**ТЕМА 6 ПРОЕКТИРОВАНИЕ И НОРМАЛИЗАЦИЯ БАЗЫ ДАННЫХ (БД)**

1. **Введение в процесс проектирования базы данных (БД)**

Проектирование базы данных (БД) - это процесс определения структуры и организации данных, которые будут храниться и управляться в информационной системе. Этот процесс включает в себя определение таблиц, столбцов, связей между таблицами и другие аспекты, необходимые для эффективного хранения и обработки данных.

Шаги проектирования базы данных:

1. Определение требований: Сначала необходимо понять, какие данные будут храниться в БД и как они будут использоваться. Это включает в себя анализ бизнес-процессов и потребностей пользователей.

2. Создание схемы данных: На этом этапе определяются таблицы, их структура (столбцы и их типы), а

также связи между таблицами.

3. Нормализация: Процесс нормализации позволяет минимизировать избыточность данных и уменьшить возможность ошибок. Он включает в себя разделение данных на более мелкие, связанные таблицы.

|  |  |
| --- | --- |
| 4. | Индексы и ключи: Определяются первичные и внешние ключи для обеспечения целостности данных и ускорения запросов. Также создаются индексы для оптимизации производительности. |

5. Безопасность и доступ: Устанавливаются права доступа к данным для разных пользователей и ролей.

Обеспечивается безопасность данных.

|  |  |
| --- | --- |
| 6. | Оптимизация производительности: Проводится оптимизация запросов, индексов и структуры данных для достижения высокой производительности системы. |

7. Тестирование и оптимизация: База данных подвергается тестированию, и проводится оптимизация на

основе полученных результатов.

8. Резервное копирование и восстановление: Разрабатывается стратегия резервного копирования данных и

восстановления в случае сбоев.

1. **Рекомендации, лучшие и худшие практики по проектированию базы данных (БД)**

Лучшие практики:

1. Соблюдение нормализации: Следуйте нормализации данных, чтобы избегать избыточности и

обеспечивать целостность данных.

2. Использование подходящих типов данных: Выбирайте подходящие типы данных для каждого столбца, чтобы экономить место и обеспечивать точность данных.

3. Использование индексов: Создавайте индексы для часто используемых столбцов, чтобы ускорить

выполнение запросов.

4. Оптимизация запросов: Написание эффективных SQL-запросов с учетом индексов и структуры данных.

5. Управление безопасностью: Устанавливайте строгое управление доступом к данным и обеспечивайте

безопасность данных.

6. Регулярное обслуживание: Проводите регулярное обслуживание и мониторинг базы данных для

выявления проблем и их решения.

Худшие практики:

1. Избыточные данные: Избегайте хранения избыточных данных, так как это может привести к проблемам с целостностью и производительностью.
2. Неоптимизированные запросы: Плохо оптимизированные SQL-запросы могут привести к медленной производительности и перегрузке сервера.

3. Отсутствие резервного копирования: Не забывайте о регулярном создании резервных копий данных для

предотвращения потери информации.

4. Отсутствие безопасности: Не игнорируйте вопросы безопасности, так как это может привести к утечке

данных и нарушениям.

1. **Нормальные формы и нормализация базы данных (БД)**

Нормализация - это процесс организации данных в БД для минимизации избыточности и обеспечения целостности данных. Существует несколько нормальных форм, включая первую, вторую и третью.

Первая нормальная форма (1NF)

1NF требует, чтобы каждый столбец таблицы содержал только атомарные (неделимые) значения, то есть значения не могут быть множествами или массивами. Пример нарушения 1NF:

CREATE TABLE students (

student\_id serial PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

subjects TEXT[]

);

Здесь столбец "subjects" содержит массив значений, что нарушает 1NF. Для исправления этого, можно создать отдельную таблицу для предметов и использовать внешний ключ.

Вторая нормальная форма (2NF)

2NF требует, чтобы каждый столбец, не являющийся ключом, полностью зависел от первичного ключа. Пример нарушения 2NF:

CREATE TABLE orders (

order\_id serial PRIMARY KEY,

customer\_id INT,

product\_id INT,

product\_name VARCHAR

(100),

quantity INT

);

Здесь "product\_name" и "quantity" зависят от "product\_id", а не от первичного ключа "order\_id". Для исправления этого, можно разделить таблицу на две: "orders" и "order\_items".

Третья нормальная форма (3NF)

3NF требует, чтобы каждый столбец, не являющийся ключом, не зависел от других столбцов, не являющихся ключами. Пример нарушения 3NF:

CREATE TABLE employees (

employee\_id serial PRIMARY KEY,

department\_id INT,

department\_name VARCHAR(50)

)

Здесь "department\_name" зависит от "department\_id", но не от первичного ключа "employee\_id". Для исправления этого, можно создать отдельную таблицу "departments".