Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Базы данных

ОТЧЁТ

по лабораторной работе №3

«Реализация SQL-запросов для создания схемы базы данных»

Выполнил                                                                   Студенты: гр.310901                                                                                 Усов А. М.

Проверила                                                                                 Воробей А. В.

Минск 2025

Создание и подключение к базе данных PostgreSQL:

CREATE DATABASE dreamdb

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'en\_US.UTF-8'

LC\_CTYPE = 'en\_US.UTF-8'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1;

CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

Скрипты создания таблиц:

CREATE TABLE Role (

RoleId INT PRIMARY KEY,

Title VARCHAR(50) NOT NULL,

Description VARCHAR(255) NOT NULL);

CREATE TABLE User\_ (

ConsumerUserId UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

Email VARCHAR(50) NOT NULL,

Password VARCHAR(255) NOT NULL,

Name VARCHAR(50) NOT NULL,

RoleId INT NOT NULL,

ProducerUserId UUID,

FOREIGN KEY (RoleId) REFERENCES Role(RoleId),

FOREIGN KEY (ProducerUserId) REFERENCES ProducerUser(ProducerUserId));

CREATE TABLE ProducerUser (

ProducerUserId UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

SecretKey VARCHAR(255) NOT NULL,

ProducerId UUID NOT NULL);

CREATE TABLE Producer (

ProducerId UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

Title VARCHAR(50) NOT NULL,

Description VARCHAR(1000),

Rating DECIMAL NOT NULL CHECK (Rating >= 0 AND Rating <= 5));

CREATE TABLE DreamCategory (

DreamCategoryId UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

Title VARCHAR(100) NOT NULL,

Description VARCHAR(1000));

CREATE TABLE Dream (

DreamId UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

Title VARCHAR(100) NOT NULL,

Description VARCHAR(1000),

ProducerUserId UUID NOT NULL,

ProducerId UUID NOT NULL,

Price DECIMAL NOT NULL CHECK (Price >= 0),

ImageFileName VARCHAR(255) NOT NULL,

FOREIGN KEY (ProducerUserId) REFERENCES ProducerUser(ProducerUserId),

FOREIGN KEY (ProducerId) REFERENCES Producer(ProducerId));

CREATE TABLE DreamCategoryDreamXRef (

DreamId UUID,

DreamCategoryId UUID,

PRIMARY KEY (DreamId, DreamCategoryId),

FOREIGN KEY (DreamId) REFERENCES Dream(DreamId),

FOREIGN KEY (DreamCategoryId) REFERENCES DreamCategory(DreamCategoryId));

CREATE TABLE File\_ (

FileName VARCHAR(255) PRIMARY KEY,

Path VARCHAR(255) NOT NULL,

UpdateData TIMESTAMP NOT NULL,

File BYTEA NOT NULL);

CREATE TABLE Order\_ (

OrderId UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

CreatedAt TIMESTAMP NOT NULL,

UserId UUID NOT NULL,

FOREIGN KEY (UserId) REFERENCES User\_(ConsumerUserId));

-- Many-to-many: Order <-> Dream

CREATE TABLE OrderDreamXRef (

OrderId UUID,

DreamId UUID,

Count INT NOT NULL CHECK (Count > 0),

PRIMARY KEY (OrderId, DreamId),

FOREIGN KEY (OrderId) REFERENCES Order\_(OrderId),

FOREIGN KEY (DreamId) REFERENCES Dream(DreamId));

CREATE TABLE Payment (

PaymentId UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

OrderId UUID NOT NULL,

PaymentDate TIMESTAMP NOT NULL,

Message VARCHAR(1000),

ChequeFileName VARCHAR(255) NOT NULL,

FOREIGN KEY (OrderId) REFERENCES Order\_(OrderId),

FOREIGN KEY (ChequeFileName) REFERENCES File\_(FileName));

CREATE TABLE Review (

ReviewId UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

DreamId UUID NOT NULL,

UserId UUID NOT NULL,

Rating DECIMAL NOT NULL CHECK (Rating >= 1 AND Rating <= 5),

Comment VARCHAR(1000),

CreatedAt TIMESTAMP NOT NULL,

FOREIGN KEY (DreamId) REFERENCES Dream(DreamId),

FOREIGN KEY (UserId) REFERENCES User\_(ConsumerUserId));

