

*Санкт-Петербургский Национальный Исследовательский Университет
Информационных Технологий, Механики и Оптики*

ПИиКТ

*Лабораторная работа 4
по дисциплине
«Информационные системы и базы данных»*

Выполнили: Студенты группы Р33113

Мансуров Б.Б.

Преподаватель: Николаев В.В.

Санкт-Петербург

2020г

Задание

Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

- опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
- приведите отношения в 3NF (как минимум). Постройте схему на основе NF (как минимум). Постройте схему на основе полученных отношений;
- опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе NF;
- преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF;
- какие денормализации будут полезны для вашей схемы? Приведите подробное описание;

Вариант 618

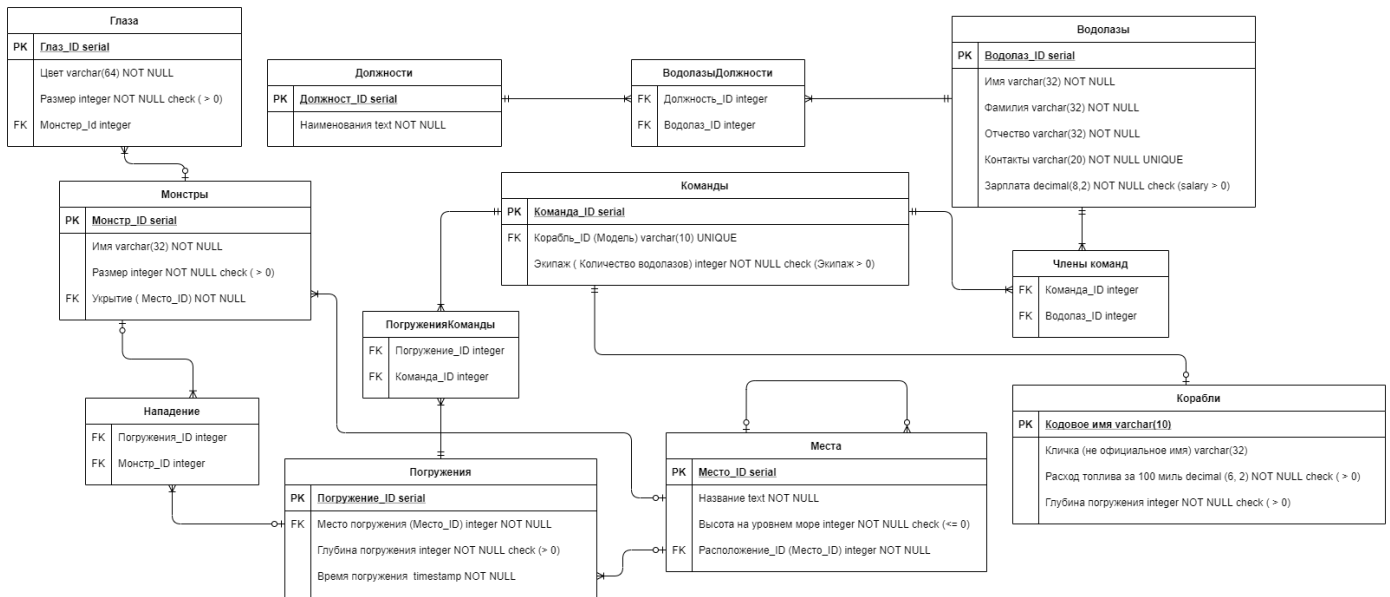
Два темных неподвижных круга - чем еще они могли быть? Они придавали каждому погружению элемент риска: Чудовище в любой момент могло покинуть свое укрытие и, распугивая рыбешку, ринуться за более крупной добычей. И никто не убедил бы Бобби и Дэвида, что осока не скрывает ничего более опасного, чем, скажем, украденный велосипед... Тем не менее дно Хрустального источника оставалось недостижимым.

Выполнение

Список сущностей и их классификация

1. Водолазы – стержневая сущность
2. Команда – ассоциативная сущность
3. Корабли – стержневая сущность
4. Монстры – ассоциативная сущность
5. Места – ассоциативная сущность
6. Расположения – стержневая сущность
7. Координаты – характеристическая сущность
8. Погружения – ассоциативная сущность
9. Должности – стержневая сущность
10. Глаза – характеристическая сущность
11. Нападения – ассоциативная сущность

Даталогическая модель



Опишите функциональные зависимости для отношений полученной схемы

Водолазы

- Водолаз_id -> Имя
- Водолаз_id -> Фамилия
- Водолаз_id -> Отчество
- Водолаз_id -> Контакты
- Водолаз_id -> Зарплата
- Контакты (номер телефона) -> Зарплата
- Контакты (номер телефона) -> Имя
- Контакты (номер телефона) -> Фамилия
- Контакты (номер телефона) -> Отчество

Должности

- Должность_id -> Наименование

Корабли

- Кодовое имя -> имя (неофициальное)
- Кодовое имя -> Расход топлива на 100 миль
- Кодовое имя -> Глубина погружения
- Имя -> Расход топлива

