Базы данных

Лекция 3.

Язык запросов SQL. Введение

SQL – Structured Query Language

- **SQL** – это структурированный язык запросов к реляционным базам данных.
- SQL – декларативный язык, основанный на операциях реляционной алгебры.
- Стандарты SQL, определённые Американским национальным институтом стандартов (ANSI):

✔ SQL-1 (SQL/89) – первый вариант стандарта.

✔ **SQL-2 (SQL/92) – основной расширенный стандарт.**

✔ SQL-3 (SQL/1999, 2003,…, 2016) – относится к объектно-реляционной модели данных:

1999: поддержка регулярных выражений, рекурсивных запросов, триггеры, базовые процедурные расширения, нескалярные типы данных и некоторые объектно-ориентированные возможности.

2003: расширения для работы с XML-данными, оконные функции (применяемые для работы с OLAP- БД), генераторы последовательностей и основанные на них типы данных.

2006: расширение функциональности работы с XML-данными, возможность совместно использовать в запросах SQL и XQuery.

2011: хронологические базы данных (PERIOD FOR), поддержка конструкции FETCH[8].

2016: защита на уровне строк, полиморфные табличные функции, JSON.

Работа с SQL

- Особенности синтаксиса:
  - ✔ В командах SQL не различаются прописные и строчные буквы (кроме содержимого символьных строк).
  - ✔ Каждая команда может занимать несколько строк и заканчивается символом ';'.
  - ✔ Символ и символьная строка заключается в одинарные кавычки:
    'A', '2' , 'строка', 'другая строка'
  - ✔ Однострочный комментарий начинается с символов '--'.
  - ✔ Многострочный комментарий заключается в символы /* ... */.
- При запуске СУБД необходимо указать логин (имя пользователя) и пароль, например:

- СУБД Oracle 11g запускается из меню Пуск: Oracle -> Go to Home Page
  В окне браузера выбрать пункт меню Log On.
- Клиент MySQL запускается через консоль (программа «Командная строка»):
  mysql –uимя_пользователя -рпароль
  mysql> use имя_базы_данных;
- СУБД Postgres 11 запускается из меню Пуск: PostgreSQL 11 -> pgAdmin 4.

Подмножества языка SQL. Команды DDL

- **DDL** (Data Definition Language) – команды создания / изменения / удаления объектов базы данных (*create/alter/drop*);
- **DML** (Data Manipulation Language) – команды добавления / модификации / удаления данных (*insert/update/delete*), а также команда извлечения данных *select;*
- DCL (Data Control Language) – команды управления данными (установка / снятие ограничений целостности). Входит в подмножество DDL.

CREATE – создание объекта.

ALTER – изменение структуры объекта.

DROP – удаление объекта.

Общий вид синтаксиса команд DDL:

create
alter *тип_объекта имя_объекта [параметры];*
drop

Создание таблиц

Упрощенный синтаксис, диалект PostgreSQL:

CREATE TABLE [IF NOT EXISTS] *имя_таблицы*
(*имя_поля* *тип_данных* [(*размер*)]
[ограничения_*целостности_поля*…]
.,..
[, *ограничения_целостности_таблицы* .,..]
)
[*параметры*];

ограничения_целостности (ОЦ):

[CONSTRAINT *constraint_name*]
{ NOT NULL | NULL |
CHECK (*выражение*) |
DEFAULT *выражение* |
UNIQUE [(*поле1* [,…, *полеN*])] |
PRIMARY KEY [(*поле1* [,…, *полеN*])] |
REFERENCES *имя_таблицы* [([(*поле1* [,…, *полеN*])])]
[ON DELETE *action*] [ON UPDATE *action*] }

Типы данных PostgreSQL

Числовые типы:

Имя	Размер	Описание	Диапазон
smallint	2 байта	целое в небольшом диапазоне	-32768 .. +32767
integer	4 байта	типичный выбор для целых чисел	-2147483648 .. +2147483647
bigint	8 байт	целое в большом диапазоне	-9223372036854775808 .. 9223372036854775807
decimal	переменный	вещественное число с указанной точностью	до 131072 цифр до десятичной точки и до 16383 — после
numeric	переменный	вещественное число с указанной точностью	до 131072 цифр до десятичной точки и до 16383 — после
real	4 байта	вещественное число с переменной точностью	точность в пределах 6 десятичных цифр
double precision	8 байт	вещественное число с переменной точностью	точность в пределах 15 десятичных цифр
smallserial	2 байта	небольшое целое с автоувеличением	1 .. 32767
serial	4 байта	целое с автоувеличением	1 .. 2147483647
bigserial	8 байт	большое целое с автоувеличением	1 .. 9223372036854775807

Типы данных PostgreSQL: символьные

Имя	Описание
character varying(*n*), varchar(*n*)	строка ограниченной переменной длины
character(*n*), char(*n*)	строка фиксированной длины, дополненная пробелами
text	строка неограниченной переменной длины

Длина ***n*** указана в символах, а не в байтах!
Для типа CHAR значение ***n*** по умолчанию 1. При при сравнении двух значений типа CHAR дополняющие пробелы игнорируются, при приведении значения CHAR к другому символьному типу дополняющие пробелы отбрасываются.
Для типа VARCHAR максимальный размер 1 Гб, а значение ***n*** по умолчанию не существует. Если не указывать ***n***, в Postgres Pro этот тип будет принимать строки любого размера, что не соответствует стандарту SQL.
Попытка сохранить в столбце символьного типа более длинную строку приведёт к ошибке, если только все лишние символы не являются пробелами (тогда они будут усечены до максимально допустимой длины).
При попытке явно привести значение к типу CHAR или VARCHAR часть строки, выходящая за границу в ***n*** символов, удаляется, не вызывая ошибки.

Типы данных PostgreSQL: календарные

Имя	Размер	Описание	Наименьшее значение	Наибольшее значение	Точность
timestamp [(*p*)] [without time zone]	8 байт	дата и время (без часового пояса)	4713 до н. э.	294276 н. э.	1 мкс
timestamp [(*p*)] with time zone	8 байт	дата и время (с часовым поясом)	4713 до н. э.	294276 н. э.	1 мкс
date	4 байта	дата (без времени суток)	4713 до н. э.	5874897 н. э.	1 день
time [(*p*)] [without time zone]	8 байт	время суток (без даты)	00:00:00	24:00:00	1 мкс
time [(*p*)] with time zone	12 байт	время дня (без даты), с часовым поясом	00:00:00+1459	24:00:00-1459	1 мкс
interval [*поля*] [(*p*)]	16 байт	временной интервал	-178000000 лет	178000000 лет	1 мкс

Необязательное значение точности ***p***, определяющее, сколько знаков после запятой должно сохраняться в секундах. По умолчанию точность не ограничивается. Допустимые значения ***p*** лежат в интервале от 0 до 6.

Значения даты и времени принимаются в любом разумном формате (в виде строки):
'2019-02-25', '20/04/2016', ‘January 8, 1999’, '10:15:00‘, ‘04:05 PM’

(current_date + 1) – завтра;

(<дата1> – <дата2>) – количество дней, прошедших между двумя датами;

(current_time + '1 hour') – через час;

(current_time – '20 minute') – 20 минут назад.

