# Оптовая реализация бытовой техники

# Создание и заполнение базы данных в СУБД Oracle

CREATE TABLE dsuppliers(dno INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

legal\_name\_Supplier\_company VARCHAR2(70) NOT NULL,

tel\_no VARCHAR2(17),

email\_address VARCHAR2(70) NOT NULL,

Date\_last\_delivery DATE NOT NULL,

CONSTRAINT ch\_telno\_dsuppliers

CHECK(REGEXP\_LIKE(tel\_no, '^\+375\(\d{2}\)\d{3}-\d{2}-\d{2}$')),

CONSTRAINT uni\_telno\_dsuppliers UNIQUE(tel\_no));

Изображение выглядит как текст, монитор, внутренний, снимок экрана

Автоматически созданное описание

CREATE TABLE iproducts(ino INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

product\_name VARCHAR2(60) NOT NULL,

quantity VARCHAR2(60) NOT NULL,

guarante VARCHAR2(60) NOT NULL,

dno INTEGER,

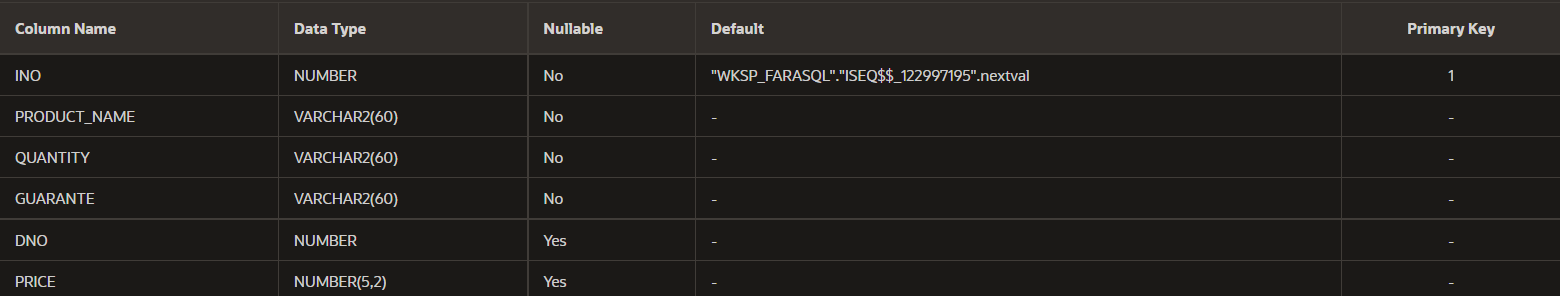
price NUMBER(5,2),

CONSTRAINT dno\_fk\_iproducts

FOREIGN KEY(dno)

REFERENCES dsuppliers

);



CREATE TABLE сshopping\_cart( ino INTEGER,

lno INTEGER,

quantity VARCHAR2(20) NOT NULL,

PRIMARY KEY(ino,lno),

CONSTRAINT ino\_fk\_сshopping\_cart

FOREIGN KEY(ino)

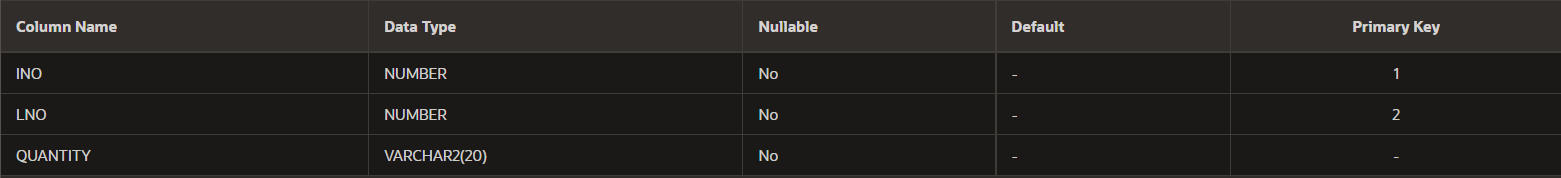
REFERENCES iproducts,

CONSTRAINT lno\_fk\_сshopping\_cart

FOREIGN KEY(lno)

REFERENCES lorders

);



CREATE TABLE lorders(lno INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

zno INTEGER,

mno INTEGER,

order\_date DATE NOT NULL,

CONSTRAINT zno\_fk\_lorders

FOREIGN KEY(zno)

REFERENCES zclient,

CONSTRAINT mno\_fk\_lorders

FOREIGN KEY(mno)

REFERENCES mdealer

);

Изображение выглядит как текст

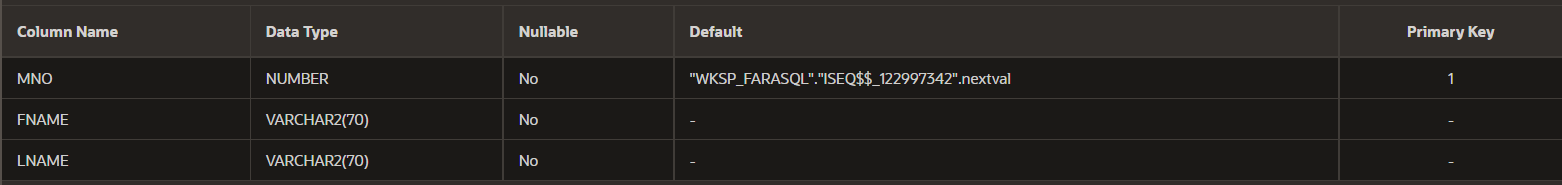
Автоматически созданное описание

CREATE TABLE mdealer(mno INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

fname VARCHAR2(70) NOT NULL,

lname VARCHAR2(70) NOT NULL

);



CREATE TABLE zclient(zno INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

legal\_name\_client\_company VARCHAR2(60) NOT NULL,

tel\_no VARCHAR2(17),

email\_address VARCHAR2(60) NOT NULL,

date\_of\_last\_order DATE NOT NULL,

CONSTRAINT ch\_telno\_zclient

CHECK(REGEXP\_LIKE(tel\_no, '^\+375\(\d{2}\)\d{3}-\d{2}-\d{2}$')),

CONSTRAINT uni\_telno\_zclient

UNIQUE(tel\_no)

);

Изображение выглядит как текст, снимок экрана, внутренний, черный

Автоматически созданное описание

CREATE TABLE fdirector(fno INTEGER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

zno INTEGER,

fname VARCHAR2(60) NOT NULL,

lname VARCHAR2(60) NOT NULL,

tel\_no VARCHAR2(17) NOT NULL,

email\_address VARCHAR2(65) NOT NULL,

dob DATE,

CONSTRAINT ch\_telno\_fdirector

CHECK(REGEXP\_LIKE(tel\_no, '^\+375\(\d{2}\)\d{3}-\d{2}-\d{2}$')),

CONSTRAINT uni\_telno\_fdirector

UNIQUE(tel\_no),

CONSTRAINT zno\_fk\_fdirector

FOREIGN KEY(zno)

REFERENCES zclient

);

Изображение выглядит как текст, снимок экрана, внутренний

Автоматически созданное описание

Создание последовательности. Использование последовательностей полезно для автоматической генерации уникальных первичных ключей для данных, а также для координирования ключей между различными строками или таблицами.

