Лабораторная работа № 2

Тема: Управление базами данных с помощью SQL. Особенности многотабличных запросов. Применение вложенных запросов. Изменение структуры БД с помощью операторов DDL .Создание и изъятия таблиць.Створення и удаления уникальных индексов.

Цель: Применяя SQL-операторы EXISTS , ANY , ALL , SOME научиться структурировать запросы, создавать более понятные запросы. Научиться строить вложенные запросы. Усвоить на практике операции объединения. Применяя DD L операторы CREATE , DROP , ALTER научиться создавать и определять, удалять и изменять объекты БД.

Содержание работы по варианту индивидуального задания :

1. С помощью SQL - запроса вывести данные из двух таблиц, например, о преподавателей и студентов, чьи фамилии находятся между буквами "К" и "С", кроме того отметить, кто студент, а кто преподаватель.
2. Вывести с помощью SQL - запроса информацию из таблиц преподавателей и предметов (связанные между собой), чтобы каждому преподавателю отвечал предмет, который он преподавал.
3. Вывести список оценок, которые получил тот или иной студент (создать SQL - запрос на основе таблиц о студентах, успешность и предметы).
4. Необходимо получить информацию о студенте Петрова-номер в списке и балл успеваемости. Номер находится в таблице о студентах, бал в таблице успеваемости. Примените вложенный запрос.
5. Вывести из таблицы успеваемости с помощью SQL - запроса номера студенческих билетов и код предмета, только если в ней присутствуют отличные оценки.
6. С помощью DDL -оператора создайте таблицу с определенным им ' ям и несколькими полями различных типов и размеров.
7. С помощью DDL -оператора измените структуру существующей таблицы. Добавьте новое поле типа INTEGER , увеличьте размер существующего поля, удалите ненужное поле.
8. Удалите таблицу из своей БД с помощью DDL -оператора.
9. Создайте индекс в таблице БД.
10. Создайте уникальный индекс для таблицы своей БД.
11. Удалите созданный ранее индекс.
12. Создайте таблицу так, чтобы для двух полей нельзя было бы установить неопределенное значение, и установите первичный ключ с помощью необходимого ограничения.

теоретические сведения

Выше рассматривались запросы, которые использовали в качестве источника данных одну таблицу. Однако часто возникает необходимость в выборе информации из нескольких таблиц. Решается с помощью многотабличных запросов. Одним из вариантов такого вывода является объединение результатов нескольких запросов, которые выполняются независимо друг от друга.

Для размещения нескольких запросов вместе и объединения их вывода применяют UNION , объединяющий вывода двух или более SQL запросов в единый набор строк и столбцов. Например, чтобы получить информацию об участниках конференции вузов применяется запрос:

SELECT SFAM , SIMA, SOTCH

FROM STUDENTS

WHERE NOT (ISNULL (KONF))

UNION

SELECT SFAM , SIMA, SO TCH

FROM TEACHERS

WHERE NOT (ISNULL (KONF))

Обратите внимание , отсутствие «;» после первой инструкции означает , что дальше будет еще один или несколько запросов . Если необходимо соединить два или более столбцов учитывается, что они должны быть совместимыми для объединения. Для каждого из запросов необходимо включение равенстве столбцов, в том же порядке, и при этом должна быть совместимость типов. Нельзя применять агрегатные функции в инструкции SELECT запроса на объединение. UNION автоматически исключает дубликаты строк из вывода. Возможно применять константы и выражения в инструкции SELECT с UNION . например :

SELECT " Студент " , SFAM

FROM STUDENTS

UNION

SELECT " Преподаватель " TFAM

FROM TEACHERS ;

Результат рассмотрим на рисунке 1.

