Тестовое задание на позицию “Стажера SQL”

Задание № 1

Сложность: низкая.

Задание является обязательным.

Расскажите нам, почему вам интересно работать в компании Wildberries и почему хотите стать специалистом SQL.

Задание № 2

Сложность: низкая.

Задание является обязательным.

Требуется спроектировать базу данных автомата по продажи напитков.

База данных должна удовлетворять следующим условиям:

* База данных должна быть разработана на postgresql;
* База данных должна содержать сведения об ассортименте: какие напитки продаются, какое количество напитков можно продать и какова стоимость каждого напитка;
* Т.к. это автомат для продажи напитков, в нем должен быть реализован кошелек покупателя, где содержится информация о том, какие монеты и в каком количестве покупатель опустил в аппарат;
* Так же необходимо предусмотреть второй кошелек самого автомата, который содержит данные о том, сколько автомат содержит монет и какого номинала. Данный кошелек требуется для выдачи сдачи автоматом;
* В кошелек покупателя и в кошелек автомата можно опускать монеты только строго определенного достоинства. Для тестового задания это монеты номиналом 10 р., 5 р., 2 р., 1 р. При этом база должна быть спроектирована таким способом, чтобы позволять гибко настраивать перечень допустимых монет;
* Помимо указанных выше таблиц, можете добавить любое количество дополнительных таблиц, если оно того требует;
* В рамках данного задания необходимо выполнить следующие дополнительные требования:
  + Необходимо нарисовать модель базы данных в любой системе проектирования либо от руки. Необходимо прислать только рисунок;
  + Необходимо прислать рабочий скрипт создания всех объектов базы данных с названием файла task2.sql;
  + В скрипте добавить комментарии к полям, таблицам.
  + Необходимо, чтобы файл так же содержал скрипт первичного заполнения всех необходимых таблиц. При этом кошелек покупателя должен быть пустым, a кошелек автомата содержать набор монет каждого номинала по 10 шт.
  + В автомате должны быть доступны напитки: чай – 25 р., капучино – 39 р., какао – 23 р., шоколад – 31 р.;

Задание № 3

Сложность: средняя.

Задание является не обязательным.

Для тех таблиц, которые были разработаны в задании № 2, необходимо написать скрипты создания индексов с названием по шаблону **IX\_<Название таблицы>\_<список полей в том же порядке, что в индексе>**.

Добавить первичные ключи по шаблону **PK\_<Название таблицы>\_<список полей в том же порядке, что в индексе>**.

Добавить внешние ключи по шаблону **FK\_<Название таблицы>\_<список полей в том же порядке, что в индексе>**.

Рабочие скрипты индексов должны быть добавлены в тот же самый файл, где создаются сами таблицы (task2.sql). Где каждый индекс требуется разместить непосредственно под своей таблицей.

Задание № 4

Сложность: низкая.

Задание является обязательным.

Необходимо разработать работающую функцию, которая будет вызываться автоматически, когда пользователь опускает монеты в автомат для пополнения кошелька покупателя.

Функция должна удовлетворять следующим условиям:

* Функция должна быть разработана на postgresql.
* Функция должна быть в отдельном файле c названием task3.sql;
* В функции добавить пошаговые комментарии.
* Функция должна быть написана понятным структурированным образом;

Задание № 5

Сложность: высокая.

Задание не является обязательным, если решено 6 задание.

Необходимо разработать работающую функцию, которая реализует операцию по продаже выбранного напитка.

Функция должна удовлетворять следующим условиям:

* Функция должна быть разработана на postgresql;
* При продаже необходимо убедиться, что напиток есть в продаже;
* Нужно проверить, что сумма на кошельке покупателя достаточна для продажи напитка;
* Проверить, что после продажи напитка автомат сможет дать сдачу (у автомата есть требуемая сумма и монеты подходящего достоинства);
* После совершения покупки количество напитков в наличии должно уменьшиться;
* После совершения покупки, все монеты из кошелька покупателя должны перейти в кошелек автомата;
* В случае если покупку совершить невозможно, функция в качестве результата должна вернуть – -1, в случае успеха – 0;
* Функция должна быть в отдельном файле c названием task5.sql;
* В функции добавить пошаговые комментарии.
* Функция должна быть написана понятным структурированным образом;

Задание № 6

Сложность: низкая.

Задание не является обязательным, если решено 5 задание.

Необходимо разработать работающую функцию, которая реализует операцию выдачи сдачи после покупки напитка.

Функция должна удовлетворять следующим условиям:

* Функция должна быть разработана на postgresql;
* Функция в качестве результата должна вернуть набор данных: монеты какого номинала и какое их количество автомат должен вернуть покупателю;
* Если подходящую сумму невозможно вернуть, функция должна вернуть пустое множество;
* Сдача выдается только из кошелька автомата;
* Главное требование: автомат при выдаче сдачи должен стремиться сначала отдавать самые крупные монеты, потом более мелкие, потом еще более мелкие. Например, если нам нужно сдать 23 рубля сдачи, то мы сначала стремимся выдать как можно большую сумму 10-ти рублевыми монетами, потом пяти, двух и только в конце рублевыми монетами;
* В случае успешного выполнения, операция должна уменьшить количество монет в кошельке автомата на то количество, которое возвращается как сдача;
* Функция должна быть в отдельном файле с названием task6.sql.
* В функции добавить пошаговые комментарии.
* Функция должна быть написана понятным структурированным образом;

Задание № 7

Сложность: низкая.

Задание является обязательным.

Необходимо выяснить из нескольких источников и объяснить, что такое «План запроса», какова его цель применения в повседневной работе sql-программиста.

Необходимо уложится в один короткий абзац.

Задание № 8

Сложность: низкая.

Задание является обязательным.

Необходимо выяснить из нескольких источников и объяснить, что такое «Нормализация и денормализация базы данных», какова цель их применения, в чем заключается разница между денормализацией и ненормализацией.

Необходимо уложится в один короткий абзац.

Ваши комментарии к решениям: