

基础 SQL

数据类型:

- **char**(n): 固定长度的字符串, 用户指定长度 n 。也可以使用全称 **character**。
- **varchar**(n): 可变长度的字符串, 用户指定最大长度 n , 等价于全称 **character varying**。
- **int**: 整数类型(和机器相关的整数的有限子集), 等价于全称 **integer**。
- **smallint**: 小整数类型(和机器相关的整数类型的子集)。
- **numeric**(p, d): 定点数, 精度由用户指定。这个数有 p 位数字(加上一个符号位), 其中 d 位数字在小数点右边。所以在一个这种类型的字段上, **numeric**(3, 1)可以精确储存 44.5, 但不能精确存储 444.5 或 0.32 这样的数。
- **real, double precision**: 浮点数与双精度浮点数, 精度与机器相关。
- **float**(n): 精度至少为 n 位的浮点数。

create table

create table 命令的通用形式是:

```
create table r
(A1 D1,
 A2 D2,
 ...,
 An Dn,
 <完整性约束1>,
 ...,
 <完整性约束k>);
```

查询并去重

select delete ... from ...

排序

select ... from ... order by ... asc(升)
desc(降)

更新数据

update 表名 set 列 = 新值 where 列 = 某值

删除表中的行

delete from 表 where 列 = 某值

删除表中所有数据

delete from 表

删表

drop table 表

提取带有特征的行 (like)

select 列 from 表 where 列 (not)

like ' %g % ' 包含 on
h % 以 h 开头

in ('g', 'on', 'h')

between A and B
[A, B)

别名

as

① 可省略 as

② 表名可不' '，列别名必加' ')

连接

select 列

from 表 A

inner / left / right / full join 表 B

on A.主键 = B.主键

```
select name, course_id
from instructor, teaches
where instructor.ID = teaches.ID;
```

该查询可以用 SQL 的自然连接运算更简洁地写作：

```
select name, course_id
from instructor natural join teaches;
```

```
select name, title
from instructor natural join teaches, course
where teaches.course_id = course.course_id;
```

```
select name, title
from instructor natural join teaches natural join course;
```

为了说明原因，注意 *instructor* 和 *teaches* 的自然连接包括属性 (*ID*, *name*, *dept_name*, *salary*, *course_id*, *sec_id*)，而 *course* 关系包含的属性是 (*course_id*, *title*, *dept_name*, *credits*)。作为这二者自然连接的结果，

⊖ 其结果是不可能用包含了原始关系名的属性名来指代自然连接结果中的属性，例如 *instructor.name* 或 *teaches.course_id*，但是我们可以使用诸如 *name* 和 *course_id* 那样的属性名，而不带关系名。

40 第一部分 关系数据库

需要来自这两个输入的元组既要在属性 *dept_name* 上取值相同，还要在 *course_id* 上取值相同。该查询将忽略所有这样的 (教师姓名, 课程名称) 对：其中教师所讲授的课程不是他所在系的课程。而前一个查询会正确输出这样的对。

```
select name, title
from (instructor natural join teaches) join course using (course_id);
```