Без генератора последовательностей порядковые номера можно создавать лишь программным способом.

CREATE SEQUENCE dsuppliers\_seq

START WITH 1

INCREMENT BY 1

NOCACHE;

CREATE SEQUENCE iproducts\_seq

START WITH 1

INCREMENT BY 1

NOCACHE;

CREATE SEQUENCE сshopping\_cart\_seq

START WITH 1

INCREMENT BY 1

NOCACHE;

CREATE SEQUENCE lorders\_seq

START WITH 1

INCREMENT BY 1

NOCACHE;

CREATE SEQUENCE mdealer\_seq

START WITH 1

INCREMENT BY 1

NOCACHE;

CREATE SEQUENCE zclient\_seq

START WITH 1

INCREMENT BY 1

NOCACHE;

CREATE SEQUENCE fdirector\_seq

START WITH 1

INCREMENT BY 1

NOCACHE;

Вставка строк в таблицу осуществляется с помощью DML-инструкции INSERT

begin

insert into dsuppliers(legal\_name\_Supplier\_company, tel\_no, email\_address, Date\_last\_delivery) values('Железо','+375(29)111-11-61','Железо@mailru','10.05.2021');

insert into dsuppliers(legal\_name\_Supplier\_company, tel\_no, email\_address, Date\_last\_delivery) values('Гарант','+375(44)225-10-72','Гарант@mailru','08.09.2021');

insert into dsuppliers(legal\_name\_Supplier\_company, tel\_no, email\_address, Date\_last\_delivery) values('Алмаз','+375(44)223-81-71','Алмаз@mailru','09.07.2021');

insert into dsuppliers(legal\_name\_Supplier\_company, tel\_no, email\_address, Date\_last\_delivery) values('Такса','+375(29)345-11-61','Такса@mailru','07.03.2021');

insert into dsuppliers(legal\_name\_Supplier\_company, tel\_no, email\_address, Date\_last\_delivery) values('Гепард','+375(44)667-45-51','Гепард@mailru','01.11.2020');

insert into dsuppliers(legal\_name\_Supplier\_company, tel\_no, email\_address, Date\_last\_delivery) values('Ветер','+375(29)678-01-71','Ветер@mailru','03.01.2019');

end;

/

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание

begin

insert into iproducts(product\_name, quantity, guarante, dno, price) values('Холодильник','15','5лет',1,'240');

insert into iproducts(product\_name, quantity, guarante, dno, price) values('Пылесос','9','2года',2,'200');

insert into iproducts(product\_name, quantity, guarante, dno, price) values('Стиральная машина','19','7лет',3,'70');

insert into iproducts(product\_name, quantity, guarante, dno, price) values('Утюг','10','2года',4,'50');

insert into iproducts(product\_name, quantity, guarante, dno, price) values('Чайник','20','7лет',5,'134');

insert into iproducts(product\_name, quantity, guarante, dno, price) values('Рэдмонд','10','5лет',6,'321');

end;

/

Изображение выглядит как текст, монитор, снимок экрана

Автоматически созданное описание

DELETE FROM dsuppliers (удаление данных с таблиц)

ALTER TABLE iproducts

DROP COLUMN price; (удаление столбца в созд таблицу)

ALTER TABLE iproducts

ADD price NUMBER(5,2) NULL; (добавления столбца в созд таблицу)

begin

insert into mdealer(fname, lname) values('Сангалиев','Ернур');

insert into mdealer(fname, lname) values('Рахатулы','Алишер');

insert into mdealer(fname, lname) values('Аугамбаев','Рустем');

insert into mdealer(fname, lname) values('Игоревич','Алексей');

insert into mdealer(fname, lname) values('Аргинова','Айнур');

insert into mdealer(fname, lname) values('Шафихов','Шапагат');

end;

/

Изображение выглядит как текст

Автоматически созданное описание

begin

insert into lorders(mno, order\_date) values(1,'12.02.2020');

insert into lorders(mno, order\_date) values(2,'10.05.2019');

insert into lorders(mno, order\_date) values(3,'08.09.2021');

insert into lorders(mno, order\_date) values(4,'05.07.2017');

insert into lorders(mno, order\_date) values(5,'11.10.2016');

end;

/

Изображение выглядит как текст, внутренний, снимок экрана

Автоматически созданное описание

begin

insert into сshopping\_cart(ino, lno, quantity) values(21,1,'3');

insert into сshopping\_cart(ino, lno, quantity) values(22,2,'4');

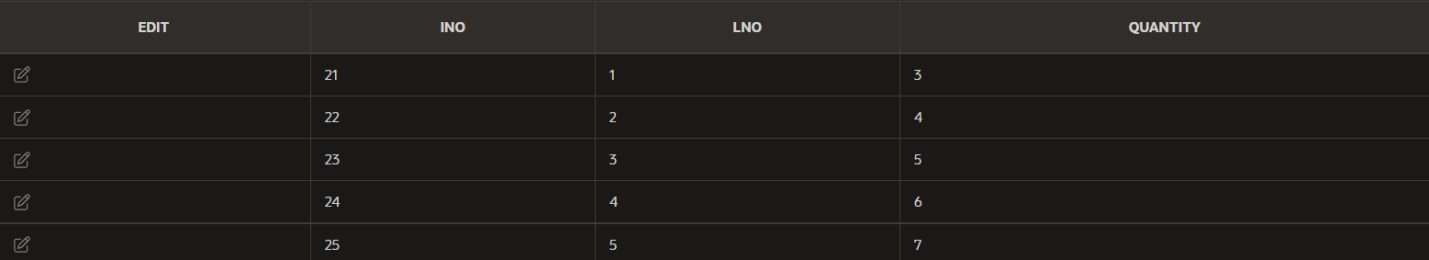
insert into сshopping\_cart(ino, lno, quantity) values(23,3,'5');

insert into сshopping\_cart(ino, lno, quantity) values(24,4,'6');

insert into сshopping\_cart(ino, lno, quantity) values(25,5,'7');

end;

/



begin

insert into zclient(legal\_name\_client\_company, tel\_no, email\_address, date\_of\_last\_order) values('Мечта','+375(21)222-12-15','Мечта@mailru','12.15.2021');

insert into zclient(legal\_name\_client\_company, tel\_no, email\_address, date\_of\_last\_order) values('Технодом','+375(21)333-99-27','Технодом@mailru','07.22.2021');

insert into zclient(legal\_name\_client\_company, tel\_no, email\_address, date\_of\_last\_order) values('Дом быта','+375(21)443-19-58','Дом быта@mailru','06.16.2021');

insert into zclient(legal\_name\_client\_company, tel\_no, email\_address, date\_of\_last\_order) values('Все для дома','+375(21)938-37-09','Все для дома@mailru','08.15.2021');

insert into zclient(legal\_name\_client\_company, tel\_no, email\_address, date\_of\_last\_order) values('Техком','+375(21)159-02-49','Техком@mailru','10.18.2021');

insert into zclient(legal\_name\_client\_company, tel\_no, email\_address, date\_of\_last\_order) values('Фора кз','+375(21)158-10-36','Фора кз@mailru','01.17.2021');

end;

