



Базы данных

Язык запросов SQL.
Введение



SQL – Structured Query Language

- **SQL** – это структурированный язык запросов к реляционным базам данных (БД).
- 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, SQL/2003) – относится к объектно-реляционной модели данных.
- Подмножества языка SQL:
 - ✓ **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.



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 заключаются в одинарные кавычки:
 'A', '2', 'строка', 'другая строка'
 - ✓ Однострочный комментарий начинается с символов '--'.
 - ✓ Многострочный комментарий заключается в символы /* ... */.
- Запуск MySQL: клиент mysql запускается с использованием программы «Командная строка»:

```
mysql -uимя_пользователя -рпароль
```

```
mysql> use имя_базы_данных;
```



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 ,... ]
  )
[ параметры ];
```

ограничения_целостности (ОЦ):

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



Типы данных

- Символьные типы:
 - ✓ **CHAR** [(длина)] – строка фиксированной длины.
Длина по умолчанию – 1, максимальная длина 2000 б.
Строка дописывается до указанной длины пробелами.
 - ✓ **VARCHAR2** (длина) – строка переменной длины.
Максимальная длина 4000 б. Хранятся только значащие символы.
- Числовой тип:
 - ✓ **NUMBER** [(точность[, масштаб])] – используется для представления чисел с заданной точностью.
Точность по умолчанию 38, масштаб по умолчанию – 0.
number(4) – числа от -999 до 9999
number(8,2) – числа от -99999.99 до 999999.99
- **DATE** – дата и время с точностью до секунды. Занимает 7 байт.
 - ✓ sysdate – функция получения текущих даты и времени.
 - ✓ Тип date поддерживает арифметику дат:
sysdate+1 – завтра
(дата1 – дата2) – количество дней, прошедших между двумя датами
(sysdate – 0.5) – 12 часов назад

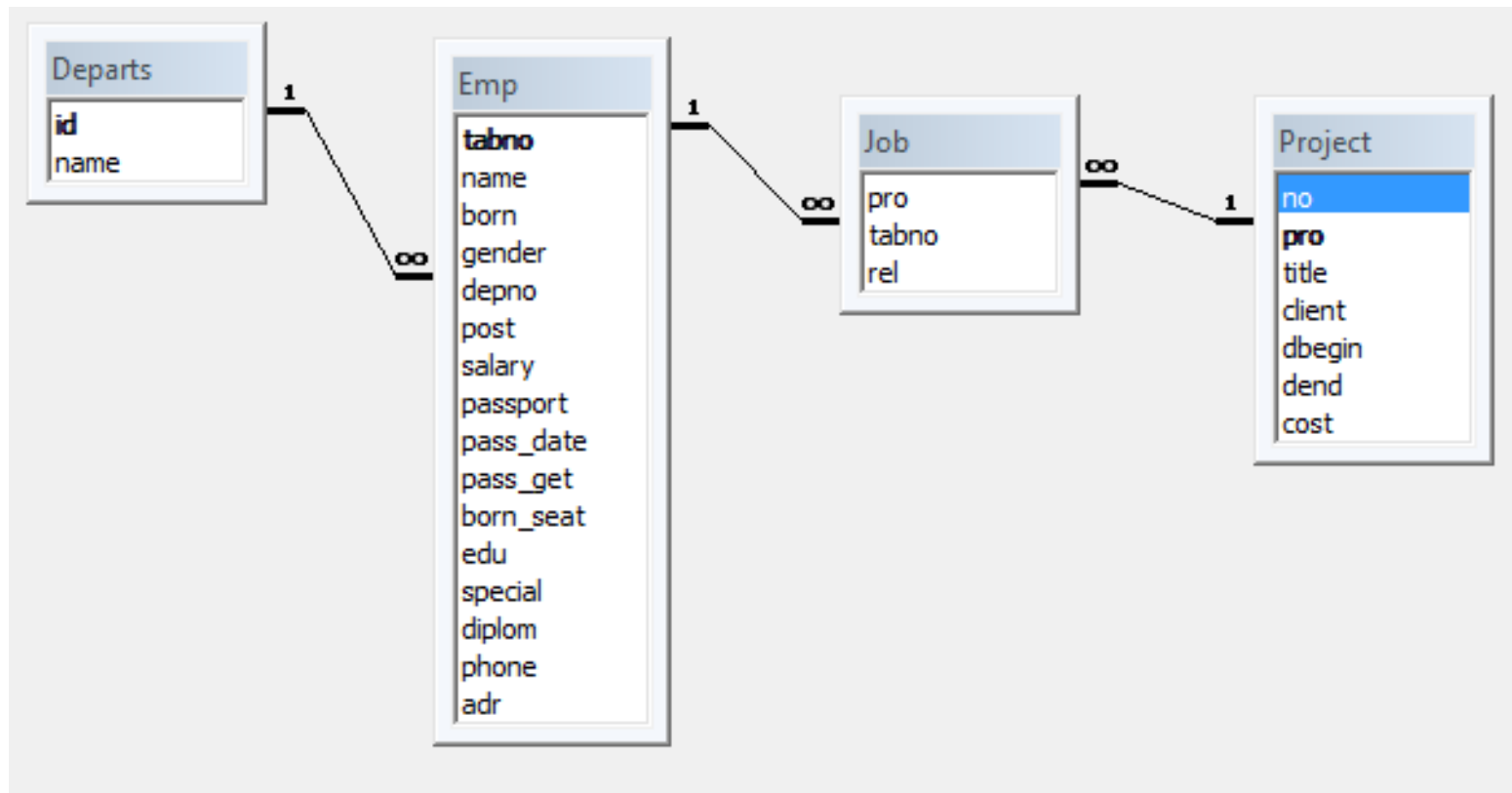


Ограничения целостности

В СУБД Oracle поддерживаются следующие ограничения целостности:

- ✓ уникальность (значений атрибута или комбинации значений атрибутов):
UNIQUE (*имя_атрибута1* [, *имя_атрибута2*,...])
- ✓ обязательность / необязательность:
NOT NULL / NULL
- ✓ первичный ключ:
PRIMARY KEY(*имя_атрибута1* [, *имя_атрибута2*,...])
- ✓ внешний ключ:
FOREIGN KEY(*имя_атрибута1* [, *имя_атрибута2*,...]) **REFERENCES**
имя_таблицы [(*имя_атрибута1* [, *имя_атрибута2*,...])]
- ✓ условие на значение поля:
CHECK (*условие*)
Например: check (salary>=4500), check (date2 > date1)

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;

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 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,  
                  gender char not null,  
                  depno number(4) not null constraint fk_depart REFERENCES depart,  
                  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') )
);
```



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 };
```

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', to_date('11.07.1969', 'dd.mm.yyyy'),  
        'М', 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;