- *Имя -> Глубина погружения*

Команды

- *Команда_id -> Корабль_id*
- *Команда_id -> Экипаж*
- *Корабль_id -> Экипаж*

Монстры

- *Монстр_id -> Имя*
- *Монстр_id -> Размер*
- *Монстр_id -> Укрытие*

Глаза

- *Глаз_id -> Цвет*
- *Глаз_id -> Размер*
- *Глаз_id -> Монстр_id*

Место

- *Место_id -> Название*
- *Место_id -> Высота над уровнем море*
- *Место_id -> Расположение (Место_id)*

Погружение

- *Погружение_id -> Место погружения (Место_id)*
- *Погружение_id -> Глубина погружения*
- *Погружение_id -> Время погружения*

Остальных таблиц не рассматриваем, так как они служебные таблицы для many – to – many и они уникальны только в паре

Приведите отношения в 3НФ (как минимум). Постройте схему на основе полученных отношений

1НФ: отношение на пересечение каждой строки и столбца – одно значение. Все мои таблицы удовлетворяют данным условиям.

2НФ: отношение в 1НФ, каждый атрибут которого, отличный от атрибута первичного ключа, является полностью функционально независимым от любого потенциального ключа. То есть: нет частичных зависимостей от первичного и потенциальных ключей

Чтобы привести к 2НФ надо убрать частичные зависимости

- Удалить частично – зависимые атрибуты
- Новое отношение: удаленные атрибуты плюс копия детерминанты

Все таблицы уже удовлетворяют условиям 2НФ, а, следовательно, преобразование не требуется.

3НФ: отношение в 1НФ и 2НФ, при этом не имеет атрибутов, которые не входят в первичный ключ и находятся в транзитивной функциональной зависимости от первичного ключа. (Должны избавиться от атрибутов, которые зависят не от первичного ключа).

Опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3НФ

Изменений в таблицах, а, следовательно, и в функциональных зависимостях не произошло, так как никаких преобразований не было.

Преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF

Таблицы, которые не соответствуют 3НФ в моей базе данных имеется. Корабли, Команды, Водолазы. Так как в этих таблицах имеется функциональная зависимость между не ключевыми атрибутами на пример таблица Корабли:

Корабли	
PK	Кодовое имя varchar(10)
	Имя varchar(32)
	Расход топлива за 100 миль decimal (6, 2) NOT NULL check (> 0)
	Глубина погружения integer NOT NULL check (> 0)

Кодовое имя -> Имя

Кодовое имя -> Расход топлива

Кодовое имя -> Глубина погружения

Имя -> Расход топлива

Имя -> Глубина погружения

Следовательно,

для того чтобы привести таблицу к 3НФ нужно разбить ее на:

Кодовое имя -> Имя в новую таблицу

Корабли_Модель_Имя	
PK	Кодовое имя varchar(10)
	Имя varchar(32)

Кодовое имя -> Расход топлива

Кодовое имя -> Глубина погружения

Корабли	
PK	Кодовое имя varchar(10)
	Расход топлива за 100 миль decimal (6, 2) NOT NULL check (> 0)
	Глубина погружения integer NOT NULL check (> 0)

хотя интуиция подсказывает, что это лишнее разбиение, совсем не улучшает проект базы данных. На этот счет есть более строгое определение 3НФ, которое учитывает, что в таблице может быть несколько возможных ключей. Аналогично для таблицы Команды.

Таблица находится в нормальной форме Бойса – Кодда (НФБК), если и только если любая функциональная зависимость между его полями сводится к полной функциональной зависимости от возможного ключа.

В соответствии с этой формулировкой таблицы Корабли и Команды находятся в НФБК.

Какие денормализация будут полезны для вашей схемы

Водолазы

- Водолаз_id -> Имя
- Водолаз_id -> Фамилия
- Водолаз_id -> Отчество
- Водолаз_id -> Контакты
- Водолаз_id -> Зарплата
- Контакты (номер телефона) -> Зарплата
- Контакты (номер телефона) -> Имя
- Контакты (номер телефона) -> Фамилия
- Контакты (номер телефона) -> Отчество

Имеется ввиду что контакты (номер телефона) unique

Если мы будем следовать нормальным формам, то следует декомпонировать таблицу Водолазы. Но думаю лучше будет оставить все как есть, потому что декомпозиция требует памяти и скорость выполнения существенно понизится из-за того, что нам потребуется соединение таблиц для того чтобы получить контакты водолаза.

Вывод:

Много времени потратил на то чтобы понять сами функциональные зависимости потому что на разных источниках правила описывают по-разному и все объясняют на таблицах где все интуитивно понятно. Оказывается, если ты знаешь предметную область, то нормализация существенно упрощается.

При выполнении лабораторной работы я получил первичные знания о нормализации, функциональных и многозначных зависимостях. На своем опыте нормализовал таблицы и получил навыки нормализации таблиц баз данных.