1) **Select name\_product,name\_company,price,quantity**

**from market,company, product**

**where market.id\_product=company.id\_product and company.id\_company=product.id\_companies and market.id\_product = product.id\_product**

// вывод маркет продуктов

2) **Select SUM(PRICE) as SUM\_of\_prod**

**FROM market**

**WHERE market.quantity < 50**

**union**

**SELECT count(quantity) AS avg\_of\_quan**

**FROM market x, company z**

**WHERE x.id\_product = z.id\_product and z.name\_product like 'A%'**

//вывод на какую сумму в маркете товаров и количество продуктовв начинающиеся на A

3)

**Delete from market**

**Where id\_product in (Select \* from prosrochka)**

////////////

**Insert into existed\_product**

**select id\_product**

**from company**

**where company.id\_product in(select\* from prosrochka\_id)**

**Create view prosrochka\_id AS**

**select id\_product**

**from market**

**where date\_end-date\_start>5**

// вывод просроченных продуктов и добавление их в existed product и удаление с маркета;

4) **insert into perevod**

**values(1,2,50,'11/12/2022');**

/////////////////////////////

**create or replace TRIGGER perevod\_trigger**

**AFTER INSERT**

**ON perevod**

**FOR EACH ROW**

**declare**

**pragma autonomous\_transaction;**

**BEGIN**

**update clients**

**set balance = (case when id\_client =:new.ID\_PEREVEL then balance-:new.SUMMA\_PEREVODA**

**when id\_client=:new.ID\_POLUCHIL then balance+:new.SUMMA\_PEREVODA**

**end)**

**WHERE id\_client in (:new.ID\_PEREVEL,:new.ID\_POLUCHIL);**

**commit;**

**END;**

//////////Триггер на перевод.

5)**Drop view prosrochka\_id**

//Дропнуть 4 пункт

6) //выбор клиентом продукта. он отправляет на сервер количество выбранного продукта и его айди. а мы делаем инсерт в баскет.

**insert all**

**into basket("nameProduct,k,price\*k"); прайс и имя берем с маркета с помощью айди**

7) **Select id\_client, sum(price)**

**from basket**

**group by id\_client**

//вывод тотал прайса в баскете каждого клиента

8)Когда пользователь нажимает кнопку оплатить создается payment.И как только прошла оплата

создается ЧЕК И баскет делаю пустым для этого клиента.

При добавлении в чек дата дефаулт сегодняшняя.

**create or replace TRIGGER ClearBasket**

**AFTER INSERT**

**ON CHECK\_OF\_CLIENTS**

**FOR EACH ROW**

**declare**

**pragma autonomous\_transaction;**

**BEGIN**

**DELETE FROM BASKET**

**WHERE basket.id\_client = :new.id\_client;**

**commit;**

**END;**

//чистка баскета.

**insert into CHECK\_OF\_CLIENTS**

**values (2,'NameProduct',6\*268,current\_date,2);**

//proverka

//////////Rauan///////////

🎴

**Select DISTINCT x.name\_company, y.name\_product, z.date\_start**

**from product x, company y, market z**

🎴

**Select date\_start, SUM(price)**

**from market**

**group by date\_start**

**order by date\_start**

🎴

**Select first\_name**

**from clients**

**where id\_client in (**

**Select id\_client**

**from market**

**where price < 400**

**Intersect**

**Select id\_client**

**from clients**

**where balance > 400**

**)**

// покупатели которые могут купить продукт ценой меньше 400

🎴

**Select id\_client, sum(price)**

**from basket**

**group by id\_client**

// вывод суммы общей покупки клиента

🎴

**select id\_client, price**

**from market**

**where price > 300 and price < 570**

**MINUS**

**select id\_client, balance**

**from clients**

**where id\_client > 40**

/////////Rauan End/////////////////

9)

**insert into dolzhniki**

**select client\_check\_sum.id\_client,client\_check\_sum.total\_cl-clients.balance**

**from clients,client\_check\_sum**

**where clients.id\_client=client\_check\_sum.id\_client AND client\_check\_sum.total\_cl>clients.balance**

**create view client\_check\_sum AS**

**Select id\_client, sum(price) as total\_cl**

**from basket**

**group by id\_client**

//вывод у кого не хватает денег и сразу добавить в таблицу должников.

10)

**insert into dohod**

**values('2022',(select sum(price)**

**from CHECK\_OF\_CLIENTS**

**where date\_of\_check like '%2022' )+(select sum(ZADOLZHNOST)**

**from DOLZHNIKI)-(select sum(QUANTITY\*DATE\_START)**

**from market**

**where id\_product in(select \* from EXISTED\_PRODUCT)**

**))**

//подсчет дохода сумма чеков и должников минус потерянные продукты

11) **select \***

**from totals\_perevods**

**where tt>200**

**create view totals\_perevods as**

**select id\_perevel,sum(SUMMA\_PEREVODA) as tt**

**from perevod**

**group by id\_perevel**

//вывод тех кто перевел больше 200 денег

12) **insert into CHECK\_OF\_CLIENTS**

**values (1,'Zaam-Dox',234,current\_date,1)**

//////////////////////////////////

**create or replace trigger update\_quantity\_market**

**AFTER**

**DELETE on basket**

**for each row**

**declare**

**pragma autonomous\_transaction;**

**begin**

**UPDATE market**

**SET quantity = quantity-:old.quantity**

**WHERE :old.id\_product = market.id\_product;**

**commit;**

**end;**

//изменение продуктов в маркете

13)

**select market.id\_product ,name\_product,quantity**

**from market,company**

**where quantity<200 and market.id\_product=company.id\_product**

//вывод продуктов которые кончаются и отсюда эти айди взять и закинуть в таблицу для запроса продуктов у компании и можно добавить триггер чтобы автоматически добавлялся продукт в маркете.

**insert into zapros\_na\_p**

**select market.id\_product ,quantity+100**

**from market,company**

**where quantity<200 and market.id\_product=company.id\_product**

////////

14)

**insert into sales**

**values(5,50)**

**create or replace TRIGGER to\_sales**

**AFTER INSERT**

**ON sales**

**FOR EACH ROW**

**declare**

**pragma autonomous\_transaction;**

**BEGIN**

**UPDATE market**

**SET price = price\*(:new.procent/100)**

**WHERE market.id\_product = :new.id\_product;**

**commit;**

**END;**

///сделать скидку на товар тригер

15) **delete from sales**

**where sales.id\_product=5**

**create or replace trigger back\_to\_old\_price**

**AFTER DELETE on sales**

**for each row**

**declare**

**pragma autonomous\_transaction;**

**begin**

**UPDATE market**

**SET price = (price\*100)/(100-:old.procent)**

**WHERE market.id\_product = :old.id\_product;**

**commit;**

**end;**

//убрать скидку на товар