

ВВОД, УДАЛЕНИЕ И
ИЗМЕНЕНИЕ ЗНАЧЕНИЙ
ПОЛЕЙ

КОМАНДЫ МОДИФИКАЦИИ ЯЗЫКА DML

- Значения могут быть помещены и удалены из полей, тремя командами языка DML (Язык Манипулирования Данными):
- INSERT (ВСТАВИТЬ),
- UPDATE (МОДИФИЦИРОВАТЬ),
- DELETE (УДАЛИТЬ).
- Не смущайтесь, все они упоминались ранее в SQL, как команды модификации.

ВВОД ЗНАЧЕНИЙ

- Все строки в SQL вводятся с использованием команды модификации INSERT.

В самой простой форме,

- INSERT использует следующий синтаксис:
- INSERT INTO
- VALUES (, . . .);

- Так, например, чтобы ввести строку в таблицу Продавцов, вы можете использовать следующее условие:
- INSERT INTO Salespeople
- VALUES (1001, 'Peel', 'London', .12);

- Команды DML не производят никакого вывода, но ваша программа должна дать вам некоторое подтверждение того, что данные были использованы.
- Имя таблицы (в нашем случае — Salespeople (Продавцы)), должно быть предварительно определено в команде CREATE TABLE (см. Главу 17), а каждое значение пронумерованное в предложении значений, должно совпадать с типом данных столбца, в который оно вставляется. В ANSI, эти значения не могут составлять выражений, что означает что 3 — это доступно, а выражение $2+1$ — нет.
- Значения, конечно же, вводятся в таблицу в поименном порядке, поэтому первое значение с именем, автоматически попадает в столбец 1, второе в столбец 2, и так далее.

ВСТАВКА ПУСТЫХ УКАЗАТЕЛЕЙ (NULL)

- Если вам нужно ввести пустое значение (NULL), вы вводите его точно также как и обычное значение.
- Предположим, что еще не имелось поля city для мистера Peel. Вы можете вставить его строку со значением=NULL в это поле, следующим образом:
- INSERT INTO Salespeople
- VALUES (1001, 'Peel', NULL, .12);
- Так как значение NULL — это специальный маркер, а не просто символьное значение, он не включается в одиночные кавычки

ИМЕНОВАНИЕ СТОЛБЦА ДЛЯ ВСТАВКИ (INSERT)

- Вы можете также указывать столбцы, куда вы хотите вставить значение имени.
- Это позволяет вам вставлять имена в любом порядке.
- Предположим, что вы берете значения для таблицы Заказчиков из отчета, выводимого на принтер, который помещает их в таком порядке: city, cname, и cnum, и для упрощения, вы хотите ввести значения в том же порядке:
- `INSERT INTO Customers (city, cname, cnum)`
- `VALUES ('London', 'Honman', 2001);`

- Обратите внимание, что столбцы rating и snum — отсутствуют.
- Это значит, что эти строки автоматически установлены в значение по умолчанию.
- По умолчанию может быть введено или значение NULL или другое значение, определяемое как значение по умолчанию.
- Если ограничение запрещает использование значения NULL в данном столбце, и этот столбец не установлен как по умолчанию, этот столбец должен быть обеспечен значением для любой команды INSERT, которая относится к таблице (смотри Главу 18 для информации об ограничениях на NULL и на "по умолчанию").

ВСТАВКА РЕЗУЛЬТАТОВ ЗАПРОСА

- Вы можете также использовать команду INSERT чтобы получать или выбирать значения из одной таблицы и помещать их в другую, чтобы использовать их вместе с запросом.
- Чтобы сделать это, вы просто заменяете предложение VALUES (из предыдущего примера) на соответствующий запрос:
- INSERT INTO Londonstaff SELECT *
- FROM Salespeople
- WHERE city = 'London';

- Здесь выбираются все значения произведенные запросом — то есть все строки из таблицы Продавцов со значениями `city = "London"` — и помещаются в таблицу на- зываемую `Londonstaff`.
- Чтобы это работало, таблица `Londonstaff` должна отвечать следующим условиям: *
Она должна уже быть создана командой `CREATE TABLE`. *
- Она должна иметь четыре столбца которые совпадают с таблицей Продавцов в терминах типа данных; то есть первый, второй, и так далее, столбцы каждой таблицы, должны иметь одинаковый тип данных (причем они не должны иметь одинаковых имен).

