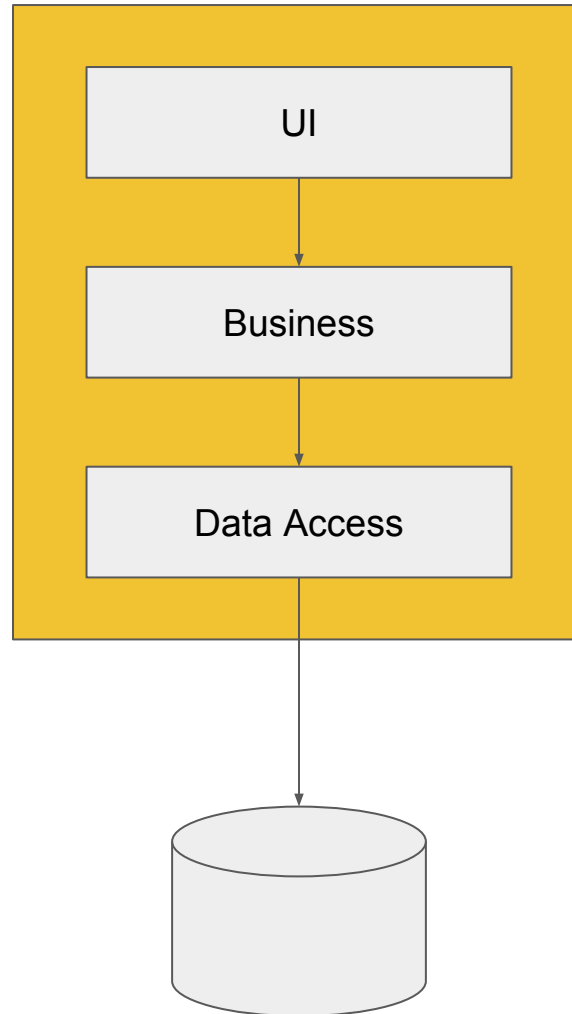
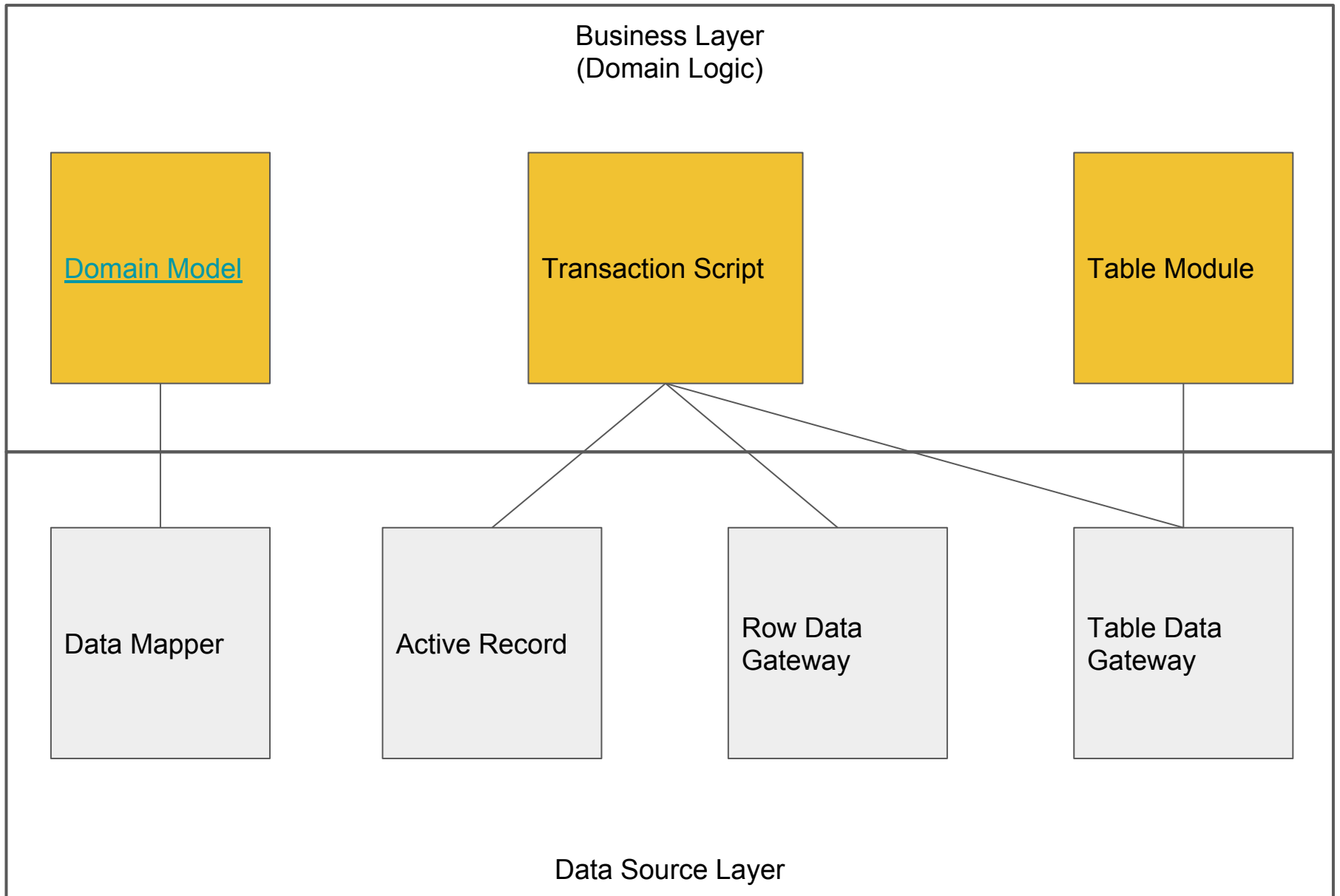


