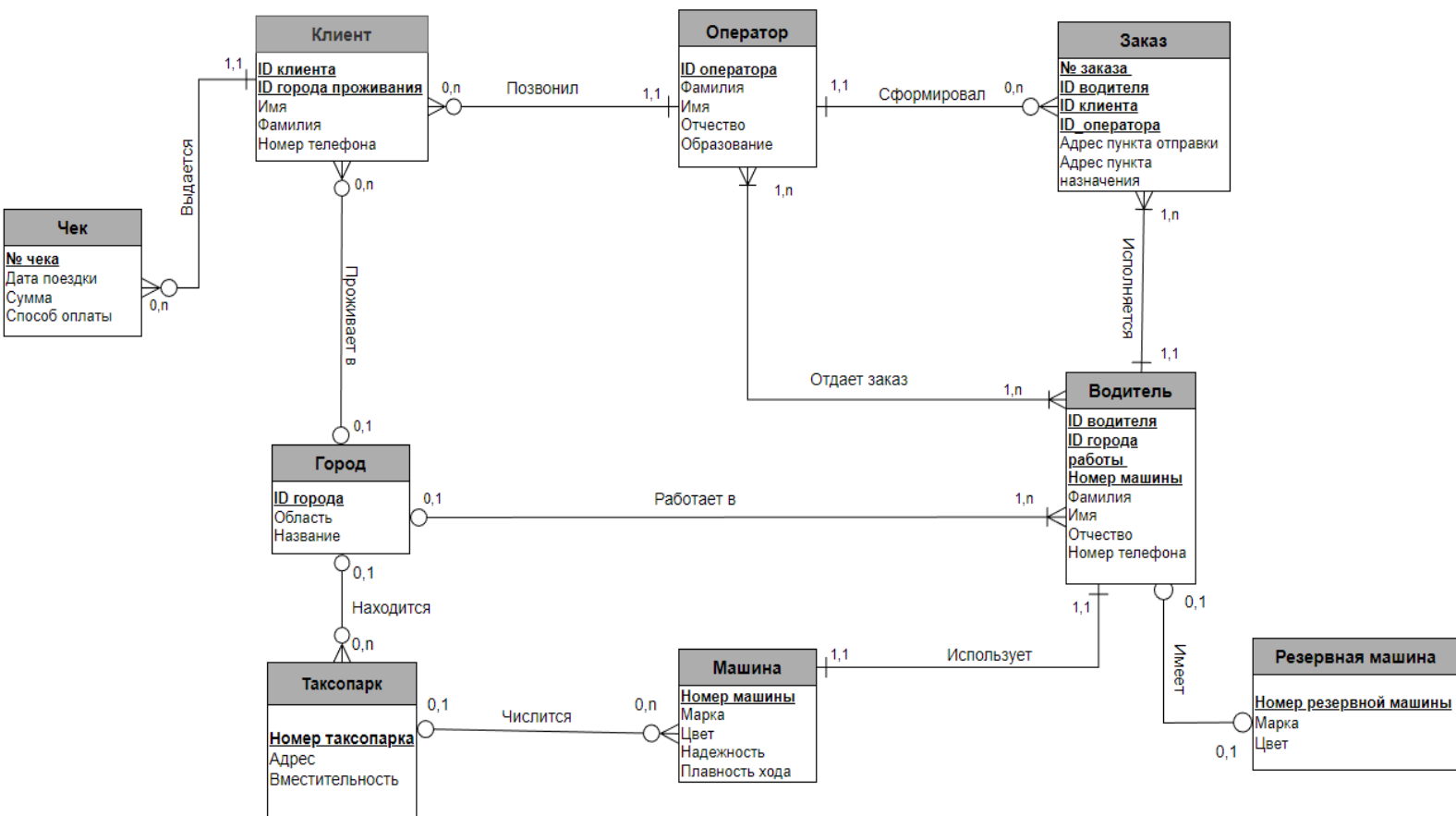


Задание 3.

Преобразование ER-диаграммы в физическую схему базы данных.

