**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**ФГБОУ ВО**

**ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ**

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МОДЕЛИРОВАНИЯ  
 И УПРАВЛЕНИЯ**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2**

**по дисциплине**

**«РАЗРАБОТКА БАЗ ДАННЫХ В СУБД ORACLE»**

**Автор \_\_\_\_\_\_\_\_\_\_\_\_** Москалев В.А 04.01.2025**\_\_\_\_\_\_\_\_\_группа\_\_\_** ЗИП-23м **\_\_**

инициалы, фамилия, дата

**Специальность\_\_\_\_**09.04.03 Прикладная информатика**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

шифр, наименование

**Шифр\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**237302**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

зачетной книжки номер варианта

**Отчет защищен\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

дата подпись

**Принял\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

подпись дата инициалы фамилия

**Воронеж 2025 г.**



Схема БД

Таблица Клиенты (CLIENT)

- "ID": NUMBER Уникальный идентификатор клиента, первичный ключ;

- "FIRST\_NAME": VARCHAR2(100) Имя клиента;

- "LAST\_NAME": VARCHAR2(100) Фамилия клиента;

- "SURNAME": VARCHAR2(100) Отчество клиента;

- "PASSPORT\_NUM": VARCHAR2(100) Серия и номер паспорта клиента;

Таблица Вкладов (BANK\_ACCOUNT)

- "ID" NUMBER(10,0) Уникальный идентификатор вклада, первичный ключ;

- "CLIENT\_ID" NUMBER(10,0) Уникальный идентификатор клиента, внешний ключ на таблицу CLIENT;

- "DEPOSITE" NUMBER(38,2) Сумма депозита в формате 100,00;

- "RATE" NUMBER(10,2) Процентная ставка депозита;

- "DAY\_OF\_DEPOSITE" NUMBER(10,0) Период депозита в днях;

- "INCOME" NUMBER(38,2) Доход по истечению срока вклада, с учетом налога;

- "CAPITALIZATION" CHAR(1) Имеется ли ежемесячная капитализация;

- "CURRENCY" VARCHAR2(20) Тип валюты вклада;

- "TAX" NUMBER(38,2) Удержанный налог;

- "DT\_SYS" DATE Дата вставки записи

Таблица CLIENT имеет связь один-ко-многим к таблице BANK\_ACCOUNT через внешний ключ CLIENT\_ID.

Тоесть у одного клиента может быть много счетов.

В логике скриптов учтен расчет колонки INCOME по тригеру.

Тоесть расчет проводиться в зависимости от параметра CAPITALIZATION

CREATE TABLE "UNIVER"."CLIENT"

(

"ID" NUMBER,

"FIRST\_NAME" VARCHAR2(100) NOT NULL ENABLE,

"LAST\_NAME" VARCHAR2(100) NOT NULL ENABLE,

"SURNAME" VARCHAR2(100) NOT NULL ENABLE,

"PASSPORT\_NUM" VARCHAR2(100) NOT NULL ENABLE,

CONSTRAINT "CLIENT\_PK" PRIMARY KEY ("ID"),

CONSTRAINT "CLIENT\_UN" UNIQUE ("PASSPORT\_NUM"));

CREATE UNIQUE INDEX "UNIVER"."CLIENT\_PK" ON "UNIVER"."CLIENT" ("ID") PCTFREE 10 INITRANS 2 MAXTRANS 255 TABLESPACE "USERS";

CREATE UNIQUE INDEX "UNIVER"."CLIENT\_UN" ON "UNIVER"."CLIENT" ("PASSPORT\_NUM") PCTFREE 10 INITRANS 2 MAXTRANS 255 TABLESPACE "USERS"

COMMENT

ON COLUMN "UNIVER"."CLIENT".FIRST\_NAME IS 'Имя клиента';

COMMENT

ON COLUMN "UNIVER"."CLIENT".LAST\_NAME IS 'Фамилия клиента';

COMMENT

ON COLUMN "UNIVER"."CLIENT".SURNAME IS 'Отчество клиента';

COMMENT

ON COLUMN "UNIVER"."CLIENT".PASSPORT\_NUM IS 'Серия и номер паспорта клиента'; CREATE TABLE "UNIVER"."CLIENT"

(

"ID" NUMBER,

"FIRST\_NAME" VARCHAR2(100) NOT NULL ENABLE,

"LAST\_NAME" VARCHAR2(100) NOT NULL ENABLE,

"SURNAME" VARCHAR2(100) NOT NULL ENABLE,

"PASSPORT\_NUM" VARCHAR2(100) NOT NULL ENABLE,

CONSTRAINT "CLIENT\_PK" PRIMARY KEY ("ID"),

CONSTRAINT "CLIENT\_UN" UNIQUE ("PASSPORT\_NUM"));

CREATE UNIQUE INDEX "UNIVER"."CLIENT\_PK" ON "UNIVER"."CLIENT" ("ID") PCTFREE 10 INITRANS 2 MAXTRANS 255 TABLESPACE "USERS";

CREATE UNIQUE INDEX "UNIVER"."CLIENT\_UN" ON "UNIVER"."CLIENT" ("PASSPORT\_NUM") PCTFREE 10 INITRANS 2 MAXTRANS 255 TABLESPACE "USERS"

COMMENT

ON COLUMN "UNIVER"."CLIENT".FIRST\_NAME IS 'Имя клиента';

COMMENT

ON COLUMN "UNIVER"."CLIENT".LAST\_NAME IS 'Фамилия клиента';

COMMENT

ON COLUMN "UNIVER"."CLIENT".SURNAME IS 'Отчество клиента';

COMMENT

ON COLUMN "UNIVER"."CLIENT".PASSPORT\_NUM IS 'Серия и номер паспорта клиента';

-- "UNIVER".BANK\_ACCOUNT definition

CREATE TABLE "UNIVER"."BANK\_ACCOUNT"

(

"ID" NUMBER(10,0) NOT NULL ENABLE,

"CLIENT\_ID" NUMBER(10,0) NOT NULL ENABLE,

"DEPOSITE" NUMBER(38,2) NOT NULL ENABLE,

"RATE" NUMBER(10,2) NOT NULL ENABLE,

"DAY\_OF\_DEPOSITE" NUMBER(10,0) NOT NULL ENABLE,

"INCOME" NUMBER(38,2) NOT NULL ENABLE,

"CAPITALIZATION" CHAR(1) NOT NULL ENABLE,

"CURRENCY" VARCHAR2(20) NOT NULL ENABLE,

"TAX" NUMBER(38,2) NOT NULL ENABLE,

"DT\_SYS" DATE,

CONSTRAINT "BANK\_ACCOUNT\_PK" PRIMARY KEY ("ID")

)

CREATE UNIQUE INDEX "UNIVER"."BANK\_ACCOUNT\_PK" ON "UNIVER"."BANK\_ACCOUNT" ("ID");

COMMENT

ON COLUMN "UNIVER".BANK\_ACCOUNT.ID IS 'Уникальный идентификатор записи';

COMMENT

ON COLUMN "UNIVER".BANK\_ACCOUNT.CLIENT\_ID IS 'Идентификатор пользователя. Ссылка на таблицу client';

COMMENT

ON COLUMN "UNIVER".BANK\_ACCOUNT.DEPOSITE IS 'Сумма вклада';

COMMENT

ON COLUMN "UNIVER".BANK\_ACCOUNT.RATE IS 'Процентная ставка вклада';

COMMENT

ON COLUMN "UNIVER".BANK\_ACCOUNT.DAY\_OF\_DEPOSITE IS 'Период депозита в днях';

COMMENT

ON COLUMN "UNIVER".BANK\_ACCOUNT.INCOME IS 'Доход по истечению срока вклада, с учетом налога';

COMMENT

ON COLUMN "UNIVER".BANK\_ACCOUNT.CAPITALIZATION IS 'Имеется ли ежемесячная капитализация';

COMMENT

ON COLUMN "UNIVER".BANK\_ACCOUNT.CURRENCY IS 'Тип валюты вклада';

COMMENT

ON COLUMN "UNIVER".BANK\_ACCOUNT.TAX IS 'Удержанный налог';

-- "UNIVER".BANK\_ACCOUNT foreign keys

ALTER TABLE "UNIVER"."BANK\_ACCOUNT"

ADD CONSTRAINT "FK\_CLIENT" FOREIGN KEY ("CLIENT\_ID")

REFERENCES "UNIVER"."CLIENT" ("ID") ON DELETE CASCADE ENABLE;

1 В соответствии с выбранным вариантом задания, создать не менее 2

процедур обеспечивающих логику программы, описанную в лабораторной работе №1 (функции, которая выполняет база данных).