Domain Logic and Data Source Patterns

Based on *Martin Fowler et. al, Patterns of Enterprise
Application Architecture, Addison
Wesley, 2003 (Chapters 10 and 9)*

3-TIER Architecture





Active Record Vs Row Data Gateway

Active Record

1. One Object Per Row
2. Has Fields
3. Has Insert, Update, Delete, Methods
4. Has either a static find(identifier) method or separate Finder class
5. Must change implementation if DataBase changes

Has Business Logic

=> generating a new file overwrites your business Logic code

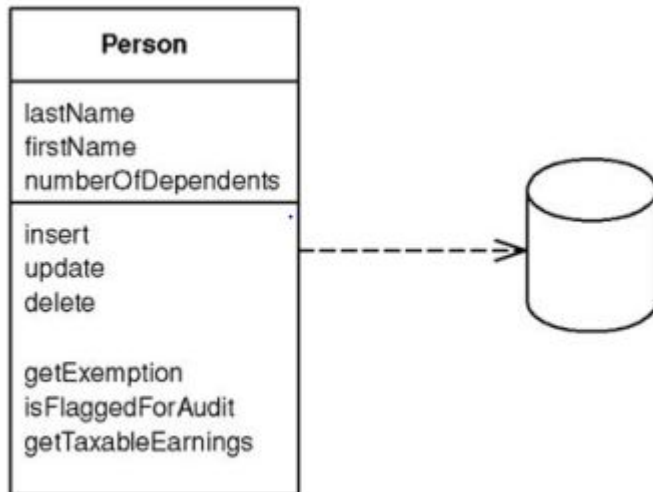
Row Data Gateway

NO Business Logic

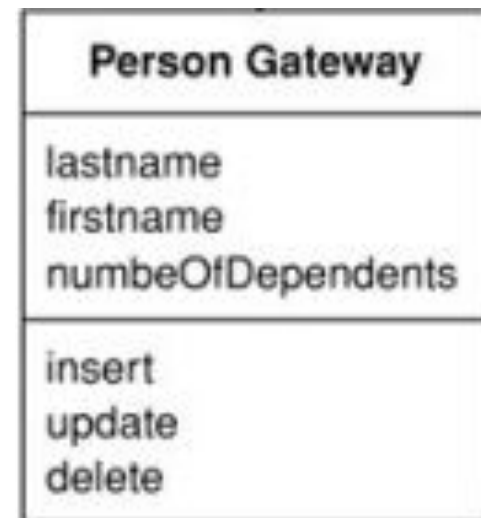
=> Can be generated based on DB

Active Record Vs Row Data Gateway

Active Record



Row Data Gateway



Business Layer
(Domain Logic)