Заполнение таблиц данными:

INSERT INTO Role (RoleId, Title, Description) VALUES

(1, 'user', 'Regular consumer user'),

(2, 'producer', 'Content producer'),

(3, 'producer\_admin', 'Admin of a producer group'),

(4, 'admin', 'System administrator');

INSERT INTO Producer (ProducerId, Title, Description, Rating) VALUES

(default, 'DreamWorks Studio', 'Creates surreal dream experiences', 4.8),

(default, 'Lucid Labs', 'Scientifically optimized dreams', 4.6),

(default, 'NightScape', 'Fantasy and adventure dreams', 4.7),

(default, 'CalmMind', 'Relaxing and meditative dreams', 4.9);

INSERT INTO ProducerUser (ProducerUserId, SecretKey, ProducerId) VALUES

(default, 'sk\_prod\_abc123', (SELECT ProducerId FROM Producer WHERE Title = 'DreamWorks Studio')),

(default, 'sk\_prod\_def456', (SELECT ProducerId FROM Producer WHERE Title = 'Lucid Labs')),

(default, 'sk\_prod\_ghi789', (SELECT ProducerId FROM Producer WHERE Title = 'NightScape')),

(default, 'sk\_prod\_jkl000', (SELECT ProducerId FROM Producer WHERE Title = 'CalmMind'));

INSERT INTO User\_ (Email, Password, Name, RoleId, ProducerUserId) VALUES

('alice@example.com', 'pass123', 'Alice', 1, NULL),

('bob@example.com', 'pass123', 'Bob', 1, NULL),

('charlie@example.com', 'pass123', 'Charlie', 1, NULL),

('diana@example.com', 'pass123', 'Diana', 1, NULL),

('eve@example.com', 'pass123', 'Eve', 1, NULL),

('frank@example.com', 'pass123', 'Frank', 1, NULL),

('grace@example.com', 'pass123', 'Grace', 1, NULL),

('heidi@example.com', 'pass123', 'Heidi', 1, NULL),

('ivan@example.com', 'pass123', 'Ivan', 1, NULL),

('judy@example.com', 'pass123', 'Judy', 1, NULL),

('producer1@example.com', 'pass123', 'Mike', 2, (SELECT ProducerUserId FROM ProducerUser LIMIT 1 OFFSET 0)),

('producer2@example.com', 'pass123', 'Nina', 2, (SELECT ProducerUserId FROM ProducerUser LIMIT 1 OFFSET 1)),

('admin@example.com', 'adminpass', 'Admin', 4, NULL),

('producer\_admin@example.com', 'prodadmin', 'Paul', 3, (SELECT ProducerUserId FROM ProducerUser LIMIT 1 OFFSET 2));

INSERT INTO DreamCategory (Title, Description) VALUES

('Adventure', 'Exciting journeys and quests'),

('Fantasy', 'Magical worlds and creatures'),

('Relaxation', 'Peaceful and calming environments'),

('Sci-Fi', 'Futuristic and space-based dreams'),

('Romance', 'Emotional and loving experiences');

INSERT INTO Dream (Title, Description, ProducerUserId, ProducerId, Price, ImageFileName) VALUES

('Journey to Mars', 'Walk on the red planet with AI companions', (SELECT ProducerUserId FROM ProducerUser LIMIT 1), (SELECT ProducerId FROM Producer LIMIT 1), 29.99, 'image\_' || uuid\_generate\_v4() || '.jpg'),

('Underwater Palace', 'Explore a coral kingdom ruled by merfolk', (SELECT ProducerUserId FROM ProducerUser OFFSET 1 LIMIT 1), (SELECT ProducerId FROM Producer OFFSET 1 LIMIT 1), 24.99, 'image\_' || uuid\_generate\_v4() || '.jpg'),

('Forest of Whispers', 'A serene forest with talking trees', (SELECT ProducerUserId FROM ProducerUser OFFSET 2 LIMIT 1), (SELECT ProducerId FROM Producer OFFSET 2 LIMIT 1), 19.99, 'image\_' || uuid\_generate\_v4() || '.jpg'),

