【MySQL】约束(三)

> 🚚MySQL学习·第三站~
> 🚩本文已收录至专栏：MySQL通关路
> ❤️每章节附章节思维导图，文末附全文思维导图，感谢各位点赞收藏支持~
> ⭐学习汇总贴，超详细思维导图：【MySQL】学习汇总(完整思维导图)

一.引入

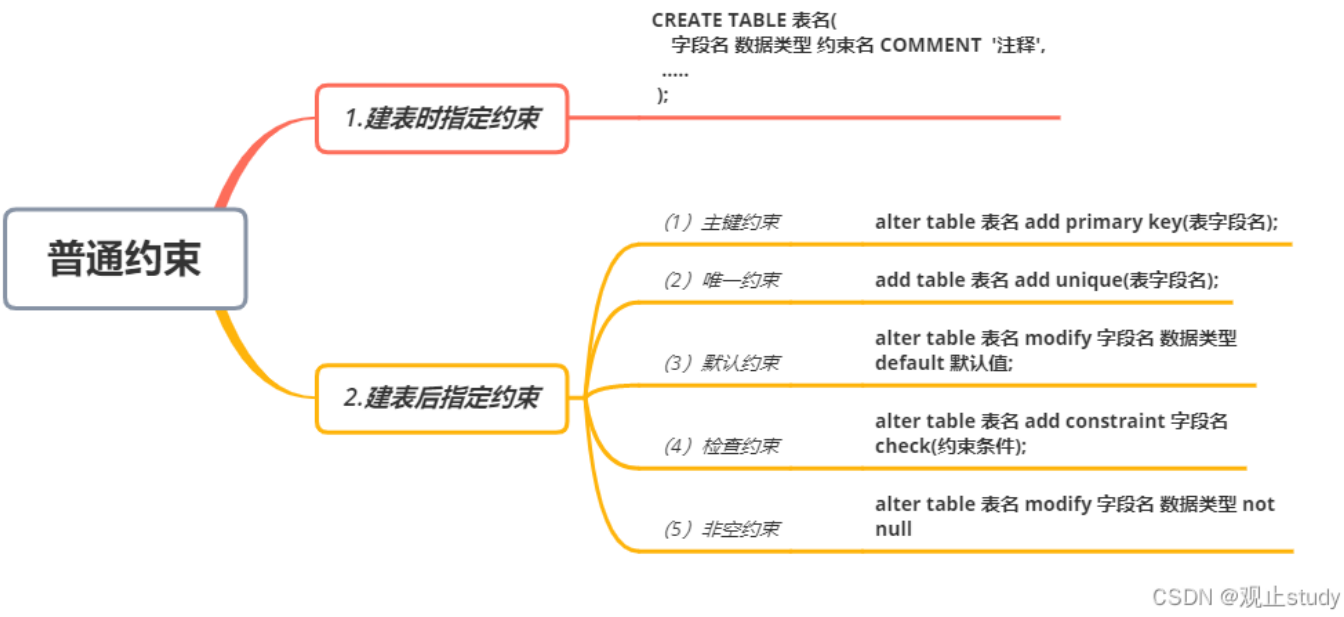
约束是**作用于表中字段上**的**规则**，用于**限制存储在表中的数据**。 使用约束可以保证数据库中数据的正确、有效性以及完整性。我们可以在创建表或修改表的时候在表字段上添加约束。

约束分为如下几种：

约束	描述	关键字
非空约束	限制该字段的数据不能为null	`NOT NULL`
唯一约束	保证该字段的所有数据都是唯一、不重复的、可以为空	`UNIQUE`
主键约束	主键是一行数据的唯一标识，要求非空且唯一	`PRIMARY KEY`
默认约束	保存数据时，如果未指定该字段的值，则采用默认值	`DEFAULT`
检查约束	插入数值时根据指定条件校验合法性(8.0.16版本之后开始)	`CHECK`
外键约束	用来让两张表的数据之间建立连接，保证数据的一致性和完整性	`FOREIGN KEY`

我们先通过一个示例介绍前五个约束，最后再介绍外键约束~

二.普通约束



(1) 示例说明

在开发过程中，如果我们想要持久化存储数据就不可避免的需要建立相关表。而建表以后对于插入其中的数据，如果我们后台没有校验，数据库也没有约束的话将显得十分混乱，接下来我们通过指定约束来规范数据。

