**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**ФГБОУ ВО**

**ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ**

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МОДЕЛИРОВАНИЯ  
 И УПРАВЛЕНИЯ**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1**

**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.**

Банковский аккаунт

КЛИЕНТ

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

Рис.1 Таблицы БД

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. Создать запросы на выборку данных производящих выбор всех строк из    таблиц,    выбор    нескольких    строк    с    помощью    предиката    WHERE с различными операторами: BETWEEN, LIKE, AND, OR, NOT >, <, =, <> (не менее 8-ми запросов).

Решение:

1) **SELECT** \* **FROM** BANK\_ACCOUNT ba **WHERE** DEPOSITE **BETWEEN** 100000 **AND** 200000

2) **SELECT** \* **FROM** BANK\_ACCOUNT ba **WHERE** CURRENCY **LIKE** 'RUB'

3) **SELECT** \* **FROM** BANK\_ACCOUNT ba **WHERE** CURRENCY **LIKE** 'RUB' **AND** CURRENCY **LIKE** 'USD'

4) **SELECT** \* **FROM** BANK\_ACCOUNT ba **WHERE** CURRENCY **LIKE** 'RUB' **OR** CAPITALIZATION **LIKE** 'Y'

5) **SELECT** \* **FROM** BANK\_ACCOUNT ba **WHERE** CAPITALIZATION <> 'Y'

6) **SELECT** \* **FROM** BANK\_ACCOUNT ba **WHERE** DEPOSITE <> 10000

7) **SELECT** \* **FROM** BANK\_ACCOUNT ba **WHERE** RATE **BETWEEN** 15 **AND** 21

8) **SELECT** \* **FROM** BANK\_ACCOUNT ba **WHERE** DAY\_OF\_DEPOSITE >=365

2. Создать запросы, производящие соединение таблиц: внутреннее, левое внешнее, правое внешнее, полное внешнее, декартовое произведение (не менее 4-х запросов).

Решение:

1) **SELECT** \* **FROM** BANK\_ACCOUNT ba

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

2) **SELECT** c.ID **FROM** BANK\_ACCOUNT ba

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

3) **SELECT** c.ID **FROM** BANK\_ACCOUNT ba

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

4) **SELECT** \* **FROM** BANK\_ACCOUNT ba

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

5) **SELECT** \* **FROM** BANK\_ACCOUNT ba

**CROSS** **JOIN** CLIENT c

3. С помощью оператора SELECT произвести следующие реляционные операции: объединение, разность, пересечение (не менее 2-х запросов на каждую операцию).

Решение:

1) **SELECT** \* **FROM** BANK\_ACCOUNT ba

**WHERE** DEPOSITE =100000

**UNION** **ALL**

**SELECT** \* **FROM** BANK\_ACCOUNT ba

**WHERE** RATE = 25

2) **SELECT** \* **FROM** BANK\_ACCOUNT ba

**WHERE** DEPOSITE =100000

**MINUS**

**SELECT** \* **FROM** BANK\_ACCOUNT ba

**WHERE** RATE = 25

3) **SELECT** \* **FROM** BANK\_ACCOUNT ba

**WHERE** DEPOSITE =100000

**UNION**

**SELECT** \* **FROM** BANK\_ACCOUNT ba

**WHERE** DEPOSITE <> 200000;

4. Создать запросы, содержащие подзапросы и операторы IN, ANY, ALL (не менее 4-х запросов).

Решение:

1) **SELECT** \* **FROM** BANK\_ACCOUNT ba

**WHERE** ba.CLIENT\_ID **IN**

(**SELECT** ID **FROM** CLIENT c **WHERE** c.SURNAME **LIKE** 'Petrovich')

2) **SELECT** \* **FROM** BANK\_ACCOUNT ba

**WHERE** ba.CLIENT\_ID = **ANY**

(**SELECT** ID **FROM** CLIENT c **WHERE** c.SURNAME **LIKE** 'Petrovich' **OR** c.SURNAME **LIKE** 'Futher')

3) **SELECT** \* **FROM** BANK\_ACCOUNT ba

**WHERE** ba.DEPOSITE >

**ALL** (**SELECT** ba2.DEPOSITE **FROM** BANK\_ACCOUNT ba2 **WHERE** ba2.CLIENT\_ID = 1)

4) **SELECT** \* **FROM** BANK\_ACCOUNT ba

**WHERE** ba.CAPITALIZATION **IN** ('Y','N')

5. Создать запросы, содержащие группировку строк с применением агрегатных функций: SUM, COUNT, MIN, MAX, AVG (не менее 2-х запросов).

Решение:

1) **SELECT** **COUNT**(c.ID) **FROM** BANK\_ACCOUNT ba

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

**WHERE** c.ID = 1

2) **SELECT** DEPOSITE **FROM** BANK\_ACCOUNT ba

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

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

6. Создать запросы производящих изменение значение во всех строках таблицы и в некоторых строках таблицы (не менее 2-х запросов).

Решение:

1) **UPDATE** BANK\_ACCOUNT **SET** DEPOSITE = DEPOSITE + 1

2) **UPDATE** BANK\_ACCOUNT **SET** DEPOSITE = DEPOSITE + 100000

**WHERE** DEPOSITE >=200000

7. Создать запросы производящие удаление некоторых строк в таблицы и всех строк таблицы.

Решение:

1) **TRUNCATE** **TABLE** client;

2) **DELETE** **FROM** BANK\_ACCOUNT **WHERE** DEPOSITE >= 20000000;