Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Базы данных (БД)

**ОТЧЕТ**

по лабораторной работе №2

Выполнил

студент: гр. 851006 Верещагин Н.В.

Проверил: Фадеева Е.Е.

Минск 2021

1. **Какие каскадные операции необходимы в этой базе данных? Опишите их.**

В исходной:

* + Установить каскадное обновление атрибута и установку значения по умолчанию (NULL) sp\_parent отношения site\_pages при изменении атрибута sp\_id отношения site\_pages.
  + Установить каскадное обновление атрибутов pa\_from, pa\_to отношения payments\_archive при изменении атрибута a\_id отношения accounts.
  + Установить каскадное обновление атрибутов po\_from и po\_to отношения payments\_operational при изменении атрибута a\_id отношения accounts.
  + Установить запрет каскадной операции и предусмотреть обновление атрибута a\_owner отношения accounts при изменении атрибута o\_id отношения owners.
  + Установить каскадное удаление: при удалении записи в отношении owners, удаляются все соответствующие записи в других отношениях.
  + Добавить каскадное обновление атрибута ams\_a\_id отношения accounts\_m2m\_statuses при изменении атрибута a\_id отношения accounts.
  + Добавить каскадное обновление атрибута ams\_s\_id отношения accounts\_m2m\_statuses при изменении атрибута s\_id отношения statuses.

В измененной:

* Установить каскадное удаление: при удалении записи в отношении owners, удаляются все соответствующие записи в других отношениях.
* Установить каскадное обновление атрибутов pa\_from, pa\_to отношения payments\_archive при изменении атрибута a\_id отношения accounts.
* Установить каскадное обновление атрибутов po\_from и po\_to отношения payments\_operational при изменении атрибута a\_id отношения accounts.
* Установить каскадное обновление атрибута и установку значения по умолчанию (NULL) sp\_parent отношения site\_pages при изменении атрибута sp\_id отношения site\_pages.
* Установить каскадное обновление ow\_id, отношение passport и license, при изменении соответствующего атрибута в отношении owners.
* Установить каскадное обновление и установку значения по умолчанию (NULL) атрибута ow\_of\_id, отношения owners, при изменении атрибута of\_id, отношения offices.

1. **Существует ли возможность аномалий операций вставки, обновления, удаления данных? Составьте список таких случаев и внесите в базу данных исправления, позволяющие избежать таких аномалий.**

В исходной:

* + Аномалия обновления o\_name отношения owners (невозможно отличить владельцев с одинаковым именем).
  + Аномалия обновления и удаления of\_city отношения offices (невозможно отличить одноименные населенные пункты).
  + Аномалия удаления sp\_parent отношения site\_pages (необходимо сохранять url).

1. **Можно ли использовать схемы «звезда» или «снежинка» с этой базой данных, чтобы избежать некоторых аномалий операций с данными? Переработайте схему, сравните новую с исходной и составьте список аномалий работы с данными, которые были устранены (или, наоборот, появились).**

Можно, но они не уменьшат количество аномальных операций, а только увеличат размер самой базы данных. Поэтому, как для исходной модели, так и для измененной модели БД, использовать схемы «звезда» и «снежинка» является нерациональным решением.

1. **Составьте список всех функциональных зависимостей в базе данных.**

В исходной:

* + - Функциональная зависимость {ams\_a\_id, ams\_s\_id} → {ams\_last\_update} в отношении accounts\_m2m\_statuses.
    - Функциональная зависимость {sp\_name} → {sp\_parent} в отношении site\_pages.
    - Функциональная зависимость {PK id} → {атрибут отношения} для каждого отношения.

В измененной:

* + - Функциональная зависимость {pa\_identifying\_num} → {pa\_first\_name, pa\_middle \_name, pa\_last \_name} для отношения passport.
    - Функциональная зависимость {li\_reg\_num} → {li\_name, li\_reg\_date} для отношения license.
    - Функциональная зависимость {PK id} → {атрибут отношения} для каждого отношения.

1. **Существуют ли отношения, имеющие многозначные зависимости? Если «да», как можно переработать схему, чтобы избежать таких зависимостей?**

Многозначные зависимости в обоих моделях БД отсутствуют.