字段名	字段含义	字段类型	约束条件	约束关键字
id	ID唯一标识	int	主键，并且自动增长	PRIMARY KEY, AUTO_INCREMENT
name	姓名	varchar(10)	不为空，并且唯一	NOT NULL , UNIQUE
age	年龄	int	大于0，并且小于等 于120	CHECK
status	状态	char(1)	如果没有指定该值， 默认为1	DEFAULT
test	测试	int	无约束对比测试字段	无

(2) 建表时指定约束

在创建表的时候为字段添加约束时，我们只需要**在字段类型之后加上约束的关键字**即可。例如：

```
CREATE TABLE user(
    id int AUTO_INCREMENT PRIMARY KEY  COMMENT  'ID唯一标识',
    name varchar(10) NOT NULL UNIQUE  COMMENT  '姓名' ,
    age int check (age > 0 && age <= 120)  COMMENT  '年龄' ,
    status char(1) default  '1'  COMMENT  '状态',
    test int  COMMENT  '无约束对比测试字段'
 );
```

特别说明：

- 当我们使用`check`约束时，约束条件必须写在`()`中。
- 当我们使用`default`约束指定默认值时，**默认值必须符合字段数据类型**。
- 每个表只能为一个字段指定主键`PRIMARY KEY`约束。
- 在一些正常情况下，**可以在一个字段上使用多个约束条件**。

```
mysql> CREATE TABLE user(
    ->     id int AUTO_INCREMENT PRIMARY KEY  COMMENT  'ID唯一标识',
    ->     name varchar(10) NOT NULL UNIQUE  COMMENT  '姓名' ,
    ->     age int check (age > 0 && age <= 120)  COMMENT  '年龄' ,
    ->     status char(1) default  '1'  COMMENT  '状态',
    ->     test int  COMMENT  '无约束对比测试字段'
    -> );
Query OK, 0 rows affected, 1 warning (0.03 sec)

mysql> desc user;
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| id     | int         | NO   | PRI | NULL    | auto_increment |
| name   | varchar(10) | NO   | UNI | NULL    |                |
| age    | int         | YES  |     | NULL    |                |
| status | char(1)     | YES  |     | 1       |                |
| test   | int         | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)                          CSDN @观止study
```

(3) 建表后指定约束

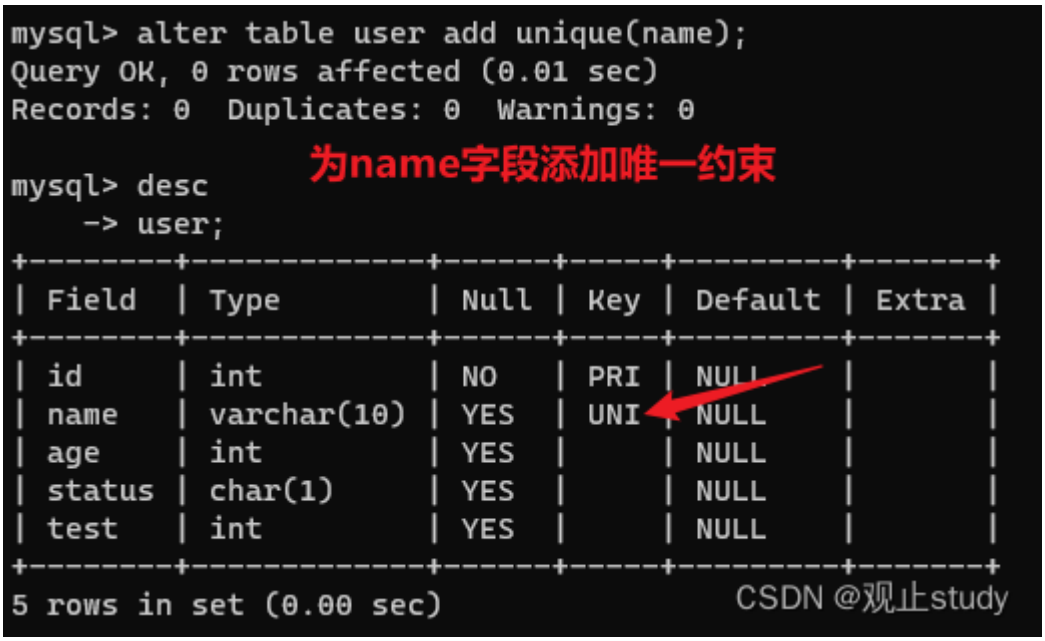
我们重新创建一张没有约束的user表来演示~

- 添加**主键约束**（`primary key`）

```
alter table 表名 add primary key(表字段名);
```

- 添加**唯一约束**（`unique`）

```
add table 表名 add unique(表字段名);
```



- 添加**非空约束**(`not null`)

```
alter table 表名 modify 字段名 数据类型 not null
```

```
mysql> alter table user modify name varchar(10) not null;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

不允许name为null