/

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание

(Тут изменил zno с - на числа по порядку 1 до 6)

UPDATE lorders (назв таблицы)

SET zno = 6 (какой столбец и на что хочешь поменять)

WHERE lno = 6 (указываешь какое поле ты выбираешь через id-ишку(primary key),в моем случае это lno)

begin

insert into fdirector(zno, fname, lname, tel\_no, email\_address, dob) values(1,'Аязбаев','Данияр','+375(25)105-02-27','Аязбаев@mailru','12.12.2000');

insert into fdirector(zno, fname, lname, tel\_no, email\_address, dob) values(2,'Сейтбенбетов','Жанболат','+375(25)014-74-25','Сейтбенбетов@mailru','03.15.2000');

insert into fdirector(zno, fname, lname, tel\_no, email\_address, dob) values(3,'Дюсеков','Абылай','+375(25)062-03-18','Дюсеков@mailru','11.26.2000');

insert into fdirector(zno, fname, lname, tel\_no, email\_address, dob) values(4,'Балгожин','Алдияр','+375(25)036-06-15','Балгожин@mailru','01.05.2001');

insert into fdirector(zno, fname, lname, tel\_no, email\_address, dob) values(5,'Камбарбай','Бекзат','+375(25)194-28-29','Камбарбай@mailru','09.14.2002');

insert into fdirector(zno, fname, lname, tel\_no, email\_address, dob) values(6,'Муратбеков','Динмухамед','+375(71)328-36-96','Муратбеков@mailru','03.19.1999');

end;

/

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание

Данные могут отличаться, так как, проводились работы с триггерами.

# Создание запросов

«Крупнейшие партии» (условная выборка)

SELECT Z.legal\_name\_client\_company, I.product\_name, C.quantity

FROM zclient Z INNER JOIN (lorders L INNER JOIN (сshopping\_cart C INNER JOIN iproducts I ON C.ino = I.ino) ON C.lno = L.lno) ON Z.zno = L.zno

WHERE C.quantity>6;

Изображение выглядит как текст, монитор, экран, черный

Автоматически созданное описание

«Сводка по технике, закупленной каждым клиентом» (итоговый запрос)

Select Z.legal\_name\_client\_company, sum(Distinct C.quantity) as Total\_teh

From zclient Z Inner Join (lorders L Inner Join сshopping\_cart C On l.lno = C.lno) On Z.zno = l.zno

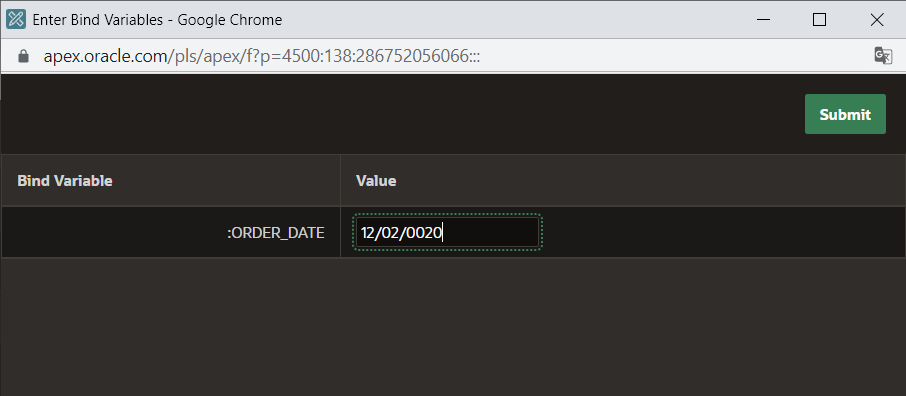
Group by Z.legal\_name\_client\_company;

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание

«Продукция, реализованная в заданный период времени» (параметрический запрос)

SELECT \* FROM lorders WHERE order\_date=:order\_date;



Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание

«Общий список клиентов и производителей с количеством продаж» (запрос на объединение)

Select I.quantity As Col\_prodazh

From dsuppliers D Inner Join iproducts I on D.dno=I.dno

Union

select Z.legal\_name\_client\_company

from zclient Z;

Изображение выглядит как текст, монитор, снимок экрана, внутренний

Автоматически созданное описание

«Количество проданных товаров фирм-производителей по кварталам» (запрос по полю с типом дата)

select d.LEGAL\_NAME\_SUPPLIER\_COMPANY, to\_char(l.order\_date, 'Q'), SUM(c.quantity)

FROM lorders L INNER JOIN (сshopping\_cart C INNER JOIN (iproducts I Inner join dsuppliers D On I.dno = D.dno) On C.ino = I.ino) on C.lno = L.lno

GROUP BY d.LEGAL\_NAME\_SUPPLIER\_COMPANY,to\_char(l.order\_date, 'Q')

Изображение выглядит как текст, монитор, экран, внутренний

Автоматически созданное описание

С внешним соединением таблиц, используя FULL JOIN, LEFT JOIN или RIGHT JOIN, при этом обязательным является наличие в БД данных, которые будут выводиться именно с выбранным оператором внешнего соединения;

Select C.lno, C.quantity, I.ino, I.product\_name

From сshopping\_cart C

Right join iproducts I On C.ino = I.ino

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание

С использованием предиката IN с подзапросом;

SELECT Z.zno, Z.legal\_name\_client\_company, Z.tel\_no, Z.email\_address, Z.date\_of\_last\_order

FROM zclient Z

WHERE Z.zno IN (SELECT F.zno FROM fdirector F WHERE lname= 'Абылай');

Изображение выглядит как текст, монитор, экран, снимок экрана

Автоматически созданное описание

С использованием предиката ANY/ALL с подзапросом;

SELECT I.product\_name, C.quantity

FROM iproducts I, сshopping\_cart C

WHERE C.quantity>= ALL(Select quantity FROM сshopping\_cart) AND I.ino = C.ino

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание

С использованием предиката EXISTS/NOT EXISTS с подзапросом.

