Проанализируйте модели базы данных «Банк» (исходный вариант (см. ниже) и вариант, изменённый в предыдущей работе). Для обеих вариантов выполните следующие задачи:

1. Какие каскадные операции необходимы в этой базе данных? Опишите их.
2. Существует ли возможность аномалий операций вставки, обновления, удаления данных? Составьте список таких случаев и внесите в базу данных исправления, позволяющие избежать таких аномалий.
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).
4. Составьте список всех функциональных зависимостей в базе данных.
5. Существуют ли отношения, имеющие многозначные зависимости? Если «да», как можно переработать схему, чтобы избежать таких зависимостей?
6. Нарушает ли схема какие-либо «требования нормализации»? Если «да», доработайте схему, чтобы избежать таких нарушений.
7. Существуют ли какие-либо потенциальные проблемы с производительностью базы данных? Если «да», запишите их.
8. Для каждого отношения в базе данных определите, в какой нормальной форме оно находится. Запишите ответ.
9. Есть ли отношения с возможной, но ненужной дальнейшей нормализацией? Составьте список.
10. Можно ли добиться некоторого повышения производительности за счёт денормализации схемы? Обоснуйте своё мнение.
11. Можно ли добиться некоторого повышения производительности, добавив в схему кэширующие отношения? Обоснуйте своё мнение.
12. Добавьте в базу данных все необходимые индексы, представления, хранимые процедуры и т. д.
13. Добавьте недостающие комментарии в базу данных.
14. Обновите логическую модель, экспортируйте окончательную версию модели в виде SQL-кода и импортируйте в реальную СУБД.

База данных представляет следующие сущности и атрибуты (см. Рисунок A) + изменения, внесённые вами в лабораторной 1:

* Account – *Учётная запись (описывает учётную запись)*:
  + id *(идентификатор аккаунта)*;
  + balance (*баланс счёта*, MONEY);
  + account owner (*владелец счёта*, FK);
  + system account (*флаг, указывающий, что эта учётная запись не принадлежит человеку*).
* Status – *Статус (статус аккаунта, например, «Активен», «Заблокирован» и т.д.)*:
  + id *(идентификатор статуса)*;
  + name (*название статуса*).
* Transaction operational – *Текущие транзакции (для транзакций в текущем месяце)*:
  + id (*идентификатор транзакции*);
  + source account (*исходный счёт*, FK);
  + destination account (*целевой счёт,* FK);
  + date and time (*дата и время транзакции*);
  + sum (*общая сумма транзакции*).
* Transaction archive – *Архив транзакций (для транзакций до текущего месяца)*:
  + id (*идентификатор транзакции*);
  + source account (*исходный счёт*, FK);
  + destination account (*целевой счёт,* FK);
  + date and time (*дата и время транзакции*);
  + sum (*общая сумма транзакции*).
* Account owner – *Владелец счёта (клиент банка)*:
  + id (*идентификатор владельца аккаунта*);
  + name (*имя владельца аккаунта*).
* Site page – *Страница сайта (страница сайта банка)*:
  + id (*идентификатор страницы*);
  + parent page (*родительская страница,* FK);
  + name (*название страницы*).
* Office – *Офис* (*офис банка*):
  + id (*идентификатор офиса*);
  + city (*местонахождение офиса*);
  + name (*название офиса*);
  + total sells sum (*сумма прибыли офиса*, MONEY).