```
mysql> desc user;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| id     | int         | NO   | PRI | NULL    |       |
| name   | varchar(10) | NO   | UNI | NULL    |       |
| age    | int         | YES  |     | NULL    |       |
| status | char(1)     | YES  |     | NULL    |       |
| test   | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```


- 添加**默认约束**（`default`）

```
alter table 表名 modify 字段名 数据类型 default 默认值;
```

```
mysql> alter table user modify status char(1) default '观';
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

为status设置未填默认值观

```
mysql> desc user;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| id     | int         | NO   | PRI | NULL    |       |
| name   | varchar(10) | NO   | UNI | NULL    |       |
| age    | int         | YES  |     | NULL    |       |
| status | char(1)     | YES  |     | 观      |       |
| test   | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```


- 添加**检查约束**(`check`)

```
alter table 表名 add constraint 字段名 check(约束条件);
```

```
mysql> alter table user add constraint age check(age>=0&&age<=100)
Query OK, 0 rows affected, 1 warning (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 1
```

约束年龄范围

(4) 插入测试

上述我们已经完成了约束的创建，接下来我们来测试一下约束是否生效~

- 主键 + 自增

```
mysql> insert into user(name,age,status) values ('Tom1',19,'1'),
    ->  ('Tom2',25,'0');
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from user;
+----+------+------+--------+------+
| id | name | age  | status | test |
+----+------+------+--------+------+
|  1 | Tom1 |   19 | 1      | NULL |
|  2 | Tom2 |   25 | 0      | NULL |
+----+------+------+--------+------+
2 rows in set (0.00 sec)
```

不指定id自动增长且唯一

对比字段，可为null

- 唯一 + 非空

```
mysql> insert into user(name,age,status) values ('Tom1',19,'1');
ERROR 1062 (23000): Duplicate entry 'Tom1' for key 'user.name'
mysql> insert into user(name,age,status) values ('Tom3',19,'1');
Query OK, 1 row affected (0.00 sec)

mysql> insert into user(name,age,status) values ('Tom3',19,'1');
ERROR 1062 (23000): Duplicate entry 'Tom3' for key 'user.name'
mysql> insert into user(age,status) values (19,'1');
ERROR 1364 (HY000): Field 'name' doesn't have a default value
mysql> select * from user;
+----+------+------+--------+------+
| id | name | age  | status | test |
+----+------+------+--------+------+
|  1 | Tom1 |   19 | 1      | NULL |
|  2 | Tom2 |   25 | 0      | NULL |
|  7 | Tom3 |   19 | 1      | NULL |
+----+------+------+--------+------+
3 rows in set (0.00 sec)
```

首次插入成功

不能重复

不能为null

- 默认约束

```
mysql> insert into user(name,age,status) values ('Tom4',19,'0');
Query OK, 1 row affected (0.00 sec)

mysql> insert into user(name,age) values ('Tom5',19);
Query OK, 1 row affected (0.00 sec)

mysql> select * from user;
+----+------+------+--------+------+
| id | name | age  | status | test |
+----+------+------+--------+------+
|  1 | Tom1 |   19 | 1      | NULL |
|  2 | Tom2 |   25 | 0      | NULL |
|  7 | Tom3 |   19 | 1      | NULL |
| 13 | Tom4 |   19 | 0      | NULL |
| 14 | Tom5 |   19 | 1      | NULL |
+----+------+------+--------+------+
5 rows in set (0.00 sec)
```