SELECT F.fname, F.lname

FROM fdirector F

WHERE NOT EXISTS (SELECT 1 FROM zclient Z

WHERE Z.zno = F.zno);

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание

# Создание и использование представлений

CREATE OR REPLACE VIEW price\_more200 AS

SELECT \*FROM iproducts WHERE price>=200

WITH CHECK OPTION;

insert into price\_more200(product\_name, quantity, guarante, dno, price) values('Яндекс','25','1год',20,'180'); -

Изображение выглядит как текст

Автоматически созданное описание

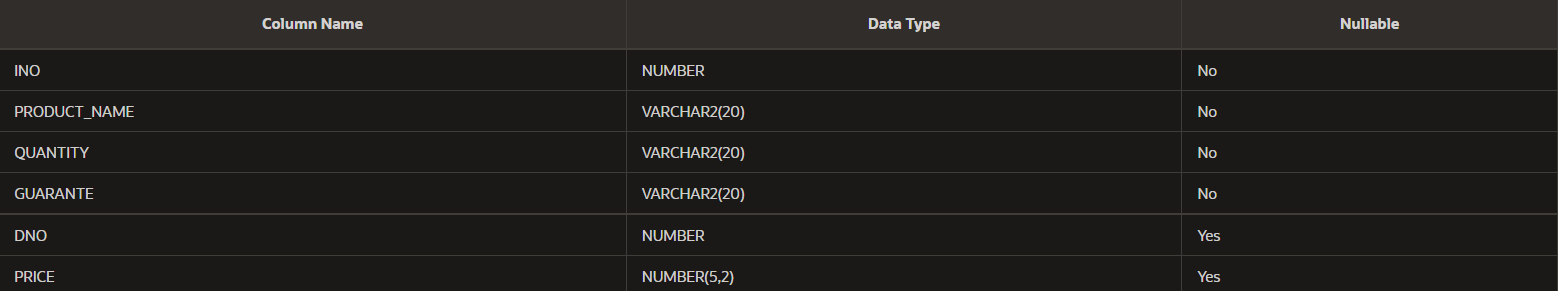
Изображение выглядит как текст, экран, отображаемый

Автоматически созданное описание

CREATE OR REPLACE VIEW price\_200 AS

SELECT \*FROM iproducts WHERE price>=200

insert into price\_200(product\_name, quantity, guarante, dno, price) values('Алиса','30','1год',20,'190'); +



Изображение выглядит как текст, экран

Автоматически созданное описание

Создаю вертикальное или смешанное не обновляемое представление, также доказываю запросом.

CREATE OR REPLACE VIEW info\_product AS

SELECT product\_name as products, C.quantity, I.price FROM сshopping\_cart C

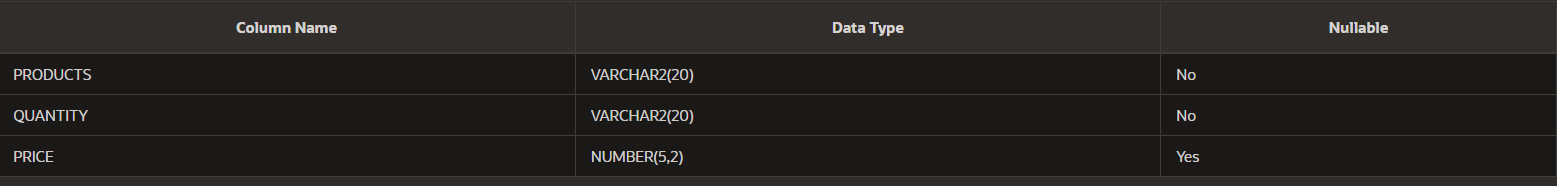
Inner join iproducts I On C.ino = I.ino

UPDATE info\_product SET products='Казан' where quantity=7 -

DELETE from info\_product where products='Утюг' +

Изображение выглядит как стол

Автоматически созданное описание



Создаю обновляемое представление для работы с одной из родительских таблиц индивидуальной БД и через него разрешить работу с данными только в рабочие дни (с понедельника по пятницу) и в рабочие часы (с 9:00 до 17:00).

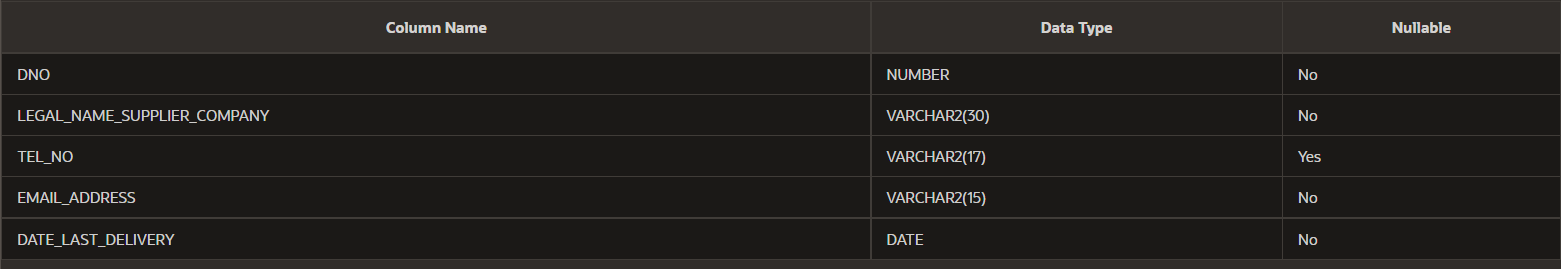
CREATE OR REPLACE VIEW v\_dsuppliers AS

Select\*From dsuppliers where (select to\_char(sysdate,'d') from dual) IN ('2','3','4','5','6') AND (select EXTRACT(hour FROM systimestamp) from dual)+3>=9

AND (select EXTRACT(hour FROM systimestamp) from dual)+3<=16

WITH CHECK OPTION;

UPDATE v\_dsuppliers SET legal\_name\_Supplier\_company='Камень' where dno=22



# 

# процедурЫ, функциИ и пакетЫ PL/SQL

Написал процедуру, которая изменяет домашний адрес клиента по указанной в качестве параметра фамилии. В случае обнаружения однофамильцев выводить их количество и уточняющие данные.

create or replace PROCEDURE change\_emaddress (client\_name IN VARCHAR2) IS

CURSOR cur (client\_name2 IN VARCHAR2)

IS

(SELECT legal\_name\_client\_company, tel\_no, email\_address FROM zclient

WHERE legal\_name\_client\_company = client\_name2)

FOR UPDATE;

cursor\_row cur%ROWTYPE;

num NUMBER;

BEGIN

SELECT COUNT(\*) INTO num FROM zclient

WHERE legal\_name\_client\_company = client\_name;

IF(num = 0) THEN

DBMS\_OUTPUT.put\_line ('Нет клиентов с таким именем (' || num || ') :');

END IF;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Несколько записей было изменено (' || num || ') :');

END IF;

OPEN cur(client\_name);

FETCH cur INTO cursor\_row;

WHILE cur%FOUND

LOOP

UPDATE zclient

SET email\_address = 'new\_email@google.com'

WHERE CURRENT OF cur;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Номер: ' || cursor\_row.tel\_no || ', Предыдущий почтовый адрес: ' || cursor\_row.email\_address);

END IF;

FETCH cur INTO cursor\_row;

END LOOP;

CLOSE cur;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Нет клиентов с таким именем');

END;

Вывод

begin

change\_emaddress('Мечта');

end;

/

Создал функцию, подсчитывающую общее количество реализованной продукции заданного производителя. Если какой-либо продукции реализовано менее 1 единицы в год, то удалить информацию об этой продукции во вспомогательную таблицу.

create or replace FUNCTION totals\_prod(legal\_name\_Supplier\_company2 IN VARCHAR2)

RETURN NUMBER AS

pragma autonomous\_transaction;

err\_data EXCEPTION;

CURSOR pr\_cur (legal\_name\_Supplier\_company3 IN VARCHAR2)

