Проанализируйте модель базы данных «Банк» (см. ниже). Выполните следующие задачи:

1. Видите ли вы какие-либо недостатки на инфологической и даталогической моделях? Если да, запишите их.
   1. Не предусмотрена поддержка разных валют;
   2. Не предусмотрено владение счетом юридическим лицом;
   3. К счету не привязан инструмент для совершения транзакций. Пример, платежная карточка;
   4. Недостаточно данных о владельце аккаунта.
2. Какие вопросы вы хотите задать заказчику для улучшения модели? Запишите эти вопросы и дайте на них ответ от лица заказчика.
   1. Имеет ли банк физические отделения;
   2. Для чего имеется сущность «site\_pages»;
   3. Для чего имеется сущность «offices»;
   4. Банк является международным;
   5. Каким образом банк производит транзакции.
   6. Банк осуществляет работу на одном языке.
3. Какие отношения отсутствуют в базе данных? Добавьте эти отношения.
   1. Отношение «терминал».
   2. Отношение «тип терминала».
   3. Отношение «физический владелец»;
   4. Отношение «паспорт»
   5. Отношение «юредический владелец»;
   6. Отношение «тип аккаунта»;
   7. Отношение «банковская карта»;
   8. Отношение «тип банковской карты»;
   9. Отношение «тип валюты».
   10. Отношение «курс обмена».
4. Какие атрибуты отсутствуют в существующих отношениях? Добавьте недостающие атрибуты.
   1. Сущность «payments\_operational» атрибут «комиссия»;
   2. Сущность «payments\_operational» атрибут «дата»;
   3. Сущность «payments\_archive» атрибут «комиссия»;
   4. Сущность «payments\_archive» атрибут «дата»;
5. Все ли объекты в базе данных соответствуют единым соглашениям об именовании объектов?
   1. Не соблюдены соглашения названия атрибутов на даталогическая модели:
      1. Сущность «offices» у атрибутов имеет префикс «of»;
      2. Сущность «accounts» у атрибутов имеет префикс «a»;
      3. Сущность «payments\_operation» у атрибутов имеет префикс «po».
6. Все ли атрибуты в базе данных имеют оптимальные типы данных? Предложите оптимизацию.
   1. Не все атрибуты в базе данных имею оптимальные типы данных:
      1. Атрибут «of\_city» сущности «offices» имеет тип «nvchar(50)». Размера типа может не хватит для хранения названия города.
         1. Город «Llanfairpwll — Moelfre — Harlech Castle — Machynlleth». Длина – 54 символа.
      2. Ключи, представляющие собой набор цифр оптимальное хранить в типе «vchar» для оптимизации затрат памяти.
         1. Сущность «accounts» атрибут «a\_id».
         2. Сущность «payments\_operational» атрибуты «po\_from» и «po\_to».
         3. Сущность «payments\_archive» атрибуты «po\_from» и «po\_to».
7. Все ли отношения в базе данных имеют оптимальные ключи? Если нет, предложите оптимизацию.
   1. В сущности «accounts» первичным ключ является искусственным. Можно использовать естественный ключ.
8. Есть ли в базе данных избыточные или отсутствующие связи? Внесите необходимые исправления.
   1. Связь между сущностями «payments\_operational» и «accounts» избыточна. Она не позволяет хранить транзакции между разными банками.
   2. Связь между сущностями « payments\_archive » и «accounts» избыточна. Она не позволяет хранить архив транзакций между разными банками

Все ответы оформите в отчёт по лабораторной работе. В отчёт добавить скриншоты исходных и изменённых моделей.

Это – предельно упрощённая модель базы данных некоего воображаемого банка. Она содержит только основные элементы, а также содержит некоторое количество намеренно допущенных ошибок, поэтому ваша задача – найти и исправить их.

База данных содержит следующие сущности и атрибуты (см. Рисунок A):

* 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).