可以指定值

不指定则是默认值

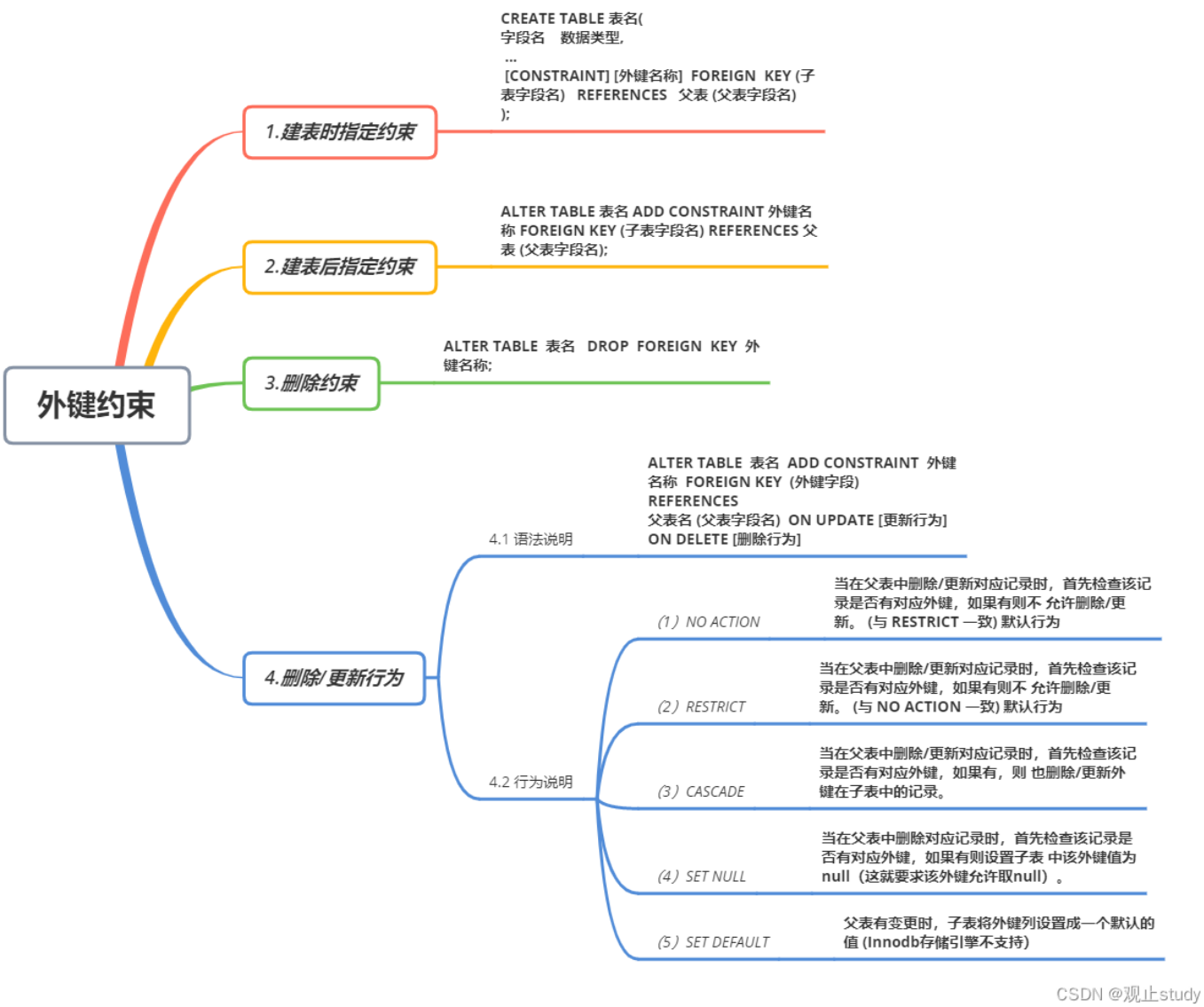
- 检查约束

```
mysql> insert into user(name,age,test) values ('Tom5',69,-69);
Query OK, 1 row affected (0.00 sec)

mysql> insert into user(name,age,test) values ('Tom5',-1,-69);
ERROR 3819 (HY000): Check constraint 'user_chk_1' is violated.
mysql> insert into user(name,age,test) values ('Tom5',130,-69);
ERROR 3819 (HY000): Check constraint 'user_chk_1' is violated.
mysql> insert into user(name,age,test) values ('Tom1',2,130);
Query OK, 1 row affected (0.00 sec)

mysql> select * from user;
+----+------+------+--------+------+
| id | name | age  | status | test |
+----+------+------+--------+------+
| 17 | Tom5 |   69 | 1      |  -69 |
| 18 | Tom1 |    2 | 1      |  130 |
+----+------+------+--------+------+
2 rows in set (0.00 sec)
```

