

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Информационные системы и базы данных»

Лабораторная работа №1

Вариант 183

Студент

Кузнецов М. А.

P33131

Преподаватель

Николаев В. В.

Описание задания

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 полученных отношений.
- Опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум).
- Преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF.
- Какие денормализации будут полезны для вашей схемы? Приведите подробное описание

Описание предметной области

Джизирак послушно проследовал за Ярланом Зеем в здание. Свой мозг в эти минуты он мог бы сравнить с губкой -- все впитывающей и ничего не подвергающей сомнению. Какое-то воспоминание или даже всего лишь отдаленное эхо воспоминания предупреждало его о том, что именно должно сейчас вот произойти, и он знал, что в былые времена при виде этого он сжался бы от ужаса. Теперь же он совсем не испытывал страха. Он не только сознавал себя под защитой понимания того, что все здесь происходящее -- нереально, но и присутствие Ярлана Зея казалось неким талисманом против любых опасностей, которые могли бы ему встретиться.

Список сущностей

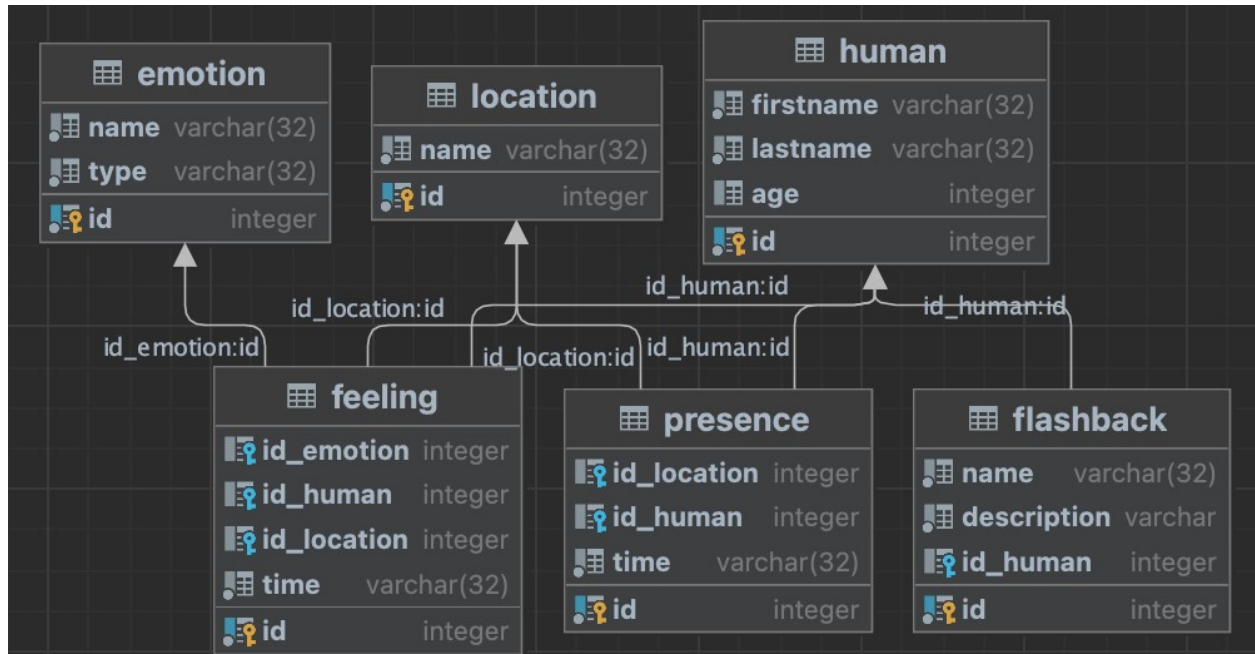
Стержневые:

- Локация – название
- Человек – *имя, фамилия, возраст*
- Эмоция – *название, тип (положительная, нейтральная,*

негативная) Характеристические:

- Воспоминание – *название, кому принадлежит, о чем*
- Нахождение – *локация, кто находился, когда* Ассоциативные:
- Чувства – *кто, какие эмоции испытывает, когда*

