|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №3**

**«Создание объектов базы данных средствами языка DDL»**

**ДИСЦИПЛИНА: «Базы данных»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-52Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Глебов С.А. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2022

**Цель:** закрепить навыки использования операторов определения данных, научится определять ограничения, индексы, просмотры.

**Задачи:**

* создать таблицы базы данных выбранной предметной области, используя СУБД Firebird,
* определить домены, столбцы, вычисляемые поля, ограничения первичного и внешнего ключей,
* организовать необходимые индексы и просмотры.

**Предметная область** – социальная сеть.

Создать базу данных под управлением SQL-сервера Firebird. Создать таблицы, ограничения, вычисляемые столбцы, просмотры предметной области в соответствии.

**Результат:**



Рис. 1. Связи таблиц

**Создание таблиц:**

CREATE TABLE Users(

Id BIGINT NOT NULL PRIMARY KEY,

Name VARCHAR(20) NOT NULL,

Avatar VARCHAR(20) NOT NULL,

Gender VARCHAR(1) NOT NULL,

BirthDate DATE NOT NULL

);

CREATE TABLE Friends(

Id BIGINT NOT NULL PRIMARY KEY,

FirstFriendId BIGINT NOT NULL,

SecondFriendId BIGINT NOT NULL,

FOREIGN KEY (FirstFriendId)

REFERENCES Users(Id)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (SecondFriendId)

REFERENCES Users(Id)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT id\_inequality CHECK (FirstFriendId <> SecondFriendId)

);

CREATE TABLE Groups(

Id BIGINT NOT NULL PRIMARY KEY,

Name VARCHAR(20) NOT NULL,

Avatar VARCHAR(20) NOT NULL,

OwnerId BIGINT NOT NULL,

FOREIGN KEY (OwnerId)

REFERENCES Users(Id)

ON DELETE SET NULL

ON UPDATE CASCADE

);

CREATE TABLE GroupPosts(

Id BIGINT NOT NULL PRIMARY KEY,

AuthorId BIGINT NOT NULL,

GroupId BIGINT NOT NULL,

Text VARCHAR(255) NOT NULL,

FOREIGN KEY (AuthorId)

REFERENCES Users(Id)

ON DELETE SET NULL

ON UPDATE CASCADE,

FOREIGN KEY (GroupId)

REFERENCES Groups(Id)

ON DELETE SET NULL

ON UPDATE CASCADE

);

CREATE TABLE GroupMembers(

Id BIGINT NOT NULL PRIMARY KEY,

GroupId BIGINT NOT NULL,

MemberId BIGINT NOT NULL,

FOREIGN KEY (GroupId)

REFERENCES Groups(Id)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (MemberId)

REFERENCES Users(Id)

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE GroupChats(

Id BIGINT NOT NULL PRIMARY KEY,

Name VARCHAR(20) NOT NULL,

Avatar VARCHAR(20),

OwnerId BIGINT NOT NULL,

FOREIGN KEY (OwnerId)

REFERENCES Users(Id)

ON DELETE SET NULL

ON UPDATE CASCADE

);

CREATE TABLE GroupChatMembers(

Id BIGINT NOT NULL PRIMARY KEY,

GroupChatId BIGINT NOT NULL,

MemberId BIGINT NOT NULL,

FOREIGN KEY (GroupChatId)

REFERENCES GroupChats(Id)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (MemberId)

REFERENCES Users(Id)

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE Messages(

Id BIGINT NOT NULL PRIMARY KEY,

SenderId BIGINT NOT NULL,

RecipientId BIGINT,

GroupChatId BIGINT,

Text VARCHAR(255) NOT NULL,

SendTime TIMESTAMP NOT NULL,

FOREIGN KEY (SenderId)

REFERENCES Users(Id)

ON DELETE SET NULL

ON UPDATE CASCADE,

FOREIGN KEY (RecipientId)

REFERENCES Users(Id)

ON DELETE SET NULL

ON UPDATE CASCADE,

FOREIGN KEY (GroupChatId)

REFERENCES GroupChats(Id)