可以看到age必须是约束内的值，
而未指定约束字段则可以随意设置

CSDN @观止study

三.外键约束



CSDN @观止study

> 外键约束通过作用于两张表之间，用于相互约束彼此的行为，从而保证数据的一致性和完整性。

(1) 引入

例如下述，左侧的emp表是员工表，里面存储员工的基本信息，包含员工的ID、姓名、年龄、职位、薪资、入职日 期、上级主管ID、部门ID，在员工的信息中存储的是部门的ID dept_id，而这个部门的ID是关联的 部门表dept的主键id，emp表的dept_id就是外键,关联的是另一张dept表的主键。

id	name	age	job	salary	entrydate	managerid	dept_id
1	金庸	66	总裁	20000	2000-01-01	<null>	5
2	张无忌	20	项目经理	12500	2005-12-05	1	1
3	杨逍	33	开发	8400	2000-11-03	2	1
4	韦一笑	48	开发	11000	2002-02-05	2	1
5	常遇春	43	开发	10500	2004-09-07	3	1

员工表 emp (子表)

id	name
1	研发部
2	市场部
3	财务部
4	销售部
5	总经办

部门表 dept (父表)

我们可以看出这两张表之间存在着一种关联关系，但它们只是在逻辑上存在这样一层关系；在数据库层面，并未建立外键关联。也就说emp表中的dept_id值可以为任意数值，即是dept表中不存在。而dept表中字段被emp使用了也可以随意删除。因此无法保证数据的一致性和完整性的。这时候就需要我们严格的进行手动维护或者**使用外键约束**。

(2) 添加外键

特别说明：

- 添加外键时必须得确保需要关联的父表已经创建
- 一个表可存在多个外键，且可以关联多个表

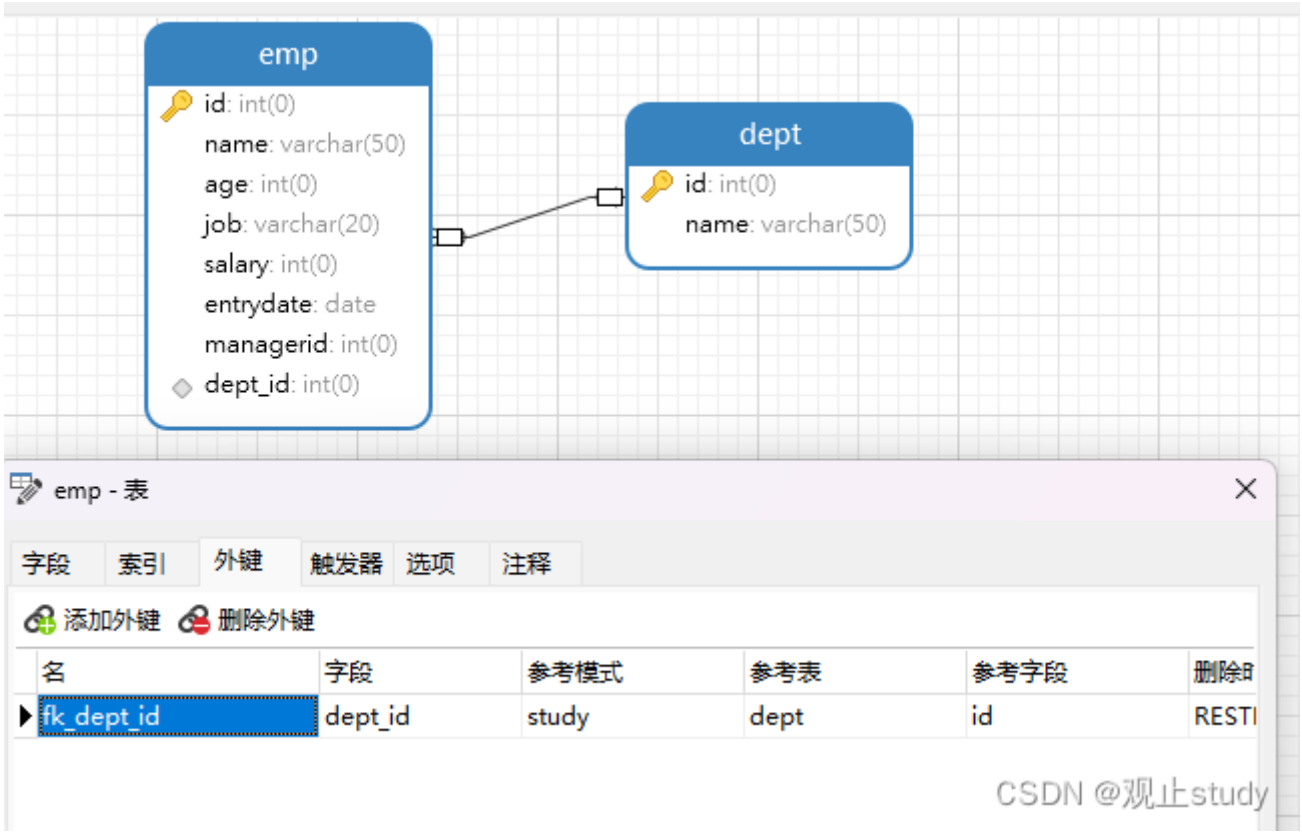
(2.1) 建表时添加

- 语法说明

```
CREATE TABLE 表名(
字段名    数据类型,
 ...
 [CONSTRAINT] [外键名称]  FOREIGN  KEY (子表字段名)   REFERENCES   父表 (父表字段名)
);
```