рисунок 1

              Важной особенностью запросов SQL является их способность определять связи между таблицами и выводить информацию с них в терминах этих связей. Эти операции называются объединением. В многотабличных запросам, таблицы, которые представлены в виде списка в инструкции FROM , отделяются друг от друга запятыми. Предикат запроса может ссылаться на любой столбец любой таблицы, и может использоваться для связи между ними. Теперь возникает необходимость принятия имен столбцов и таблиц, поскольку в многотабличных запросе возможно возникновение неоднозначности. Допускается создавать запросы, о ' объединяют более двух таблиц. Например, необходимо вывести список оценок, которые выставил тот или иной преподаватель:

SELECT TEACHERS . TFAM , USP . OCENKA

FROM TEACHERS , PREDMET , USP

WHERE TEACHERS . TNUM = PRE DMET . TNUM

AND PREDMET . PNUM = USP . PNUM ;

Запросы способны управлять другими запросами, это происходит, если разместить запрос внутрь другого предиката, который использует вывода внутреннего запроса для установления верного или неверного значения предиката. Например, нужно вывести всю информацию о студенте с фамилией Поляков, если неизвестный его номер. Номер получается из таблицы с данными о студентах, а затем применить результат в таблицу успеваемости таким запросом:

SELECT \*

FROM USP

WHERE SNUM =

(SELECT S NAM

FROM STUDENTS

WHERE SFAM = ' Поляков ');

Чтобы выполнить запрос SQL сначала оценивает внутренний запрос ( его называют подзапросом ) внутри предложения WHERE . Подзапрос должен выбрать только одно поле, а тип данных этого поля должен совпадать с тем значением, с которым он сравнивается в предикатив.

              EXISTS , ANY , ALL , SOME -специальные операторы, которые всегда берут подзапросы в качестве аргументов.

EXISTS -Использует, чтобы указать предикату на то, чтобы выполнять вывода или не выполнять вывода в подзапросе, при этом E XISTS дает в качестве результата значение ИСТИНА или ЛОЖЬ. Например, можно решить, получать данные из таблицы успеваемости, если в ней присутствуют неудовлетворительные оценки. Это реализуется следующим образом:

SELECT \*

FROM USP

WHERE USP . OCENCA = 2

AND EXISTS

( SELECT \*

FROM USP

WHERE USP. OCENCA = 2 )

ANY , ALL и SOME , напоминают EXISTS , но отличаются тем , что применяются вместе с реляционными операторами , аналогично IN в подзапросах . например :

SELECT \*

FROM USP

WHERE OCENCA <> ALL

( SELECT OCENCA

FROM USP

WHERE UDATE = 10/06/20 09);

В этом случае подзапрос выберет все оценки 10.06.2009. После этого основной запрос выведет все записи с оценкой, не совпадает ни с одним из них.

Аналогичный результат можно получить следующим образом:

SELECT \*

FROM USP

WHERE NOT OCENCA = ANY

( SELECT OCENC A

FROM USP

WHERE UDATE = 10 /06/2009)

Если пригласить значение с помощью команды ALL , не равны набора значений, то это тоже самое, что признать факт отсутствия этого значения в наборе.

DDL

Для изменения структуры БД в SQL предусмотрено DDL ( Data Definition Language ) - язык определения данных. С помощью операторов DDL возможно следующее:

* Создать новую базу данных;
* Определить структуру новой таблицы и создать эту таблицу;
* Удалить существующую таблицу;
* Изменить определение существующей таблицы;
* Определить представления данных;
* Обеспечить условия безопасности БД;
* Создать индексы для доступа к таблицам;
* Управлять размещением данных на устройствах хранения.

DDL базируется на трех командах SQL:

* CREATE - позволяет определить и создать о ' объект базы данных;
* DROP - применяется для удаления существующего о ' объекта базы данных;
* ALTER -за помощью которого можно изменить определение о ' объекта БД.

Команда CREATE TABLE - создает пустую таблицу с определенным им ' ям и определенным набором полей, которые указываются в определенном порядке, а также определяются типы полей . К примеру:

CRE ATE TABLE STUDENTS

(SNUM INTEGER,

SFAM CHAR (20),

SIMA CHAR (10),

SOTCH CHAR (15) ) ;

Порядок расположения полей в таблице определяется тем, в какой последовательности они указаны в команде создания таблицы.

Команда ALTER TABLE -средство для изменения определения существующей таблицы, которой можно добавлять поля к существующей таблице ( ADD ) , удалять поля ( DROP ) или изменять их размер ( ALTER ) . ALTER TABLE не действует, если необходимо переопределения. К примеру:

ALTER TABLE STUDENTS

ADD COURS INTEGER,

SPEC CHAR (10);

В таблицу STUDENTS добавляются два поля для хранения информации о курсе и специальность студента.

