

Министерство образования Республики Беларусь
Учреждение образования «Полоцкий государственный университет»

Факультет информационных
технологий
Кафедра технологий
программирования

Лабораторная работа №7 по курсу «Базы данных»

«Создание базы данных в СУБД»
Вариант «Проектный менеджер»

Выполнил

Студент гр.
21-ИТ-1

Проверил

Ланцев Е.Н.
Преподаватель
Дьякова А.С.

Полоцк, 2023г.

Цель работы: Изучить основные понятия нормализации баз данных.

Ход работы

Задание:

1. БД должна содержать минимум 10 таблиц
2. БД должна соответствовать минимум 3 нормальной форме
3. В описании типов данных атрибутов необходимо использовать типы, поддерживаемые СУБД MS SQL Server
4. В описании атрибутов также использовать каждое доступное в MS SQL Server ограничение столбцов как минимум по 2 раза
5. Используемые ограничения столбцов должны выполнять определённую функцию и иметь логический смысл

Дополнительное задание:

1. Определить не менее 3 типов пользователей БД (например, неавторизованные пользователи, сторонние приложения, администраторы)
2. Для каждого типа пользователей определить, к каким таблицам в БД он имеет доступ и в каком объёме (чтение, изменение, добавление, удаление)

Ход решения:

Запрос для создания 10 таблиц: Проекты, Пользователи, Команды, Статусы Проектов, Финансовый отчет, Статусы Задачи, Роли, Команда, Команды Пользователя, Комментарий под задачей:

Данный запрос создает таблицу “Projects”

```
CREATE TABLE IF NOT EXISTS projects (  
  
    id TEXT UNIQUE,  
  
    name TEXT NOT NULL,  
  
    description TEXT,  
  
  
  
    fk_project_manager_id TEXT UNIQUE,  
    FOREIGN KEY (fk_project_manager_id) REFERENCES users(id),  
  
  
  
    fk_project_status_id TEXT,  
    FOREIGN KEY (fk_project_status_id) REFERENCES project_statuses(id),
```

```
PRIMARY KEY (id, fk_project_manager_id);
```

Данный запрос создает таблицу “Tasks”

```
CREATE TABLE IF NOT EXISTS tasks (  
  
id TEXT PRIMARY KEY,  
  
title TEXT NOT NULL,  
  
description TEXT,  
  
  
fk_assignee_id TEXT,  
  
FOREIGN KEY (fk_assignee_id) REFERENCES users(id),  
  
  
  
fk_project_id TEXT,  
  
FOREIGN KEY (fk_project_id) REFERENCES projects(id),  
  
  
  
fk_task_status_id TEXT,  
  
FOREIGN KEY (fk_task_status_id) REFERENCES task_statuses(id)  
  
);
```

Данный запрос создает таблицу “TaskComments”

```
CREATE TABLE IF NOT EXISTS task_comments (  
  
id TEXT PRIMARY KEY,  
  
  
  
content TEXT NOT NULL,  
  
fk_task_id TEXT NOT NULL,  
  
FOREIGN KEY (fk_task_id) REFERENCES tasks(id),  
  
  
  
fk_author_id TEXT NOT NULL,  
  
FOREIGN KEY (fk_author_id) REFERENCES users(id)  
  
);
```

Данный запрос создает таблицу “Users”

```
CREATE TABLE IF NOT EXISTS users (  
  
id TEXT PRIMARY KEY,  
  
name TEXT NOT NULL,  
  
email TEXT  
  
);
```

Данный запрос создает таблицу “ProjectStatuses”

```
CREATE TABLE IF NOT EXISTS project_statuses (  
  
id TEXT PRIMARY KEY,  
  
title TEXT UNIQUE NOT NULL,  
  
description TEXT  
  
);
```

Данный запрос создает таблицу “Roles”

```
CREATE TABLE IF NOT EXISTS roles (  
  
id TEXT PRIMARY KEY,  
  
title TEXT UNIQUE NOT NULL,  
  
description TEXT  
  
);
```

Данный запрос создает таблицу “TaskStatuses”

```
CREATE TABLE IF NOT EXISTS task_statuses (  
  
id TEXT PRIMARY KEY,  
  
title TEXT UNIQUE NOT NULL,  
  
description TEXT,  
  
staging BOOLEAN NOT NULL  
  
);
```

Данный запрос создает таблицу “FinancialReports”

```
CREATE TABLE IF NOT EXISTS financial_reports (  
  
id TEXT UNIQUE,  
  
value INT NOT NULL,  
  
additional_info TEXT,  
  
  
fk_project_id TEXT UNIQUE,  
  
FOREIGN KEY (fk_project_id) REFERENCES projects(id),  
  
  
  
PRIMARY KEY (id, fk_project_id)  
  
);
```

Данный запрос создает таблицу “UserTeams”

```
CREATE TABLE IF NOT EXISTS user_teams (  
  
fk_user_id TEXT,  
  
fk_team_id TEXT,  
  
FOREIGN KEY (fk_user_id) REFERENCES users(id),  
  
FOREIGN KEY (fk_team_id) REFERENCES teams(id),  
  
PRIMARY KEY (fk_user_id, fk_team_id)  
  
);
```

Данный запрос создает пользователя ‘adminaa’

```
REATE USER adminaa WITH PASSWORD 'im_so_sorry';  
  
GRANT ALL PRIVILEGES ON DATABASE lablab6 to adminaa;
```

Данный запрос создает пользователя ‘make_me_happy’ и устанавливает ему доступ к таблице ‘Teams’

```
CREATE USER hell WITH PASSWORD 'make_me_happy';  
  
GRANT ALL PRIVILEGES ON TABLE teams TO hell;
```

Ответы на контрольные вопросы:

1. В чём заключается проектирование базы данных на физическом уровне?

Физический уровень проектирования баз данных - это один из этапов процесса

проектирования баз данных, который фокусируется на определении способов хранения данных на физическом уровне, то есть на уровне, который непосредственно связан с аппаратным обеспечением компьютерной системы и структурой хранения данных на дисках или других устройствах хранения, а также с типом и особенностями используемой СУБД.

2. Что такое СУБД?

СУБД — комплекс программ, позволяющих создать базу данных и манипулировать данными

3. Что такое интегрированная среда разработки?

Интегрированная среда разработки (IDE) – это программное приложение, которое помогает программистам эффективно разрабатывать программный код.

4. Что такое SQL?

Язык SQL (Structured Query Language) для выполнения запросов и манипуляции данными. Декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

5. Для чего используется ограничение IDENTITY?

IDENTITY - это не столько ограничение, сколько свойство столбца. Оно позволяет нам автоматически увеличивать значение при вставке новой строки. Это работает только с числовыми типами данных. Обычно используется для первичных ключей. Например, если в таблице нет записей, то при добавлении новой MS SQL Server автоматически присвоит столбцу IDENTITY номер 1.

6. В чём заключается проектирование на этапе физической организации данных?

Физическое проектирование – это процедура создания описания конкретной реализации БД с описанием структуры хранения данных, методов доступа к данным.

7. Что такое кэширование?

Кэширование - это сохранение какой-либо информации в оперативной памяти сервера баз данных для более быстрого доступа. Например, мы можем кэшировать данные таблиц, которые часто используются. Помимо этого, мы можем кэшировать запросы и результаты запросов. Это помогает быстро производить повторяющиеся запросы к БД.