IS

SELECT ds.legal\_name\_Supplier\_company as name, sum(cc.quantity) as sm

FROM dsuppliers ds join iproducts ip

on ip.dno = ds.dno

join сshopping\_cart cc

on cc.ino = ip.ino

where ds.legal\_name\_Supplier\_company = legal\_name\_Supplier\_company3

GROUP BY ds.legal\_name\_Supplier\_company;

pr\_cursor\_row pr\_cur%ROWTYPE;

CURSOR it\_cur (legal\_name\_Supplier\_company4 IN VARCHAR2)

IS

SELECT ip.product\_name as name, sum(cc.quantity) as sm, min(lo.order\_date) as mindate

FROM dsuppliers ds join iproducts ip

on ip.dno = ds.dno

join сshopping\_cart cc

on cc.ino = ip.ino

join lorders lo

on lo.lno = cc.lno

where ds.legal\_name\_Supplier\_company = legal\_name\_Supplier\_company4

GROUP BY ip.product\_name;

it\_cursor\_row it\_cur%ROWTYPE;

BEGIN

OPEN pr\_cur(legal\_name\_Supplier\_company2);

FETCH pr\_cur INTO pr\_cursor\_row;

IF (pr\_cur%NOTFOUND) THEN

RAISE err\_data;

END IF;

OPEN it\_cur(legal\_name\_Supplier\_company2);

FETCH it\_cur INTO it\_cursor\_row;

WHILE it\_cur%FOUND

LOOP

IF ( ( it\_cursor\_row.sm / (EXTRACT( YEAR FROM SYSDATE ) - EXTRACT( YEAR FROM it\_cursor\_row.mindate ) + 1) ) < 1) THEN

DELETE FROM сshopping\_cart cc where cc.ino = it\_cursor\_row.name;

DELETE FROM lorders lo WHERE lo.lno = (select lno from сshopping\_cart cc where cc.ino = it\_cursor\_row.name);

DELETE FROM iproducts ip where ip.ino = it\_cursor\_row.name;

COMMIT;

END IF;

FETCH it\_cur INTO it\_cursor\_row;

END LOOP;

CLOSE it\_cur;

RETURN pr\_cursor\_row.sm;

EXCEPTION

WHEN err\_data THEN

DBMS\_OUTPUT.PUT\_LINE('Информация не найдена');

RETURN 0;

WHEN OTHERS THEN

raise\_application\_error(-2001,'Ошибка');

RETURN 0;

END;

Вывод

DECLARE

cntr NUMBER := 0;

BEGIN

cntr := totals\_prod('Гарант');

DBMS\_OUTPUT.PUT\_LINE('Реализовано продукции: ' || cntr);

END;

Написал перегруженные программы, используя для этого ранее созданную процедуру или функцию.

create or replace PROCEDURE change\_emaddress(client\_name IN VARCHAR2, new\_address IN VARCHAR2) IS

CURSOR cur (client\_name2 IN VARCHAR2)

IS

(SELECT legal\_name\_client\_company, tel\_no, email\_address FROM zclient

WHERE legal\_name\_client\_company = client\_name2)

FOR UPDATE;

cursor\_row cur%ROWTYPE;

num NUMBER;

BEGIN

SELECT COUNT(\*) INTO num FROM zclient

WHERE legal\_name\_client\_company = client\_name;

IF(num = 0) THEN

DBMS\_OUTPUT.put\_line ('Нет клиентов с таким именем (' || num || ') :');

END IF;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Несколько записей было изменено (' || num || ') :');

END IF;

OPEN cur(client\_name);

FETCH cur INTO cursor\_row;

WHILE cur%FOUND

LOOP

UPDATE zclient

SET email\_address = 'new\_address'

WHERE CURRENT OF cur;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Номер: ' || cursor\_row.tel\_no || ', Предыдущий почтовый адрес: ' || cursor\_row.email\_address);

END IF;

FETCH cur INTO cursor\_row;

END LOOP;

CLOSE cur;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Нет клиентов с таким именем');

END;

Создал локальную программу, изменив код ранее написанной процедуры или функции.

create or replace PROCEDURE change\_emaddressl (client\_name IN VARCHAR2) IS

CURSOR cur (client\_name2 IN VARCHAR2)

IS

(SELECT legal\_name\_client\_company, tel\_no, email\_address FROM zclient

WHERE legal\_name\_client\_company = client\_name2)

FOR UPDATE;

cursor\_row cur%ROWTYPE;

num NUMBER;

FUNCTION local\_get\_count(c\_name IN VARCHAR2) RETURN NUMBER IS

cnt NUMBER;

BEGIN

SELECT count(\*)

INTO cnt

FROM zclient

WHERE legal\_name\_client\_company = c\_name;

RETURN cnt;

END local\_get\_count;

BEGIN

num := local\_get\_count(client\_name);

IF(num = 0) THEN

DBMS\_OUTPUT.put\_line ('Нет клиентов с таким именем (' || num || ') :');

END IF;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Несколько записей было изменено (' || num || ') :');

END IF;

OPEN cur(client\_name);

FETCH cur INTO cursor\_row;

WHILE cur%FOUND

LOOP

UPDATE zclient

SET email\_address = 'nSome\_value'

WHERE CURRENT OF cur;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Номер: ' || cursor\_row.tel\_no || ', Предыдущий почтовый адрес: ' || cursor\_row.email\_address);

END IF;

FETCH cur INTO cursor\_row;

END LOOP;

CLOSE cur;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Нет клиентов с таким именем');

END;

Объединил все процедуры и функции, в том числе перегруженные, в пакет.

CREATE OR REPLACE PACKAGE package5 IS

PROCEDURE change\_emaddress(client\_name IN VARCHAR2);

PROCEDURE change\_emaddress(client\_name IN VARCHAR2, new\_address IN VARCHAR2);

PROCEDURE change\_emaddressl(client\_name IN VARCHAR2);

FUNCTION totals\_prod(legal\_name\_Supplier\_company2 IN VARCHAR2) RETURN NUMBER;

END package5;

CREATE OR REPLACE PACKAGE BODY package5 IS

PROCEDURE change\_emaddress(client\_name IN VARCHAR2) IS

CURSOR cur (client\_name2 IN VARCHAR2)

IS

(SELECT legal\_name\_client\_company, tel\_no, email\_address FROM zclient

WHERE legal\_name\_client\_company = client\_name2)

FOR UPDATE;

cursor\_row cur%ROWTYPE;

num NUMBER;

BEGIN

SELECT COUNT(\*) INTO num FROM zclient

WHERE legal\_name\_client\_company = client\_name;

IF(num = 0) THEN

DBMS\_OUTPUT.put\_line ('Нет клиентов с таким именем (' || num || ') :');

END IF;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Несколько записей было изменено (' || num || ') :');

END IF;

OPEN cur(client\_name);

FETCH cur INTO cursor\_row;

WHILE cur%FOUND

LOOP

UPDATE zclient

SET email\_address = 'new\_email@google.com'

WHERE CURRENT OF cur;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Номер: ' || cursor\_row.tel\_no || ', Предыдущий почтовый адрес: ' || cursor\_row.email\_address);

END IF;