ON DELETE SET NULL

ON UPDATE CASCADE,

CONSTRAINT end\_point\_choice CHECK (RecipientId IS NULL AND GroupChatId IS NOT NULL OR RecipientId IS NOT NULL AND GroupChatId IS NULL)

);

ALTER TABLE Users ADD CONSTRAINT gender\_check CHECK (Gender IN ('М', 'Ж'));

ALTER TABLE Users ADD FriendCount GENERATED ALWAYS AS ((SELECT COUNT(\*) FROM Friends WHERE Friends.FirstFriendId = Users.Id OR Friends.SecondFriendId = Users.Id));

**Добавление записей:**

INSERT INTO Users(Id, Name, Avatar, Gender, Birthdate)

VALUES (1, 'Иван Шведов', 'rock.png', 'М', '2002-04-07');

INSERT INTO Users(Id, Name, Avatar, Gender, Birthdate)

VALUES (2, 'Денис Кувшинов', 'cyberpunk.png', 'М', '2002-08-14');

INSERT INTO Users(Id, Name, Avatar, Gender, Birthdate)

VALUES (3, 'Александр Карпуткин', 'study.jpg', 'М', '2002-12-08');

INSERT INTO Users(Id, Name, Avatar, Gender, Birthdate)

VALUES (4, 'Виктория Дронова', 'flower.png', 'Ж', '2002-03-01');

INSERT INTO Friends(Id, FirstFriendId, SecondFriendId)

VALUES (1, 1, 4);

INSERT INTO Friends(Id, FirstFriendId, SecondFriendId)

VALUES (2, 4, 3);

INSERT INTO Friends(Id, FirstFriendId, SecondFriendId)

VALUES (3, 1, 2);

INSERT INTO Friends(Id, FirstFriendId, SecondFriendId)

VALUES (4, 3, 1);

INSERT INTO Groups(Id, Name, Avatar, OwnerId)

VALUES (1, 'Sport', 'Fist.png', 2);

INSERT INTO GroupPosts(Id, AuthorId, GroupId, Text)

VALUES (1, 2, 1, 'Первое занятие состоится 1 ноября');

INSERT INTO GroupMembers(Id, GroupId, MemberId)

VALUES (1, 1, 1);

INSERT INTO GroupMembers(Id, GroupId, MemberId)

VALUES (2, 1, 2);

INSERT INTO GroupMembers(Id, GroupId, MemberId)

VALUES (3, 1, 4);

INSERT INTO GroupChats(Id, Name, Avatar, OwnerId)

VALUES (1, 'Study', 'book.jpg', 3);

INSERT INTO GroupChatMembers(Id, GroupChatId, MemberId)

VALUES (1, 1, 1);

INSERT INTO GroupChatMembers(Id, GroupChatId, MemberId)

VALUES (2, 1, 3);

INSERT INTO GroupChatMembers(Id, GroupChatId, MemberId)

VALUES (3, 1, 4);

INSERT INTO Messages(Id, SenderId, RecipientId, GroupChatId, Text, SendTime)

VALUES (1, 2, 1, NULL, 'Пойдешь играть?', '2022-10-20 18:01');

INSERT INTO Messages(Id, SenderId, RecipientId, GroupChatId, Text, SendTime)

VALUES (2, 1, 2, NULL, 'Нет, я немного занят', '2022-10-20 18:04');

INSERT INTO Messages(Id, SenderId, RecipientId, GroupChatId, Text, SendTime)

VALUES (3, 4, NULL, 1, 'Что задали?', '2022-10-21 14:32');

**Создание представлений:**

CREATE VIEW DirectMessages AS

SELECT Sender, Users.Name AS Recipient, Text, SendTime

FROM

(SELECT Users.Name as Sender, RecipientId, Text, SendTime

FROM Messages

JOIN Users

ON Messages.SenderId = Users.Id)

JOIN Users

ON RecipientId = Users.Id

CREATE VIEW ChatMessages AS

SELECT Sender, GroupChats.Name AS Chat, Text, SendTime

FROM

(SELECT Users.Name as Sender, GroupChatId, Text, SendTime

FROM Messages

JOIN Users

ON Messages.SenderId = Users.Id)

