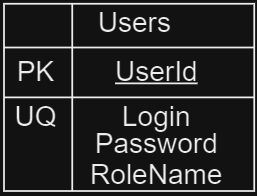
Защита данных

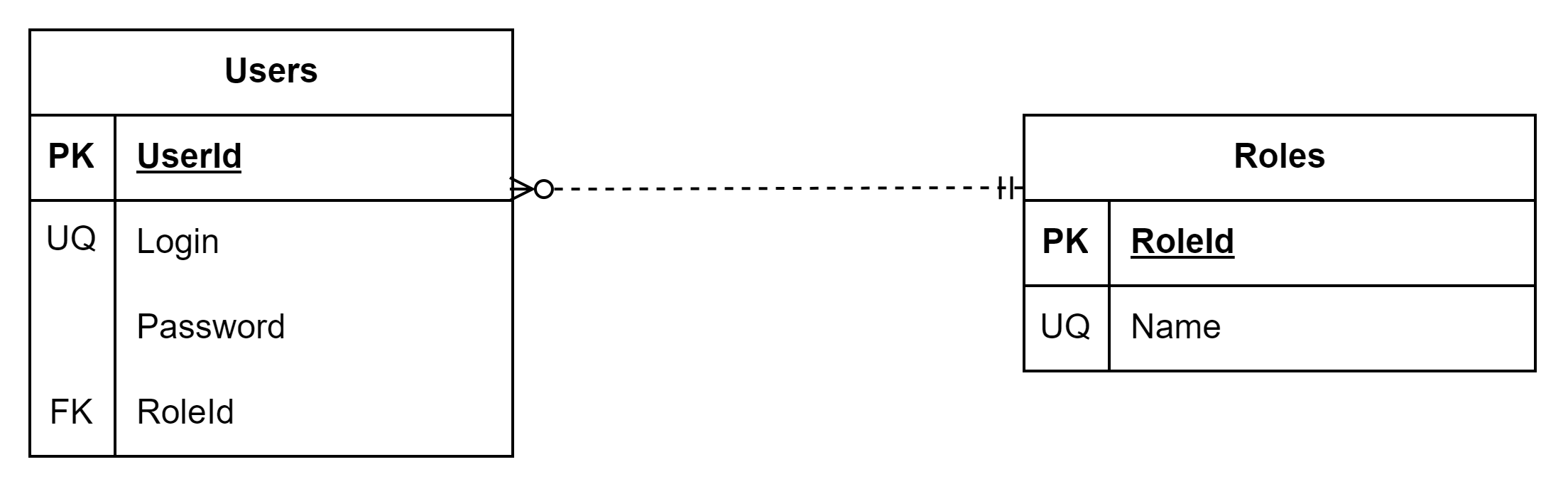
1. Защита данных на уровне клиентского приложение – требует создания в БД таблиц пользователем
2. Защита данных на уровне сервера – требует создания пользователей сервера и БД и назначения им прав доступа

