

SQL

Манипулирование данными



АКАДЕМИЯ

Рассматриваемые вопросы

Часть	Темы
1	Описание команд манипулирования данными DML
2	Вставка строк в таблицу
3	Обновление строк в таблице
4	Удаление строк из таблицы
5	Управление транзакциями

Язык манипулирования данными (DML)

- Команды DML выполняются при следующих операциях:
 - Вставка новых строк в таблицу
 - Изменение существующих строк в таблице
 - Удаление существующих строк из таблицы
- Транзакция – это совокупность команд DML, образующих логическую единицу работы.

Синтаксис команды INSERT

- Для вставки новых строк в таблицу используется команда INSERT:

INSERT INTO таблица [(столбец [, столбец ...])]

VALUES (значение [, значение ...])

- Этот синтаксис позволяет вставлять в таблицу только по одной строке

```
INSERT INTO students (student_id, surname, name, kurs)
```

```
VALUES (30, 'Иванов', 'Иван', 5);
```

- Символьные значения и даты заключаются в апострофы

Создание скрипта

- Использование в команде SQL переменных подстановки с одним амперсантом (&) для ввода запрашиваемых у пользователя значений.
- Амперсант показывает, куда вставляется значение переменной

```
INSERT INTO students (student_id, surname, name, kurs)  
VALUES (&student_id, &surname, &name, &kurs);
```

Команды с переменными можно сохранить в файле для последующего использования. После запуска скрипт-файла пользователю выдается приглашение на ввод значений переменных подстановки с одним амперсантом.

Этот скрипт-файл может быть выполнен неоднократно, и при каждом его выполнении могут ыть введены различные значения.

Копирование строк из другой таблицы

- Команда INSERT должна включать подзапрос
- Предложение VALUES не используется
- Количество столбцов, указанных в предложении INSERT, должно совпадать с количеством столбцов в подзапросе.

```
INSERT INTO students (student_id, surname, name, kurs)
```

```
SELECT student_id, surname, name, kurs
```

```
FROM students
```

```
WHERE kurs=3;
```

Синтаксис команды UPDATE

- Для обновления существующих строк используется команда UPDATE

UPDATE таблица

SET столбец1=значение1 [, столбец2=значение2, ...]

[WHERE условие];

- В случае необходимости можно одновременно обновлять несколько строк

Предложение WHERE позволяет изменить конкретную строку или строки (те, которые удовлетворяют условию)

Если предложение WHERE отсутствует, обновятся все строки таблицы

Обновление нескольких столбцов с помощью подзапроса

Изменение стипендии и курса служащего с идентификатором 25, чтобы они стали такими же, как у студента с идентификатором 40.

UPDATE student

SET stipend =(select stipend from student where student_id=40),

 kurs = (select kurs from student where student_id=40)

WHERE student_id=25;

Обновление строк на основе значений из другой таблицы

Для изменения строк таблицы на основе значений из другой таблицы используйте подзапросы в командах UPDATE

```
UPDATE copy_emp  
SET department_id = (select department_id  
                        from employees  
                        where employee_id = 100)  
WHERE job_id = (select job_id  
                from employees  
                where employee_id = 200);
```

Таблица COPY_EMP обновляется на основе значений из таблицы EMPLOYEES. Номер отдела всех служащих, имеющих такую же должность как служащий под номером 200, меняется на текущий номер отдела служащего под номером 100.

Команда DELETE

Для удаления строк используется команда DELETE

```
DELETE [FROM] таблица  
[WHERE      условие];
```

Удаление строк из таблицы

- Конкретная строка или строки удаляются с помощью предложения WHERE

DELETE FROM student

WHERE student_id = 30;

Удаляется 1 строка, где идентификатор студента равен 30

- Если предложение WHERE отсутствует, удаляются все строки таблицы

DELETE FROM student;

Удаляются все строки таблицы student

Удаление строк на основе значений из другой таблицы

Для удаления строк на основе значений из другой таблицы используйте подзапросы в командах DELETE

```
DELETE FROM student
```

```
WHERE univ_id in (select univ_id
```

```
from university
```

```
where rating = 400);
```