('Cyberpunk City', 'Neon-lit metropolis with hovercars', (SELECT ProducerUserId FROM ProducerUser OFFSET 3 LIMIT 1), (SELECT ProducerId FROM Producer OFFSET 3 LIMIT 1), 34.99, 'image\_' || uuid\_generate\_v4() || '.jpg'),

('Love in Paris', 'Romantic evening by the Eiffel Tower', (SELECT ProducerUserId FROM ProducerUser LIMIT 1), (SELECT ProducerId FROM Producer LIMIT 1), 27.99, 'image\_' || uuid\_generate\_v4() || '.jpg'),

('Haunted Mansion', 'Solve mysteries in a spooky old house', (SELECT ProducerUserId FROM ProducerUser OFFSET 1 LIMIT 1), (SELECT ProducerId FROM Producer OFFSET 1 LIMIT 1), 22.99, 'image\_' || uuid\_generate\_v4() || '.jpg'),

('Sky Islands', 'Floating islands with airships', (SELECT ProducerUserId FROM ProducerUser OFFSET 2 LIMIT 1), (SELECT ProducerId FROM Producer OFFSET 2 LIMIT 1), 25.99, 'image\_' || uuid\_generate\_v4() || '.jpg'),

('Time Traveler''s Dilemma', 'Change history without breaking reality', (SELECT ProducerUserId FROM ProducerUser OFFSET 3 LIMIT 1), (SELECT ProducerId FROM Producer OFFSET 3 LIMIT 1), 30.99, 'image\_' || uuid\_generate\_v4() || '.jpg');

INSERT INTO DreamCategoryDreamXRef (DreamId, DreamCategoryId)

SELECT d.DreamId, dc.DreamCategoryId

FROM Dream d, DreamCategory dc

WHERE d.Title = 'Journey to Mars' AND dc.Title = 'Sci-Fi'

UNION ALL

SELECT d.DreamId, dc.DreamCategoryId FROM Dream d, DreamCategory dc WHERE d.Title = 'Underwater Palace' AND dc.Title = 'Fantasy'

UNION ALL

SELECT d.DreamId, dc.DreamCategoryId FROM Dream d, DreamCategory dc WHERE d.Title = 'Forest of Whispers' AND dc.Title = 'Relaxation'

UNION ALL

SELECT d.DreamId, dc.DreamCategoryId FROM Dream d, DreamCategory dc WHERE d.Title = 'Cyberpunk City' AND dc.Title = 'Sci-Fi'

UNION ALL

SELECT d.DreamId, dc.DreamCategoryId FROM Dream d, DreamCategory dc WHERE d.Title = 'Love in Paris' AND dc.Title = 'Romance'

UNION ALL

SELECT d.DreamId, dc.DreamCategoryId FROM Dream d, DreamCategory dc WHERE d.Title = 'Haunted Mansion' AND dc.Title = 'Fantasy'

UNION ALL

SELECT d.DreamId, dc.DreamCategoryId FROM Dream d, DreamCategory dc WHERE d.Title = 'Sky Islands' AND dc.Title = 'Adventure'

UNION ALL

SELECT d.DreamId, dc.DreamCategoryId FROM Dream d, DreamCategory dc WHERE d.Title = 'Time Traveler''s Dilemma' AND dc.Title = 'Sci-Fi';

INSERT INTO File\_ (FileName, Path, UpdateData, File)

SELECT ImageFileName, '/images/', NOW(), ''::bytea FROM Dream

UNION ALL

SELECT 'cheque\_' || uuid\_generate\_v4() || '.pdf', '/cheques/', NOW(), ''::bytea FROM generate\_series(1, 10);

INSERT INTO Order\_ (CreatedAt, UserId)

SELECT NOW() - (random() \* 30 || ' days')::interval, ConsumerUserId

FROM User\_

WHERE RoleId = 1

ORDER BY random()

LIMIT 10;

INSERT INTO OrderDreamXRef (OrderId, DreamId, Count)

SELECT o.OrderId, d.DreamId, (random() \* 3 + 1)::int

FROM Order\_ o

CROSS JOIN Dream d

WHERE random() < 0.3

ORDER BY random()

LIMIT 15;

