

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.

2. 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).

3. 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 уменьшают 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.

4. 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 $\{sp_name\} \rightarrow \{sp_parent\}$ в отношении `site_pages`.
- Функциональная зависимость $\{PK\ id\} \rightarrow \{\text{атрибут отношения}\}$ для каждого отношения.

В измененной:

- Функциональная зависимость $\{pa_identifying_num\} \rightarrow \{pa_first_name, pa_middle_name, pa_last_name\}$ для отношения `passport`.
- Функциональная зависимость $\{li_reg_num\} \rightarrow \{li_name, li_reg_date\}$ для отношения `license`.
- Функциональная зависимость $\{PK\ id\} \rightarrow \{\text{атрибут отношения}\}$ для каждого отношения.

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

Многозначные зависимости в обоих моделях БД отсутствуют.

6. 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 $\{sp_id\} \rightarrow \{sp_name\} \rightarrow \{sp_parent\}$.
- 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`.

7. 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.

8. 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

9. 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.

10. 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.

11. 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.

12. 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.

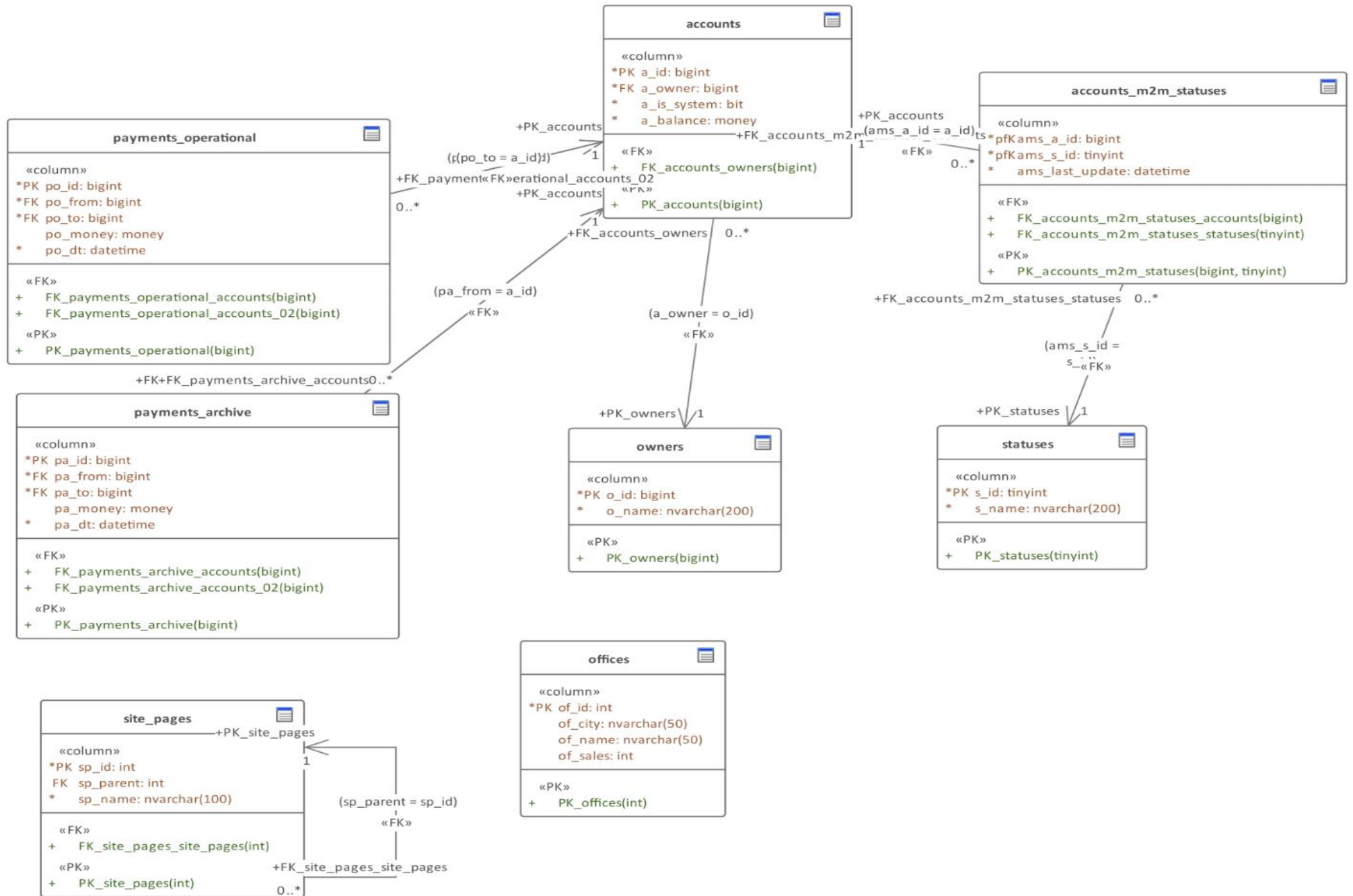
Триггеры:

- На изменение валюты и баланса.

Представление:

- Транзакциях пользователя.
- Полной информации об аккаунтах владельца.

Исходная модель БД:



Измененная модель БД:

