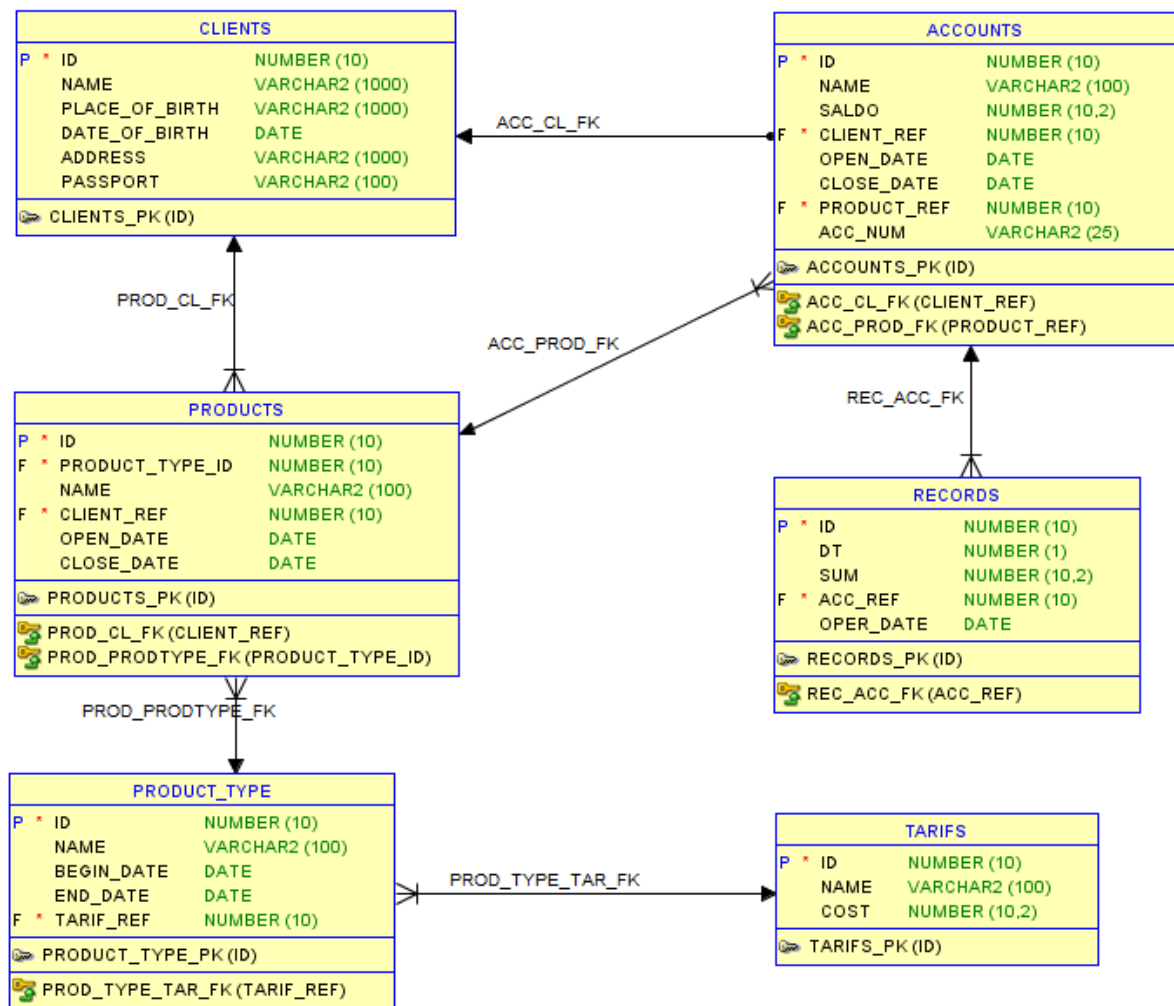


Тестовое задание курса разработчика.

Ниже представлена упрощенная модель учета операций в Банке.



Описание модели.

1. [CLIENTS] – таблица содержит основную информацию по клиентам Банка:
 - ID – уникальный идентификатор (первичный ключ);
 - NAME – ФИО клиента;
 - PLACE_OF_BIRTH – место рождения клиента;
 - DATE_OF_BIRTH – дата рождения клиента;
 - ADDRESS – адрес проживания клиента;
 - PASSPORT – паспортные данные клиента;

