Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет

ИТМО»

**Факультет Программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 3**

по дисциплине

‘Информационные системы и базы данных’

Вариант №- строковый

*Выполнил:*

Студент группы P3132

Копалина Майя Алексеевна

*Проверил:*

Харитонова Анастасия Евгеньевна

г. Санкт-Петербург

2023 год

Задание

Лабораторная работа #3

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 основеNF (как минимум). Постройте схему на основе полученных отношений;
* опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основеNF;
* преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF;

Если ваша схема находится уже в BCNF, докажите это.

Какие денормализации будут полезны для вашей схемы? Приведите подробное описание;

Придумайте функцию, связанную с вашей предметной областью, согласуйте ее с преподавателем и реализуйте на языке PL/pgSQL.

|  |
| --- |
|  |
| Текст задания: |
|  |  |
|  | У Тима похолодело внутри, и тут голова ринулась вперед, пасть раскрылась. |
|  | Зубы ударились о металл, Тим ощутил горячее, зловонное дыхание зверя и увидел толстый язык, |
|  | просовывающийся в машину сквозь отверстие на месте разбитого стекла. |
|  | Динозавр начал шлепать мокрым языком по обшивке кабины... |
|  | капли горячей пенистой слюны упали на кожу Тима... затем тиранозавр взревел... оглушительно... |
|  |  |

|  |
| --- |
|  |
| Описание: |
|  |  |
|  | Существует человек, имеющий определенное имя. |
|  | Он сидит в машине, которая может быть любой марки. |
|  | Тим наблюдает за динозавром. Они встретились. |
|  | У каждого динозавра есть пасть, в которой находятся зубы и язык. |
|  | Зубы могут быть грязные и почищенные, а язык может иметь разную величину. |
|  | Во рту выделяется слюна, которая в зависимости от состояния зверя может быть обильной и умеренной. |
|  | Язык касается той же самой машины. |
|  |  |

**Список сущностей**

|  |
| --- |
| Стержневые: |
|  | 1. Человек |
|  | 2. Динозавр |
|  | 3. Машина |

|  |
| --- |
| Ассоциативные: |
|  | 4. Встреча |
|  | 5. Касание |

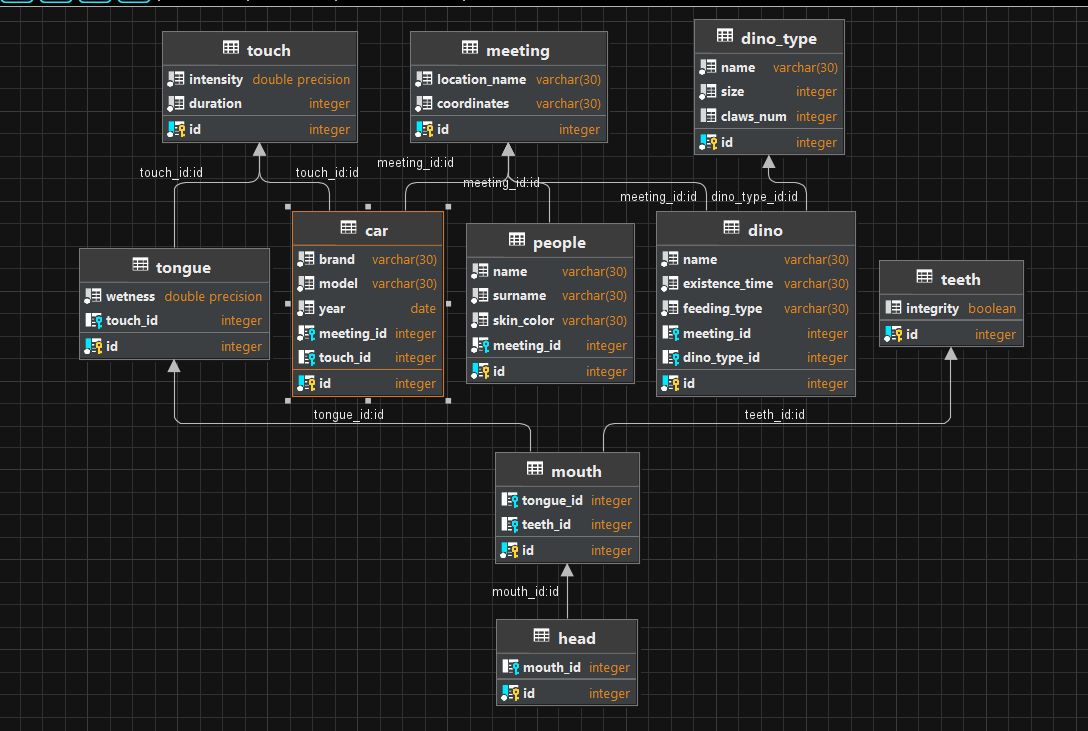
|  |
| --- |
| Характеристики: |
|  | 6. Голова - характеристика сущности Динозавр |
|  | 7. Пасть - характеристика сущности Голова |
|  | 8. Зубы - характеристика сущности Пасть |
|  | 9. Язык - характеристика сущности Пасть |
|  | 10. Вид динозавров - характеристика сущности Динозавр |

