**Лабораторные работы по курсу**

**Базы данных**

**Лабораторная работа 5**

**«Операторы модификации базы данных»**

**Москва, 2024**

Оглавление

[1. Теоретическая часть 3](#_Toc144807930)

[1.1. Добавление данных в таблицу 3](#_Toc144807931)

[1.2. Изменение значений 4](#_Toc144807932)

[1.3. Удаление значений 4](#_Toc144807933)

[2. Практическая часть 5](#_Toc144807934)

[2.1. Задание 1. 5](#_Toc144807935)

[2.2. Задание 2. 5](#_Toc144807936)

[2.3. Задание 3. 5](#_Toc144807937)

[2.4. Задание 4. 5](#_Toc144807938)

[2.5. Задание 5. 5](#_Toc144807939)

[Список литературы 5](#_Toc144807940)

1. Теоретическая часть

В предыдущей лабораторной работе рассматривались вопросы, связанные с проектированием баз данных. После выполнения оператора CREATE TABLE создается таблица, не содержащая никаких данных. На следующем этапе ее необходимо заполнить. В данной лабораторной работе обсуждаются темы, связанные с изменением данных в таблицах – вставкой, обновлением и удалением записей.

* 1. Добавление данных в таблицу

Для добавления данных в таблицу существует оператор SQL INSERT. Его сокращенный синтаксис представлен ниже.

INSERT INTO *имя\_таблицы* [ ( *имя\_столбца1* [, ...] ) ]

{ DEFAULT VALUES | VALUES ( { *выражение1* | DEFAULT }[, ...] ) [, ...] |*запрос* }

Данный оператор добавляет строки в таблицу. С помощью INSERT возможно добавить одну или несколько строк, указанных напрямую, либо ноль и более строк, возвращенных с помощью дополнительного запроса.

Параметры:

*имя\_таблицы –* таблица, в которую добавляются данные

*имя\_столбца1,…-* список столбцов в таблице, которые заполняются данными из списка выражений записанного послеVALUES. Имя столбцов можно не указывать, тогда таблица начнет заполняться с крайне левого столбца и далее по порядку, пока не закончится список выражений. В случае если указаны только некоторые столбцы, остальные будут заполнены выражениями по умолчанию или NULL.

DEFAULT VALUES - все столбцы таблицы получат значение по умолчанию,

VALUES ( *выражение1,…*) – значение выражения которое будет присвоено заданным столбцам ( *имя\_столбца1,…).* Тип столбца должен соответствовать типу записываемого в него выражения.

VALUES DEFAULT – столбцу будет присвоено выражение по умолчанию. Аналогичное действие происходит при отсутствии имени столбца в списке ( *имя\_столбца1,…),* однако для наглядности часто используют именно этот формат.

*Запрос –* запрос (оператор SELECT), который формирует строки для добавления в таблицу.

Например, добавим в таблицу *Students\_groups* восемь записей.

INSERT INTO Students\_group (students\_group\_number, enrolment\_status, structural\_unit\_number) VALUES

('ИВТ-41', 'Очная', 1),

('ИВТ-42', 'Очная', 1),

('ИВТ-43', 'Очная', 1),

('ИВТ-21В', 'Заочная', 1),

('ИБ-21', 'Очная', 3),

('ИТД-31', 'Очная', 4),

('ИТД-32', 'Очная', 4),

('ИТД-33', 'Очная', 4);

В качестве строк для добавления в таблицу могут быть использованы значения, сформированные в результате запроса к другой таблице. Для иллюстрации приведем следующий пример. Создадим ещё одну таблицу, которая будет содержать информацию о должниках по предметам (студентам, имеющим хотя бы одну оценку 2).

Для этого создадим еще таблицу *Debtor\_students:*

CREATE TABLE debtor\_students

(

id SERIAL PRIMARY KEY,

surname VARCHAR(30) NOT NULL,

name VARCHAR(30) NOT NULL,

patronymic VARCHAR(30) NULL,

group\_id VARCHAR(7) NOT NULL,

debt\_number INTEGER NOT NULL

)

Данная таблица будет хранить информацию о ФИО и группе студента, а также о числе его долгов.

Для заполнения таблицы составим SQL запрос.

INSERT INTO debtor\_students (surname, name, patronymic, students\_group\_number, debt\_number)

(

SELECT surname, name, patronymic, students\_group\_number, COUNT(\*) AS "Number of debts"

FROM student

INNER JOIN field\_comprehension ON field\_comprehension.student\_id = student.student\_id

WHERE field\_comprehension.mark = 2

GROUP BY surname, name, patronymic, students\_group\_number

);

Данный запрос выбирает всех студентов, которые имеют хотя бы одну оценку 2, и записывает данные значения в созданную таблицу с должниками.

* 1. Изменение значений

Для изменения существующих значений в базе данных существует оператор UPDATE.

