

БАЗЫ ДАННЫХ

Язык запросов SQL (Structured Query Language).
Введение

История версий стандарта SQL

- 1986. SQL-86, SQL-87. Первый вариант стандарта, принятый институтом ANSI и одобренный ISO в 1987 году.
- 1989. SQL-89, [FIPS 127-1](#). Немного доработанный вариант предыдущего стандарта.
- [1992. SQL-92](#). SQL2, FIPS 127-2. Значительные изменения (ISO 9075); уровень *Entry Level* стандарта SQL-92 был принят как стандарт FIPS 127-2.
- [1999. SQL:1999](#) SQL:1999. SQL3. Добавлена поддержка [регулярных выражений](#), [рекурсивных запросов](#), поддержка [триггеров](#) SQL:1999. SQL3. Добавлена поддержка регулярных выражений, рекурсивных запросов, поддержка триггеров, базовые процедурные расширения, нескаллярные типы данных и некоторые [объектно-ориентированные](#) возможности.
- [2003. SQL:2003](#) SQL:2003. Введены расширения для работы с [XML](#) SQL:2003. Введены расширения для работы с XML-данными, оконные функции (применяемые для работы с [OLAP](#)-базами данных), генераторы последовательностей и основанные на них типы данных.
- [2006. SQL:2006](#). Функциональность работы с XML-данными значительно расширена. Появилась возможность совместно использовать в запросах SQL и [XQuery](#).
- [2008. SQL:2008](#). Улучшены возможности оконных функций, устранены некоторые неоднозначности стандарта SQL:2003

Соответствие стандарту

Впервые понятие «уровня соответствия» было предложено в стандарте SQL-92. ANSI определяли четыре уровня соответствия реализации этому стандарту, где каждый последующий уровень соответствия заведомо подразумевал соответствие предыдущему уровню :

- *Entry* (базовый)
- *Transitional* (переходный) — проверку на соответствие этому уровню проводил только NIST
- *Intermediate* (промежуточный)
- *Full* (полный)

Положение изменилось с введением стандарта SQL:1999. в котором стандарт приобрёл модульную структуру — основная часть стандарта была вынесена в раздел «SQL/Foundation», все остальные были выведены в отдельные модули. Соответственно, остался только один уровень совместимости — что означает поддержку этой основной части. Поддержка остальных возможностей оставлена на усмотрение производителей СУБД. Аналогичное положение имело место и с последующими версиями стандарта.

Расширения SQL

- [InterBase](#)/[Firebird](#) – краткое название языка – [PSQL](#).
Procedural SQL
- IBM [DB2](#) – краткое название языка – [SQL PL](#). SQL Procedural Language (расширяет [SQL/PSM](#) SQL Procedural Language (расширяет SQL/PSM)); также в DB2 хранимые процедуры могут писаться на обычных языках программирования: [Си](#) SQL Procedural Language (расширяет SQL/PSM); также в DB2 хранимые процедуры могут писаться на обычных языках программирования: Си, [Java](#) и т. д.
- [MS SQL Server](#)/[Sybase ASE](#) – краткое название языка – [Transact-SQL](#). Transact-SQL
- [MySQL](#) MySQL – краткое название языка – [SQL/PSM](#).
SQL/Persistent Stored Module
- [Oracle](#) Oracle – краткое название языка – [PL/SQL](#). Procedural Language/SQL (основан на языке [Ada](#))
- [PostgreSQL](#) PostgreSQL – краткое название языка – [PL/pgSQL](#).
Procedural Language/PostgreSQL Structured Query Language (очень похож на Oracle PL/SQL)

Основные подмножества SQL (1/2)

Аббревиатура	Описание	Действия	Основные команды
DDL	(Data Definition Language) язык определения данных	Создание, изменение и удаление объектов БД	CREATE, ALTER, DROP
DML	(Data Manipulation Language) язык манипулирования данными	Вставка, изменение и удаление строк таблицы	INSERT, UPDATE, DELETE
DQL	(Data Query Language) язык запросов	Выборка строк таблицы	SELECT

Основные подмножества SQL (2/2)

Аббревиатура	Описание	Действия	Основные команды
DCL	(Data Control Language) язык управления данными	Управление доступом	GRANT, REVOKE
TPL	(Transaction Processing Language) язык обработки транзакций	Фиксация или откат транзакции, сохранение промежуточной точки	COMMIT, ROLLBACK, SAVEPOINT
CCL	(Cursor Control Language) язык управления курсорами	Выборка данных при помощи курсора, его открытие и закрытие	OPEN, CLOSE, FETCH

Команды DDL

CREATE – создание объекта.

