

Санкт-Петербургский государственный университет
Математико-Механический факультет
Кафедра Прикладной кибернетики

Еричев Алексей Олегович

ПРОЕКТИРОВАНИЕ СТРУКТУР ХРАНЕНИЯ ДАННЫХ С ОБЕСПЕЧЕНИЕМ
ВЫСОКОЙ ПРОИЗВОДИТЕЛЬНОСТИ.

Курсовая работа

Научный руководитель:

Системный аналитик «TaskData»

Д. А. Богданов

Рецензент:

Заведующий кафедрой

прикладной кибернетики,

д. ф.-м. н., профессор Н. В. Кузнецов

Санкт-Петербург

2020

Оглавление

| | | |
|------------------------------------|-------------------------------------------------|-----------|
| 1. | Введение | 3 |
| 2. | Постановка задачи. Техническое задание. | 4 |
| 3. | Системная постановка | 5 |
| 3.1. | Описание решения | 5 |
| 3.2. | Характеристики полей таблиц | 8 |
| 3.3. | Матрицы валидаций таблиц | 11 |
| 4. | Заключение | 22 |
| Список литературы | | 23 |
| 5. | Приложение | 24 |

1. Введение

В наше время объёмы обрабатываемой цифровой информации заметно возросли. Однако, не смотря на огромное количество проводимых операций, структура данных при работе в отдельно взятой системе часто остается постоянной. Поэтому возникла острая необходимость научиться эффективно хранить данные одинаковой структуры.

Данная курсовая работа посвящена описанию системы, обеспечивающей оптимальное хранение данных бизнес-процессов схемы биллинга работы продуктового склада. В ходе изучения данной темы была спроектирована база данных, обеспечивающая хранение информации о договорах, выставленных счетах и проведенных операциях оплаты, а так же поставках продуктов на склад. В процессе описания системы были приведены подробные пояснения и матрицы валидаций для всех таблиц данной базы.

После описания всех необходимых структур была создана база данных PostgreSQL, реализующая требуемый функционал, которая находится в приложении.

2. Постановка задачи. Техническое задание.

Необходимо спроектировать и описать базу данных, для обеспечения следующих функциональных возможностей:

1. Хранение договора клиента и ведение его баланса
2. Хранение услуг и ведение их стоимости
3. Проведение начислений за предоставленные услуги
4. Проведение платежей
5. Перевод денежных средств с договора на договор
6. Ввод остатков по клиентам при переносе их из другой системы
7. Хранение выставленных счетов клиенту
8. Погашение выставленных счетов
9. Хранение информации о поставках продуктов на склад
10. Хранение информации о количестве продуктов и сроке их хранения

3. Системная постановка

3.1. Описание решения

В ходе изучения материалов бизнес-процессов биллинга[5] была составлена схема базы данных (см. рис.1), содержащая таблицы нескольких видов: опорные или базисные таблицы, таблицы первичных документов, таблицы поставок продуктов на склад, а так же справочные и вспомогательные таблицы.

Базисными таблицами, обеспечивающими корректную работу схемы, биллинга являются:

- Договоры (Contract) — минимально необходимая информация для описания договора.
- Проводки (Trx) — журнал проведенных операций по списанию и пополнению денежных средств.
- Используемая сторона учета (Ledgertype) — указывает на то куда идет проводка (дебит, кредит).
- Счета выставленные клиенту (Invoice) — счета, которые выставляются клиенту за приобретение товара.

Для удобства работы были добавлены таблицы первичных документов. Под первичными документами подразумеваются документы, на основании которых делаются проводки на по договору клиента:

- Остатки (Remain) — входящие остатки при миграции из другой системы.
- Платежи (Payment) — поступающие от клиентов денежные средства.
- Переводы (Transfer) — перевод денежных средств со счета на счет.
- Начисления (Charge) — начисления за приобретенные товары.
- Начисления в счете (Invoice_trx) — начисления за приобретенные товары, включенные в счет.
- Платежи по счету (Invoice_cover_trx) — оплаченные счета.

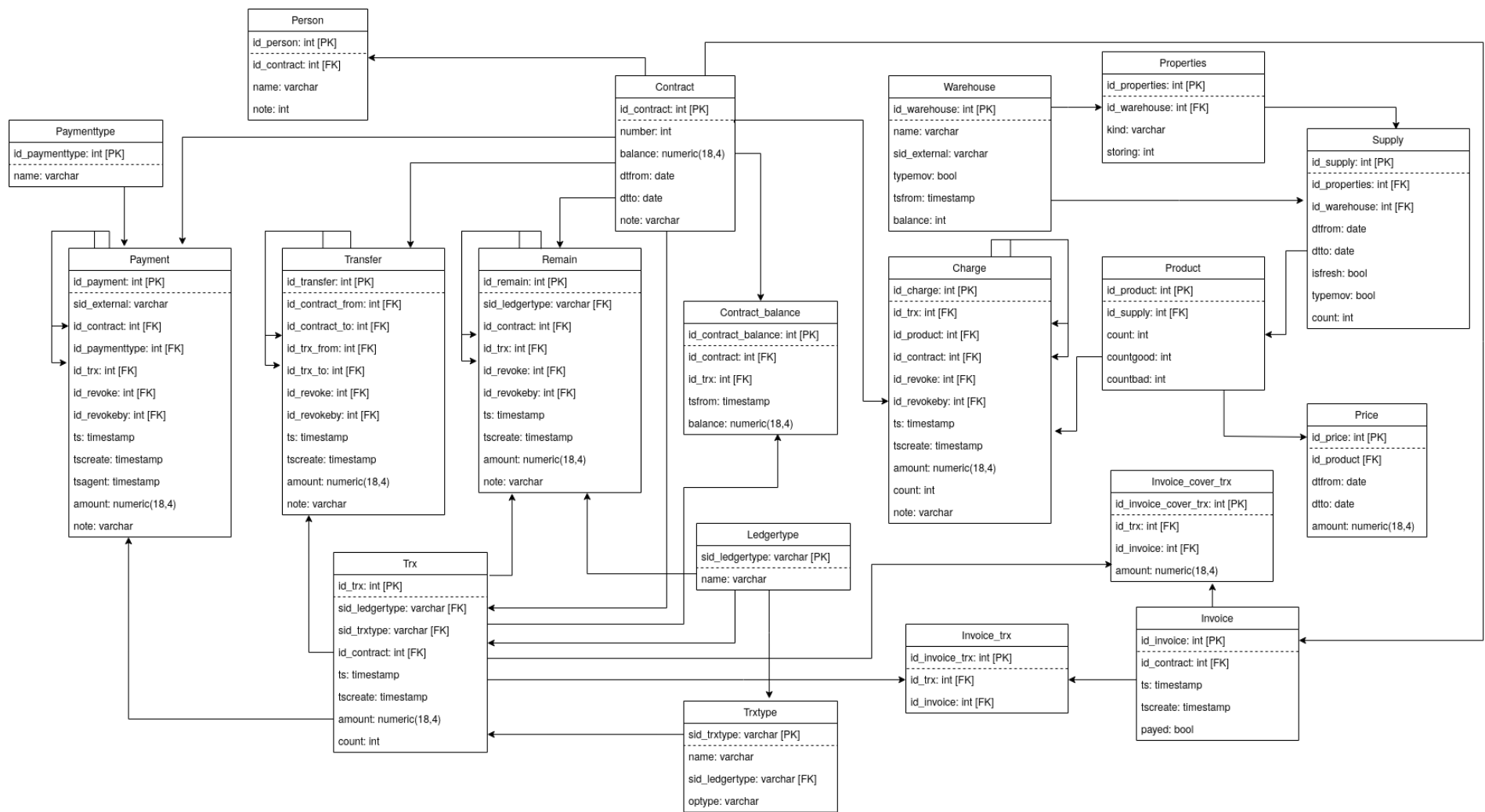


Рис. 1. Схема базы данных.

Отдельно была организована работа учета количества продуктов на складе через таблицы:

- Изменение состояния склада (Warehouse) — содержит информацию о поступающих и выбывающих товарах со склада.
- Свойства товара (Properties) — характеристики поступающих и выбывающих товаров.
- Поставки (Supply) — таблица сформированных поставок, содержащая информацию о свежести товара.
- Товары (Product) — имеющиеся на складе товары с информацией о количестве свежей и просроченной продукции.

В схеме присутствуют следующие справочные таблицы:

- Типы платежей (Paymenttype) — деление платежей по типам. (Наличные, безналичный расчет, платежные агенты и т.п.)
- Цены (Price) — стоимость товаров в данный период времени, так же хранит более ранние изменения стоимости.
- Тип проводки (Trxtype) — указывает на используемый тип проводки для первичных документов, а так же сторону учета используемую по умолчанию. В случае если сторона по умолчанию не указана, выбор происходит при проведении документа.

Так же добавлены две вспомогательные таблицы:

- История баланса (Contract_balance) — история изменения баланса договора с привязкой к проводкам.
- Клиенты (Person) — информация о клиенте, с которым заключен договор.

3.2. Характеристики полей таблиц

При описании данной системы использовался следующий набор принципов, распространяющийся на все таблицы:

- Все внешние ключи имеют формат <первичный ключ>_<имя таблицы>. В случае если внешних ключей несколько или они указывают на саму таблицу допускается либо дополнение к названию вида `id_<имя таблицы>_<поясняющее дополнение>` или же `id_<поясняющее дополнение>`.
- Поля с денежными средствами определены как `numeric(18,4)`.
- Поля с датой имеют префикс `dt` в обязательном порядке.
- Поля с временем содержат дату и имеют префикс `ts` в обязательном порядке.
- В случае если имеется временной интервал (`dtfrom`, `dtto` или `tsfrom`, `tsto`), то первая дата всегда задана и по умолчанию равна `now()`, вторая дата может быть пустой и в этом случае интервал считается действующим на данный момент.
- В части случаев у справочников вместо числового первичного ключа используется текстовый мнемонический ключ. Такие ключи обозначены как `sid`.

У базисных таблиц, первичных документов, а так же таблиц поставок продуктов есть общие поля:

- `id_contract` (`id_contract_from`, `id_contract_to`) — указывает на договор или договора документа.
- `id_trx` (`id_trx_from`, `id_trx_to`) — указывает на проводку или проводки документа.
- `ts` — время документа.
- `tscreate` — время создания документа.
- `amount` — сумма документа.
- `sid_external` — используется для связи с внешними системами.
- `note` — текстовое примечание.
- `id_revoke` — корректируемый документ. Указывает на тот документ который корректируется этим.

- *id_revokedby* — корректирующий документ. Указывает на тот документ который откорректировал текущий.

Корректируемые и корректирующие документы используются для корректного исправления документа в котором есть ошибка. Корректирующий документ содержит исправленную информацию и аннулирует корректируемый документ. Для этого используются два поля *id_revoke* — заполняется у корректирующего документа, а *id_revokedby* у корректируемого.