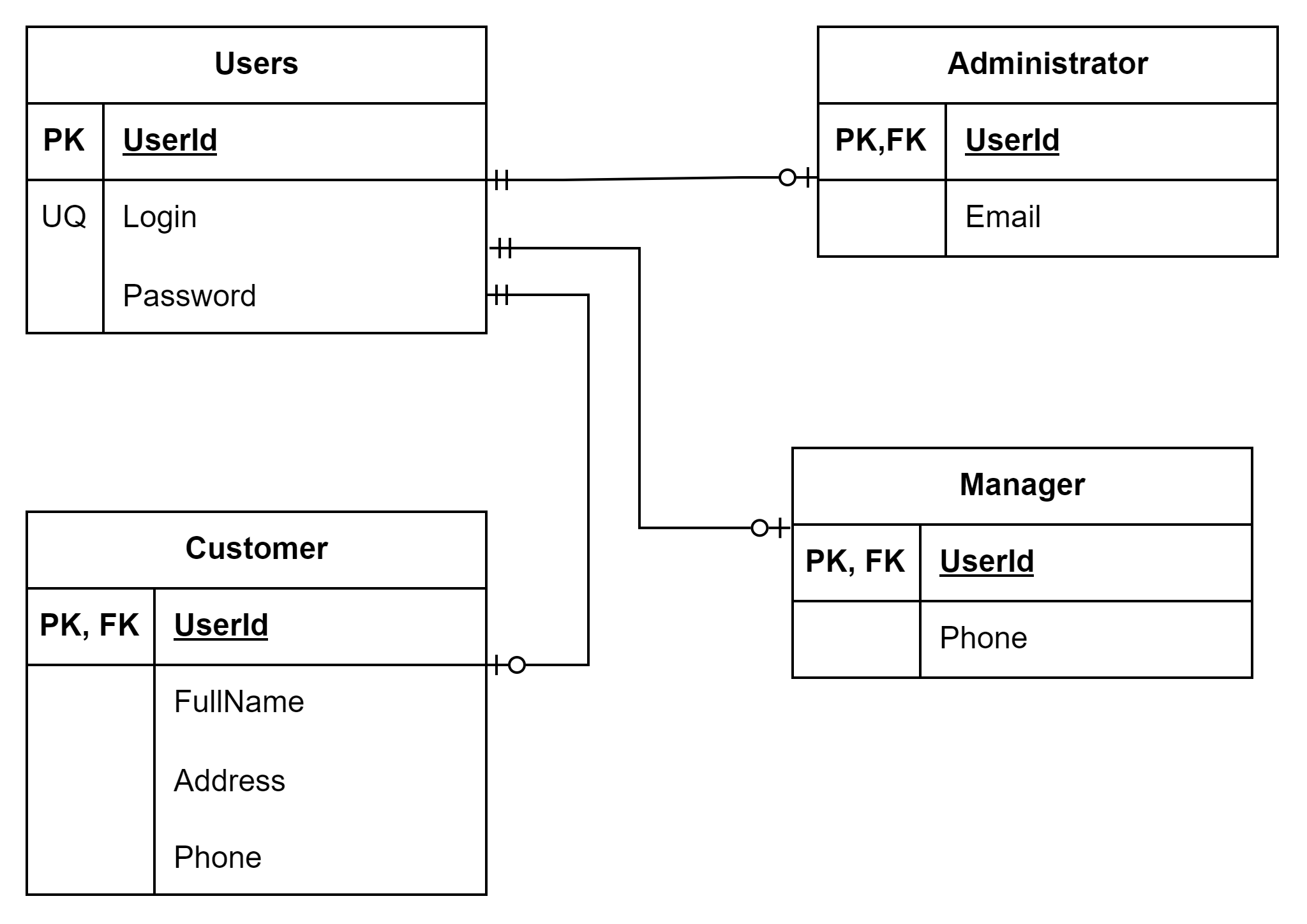
**Проектирование таблицы пользователей:**

-Все в одной таблице



-разделение таблиц

-хранение информации о роли с указанием дополнительных атрибутов роли



Пароли в таблицах пользователей можно зашифровать использование хэширование:

UPDATE Users

SET [Password] = HASHBYTES(‘SHA2\_256’, [Password])

WHERE UserId = …

SELECT IdGame,

[Name]

HASHBYTES(‘SHA2\_256’,[Platform])

FROM [GamesN]

**Система безопасности MSSQL:**

-этапы предоставления доступа к даным

1) аутентификация пользователя и предоставления доступа к серверу

Аутентификация – проверка личности по логину(имя для входа) и паролю

2) предоставление доступа к БД

3) определение ролей, к которым пренадлежит пользоватль и авторизация доступа пользователя к логическим объектам БД на основе прав доступа логика и ролей пользователя

* Авторизация – предоставления прав на вспомагательных действий и контроль доступа поля аутентефикации
* Право доступа – определяет отношения между объектом и субъектом доступа
* Субъект доступа – логин\роль\учетная запись
* Объект доступа – логичекий объект БД, доступ к которому пытается получить пользователь