1. **Нарушает ли схема какие-либо «требования нормализации»? Если «да», доработайте схему, чтобы избежать таких нарушений.**

В исходной:

* + - В отношении site\_pages существует нарушение требований 1НФ, поскольку атрибут sp\_name нужно разбить на два атрибута: имя (name), адрес (url). А также нарушение требований 3НФ, поскольку существует транзитивная зависимость {sp\_id}→{sp\_name}→{sp\_parent}.
    - В отношении owners существует нарушение требований 1НФ, поскольку атрибут o\_name следует разбить на три атрибута: имя (first\_name), отчество (middle\_name), фамилия (last\_name).

В измененной:

* + - Для отношения passport существует нарушение требований 1НФ, поскольку атрибут pa\_birthday\_date следует разбить на три атрибута: pa\_birthday\_year, pa\_birthday\_month, pa\_birthday\_date.

1. **Существуют ли какие-либо потенциальные проблемы с производительностью базы данных? Если «да», запишите их.**

В исходной:

* + - Таблицу payments\_operational нужно оптимизировать на запись, убрав все индексы.
    - Проблема поиска в отношениях payments\_operational и payments\_archive при значительном увеличении количества данных.
    - К данным в таблице site\_pages обращение будет происходить часто, поэтому нужно произвести оптимизацию для чтения, добавив индекс (parent\_page, name).

В измененной:

* + - Такие же изменения, как и для исходной таблицы.

1. **Для каждого отношения в базе данных определите, в какой нормальной форме оно находится. Запишите ответ.**

В исходной:

* payments\_archive: 5 НФ
* payments\_operational: 5 НФ
* statuses: 6 НФ
* offices: 5 НФ
* site\_pages: 6 НФ
* accounts: 5 НФ
* owners: 0 НФ

В измененной:

* payments\_archive: 5 НФ
* payments\_operational: 5 НФ
* statuses: 6 НФ
* offices: 5 НФ
* site\_pages: 6 НФ
* accounts: 5 НФ
* owners: 5 НФ
* passport: 5 НФ
* license: 5 НФ

1. **Есть ли отношения с возможной, но ненужной дальнейшей нормализацией? Составьте список.**

В исходной:

* + - Для отношения site\_pages атрибут sp\_name возможно разбить на два атрибута: имя (name), адрес (url).
    - В отношении owners атрибут o\_name возможно разбить на три атрибута: имя (first\_name), отчество (middle\_name), фамилия (last\_name).

В измененной: отсутствует.

1. **Можно ли добиться некоторого повышения производительности за счёт денормализации схемы? Обоснуйте своё мнение.**

В исходной:

* + - Нет, поскольку при совмещении отношений, работа с данными будет только увеличиваться.

В измененной:

* + - Можно хранить атрибуты отношения license в отношении owners, но это не безопасно, а также не все владельцы счета имеют лицензию.
    - Также можно хранить атрибуты отношения passport в отношении owners, но это не безопасно.

1. **Можно ли добиться некоторого повышения производительности, добавив в схему кэширующие отношения? Обоснуйте своё мнение.**

В исходной:

* + - Кэшировать данные payment\_arhive и payment\_operational за последние 500 записей.

В измененной:

* + - Кэшировать данные passport и license за промежуток времени.
    - Повторить такие же изменения, как и в исходной таблице.

1. **Добавьте в базу данных все необходимые индексы, представления, хранимые процедуры и т. д.**

В исходной:

Индексы:

* В отношении offices, по атрибуту of\_name.
* В отношении payments\_archive, по атрибуту po\_dt.
* В отношении payments\_operational, по атрибуту pa\_dt.

Хранимые процедуры:

* Процедура переноса из payment\_operational в payment\_archive.

Триггеры:

* На изменение валюты и баланса.

В измененной:

Индексы:

* В отношении passport, по атрибуту pa\_last\_name.
* В отношении license, по атрибуту li\_reg\_date.
* В отношении offices, по атрибуту of\_name.
* В отношении payments\_archive, по атрибуту po\_date.
* В отношении payments\_operational, по атрибуту pa\_date.