FETCH cur INTO cursor\_row;

END LOOP;

CLOSE cur;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Нет клиентов с таким именем');

END;

PROCEDURE change\_emaddress (client\_name IN VARCHAR2, new\_address IN VARCHAR2) IS

CURSOR cur (client\_name2 IN VARCHAR2)

IS

(SELECT legal\_name\_client\_company, tel\_no, email\_address FROM zclient

WHERE legal\_name\_client\_company = client\_name2)

FOR UPDATE;

cursor\_row cur%ROWTYPE;

num NUMBER;

BEGIN

SELECT COUNT(\*) INTO num FROM zclient

WHERE legal\_name\_client\_company = client\_name;

IF(num = 0) THEN

DBMS\_OUTPUT.put\_line ('Нет клиентов с таким именем (' || num || ') :');

END IF;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Несколько записей было изменено (' || num || ') :');

END IF;

OPEN cur(client\_name);

FETCH cur INTO cursor\_row;

WHILE cur%FOUND

LOOP

UPDATE zclient

SET email\_address = 'new\_address'

WHERE CURRENT OF cur;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Номер: ' || cursor\_row.tel\_no || ', Предыдущий почтовый адрес: ' || cursor\_row.email\_address);

END IF;

FETCH cur INTO cursor\_row;

END LOOP;

CLOSE cur;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Нет клиентов с таким именем');

END;

PROCEDURE change\_emaddressl(client\_name IN VARCHAR2) IS

CURSOR cur (client\_name2 IN VARCHAR2)

IS

(SELECT legal\_name\_client\_company, tel\_no, email\_address FROM zclient

WHERE legal\_name\_client\_company = client\_name2)

FOR UPDATE;

cursor\_row cur%ROWTYPE;

num NUMBER;

FUNCTION local\_get\_count(c\_name IN VARCHAR2) RETURN NUMBER IS

cnt NUMBER;

BEGIN

SELECT count(\*)

INTO cnt

FROM zclient

WHERE legal\_name\_client\_company = c\_name;

RETURN cnt;

END local\_get\_count;

BEGIN

num := local\_get\_count(client\_name);

IF(num = 0) THEN

DBMS\_OUTPUT.put\_line ('Нет клиентов с таким именем (' || num || ') :');

END IF;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Несколько записей было изменено (' || num || ') :');

END IF;

OPEN cur(client\_name);

FETCH cur INTO cursor\_row;

WHILE cur%FOUND

LOOP

UPDATE zclient

SET email\_address = 'nSome\_value'

WHERE CURRENT OF cur;

IF(num > 1) THEN

DBMS\_OUTPUT.put\_line ('Номер: ' || cursor\_row.tel\_no || ', Предыдущий почтовый адрес: ' || cursor\_row.email\_address);

END IF;

FETCH cur INTO cursor\_row;

END LOOP;

CLOSE cur;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Нет клиентов с таким именем');

END;

FUNCTION totals\_prod(legal\_name\_Supplier\_company2 IN VARCHAR2)

RETURN NUMBER AS

pragma autonomous\_transaction;

err\_data EXCEPTION;

CURSOR pr\_cur (legal\_name\_Supplier\_company3 IN VARCHAR2)

IS

SELECT ds.legal\_name\_Supplier\_company as name, sum(cc.quantity) as sm

FROM dsuppliers ds join iproducts ip

on ip.dno = ds.dno

join сshopping\_cart cc

on cc.ino = ip.ino

where ds.legal\_name\_Supplier\_company = legal\_name\_Supplier\_company3

GROUP BY ds.legal\_name\_Supplier\_company;

pr\_cursor\_row pr\_cur%ROWTYPE;

CURSOR it\_cur (legal\_name\_Supplier\_company4 IN VARCHAR2)

IS

SELECT ip.product\_name as name, sum(cc.quantity) as sm, min(lo.order\_date) as mindate

FROM dsuppliers ds join iproducts ip

on ip.dno = ds.dno

join сshopping\_cart cc

on cc.ino = ip.ino

join lorders lo

on lo.lno = cc.lno

where ds.legal\_name\_Supplier\_company = legal\_name\_Supplier\_company4

GROUP BY ip.product\_name;

it\_cursor\_row it\_cur%ROWTYPE;

BEGIN

OPEN pr\_cur(legal\_name\_Supplier\_company2);

FETCH pr\_cur INTO pr\_cursor\_row;

IF (pr\_cur%NOTFOUND) THEN

RAISE err\_data;

END IF;

OPEN it\_cur(legal\_name\_Supplier\_company2);

FETCH it\_cur INTO it\_cursor\_row;

WHILE it\_cur%FOUND

LOOP

IF ( ( it\_cursor\_row.sm / (EXTRACT( YEAR FROM SYSDATE ) - EXTRACT( YEAR FROM it\_cursor\_row.mindate ) + 1) ) < 1) THEN

DELETE FROM сshopping\_cart cc where cc.ino = it\_cursor\_row.name;

DELETE FROM lorders lo WHERE lo.lno = (select lno from сshopping\_cart cc where cc.ino = it\_cursor\_row.name);

DELETE FROM iproducts ip where ip.ino = it\_cursor\_row.name;

COMMIT;

END IF;

FETCH it\_cur INTO it\_cursor\_row;

END LOOP;

CLOSE it\_cur;

RETURN pr\_cursor\_row.sm;

EXCEPTION

WHEN err\_data THEN

DBMS\_OUTPUT.PUT\_LINE('Информация не найдена');

RETURN 0;

WHEN OTHERS THEN

raise\_application\_error(-2001,'Ошибка');

RETURN 0;

END;

END package5;

Написал анонимный PL/SQL-блок, в котором будут вызовы реализованных функций и процедур пакета с различными характерными значениями параметров для проверки правильности работы основных задач и обработки исключительных ситуаций.

DECLARE

cntr NUMBER := 0;

BEGIN

package5.change\_emaddress('Мечта');

package5.change\_emaddress('Техком','Фара1');

package5.change\_emaddressl('Дом быта');

cntr := package5.totals\_prod('Гарант');

DBMS\_OUTPUT.PUT\_LINE('Реализовано деталей: ' || cntr);

END;

Изображение выглядит как текст

Автоматически созданное описание

# Триггеры

Написал DML-триггер, регистрирующий изменение данных (вставку, обновление, удаление) в одной из таблиц БД. Во вспомогательную таблицу LOG1 записывать, кто, когда (дата и время) и какое именно изменение произвел, для одного из столбцов сохранять старые и новые значения.