JOIN GroupChats

ON GroupChatId = GroupChats.Id

**Создание домена:**

CREATE DOMAIN Gender VARCHAR(1) CHECK (VALUE IN ('М', 'Ж'));

ALTER TABLE Users ADD Gender\_temp VARCHAR(1);

UPDATE Users SET Gender\_temp = Gender;

ALTER TABLE Users DROP Gender;

ALTER TABLE Users ADD GivenGender Gender;

UPDATE Users SET GivenGender = Gender\_temp;

ALTER TABLE Users DROP Gender\_temp;

**Создание индекса:**

CREATE INDEX UserNames

ON Users(name);

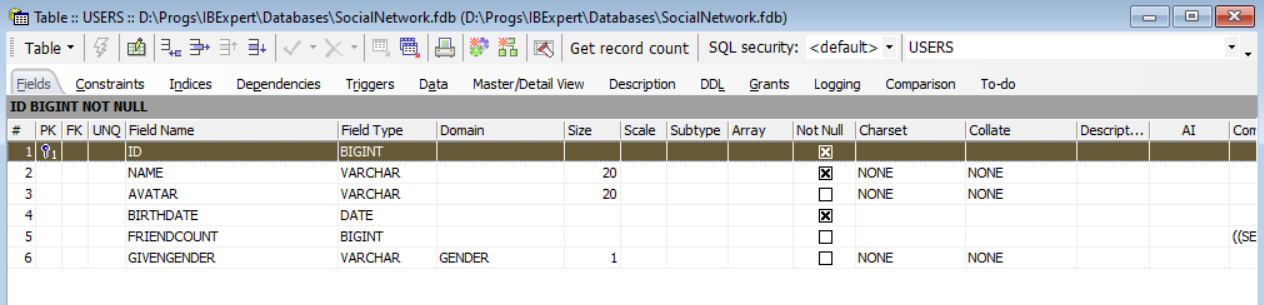


Рис. 2. Пример созданной таблицы

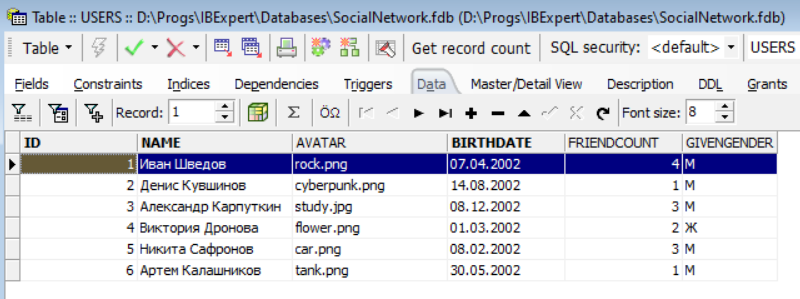


Рис. 3. Пример созданных записей

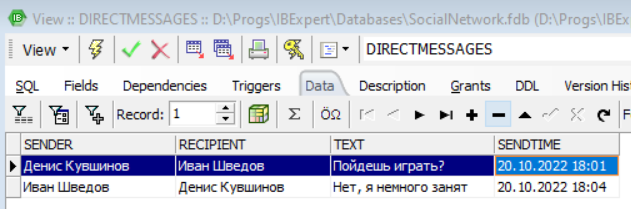


Рис. 4. Представление личных сообщений

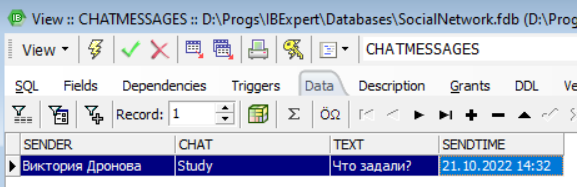


Рис. 5. Представление сообщений чата

**Вывод:** в ходе выполнения лабораторной работы были получены практические навыки использования операторов определения данных, ограничений, просмотров.