1. Предметная область: служба такси
2. ER-диаграмма предметной области сделана при помощи Draw.io.



3. Физическая схема реляционной базы данных

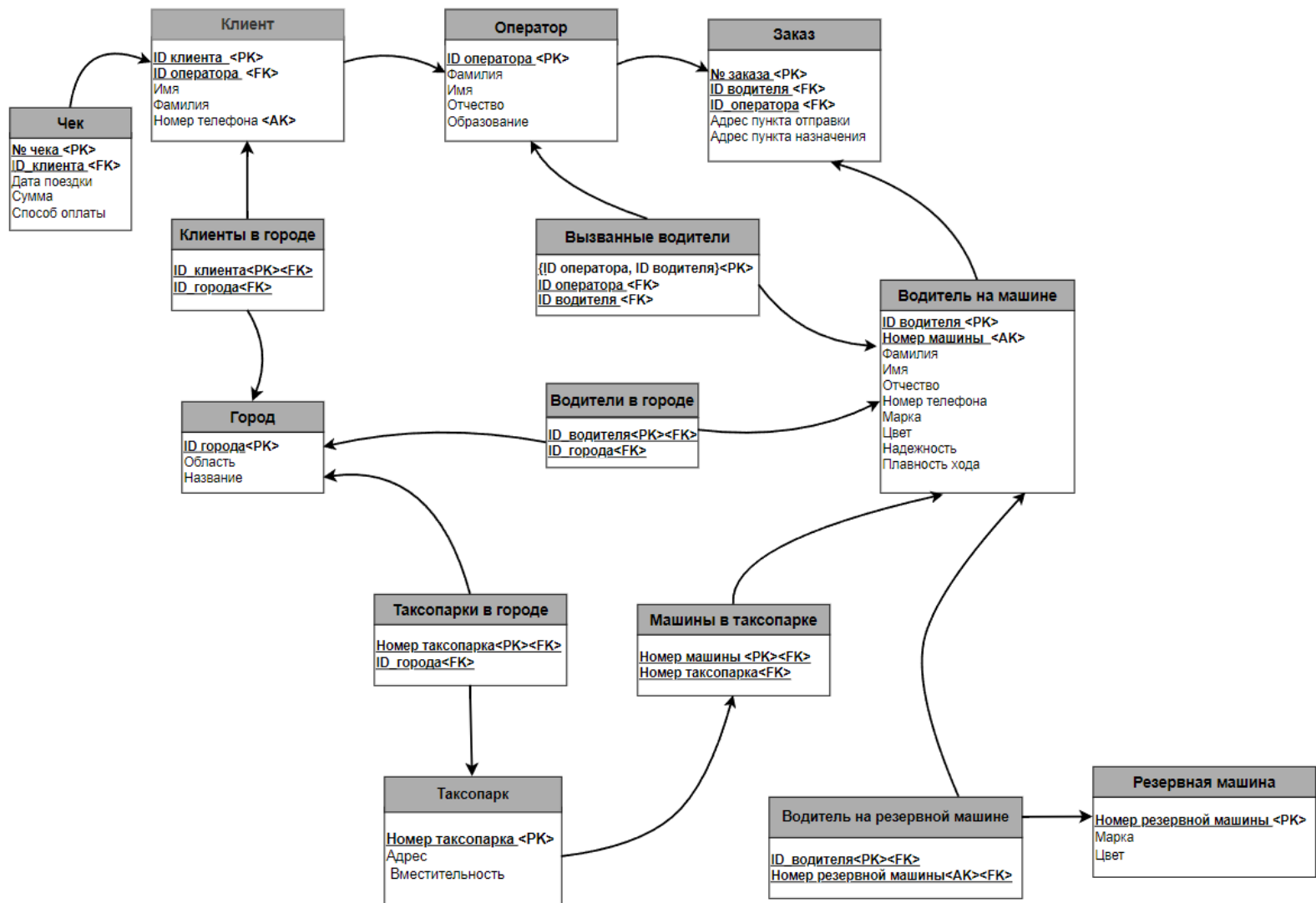
Используемые обозначения:

РК – первичный ключ

FK – внешний ключ

АК – альтернативный ключ

Стрелки – связи между отношениями.