- 示例演示

```
# 必须先创建需要关联的父表 (dept)
create table emp(
 id  int auto_increment comment 'ID' primary key,
 name varchar(50) not null comment '姓名',
 age  int comment '年龄',
 job varchar(20) comment '职位',
 salary int comment '薪资',
 entrydate date comment '入职时间',
 managerid int comment '直属领导ID',
 dept_id int comment '部门ID',
 CONSTRAINT fk_dept_id foreign key (dept_id) references dept(id)
 )comment '员工表';
```

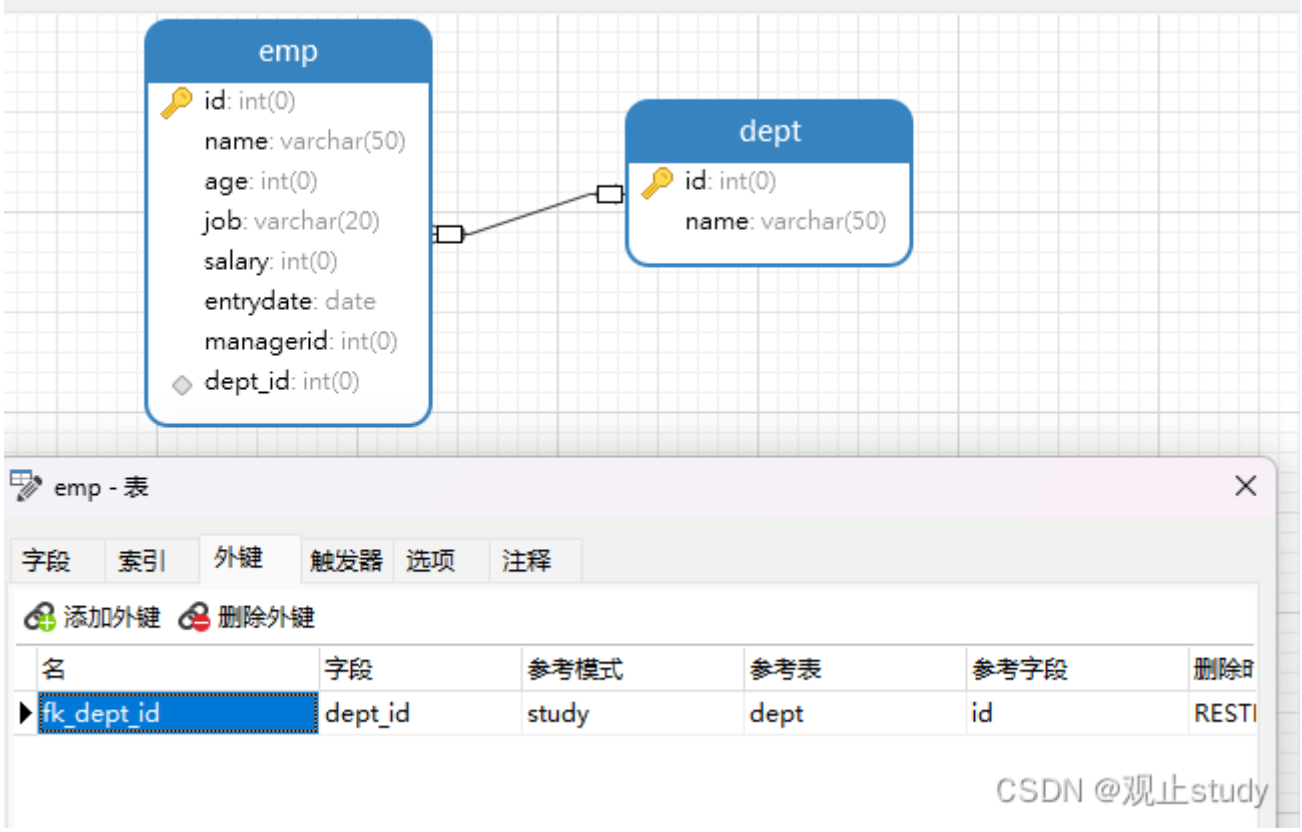


(2.2) 建表后添加

- 语法说明

```
ALTER TABLE 表名 ADD CONSTRAINT 外键名称 FOREIGN KEY (子表字段名) REFERENCES 父表 (父表字段名);
```