|  |  |
| --- | --- |
|  | |
| Связи: | |
|  | | динозавр 1-1 голова |
|  | | пасть 1-1 язык |
|  | | пасть1-М зубы |
|  | | динозавр 1-М вид динозавра (тиранозавр) |
|  | | машина М-М язык (через ассоциацию касание) |
|  | | человек М-М динозавр (через ассоциацию встреча) |
|  | | машина М-М динозавр (через ассоциацию встреча) |
|  | |  |
| Атрибуты сущностей: | |
|  | | Человек - имя, фамилия, цвет кожи, ID, встреча | |
|  | | Динозавр - название, время существования, тип питания, место обитания, ID, встреча | |
|  | | Машина - марка, модель, год выпуска, ID, встреча, касание | |
|  | | Вид динозавров - название, размер, количество когтей, ID | |
|  | | Встреча - название локации, координаты, ID | |
|  | | Касание - интенсивность, продолжительность, ID | |
|  | | Зубы - целостность, ID | |
|  | | Пасть - язык, зубы, ID | |
|  | | Язык - влажность, касание, ID | |
|  | | Голова - пасть, ID | |
|  | | Язык - касание, влажность, ID | |

**Инфологическая модель**

****

**Даталогическая модель**

****

## Функциональные зависимости для отношений схемы (минимальное множество)

People: id – (name, surname, skin\_color, meeting\_id)

Car: id – (brand, model, year, meeting\_id, touch\_id)

Dino: id – (name, existence\_time, feeding\_type, meeting\_id)

Touch: id – (intensity, duration)

Meeting: id – (location\_name, coordinates)

Dino\_type: id – (name, size, claws\_num)

Tongue: id – (wetness, touch\_id)

Teeth: id – (integrity)

Mouth: (tongue\_id, teeth\_id) – ()

Head: (mouth\_id) – ()

**Нормализация**

Пройдёмся эволюционно:

Первая нормальная форма (1НФ):

На пересечении строки и столбца содержится только одно значение.

Это ключевое определение реляционной базы данных и при проектировании первой лабораторной работы это было учтено. То есть все атрибуты атомарны, и нет повторяющихся групп. Следовательно, изменения не требуются.

Вторая нормальная форма (2НФ):

1. Отношения находятся в 1НФ;
2. Каждый атрибут, не входящий в первичный ключ, полностью функционально зависит от первичного ключа

В каждой таблице атрибуты, не включенные в первичный ключ, непосредственно, зависят от него. Следовательно, схема удовлетворяет условиям 2НФ и изменения не требуются.

Третья нормальная форма (3НФ):

1. Отношения находятся в 2НФ;
2. Нет атрибутов, не входящих в первичный ключ, которые находятся в транзитивной зависимости от первичного ключа.

Транзитивных зависимостей нет. Моя модель удовлетворяет 3НФ, так как все неключевые атрибуты зависят только от первичных ключей, и не содержат транзитивных зависимостей.

**BCNF**

Отношения находятся в BCNF, так как отношения не имеют двух и более потенциальных ключей, не имеют общих атрибутов, ключи не являются составными.

Если для каждой функциональной зависимости X →Y, X является суперключом. Моя модель удовлетворяет BCNF, так как для всех функциональных зависимостей X является суперключом.

**Денормализация**

Объединение связанных таблиц:

Объединение таблиц может уменьшить количество операций JOIN и ускорить обработку запросов. Например, можно рассмотреть объединение таблиц dino и dino\_type, если часто запрашиваются данные о динозавре и его типе.

Добавление избыточных атрибутов:

Добавление избыточных атрибутов может улучшить производительность запросов. Например, можно добавить атрибут количества зубов, чтобы их не пересчитывать каждый раз. Это позволит избежать операций подсчёта при каждом запросе, однако необходимо его обновлять при добавлении или удалении зубов.

**Строковая функция на языке PL/pgSQL**

Строковые функции используются для возвращения уже отредактированного значения. Другими словами, например, у Вас в колонке все значения состоят более чем из 20 символов, а Вам нужно возвратить всего первые 5 символов, это можно сделать с помощью, так называемых строковых функций.

create table log\_table  
(  
 time DATE,  
 type CHARACTER VARYING(200) NOT NULL,  
 primary\_key INT  
);  
CREATE OR REPLACE FUNCTION *insert\_log\_table*()  
RETURNS TRIGGER AS $$  
BEGIN  
 IF TG\_OP = 'INSERT' THEN  
 INSERT INTO log\_table (time,type,primary\_key)  
 VALUES (*NOW*(),'CREATE',new.id);  
 ELSE  
 INSERT INTO log\_table (time,type,primary\_key)  
 VALUES (*NOW*(),'UPDATE',new.id);  
 END IF;  
 RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
  