Кроме общих полей, у базисных таблиц есть свои специфичные поля:

- Договор (Contract): *balance* — показывает сумму на которую заключен договор.
- Журнал проведенных операций (Trx):
 - *sid_trxtype* — указывает тип проведенной операции при проведении документа.
 - *count* — указывает на количество проведенных операций.
- Счета (Invoice): *payed* — указывает на то, оплачен ли счет или нет.

Так же свой специфичные поля есть у первичных документов:

- Документы начислений (Charge): *count* — указывает на количество проданных видов товаров.
- Документы остатков (Remain): *sid_ledgertype* — указывает используемую сторону учета при проведении документа.
- Документы платежей (Payment): *tsagent* — время поступления оплаты по договору.

Специфические поля таблиц поставок:

- Документы изменения состояния склада (Warehouse):
 - *name* — название поставленного/проданного товара.
 - *typemov* — указывает, данный товар поступил на склад или выбыл.
 - *balance* — текущее количество данного товара на складе.
- Документы характеристик товара (Properties):
 - *kind* — вид поставленного или проданного товара.
 - *storing* — время хранения данного товара на складе.

- Поставки (Supply):
 - *isfresh* — указывает, товар данной поставки свежий или просроченный.
 - *typemov* — указывает, данный товар поступил на склад или выбыл.
 - *count* — количество поступивших или выбывших единиц товара данной поставки на складе.
- Продукты (Product):
 - *count* — общее количество данного товара на складе.
 - *countgood* — количество свежих единиц данного товара на складе.
 - *countbad* — количество просроченных единиц данного товара на складе.

Специфические поля справочных таблиц:

- Типы проводок (Trxtype):
 - *name* — название типа проводки.
 - *optype* — указывает, на тип проведенной операции.

3.3. Матрицы валидаций таблиц

tab. Contract

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|---------------------------------------------------------------------------------|---------------|------------|----------------------------------------|
| id_contract | Уникальный идентификатор записи договора в таблице | int | 42 | не пусто |
| number | номер договора | int | 32 | не пусто |
| balance | сумма, на которую заключен договор | numeric(18,4) | 44,32 | ≥ 0 |
| dtfrom | дата начала действия договора | date | 01.01.2001 | не пусто |
| dtto | дата окончания действия договора; если не заполнена, договор с продлением | date | 02.01.2001 | если заполнена, то не меньше dtfrom |
| note | примечания, комментарии | varchar | примечание | |

tab. Trx

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|------------------------------------------------------------------------------------------------------------------|---------------|------------------------|---------------------------------|
| id_trx | Уникальный идентификатор проведенной операции в таблице | int | 5454 | не пусто |
| sid_ledgertype | Текстовый мнемонический идентификатор, указывающий используемую сторону учета при проведении документа в таблице | varchar | cli | не пусто |
| sid_trxtype | Текстовый мнемонический идентификатор, указывающий тип проводки в таблице | varchar | dbt | не пусто |
| id_contract | Уникальный идентификатор записи договора в таблице | int | 434 | не пусто |
| ts | Дата проведения операции с документом | timestamp | 2004-10-19 10:23:54 | не пусто, по умолчанию now() |
| tscreate | Дата создания документа | timestamp | 2004-10-19 11:23:54 | не пусто, по умолчанию now() |
| amount | Сумма документа | numeric(18,4) | 46464,8787 | не пусто |
| count | количество проданных товаров | double | 654 | ≥ 0 , по умолчанию 1 |

tab. Ledgertype

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|------------------------------------------------------------------------------------------------------------------|---------|--------|-----------|
| sid_ledgertype | Текстовый мнемонический идентификатор, указывающий используемую сторону учета при проведении документа в таблице | varchar | cli | не пусто |
| name | Используемая сторона учета (дебит/кредит) | varchar | дебит | не пусто |

tab. Invoice

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|----------------------------------------------------|-----------|------------------------|----------------------------|
| id_invoice | Уникальный идентификатор счета в таблице | int | 87800 | не пусто |
| id_contract | Уникальный идентификатор записи договора в таблице | int | 43334 | не пусто |
| ts | Время счета | timestamp | 2004-10-19 13:23:54 | не пусто |
| tscreate | Время создания счета | timestamp | 2004-10-19 11:23:54 | не пусто |
| payed | Статус счета: 0 - не оплачен; 1 - оплачен; | bool | 1 | 0 или 1, по умолчанию 0 |

tab. Remain

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|------------------------------------------------------------------------------------------------------------------|---------------|------------------------|-----------|
| id_remain | Уникальный идентификатор записи входящего остатка в таблице | int | 87800 | не пусто |
| sid_ledgertype | Текстовый мнемонический идентификатор, указывающий используемую сторону учета при проведении документа в таблице | varchar | cli | не пусто |
| id_contract | Уникальный идентификатор записи договора в таблице | int | 434 | не пусто |
| id_trx | Уникальный идентификатор проведенной операции в таблице | int | 5454 | не пусто |
| id_revoke | Уникальный идентификатор редактируемого документа в таблице | int | 87877878 | не пусто |
| id_revokedby | Уникальный идентификатор редактирующего документа в таблице | int | 111111 | не пусто |
| ts | Время договора | timestamp | 2004-10-19 13:23:54 | не пусто |
| tscreate | Время создания договора | timestamp | 2004-10-19 11:23:54 | не пусто |
| amount | Сумма договора | numeric(18,4) | 898383,8930 | не пусто |
| note | Примечание | varchar | комментарий | |