INSERT INTO Payment (OrderId, PaymentDate, Message, ChequeFileName)

SELECT o.OrderId, o.CreatedAt + '1 day'::interval, 'Payment confirmed', f.FileName

FROM Order\_ o

JOIN File\_ f ON f.FileName LIKE 'cheque%'

WHERE NOT EXISTS (SELECT 1 FROM Payment p WHERE p.OrderId = o.OrderId)

LIMIT 10;

INSERT INTO Review (DreamId, UserId, Rating, Comment, CreatedAt)

SELECT d.DreamId, u.ConsumerUserId, (random() \* 4 + 1)::decimal(2,1),

'Great experience! ' || md5(random()::text),

NOW() - (random() \* 10 || ' days')::interval

FROM Dream d

CROSS JOIN User\_ u

WHERE u.RoleId = 1

AND random() < 0.4

LIMIT 20;

Создан файл с резервной копией базы данных:

root@bde27e99e254:/# **pg\_dump -U DreamsShop\_Client -h localhost -p 5432 -F c -b -v -f "dreamdb\_backup.dump" DB-bsuir-course**

pg\_dump: last built-in OID is 16383

pg\_dump: reading extensions

pg\_dump: identifying extension members

pg\_dump: reading schemas

pg\_dump: reading user-defined tables

pg\_dump: reading user-defined functions

pg\_dump: reading user-defined types

pg\_dump: reading procedural languages

pg\_dump: reading user-defined aggregate functions

pg\_dump: reading user-defined operators

pg\_dump: reading user-defined access methods

pg\_dump: reading user-defined operator classes

pg\_dump: reading user-defined operator families

pg\_dump: reading user-defined text search parsers

pg\_dump: reading user-defined text search templates

pg\_dump: reading user-defined text search dictionaries

pg\_dump: reading user-defined text search configurations

pg\_dump: reading user-defined foreign-data wrappers

pg\_dump: reading user-defined foreign servers

pg\_dump: reading default privileges

pg\_dump: reading user-defined collations

pg\_dump: reading user-defined conversions

pg\_dump: reading type casts

pg\_dump: reading transforms

pg\_dump: reading table inheritance information

pg\_dump: reading event triggers

pg\_dump: finding extension tables

pg\_dump: finding inheritance relationships

pg\_dump: reading column info for interesting tables

pg\_dump: finding table default expressions

pg\_dump: finding table check constraints

pg\_dump: flagging inherited columns in subtables

pg\_dump: reading partitioning data

pg\_dump: reading indexes

pg\_dump: flagging indexes in partitioned tables

pg\_dump: reading extended statistics

pg\_dump: reading constraints

pg\_dump: reading triggers

pg\_dump: reading rewrite rules

pg\_dump: reading policies

pg\_dump: reading row-level security policies

pg\_dump: reading publications

pg\_dump: reading publication membership of tables

pg\_dump: reading publication membership of schemas

pg\_dump: reading subscriptions

pg\_dump: reading subscription membership of tables

pg\_dump: reading large objects

pg\_dump: reading dependency data

pg\_dump: saving encoding = UTF8

pg\_dump: saving "standard\_conforming\_strings = on"

pg\_dump: saving "search\_path = "

pg\_dump: saving database definition

pg\_dump: dumping contents of table "public.dream"

pg\_dump: dumping contents of table "public.dreamcategory"

pg\_dump: dumping contents of table "public.dreamcategorydreamxref"

pg\_dump: dumping contents of table "public.file\_"

pg\_dump: dumping contents of table "public.order\_"

pg\_dump: dumping contents of table "public.orderdreamxref"

pg\_dump: dumping contents of table "public.payment"

pg\_dump: dumping contents of table "public.producer"

pg\_dump: dumping contents of table "public.produceruser"

pg\_dump: dumping contents of table "public.review"

pg\_dump: dumping contents of table "public.role"

pg\_dump: dumping contents of table "public.user\_"

Файл бэкапа в контейнере:

root@bde27e99e254:/# **ls**

bin dev **dreamdb\_backup.dump** home lib64 mnt proc run srv tmp var

boot docker-entrypoint-initdb.d etc lib media opt root sbin sys usr

root@bde27e99e254:/#