CREATE TRIGGER before\_insert\_car  
BEFORE INSERT OR UPDATE ON car  
FOR EACH ROW  
EXECUTE FUNCTION *insert\_log\_table*();  
INSERT INTO meeting VALUES (35, 'Северо-запад', '(4,5)'),  
 (48, 'Юго-восток', '(9,0)');  
INSERT INTO touch VALUES (1, 6.7, 10),  
 (2, 1.8, 5);  
  
  
INSERT INTO car VALUES (1, 'BMW', 'X7', '2023-03-16', 35, 1),  
 (2, 'Mercedes', 'X6', '2022-12-05',48, 2);

**Реализация модели на SQL (1-ая лабораторная)**

insert into s368346.testbd(id**,**value) values (**1,**'test')**;**select \* from s368346.testbd**;**CREATE TABLE IF NOT EXISTS people  
(  
 id serial PRIMARY KEY**,** name varchar(**30**) NOT NULL**,** surname varchar(**30**) NOT NULL**,** skin\_color varchar(**30**) NOT NULL  
)**;**CREATE TABLE IF NOT EXISTS dino  
(  
 id serial PRIMARY KEY**,** name varchar(**30**) NOT NULL**,** existence\_time varchar(**30**) NOT NULL**,** feeding\_type varchar(**30**) NOT NULL**,** dino\_type\_id int REFERENCES dino\_type(id)  
)**;**CREATE TABLE IF NOT EXISTS car  
(  
 id serial PRIMARY KEY**,** brand varchar(**30**) NOT NULL**,** model varchar(**30**) NOT NULL**,** year date NOT NULL  
)**;**CREATE TABLE IF NOT EXISTS dino\_type  
(  
 id serial PRIMARY KEY**,** name varchar(**30**) NOT NULL**,** size int NOT NULL**,** claws\_num int CHECK (claws\_num > **0**)  
)**;**CREATE TABLE IF NOT EXISTS meeting  
(  
 id serial PRIMARY KEY**,** location\_name varchar(**30**) NOT NULL**,** coordinates varchar(**30**) NOT NULL**,** people\_id int REFERENCES people(id)**,** dino\_id int REFERENCES dino(id) **,** car\_id int REFERENCES car(id)  
)**;**CREATE TABLE IF NOT EXISTS touch  
(  
 id serial PRIMARY KEY**,** intensity float NOT NULL**,** duration int NOT NULL**,** tongue\_id int REFERENCES tongue(id)**,** car\_id int REFERENCES car(id)  
)**;**CREATE TABLE IF NOT EXISTS mouth  
(  
 id serial PRIMARY KEY**,** tongue\_id int REFERENCES tongue(id)**,** teeth\_id int REFERENCES teeth(id)  
)**;**CREATE TABLE IF NOT EXISTS teeth  
(  
 id serial PRIMARY KEY**,** integrity boolean  
)**;**CREATE TABLE IF NOT EXISTS tongue  
(  
 id serial PRIMARY KEY**,** wetness float NOT NULL  
)**;**CREATE TABLE IF NOT EXISTS head  
(  
 id serial PRIMARY KEY**,** mouth\_id int REFERENCES mouth(id) ON DELETE CASCADE  
)**;**INSERT INTO car VALUES (**1,** 'BMW'**,** 'X7'**, 2023**-**03**-**16**)**,** (**2,** 'Mercedes'**,** 'X6'**, 2022**-**12**-**05**)**;**INSERT INTO dino VALUES (**11,** 'Динозавр А'**,** 'юрский период'**,** 'травоядный')**,** (**12,** 'Динозавр Е'**,** 'меловой период'**,** 'хищник')**;**INSERT INTO dino\_type VALUES (**33,** 'Диплодок'**, 1200, 5**)**,** (**44,** 'Бронтозавр'**, 3000, 4**)**;**INSERT INTO head VALUES (**55**)**,** (**45**)**;**INSERT INTO meeting VALUES (**35,** 'Северо-запад'**,** '(4,5)')**,** (**48,** 'Юго-восток'**,** '(9,0)')**;**INSERT INTO mouth VALUES (**10**)**,** (**19**)**;**INSERT INTO people VALUES (**18,** 'Саша'**,** 'Иванова'**,** 'светлая кожа')**,** (**20,** 'Оля'**,** 'Решетняк'**,** 'смуглая кожа')**;**INSERT INTO teeth VALUES (**9,** true)**,** (**99,** true)**;**INSERT INTO tongue VALUES (**1, 56.9**)**,** (**2, 61.8**)**;**INSERT INTO touch VALUES (**1, 6.7, 10**)**,** (**2, 1.8, 5**)**;**/\* drop table people;  
drop table dino;  
drop table car;  
drop table dino\_type;  
drop table head;  
drop table meeting;  
drop table mouth;  
drop table teeth;  
drop table tongue;  
drop table touch;

\*/

**Вывод**

При выполнении лабораторной работы я узнала понятие нормализации и денормализации. Научилась определять функциональные зависимости модели, писать функции на языке PL/pgSQL.