tab. Payment

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|-----------------------------------------------------------------|---------------|------------------------|-----------|
| id_payment | Уникальный идентификатор поступления денежных средств в таблице | int | 87800 | не пусто |
| sid_external | Идентификатор для связи с внешними системами | varchar | orcl | не пусто |
| id_contract | Уникальный идентификатор записи договора в таблице | int | 434 | не пусто |
| id_paymenttype | Уникальный идентификатор записи вида платежа в таблице | int | 939847 | не пусто |
| id_trx | Уникальный идентификатор проведенной операции в таблице | int | 5454 | не пусто |
| id_revoke | Уникальный идентификатор редактируемого документа в таблице | int | 87877878 | не пусто |
| id_revokedby | Уникальный идентификатор редактирующего документа в таблице | int | 111111 | не пусто |
| ts | Время договора | timestamp | 2004-10-19 13:23:54 | не пусто |
| tscreate | Время создания договора | timestamp | 2004-10-19 11:23:54 | не пусто |
| tsagent | Время поступления оплаты договора | timestamp | 2004-10-19 12:23:54 | |
| amount | Сумма договора | numeric(18,4) | 898383,8930 | не пусто |
| note | Примечание | varchar | комментарий | |

tab. Transfer

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|----------------------------------------------------------------------------------------------|---------------|------------------------|-----------|
| id_transfer | Уникальный идентификатор перевода денежных средств в таблице | int | 878 | не пусто |
| id_contract_from | Уникальный идентификатор записи договора, с которого переносятся денежные средства в таблице | int | 434 | не пусто |
| id_contract_to | Уникальный идентификатор записи договора, на который переносятся денежные средства в таблице | int | 434444 | не пусто |
| id_trx_from | Уникальный идентификатор проведенной операции списания денежных средств в таблице | int | 5454 | не пусто |
| id_trx_to | Уникальный идентификатор проведенной операции начисления денежных средств в таблице | int | 5333454 | не пусто |
| id_revoke | Уникальный идентификатор редактируемого документа в таблице | int | 87877878 | не пусто |
| id_revokedby | Уникальный идентификатор редактирующего документа в таблице | int | 111111 | не пусто |
| ts | Время договора | timestamp | 2004-10-19 13:23:54 | не пусто |
| tscreate | Время создания договора | timestamp | 2004-10-19 11:23:54 | не пусто |
| amount | Сумма договора | numeric(18,4) | 898383,8930 | не пусто |
| note | Примечание | varchar | комментарий | |

tab. Charge

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|-------------------------------------------------------------|---------------|------------------------|-----------------------------|
| id_charge | Уникальный идентификатор начисления в таблице | int | 878 | не пусто |
| id_trx | Уникальный идентификатор проведенной операции в таблице | int | 5454 | не пусто |
| id_product | Уникальный идентификатор записи продукта в таблице | int | 8787 | не пусто |
| id_contract | Уникальный идентификатор записи договора в таблице | int | 434 | не пусто |
| id_revoke | Уникальный идентификатор редактируемого документа в таблице | int | 87877878 | не пусто |
| id_revokedby | Уникальный идентификатор редактирующего документа в таблице | int | 111111 | не пусто |
| ts | Время договора | timestamp | 2004-10-19 13:23:54 | не пусто |
| tscreate | Время создания договора | timestamp | 2004-10-19 11:23:54 | не пусто |
| amount | Сумма договора | numeric(18,4) | 898383,8930 | не пусто |
| count | количество проданных видов товаров | int | 77777 | не пусто, по умолчанию 1 |
| note | Примечание | varchar | комментарий | |

tab. Invoice_cover_trx

| Название атрибута | Описание | Тип | Пример | Валидация |
|---------------------|------------------------------------------------------------|---------------|----------|-----------|
| id_invoice_covertrx | Уникальный идентификатор записи платежа по счету в таблице | int | 43334 | не пусто |
| id_trx | Уникальный идентификатор проведенной операции в таблице | int | 5454 | не пусто |
| id_invoice | Уникальный идентификатор счета в таблице | int | 87800 | не пусто |
| amount | Время счета | numeric(10,2) | 23535,23 | ≥ 0 |

tab. Invoice_trx

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|------------------------------------------------------------------------|-----|--------|-----------|
| id_invoice_trx | Уникальный идентификатор записи начисления, включенного в счет таблице | int | 43334 | не пусто |
| id_trx | Уникальный идентификатор проведенной операции в таблице | int | 5454 | не пусто |
| id_invoice | Уникальный идентификатор счета в таблице | int | 87800 | не пусто |

tab. Warehouse

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|---------------------------------------------------------------|-----------|------------------------|-----------|
| id_warehouse | Уникальный идентификатор изменения состояния склада в таблице | int | 989 | не пусто |
| name | Название поставленного/проданного товара | varchar | Яблоки | не пусто |
| sid_external | Идентификатор для связи с внешними системами | varchar | psql | не пусто |
| typemov | тип движения товара: 0 - продан; 1 - поступил; | bool | 0 | 0 или 1 |
| tsfrom | Время изменения количества товара на складе | timestamp | 2004-10-19 11:23:54 | не пусто |
| balance | Текущее количество данного товара на складе | int | 8989 | ≥ 0 |

tab. Properties

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|---------------------------------------------------------------|---------|--------|-----------|
| id_properties | Уникальный идентификатор свойств товара в таблице | int | 42 | не пусто |
| id_warehouse | Уникальный идентификатор изменения состояния склада в таблице | int | 909 | не пусто |
| kind | Вид товара | varchar | овоци | не пусто |
| storing | Время хранения товара (в днях) | int | 34 | ≥ 0 |