- Общее правило то, что всталяемые столбцы таблицы, должны совпадать со столбцами выводимыми подзапросом, в данном случае, для всей таблицы Продавцов.
- Londonstaff — это теперь независимая таблица, которая получила некоторые значения из таблицы Продавцов (Salespeople).
- Если значения в таблице Продавцов будут вдруг изменены, это никак не отразится на таблице Londonstaff (хотя вы могли бы создать такой эффект с помощью Представления (VIEW))

- Так как или запрос или команда INSERT могут указывать столбцы по имени, вы можете, если захотите, переместить только выбранные столбцы, а также переупорядочить только те столбцы, которые вы выбрали.
- Предположим, например, что вы решили сформировать новую таблицу с именем Daytotals, которая просто будет следить за общим количеством долларов сумм приобретений упорядоченных на каждый день. Вы можете ввести эти данные независимо от таблицы Заказов, но сначала вы должны заполнить таблицу Daytotals информацией ранее представленной в таблице Заказов.
- Понимая, что таблица Заказов охватывает последний финансовый год, а не только несколько дней, как в нашем примере, вы можете видеть преимущество использования следующего условия INSERT в подсчете и вводе значений

- INSERT INTO Daytotals (date, total)
- SELECT odate, SUM (amt)
- FROM Orders
- GROUP BY odate;

- Обратите внимание что, как предложено ранее, имена столбцов таблицы Заказов и таблицы Daytotals — не должны быть одинаковыми.
- Кроме того, если дата приобретения и общее количество — это единственные столбцы в таблице, и они находятся в данном порядке, их имена могут быть исключены из вывода из-за их очевидной простоты.

УДАЛЕНИЕ СТРОК ИЗ ТАБЛИЦ

- Вы можете удалять строки из таблицы командой модификации — DELETE. Она может удалять только введённые строки, а не индивидуальные значения полей, так что параметр поля является необязательным или недоступным.
- Чтобы удалить все содержание таблицы Продавцов, вы можете ввести следующее условие:
- DELETE FROM Salespeople;

- Теперь, когда таблица пуста, ее можно окончательно удалить командой DROP TABLE.
- Обычно, вам нужно удалить только некоторые определенные строки из таблицы.
- Чтобы определить какие строки будут удалены, вы используете предикат, так же как вы это делали для запросов. Например, чтобы удалить продавца Axelrod из таблицы, вы можете ввести
- DELETE FROM Salespeople
- WHERE snum = 1003;

- Мы использовали поле `snm` вместо поля `sname` потому, что это лучшая тактика при использовании первичных ключей когда вы хотите чтобы действию подвергалась одна и только одна строка.
- Для вас — это аналогично действию первичного ключа. Конечно, вы можете также использовать `DELETE` с предикатом, который бы выбирал группу строк, как показано в этом примере:
 - `DELETE FROM Salespeople`
 - `WHERE city = 'London';`

ИЗМЕНЕНИЕ ЗНАЧЕНИЙ ПОЛЯ

- Теперь, когда вы уже можете вводить и удалять строки таблицы, вы должны узнать, как изменять некоторые или все значения в существующей строке.
- Это выполняется командой UPDATE. Эта команда содержит предложение UPDATE, в котором указано имя используемой таблицы и предложение SET, которое указывает на изменение, которое нужно сделать для определенного столбца.
- Например, чтобы изменить оценки всех заказчиков на 200, вы можете ввести
- UPDATE Customers
- SET rating = 200;

МОДИФИЦИРОВАНИЕ ТОЛЬКО ОПРЕДЕЛЕННЫХ СТРОК

- Конечно, вы не всегда захотите указывать все строки таблицы для изменения единственного значения, так что UPDATE, наподобие DELETE, может брать предикаты.
- Вот как например можно выполнить изменение одинаковое для всех заказчиков продавца Peel (имеющего snum=1001):
 - UPDATE Customers
 - SET rating = 200

