Отчет по лабораторной работе №1.

Вариант 0900909889899. Раевский Григорий, Р3121.

1. Текст задания.

В редких и трудно предвидимых случаях Шут буквально вверх дном переворачивал город какой-нибудь своей проделкой, которая могла быть не более чем тонко задуманной дурацкой шуткой или же рассчитанным выпадом против популярного в данный момент убеждения, а то и всего образа жизни. Принимая все это во внимание, можно было утверждать, что титул <шут> оказался в высшей степени удачным. В свое время, еще когда существовали короли и их дворы, шуты решали именно такие задачи и преследовали те же цели.

2. Описание предметной области.

Королевство состоящее из сущностей, выделенных из текста задания(а так же добавленных мною). Состоит из королей, шутов, дворян, крестьян и их возможных местоположений.

3. Список сущностей и их классификация.

* cities – является стержневой сущностью и описывает конкретный город.
* human – является стержневой сущностью, она описывает конкретного человека(имя, возраст, пол, рост, вес).
* Royal\_court – является ассоциативной сущностью, она связывает подчиненных и короля.
* Subordinate – является ассоциативной сущностью, определяющей связь между хозяином и подчиненным.
* Skills, statuses – являются характеристическими сущностями, они нужны для дополнительного описания каждого человека

4. Выводы

Во-первых, составление базы данных на основе отрывка из научно-фантастического романа оказалось не самой простой задачей. Мне не хватало того отрывка текста, так что пришлось придумать дополнительные сущности и атрибуты.

Во-вторых, в процессе создания базы данных возникли сложности с созданием внешних ключей и организацией связей 1-1,1-м и т.д. Это было связано с непониманием ошибок, которые выдает PostgreSQL. Позже эти проблемы решились (но это не точно).

Запросы

1. SELECT h1.name AS king\_or\_queen, h2.name AS subordinate\_name

FROM subordinates s

JOIN human h1 ON s.person\_id = h1.id

JOIN human h2 ON h2.id = ANY(s.subordinate\_ids)

1. SELECT human.name, skills.skill

FROM human

JOIN royal\_court ON human.originlocationid = royal\_court.locationid

JOIN skills ON human.id = skills.humanid

WHERE royal\_court.king\_or\_queen IN ('King', 'Queen');

1. SELECT h.name, h.date\_of\_birth, c.city AS city

FROM human h

JOIN subordinates s ON h.id = ANY(s.subordinate\_ids)

JOIN royal\_court r ON s.person\_id = r.court\_id

JOIN cities c ON h.originlocationid = c.id

WHERE r.court\_id = 1;

1. Сколько человек родилось в каждом городе:

SELECT cities.city AS city, COUNT(human.id) AS born\_here

FROM human

JOIN cities ON human.originlocationid = cities.id

GROUP BY city

ORDER BY born\_here ASC;

1. Вывести города, где родилось> 2 чел(создать представление ):

CREATE VIEW born\_more\_than\_2 AS

SELECT cities.city AS city, COUNT(human.id) AS born\_here

FROM human

JOIN cities ON human.originlocationid = cities.id

GROUP BY city

Having count(human.id) >=2

ORDER BY born\_here ASC;