tab. Supply

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|---------------------------------------------------------------|------|------------|------------------------------|
| id_supply | Уникальный идентификатор поставки в таблице | int | 42 | не пусто |
| id_properties | Уникальный идентификатор характеристик товара в таблице | int | 33 | не пусто |
| id_warehouse | Уникальный идентификатор изменения состояния склада в таблице | int | 989 | не пусто |
| dtfrom | Дата поступления на склад | date | 1999-01-08 | не пусто, не будущее |
| dtto | Дата истечения срока годности | date | 1999-01-09 | не меньше dtfrom |
| isfresh | признак просрочки товара: 0 - просроченный; 1 - свежий; | bool | 0 | 0 или 1 |
| typemov | тип движения товара: 0 - продан; 1 - поступил; | bool | 0 | 0 или 1 |
| count | количество поступивших/проданных единиц товара | int | 2 | ≥ 0 , по умолчанию 1 |

tab. Product

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|----------------------------------------------------------------------------------------------------------|-----|--------|------------------------------|
| id_product | Уникальный идентификатор записи продукта в таблице | int | 8787 | не пусто |
| id_supply | Уникальный идентификатор записи поставки в таблице | int | 8888 | не пусто |
| count | количество товара – это сумма всех с одинаковым id_properties из tab. Supply | int | 7676 | ≥ 0 , по умолчанию 1 |
| countgood | количество свежего товара – это сумма всех с одинаковым id_properties из tab. Supply и isfresh = 1 | int | 757 | ≥ 0 |
| countbad | количество просроченного товара – это сумма всех с одинаковым id_properties из tab. Supply и isfresh = 0 | int | 8883 | ≥ 0 |

tab. Paymenttype

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|-------------------------------------------------|---------|----------|-----------|
| id_paymenttype | Уникальный идентификатор вида платежа в таблице | int | 56923876 | не пусто |
| name | Название вида платежа | varchar | наличные | не пусто |

tab. Price

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|----------------------------------------------------|---------------|------------|-------------------------|
| id_price | Уникальный идентификатор стоимости услуг в таблице | int | 42666 | не пусто |
| id_product | Уникальный идентификатор записи продукта в таблице | int | 8787 | не пусто |
| dtfrom | Дата поступления на склад | date | 1999-01-08 | не пусто, не будущее |
| dtto | Дата истечения срока годности | date | 1999-01-09 | не меньше dtfrom |
| amount | Стоимость товара за одну условную единицу | numeric(10,2) | 23222,33 | не пусто |

tab. Trxtype

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------|---------|------------------|-----------|
| sid_trxtype | Текстовый мнемонический идентификатор, указывающий тип проводки в таблице | varchar | dbt | не пусто |
| name | Название типа проводки | varchar | дебит | не пусто |
| sid_ledgertype | Текстовый мнемонический идентификатор, указывающий используемую сторону учета по умолчанию при проведении документа в таблице | varchar | cli | не пусто |
| optype | Тип операции | varchar | Списание средств | не пусто |

tab. Contract_balance

| Название атрибута | Описание | Тип | Пример | Валидация |
|---------------------|------------------------------------------------------------|---------------|------------------------|---------------------------------|
| id_contract_balance | Уникальный идентификатор записи баланса договора в таблице | int | 87800 | не пусто |
| id_contract | Уникальный идентификатор записи договора в таблице | int | 434 | не пусто |
| id_trx | Уникальный идентификатор проведенной операции в таблице | int | 5454 | не пусто |
| ts | Время изменения баланса | timestamp | 2004-10-19 13:23:54 | не пусто, по умолчанию now() |
| balance | Текущий баланс | numeric(18,4) | 898383,8930 | ≥ 0 |

tab. Person

| Название атрибута | Описание | Тип | Пример | Валидация |
|-------------------|----------------------------------------------------|---------|----------------------------|-----------|
| id_person | Уникальный идентификатор записи клиента в таблице | int | 314 | не пусто |
| id_contract | Уникальный идентификатор записи договора в таблице | int | 444 | не пусто |
| name | ФИО | varchar | Еричев Алексей Олегович | не пусто |
| note | примечание | varchar | комментарий | |

4. Заключение

В ходе проведённой работы была реализована реляционная база данных, обеспечивающая корректную работу схемы биллинга склада продуктов. В ходе создания системы базы данных были приведены подробные описания и матрицы валидаций для всех её таблиц. Достигнута цель обеспечения высокой эффективности хранения данных.

Список литературы

1. **Коннолли Томас, Бегг Каролин** *Базы данных. Проектирование, реализация и сопровождение. Теория и практика.* — Вильямс, 2017. — 1440 стр.
2. **Гектор Гарсиа-Молина, Джефффри Ульман, Дженнифер Уидом** *Системы баз данных. Полный курс* — Вильямс, 2017. — 1088 стр.
3. **Мартин Клеппман** *Высоконагруженные приложения. Программирование, масштабирование, поддержка.* — СПб.: Питер, 2018. — 640 стр.
4. **Regina O. Obe, Leo S. Hsu** *PostgreSQL Up and Running, 3rd Edition.* — O'Reilly Media, Inc., 2017. — 314 p.
5. **Hunter, Jane M. and Thiebaud, Maud E.** *Telecommunications Billing Systems: Implementing and Upgrading for Profitability.* — N.Y.: McGraw-Hill, 2003. — 458 p.

