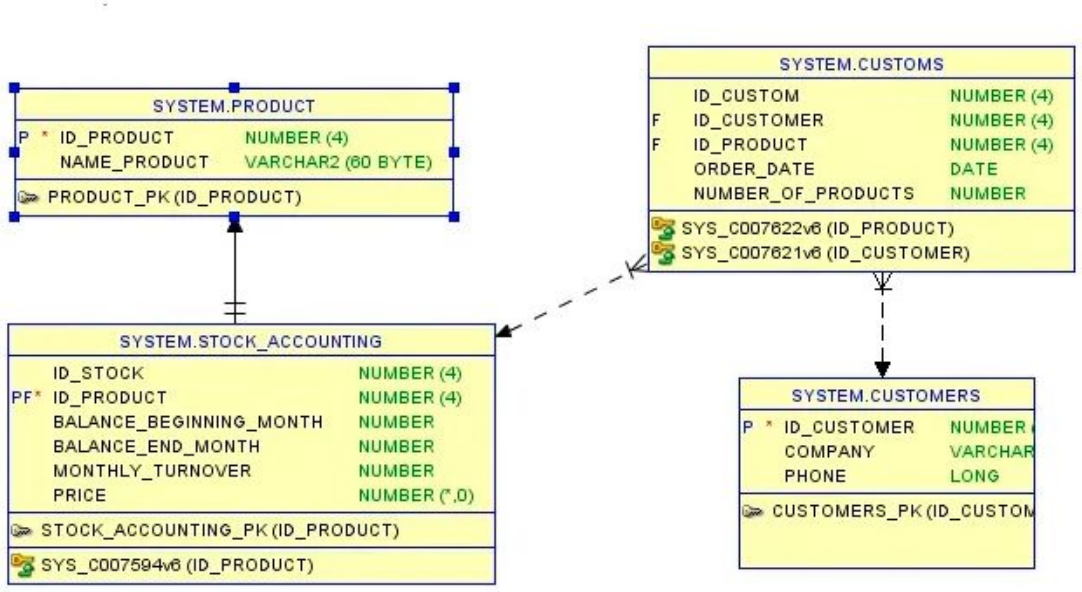
Сначала нужно описать как создавал таблицы, по какому принципу отталкивался, важно дать ей понять, что поля в таблицах были созданы в перспективе возможности выполнения лабы второй - это на тот случай, если скажет, что в реале еще кучу всякой хероты добавляют.



Если смотреть на модель данных в таблицах видно, что в таблице “Customs” нет первичных ключей, что значит, что предполагается возможность возникновения одинаковый записей, это даже логически обусловлено.
Разберу составные поля каждой таблицы(какой в них смысл и тп):

Product:
Ясно, что в данной таблице находятся записи имен продуктов, каждое имя может быть внесено только один раз. Если по простому, то это просто список всех продуктов, которые были зафиксированы в том или ином складе этой базы данных.

Customers:
Тут тоже ясно, что заказчиков с одинаковым именем быть не может, поэтому у каждой компании свой первичный ключ, который связывает название компании, оформившая заказ и номер для связи.

Customs:
Тут под id_custom подразумевается количество заказанных товаров на складе одной компании, так сделанно для того, чтобы при выводе заказанных компанией товаров можно было сразу видеть все ли выборка вывела или нет.
Другие id для всязи с другой таблицей.
По остальным полям думаю понятно.

Stock_accouting:
Тут мне кажется не нужно объяснить.

Вторая лаба:

1)

SELECT name_product, company FROM PRODUCT, CUSTOMERS, CUSTOMS

//из каких таблиц выборка

WHERE PRODUCT.id_product = (SELECT id_product FROM PRODUCT WHERE NAME_PRODUCT = 'Краска белила цинковые')

//конкретная выборка из списка продуктов

AND customs.id_product = product.id_product

AND customers.id_customer = customs.id_customer

//нужно для связи выборки в таблицах, чтобы в соответствии с найденным айдишником в продакт нашлись и остальные данные из других таблиц, по такому же принципу и остальные приравнения айдишников работают