4. Преобразования

Водитель-Машина

Степень связи – 1:1(один к одному)

Класс принадлежности экземпляров сущности Водитель – обязательный, сущности Машина – обязательный.

Водитель обязательно передвигается на одной и только одной машине, на каждой машине передвигается один и только один водитель.

ER-диаграмма сущностей Водитель и Машина трансформируется в одно реляционное отношение Водитель на машине.

Идентифицирующие свойства этих сущностей становятся первичным и альтернативным ключами в отношении Водитель на машине.

При этом, ID водителя становится первичным ключом, а Номер машины - альтернативным.

Водитель на машине-Резервная машина

Степень связи – 1:1(один к одному)

Класс принадлежности экземпляров сущности Водитель на машине – необязательный, сущности Резервная машина – необязательный.

Водитель может иметь в запасе одну резервную машину (может не иметь совсем), у каждой резервной машины может быть один водитель (может и не быть вовсе).

При преобразовании в реляционную схему в данном случае требуется уже не два, а три отношения – Водитель, Водитель на резервной машине, Резервная машина

Два из этих отношений соответствуют сущностям ER-диаграммы (Водитель и Резервная машина). Идентифицирующие свойства этих сущностей становятся первичными ключами соответствующих отношений. К этим двум отношениям добавляется третье отношение Водитель на резервной машине. В этом отношении вводятся два атрибута, соответствующие первичным ключам первых двух отношений (ID_водителя и Номер резервной машины, соответственно). Эти два атрибута становятся внешними ключами, при этом один из них (например, ID_водителя) становится первичным ключом отношения Водитель на резервной машине, а другой (например, Номер резервной машины) альтернативным ключом этого отношения.

Водитель на машине-Заказ

Степень связи – 1:n(один ко многим)

Класс принадлежности экземпляров сущности Заказ – обязательный, сущности Водитель – обязательный.

Каждый заказ выполняется одним и только одним водителем, каждый водитель обязательно выполняет один или несколько заказов.

Идентифицирующие свойства этих отношений становятся их первичными ключами (<PK>). Это соответственно, атрибут <ID_водителя> и атрибут <№ заказа>.

В отношении, соответствующее сущности с множественной связностью (Заказ), вводится атрибут ID_водителя, который становится внешним ключом(<FK>), ссылающимся на одноименный первичный ключ отношения Водитель.

Оператор – Заказ

Степень связи – 1:n(один ко многим)

Класс принадлежности экземпляров сущности Оператор – необязательный, сущности Заказ – обязательный.

Оператор может сформировать несколько заказов (может и ни одного), каждый заказ формируется одним и только одним оператором.

Идентифицирующие свойства этих сущностей становятся первичными ключами (<PK>) этих отношений (соответственно, атрибут <ID_оператора> и атрибут <№ заказа>).

В отношении Заказ вводится дополнительный атрибут <ID_оператора>, который назначается внешним ключом(<FK>), ссылающимся на одноименный первичный ключ отношения Оператор.

Оператор – Клиент

Степень связи – 1:n(один ко многим)

Класс принадлежности экземпляров сущности Клиент – обязательный, сущности Оператор – необязательный.

У клиента есть один и только один оператор, у каждого оператора может быть несколько клиентов (может и не быть вовсе).

Идентифицирующие свойства этих сущностей становятся первичными ключами (<PK>) этих отношений (соответственно, атрибут <ID_оператора> и атрибут <ID_клиента>).

В отношении Клиент вводится дополнительный атрибут <ID_оператора>, который назначается внешним ключом, ссылающимся на одноименный первичный ключ отношения Оператор.

Клиент – Чек

Степень связи – 1:n(один ко многим)

Класс принадлежности экземпляров сущности Чек – обязательный, сущности Клиент – необязательный.

Чек обязательно выдается одному и только одному клиенту, у каждого клиента может быть несколько чеков (может и не быть вовсе).