Решение:

1. –процедура переводит столбцы в верхний регистр

**CREATE** **OR** **REPLACE** **PROCEDURE** UpdateNamesToUpperCase **IS**

**BEGIN**

**UPDATE** CLIENT

**SET**

first\_name = **UPPER**(first\_name),

last\_name = **UPPER**(last\_name);

**COMMIT**; -- Применение изменений

**END** UpdateNamesToUpperCase;

--ВЫЗОВ ФУНКЦИИ

**CALL** UNIVER.UPDATENAMESTOUPPERCASE();

2)

-- процедура прибавляет tax к income и обнуляем tax

--тоесть делает income значение без учета налога

**CREATE** **OR** **REPLACE** **PROCEDURE** UpdateIncomeAndClearTax **IS**

**BEGIN**

**UPDATE** BANK\_ACCOUNT

**SET**

INCOME = INCOME + TAX , --

TAX = 0 -- Обнуляем tax

;

**COMMIT**; -- Применяем изменения

**END** UpdateIncomeAndClearTax;

--ВЫЗОВ ФУНКЦИИ

**CALL** UNIVER.UPDATEINCOMEANDCLEARTAX();

2 Создать 2 вспомогательные функции, которые необходимо вызывать

из тела процедуры.

1)

--ВЕРНУТЬ ДЕПОЗИТ КОТОРЫЙ ВЫШЕ СРЕДНЕГО ЗНАЧЕНИЯ ВСЕХ ДЕПОЗИТОВ

**CREATE** **OR** **REPLACE** **FUNCTION** GET\_MAX\_DEPOSITE\_BY\_AVG **RETURN** **NUMBER**

**IS**

lvValue **NUMBER**;

**BEGIN**

-- Выполняем запрос и сохраняем результат в переменную

**SELECT** **MAX**(ba.DEPOSITE) -- Нам нужен максимальный депозит, чтобы получить максимальную сумму

**INTO** lvValue

**FROM** BANK\_ACCOUNT ba

**JOIN** CLIENT c **ON** ba.CLIENT\_ID = c.ID

**WHERE** ba.DEPOSITE >= (**SELECT** **AVG**(DEPOSITE) **FROM** BANK\_ACCOUNT);

**RETURN** lvValue; -- Возвращаем значение

**END** GET\_MAX\_DEPOSITE\_BY\_AVG;

--ВЫЗОВ ФУНКЦИИ

**SELECT** GET\_MAX\_DEPOSITE\_BY\_AVG() **FROM** DUAL;

2) --ВЕРНУТЬ TRUE если есть клиент с 500к депозитом

**CREATE** **OR** **REPLACE** **FUNCTION** EXIST\_CLIENT\_WITH\_500\_K\_DEPOSITE **RETURN** **NUMBER**

**IS**

lvAnswer **BOOLEAN** := **FALSE**; -- lvAnswer типа BOOLEAN

lvCount **NUMBER**;

**BEGIN**

-- Выполняем запрос, чтобы посчитать количество клиентов с депозитом 500000

**SELECT** **COUNT**(\*)

**INTO** lvCount

**FROM** BANK\_ACCOUNT ba

**WHERE** ba.DEPOSITE = 500000;

-- Если найден хотя бы один клиент, то устанавливаем lvAnswer в TRUE

**IF** lvCount > 0 **THEN**

lvAnswer := **TRUE**;

**ELSE**

lvAnswer := **FALSE**;

**END** **IF**;

-- Возвращаем 1 (если TRUE) или 0 (если FALSE)

**IF** lvAnswer **THEN**

**RETURN** 1;

**ELSE**

**RETURN** 0;

**END** **IF**;

**END** EXIST\_CLIENT\_WITH\_500\_K\_DEPOSITE;

**SELECT** EXIST\_CLIENT\_WITH\_500\_K\_DEPOSITE() **FROM** DUAL;