

Язык манипулирования данными

(Data Manipulation Language, DML)

Пример простой схемы БД

Столбцы таблицы **Salespeople** (Продавцы)

Столбец	Описание
<u>snum</u>	Уникальный номер, присваиваемый каждому продавцу (номер служащего)
sname	Фамилия продавца
city	Город, где находится продавец, т. е. один из офисов компании.
comm	Комиссионное вознаграждение продавца в десятичной форме

Пример простой схемы БД

Salespeople (Продавцы)

Snum	Sname	City	Comm
1001	Peel	London	.12
1002	Serres	San Jose	.13
1004	Motika	London	.11
1007	Rifkin	Barcelona	.15
1003	Axelrod	New York	.10

Пример простой схемы БД

Столбцы таблицы Customers (Покупатели)

Столбец	Описание
<u>snum</u>	Уникальный номер, присваиваемый каждому покупателю
sname	Фамилия покупателя
city	Город, где находится покупатель. Это один из офисов компании, а не место проживания покупателя
rating	Числовой код, который показывает уровень предпочтения для покупателя. NULL обозначает покупателя, которому еще не присвоен рейтинг
snum	Номер продавца (из таблицы Salespeople), 4 прикрепленного к данному покупателю

Пример простой схемы БД

Customers (Покупатели)

Cnum	Cname	City	Rating	Snum
2001	Hoffman	London	100	1001
2002	Giovanni	Rome	200	1003
2003	Liu	San Jose	200	1002
2004	Grass	Berlin	300	1002
2006	Clemens	London	NULL	1001
2008	Cisneros	San Jose	300	1007
2007	Pereira	Rome	100	1004

Пример простой схемы БД

Столбцы таблицы **Orders (Заказы)**

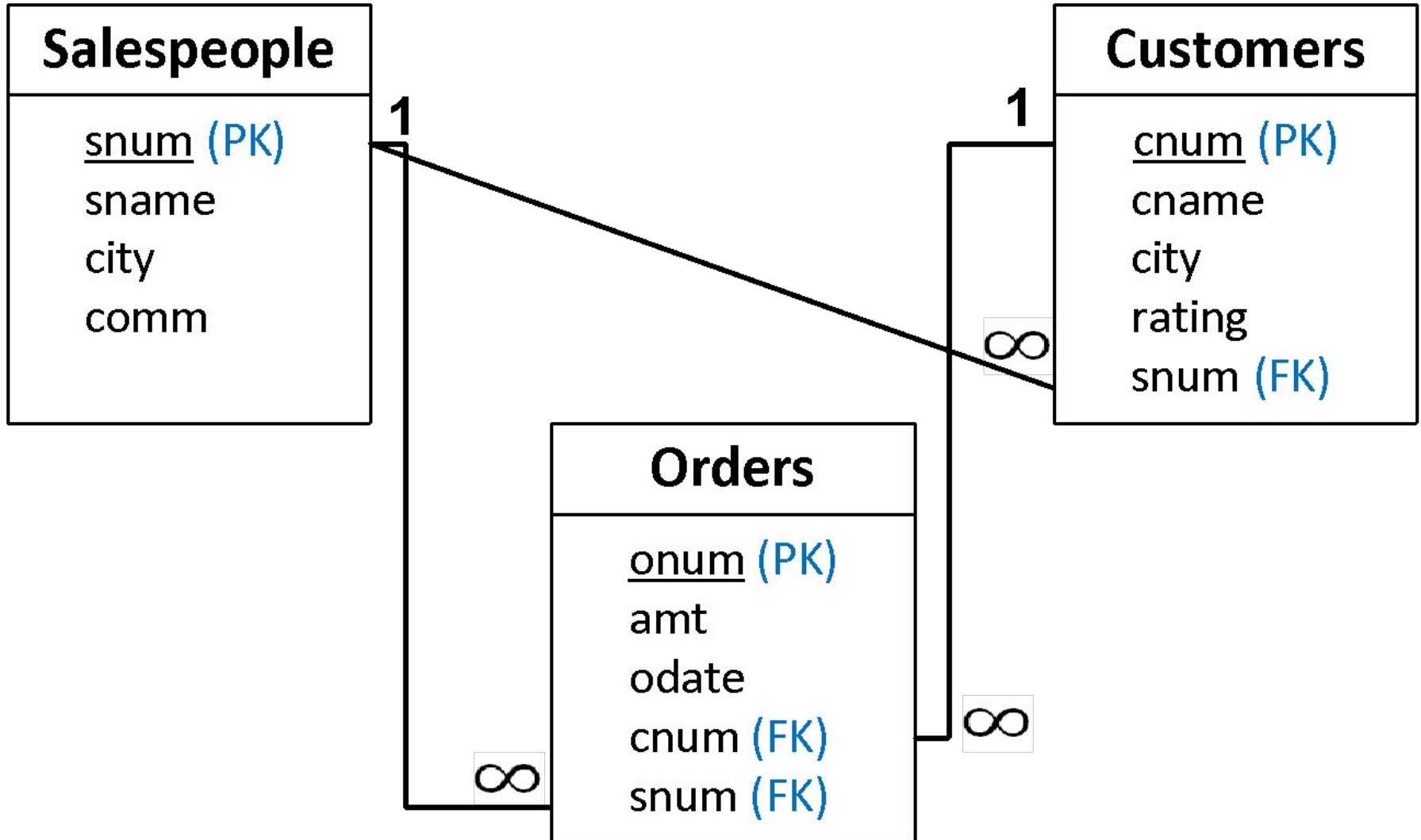
Столбец	Описание
<u>onum</u>	Уникальный номер, присваиваемый каждой покупке
amt	Сумма покупки
odate	Дата покупки
cnum	Номер покупателя (из таблицы Customers), делающего покупку
snum	Номер продавца (из таблицы Salespeople), совершившего продажу. Обычно это продавец, прикрепленный к покупателю в таблице Customers, но не всегда