UPDATE *имя\_таблицы*

SET { *имя\_столбца* = { *выражение* | DEFAULT } |

( *имя\_столбца* [, ...] ) = ( { *выражение* | DEFAULT } [, ...] ) |

( *имя\_столбца* [, ...] ) = ( *вложенный\_SELECT* )

} [, ...]

[ FROM *список\_FROM* ]

[ WHERE *условие* ]

После оператора UPDATE указывается целевая таблица, которая должна быть модифицирована. В предложении SET указывается, какие столбцы в выбранных строках таблицы должны быть обновлены, и для них задаются новые значения. Остальные столбцы сохраняют свои предыдущие значения. Возможно изменение строк таблицы на значения, из другой таблицы. Для этого в параметре FROM указывается её название, с помощью ключевого слова WHERE, возможно указать условие, при котором произойдет выборка.

Например, составим запрос, убирающий один долг у всех студентов групп ИВТ.

UPDATE debtor\_students

SET debt\_number = debt\_number - 1

WHERE students\_group\_number LIKE 'ИВТ%';

* 1. Удаление значений

Для удаления значений из базы данных существует оператор DELETE.

DELETE FROM *имя\_таблицы* [ \* ] [ [ AS ] *псевдоним* ]

[ WHERE *условие* ]

[ RETURNING \* ]

Оператор DELETE удаляет из указанной таблицы строки, удовлетворяющие условию WHERE. Если предложение WHERE отсутствует, она удаляет из таблицы все строки.

Составим запрос на удаление из списка должников всех студентов, не имеющих задолженностей (число долгов = 0).

DELETE FROM debtor\_students

WHERE debt\_number = 0;

В случае отсутствия оператора WHERE из таблицы будут удалены все значения. Аналогичного результата возможно добиться, используя оператор TRUNCATE. Этот оператор работает гораздо быстрее безусловной команды DELETE и полезна для больших таблиц.

TRUNCATE debtor\_students;

Для вывода на экран списка удаленных строк возможно использовать параметр RETURNING \*;

1. Практическая часть

Вариант выбирается в соответствии с формулой: N = (Nв списке mod 10) + 1.

* 1. Задание 1.

Напишите запрос в соответствии с вашим вариантом

|  |  |
| --- | --- |
| № | Условие запроса |
| 1 | Переведите всех учащихся 3-го курса на 4-й, изменив номер группы |
| 2 | Предположим, что одна из групп была расформирована. Составьте запрос таким образом, чтобы оставшиеся студенты были распределены по другим группам потока. |
| 3 | Уволился преподаватель, имеющий наибольший оклад. Распределите его оклад поровну между преподавателями, у которых такая же должность. Уволившемуся преподавателю поставьте оклад 0. Если преподавателей, имеющих наибольший оклад несколько выберите человека с минимальным кодом преподавателя. |
| 4 | Составьте запрос, увеличивающий значение стажа преподавателя на 1 год, для тех, чья ставка меньше 0.25. |
| 5 | Увеличьте зарплату вдвое у тех преподавателей, которые преподают больше 6 дисциплин. |
| 6 | Поставьте “!” перед фамилией студента, у которого долгов более 12 |
| 7 | Появилась ставка доцента. Переведите старшего преподавателя имеющего наибольший стаж на должность доцента. |
| 8 | Поставьте “!” перед фамилией преподавателя если средний бал по всем дисциплинам, которые он преподает меньше среднего балла по всему структурному подразделению к которому он прикреплен. |
| 9 | Пересчитайте ставки преподавателей в соответствии с средним значением (округлите до целого) ЗЕТ дисциплин, которые он ведет, из расчета, что полная ставка соответствует 6 ЗЕТ. Значение ставки округлите до сотых. |
| 10 | Рассчитайте реальные оклады преподавателей, с учетом, что полная ставка ассистента 25000 у.е., старшего преподавателя 50000 у.е., доцента 75000 у.е., профессора 100000 у.е. |

* 1. Задание 2.

