

Task 1

Так как каждая строка в таблице обязана быть уникальной, то приходим к тому, что отношение – это набор уникальных в совокупности атрибутов.

Пусть ключа нет, тогда это утверждение равносильно тому, что нет такого набора атрибутов, чтобы они являлись уникальными в совокупности (то есть такими, по которым можно однозначно идентифицировать наблюдение). Приходим к противоречию с первым утверждением о том, что каждое наблюдение обязано быть уникальным.

Значит, каждое отношение (множество атрибутов, которые в совокупности должны быть уникальны) имеет ключ. Ключ также может быть и составной (из нескольких атрибутов).

Task 2

Task2.1

Book(ISBN:string, name:string, year:int, author:string, number_of_pages:int, publisher_name:string)

**ISBN – однозначно определяет книгу*

Copy(number:int, shelf_position:int, book_isbn:int)

**считаем, что номер копии не определяет копию. А если взять вместе с ISBN, то определяется однозначно экземпляр*

Publisher(name:string, address:string)

**Издательство определяется именем*

Category(name:string, super_category_name:string)

**так как иерархия, то имя категории однозначно определяет родителя*

Book_Category_belongs (category_name:string, ISBN:int)

**не знаю точно, как лучше назвать, но суть в том, что эта таблица показывает связь между категорией и книгой. Любая пара значений category_name, ISBN может храниться там (нельзя подчеркнуть один ключ, так как тогда что-то второе должно определяться однозначно, а это не так)*

Reader(reader_number:int, first_name:string, last_name:string, address:string, birthday:date)

**читатель определяется уникальным id*

Booking(reader_number:int, copy_num:int, isbn_of_copy:int, return_date:date)

**это тоже связь многое ко многим между копией и читателем. Пара читатель – копия определяют однозначно дату. То есть, взятие книги определяется*

Task 2.2

1)

Страна({название:string})

Город({название:string, название_страны:string})

Улица({название:string, название_города:string, название_страны:string })

Дом({номер:int, название_улицы:string, название_города:string, название_страны:string })

Квартира({номер:int, номер_дома:int, название_улицы:string, название_города:string, название_страны:string })

**Будем считать, что для двух разных стран города могут повторяться, и для двух разных городов – улицы могут повторяться. Получаем: город определяется названием и страной, улица – городом (который определяется названием и страной) и тд.*

2)

Команда({название:string })

Арбитр({имя:string})

Матч({название_команды1:string, название_команды2:string, имя_арбитра:string})

**В этом случае без матча не обойтись, если мы считаем, что каждая команда может сыграть с другой командой несколько раз, и при этом может быть сменен арбитр. Все эти условия определяют, что ключ должен быть 3-х частей (к1, к2, арбитр)*

Логично, что если команды должны переиграть (то есть не меняются ключи название_команды1, название_команды2, то арбитр должен быть уже другой)

3)

Женщина({id:int, id_матери:int, id_отца:int})

Мужчина({id:int, id_матери:int, id_отца:int})

**Можно считать, что каждый человек имеет id матери и отца, но он не обязан иметь ребенка.*

Task 2.3

Атрибут({название:string, ключевой_или_нет:bool, имя_сущности:int})

**Ключ составной, так как два одинаковых атрибута могут быть у разных сущностей.*

Сущность({название:string})

Связь({название:string, тип:string, название сущности1:string, название сущности2:string, роль:string})

**То есть связь определяется двумя сущностями*

Task 3

Task3.1

City({region:string, name:string})

Station({name:string, city_region:string, city_name:string, #tracks:int })

**Станция определяется ее именем (считаем, что в разных городах станции все равно не повторяются, иначе – надо было бы сделать составной ключ)*

Train({TrainNr:int, length:int, start_station_name:string, end_station_name:string})

Connected({prev_station_name:string, trainNr:int, arrival:time, departure:time, next_station_name:string})

**Считаем, что любой поезд один раз уезжает с конкретной станции (было бы логично так делать), поэтому пара prev_station_name и trainNr будет ключом (так же можно было взять пару trainNr и next_station).*

Task 3.2

Station({statNr:int, name:string})

Room({roomNr:int, #Beds:int, stationNr:int})

**Комната должна определяться не только ее номером, но и местом.*

Patient({patientNr:int, name:string, disease:string, PersNr:int, roomNr:int, from:string, to:string})

**Для каждого admission только 1 пациент, поэтому он и ключ.*

StationPersonell({PersNr:int, #Name:string, stationNr:int})

Caregiver({persNr:int, qualification:string, Name:string, StatNr:int})

Doctor({persNr:int, area:string, rank:string, Name:string, StatNr:int })