В данном примере удаляются записи о студентах из таблицы student, рейтинг университета которых равен 400

Команда TRUNCATE

- Удаляет все строки из таблицы, оставляя таблицу пустой с сохранением структуры таблицы
- Так как это команда языка определения данных (DDL), а не языка DML, ее действие нельзя отменить простым образом
- Синтаксис:

```
TRUNCATE TABLE table_name;
```

- Пример:

```
TRUNCATE TABLE student;
```

Транзакции базы данных

Содержат что-либо из следующего:

- Команды DML, выполняющие единое согласованное изменение данных
- Одну команду DDL
- Одну команду DCL

Транзакции базы данных

- Начинаются с выполнения первой исполняемой команды SQL
- Заканчиваются одним из следующих событий:
 - Выполнение команды COMMIT или ROLBACK
 - Выполнением команды DDL или DCL (автоматическая фиксация транзакции)
 - Разрывом соединения пользователя с базой данных
 - Отказы сисетмы

Преимущества команд COMMIT и ROLLBACK

- Обеспечивают согласованность данных.
- Позволяют проверить изменения в данных прежде, чем сделать их постоянными.
- Логически группируют взаимосвязанные операции

Команды COMMIT и ROLLBACK позволяют управлять процессом изменений данных перед тем, как сделать их постоянными.

Откат изменений до маркера

- С помощью команды SAVEPOINT можно создать маркер в текущей транзакции.
- Выполнить откат изменений к маркеру, используя команду ROLLBACK TO savepoint_name.

Пример:

...

...

UPDATE ...

SAVEPOINT update_done;

...

...

INSERT...

ROLLBACK TO update_done;

Если создаётся точка сохранения с таким же именем, предыдущая уничтожается.

Неявная обработка транзакций

- Автоматическая фиксация изменений (COMMIT) происходит в следующих случаях:
 - Выполнение команды DDL
 - Выполнение команды DCL
 - Нормальное прекращение сеанса работы с базой данных без явной отправки команды COMMIT или ROLLBACK
- Автоматический откат (ROLLBACK) выполняется в случае аварийного прекращения сеанса работы или отказа системы

Итоги

В этой лекции вы познакомились с тем, как использовать следующие команды:

Команда	Описание
INSERT	Вставляет новые строки в таблицу
UPDATE	Изменяет существующие строки таблицы
DELETE	Удаляет существующие строки из таблицы
COMMIT	Делает все незафиксированные изменения постоянными
SAVEPOINT	Позволяет произвести откат до определённой точки сохранения
ROLLBACK	Отменяет все незафиксированные изменения данных

Практическое занятие



АКАДЕМИЯ



Задания

1. Создайте таблицу MY_EMPLOYEE запросом "CREATE TABLE MY_EMPLOYEE AS SELECT * FROM employees where 1=2;"
2. SQL запросами добавьте в таблицу MY_EMPLOYEE 5 записей о пяти любых сотрудниках, значения на ваше усмотрение.
3. Добавьте в таблицу MY_EMPLOYEE всех сотрудников из таблицы EMPLOYEES зарплата (salary) которых больше 7000
4. Создайте скрипт-файл для интерактивной вставки строк в таблицу MY_EMPLOYEE где email будет автоматически рассчитываться и равняться первому символу имени и фамилии. Например 'AIVANOV'
5. Удалите из таблицы MY_EMPLOYEE записи о сотрудниках имена которых начинаются на буквы 'A' 'B' 'C'
6. Установите зарплату 10 000 сотрудникам которые получают менее 10 000
7. Зафиксируйте изменения
8. Используя SQVEPOINT напишите набор скриптов которые позволят вставить запись о Иване Иванове затем удалить все записи из таблицы MY_EMPLOYEE кроме вставленной записи о Иване Иванове.
9. Зафиксируйте изменения.
10. Удалите все записи из таблицы MY_EMPLOYEE
11. Зафиксируйте изменения
12. Написанные Вами команды пришлите на проверку по адресу aspostnikov@phoenixit.ru