5. Приложение

```
CREATE SCHEMA CourseWork;
```

```
CREATE TABLE CourseWork.ledgertype (
    sid VARCHAR NOT NULL,
    name VARCHAR NOT NULL,
    CONSTRAINT CourseWork_ledgertype_pk PRIMARY KEY (sid)
);
COMMENT ON TABLE CourseWork.ledgertype IS 'Тип бухгалтерского счета';
```

```
CREATE SEQUENCE CourseWork.paymenttype_id_seq;
```

```
CREATE TABLE CourseWork.paymenttype (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.paymenttype_id_seq'),
    name VARCHAR NOT NULL,
    CONSTRAINT paymenttype_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.paymenttype IS 'Типы платежей';
```

```
ALTER SEQUENCE CourseWork.paymenttype_id_seq OWNED BY CourseWork.paymenttype.id;
```

```
CREATE SEQUENCE CourseWork.warehouse_id_seq;
```

```
CREATE TABLE CourseWork.warehouse (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.warehouse_id_seq'),
    name VARCHAR NOT NULL,
    sid_external VARCHAR NOT NULL,
    typemov BOOLEAN NOT NULL,
    tsfrom TIMESTAMP NOT NULL,
    balance BIGINT NOT NULL,
    CONSTRAINT warehouse_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.warehouse IS 'Изменения состояния склада';
```

```
ALTER SEQUENCE CourseWork.warehouse_id_seq OWNED BY CourseWork.warehouse.id;
```

```
CREATE SEQUENCE CourseWork.properties_id_seq;
```

```
CREATE TABLE CourseWork.properties (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.properties_id_seq'),
    id_warehouse BIGINT NOT NULL,
    kind VARCHAR NOT NULL,
    storing BIGINT NOT NULL,
    CONSTRAINT properties_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.properties IS 'Характеристики товара';
```



```
ALTER SEQUENCE CourseWork.properties_id_seq OWNED BY CourseWork.properties.id;
```

```
CREATE SEQUENCE CourseWork.supply_id_seq;
```

```
CREATE TABLE CourseWork.supply (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.supply_id_seq'),
    id_properties BIGINT NOT NULL,
    id_warehouse BIGINT NOT NULL,
    dtfrom DATE DEFAULT now() NOT NULL,
    dtto DATE NOT NULL,
    isfresh BOOLEAN NOT NULL,
    typemov BOOLEAN NOT NULL,
    count BIGINT DEFAULT 1 NOT NULL ,
    CONSTRAINT supply_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.supply IS 'Поствки товара';
```

```
ALTER SEQUENCE CourseWork.supply_id_seq OWNED BY CourseWork.supply.id;
```

```
CREATE SEQUENCE CourseWork.product_id_seq;
```

```
CREATE TABLE CourseWork.product (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.product_id_seq'),
    id_supply BIGINT NOT NULL,
    count BIGINT DEFAULT 1 NOT NULL,
    countgood BIGINT,
    countbad BIGINT,
    CONSTRAINT product_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.product IS 'Количественные характеристики товара';
```

```
ALTER SEQUENCE CourseWork.product_id_seq OWNED BY CourseWork.product.id;
```

```
CREATE SEQUENCE CourseWork.price_id_seq;
```

```
CREATE TABLE CourseWork.price (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.price_id_seq'),
    id_product BIGINT NOT NULL,
    dtfrom DATE NOT NULL,
    dtto DATE,
    amount NUMERIC(18,4) NOT NULL,
    CONSTRAINT price_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.price IS 'Стоимость услуг';
```

```
ALTER SEQUENCE CourseWork.price_id_seq OWNED BY CourseWork.price.id;
```

```

CREATE TABLE CourseWork.trxtype (
    sid VARCHAR NOT NULL,
    name VARCHAR NOT NULL,
    sid_ledgertype VARCHAR NOT NULL,
    optype VARCHAR NOT NULL,
    CONSTRAINT trxtype_pk PRIMARY KEY (sid)
);
COMMENT ON TABLE CourseWork.trxtype IS 'Тип проведенных операций';

```

```

CREATE SEQUENCE CourseWork.contract_id_seq;

```

```

CREATE TABLE CourseWork.contract (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.contract_id_seq'),
    number BIGINT NOT NULL,
    balance NUMERIC(18,4) NOT NULL,
    dtfrom DATE NOT NULL,
    dtto DATE,
    note VARCHAR,
    CONSTRAINT contract_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.contract IS 'Договоры с клиентами';

```

```

ALTER SEQUENCE CourseWork.contract_id_seq OWNED BY CourseWork.contract.id;

```

```

CREATE SEQUENCE CourseWork.invoice_id_seq;

```

```

CREATE TABLE CourseWork.invoice (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.invoice_id_seq'),
    id_contract BIGINT NOT NULL,
    ts TIMESTAMP NOT NULL,
    tscreate TIMESTAMP NOT NULL,
    paid BOOLEAN DEFAULT false NOT NULL,
    CONSTRAINT invoice_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.invoice IS 'Документы к оплате';

```

```

ALTER SEQUENCE CourseWork.invoice_id_seq OWNED BY CourseWork.invoice.id;

```

```

CREATE SEQUENCE CourseWork.trx_id_seq;

```

```

CREATE TABLE CourseWork.trx (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.trx_id_seq'),
    sid_ledgertype VARCHAR NOT NULL,
    sid_trxtype VARCHAR NOT NULL,
    id_contract BIGINT NOT NULL,
    ts TIMESTAMP DEFAULT now() NOT NULL,
    tscreate TIMESTAMP DEFAULT now() NOT NULL,
    amount NUMERIC(18,4) NOT NULL,

```

```

        count BIGINT DEFAULT 1,
        CONSTRAINT trx_pk PRIMARY KEY (id)
    );
COMMENT ON TABLE CourseWork.trx IS 'Проводки';

```

```

ALTER SEQUENCE CourseWork.trx_id_seq OWNED BY CourseWork.trx.id;