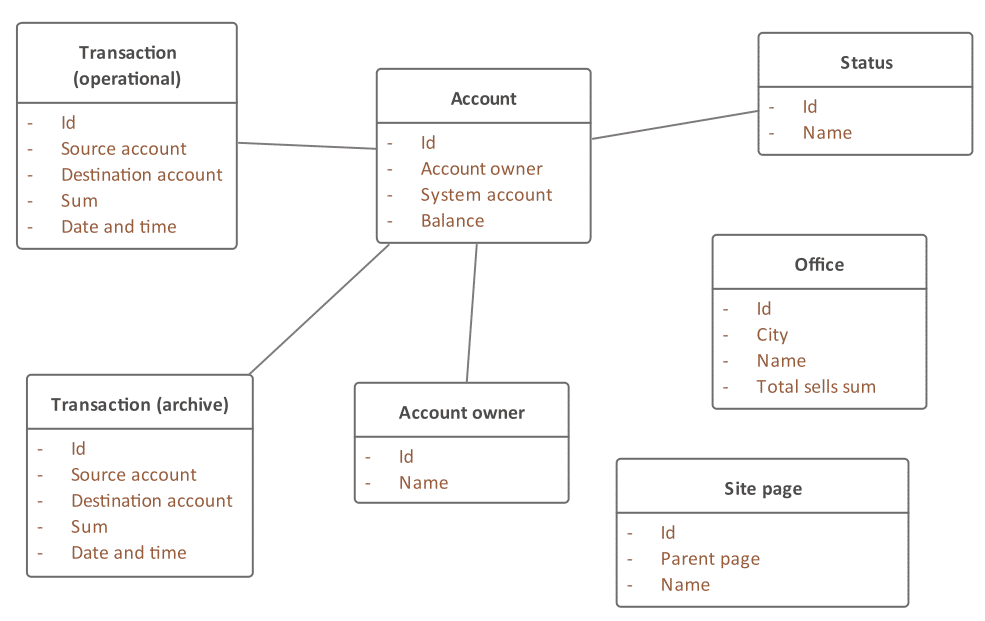


Рисунок A – Концептуальная модель

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

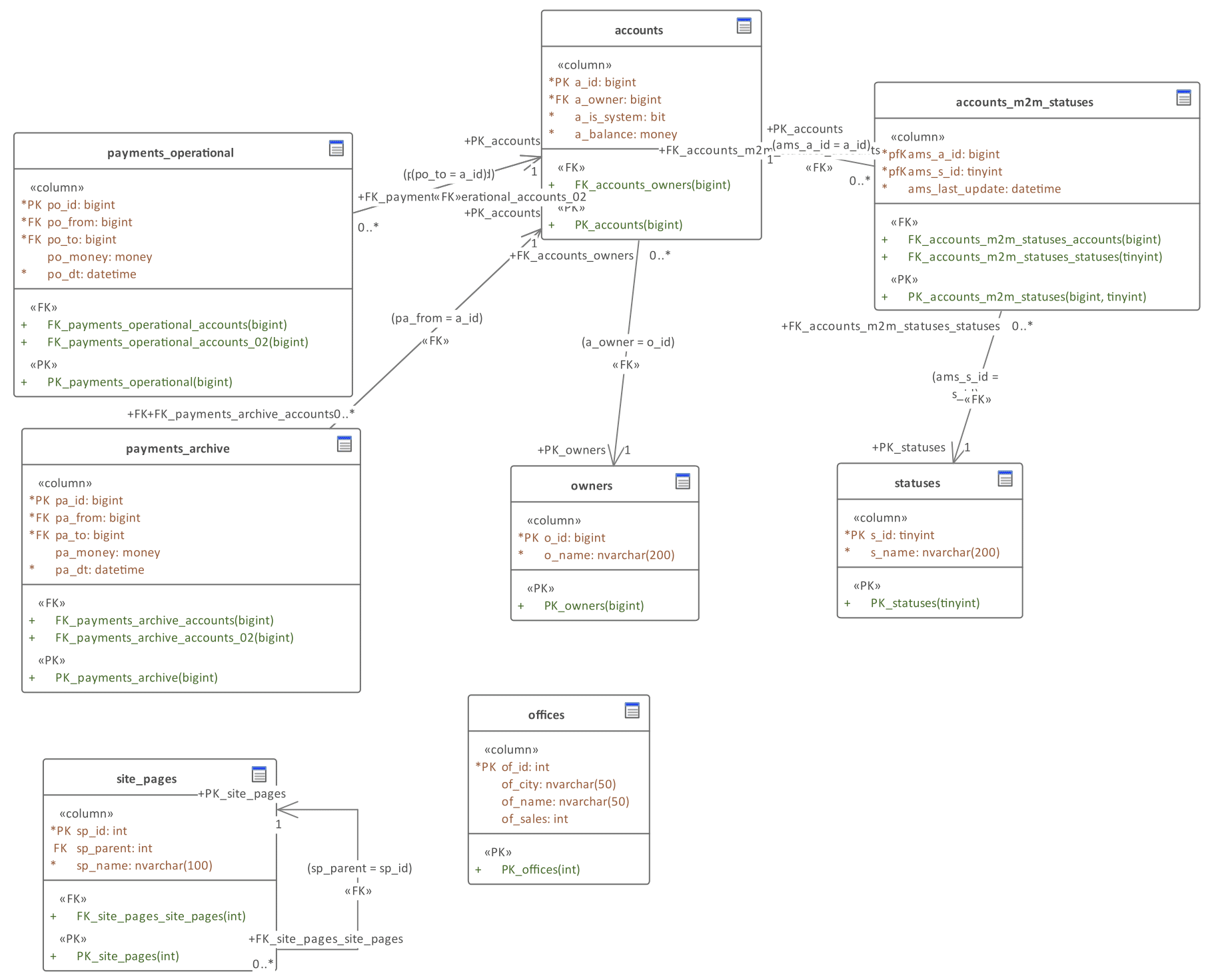