              Для удаления таблицы используют команду DROP TABLE . Таблица информации не может видалятись.Таблиця удаляется, если она пустая. К примеру:

DROP TABLE STUDENTS ;

Выполняется удаление пустой таблицы STUDENTS .

              Индексом называют упорядоченный список полей или групп полей в таблице. Индексы это полезный инструмент, который широко применяется во всех СУБД . Если создается индекс в поле БД запомнить 'Пятов есть соответствующий порядок всех значений данного поля в области памяти " памяти. Таблица, для которой создается индекс должн а уже существовать и сохранять имена индексированных полей, при этом имя индексов не может быть использовано для чего-то другого в БД, а SQL сам решает когда он необходим для работы и пользуется им автоматически. Например команда для создания индекса по полю, сохраняет фамилию студента:

              CREATE INDEX SFAMIDX ON STUDENTS ( SFAM )

              Для создания уникальных индексов используется ключевое слово UNIQUE в команде CREATE INDEX . Фактически такой индекс будет первичным ключом таблицы.

Создать уникальные индексы можно так:

              CREATE UNIQUE INDEX

              SNAMIDX ON STUDENTS ( SNAM )

Эта команда не выполнится, если в поле SNAM случатся неуникальные значения .

Для удаления созданного индекса по фамилии студента, можно воспользоваться командой:

DROP INDEX SFAMIDX

ON STUDENTS ;

              Ограничение данных - это часть определений таблицы, которая ограничивает значения, допускаются к вводу в поля таблицы. Ограничения можно указывать, когда создается меняется таблица. Существуют два основных вида ограничений. Ограничение поля и ограничения таблицы. Ограничение поля ставят в конец фрагмента команды, которая объявляет его имя 'я п и сля типа данных. Ограничение таблицы ставят в конец объявления имени таблицы. NOT NULL - предохраняет поле от пустых значений.

              Есть возможность установить уникальность в качестве ограничения столбца с помощью ключевого слова UNIQUE . А с помощью ограничения PRIMARE KEY возможно ограничивать таблицу или отдельные столбцы таблицы. Синтаксис и определения его уникальности такие же которые UNIQUE - первичные ключи не допускают NULL значений, поэтому перед такими ограничениями необходимо объявить NOT NULL . К примеру:

              CREATE TABLE YSP

              (UNUM INTEGER NOT NULL PRIMARY KEY,

              OCENKA INTEGER,

              UDATE DATE,

              SNUM INTEGER NOT NULL,

              PNUM INTEGER NOT NULL,

              UNIQUE (SNUM, PNUM))

Как видно, лучше накладывать ограничения на поле, которое создает уникальный идентификатор строки, и сохранить ограничения UNIQUE для поля или группы полей, которые должны быть уникальными логично, а не применяться для идентификации строк.

Контрольные вопросы к лабораторной работе 2:

1. Запросы называют многотабличных зап тами?
2. Что значит UNION в инструкции SELECT ?
3. Какие функции н е можно применять в инструкции SELECT запроса на объединение ?
4. Разрешается создавать запросы, которые об ' объединяют более двух таблиц?
5. Запрос называют подзапросом?
6. Условия накладываются на значение получаемого в подзапросе?
7. Как отделяются друг от друга таблицы в многотабличных запросах, которые представлены в виде списка в инструкции FROM ?
8. Действия испол уються сначала для запроса, который использует вывода внутреннего запроса ?
9. Когда возникает необходимость принятия имен столбцов с именем таблицы через точку в запросах?
10. Какие специальные операторы всегда берут подзапросы в качестве аргументов ?
11. В каких случаях используют оператор EXISTS ?
12. Чем отличаются операторы ANY , ALL и SOME от EXISTS ?
13. Какие возможности DDL ?
14. На каких командах базируется DDL ?
15. С помощью какой команды можно создать таблицу?
16. С помощью какой команды можно удалить таблицу?
17. С помощью какой команды можно изменить определение о ' объекта БД?
18. Что значит команда CREATE TABLE ?
19. Что значит команда ALTER TABLE ?
20. Что значит команда DROP TABLE ?
21. Действия над таблицей необходимо выполнить перед тем как удалить ее?
22. Что называют индексом?
23. Что означает ограничение данных?
24. Когда нельзя создать уникальный индекс в таблице с данными?
25. Что означает ограничение NOT NULL в конце объявления имени таблицы?