Transaction Script

Table Data
Gateway

In the original Pattern:
The Gateway takes as parameters
only Wrappers over primitive types

Person Gateway

```
find (id) : RecordSet  
findWithLastName(String) : RecordSet  
update (id, lastname, firstname, numberOfDependents)  
insert (lastname, firstname, numberOfDependents)  
delete (id)
```

Data Source Layer

Business Layer
(Domain Logic)

gateway.find(idFromUI)

Transaction1

Transaction2

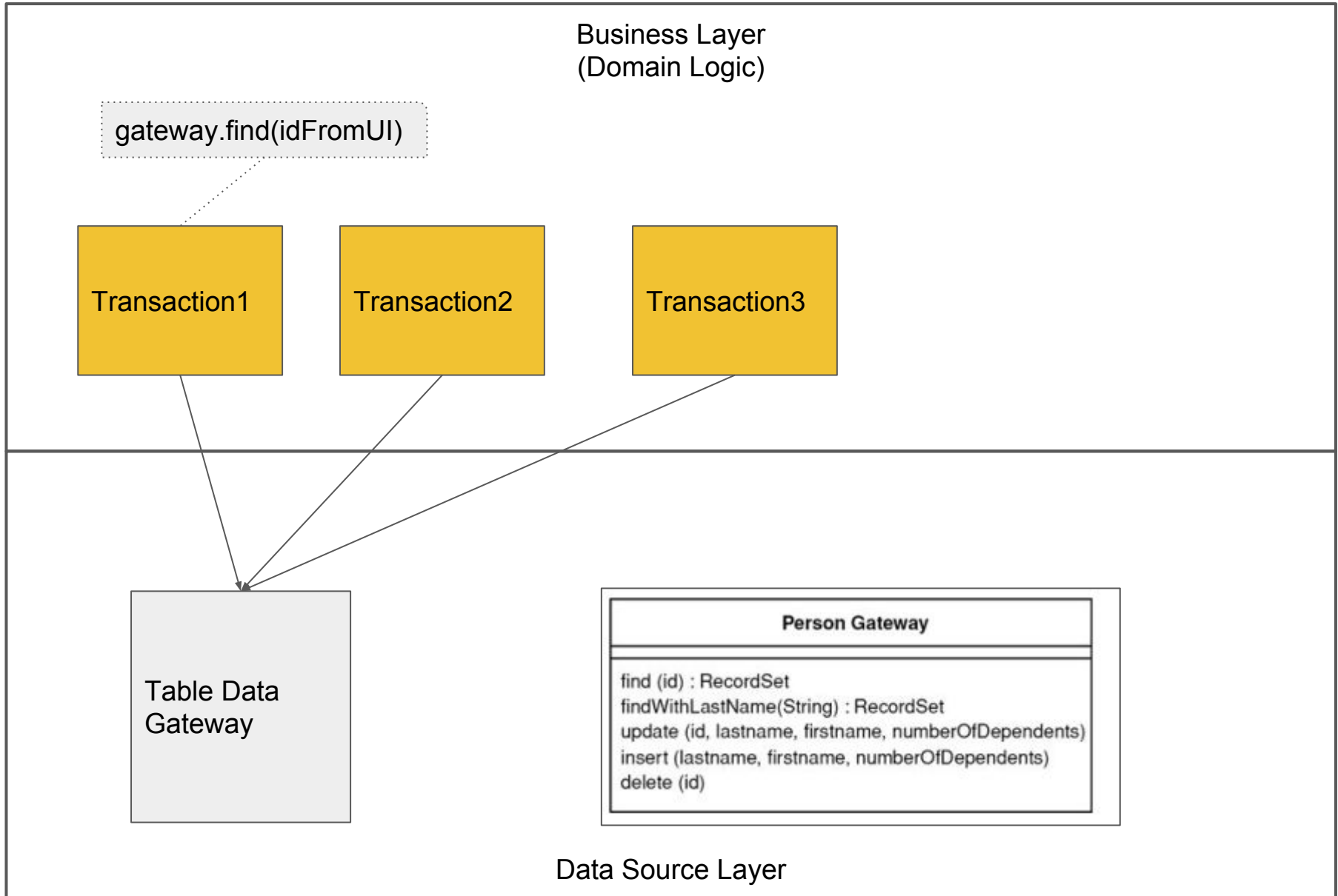
Transaction3

Table Data
Gateway

Person Gateway

```
find (id) : RecordSet  
findWithLastName(String) : RecordSet  
update (id, lastname, firstname, numberOfDependents)  
insert (lastname, firstname, numberOfDependents)  
delete (id)
```

Data Source Layer



Business Layer
(Domain Logic)

Table Module

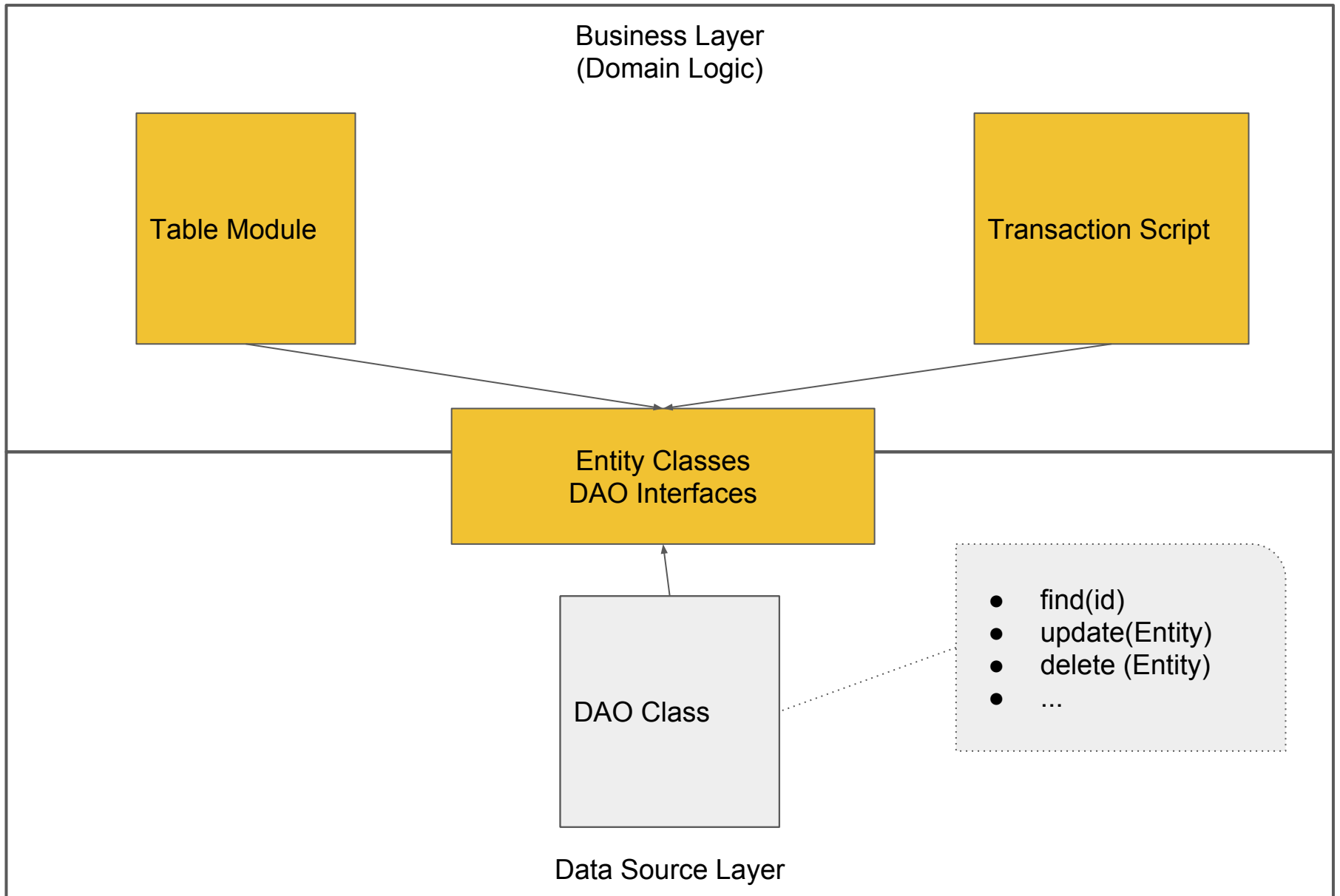
- findPerson(id)
- calculatePersonBMI(id)
- getAllPersonFirstNames()
- ...