- 示例演示



(2.3) 外键测试

添加外键约束后我们无法随意删除父表中的数据，必须先删除所有子表中关联到改记录的数据才能删除父表中数据。

```
mysql> delete from dept where id = 1;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`study`.`emp`, CONSTRAINT `fk_
dept_id` FOREIGN KEY (`dept_id`) REFERENCES `dept` (`id`))
mysql> delete from emp where dept_id = 1;
Query OK, 5 rows affected (0.00 sec)

mysql> delete from dept where id = 1;
Query OK, 1 row affected (0.00 sec)
```

emp中使用了该记录无法直接删除dept表中记录

必须先清除emp表的关联数据

然后才能删除成功dept表中数据

(3) 删除外键

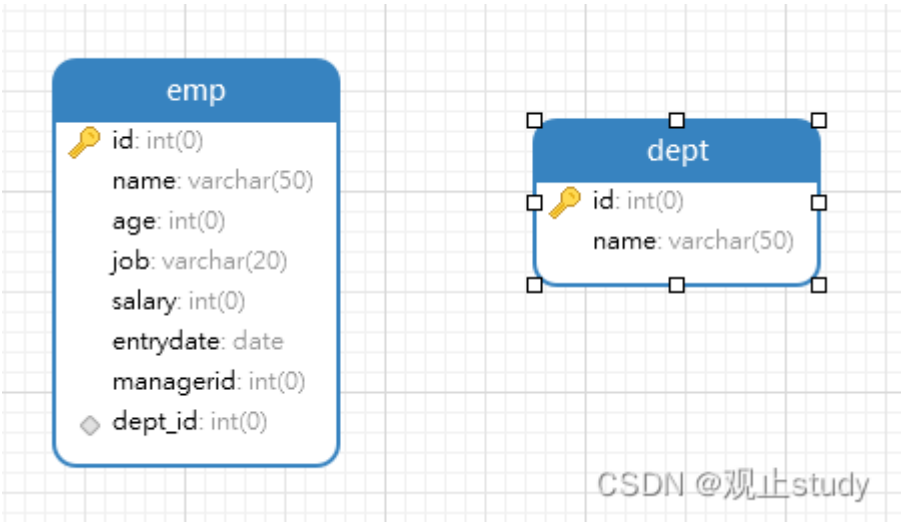
- 语法

```
ALTER TABLE  表名  DROP  FOREIGN  KEY  外键名称;
```

- 示例演示

```
mysql> alter table emp drop foreign key fk_dept_id;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

删除外键



(4) 删除/更新行为

在添加了外键之后，再删除**父表数据**时产生的约束行为，我们就称为删除/更新行为。具体的删除/更新行为有以下几种:

行为	说明
`NO ACTION`	当在父表中删除/更新对应记录时，首先检查该记录是否有对应外键，如果有则不 允许删除/更新。 (与 RESTRICT 一致) **默认行为**
`RESTRICT`	当在父表中删除/更新对应记录时，首先检查该记录是否有对应外键，如果有则不 允许删除/更新。 (与 NO ACTION 一致) **默认行为**
`CASCADE`	当在父表中删除/更新对应记录时，首先检查该记录是否有对应外键，如果有，则 也删除/更新外键在子表中的记录。
`SET NULL`	当在父表中删除对应记录时，首先检查该记录是否有对应外键，如果有则设置子表 中该外键值为null（这就要求该外键允许取null）。
`SET DEFAULT`	父表有变更时，子表将外键列设置成一个默认的值 (Innodb存储引擎不支持)

注意事项：`NO ACTION`与`RESTRICT`为外键约束的默认行为，也就是在一些情况下阻止我们删除/更新数据，添加外键时默认生效。如果需要修改为其他几种行为则需要我们在添加外键时手动设置。

- 修改删除/更新语法：

```
ALTER TABLE 表名 ADD CONSTRAINT 外键名称 FOREIGN KEY (外键字段) REFERENCES
父表名 (父表字段名) ON UPDATE [更新行为] ON DELETE [删除行为]
```

- `CASCADE`行为演示

