1. Вьюшка Client\_Offer (Коммерческое предложение)

Атрибуты: Client\_name, Client.surname, Client\_Request.request, Client.email, Product.name AS Product

Родительские таблицы: Client, Client\_Request, Request\_Product, Product

* в поле Product попадают только те продукты, что находятся в одной категории с заказом клиента

1. Вьюшка Statement\_Invoice (Ведомость отгрузки)

Атрибуты: Client.adress, Product.name, Invoice\_Product.count, Invoice\_Product.loaded

Родительские таблицы: Invoice, Client\_Order, Client, Invoice\_Product, Product

* В ведомость попадают только те заказы, что утверждены начальством (agreed = TRUE) и только те товары, что еще не отгружены (loaded = FALSE)
* Цель данной вьюшки – показать кладовщику кому, что и сколько необходимо доставить ~~минимизируя транспортные расходы~~
* Когда кладовщик отгружает товар, он ставит галочку на Invoice\_Product.loaded, тем самым запуская триггер update\_Products\_Ordersи подтверждает то, что товар уже отправлен

1. Триггер delete\_Client

Привязан: Client

Условие: После удаления

* После удаления строки из таблицы помещает клиента в архив, который по структуре полностью идентичен основной таблице

1. Триггер create\_Order

Привязан: Client\_Request

Условие: После обновления

* Если после обновления таблицы Client\_Request checked = TRUE AND approved = TRUE, то данная заявка становится заказом, и строка переходит в Client\_Order с новым id, который может не совпадать с id заявки
* Все соответствующие товары переносе заявки в заказ также переходят из Request\_Product в Order\_Product с учетом нового id сформированного заказа
* При создании заказа количество остатка rest в таблице Order\_Product равен количеству заказанного продукта, так как пока мы клиенту ничего не доставили

1. Триггер update\_Products\_Orders

Привязан: Invoice\_Product

Условие: После обновления

* Триггер сработает только тогда, когда кладовщик поставит loaded = TRUE через вьюшку Statement\_Invoice
* В результате пересчитывается общее количество товара на складе в зависимости от количества отгруженного товара (Product.count = Product.count - NEW.count) и остаток соответствующего заказа

(rest = rest - NEW.count)

* Если после пересчета остатка, все остатки данного заказа равны нулю, то запускается процедура удаления заказа delete\_order(@order), где @order – id удаляемого заказа

1. Процедура delete\_order(IN del\_order\_id INT)

* Из Client\_Order удаляется заказ по del\_order\_id
* Из Order\_Product удаляется продукт по del\_order\_id
* В Contract\_Archive помещается контракт по del\_order\_id
* Из Contract удаляется контракт по del\_order\_id