

База Данных

Лекция 6

SQL.

**Язык манипулирования
данными**

SQL

- SQL (Structured Query Language) - это язык программирования специального назначения, предназначенный для управления данными, хранящимися в системе управления реляционными базами данных (RDBMS).
- Ключевые слова SQL не чувствительны к регистру: `select` - это то же самое, что `SELECT`
- Все SQL-запросы должны заканчиваться точкой с запятой “;”
- Точка с запятой - это стандартный способ разделения каждой инструкции SQL в СУБД, которые позволяют выполнять более одной инструкции SQL в одном вызове.

Комментарии SQL

- Комментарии используются для объяснения разделов инструкций SQL или для предотвращения выполнения инструкций SQL.
- Однострочные комментарии начинаются с --.
- Любой текст между -- и концом строки будет проигнорирован (не будет выполнен).

--Select all:

```
SELECT * FROM Students;
```

Комментарии SQL

- Многострочные комментарии начинаются с /* и заканчиваются на */.

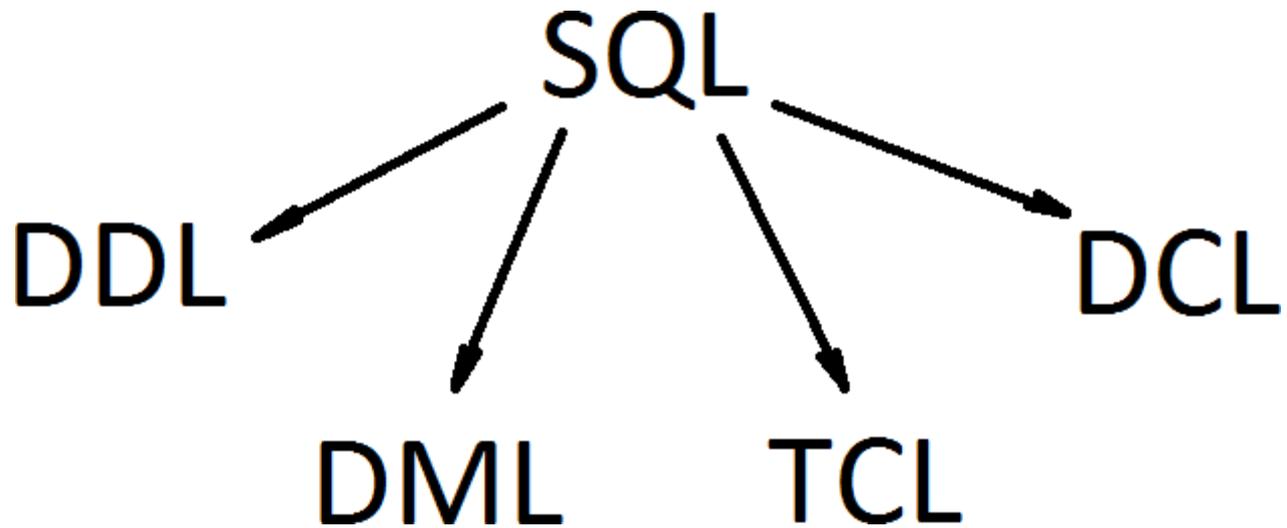
- Любой текст между /* и */ будет проигнорирован.

```
/*Select all the columns of all the records in the  
Students table:*/  
SELECT * FROM Students;
```

- Чтобы проигнорировать только часть утверждения, также используйте /*
*/ комментарий.

```
SELECT Iname, /*fname,*/ gpa FROM Students;
```

Структура SQL



- DDL (Язык определения данных)
- DML (Язык манипулирования данными)
- TCL (Язык управления транзакциями)
- DCL (Язык управления данными)

Прошлая лекция

Язык определения данных (DDL) определяет конструкции, которые структурируют данные в базе данных.

Инструкции DDL:

- CREATE
- ALTER
- DROP

Язык манипулирования данными

Язык манипулирования данными (DML) является подязыком SQL и используется для выбора, вставки, удаления и обновления данных в базе данных.

Операторы DML:

- INSERT
- UPDATE
- DELETE
- SELECT

DDL против DML

Когда таблица создается (с помощью инструкций DDL), она не содержит данных (data - DML).

stud_id	f_name	l_name	bdate	group_id

Вставлять (INSERT)

Оператор INSERT используется для вставки новых записей в таблицу.

Синтаксис:

```
INSERT INTO table_name VALUES (values);
```

- В SQL все строки и даты должны быть заключены в одинарные кавычки.
- Значения данных перечислены в том порядке, в котором столбцы отображаются в таблице, разделенные запятыми.

ВСТАВИТЬ: пример 2

Можно вставить подмножество данных в таблицу. Мы можем указать, какие столбцы мы хотели бы вставить:

```
INSERT INTO table_name (column(s))  
VALUES (values);
```

Вставьте новую запись в таблицу Students для студента с id=2 и без даты рождения:

```
INSERT INTO Students (stud_id, f_name,  
I_name, group_id)  
VALUES (2, 'Firstname2', 'Lastname2', 1);
```

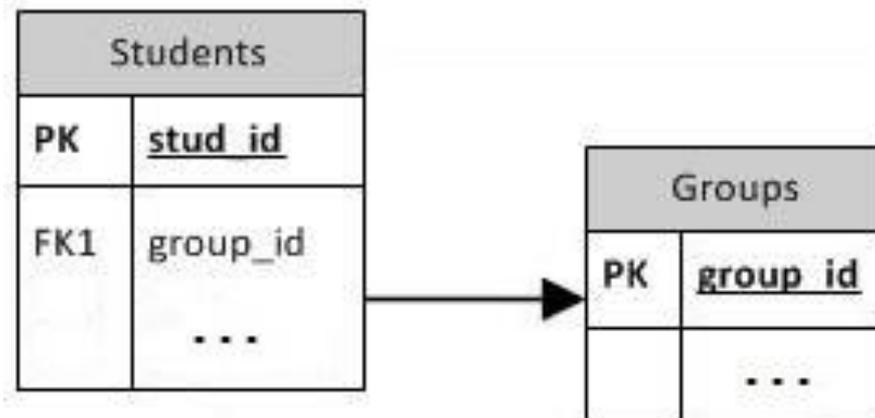
ВСТАВИТЬ: пример 3

Концептуально данные вставляются по одной строке за раз. Однако вы можете вставить несколько строк одной командой:

```
INSERT INTO Students (stud_id, f_name,  
                      l_name, group_id)  
VALUES (1, 'FirstName1', 'LastName1', 1),  
       (2, 'FirstName2', 'LastName2', 1);
```

Ссылочная целостность (СОЗДАНИЕ ТАБЛИЦЫ)

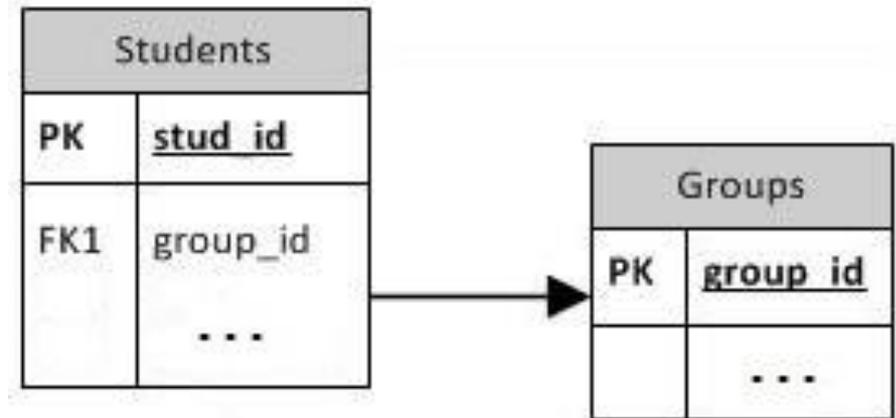
- Условие для использования ссылочной целостности:
- Таблица, на которую ссылается FK, должна быть создана до определения внешних ключей
- Например, сначала должна быть создана таблица Groups, а затем может быть создана таблица Students (с ее FK).



Ссылочная целостность (СОЗДАНИЕ ТАБЛИЦЫ)

```
CREATE TABLE Groups(  
  group_id int,  
  group_name varchar(15),  
  PRIMARY KEY (group_id));
```

```
CREATE TABLE Students(  
  stud_id int,  
  first_name varchar(20),  
  last_name varchar(20),  
  bdate date,  
  group_id int,  
  PRIMARY KEY (stud_id),  
  FOREIGN KEY (group_id) REFERENCES Groups(group_id));
```



Ссылочная целостность (INSERT)

Строки не могут быть вставлены в таблицу Students без соответствующих строк в таблице Groups. Эта запись недоступна, пока таблица Groups пуста:

```
INSERT INTO Students
```

```
VALUES (1, 'FirstName1', 'LastName1', '31.12.1994', 1);
```

ОШИБКА: вставка или обновление таблицы "students" нарушает ограничение внешнего ключа "students_group_id_fkey" ПОДРОБНЕЕ: Ключ (group_id)=(1) отсутствует в таблице "groups".

ОБНОВЛЕНИЕ (UPDATE)

SQL предоставляет инструкцию UPDATE для изменения значений в таблицах.

Синтаксис:

```
UPDATE table_name
```

```
    SET column = value [, column = value, ...]
```

```
    [WHERE condition];
```

- Условие в части WHERE инструкции UPDATE известно как условие выбора.
- Условие выбора аналогично операторам if в традиционном языке программирования.

ОБНОВЛЕНИЕ: пример 1

Предположим, у нас есть студент с `stud_id=2` и `group_id=1`, который необходимо обновить до `group_id=2`.

Мы бы записали это как

```
UPDATE Students  
SET group_id=2  
WHERE stud_id=2;
```

ОБНОВЛЕНИЕ: пример 2, 3

SQL-команда для изменения имени студентки с `stud_id=1` на Aigul:

```
UPDATE Students  
SET f_name='Aigul'  
WHERE stud_id=1;
```

Команда SQL для изменения даты рождения студента с `stud_id=2` на 15.10.1994:

```
UPDATE Students  
SET bdate='15.10.1994'  
WHERE stud_id=2;
```

ОБНОВЛЕНИЕ: пример 4

- Каждый столбец может быть обновлен отдельно; другие столбцы не затрагиваются.
- Вы можете обновить отдельные строки, все строки в таблице или подмножество всех строк.
- Например, эта команда обновляет всех учащихся, у которых `group_id=1`, до `group_id=2`:

```
UPDATE Students  
SET group_id=2  
WHERE group_id=1;
```

ОБНОВЛЕНИЕ: пример 5

- Выражение для нового значения может ссылаться на существующее значение(значения) в строке.
- Если вы хотите повысить цену на все продукты на 10%, вы можете использовать:
`UPDATE Products SET price = price*1.10;`
- Мы также опустили предложение WHERE. Если оно опущено, это означает, что обновляются все строки в таблице. Если оно присутствует, обновляются только те строки, которые соответствуют условию WHERE.

УДАЛИТЬ (DELETE): пример 1

SQL предоставляет инструкцию DELETE для удаления данных из таблиц.

Синтаксис:

```
DELETE FROM table_name [WHERE  
condition];
```

УДАЛИТЬ (DELETE): пример 2

- Чтобы удалить все строки в таблице Students, выполните следующие действия:
`DELETE FROM Students;`
- Чтобы удалить студента с stud_id=2:
`DELETE FROM Students
WHERE stud_id=2;`

УДАЛИТЬ (DELETE): пример 3

- Чтобы удалить все строки из таблицы Товаров, цена которых равна 10:

```
DELETE FROM Products  
WHERE price=10;
```

Следующая лекция: ВЫБРАТЬ (SELECT)

SQL позволяет запрашивать данные с помощью инструкции SELECT.

Синтаксис:

```
SELECT attribute(s) FROM table(s);
```

quiz

<https://www.w3schools.com/quiztest/quiztest.asp?qtest=SQL>

Упражнение 3

Создайте базу данных и таблицу.
Определите имя поля, размер,
тип данных, первичный ключ в
тетради.

UserName	FirstName	SecondName	DateOfBirth
gem123	John	Smith	01/01/1995
purpleSky	Muhammed	Ali	23/02/1956
OpenWindow	Sunny	Amir	03/03/1997
bluebird127	Raziya	Bello	04/03/1982

Упражнение 4. Создайте базу данных, определите внешний ключ, добавьте 3 таблицы

A social media website has a relational database, WEBDATA, that stores the site's information.

The database has three tables to store users' details, and details of the images and text that they post.

USER(UserName, FirstName, SecondName, DateOfBirth) PHOTO(PhotoID, UserName, Comment, UploadDate) TEXTPOST(PostID, UserName, DateOfPost, TheText)

Data Manipulation Language, DML

Пример:

- **SELECT * FROM Customers;**

Выводит все строки и таблицы с Customers table

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Student	Avda. de la Constituciyn 2222	Taraz	05021	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	Berlin	WA1 1DP	UK
5	Berglunds snabbkцp	Christina Berglund	Berguvsvдgen 8	Lulee	S-958 22	Sweden

Data Manipulation Language, DML

Пример :

- **SELECT CustomerName, Country FROM Customers;**

Выводит записи **CustomerName** ,
Country fields from
Customers table

CustomerName	Country
Alfreds Futterkiste	Germany
Ana Trujillo Emparedados y helados	Mexico
Antonio Moreno Taqueria	Mexico
Around the Horn	UK
Berglunds snabbkupp	Sweden

Data Manipulation Language, DML

пример:

- **SELECT CustomerName, City FROM Customers
WHERE City='Berlin';**

Показывает записи **CustomerName**,
City fields, from **Customers** table
where the **City** field value is equal to
the word '**Berlin**'

CustomerName	City
Alfreds Futterkiste	Berlin
Around the Horn	Berlin

Activity

- **Go to this link**

http://sqlzoo.net/wiki/SELECT_from_Nobel_Tutorial

perform the task of 1, 2, 5, 8, 12

Show your answers for teacher

Data Manipulation Language, DML

To change the values in one or more columns of the table used UPDATE statement.

- **UPDATE table_name SET Field = new_Value
WHERE selection condition;**

Data Manipulation Language, DML

Example:

- **UPDATE Customers SET ContactName='Student', City='Taraz' WHERE CustomerID=2;**

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constituciyn 2222	Mїxico D.F.	05021	Mexico
3	Antonio Moreno Taqueria	Antonio Moreno	Mataderos 2312	Mїxico D.F.	05023	Mexico

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Student	Avda. de la Constituciyn 2222	Taraz	05021	Mexico
3	Antonio Moreno Taqueria	Antonio Moreno	Mataderos 2312	Mїxico D.F.	05023	Mexico

After **UPDATE** statements, records fields **CustomerName**, **City** in **Customers** table has changed

Activity

- **Go to this link**

http://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

**Update London to Berlin for CustomerID = 4
Perform the task. Show your answers for
teacher**

Data Manipulation Language, DML

- To add records to the table, use the **INSERT** statement

```
INSERT INTO table_name (field1, field2, field3, ...)  
VALUES (value1, value2, value3, ...);
```

Data Manipulation Language, DML:

Example:

- **INSERT INTO Customers** (CustomerName, ContactName, Address, City, PostalCode, Country) **VALUES** ('Student', 'Anuar Samatov', 'Satpayev 2', 'Taraz', '000000', 'Kazakhstan');

91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
92	Bake	Askar Nagay	Abai 1	Taraz	200000	Kazakhstan
93	Student	Anuar Samatov	Satpayev 2	Taraz	000000	Kazakhstan

After the **INSERT INTO** proposals at the end of the table create a new record with the given values.

Activity

Go to this link

http://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

```
INSERT VALUES ('Bala', 'Askar Nagay',  
'Abai 1', 'Taraz', '200000', 'Kazakhstan');
```

Data Manipulation Language, DML:

To delete rows from a table, use a **DELETE** statement

```
DELETE FROM table-name  
WHERE selection condition
```

Data Manipulation Language, DML

Example:

DELETE FROM Customers WHERE CustomerID=3;

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Student	Avda. de la Constituciyn 2222	Taraz	05021	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	Berlin	WA1 1DP	UK

After the proposal **DELETE FROM**, the third record with values completely delete.

Activity

Go to this link

http://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

Delete row where CustomerID=12

Perform the task. Show your answers for teacher

КНИГИ

- Коннолли, Томас М. Системы баз данных: практический подход к проектированию, внедрению и управлению / Томас М. Коннолли, Кэролин Э. Бегг.- Соединенные Штаты Америки: Pearson Education
- Гарсия-Молина, Х. Система баз данных: Полная книга / Эктор Гарсия-Молина.- Соединенные Штаты Америки: Пирсон Прентис Холл
- Шарма, Н. Основы баз данных: Книга для сообщества от сообщества / Нирадж Шарма, Ливиу Перниу.- Канада
- www.postgresql.org/docs/manuals/

Вопрос

Третья нормальная форма основана на концепции ...

- Транзитивной зависимости
- Частичной зависимости
- Внешней зависимости
- Ни один из приведенных

Вопрос

Таблица в 1NF, в которой первичный ключ состоит из двух из трех его атрибутов:

- Всегда нарушает 2NF
- Никогда не нарушает 2NF
- Может нарушать 2NF
- None

Вопрос

Таблица содержит поля F1, F2, F3, F4 и F5 со следующими функциональными зависимостями:

F1 -> F3, F2 -> F4,
(F1, F2) -> F5.

С точки зрения нормализации, эта таблица находится в ...

- 1NF
- 2NF
- 3NF
- UNF