ALTER – изменения структуры объекта.

DROP – удаление объекта.

Общий вид синтаксиса команд DDL:

create

alter

drop

тип_объекта имя_объекта [параметры];

Создание таблиц

```
CREATE TABLE [имя_схемы.]имя_таблицы  
  ( имя_поля тип_данных [(размер)] [NOT NULL]  
    [DEFAULT выражение]  
    [ограничения_целостности_поля...]  
    ,...  
    [, ограничения_целостности_таблицы ,...]  
  );
```

ограничения_целостности (ОЦ):

```
[CONSTRAINT имя_ОЦ] название_ОЦ [параметры]
```

Числовые типы

точные целочисленные

tinyint	1-байтовое целое без знака	MS SQL
smallint	2-байтовое целое со знаком	SQL/89
integer, int	4-байтовое целое со знаком	SQL/92
bigint	8-байтовое целое со знаком	MS SQL

точные вещественные

numeric(p,s), decimal(p,s)	Число из p цифр, содержащее s цифр в дробной части	SQL/99
-------------------------------	--	--------

Символьные (однобайтовые)

character(n), char(n)	Строка однобайтовых символов фиксированной длины n	SQL/89
character varying(n), varchar(n)	Строка однобайтовых символов переменной длины максимальным размером n	SQL/92
text	Строка однобайтовых символов до 2 Гб (данные хранятся на отдельных страницах по 8 Кб)	MS SQL

Символьные (Unicode)

national character(n), national char(n), nchar(n)	Строка Unicode символов фиксированной длины n	SQL/92
national character varying(n), national char varying(n), nvarchar(n)	Строка Unicode символов переменной длины максимальным размером n	SQL/92
ntext	Строка Unicode символов до 2 Гб (данные хранятся на отдельных страницах)	MS SQL

Двоичные типы данных

binary(n)	Блок двоичных данных фиксированной длины n от 1 до 8000 байт	MS SQL
varbinary(n)	Блок двоичных данных переменной длины максимальным размером n от 1 до 8000 байт	MS SQL
image	Блок двоичных данных до 2 Гб (данные хранятся на отдельных страницах по 8 Кб)	MS SQL

Дата и время

date	календарная дата (день, месяц и год)	SQL/92
time	Время в течение суток	SQL/92
timestamp	Дата и время	SQL/92
datetime	Дата и время с 1 янв. 1753 г. по 31 дек. 9999 г. точность 3.33 мс 8 байт	MS SQL
smalldatetime	Дата и время с 1 янв. 1900 г. до 6 июня 2079 г. точность 1 мин. 4 байта	MS SQL

Ограничения целостности

Ограничения целостности:

✓ первичный ключ:

PRIMARY KEY(*имя_атрибута1* [, *имя_атрибута2*,...])

✓ внешний ключ:

FOREIGN KEY(*имя_атрибута1* [, *имя_атрибута2*,...]) **REFERENCES**
имя_таблицы [(*имя_атрибута1* [, *имя_атрибута2*,...])]

✓ уникальность (значений атрибута или комбинации значений атрибутов):

UNIQUE (*имя_атрибута1* [, *имя_атрибута2*,...])

✓ обязательность / необязательность:

NOT NULL / NULL

✓ условие на значение поля:

CHECK (*условие*)

Например: `check (salary >= 4500)`, `check (date2 > date1)`

Ограничения целостности: первичный ключ таблицы

→ ID_Дисциплина INTEGER NOT NULL **PRIMARY KEY**

Key	Id	Name	Data Type	Size	Nulls	Default
		ID_План	int	4	<input type="checkbox"/>	
		ID_Дисциплина	int	4	<input type="checkbox"/>	
		Семестр	int	4	<input type="checkbox"/>	
		Отчетность	char	1	<input type="checkbox"/>	
		Количество_часов	int	4	<input checked="" type="checkbox"/>	
		ID_преподавателя	int	4	<input checked="" type="checkbox"/>	

→ **PRIMARY KEY** (ID_Дисциплина, ID_Студент)

Key	Id	Name	Data Type	Size	Nulls	Default
		ID_Студент	int	4	<input type="checkbox"/>	
		ID_Дисциплина	int	4	<input type="checkbox"/>	
		Оценка	int	4	<input type="checkbox"/>	
		Дата_сдачи	datetime	8	<input type="checkbox"/>	

Ограничения целостности: внешний ключ таблицы

FOREIGN KEY (<список столбцов внешнего ключа>)

REFERENCES <имя родительской таблицы>

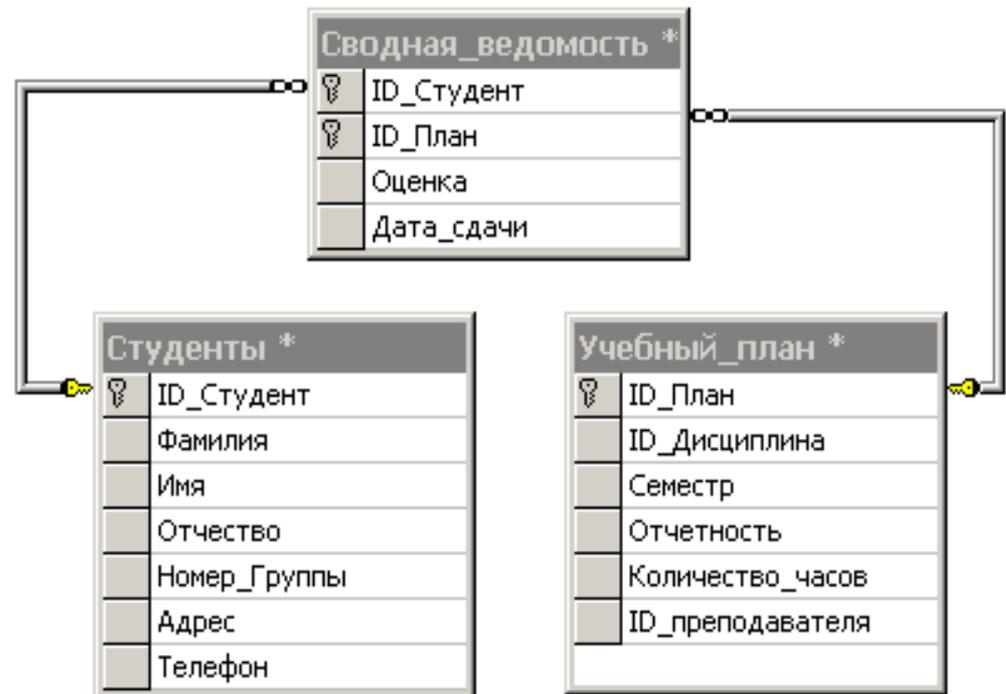
[[<список столбцов родительской таблицы>]

[**ON DELETE** {**RESTRICT** | **CASCADE** | **SET DEFAULT** | **SET NULL**}]

[**ON UPDATE** {**RESTRICT** | **CASCADE** | **SET DEFAULT** | **SET NULL**}]

```
ALTER TABLE Сводная_ведомость
ADD FOREIGN KEY (ID_План)
REFERENCES Учебный_план
```

```
ALTER TABLE Сводная_ведомость
ADD FOREIGN KEY
(ID_Студент)
REFERENCES Студенты
```



Ограничения целостности: определение уникального столбца

На уровне столбца:

CREATE TABLE	Дисциплины
(ID_Дисциплина	INTEGER NOT NULL PRIMARY KEY,
Наименование	VARCHAR(20) NOT NULL UNIQUE)

На уровне таблицы:

CREATE TABLE	Дисциплины
(ID_Дисциплина	INTEGER NOT NULL,
Наименование	VARCHAR(20) NOT NULL,
PRIMARY KEY	(ID_Дисциплина),
UNIQUE	(Наименование))

Ограничения целостности: определение проверочных ограничений

На уровне столбца:

Семестр **INTEGER NOT NULL**
CHECK ((Семестр >= 1) **AND** (Семестр <= 10))

На уровне таблицы:

CHECK ((Семестр >= 1) **AND** (Семестр <= 10)))

Ограничения целостности: определение значения по умолчанию

Год_поступления **INTEGER DEFAULT YEAR(GETDATE())**

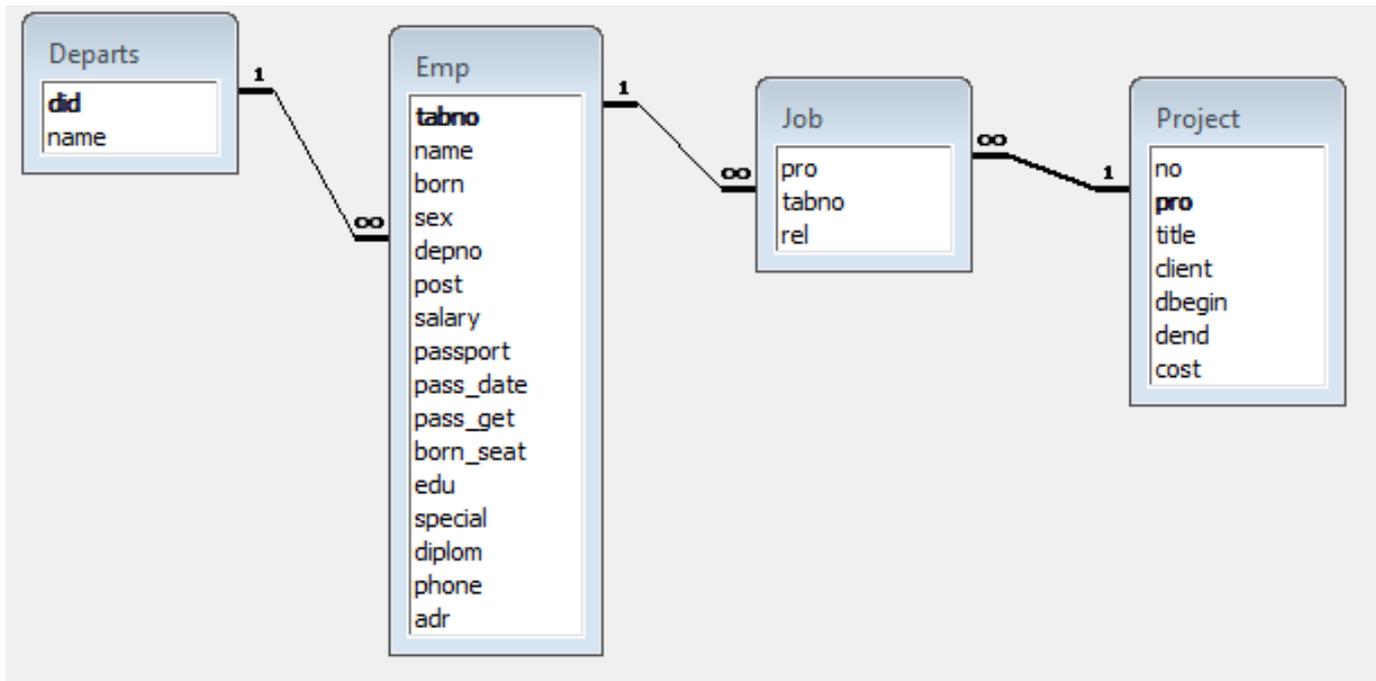
Функция **GETDATE()**

- задает текущую дату

Функция **YEAR()**

- выделяет из даты компонент,
определяющий год

Пример БД: проектная организация



Departs – отделы,
Emp – сотрудники,

Project – проекты,
Job – участие в проектах.

Пример БД: проектная организация

Emp – сотрудники:

tabno – табельный номер сотрудника, первичный ключ;

name – ФИО сотрудника, обязательное поле;

born – дата рождения сотрудника, обязательное поле;

sex – пол сотрудника, обязательное поле;

depno – номер отдела, обязательное поле, внешний ключ;

post – должность сотрудника;

salary – оклад, больше МРОТ;

passport – серия и номер паспорта, уникальный обязательный атрибут;

pass_date – дата выдачи паспорта, обязательное поле;

pass_get – кем выдан паспорт, обязательное поле;

born_seat – место рождения сотрудника;

edu – образование сотрудника;

special – специальность по образованию;

diplom – номер диплома;

phone – телефоны сотрудника;

adr – адрес сотрудника;

edate – дата вступления в должность, обязательное поле.

Пример БД: проектная организация

Departs – отделы:

did – номер отдела, первичный ключ;

name – название отдела, обязательное поле.

Project – проекты:

No – номер проекта, первичный ключ;

title – название проекта, обязательное поле;

pro – краткое название проекта, обязательное уникальное поле;

client – заказчик, обязательное поле;

dbegin – дата начала выполнения проекта, обязательное поле;

dend – дата завершения проекта, обязательное поле;

cost – стоимость проекта, обязательное поле.

Job – участие в проектах:

pro – краткое название проекта, внешний ключ;

tabNo – номер сотрудника, участвующего в проекте, внешний ключ;

rel – роль сотрудника в проекте; может принимать одно из трех значений:

'исполнитель', 'руководитель', 'консультант'.

Первичный ключ – комбинация полей **pro** и **tabNo**.

Создание таблиц БД проектной организации

Таблица «Отделы» (Depart):

```
create table depart (did number(4) constraint pk_depart PRIMARY KEY,  
                    name varchar2(100) not null
```

```
);
```

Таблица «Сотрудники» (Emp):

```
create table emp ( tabno number(6) constraint pk_emp PRIMARY KEY,  
                  name varchar2(100) not null,  
                  born date not null,  
                  sex char  not null,  
                  depno number(4) not null constraint fk_depart REFERENCES departs,  
                  post varchar(50) not null,  
                  salary number(8,2) not null constraint check_sal check (salary > 4630),  
                  passport char(10) not null constraint passp_uniq UNIQUE,  
                  pass_date date not null,          pass_get varchar2(100) not null,  
                  born_seat varchar2(100),          edu varchar2(30),  
                  special varchar2(100),            diplom varchar2(40),  
                  phone varchar2(30),              adr varchar2(80),  
                  edate date not null default trunc(sysdate),  
                  chief number(6) constraint fk_emp REFERENCES emp
```

```
);
```

Создание таблиц БД проектной организации

Таблица «Проекты» (Project):

```
create table project (No number(5) constraint pk_project primary key,
                    title varchar2(200) not null,
                    pro varchar(15) not null constraint pro_uniq unique,
                    client varchar(100) not null,
                    dbegin date not null,
                    dend date not null,
                    cost number(9)
);
```

Таблица «Участие в проектах» (Job):

```
create table job ( pro varchar(15) not null references project (abbr),
                 tabNo number(6) not null references emp,
                 rel varchar(20) default 'исполнитель',
                 primary key (tabno, pro),
                 check ( rel IN ('исполнитель', 'руководитель', 'консультант') )
);
```

Подмножество команд DML

- **INSERT** – добавление строк в таблицу.
 - ✓ Добавляет одну или несколько строк в указанную таблицу.
- **UPDATE** – изменение данных.
 - ✓ Изменяет значения одного или нескольких полей в записях указанной таблицы.
 - ✓ Можно указать условие, по которому выбираются обновляемые строки.
 - ✓ Если условие не указано, обновляются все строки таблицы.
 - ✓ Если ни одна строка не удовлетворяет условию, ни одна строка не будет обновлена.
- **DELETE** – удаление строк из таблицы.
 - ✓ Удаляет одну или несколько строк из таблицы.
 - ✓ Можно указать условие, по которому выбираются удаляемые строки.
 - ✓ Если условие не указано, удаляются все строки таблицы.
 - ✓ Если ни одна строка не удовлетворяет условию, ни одна строка не будет удалена.

Добавление данных

INSERT – добавление строк в таблицу:

```
INSERT INTO имя_таблицы [(список_полей_таблицы)]  
    { VALUES (список_выражений) | запрос };
```

Примеры:

-- Добавить в таблицу "Отделы" новую запись (все поля):

```
insert into depart  
values(7, 'Договорной отдел');
```

-- Добавить в таблицу "Сотрудники" новую запись (не все поля):

```
insert into emp (tabno, name, born, sex, depno, passport, pass_date_pass_get,  
    post, salary, phone)  
values( 301, 'САВИН АНДРЕЙ ПАВЛОВИЧ', to_date('11.07.1969', 'dd.mm.yyyy'),  
    'M', 5, '4405092876', to_date('15.02.1999', 'dd.mm.yyyy'),  
    'ОВД "Митино" г.Москвы', 'программист', 38050, '121-34-11');
```

Замечание: значение по умолчанию используется только тогда, когда значение поля не вводится в явном виде.

Изменение данных

UPDATE – изменение данных:

UPDATE *имя_таблицы*

SET *имя_поля1 = выражение1* [, *имя_поля2 = выражение2,...*]
[**WHERE** *условие*];

Примеры:

-- Изменить статус сотрудника Бобкова Л.П., табельный номер 74, по отношению к проекту 30."Система автоматизированного управления предприятием":

update job

set rel = 'консультант'
where tabno = 74 and pro = 30;

-- Перевести сотрудника Жаринова А.В., табельный номер 68, на должность ведущего программиста и повысить оклад на три тысячи рублей:

update emp

set post = 'ведущий программист', salary = salary+3000
where tabno = 68;

Удаление данных

DELETE – удаление строк из таблицы:

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

Примеры.

-- Удалить сведения о том, что сотрудник Афонасьев В.Н., табельный номер 147, участвует в проектах:

```
delete from job  
where tabno=147;
```

-- Удалить сведения о сотруднике Афонасьеве В.Н., табельный номер 147:

```
delete from emp  
where tabno = 147;
```

Замечание: отменить удаление данных можно командой
ROLLBACK;