Создание таблицы LOG1

create table LOG1(

operation\_name char(1),

pk varchar2(20),

column\_name varchar2(20),

old\_value varchar2(20),

new\_value varchar2(20),

username varchar2(20),

op\_date date

)

процедура, заносящая изменения в таблицу acc\_changes

create or replace procedure logging(

operation\_name in char,

pk in varchar2,

column\_name in varchar2,

old\_value in varchar2,

new\_value in varchar2

) is

pragma autonomous\_transaction;

begin

if old\_value <> new\_value or operation\_name in ('I', 'D') then

insert into LOG1 values (

operation\_name, pk, column\_name, old\_value, new\_value, USER, SYSDATE);

commit;

end if;

end;

Триггер следящий за изменениями

create or replace trigger clients\_log

after insert or update or delete

on zclient

for each row --тригер уровня строки

declare

op char(1) := 'I'; --локальные переменные в тригере

begin

case

when inserting

then

op := 'I';

logging (op, :NEW.zno, 'LEGAL\_NAME\_CLIENT\_COMPANY', null, :NEW.LEGAL\_NAME\_CLIENT\_COMPANY);

logging (op, :NEW.zno, 'EMAIL\_ADDRESS', null, :NEW.EMAIL\_ADDRESS);

logging (op, :NEW.zno, 'TEL\_NO', null, :NEW.TEL\_NO);

when updating('LEGAL\_NAME\_CLIENT\_COMPANY') or updating('EMAIL\_ADDRESS') or updating('TEL\_NO')

then

op := 'U';

logging (op, :NEW.zno, 'LEGAL\_NAME\_CLIENT\_COMPANY', :OLD.LEGAL\_NAME\_CLIENT\_COMPANY, :NEW.LEGAL\_NAME\_CLIENT\_COMPANY);

logging (op, :NEW.zno, 'EMAIL\_ADDRESS', :OLD.EMAIL\_ADDRESS, :NEW.EMAIL\_ADDRESS);

logging (op, :NEW.zno, 'TEL\_NO', :OLD.TEL\_NO, :NEW.TEL\_NO);

when deleting

then

op := 'D';

logging (op, :OLD.zno, 'LEGAL\_NAME\_CLIENT\_COMPANY', :OLD.LEGAL\_NAME\_CLIENT\_COMPANY, null);

logging (op, :OLD.zno, 'EMAIL\_ADDRESS', :OLD.EMAIL\_ADDRESS, null);

logging (op, :OLD.zno, 'TEL\_NO', :OLD.TEL\_NO, null);

else

null;

end case;

end clients\_log;

Проверка

insert into zclient(legal\_name\_client\_company, tel\_no, email\_address, date\_of\_last\_order) values ('TEST','+375(21)232-12-15','test@mailru','10.15.2021');

update zclient set tel\_no='+375(29)660-71-82' where legal\_name\_client\_company='TEST';

delete from zclient where legal\_name\_client\_company='TEST';

select\*from log1

Изображение выглядит как текст, снимок экрана, монитор, внутренний

Автоматически созданное описание

Написал DDL-триггер, протоколирующий действия пользователей по созданию, изменению и удалению таблиц в схеме во вспомогательную таблицу LOG2 в определенное время и запрещающий эти действия в другое время.

CREATE TABLE hog (

id INT GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY NOT NULL,

operation\_name CHAR(6),

handler\_uname CHAR(100),

table\_name VARCHAR2(100),

operation\_date timestamp);

--процедура

CREATE OR REPLACE PROCEDURE hog\_procedure(

operation\_name IN CHAR,

table\_name IN VARCHAR2)

IS

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

IF operation\_name IN ('CREATE','ALTER','DROP')

THEN

INSERT INTO hog

(operation\_name, handler\_uname, table\_name, operation\_date)

VALUES (operation\_name, USER, table\_name, SYSDATE);

COMMIT;

END IF;

END;

/

--триггер

CREATE OR REPLACE TRIGGER hog\_trigger

BEFORE CREATE OR ALTER OR DROP

ON SCHEMA

DECLARE

OPERATION CHAR (6) := 'CREATE';

L\_SYSEVENT VARCHAR(10);

MANIPULATING\_TABLE\_NAME VARCHAR(20);

BEGIN

IF to\_number(to\_char(SYSDATE, 'hh24')) between 1 and 7

THEN

RAISE\_APPLICATION\_ERROR(

NUM => -20000,

MSG => 'Rejected.');

ELSE

SELECT ORA\_SYSEVENT INTO L\_SYSEVENT FROM DUAL;

SELECT ORA\_DICT\_OBJ\_NAME INTO MANIPULATING\_TABLE\_NAME FROM DUAL;

IF (L\_SYSEVENT = 'CREATE')

THEN

OPERATION := 'CREATE';

hog\_procedure(OPERATION, MANIPULATING\_TABLE\_NAME);

ELSE NULL;

END IF;

IF (L\_SYSEVENT = 'ALTER')

THEN

OPERATION := 'ALTER';

hog\_procedure(OPERATION, MANIPULATING\_TABLE\_NAME);

ELSE NULL;

END IF;

IF (L\_SYSEVENT = 'DROP')

THEN

OPERATION := 'DROP';

hog\_procedure(OPERATION, MANIPULATING\_TABLE\_NAME);

ELSE NULL;

END IF;

END IF;

END acc\_TRIGGER;

/

Пример

create table фара( test int );

drop table фара

Вывод

select\*from hog

Изображение выглядит как текст, снимок экрана, монитор, черный

Автоматически созданное описание

Написал системный триггер, добавляющий запись во вспомогательную таблицу LOG3, когда пользователь подключается или отключается. В таблицу логов записывается имя пользователя (USER), тип активности (LOGON или LOGOFF), дата (SYSDATE), количество записей в основной таблице БД.

CREATE TABLE LOG3(

log3\_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

username VARCHAR2(50),

action\_type VARCHAR2(10),

action\_date DATE,

row\_count INTEGER,

CONSTRAINT table\_action\_type\_check\_constraint CHECK( action\_type IN ('LOGON', 'LOGOFF') )

);

create or replace TRIGGER user\_log\_logoff

BEFORE LOGOFF ON SCHEMA

DECLARE

CURSOR cur IS

SELECT COUNT(1) val FROM log2;

cur\_row cur%ROWTYPE;

BEGIN

OPEN cur;

FETCH cur INTO cur\_row;

INSERT INTO LOG3 (username, action\_date, action\_type, row\_count) VALUES (

USER, SYSDATE, 'LOGOFF', cur\_row.val

);

CLOSE cur;

END user\_log\_logoff;

create or replace TRIGGER user\_log\_logon

AFTER LOGON ON SCHEMA

DECLARE

CURSOR cur IS

SELECT COUNT(1) val FROM log2;

cur\_row cur%ROWTYPE;

BEGIN

OPEN cur;

FETCH cur INTO cur\_row;

INSERT INTO LOG3 (username, action\_date, action\_type, row\_count) VALUES (

USER, SYSDATE, 'LOGON', cur\_row.val

);

CLOSE cur;

END user\_log\_logon;

---------------------------------------------------------------------------------------------------------------------------------------------------4

Отслеживать количество единиц товаров на складе при совершении оптовых продаж.

Выдавать сообщение о необходимости пополнения склада, если остав-шееся количество товара меньше, чем его среднее значение по продажам за по-следний месяц.