Даталогическая модель



Реализация модели на SQL

```

CREATE TABLE Human
(
    ID          SERIAL PRIMARY KEY,
    Firstname   VARCHAR(32) NOT NULL,
    Lastname    VARCHAR(32) NOT NULL,
    Age         INTEGER DEFAULT 0
);

CREATE TABLE Location
(
    ID          SERIAL PRIMARY KEY,
    Name        VARCHAR(32) NOT NULL
);

CREATE TABLE Emotion
(
    ID          SERIAL PRIMARY KEY,
    Name        VARCHAR(32) NOT NULL,
    Type        VARCHAR(32) NOT NULL
);

CREATE TABLE Feeling
(
    ID          SERIAL PRIMARY KEY,
    ID_Emotion  INTEGER REFERENCES Emotion,
    ID_Human    INTEGER REFERENCES Human,
    ID_Location INTEGER REFERENCES Location,
    Time        VARCHAR(32) NOT NULL
);

CREATE TABLE Flashback
(
    ID          SERIAL PRIMARY KEY,
    Name        VARCHAR(32) NOT NULL,
    Description  VARCHAR NOT NULL,
    ID_Human    INTEGER REFERENCES Human
);
    
```

```
);

CREATE TABLE Presence
(
    ID          SERIAL PRIMARY KEY,
    ID_Location INTEGER REFERENCES Location,
    ID_Human    INTEGER REFERENCES Human
);
```

```
INSERT INTO human (firstname, lastname, age)
VALUES ('Ярлан', 'Зей', 110),
       ('Джизирак', 'Каризижд', 40),
       ('Максим', 'Кузнецов', 19);

INSERT INTO location (name)
VALUES ('Капитолий'),
       ('Университет ИТМО');

INSERT INTO presence (id_location, id_human, time)
VALUES (1, 1, 'Беседа Ярлана Зея и Джизирака в Капитолее 8 сентября'),
       (1, 2, 'Беседа Ярлана Зея и Джизирака в Капитолее 8 сентября');

INSERT INTO flashback (id_human, name, description)
VALUES (2, 'Посещение Капитолия', 'Ужасные воспоминания о посещении
Джизираком Капитолия.');
```

```
INSERT INTO emotion (name, type)
VALUES ('Ужас', 'NEGATIVE'),
       ('Страх', 'NEGATIVE'),
       ('Сомнение', 'NEUTRAL');
```

```
INSERT INTO feeling (id_human, id_emotion, id_location, time)
VALUES (2, 1, 1, 'Во время беседы с Ярланом Зеем'),
       (2, 2, 1, 'Во время беседы с Ярланом Зеем'),
       (2, 3, 1, 'Во время беседы с Ярланом Зеем');
```

Минимальное множество функциональных зависимостей

Emotion:

id → name

id → type

Presence:

id → id_location

id → id_human

id → time

Location:

id → name

Feeling:

id → id_emotion

id → id_human

id → id_location

id → time

Human:

id → firstname

id → lastname

id → age

Flashback:

id → id_human

id → name

id → description

Соответствие нормальным формам

- *Первая нормальная форма: используются только скалярные значения, все атрибуты простые, повторений строк нет благодаря наличию первичных ключей*
- *Вторая нормальная форма: отношения находятся в 1НФ, и все атрибуты зависят от первичного ключа целиком, а не от его части*
- *Третья нормальная форма: отношения находятся во 2НФ, и каждый неключевой атрибут зависит только от первичного ключа, не зависит от других неключевых атрибутов*
- *Нормальная форма Бойса-Кодда: отношения находятся в 3НФ, и ключевые атрибуты не зависят от неключевых атрибутов*

Денормализация

Например, мы можем объединить таблицы “Emotion” и “Feeling”, то есть сущность “Feeling” получит вместо атрибута “id_emotion” сразу два поля сущности “Emotion” – “name” и “type”. Это незначительно может ускорить производительность. Но нарушит сразу 3-ю НФ, и вот почему:

- **3НФ:** отношения находятся во 2НФ, и каждый неключевой атрибут зависит только от первичного ключа, не зависит от других неключевых атрибутов
Не-а. Получится так, что теперь поле «type» будет зависеть от названия эмоции в этой таблице, а никак не от ключа. То есть, если у нас в этой таблице будет несколько прецедентов, в которых два или более человека испытали, например, «Страх», то так как «Страх» это «NEGATIVE», то поле «type» будет зависеть лишь от значения в атрибуте «name», но никак не от ключа. То есть, там, где появляется «Страх», всегда будет «NEGATIVE», и никак иначе.

Выводы

При выполнении лабораторной работы я:

- Познакомился с нормальными формами, применимыми к БД
- Попрактиковался в нормализации имеющейся БД
- Узнал, что такое денормализация и предложил идею по ее применению к своей БД