0415内置函数

5.8 外键

外键用于定义主表和从表之间的关系：外键约束主要定义在从表上，主表则必须是有主键约束或unique约束。当定义外键后，要求外键列数据必须在主表的主键列存在或为null。

语法：

```
foreign key (字段名) references 主表(列)
```

学生表(stu)

id	name	class_id
100	张三	10
101	李四	20

班级表(myclass)

id	name
10	c++大牛班
20	java大神班

如果将班级表中的数据都设计在每个学生表的后面，那就会出现数据冗余，所以我们只要设计成让stu->class_id和myclass->id形成关联的关系 => 外键约束

```
) at line 2
MySQL@<lesson5> $ create table if not exists student_tb( id bigint primary key, name
 varchar(32) not null, class_id bigint );
Query OK, 0 rows affected (0.05 sec)

MySQL@<lesson5> $ show tables;
+-------------------+
| Tables_in_lesson5 |
+-------------------+
| student_tb        |
+-------------------+
1 row in set (0.00 sec)

MySQL@<lesson5> $ desc student_tb;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| id       | bigint(20)  | NO   | PRI | NULL    |       |
| name     | varchar(32) | NO   |     | NULL    |       |
| class_id | bigint(20)  | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

MySQL@<lesson5> $
```

先创建一张表

```
MySQL@<lesson5> $ create table if not exists class_tb(
    -> id bigint primary key,
    -> name varchar(32) not null,
    -> teacher varchar(32) not null
    -> );
Query OK, 0 rows affected (0.04 sec)

MySQL@<lesson5> $ desc class_tb;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| id      | bigint(20)  | NO   | PRI | NULL    |       |
| name    | varchar(32) | NO   |     | NULL    |       |
| teacher | varchar(32) | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)

MySQL@<lesson5> $
```

准备好这两张表

```
MySQL@<lesson5> $ select * from student_tb;
+------+----------+----------+
| id   | name     | class_id |
+------+----------+----------+
| 1000 | zhangsan |      104 |
| 1001 | lisi     |      104 |
| 1002 | wangwu   |      104 |
| 1003 | zhangfei |      105 |
| 1004 | liubei   |      105 |
| 1005 | sunquan  |      105 |
+------+----------+----------+
6 rows in set (0.00 sec)

MySQL@<lesson5> $ select * from class_tb;
+-----+--------+---------+
| id  | name   | teacher |
+-----+--------+---------+
| 104 | 104C++ | pengge  |
| 105 | 105C++ | hangge  |
+-----+--------+---------+
2 rows in set (0.00 sec)

MySQL@<lesson5> $
```

```
MySQL@<lesson5> $ update student_tb set class_id=105 where id=1001;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MySQL@<lesson5> $ select * from student_tb;
+------+----------+----------+
| id   | name     | class_id |
+------+----------+----------+
| 1000 | zhangsan |      104 |
| 1001 | lisi     |      105 |
| 1002 | wangwu   |      104 |
| 1003 | zhangfei |      105 |
| 1004 | liubei   |      105 |
| 1005 | sunquan  |      105 |
+------+----------+----------+
6 rows in set (0.00 sec)

MySQL@<lesson5> $
```

调班

```
MySQL@<lesson5> $ insert into student_tb values(1006,"diaochan",108);
Query OK, 1 row affected (0.00 sec)

MySQL@<lesson5> $ select * from student_tb;
+------+----------+----------+
| id   | name     | class_id |
+------+----------+----------+
| 1000 | zhangsan |      104 |
| 1001 | lisi     |      105 |
| 1002 | wangwu   |      104 |
| 1003 | zhangfei |      105 |
| 1004 | liubei   |      105 |
| 1005 | sunquan  |      105 |
| 1006 | diaochan |      108 |
+------+----------+----------+
7 rows in set (0.00 sec)

MySQL@<lesson5> $ select * from class_tb;
+-----+--------+---------+
| id  | name   | teacher |
+-----+--------+---------+
| 104 | 104C++ | pengge  |
| 105 | 105C++ | hangge  |
+-----+--------+---------+
2 rows in set (0.00 sec)

MySQL@<lesson5> $
```

如果这样插入108期的学生

我们希望 class_tb 可以自动插入 108 期

所以按照刚才这种建表的方式
只有关系没有约束！

```
MySQL@<lesson5> $ desc class_tb;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| id      | bigint(20)  | NO   | PRI | NULL    |       |
| name    | varchar(32) | NO   |     | NULL    |       |
| teacher | varchar(32) | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

MySQL@<lesson5> $ create table stu_tb(
    -> id bigint auto_increment,
    -> name varchar(32) not null,
    -> class_id bigint not null,
    -> primary key(id),
    -> foreign key(class_id) references class_tb(id)
    -> );
Query OK, 0 rows affected (0.07 sec)

MySQL@<lesson5> $ desc stu_db;
ERROR 1146 (42S02): Table 'lesson5.stu_db' doesn't exist
MySQL@<lesson5> $ desc stu_tb;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| id       | bigint(20)  | NO   | PRI | NULL    | auto_increment |
| name     | varchar(32) | NO   |     | NULL    |                |
| class_id | bigint(20)  | NO   | MUL | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

MySQL@<lesson5> $
```

```
MySQL@<lesson5> $ create table stu_tb(
    -> id bigint auto_increment,
    -> name varchar(32) not null,
    -> class_id bigint not null,
    -> primary key(id),
    -> foreign key(class_id) references class_tb(id)
    -> );
```