```

```

CREATE SEQUENCE CourseWork.contract_balance_id_seq;

```

```

CREATE TABLE CourseWork.contract_balance (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.contract_balance_id_seq'),
    id_contract BIGINT NOT NULL,
    id_trx BIGINT NOT NULL,
    tsfrom TIMESTAMP DEFAULT now() NOT NULL,
    balance NUMERIC(18,4) NOT NULL,
    CONSTRAINT contract_balance_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.contract_balance IS 'История изменения баланса';

```

```

ALTER SEQUENCE CourseWork.contract_balance_id_seq OWNED BY CourseWork.contract_balance.id;

```

```

CREATE SEQUENCE CourseWork.invoice_cover_trx_id_seq;

```

```

CREATE TABLE CourseWork.invoice_cover_trx (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.invoice_cover_trx_id_seq'),
    id_trx BIGINT NOT NULL,
    id_invoice BIGINT NOT NULL,
    amount NUMERIC(18,4) NOT NULL,
    CONSTRAINT invoice_cover_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.invoice_cover_trx IS 'Покрытие счетов платежами';

```

```

ALTER SEQUENCE CourseWork.invoice_cover_trx_id_seq OWNED BY CourseWork.invoice_cover_trx.id;

```

```

CREATE SEQUENCE CourseWork.invoice_trx_id_seq;

```

```

CREATE TABLE CourseWork.invoice_trx (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.invoice_trx_id_seq'),
    id_trx BIGINT NOT NULL,
    id_invoice BIGINT NOT NULL,
    CONSTRAINT invoice_trx_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.invoice_trx IS 'Начисления включенные в счет';

```

```

ALTER SEQUENCE CourseWork.invoice_trx_id_seq OWNED BY CourseWork.invoice_trx.id;

```

```

CREATE SEQUENCE CourseWork.transfer_id_seq;

```

```

CREATE TABLE CourseWork.transfer (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.transfer_id_seq'),
    id_contract_from BIGINT NOT NULL,
    id_contract_to BIGINT NOT NULL,
    id_trx_from BIGINT NOT NULL,
    id_trx_to BIGINT NOT NULL,
    id_revoke BIGINT NOT NULL,
    id_revokedby BIGINT NOT NULL,
    ts TIMESTAMP NOT NULL,
    tscreate TIMESTAMP NOT NULL,
    amount NUMERIC(18,4) NOT NULL,
    note VARCHAR,
    CONSTRAINT transfer_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.transfer IS 'Переносы денежных средств';

```

```

ALTER SEQUENCE CourseWork.transfer_id_seq OWNED BY CourseWork.transfer.id;

```

```

CREATE SEQUENCE CourseWork.remain_id_seq;

```

```

CREATE TABLE CourseWork.remain (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.remain_id_seq'),
    sid_ledgertype VARCHAR NOT NULL,
    id_contract BIGINT NOT NULL,
    id_trx BIGINT NOT NULL,
    id_revoke BIGINT,
    id_revokedby BIGINT,
    ts TIMESTAMP NOT NULL,
    tscreate TIMESTAMP NOT NULL,
    amount NUMERIC(18,4) NOT NULL,
    note VARCHAR,
    CONSTRAINT remain_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.remain IS 'Корректировки и остатки балансов возникающие при миграции';

```

```

ALTER SEQUENCE CourseWork.remain_id_seq OWNED BY CourseWork.remain.id;

```

```

CREATE SEQUENCE CourseWork.charge_id_seq;

```

```

CREATE TABLE CourseWork.charge (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.charge_id_seq'),
    id_contract BIGINT NOT NULL,
    id_trx BIGINT NOT NULL,
    id_product BIGINT NOT NULL,
    id_revoke BIGINT,
    id_revokedby BIGINT NOT NULL,
    ts TIMESTAMP NOT NULL,
    tscreate TIMESTAMP NOT NULL,
    amount NUMERIC(18,4) NOT NULL,
    count BIGINT DEFAULT 1 NOT NULL,

```

```

        note VARCHAR,
        CONSTRAINT charge_pk PRIMARY KEY (id)
    );
COMMENT ON TABLE CourseWork.charge IS 'Начисления';

```

```

ALTER SEQUENCE CourseWork.charge_id_seq OWNED BY CourseWork.charge.id;

```

```

CREATE SEQUENCE CourseWork.payment_id_seq;

```

```

CREATE TABLE CourseWork.payment (
    id BIGINT NOT NULL DEFAULT nextval('CourseWork.payment_id_seq'),
    sid_external VARCHAR,
    id_contract BIGINT NOT NULL,
    id_paymenttype BIGINT NOT NULL,
    id_trx BIGINT NOT NULL,
    id_revoke BIGINT,
    id_revokedby BIGINT,
    ts TIMESTAMP NOT NULL,
    tscreate TIMESTAMP NOT NULL,
    tsagent TIMESTAMP,
    amount NUMERIC(18,4) NOT NULL,
    note VARCHAR,
    CONSTRAINT payment_pk PRIMARY KEY (id)
);
COMMENT ON TABLE CourseWork.payment IS 'Платежи';

```

```

ALTER SEQUENCE CourseWork.payment_id_seq OWNED BY CourseWork.payment.id;