2. [PRODUCTS] – таблица содержит информацию о продуктах, открытых для клиента в Банке:
 - ID – уникальный идентификатор (первичный ключ);
 - PRODUCT_TYPE_ID – ссылка на тип продукта;
 - NAME – наименование продукта;
 - CLIENT_REF – ссылка на клиента;
 - OPEN_DATE – дата открытия продукта;
 - CLOSE_DATE – дата закрытия продукта;
3. [PRODUCT_TYPE] - таблица содержит информацию о типах продуктов, которые доступны для открытия клиенту:
 - ID – уникальный идентификатор (первичный ключ);
 - NAME – наименование типа продукта;
 - BEGIN_DATE – дата начала действия типа продукта;
 - END_DATE – дата окончания действия типа продукта;
 - TARIF_REF – ссылка на тариф;
4. [ACCOUNTS] - таблица содержит информацию о счетах, открытых для клиента в Банке:
 - ID – уникальный идентификатор (первичный ключ);
 - NAME – наименование счета;
 - SALDO – остаток по счету;
 - CLIENT_REF – ссылка на клиента;
 - OPEN_DATE – дата открытия счета;
 - CLOSE_DATE – дата закрытия счета;
 - PRODUCT_REF – ссылка на продукт, в рамках которого открыт счет;
 - ACC_NUM – номер счета.
5. [RECORDS] – таблица содержит информацию операциях по счетам:
 - ID – уникальный идентификатор (первичный ключ);
 - DT – признак дебетования счета, может принимать значения 1 и 0, в случае когда значение равно 1 – остаток по счету уменьшается (дебет), в случае когда значение равно 0 – остаток по счету увеличивается (кредит);
 - ACC_REF – ссылка на счет, по которому происходит движение;
 - OPER_DATE – дата операции;
 - SUM – сумма операции;
6. [TARIFS] – таблица содержит информацию о тарифах за операции по счетам:
 - ID – уникальный идентификатор (первичный ключ);
 - NAME – наименование тарифа;
 - COST – сумма тарифа.

Описание процесса.

Обслуживание клиента в банке начинается с заведения карточки клиента (таблица CLIENTS), далее клиент выражает завести тот или иной продукт в банке - КРЕДИТ, ДЕПОЗИТ, КАРТА (таблица PRODUCT_TYPE). После оформления документов в банке создается экземпляр продукта (таблица PRODUCTS), в рамках продукта открывается один или несколько счетов (таблица ACCOUNTS) с остатком равным 0.

Далее в случае, если оформлен продукт типа КРЕДИТ по счету продукта проходит дебетовая операция – банк выдает деньги клиенту (в таблице RECORDS появляется запись с полем DT = 1 и суммой зачисления, запись в таблице RECORDS влияет на поле SALDO таблицы ACCOUNTS). Если оформлен продукт ДЕПОЗИТ или КАРТА по счету клиента проходит кредитовая операция – клиент вносит средства на счета (в таблице

RECORDS появляется запись с полем DT = 0 и суммой зачисления, запись в таблице RECORDS влияет на поле SALDO таблицы ACCOUNTS).

После чего клиент в случае, если ему открыт продукт типа КРЕДИТ, вносит средства на счет, погашая кредит, а если продукт типа ДЕПОЗИТ или КАРТА, может списывать средства со счета. После полного погашения продукта типа КРЕДИТ, выдача кредита может происходить снова, и клиент нужно опять осуществлять погашения. Если у клиента продукта типа ДЕПОЗИТ или КАРТА, клиент в любое время может внести средства.

Задания.

1. Подготовьте DDL-скрипты создания объектов для приведённой модели: создание таблиц, первичных, уникальных, внешних ключей и т.д.
2. Заполните таблицы примерами из приложенного скрипта



ex_table_st.sql

3. Подготовьте скрипты заполнения таблиц тестовыми данными, достаточными для выполнения заданий ниже. Сформируйте отчет, который содержит все счета, относящиеся к продуктам типа ДЕПОЗИТ, принадлежащих клиентам, у которых имеется более одного открытого продукта.
4. Сформируйте выборку, которая содержит счета, относящиеся к продуктам типа ДЕПОЗИТ или КАРТА, по которым были операции внесения средств на счет в рамках одного произвольного дня.
5. Найдите клиентов, у которых открыт продукт типа КРЕДИТ, и у которых сумма всех дебетовых операций по таким продуктам превышает сумму всех кредитовых операций.
6. В результате сбоя в базе данных разъехалась информация между остатками и операциями по счетам. Напишите нормализацию (процедуру выравнивающую данные), которая найдет такие счета и восстановит остатки по счету. Необходимо таким клиентам выплатить компенсацию в размере 1% от восстановленного остатка счета (для счетов, относящихся к продукту типа ДЕПОЗИТ).
7. Сформируйте выборку, которая содержит информацию о клиентах, которые полностью погасили кредит, но при этом не закрыли продукт (по продукту есть операция новой выдачи кредита). Выведите в выборке наименование продукта и сумму кредита и количество операций его гашения.
8. Закройте продукты (установите дату закрытия равную текущей) типа КРЕДИТ. Закрытие продукта происходит в следующих условиях:
 - клиент должен полностью погасить все долги по кредиту (остаток на счете продукта должен стать равным 0)
 - система должна проверить, что все операции по кредиту завершены (сумма всех дебетовых операций по кредитному продукту равна сумме всех кредитовых операций)
9. Закройте продукт типа КАРТА, который был открыт первым, если у клиента имеется более 20-ти карт с ненулевым остатком. Средства со счета закрываемой карты перечислить на карту, открытую второй.

10. В модель данных добавьте сумму договора по продукту. Заполните поле для всех продуктов суммой минимальной дебетовой операции по счету для продукта типа КРЕДИТ, и суммой средней кредитовой операции по счету продукта для продукта типа ДЕПОЗИТ или КАРТА.

Примечание. Порядковый номер задач не связан с их сложностью. Для решения указанных задач могут потребоваться различные навыки и уровень знаний SQL, поэтому допустимо их частичное решение. Предлагается выполнить задания, какие и как можете. Использовать можно **любую** реляционную СУБД, но **обязательно укажите, какую использовали.**

Спасибо за ваши решения. Удачи и до встречи на курсе! 😊