КОМАНДА UPDATE ДЛЯ МНОГИХ СТОЛБЦОВ

- Однако, вы не должны ограничивать себя модифицированием единственного столбца с помощью команды UPDATE.
- Предложение SET может назначать любое число столбцов, отделяемых запятыми. Все указанные назначения могут быть сделаны для любой табличной строки, но только для одной в каждый момент времени.
- Предположим, что продавец Motika ушел на пенсию, и мы хотим переназначить его номер новому продавцу:
- UPDATE Salespeople
- SET sname = 'Gibson', city = 'Boston', comm = .10\
- WHERE snum = 1004;

- Эта команда передаст новому продавцу Gibson, всех текущих заказчиков бывшего продавца Motika и порядки, в том виде в котором они были скомпонованы для Motika с помощью поля snum.
- Вы не можете, однако, модифицировать сразу много таблиц в одной команде, частично потому, что вы не можете использовать префиксы таблицы со столбцами измененными предложением SET.
- Другими словами, вы не можете сказать — "SET Salespeople.sname = Gibson" в команде UPDATE, вы можете сказать только так — "SET sname = Gibson".

ИСПОЛЬЗОВАНИЕ ВЫРАЖЕНИЙ ДЛЯ МОДИФИКАЦИИ

- Вы можете использовать скалярные выражения в предложении SET команды UPDATE, однако, включив его в выражение поля которое будет изменено. В этом их отличие от предложения VALU
- ES команды INSERT, в котором выражения не могут использоваться; это свойство скалярных выражений — весьма полезная особенность.
- Предположим, что вы решили удвоить комиссионные всем вашим продавцам. Вы можете использовать следующее выражение:
- UPDATE Salespeople
- SET comm = comm * 2;

- Всякий раз, когда вы ссылаетесь к указанному значению столбца в предложении SET, произведенное значение может получиться из текущей строки, прежде в ней будут сделаны какие-то изменения с помощью команды UPDATE
- Естественно, вы можете скомбинировать эти особенности, и сказать, — удвоить комиссию всем продавцам в Лондоне, таким предложением:
 - UPDATE Salespeople SET comm = comm * 2
 - WHERE city = 'London':

МОДИФИЦИРОВАНИЕ ПУСТЫХ (NULL) ЗНАЧЕНИЙ

- Предложение SET — это не предикат. Он может вводить пустые NULL значения также, как он вводил значения, не используя какого-то специального синтаксиса (такого, например, как IS NULL).
- Так что, если вы хотите установить все оценки заказчиков в Лондоне в NULL, вы можете ввести следующее предложение:
- UPDATE customers
- SET rating = NULL
- WHERE city = 'London'; что обнулит все оценки заказчиков в Лондоне.

РЕЗЮМЕ

- Теперь вы овладели мастерством управления содержанием вашей базы данных с помощью трех простых команд:
- INSERT — используемой чтобы помещать строки в базу данных;
- DELETE — чтобы удалять их;
- REFERENCES — чтобы изменять значения в уже вставленных строках.

- Вы обучались использованию предиката с командами UPDATE и DELETE чтобы определять, на которую из строк будет воздействовать команда.
- Конечно, предикаты как таковые — не значимы для INSERT, потому что обсуждаемая строка не существует в таблице до окончания выполнения команды INSERT.
- Однако, вы можете использовать запросы с INSERT, чтобы сразу помещать все наборы строк в таблицу. Причем это, вы можете делать со столбцами в любом порядке.

- Вы узнали, что значения по умолчанию, могут помещаться в столбцы, если вы не устанавливаете это значение явно.
- Вы также видели использование стандартного значения по умолчанию, которым является NULL.
- Кроме того, вы поняли, что UPDATE может использовать выражение значения, тогда как INSERT не может.

РАБОТА С SQL

1. Напишите команду, которая бы поместила следующие значения, в их нижеуказанном порядке, в таблицу Продавцов: city — San Jose, name — Bianco, comm — NULL, cnum — 1100.
2. Напишите команду, которая бы удалила все заказы заказчика Clemens из таблицы Порядков.
3. Напишите команду, которая бы увеличила оценку всех заказчиков в Риме на 100.
4. Продавец Serres оставил компанию. Переназначьте его заказчиков продавцу Motika.