```

```

ALTER TABLE CourseWork.trxtype ADD CONSTRAINT ledgertype_trxtype_fk
FOREIGN KEY (sid_ledgertype)
REFERENCES CourseWork.ledgertype (sid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.trx ADD CONSTRAINT ledgertype_trx_fk
FOREIGN KEY (sid_ledgertype)
REFERENCES CourseWork.ledgertype (sid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.remain ADD CONSTRAINT ledgertype_remain_fk
FOREIGN KEY (sid_ledgertype)
REFERENCES CourseWork.ledgertype (sid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.payment ADD CONSTRAINT paymenttype_payment_fk
FOREIGN KEY (id_paymenttype)
REFERENCES CourseWork.paymenttype (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.properties ADD CONSTRAINT warehouse_properties_fk
FOREIGN KEY (id_warehouse)
REFERENCES CourseWork.warehouse (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.supply ADD CONSTRAINT warehouse_supply_fk
FOREIGN KEY (id_warehouse)
REFERENCES CourseWork.warehouse (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.supply ADD CONSTRAINT properties_supply_fk
FOREIGN KEY (id_properties)
REFERENCES CourseWork.properties (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.product ADD CONSTRAINT supply_product_fk
FOREIGN KEY (id_supply)
REFERENCES CourseWork.supply (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.charge ADD CONSTRAINT product_charge_fk
FOREIGN KEY (id_product)
REFERENCES CourseWork.product (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.price ADD CONSTRAINT product_price_fk
FOREIGN KEY (id_product)
REFERENCES CourseWork.product (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.trx ADD CONSTRAINT trxtype_trx_fk
FOREIGN KEY (sid_trxtype)
REFERENCES CourseWork.trxtype (sid)

```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.trx ADD CONSTRAINT contract_trx_fk
FOREIGN KEY (id_contract)
REFERENCES CourseWork.contract (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.payment ADD CONSTRAINT contract_payment_fk
FOREIGN KEY (id_contract)
REFERENCES CourseWork.contract (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.charge ADD CONSTRAINT contract_charge_fk
FOREIGN KEY (id_contract)
REFERENCES CourseWork.contract (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.remain ADD CONSTRAINT contract_remain_fk
FOREIGN KEY (id_contract)
REFERENCES CourseWork.contract (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.transfer ADD CONSTRAINT contract_from_transfer_fk
FOREIGN KEY (id_contract_from)
REFERENCES CourseWork.contract (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.transfer ADD CONSTRAINT contract_to_transfer_fk
FOREIGN KEY (id_contract_to)
REFERENCES CourseWork.contract (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.contract_balance ADD CONSTRAINT contract_balance_fk
FOREIGN KEY (id_contract)
REFERENCES CourseWork.contract (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION

```

NOT DEFERRABLE;

```
ALTER TABLE CourseWork.invoice ADD CONSTRAINT contract_invoice_fk
FOREIGN KEY (id_contract)
REFERENCES CourseWork.contract (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE CourseWork.invoice_trx ADD CONSTRAINT invoice_invoice_trx_fk
FOREIGN KEY (id_invoice)
REFERENCES CourseWork.invoice (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE CourseWork.invoice_cover_trx ADD CONSTRAINT invoice_invoice_cover_trx_fk
FOREIGN KEY (id_invoice)
REFERENCES CourseWork.invoice (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE CourseWork.payment ADD CONSTRAINT trx_payment_fk
FOREIGN KEY (id_trx)
REFERENCES CourseWork.trx (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE CourseWork.charge ADD CONSTRAINT trx_charge_fk
FOREIGN KEY (id_trx)
REFERENCES CourseWork.trx (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE CourseWork.remain ADD CONSTRAINT trx_remain_fk
FOREIGN KEY (id_trx)
REFERENCES CourseWork.trx (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE CourseWork.transfer ADD CONSTRAINT trx_to_transfer_fk
FOREIGN KEY (id_trx_to)
REFERENCES CourseWork.trx (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE CourseWork.transfer ADD CONSTRAINT trx_from_transfer_fk
```



```

FOREIGN KEY (id_trx_from)
REFERENCES CourseWork.trx (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.invoice_trx ADD CONSTRAINT trx_invoice_trx_fk
FOREIGN KEY (id_trx)
REFERENCES CourseWork.trx (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.invoice_cover_trx ADD CONSTRAINT trx_invoice_cover_trx_fk
FOREIGN KEY (id_trx)
REFERENCES CourseWork.trx (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.contract_balance ADD CONSTRAINT trx_balance_fk
FOREIGN KEY (id_trx)
REFERENCES CourseWork.trx (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.transfer ADD CONSTRAINT revokedby_transfer_fk
FOREIGN KEY (id_revokedby)
REFERENCES CourseWork.transfer (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.transfer ADD CONSTRAINT revoke_transfer_fk
FOREIGN KEY (id_revoke)
REFERENCES CourseWork.transfer (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.remain ADD CONSTRAINT revoke_remain_fk
FOREIGN KEY (id_revoke)
REFERENCES CourseWork.remain (id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```

ALTER TABLE CourseWork.remain ADD CONSTRAINT revokedby_remain_fk
FOREIGN KEY (id_revokedby)
REFERENCES CourseWork.remain (id)
ON DELETE NO ACTION

```

```
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE CourseWork.charge ADD CONSTRAINT revoke_charge_fk  
FOREIGN KEY (id_revoke)  
REFERENCES CourseWork.charge (id)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE CourseWork.charge ADD CONSTRAINT revokedby_charge_fk  
FOREIGN KEY (id_revokedby)  
REFERENCES CourseWork.charge (id)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE CourseWork.payment ADD CONSTRAINT revoke_payment_fk  
FOREIGN KEY (id_revoke)  
REFERENCES CourseWork.payment (id)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE CourseWork.payment ADD CONSTRAINT revokedby_payment_fk  
FOREIGN KEY (id_revokedby)  
REFERENCES CourseWork.payment (id)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```