Рисунок B – Даталогическая модель для MS SQL Server

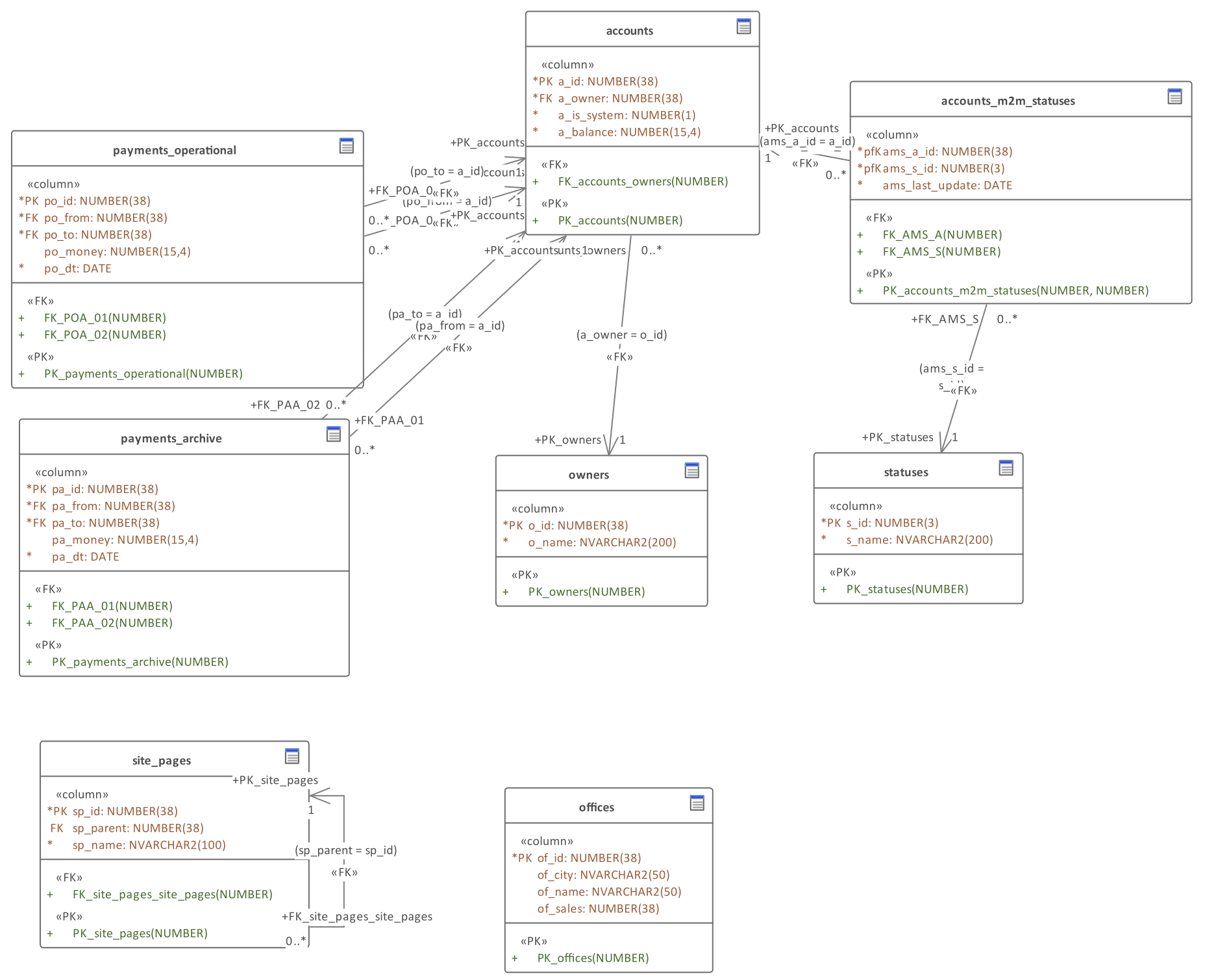


Рисунок C – Даталогическая модель для Oracle