Пример простой схемы БД

Orders (Заказы)

Onum	Amt	Odate	Cnum	Snum
3001	18.69	10/03/2000	2008	1007
3003	767.19	10/03/2000	2001	1001
3002	1900.10	10/03/2000	2007	1004
3005	5160.45	10/03/2000	2003	1002
3006	1098.16	10/03/2000	2008	1007
3009	1713.23	10/04/2000	2002	1003
3007	75.75	10/04/2000	2004	1002
3008	4723.00	10/05/2000	2006	1001
3010	1309.95	10/06/2000	2004	1002
3011	9891.88	10/06/2000	2006	1001

Пример простой схемы БД



Подразделы SQL

SELECT

Язык запросов (Queries)

CREATE

ALTER

DROP

**Язык определения данных
(DDL)**

INSERT

UPDATE

DELETE

**Язык манипулирования
данными (DML)**

GRANT

REVOKE

**Язык управления доступа
к данным (DCL)**

COMMIT

ROLLBACK

**Язык управления
транзакциями (TCL)**

Язык манипулирования данными

Команды **Data Manipulation Language (DML)** для работы с таблицами:

- INSERT — вставка строк в таблицу
- UPDATE — изменение значений полей
- DELETE — удаление строк из таблицы

Ввод значений в таблицы

Форма оператора **INSERT**:

```
INSERT INTO table_name (list_of_columns)  
VALUES (list_of_values);
```

Данные вводятся в таблицу в порядке их перечисления

Пример ввода строки в таблицу

```
INSERT INTO Salespeople  
      (snum, sname, city, comm)  
VALUES (1001, 'Peel', 'London', .12);
```

Ввод значений в таблицы

Список столбцов может быть опущен

Пример ввода строки в таблицу

Salespeople:

```
INSERT INTO Salespeople  
VALUES (1001, 'Peel', 'London', .12);
```

для определения порядка столбцов в
таблице используйте:

DESCRIBE table_name

Ввод неопределенных значений

NULL может явно вводиться в поле так же, как обычное значение:

```
INSERT INTO Salespeople  
    (snum, sname, city, comm)  
VALUES (1001, 'Peel', NULL, .12);
```

Поле не должно иметь ограничения **NOT NULL**

Ввод значений по умолчанию

Можно вводить в поле значение по умолчанию:

```
INSERT INTO Salespeople  
    (snum, sname, city, comm)  
VALUES (1001, 'Peel', DEFAULT, .12);
```

Перед вставкой для поля должно быть установлено значение по умолчанию (**DEFAULT**)

Ввод неопределенных значений

Неявный метод:

Если столбец не указан в списке столбцов – его значение для записи будет **NULL** (или **DEFAULT**)

В примере не указан столбец *rating*

```
INSERT INTO Customers  
    (cnum, cname, city, snum)  
VALUES (2009, 'Bond', London, 1001);
```

Поле не должно иметь ограничения **NOT NULL**

Значения по умолчанию и NULL

Оператор **INSERT**, не содержащий значения для столбца

- вставит **NULL**, если столбец не содержит ограничения NOT NULL и предложения DEFAULT
- вставит **значение по умолчанию**, если столбец содержит предложение DEFAULT
- выдаст **ошибку**, если столбец содержит ограничение NOT NULL, а значение по умолчанию не указано

Вставка даты

При вставке даты можно использовать:

- `SYSDATE` – текущая дата
- `TO_DATE(...)` – конкретная дата

Вставка результатов запроса

Оператор **INSERT** можно использовать для вставки результатов запроса:

```
INSERT INTO table_name  
(list_of_columns)  
query;
```

Пример перемещения значений из одной таблицы в другую:

```
INSERT INTO Londonstaff  
SELECT * FROM Salespeople  
WHERE city = 'London';
```