Ограничения целостности

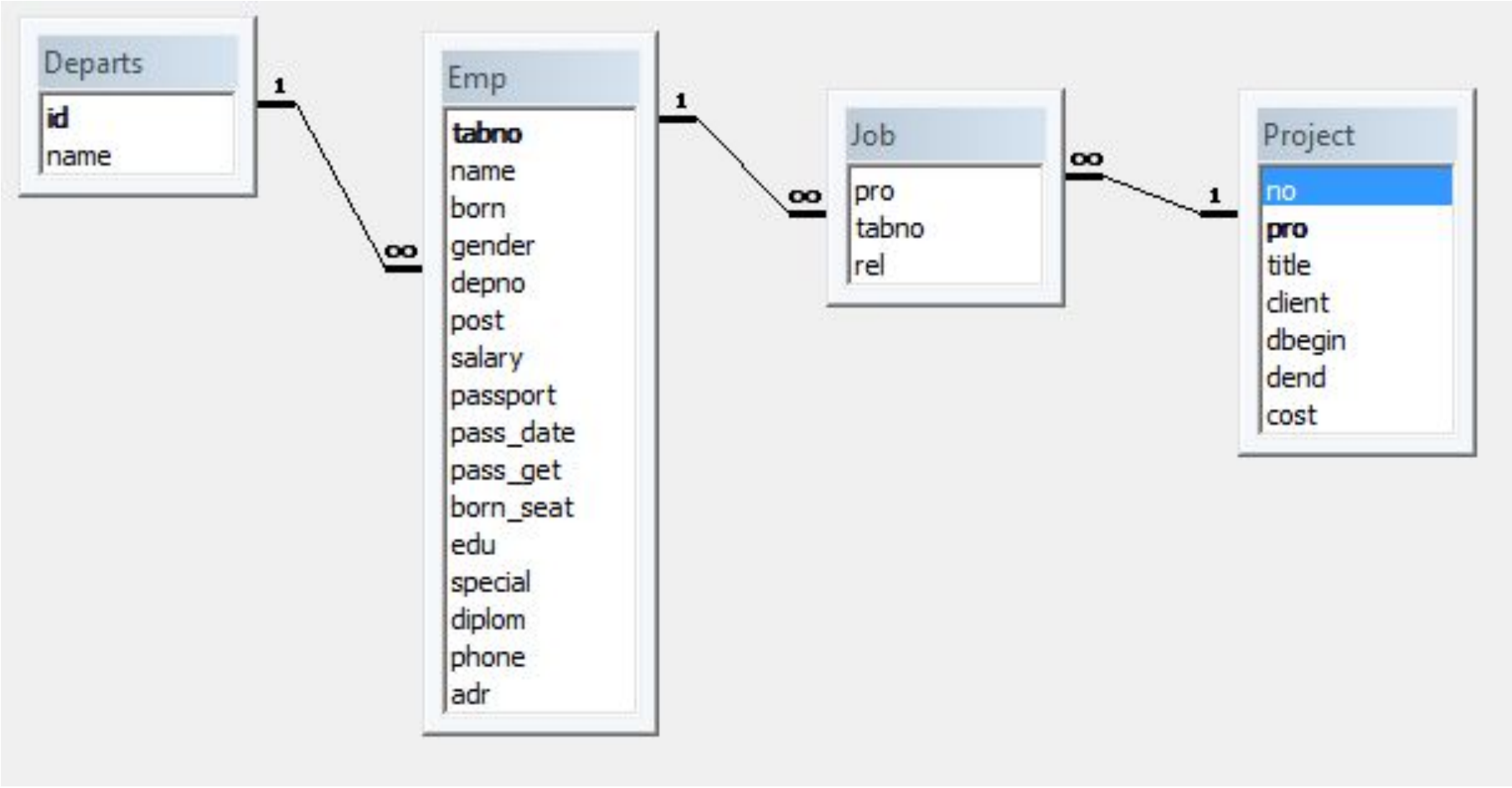
В СУБД Postgres в соответствии со стандартом SQL-92 поддерживаются следующие ограничения целостности:

- обязательность / необязательность:
  NOT NULL / NULL
- уникальность (значений атрибута или комбинации значений атрибутов):
  UNIQUE (*имя_атрибута1* [, *имя_атрибута2*,...])
- первичный ключ:
  PRIMARY KEY(*имя_атрибута1* [, *имя_атрибута2*,...])
- условие на значение поля:
  CHECK (*условие*)
  Например: check (salary>=12000), check (date2 > date1)
- внешний ключ:
  FOREIGN KEY(*имя_атрибута1* [, *имя_атрибута2*,...]) **REFERENCES**
  имя_таблицы [(*имя_атрибута1* [, *имя_атрибута2*,...])]
  [ON DELETE *action*] [ON UPDATE *action*]

action – действие, производимое при удалении строки (обновлении значения уникального поля) в родительской таблице:

NO ACTION | RESTRICT | CASCADE | SET NULL | SET DEFAULT

Пример БД: проектная организация



Departs – отделы, Project – проекты,

Emp – сотрудники, 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;
gender – 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 numeric(4) constraint pk_depart  PRIMARY KEY,
            name varchar(100) not null
);
```

Таблица «Сотрудники» (Emp):

```
create table emp (
            tabno       numeric(5) constraint pk_emp PRIMARY KEY,
      name       varchar(100) not null,
      born          date not null,
      gender       char   not null,
      depno       numeric(4) not null constraint fk_depart  REFERENCES depart,
      post          varchar(50) not null,
      salary        numeric(8,2) not null constraint check_sal  check (salary > 12000),
      passport     char(10) not null constraint passp_uniq  UNIQUE,
      pass_date  date not null, pass_get   varchar(100) not null,
      born_seat   varchar(100),edu          varchar(30),
      special       varchar(100),       diplom       varchar(40),
      phone         varchar(30),  adr           varchar(80) not null,
      edate         date  not null  default   current_date,
      chief          numeric(5)  constraint  fk_emp  REFERENCES  emp
);
```

Создание таблиц БД проектной организации

Таблица «Проекты» (Project):

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

Таблица «Участие в проектах» (Job):

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

Пример описания составного внешнего ключа

Таблица «Этапы проектов» (Stages):

```
create table stages (No numeric(2),
            pro     varchar(15)     references  project (abbr),
            dbegin date             not null,
            form    varchar(100)  not null,
            cost     numeric(9)     check(cost>0)  default 100000,
primary key (pro, No)
);
```

Таблица «Участие в этапах проектов» (Participants):

```
create table  participants
(           No      numeric(2)     not null,
            pro      varchar(15)   not null,
            tabNo numeric(6)     not null  references  emp,
            dbegin date             not null,
            dend    date,
      foreign  key (pro, No) references  project (pro, No)
);
```

Изменение структуры таблицы

Структура таблицы изменяется с помощью команды **ALTER TABLE.**
Синтаксис этой команды (для диалекта Oracle):

ALTER TABLE shema . Tablename
ADD (column datatype | DEFAULT expr | column constraint)
MODIFY (column datatype | DEFAULT expr | column constraint)
DROP COLUMN (column)
STORAGE storage options

Где:

- **ADD** - опция добавления элемента таблицы.
  - **column datatype** - тип данных столбцов таблицы.
  - DEFAULT expr - значение по умолчанию.
  - column constraint - ограничение столбца.
- **MODIFY** - опция изменения элемента таблицы.
  - **column datatype** - тип данных столбцов таблицы.
  - **DEFAULT expr** - значение по умолчанию.
  - **column constraint** - ограничение столбца.
- **DROP COLUMN column** - удаление столбца.
- **STORAGE storage options** - параметры хранения таблицы.

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* };

Для ясности можно также явно указать значения по умолчанию для отдельных столбцов или всей строки:

INSERT INTO <имя таблицы> VALUES (DEFAULT, DEFAULT);
INSERT INTO <имя таблицы> VALUES DEFAULT VALUES;

Одна команда может вставить сразу несколько строк:

INSERT INTO <имя таблицы> VALUES (), (), ();

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, gender, depno, passport, pass_date_pass_get,
      post, salary, phone)
    values( 301, 'САВИН АНДРЕЙ ПАВЛОВИЧ',  '1969.11.07',
      'М', 5, '4405092876', '1999.10.02',  'ОВД "Митино" г.Москвы',
      'программист', 68050, '915-121-34-11');
```

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 = ‘АИС Налог-2';
```

-- 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 вставку/изменение/удаление данных можно командой

ROLLBACK;