

Федеральное государственное автономное образовательное учреждение высшего
образования Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Базы данных

Лабораторная работа №3

Вариант № 666

Выполнил: студент группы Р3108,
Васильев Никита Алексеевич

Преподаватель: Афанасьев Дмитрий
Борисович

Санкт-Петербург 2024

Содержание

Текст задания	3
Функциональные зависимости	3
Нормальные формы	4
1NF	4
2NF	4
3NF	4
BCNF	4
Возможные денормализации	5
Функция и триггер на языке PL/pgSQL	6
Выводы по работе	7

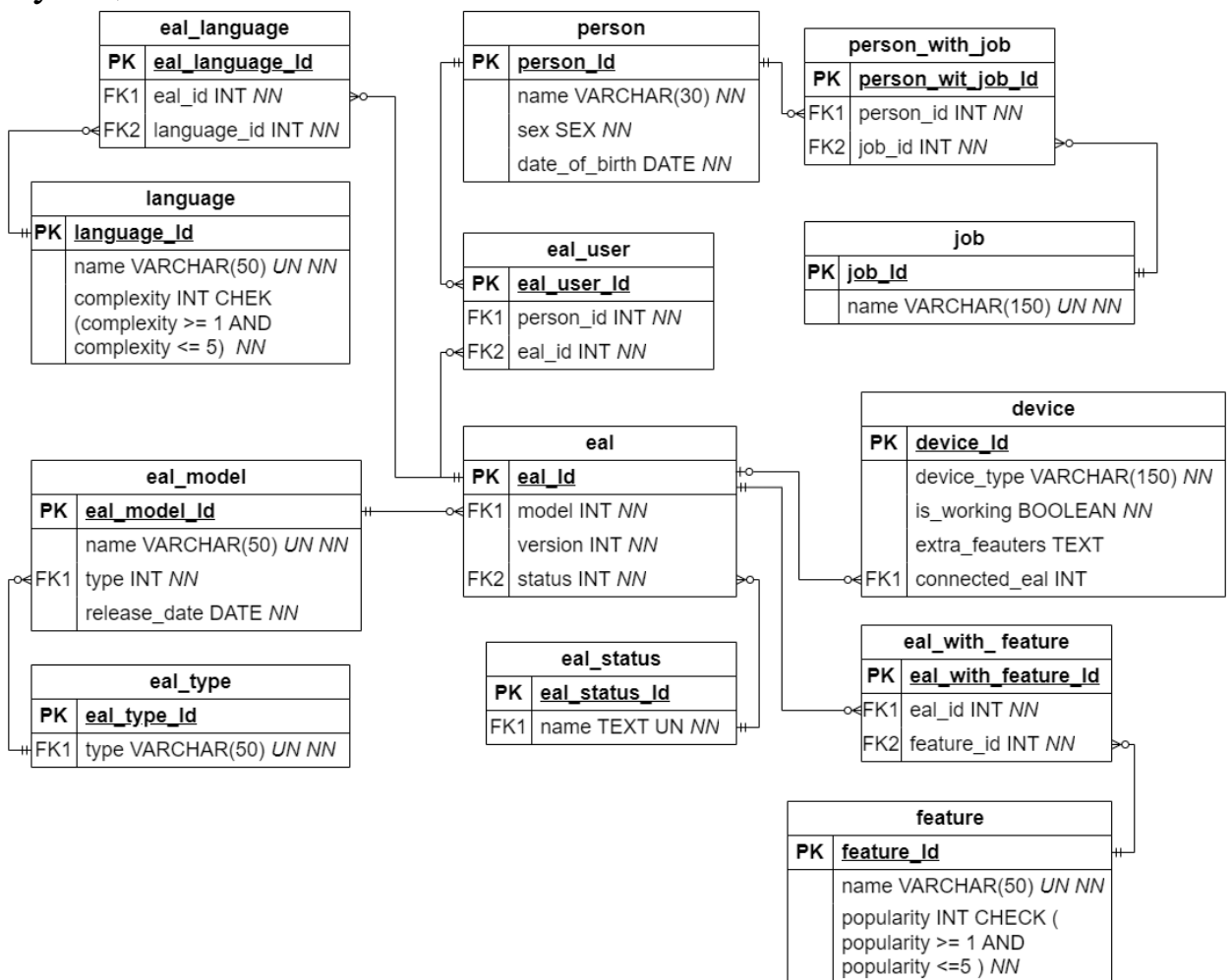
Текст задания

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 4NF (как минимум).
- Опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе 4NF;
- 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 триггер и связанную с ним функцию, относящиеся к вашей предметной области, согласуйте их с преподавателем и реализуйте на языке PL/pgSQL.

Функциональные зависимости



Сущность		
eal	id	model, version, status
device	id	device_type, is_working, extra_features, connected_eal
eal_status	id	name
person	id	name, sex, date_of_birth, job
job	id	name
eal_type	id	type
language	name	complexity
eal_model	name	type, release_date
feature	name	popularity
language	id	complexity
eal_model	id	type, release_date
feature	id	popularity
eal_language		eal_id, language_id
eal_user		eal_id, person_id
eal with feature		eal_id, feature_id

Нормальные формы

1NF

Отношение находится в 1NF, если на пересечении каждой строки и столбца — одно значение. В моей модели все атрибуты имеют одно значение.

2NF

Отношение находится в 2NF, если оно в 1NF, и атрибуты, не входящие в первичный ключ, в полной функциональной зависимости от первичного ключа отношения. В моей модели все не ключевые атрибуты полностью функционально зависят от первичных ключей.