Вставка сразу нескольких записей

Для вставки нескольких записей в одну таблицу используйте **INSERT ALL**

```
INSERT ALL  
INTO table_name (list_of_columns)  
VALUES (list_of_values);  
INTO table_name (list_of_columns)  
VALUES (list_of_values);  
INTO table_name (list_of_columns)  
VALUES (list_of_values);  
...
```

Создание новой таблицы

Создать новую таблицу и скопировать все данные из старой:

```
CREATE TABLE new_table AS  
SELECT * FROM old_table;
```

Задания на INSERT

1. Добавить покупателя **Bush** из Вашингтона с привязкой к продавцу **Bond**
2. Добавить по одному заказу каждому продавцу с любой датой текущего года
3. Добавить два заказа от текущей даты для покупателя **Bush** и созданного продавца **Bond**
4. Добавить нового продавца **Freddy** из Нью-Йорка с комиссионными равными максимальным комиссионным продавцов + 0,05
5. Добавить дополнительный заказ для каждого покупателя, которые делали заказ из Лондона. Сумма нового заказа в полтора раза больше прошлого. Новый заказ провести датой на месяц позднее даты прошлого заказа.

Изменение значений полей

Изменение всех или некоторых значений в существующих строках выполняется с помощью оператора **UPDATE**:

```
UPDATE имя таблицы  
SET { имя столбца = { выражение  
                        | NULL  
                        | DEFAULT }  
      },...  
[ { WHERE предикат } ];
```

Изменение значений полей

Пример обновления значений поля для всех строк таблицы **Customers**:

```
UPDATE Customers  
SET rating = 200;
```

Пример обновления значений поля для отдельных строк таблицы **Customers**:

```
UPDATE Customers  
SET rating = 200  
WHERE snum = 1001;
```

Изменение значений полей

Пример обновления значений группы полей для отдельной строки таблицы

Sales

```
UPDATE Salespeople
SET  sname = 'Gibson',
       city  = 'Boston',
       comm  = .10
WHERE snum = 1004;
```

Не разрешается обновлять несколько таблиц в одном операторе **UPDATE**

Изменение значений полей

В предложении **SET** оператора **UPDATE** разрешается применять выражения:

```
UPDATE Salespeople  
SET comm = comm * 2;  
WHERE city = 'London';
```

Разрешается обновлять до **NULL**-значений или значений по умолчанию:

```
UPDATE Customers  
SET rating = NULL, city = DEFAULT  
WHERE snum = 1001;
```

Изменение значений полей с подзапросами

- Обновление с помощью многостолбцового подзапроса

```
UPDATE Salespeople
SET (city, comm) =
      (SELECT city, comm
       FROM Salespeople
       WHERE snum = 1001)
WHERE snum = 1008;
```

Изменение значений полей с подзапросами

Обновление строк таблицы на основе значений из других таблиц:

```
UPDATE Salespeople
SET    city =
        (SELECT city
         FROM Customers
         WHERE cnum = 2008)
WHERE  snum =
        (SELECT snum
         FROM Customers
         WHERE cnum = 2008);
```

Контроль целостности при изменении

- Если вы пытаетесь обновить запись и новое значение столбца противоречит **ограничению**, выдается сообщение об ошибке:

```
UPDATE Orders  
SET   snum = 1017  
WHERE snum = 1007;
```

ORA-02291: integrity constraint
(WATER.FK_ORDERS_SALESPEOPLE)
violated - parent key not found

Задания на UPDATE

1. Для всех покупателей продавца *Peel* выставить рейтинг 100
2. Передать всех покупателей продавца *Peel* продавцу *Motika* и увеличить рейтинг данных покупателей на 50
3. Увеличить на 100 рейтинг покупателей, которые совершили покупки на сумму выше средней за последний год
4. Уменьшить в два раза комиссионные для всех продавцов, которые имеют общую сумму заказов меньше 100 за последний год

Удаление строк из таблицы

Для удаления строк из таблицы используется команда **DELETE**:

```
DELETE FROM ИМЯ  
таблицы  
[ { WHERE предикат } ];
```

Пример удаления всех строк из таблицы:

```
DELETE FROM Salespeople;
```

Удаление строк из таблицы

Удаление определенных строк, для указания которых используются предикаты:

```
DELETE FROM Salespeople  
WHERE snum = 1003;
```

```
DELETE FROM Salespeople  
WHERE city = 'London';
```

Удаление строк из таблицы с подзапросами

Для удаления определенных строк на основе значений из *другой таблицы* используйте подзапросы в операторе **DELETE**

Задание:

Удалить все заказы из таблицы **Orders**, выполненные продавцами из Лондона

Задания на DELETE

1. Удалить заказы продавцов, которые проживают в разных городах со своими покупателями
2. Удалить покупателей, которые не сделали ни одного заказа
3. Удалить все заказы продавца (-ов), у которого максимальный размер комиссионных