**Получение доступа к серверу**

Логин – учетная запись для подключения к SQL-серверу

Пример: ispp1112

Типы проверки подлинности:

-Windows – использует логины Windows

* защищенный
* не требует ввода пароля
* подлючение только с ПК, на котором установлен СУБД

-SQL Server

Проверить состояние имени входа: Безопасность – Имя – ПКМ, на котором установлен СУБД

Создание логина:

1. CREATE LOGIN ИмяВхода

WITH PASSQWORD = ‘пароль’

Доп настройки:

CHECK\_EXPIRATION = ON – применение политики истечения срока действия

* CHECK\_POLICY = ON – требования к паролю
* MUST\_CHANGE – смена пароля после авторизации

1. Процедура: sp\_addlogin ‘ИмяВхода’, ‘пароль’
2. Сервер – Безопасность – Имена для входа – Создать …

Если имени входа нужны права доступа на уровне сервера, ему назначаются серверные роли; их нельзя создасть, выбираются из списка доступных:

* Public – нет прав, у каждого по умолчанию
* dbcreator – создание\изменение\удаление\восстановление БД
* securityadmin – управление логинами и разрешениями
* sysadmin – все права доступа на сервере
* bulkadmin – команды BULK INSERT – вставка данных из файлов
  + sp\_addsrvrolemember ‘ИмяВхода’, ‘рольСервера’
  + sp\_dropsrvrolemember ‘Логин’,’Роль’

**Предоставление доступа на уровне БД**

1. Создание пользователя

Пользователь БД – объект, создающийся на уровне БД и использующийся для предоставления разрешений к объектам БД

Схема БД – набор объектов БД, объединямых общими пространством имен

dbo-схема по умолчанию (database owner)

Пользователь: БД – Безопасность – Пользователь

Встроеные пользователи БД:

dbo – владелец БД

guest – используется для логина, котороу нужн разрешить чтение данных

1. CREATE USER ИмяПользователяБД

FOR LOGIN ИмяВхода

2)sp\_adduser ‘ИмяВхода’, ‘ИмяПользователяБД’

3)Через оконный интерфейс

2)назначение ролей доступа

Стандартные роли:

db\_owner – владелец БД

db\_datareader – резрашение чтения данных

db\_datawriter – DML-команды

db\_denydatareader – запрет чтения

db\_denydatawriter – запрет изменениия

db\_ddladmin – DDL-команды

db\_accessadmin – добавление/удаление прав доступа к БД

db\_backupoperator – создание резервной копии БД

sp\_addrolemember ‘роль’, ‘пользователь БД’

DCL – Data Control Language – GRANT(выдача разрешений) DENY(запрет разреший) REMOVE(отзыв разрешений)

GRANT список команд

ON объект БД

TO пользователь БД

GRANT SELECT, UPDATE

ON Games

TO guest

GRANT CREATE TABLE

TO guest

DENY EXECUTE

ON Proc1

TO guest

DENY VIEW

ANY DATABASE – запрет просмотра списка БД

TO ИмяВхода

**Создание скриптов для автоматизации действий по настройке прав доступа**

DECLARE @userCount INT = 15

DECLARE @currentUser INT = 1

DECLARE @dbName VARCHAR (10)

DECLARE @password VARCHAR (20)

WHILE (@currentUser <= @userCount)

BEGIN

SET @dbName = ‘test’ + CAST(@currentUser AS VARCHAR)

SET @password = ‘pwd’ + @dbName

PRINT @dbName

EXEC (‘CREATE LOGIN [‘+@dbName+’] WITH PASSWORD = N’’’ +@password+’’’, CHECK\_POLICY=OFF)

EXEC (‘ALTER DATABASE [+@dbName+]’)

EXEC (‘DENY VIEW ANY DATABASE TO [‘+@dbName’]’) –запрет просмотра баз данных

EXEC(‘USE [‘+@dbName+’]’)

EXEC dbo.sp\_changedbowner ‘@loginame=’+@dbName)

SET @currentUser +=1

END