Идентифицирующие свойства этих сущностей становятся первичными ключами (<PK>) этих отношений (соответственно, атрибут <ID_клиента> и атрибут <№ чека>).

В отношении Чек вводится дополнительный атрибут <ID_клиента>, который назначается внешним ключом, ссылающимся на одноименный первичный ключ отношения Клиент.

Водитель на машине – Город

Степень связи – 1:n(один ко многим)

Класс принадлежности экземпляров сущности Водитель с машиной – необязательный, сущности Город – обязательный.

Каждый водитель может проживать в одном городе (может и не проживать), в каждом городе обязательно работает один или несколько водителей.

Создадим новое отношение «Водители в городе» и назначим: ID водителя – родительский и внешний ключ в отношении «Водители в городе», а ID города в отношении «Водители в городе» - внешний ключ. ID города и ID водителя в соответствующих отношениях – первичные ключи(<PK>).

Новое отношение создается потому, что иначе в отношении будут присутствовать транзитивные функциональные зависимости. То есть отношение не будет находится в 3-й нормальной форме и в нем неизбежно будут присутствовать вызванные такой функциональной зависимостью аномалии операций вставки, удаления и обновления.

Клиент – Город

Степень связи – 1:n(один ко многим)

Класс принадлежности экземпляров сущности Клиент – необязательный, сущности Город – необязательный.

Каждый клиент может проживать в одном городе (может и не проживать вовсе), в каждом городе проживает несколько клиентов (может и не проживать вовсе).

Создадим новое отношение «Клиенты в городе» и назначим: ID клиента – родительский и внешний ключ в отношении «Клиенты в городе», а ID города в отношении «Клиенты в городе» - внешний ключ. ID города и ID клиента в соответствующий отношениях – первичные ключи(<PK>).

Город - Таксопарк

Степень связи – 1:n(один ко многим)

Класс принадлежности экземпляров сущности Таксопарк – необязательный, сущности Город – необязательный.

Каждый таксопарк может находиться в одном городе (может и не находиться), в каждом городе может быть несколько таксопарков (может и не быть вовсе).

Создадим новое отношение «Таксопарки в городе» и назначим: Номер таксопарка – родительский и внешний ключ в отношении «Таксопарки в городе», а ID города в отношении «Таксопарки в городе» - внешний ключ. ID города и Номер таксопарка в соответствующий отношениях – первичные ключи(<PK>).

Таксопарк – Водитель на машине

Степень связи – 1:n(один ко многим)

Класс принадлежности экземпляров сущности Водитель на машине – необязательный, сущности Таксопарк – необязательный.

Каждый водитель на машине может числиться в одном таксопарке (может и не числиться вовсе), в каждом таксопарке может числиться несколько водителей на машинах (может не числиться ни одной).

Создадим новое отношение «Машины в таксопарке» и назначим: Номер машины – родительский и внешний ключ в отношении «Машины в таксопарке», а Номер таксопарка в отношении «Таксопарки в городе» - внешний ключ. Номер машины и Номер таксопарка в соответствующий отношениях – первичные ключи(<PK>).

Оператор – Водитель на машине

Степень связи –n:m(многие ко многим)

Класс принадлежности экземпляров сущности Оператор – обязательный, сущности Водитель – обязательный.

Каждый оператор обязательно отдает водителю один или несколько заказов, каждый водитель принимает один или несколько заказов от оператора.

Создадим новое отношение «Вызванные водители», первичным ключом которого станет составной атрибут {ID_оператора, ID_водителя}.

ID_оператора, ID_водителя – внешние ключи в отношении «Вызванные водители». ID_оператора, ID_водителя – первичные ключи в соответствующих отношениях (<PK>).

Новое отношение создается потому, что иначе в отношении будут присутствовать приводимые слева (неполные) функциональные зависимости атрибутов от составного первичного ключа. То есть отношение не будет находится во 2-й нормальной форме и в нем неизбежно будут присутствовать вызванные такой функциональной зависимостью аномалии операций вставки, удаления и обновления.