Напишите запрос в соответствии с вашим вариантом

|  |  |
| --- | --- |
| № | Условие запроса |
| 1 | Заполните поле, содержащее цвет студенческого билета следующим образом: студенты с четными номерами (кроме оканчивающихся на 0) получают синий студенческий, нечетными – белый, оканчивающиеся на 0 – зеленый. |
| 2 | Заблокируйте пропуск всем студентам, имеющим больше 4 долгов. |
| 3 | Автоматически заполните поле «Пол». *Подсказка*: обратите внимание на окончание отчества и фамилий студентов. Учтите отсутствие отчеств у ряда студентов. |
| 4 | Автоматически заполните поле «Паспорт» в формате, заданном в предыдущей лабораторной работе (XX-XXXXXX, где X – цифра). Первые две цифры паспорта – дата рождения студента, последние восемь – номер студенческого билета. В случае если дата состоит из одной цифры, добавьте после нее 0 (5 🡪 50). |
| 5 | Автоматически заполните поле «СНИЛС» в формате, заданном в предыдущей лабораторной работе (XXX-XXX-XXX X, где X – цифра). В качестве первых трех цифр возьмите число и месяц рождения студента (если они составят строку меньше трех символов, то добавить 0, если больше – обрезать). Следующие шесть взять из номера студенческого билета, разбив его по три цифры через тире. Последнюю цифру взять сгенерировать рандомно. |
| 6 | Автоматически заполните поле «ИНН» в формате, заданном в предыдущей лабораторной работе (XXXXAAAAAABB, где все указанные символы – цифры).  XXXX – за код налогового органа возьмите год рождения студента, следующие шесть цифр (AAAAAA) – номер студенческого билета. Две последние контрольные цифры вычислите по правилу:  Вычислить 1-ю контрольную цифру:   * Вычислить сумму произведений цифр ИНН (с 1-й по 10-ю) на следующие коэффициенты — 7, 2, 4, 10, 3, 5, 9, 4, 6, 8 (т.е. 7 \* ИНН[1] + 2 \* ИНН[2] + ...). * Вычислить младший разряд остатка от деления полученной суммы на 11.   Вычислить 2-ю контрольную цифру:   * Вычислить сумму произведений цифр ИНН (с 1-й по 11-ю) на следующие коэффициенты — 3, 7, 2, 4, 10, 3, 5, 9, 4, 6, 8 (т.е. 3 \* ИНН[1] + 7 \* ИНН[2] + ...). * Вычислить младший разряд остатка от деления полученной суммы на 11. |
| 7 | Автоматически заполните поле телефон преподавателя в формате, заданном в предыдущей лабораторной работе (+7(XXX)XXX–XX–XX). Укажите код оператора 903, в случае четного порядкового номера преподавателя в таблице *professors*, иначе – 967. Следующие три цифры номера – первые три цифры кода преподавателя. Предпоследнюю пару цифр сгенерируйте рандомно, вместо последней используйте порядковый номер преподавателя в таблице *professors,* отсортированной по фамилии. Если порядковый номер <10 , добавьте вначале 0 (5 🡪 05). |
| 8 | Автоматически заполните поле телефон студента в формате, заданном в предыдущей лабораторной работе (8(XXX)XXX–XX–XX). Укажите код оператора 903, в случае четного студенческого билета, иначе – 967. Следующие три цифры номера сгенерируйте рандомно. Предпоследняя пара цифр – номер группы, последняя – количество человек в группе, в которой учится студент. |
| 9 | Автоматически заполните поле, содержащее номер аудитории подразделения в формате, установленном в предыдущей лабораторной работе (AAXX, где AA могут принимать значения 29, 25, 14, 33, 87). Выбор первых двух цифр аудитории организуйте рандомно из списка допустимых значений. Последние две цифры – количество букв полном названии подразделения. |
| 10 | Назначьте случайным образом куратора каждой группе. Для этого в соответствующее поле таблицы *students\_group* добавленное в предыдущей лабораторной работе запишите код преподавателя, выбранного случайным образом из таблицы *professors*. |

* 1. Задание 3.

В зависимости от варианта, добавьте значения в исправленную вами базу данных в прошлой лабораторной работе. В каждую из таблиц необходимо добавить не менее 10 осмысленных значений.

Контрольные вопросы

1. Какие ключевые слова возможно использовать внутри оператора INSERT?
2. Каким образом возможно обновить сразу несколько значений
3. Возможно ли использовать операции объединения в операторе INSERT? Если да, то приведите пример.
4. Возможно ли изменять порядок атрибутов при вводе данных в таблицу?
5. В чем отличие между операторами DELETE и TRUNCATE?

Список использованной литературы

|  |  |
| --- | --- |
| [1] | Е. П. Моргунов, PostgreSQL. Основы языка SQL, 1-е ред., Санкт-Петербург: БХВ-Петербург, 2018, p. 336. |
| [2] | «PL/pgSQL — процедурный язык SQL,» [В Интернете]. Available: https://postgrespro.ru/docs/postgresql/15/plpgsql. [Дата обращения: 09 03 2023]. |
| [3] | «Исходный код СУБД postgres,» [В Интернете]. Available: https://github.com/postgres/postgres. [Дата обращения: 30 01 2023]. |
| [4] | Документация к PostgreSQL 15.1, 2022. |
| [5] | Е. Рогов, PostgreSQL изнутри, 1-е ред., Москва: ДМК Пресс, 2023, p. 662 . |
| [6] | Б. А. Новиков, Е. А. Горшкова и Н. Г. Графеева, Основы технологии баз данных, 2-е ред., Москва: ДМК пресс, 2020, p. 582. |