Table Data
Gateway

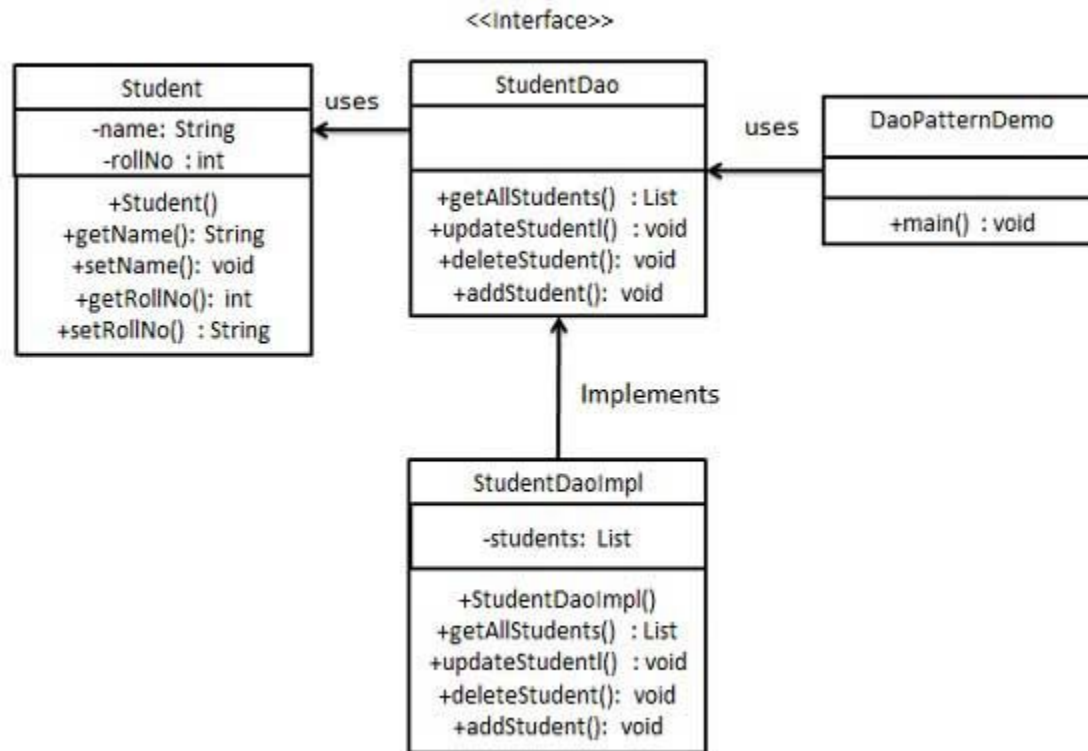
Person Gateway

```
find (id) : RecordSet  
findWithLastName(String) : RecordSet  
update (id, lastname, firstname, numberOfDependents)  
insert (lastname, firstname, numberOfDependents)  
delete (id)
```

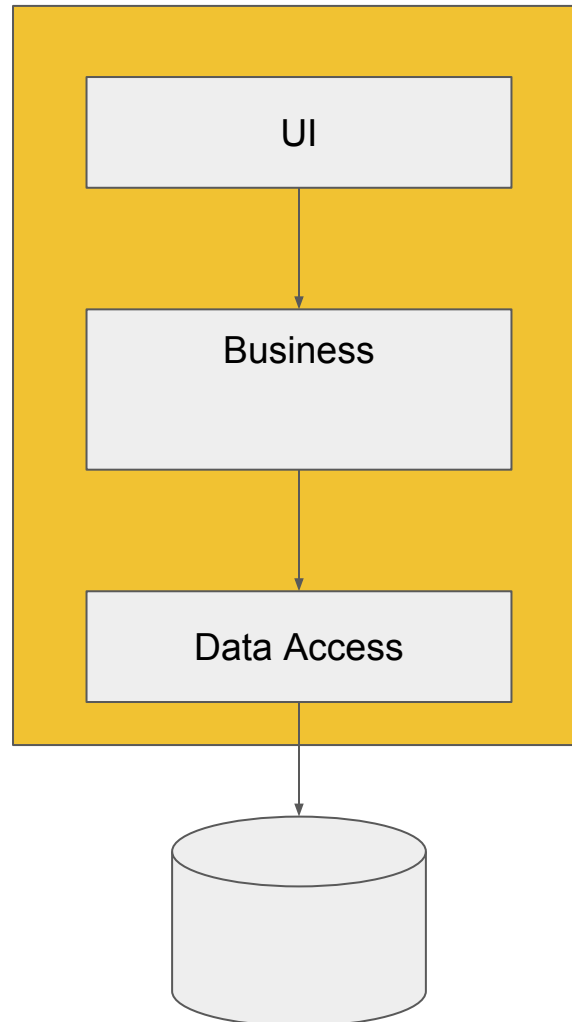
Data Source Layer



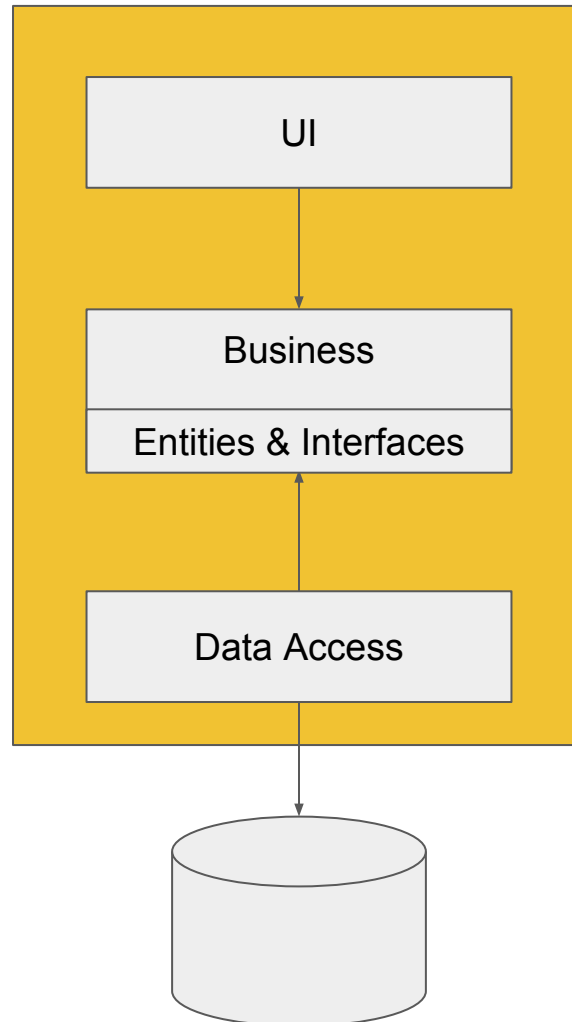
The Dao Design Pattern



3-TIER Architecture



3-TIER Architecture



Questions?

Then let's start coding