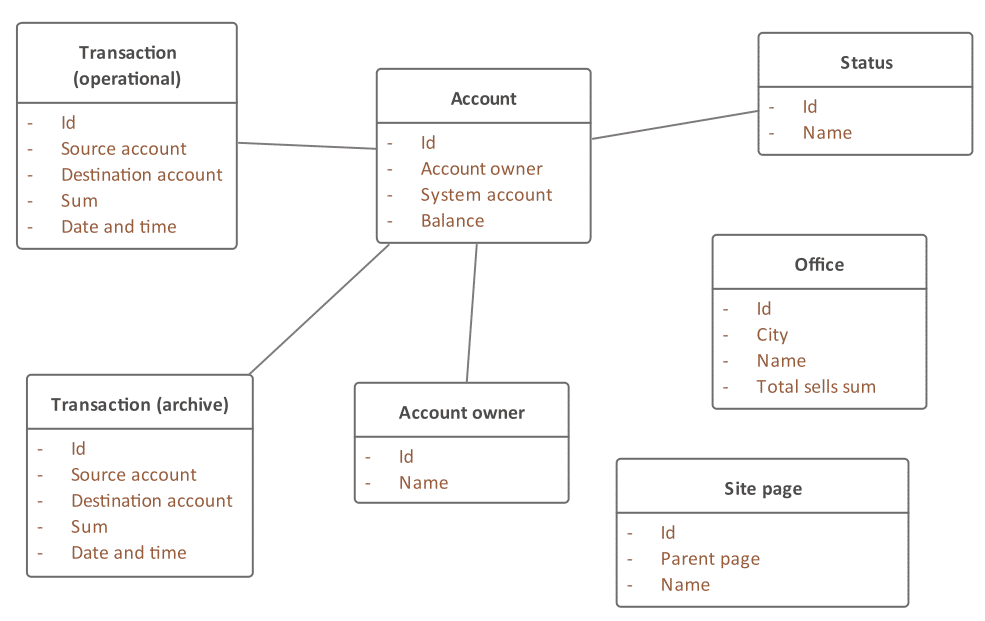


Рисунок A – Концептуальная модель

Даталогический уровень исходной базы данных выглядит следующим образом (см. Рисунок B и Рисунок C):

