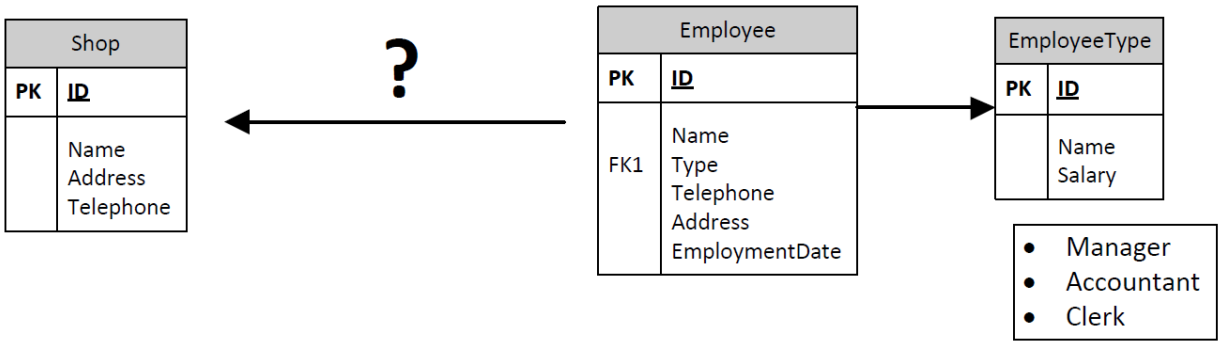
Database Design



- ID is primary key, AutoIncrement
- Type is Foreign Key to EmployeeType
- Choose column type freely
- Nullables are free to choose
- The relation between Shop and Employee is the following.
    - One Employee can work in multiple Shops
    - Its restricted by work date
    - Employee X can work at Store A today, Store B tomorrow, Store C day after, etc.
    - Employee X can't work at Store A today and Store B today.

Shopping Web Api

The objective is to create a simple web application that will list all employees at the company and also have the ability to create new employees.

Architecture

The solution consists in **3 projects**

1. Web Application Layer (Empty Web Project)
2. Business Layer
3. Data Access Layer (ADO .NET, Entity Framework, NHibernate, etc.)

View All Employees

The application flow should look like

- Controller should invoke list employees from Business Layer
- Business Layer will call a List employees from Data Access Layer
- Data Access Layer will apply the data retrieval based on the method the developer has chosen, should return a data model
- Business Layer will transform data models to business models
- Web Application will render the employees

Add Employee

The application flow should look like

- Controller should invoke add employee from Business Layer, using a Business Model as parameter
- Business Layer will transform business model to data model and invoke add employee from Data Access Layer
- Data Access Layer will insert new candidate
- Business Layer will confirm a success or error
- Web Application will render result feedback