```
mysql> alter table emp add constraint fk_emp_dept_id foreign key (dept_id) references
    -> dept(id) on update cascade on delete cascade ;
Query OK, 6 rows affected (0.04 sec)
Records: 6  Duplicates: 0  Warnings: 0          修改更新/删除行为同步

mysql> update dept set id = 9999 where name = '研发部';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0        修改父表id值

mysql> select * from dept;
+------+-----------+
| id   | name      |
+------+-----------+
|    2 | 市场部    |
|    3 | 财务部    |
|    4 | 销售部    |
|    5 | 总经办    |      父表成功修改
| 9999 | 研发部    |
+------+-----------+
5 rows in set (0.00 sec)

                                                子表自动更新外键值
mysql> select * from emp;
+----+----------+------+--------------------+--------+------------+-----------+---------+
| id | name     | age  | job                | salary | entrydate  | managerid | dept_id |
+----+----------+------+--------------------+--------+------------+-----------+---------+
|  1 | 金庸     |   66 | 总裁               |  20000 | 2000-01-01 |      NULL |       5 |
|  2 | 张无忌   |   20 | 项目经理           |  12500 | 2005-12-05 |         1 |    9999 |
|  3 | 杨逍     |   33 | 开发               |   8400 | 2000-11-03 |         2 |    9999 |
|  4 | 韦一笑   |   48 | 开
发             |  11000 | 2002-02-05 |         2 |    9999 |
|  5 | 常遇春   |   43 | 开发               |  10500 | 2004-09-07 |         3 |    9999 |
|  6 | 小昭     |   19 | 程
序员鼓励师     |   6600 | 2004-10-12 |         2 |    9999 |
+----+----------+------+--------------------+--------+------------+-----------+---------+
6 rows in set (0.00 sec)
```

```
mysql> delete from dept where id = 9999;
Query OK, 1 row affected (0.00 sec)

mysql> select * from emp;
+----+--------+------+--------+--------+------------+-----------+---------+
| id | name   | age  | job    | salary | entrydate  | managerid | dept_id |
+----+--------+------+--------+--------+------------+-----------+---------+
|  1 | 金庸   |   66 | 总裁   |  20000 | 2000-01-01 |      NULL |       5 |
+----+--------+------+--------+--------+------------+-----------+---------+
1 row in set (0.00 sec)

mysql> select * from dept;
+----+-----------+
| id | name      |
+----+-----------+
|  2 | 市场部    |
|  3 | 财务部    |
|  4 | 销售部    |
|  5 | 总经办    |
+----+-----------+
4 rows in set (0.00 sec)
```

测试删除行为

成功同步删除

- `SET NULL`行为演示

```
mysql> alter table emp add constraint fk_emp_dept_id foreign key (dept_id) reference
    -> dept(id) on update set null on delete set null;
Query OK, 1 row affected (0.04 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> update dept set id = 9999 where name = '总经办';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from emp;
+----+--------+------+--------+--------+------------+-----------+---------+
| id | name   | age  | job    | salary | entrydate  | managerid | dept_id |
+----+--------+------+--------+--------+------------+-----------+---------+
|  1 | 金庸   |   66 | 总裁   |  20000 | 2000-01-01 |      NULL |    NULL |
+----+--------+------+--------+--------+------------+-----------+---------+
1 row in set (0.00 sec)

mysql> select * from dept;
+------+-----------+
| id   | name      |
+------+-----------+
|    2 | 市场部    |
|    3 | 财务部    |
|    4 | 销售部    |
| 9999 | 总经办    |
+------+-----------+
```

修改更新/删除行为 置null

自动变为null

修改成功


mysql> select * from emp;
+----+--------+------+--------+--------+------------+-----------+---------+
| id | name | age | job | salary | entrydate | managerid | dept_id |
+----+--------+------+--------+--------+------------+-----------+---------+
| 1 | 金庸 | 66 | 总裁 | 20000 | 2000-01-01 | NULL | 9999 |
+----+--------+------+--------+--------+------------+-----------+---------+
1 row in set (0.00 sec)
mysql> delete from dept where id = 9999;
删除
Query OK, 1 row affected (0.00 sec)
mysql> select * from emp;
同样置为null
+----+--------+------+--------+--------+------------+-----------+---------+
| id | name | age | job | salary | entrydate | managerid | dept_id |
+----+--------+------+--------+--------+------------+-----------+---------+
| 1 | 金庸 | 66 | 总裁 | 20000 | 2000-01-01 | NULL | NULL |
+----+--------+------+--------+--------+------------+-----------+---------+
1 row in set (0.00 sec)
CSDN @观止study

四.全文概览