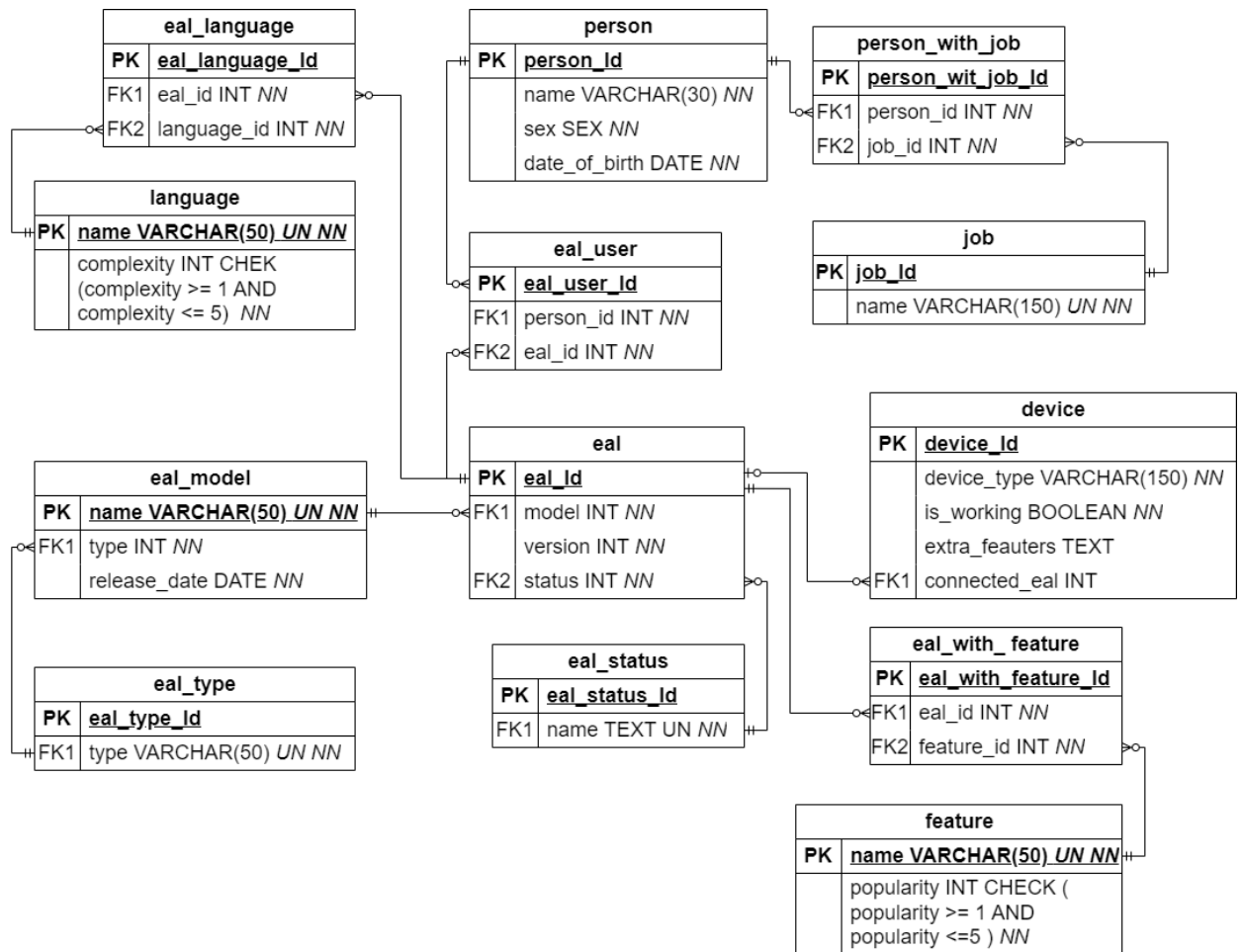
3NF

Отношение в 3NF, если оно в 1NF и 2NF, и все атрибуты, которые не входят в первичный ключ, не находятся в транзитивной функциональной зависимости от первичного ключа. Так как в моей модели присутствуют UNIQUE на некоторых атрибутах, она не соответствует 3NF. Необходимо изменить ПК для сущностей LANGUAGE, EAL_MODEL и FEATURE.

BCNF

Отношение в BCNF, если оно в 1NF, 2NF и 3NF, и для всех функциональных зависимостей отношения выполняется условие: детерминант — потенциальный ключ. В моей модели для всех функциональных зависимостей первичный ключ однозначно идентифицирует записи, находящиеся в отношении.

После преобразований даталогическая модель и функциональные зависимости выглядят следующим образом:

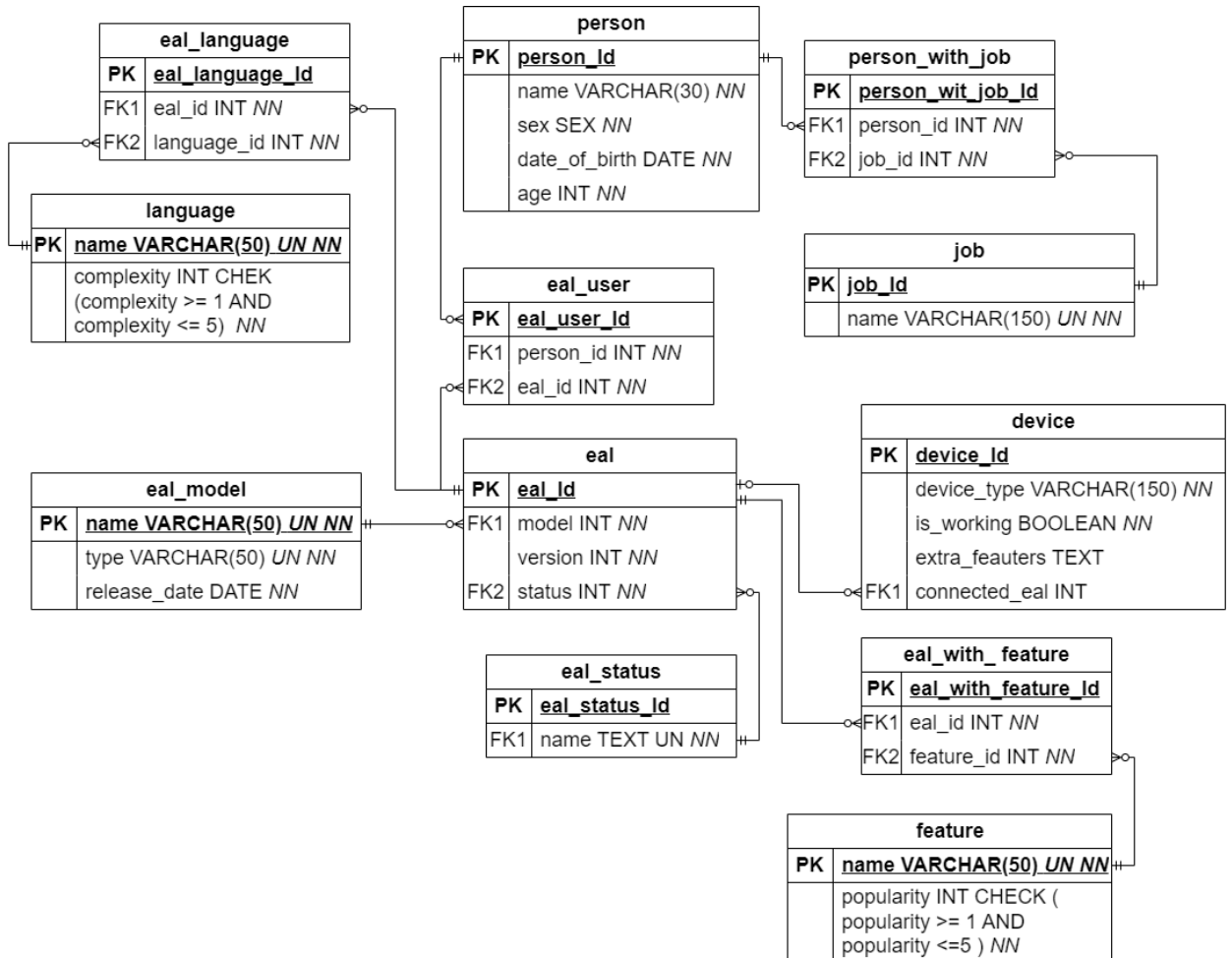


Сущность	Функциональные зависимости
eal	id model, version, status
device	id device_type, is_working, extra_features, connected_eal
eal_status	id name
person	id name, sex, date_of_birth, job
job	id name
eal_type	id type
language	name complexity
eal_model	name type, release_date
feature	name popularity
eal_language	eal_id, language_id
eal_user	eal_id, person_id
eal with feature	eal id, feature id

Возможные денормализации

- Объединение таблиц eal_type и eal_model, позволяющее ускорить запросы, если необходимо узнать о модели ЭАЛ и его типе одновременно.
- Добавление дополнительного атрибута age в таблицу person, позволяющее ускорить запрос на вычисление возраста.

Даталогическая модель при денормализации:



Функция и триггер на языке PL/pgSQL

```

CREATE OR REPLACE FUNCTION update_eal_status()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE eal SET status = 3 WHERE eal.id IN (
        SELECT eal.id FROM eal
        JOIN eal_user ON eal.id = eal_user.eal_id
        JOIN person ON eal_user.person_id = person.id
        JOIN person_with_job ON person.id = person_with_job.person_id
        JOIN job ON person_with_job.job_id = job.id
        WHERE job.id NOT IN (1, 3) AND eal.status = 1
        AND eal_user.person_id = NEW.person_id
    );
    RETURN NEW;
END
$$;

CREATE OR REPLACE TRIGGER check_job_id
AFTER INSERT OR UPDATE OF person_id ON eal_user
FOR EACH ROW
EXECUTE PROCEDURE update_eal_status();
    
```

Выводы по работе

В ходе выполнения данной лабораторной работы я познакомился с принципами нормализации и написанием функций и триггеров на языке PL/pgSQL.