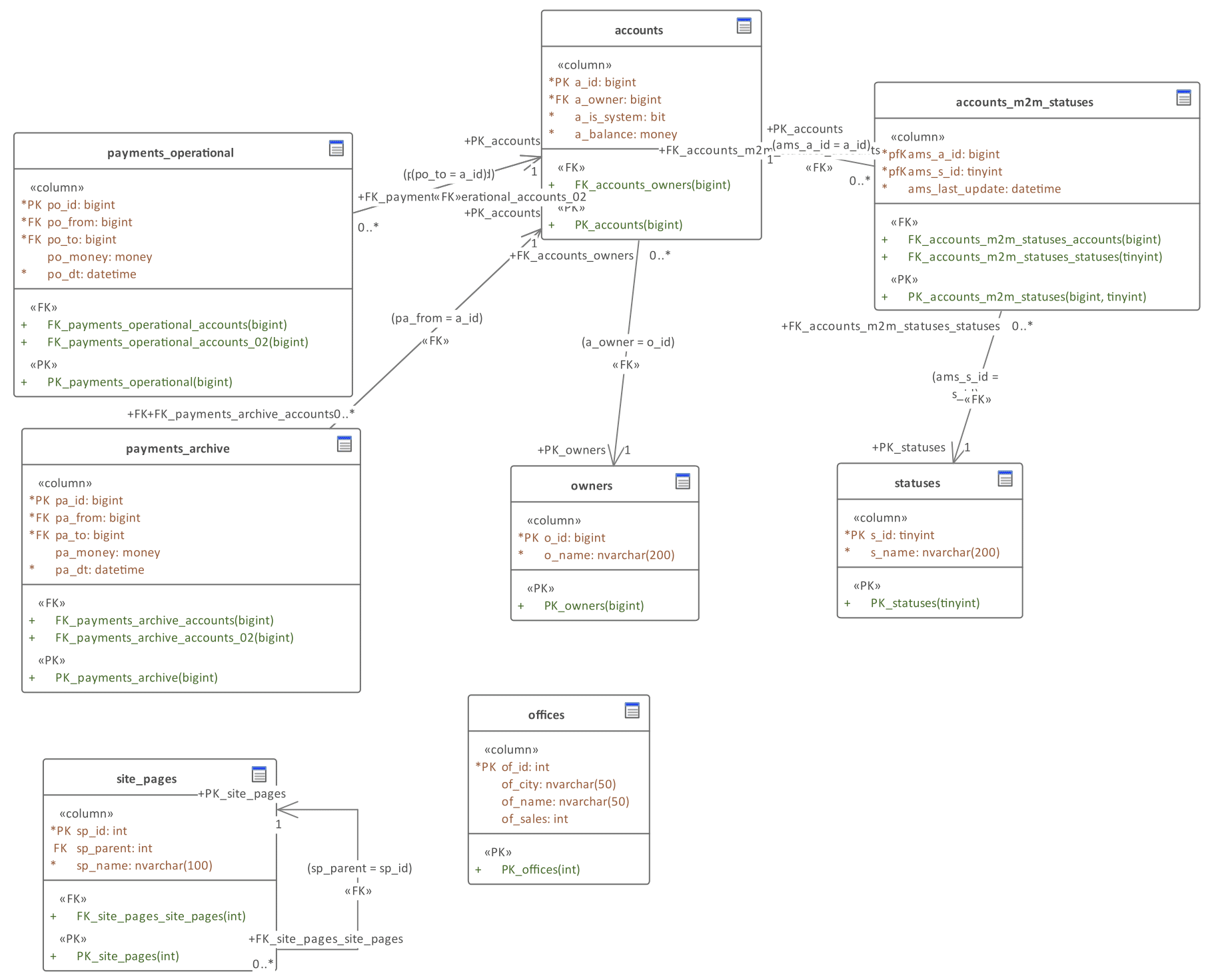


Рисунок B – Даталогическая модель для MS SQL Server

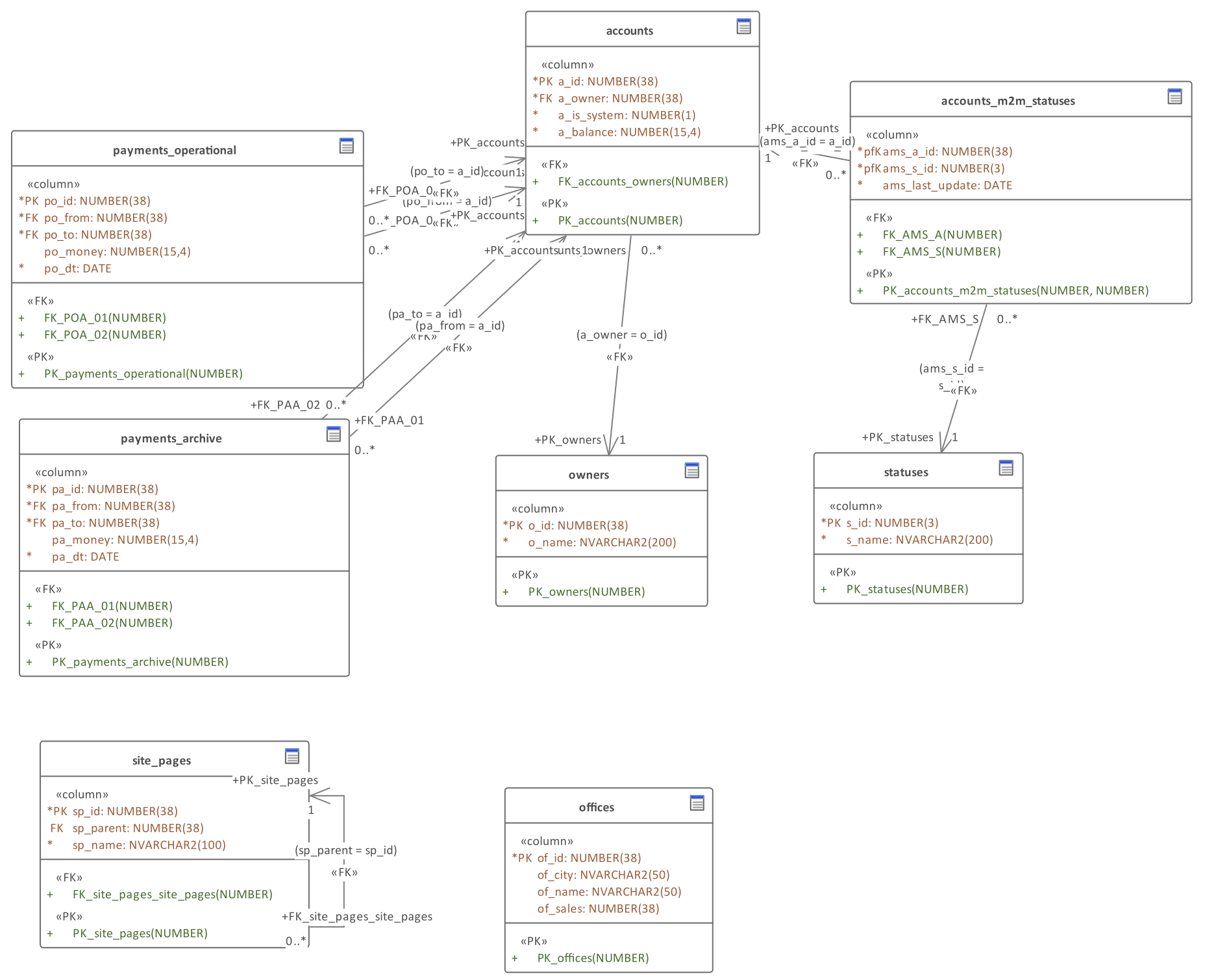


Рисунок C – Даталогическая модель для Oracle