2)

SELECT SUM(BALANCE_BEGINNING_MONTH), SUM(BALANCE_END_MONTH), SUM(MONTHLY_TURNOVER)

FROM STOCK_ACCOUNTING;

// тут довольно просто, функция по заданию, некоторые моменты думаю очевидны да и задания в целом

5)

SELECT company, phone, name_product, price FROM CUSTOMERS, PRODUCT, CUSTOMS, STOCK_ACCOUNTING

WHERE Product.id_product = (SELECT id_product FROM STOCK_ACCOUNTING WHERE price = (SELECT min(price) from STOCK_ACCOUNTING))

AND STOCK_ACCOUNTING.id_product = Product.id_product

AND CUSTOMS.id_product = Product.id_product

AND CUSTOMERS.id_customer = Customs.id_customer

AND CUSTOMS.id_custom < 2;

Вот данное решение может быть не очевидным, но все было предусмотрено еще при создании таблиц и поэтому самое сложное было только получить айдишник в складском учете соответствующий минимальной цене, то есть одновременные выборки из разных таблиц

Хитрость в том, что значение айдишника в таблице заказов удовлетворяет всем критериям в задании, заказ у заказчика только один и он минимальный - подходит, нет - выборка не выполняется

6) UPDATE "STOCK_ACCOUNTING"

SET "ID_STOCK" = 4 WHERE "ID_STOCK" = 1;

просто в таблице складского учета меняем айдишник на нужный
даже такое учла (легко просто гениально)

Третья:

2)

```
DECLARE

v_name_prod product.name_product%TYPE;

v_balance_end STOCK_ACCOUNTING.BALANCE_END_MONTH%TYPE;

//определение нужных переменных тип которых может быть изменен(%type)

CURSOR cur_company (num_balance int)

//объявление курсора с переменной(предыдущее задание такое же но без, поэтому незачем объяснять)

IS

SELECT name_product, BALANCE_END_MONTH

FROM STOCK_ACCOUNTING, PRODUCT

WHERE STOCK_ACCOUNTING.BALANCE_END_MONTH < num_balance

AND stock_accounting.id_product = product.id_product;

//внутренняя выборка

BEGIN

OPEN cur_company(100);

LOOP

FETCH cur_company INTO v_name_prod, v_balance_end;

//выборка данных из курсора с помощью оператора fetch

EXIT WHEN cur_company%NOTFOUND;

DBMS_OUTPUT.put_line('RESULT: ' ||v_name_prod||' '|| v_balance_end);
```

//вот тут курсор выводит результат выборки

END LOOP;

CLOSE cur_company;

END;

3)

CREATE TABLE "AUDIT"

("USER_NAME" VARCHAR2(50), "TIME" DATE);

//создание таблицы в которую будем записывать данные когда тригер сработал и кто виноват

CREATE OR REPLACE

//если существует с таким названием, заменить

TRIGGER ADT

AFTER INSERT OR DELETE OR UPDATE

//объявление тригера, который сработает после того, когда:
внесут или удалят или изменят в таблице

ON STOCK_ACCOUNTING

//в какой таблице

FOR EACH ROW

DECLARE

BEGIN

INSERT INTO "AUDIT"("USER_NAME",

"TIME") VALUES (USER, SYSDATE);

//что делает триггер, после того, как сработает, а именно внесет данные в нашу ранее созданную таблицу

END ADT;

UPDATE "STOCK_ACCOUNTING"

SET "ID_STOCK" = 1 WHERE "ID_STOCK" = 4;

//для проверки

4)

```
CREATE SEQUENCE AVTO START WITH 20 INCREMENT BY 1;

//создание последовательности, у нас в таблице продакт было 20 продуктов, поэтому начинаем с 20

CREATE OR REPLACE

TRIGGER IDENT

BEFORE INSERT ON PRODUCT

FOR EACH ROW

DECLARE

BEGIN

SELECT AVTO.NEXTVAL

INTO :NEW.id_product

//создание новой записи с айдишником из созданной последдовательности

FROM DUAL ;

END IDENT;
```