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------------------------------------------------------------

CREATE OR REPLACE FUNCTION get\_avr\_count(product\_id IN NUMBER)

RETURN NUMBER AS

orders\_count NUMBER;

product\_count NUMBER;

BEGIN

SELECT SUM(C.quantity) INTO product\_count

From сshopping\_cart C

Join iproducts I On C.ino = I.ino

WHERE I.ino = product\_id;

SELECT COUNT(\*) INTO orders\_count

From сshopping\_cart C

Join iproducts I On C.ino = I.ino

WHERE I.ino = product\_id;

RETURN product\_count / orders\_count;

END;

---------------------------------------------------------------------------------------------------------------------------------------------------5

Организовал слежение за минимальным оптовым количеством товара в зависимости от его вида и производителя.

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------------------------------------------------------------

CREATE OR REPLACE TRIGGER control\_items

AFTER INSERT ON сshopping\_cart

FOR EACH ROW

DECLARE

avr\_ord\_count NUMBER := get\_avr\_count(:NEW.ino);

it\_left NUMBER;

SUPPLIERS\_name VARCHAR(2);

minimum\_order\_count NUMBER;

BEGIN

SELECT D.LEGAL\_NAME\_SUPPLIER\_COMPANY INTO SUPPLIERS\_name

FROM DSUPPLIERS D

JOIN iproducts I ON D.dno = I.dno

WHERE I.ino = :NEW.ino;

IF (SUPPLIERS\_name = 'МТЗ') THEN

minimum\_order\_count := 5;

ELSE

minimum\_order\_count := 3;

END IF;

IF (:NEW.quantity < minimum\_order\_count) THEN

RAISE\_APPLICATION\_ERROR(

NUM => -2,

MSG => 'Слишком мало заказали'

);

END IF;

SELECT I.quantity INTO it\_left

FROM iproducts I

WHERE ino = :NEW.ino;

IF (it\_left < avr\_ord\_count) THEN

DBMS\_OUTPUT.put\_line ('Надо пополнить склад');

END IF;

END control\_items;

---------------------------------------------------------------------------------------------------------------------------------------------------6

Создал таблицу-отчёт, которая должна будет обновляться один раз в неделю или в месяц и содержать данные о производителе, количестве проданных еди-ниц товаров, сумме продаж.

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------------------------------------------------------------

create table otchet(

SUPPLIERS\_name varchar2(30),

PRODUCTS\_sell NUMBER,

money NUMBER

)

CREATE OR REPLACE PROCEDURE update\_otchet IS

CURSOR pr\_cur

IS

SELECT D.LEGAL\_NAME\_SUPPLIER\_COMPANY as name, sum(C.QUANTITY) as sm, sum(I.PRICE\*C.QUANTITY) as sm2

FROM DSUPPLIERS D join IPRODUCTS I

on D.dno = I.dno

join СSHOPPING\_CART C

on C.ino = I.ino

join LORDERS L

on L.lno = C.lno

where L.ORDER\_DATE > sysdate - 7

GROUP BY D.LEGAL\_NAME\_SUPPLIER\_COMPANY;

pr\_cursor\_row pr\_cur%ROWTYPE;

BEGIN

delete from otchet;

OPEN pr\_cur;

FETCH pr\_cur INTO pr\_cursor\_row;

WHILE pr\_cur%FOUND

LOOP

INSERT INTO otchet (SUPPLIERS\_name, PRODUCTS\_sell, money) VALUES (

pr\_cursor\_row.name, pr\_cursor\_row.sm, pr\_cursor\_row.sm2

);

FETCH pr\_cur INTO pr\_cursor\_row;

END LOOP;

CLOSE pr\_cur;

END;

BEGIN

DBMS\_SCHEDULER.CREATE\_PROGRAM

( program\_name => 'simple\_program1',

program\_type => 'STORED\_PROCEDURE',

program\_action => 'update\_otchet',

enabled => TRUE);

END;

BEGIN

DBMS\_SCHEDULER.CREATE\_SCHEDULE

(schedule\_name => 'simple\_schedule',

start\_date => SYSTIMESTAMP,

repeat\_interval => 'FREQ=WEEKLY; BYDAY=MON');

END;

Написал триггер INSTEAD OF для работы с необновляемым представлением, созданным после выполнения п. 2.4 задания, проверить DML-командами возможность обновления представления после включения триггера.

create or replace view infoprod\_view as (Созданное не изменяемого представления)

select ino, PRODUCT\_NAME, PRICE, DSUPPLIERS.LEGAL\_NAME\_SUPPLIER\_COMPANY

from IPRODUCTS join DSUPPLIERS using (dno)

insert into infoprod\_view values('Холодильник','240','Железо'); (Проверка предстовления на изменения)

CREATE OR REPLACE TRIGGER restrig INSTEAD OF

UPDATE OR INSERT OR DELETE ON infoprod\_view

FOR EACH ROW

DECLARE

dno\_new DSUPPLIERS.dno%TYPE;

check\_excep\_LEGAL\_NAME\_SUPPLIER\_COMPANY DSUPPLIERS.LEGAL\_NAME\_SUPPLIER\_COMPANY%TYPE;

BEGIN

IF updating THEN --\* update

IF :new.LEGAL\_NAME\_SUPPLIER\_COMPANY != :old.LEGAL\_NAME\_SUPPLIER\_COMPANY THEN

SELECT LEGAL\_NAME\_SUPPLIER\_COMPANY INTO check\_excep\_LEGAL\_NAME\_SUPPLIER\_COMPANY

FROM DSUPPLIERS WHERE LEGAL\_NAME\_SUPPLIER\_COMPANY = :new.LEGAL\_NAME\_SUPPLIER\_COMPANY;

SELECT dno INTO dno\_new

FROM DSUPPLIERS WHERE LEGAL\_NAME\_SUPPLIER\_COMPANY = :new.LEGAL\_NAME\_SUPPLIER\_COMPANY;

ELSE

SELECT dno INTO dno\_new

FROM DSUPPLIERS WHERE LEGAL\_NAME\_SUPPLIER\_COMPANY = :old.LEGAL\_NAME\_SUPPLIER\_COMPANY;

END IF;

UPDATE IPRODUCTS SET

PRODUCT\_NAME = :new.PRODUCT\_NAME,

PRICE = :new.PRICE,

dno = dno\_new

WHERE

ino = :old.ino;

END IF;

IF inserting THEN

SELECT LEGAL\_NAME\_SUPPLIER\_COMPANY INTO check\_excep\_LEGAL\_NAME\_SUPPLIER\_COMPANY

FROM DSUPPLIERS WHERE LEGAL\_NAME\_SUPPLIER\_COMPANY = :new.LEGAL\_NAME\_SUPPLIER\_COMPANY;

SELECT dno INTO dno\_new

FROM DSUPPLIERS

WHERE LEGAL\_NAME\_SUPPLIER\_COMPANY = :new.LEGAL\_NAME\_SUPPLIER\_COMPANY;

INSERT INTO IPRODUCTS (ino, PRODUCT\_NAME, PRICE, dno)

VALUES ( :new.ino, :new.PRODUCT\_NAME, :new.PRICE, dno\_new);

END IF;

EXCEPTION

WHEN no\_data\_found THEN RAISE\_APPLICATION\_ERROR(-20005,'Таких данных нет!');

END;

/

(Триггер на изменения неизменяемого представления)