Хранимые процедуры:

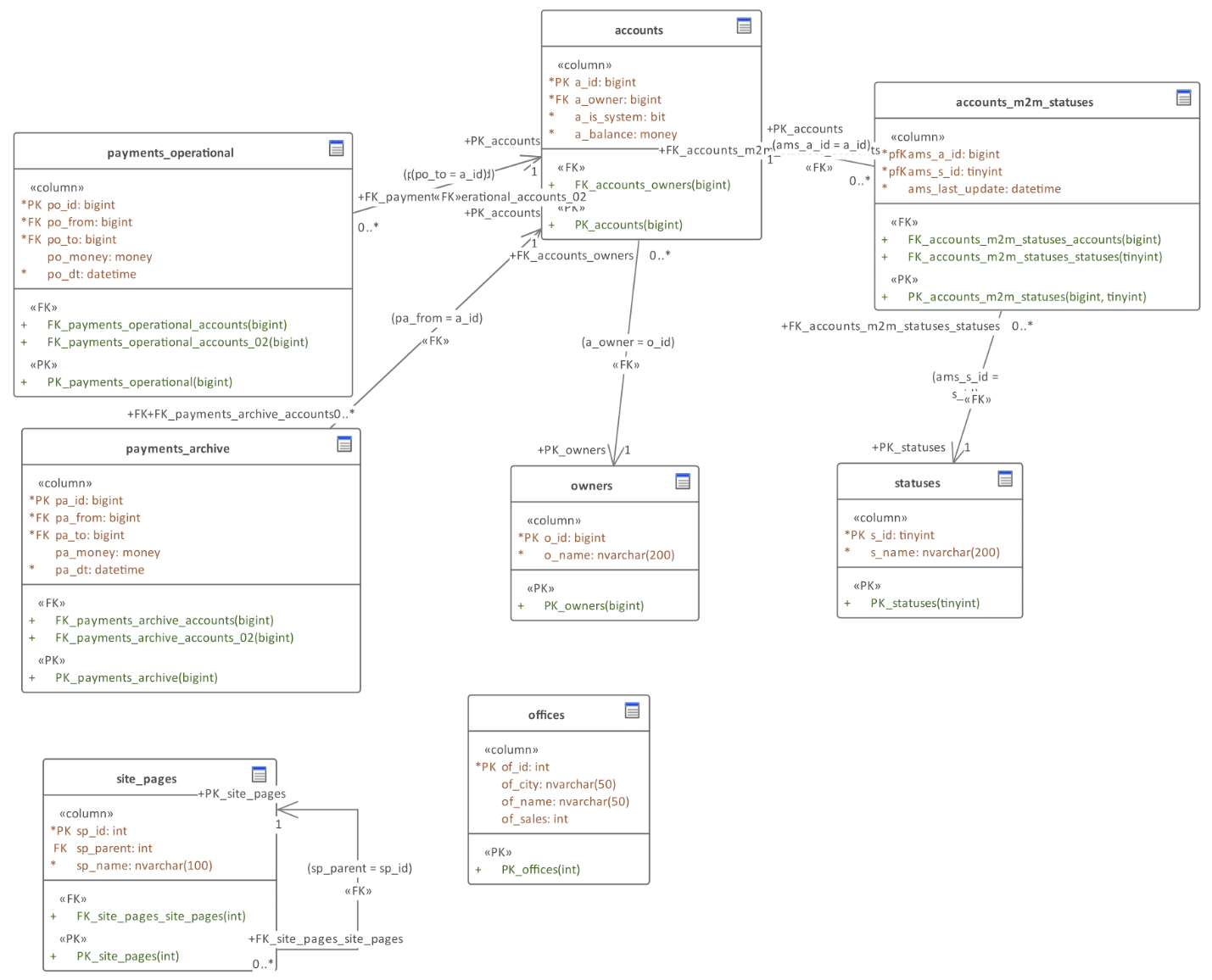
* Процедура переноса из payment\_operational в payment\_archive.

Триггеры:

* На изменение валюты и баланса.

Представление:

* Транзакциях пользователя.
* Полной информации об аккаунтах владельца.

**Исходная модель БД:**

**Измененная модель БД:**