外键一定是在从表中定义的 ！

```
MySQL@<lesson5> $ insert into stu_tb (name, class_id) values ('张飞,104);
    '> ;
    '> ^C
MySQL@<lesson5> $ insert into stu_tb (name, class_id) values ('张飞,104);
    '> ^C
MySQL@<lesson5> $ insert into stu_tb (name, class_id) values ('张飞',104);
Query OK, 1 row affected (0.00 sec)

MySQL@<lesson5> $ insert into stu_tb (name, class_id) values ('关羽',105);
Query OK, 1 row affected (0.00 sec)

MySQL@<lesson5> $ select * from stu_tb;
+----+--------+----------+
| id | name   | class_id |
+----+--------+----------+
|  1 | 张飞   |      104 |
|  2 | 关羽   |      105 |
+----+--------+----------+
2 rows in set (0.00 sec)

MySQL@<lesson5> $ insert into stu_tb (name, class_id) values ('刘备',106);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fa
ils (`lesson5`.`stu_tb`, CONSTRAINT `stu_tb_ibfk_1` FOREIGN KEY (`class_id`) REFE
RENCES `class_tb` (`id`))
MySQL@<lesson5> $
```

当我们想插入一个106期的时候

报错了

因为不符合外键约束的条件

插入失败

同样，如果我们想在班级表里面删掉104的，是删不掉的
我们必须要先删掉学生表了里面104的学生，我们才能在班级表里面删掉104

那么，什么叫做外键呢？

外键不仅仅是产生表和表之间的关联的，还有一个重要属性往往被人忽略

外键在MySQL中还具有特定的约束规则，保证表和表的数据完整性和一致性

存在约束的关联字段，才叫做外键

MySQL基本查询

```
INSERT [INTO] table_name
    [(column [, column] ...)]
    VALUES (value_list) [, (value_list)] ...

value_list: value, [, value] ...
```

```
MySQL@<lesson5> $ create table if not exists students(
    -> id int unsigned primary key auto_increment,
    -> sn int unsigned unique key not null comment "学生的学号",
    -> name varchar(64) not null comment '学生的姓名',
    -> qq varchar(64) unique key
    -> );
Query OK, 0 rows affected (0.03 sec)

MySQL@<lesson5> $ desc students;
+-------+------------------+------+-----+---------+----------------+
| Field | Type             | Null | Key | Default | Extra          |
+-------+------------------+------+-----+---------+----------------+
| id    | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| sn    | int(10) unsigned | NO   | UNI | NULL    |                |
| name  | varchar(64)      | NO   |     | NULL    |                |
| qq    | varchar(64)      | YES  | UNI | NULL    |                |
+-------+------------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)

MySQL@<lesson5> $
```

```
MySQL@<lesson5> $ # 一次插入多条
MySQL@<lesson5> $
MySQL@<lesson5> $ insert into students (sn,name,qq) values(1235,'sunquan','1235@qq.com
'),(1241,'sunshangxiang','2412@qq.com'),(4321,'yuanshao','4321@qq.com');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

MySQL@<lesson5> $ select * from students;
+----+------+--------------+-------------+
| id | sn   | name         | qq          |
+----+------+--------------+-------------+
|  1 |  123 | zhangfei     | 123@qq.com  |
|  2 | 1234 | liubei       | NULL        |
|  3 | 1235 | sunquan      | 1235@qq.com |
|  4 | 1241 | sunshangxiang| 2412@qq.com |
|  5 | 4321 | yuanshao     | 4321@qq.com |
+----+------+--------------+-------------+
5 rows in set (0.00 sec)

MySQL@<lesson5> $
```

用insert一次插入多行的值
用 , 隔开就行

```
MySQL@<lesson5> $ insert into students values (468,'liubiao','783@qq.com');
ERROR 1136 (21S01): Column count doesn't match value count at row 1
MySQL@<lesson5> $ insert into students values (6,468,'liubiao','783@qq.com');
Query OK, 1 row affected (0.01 sec)

MySQL@<lesson5> $ select * from students;
+----+------+--------------+-------------+
| id | sn   | name         | qq          |
+----+------+--------------+-------------+
|  1 |  123 | zhangfei     | 123@qq.com  |
|  2 | 1234 | liubei       | NULL        |
|  3 | 1235 | sunquan      | 1235@qq.com |
|  4 | 1241 | sunshangxiang | 2412@qq.com |
|  5 | 4321 | yuanshao     | 4321@qq.com |
|  6 |  468 | liubiao      | 783@qq.com  |
+----+------+--------------+-------------+
6 rows in set (0.00 sec)

MySQL@<lesson5> $
```

如果想整行插入，可以忽略values左边的列名称去操作

但是如果这样，auto_increment的也要带上一起写,(如果auto_increment列写NULL那还是按照默认的自增规则插入)

不建议这样搞，还是带上values左边的列名称比较好。

```
MySQL@<lesson5> $ insert students values (7,'zhouyu','534@qq.com');
ERROR 1136 (21S01): Column count doesn't match value count at row 1
MySQL@<lesson5> $ insert students values (7,23424,'zhouyu','534@qq.com');
Query OK, 1 row affected (0.00 sec)

MySQL@<lesson5> $ select * from students;
+----+-------+--------------+-------------+
| id | sn    | name         | qq          |
+----+-------+--------------+-------------+
|  1 |   123 | zhangfei     | 123@qq.com  |
|  2 |  1234 | liubei       | NULL        |
|  3 |  1235 | sunquan      | 1235@qq.com |
|  4 |  1241 | sunshangxiang | 2412@qq.com |
|  5 |  4321 | yuanshao     | 4321@qq.com |
|  6 |   468 | liubiao      | 783@qq.com  |
|  7 | 23424 | zhouyu       | 534@qq.com  |
+----+-------+--------------+-------------+
7 rows in set (0.00 sec)

MySQL@<lesson5> $
```

insert后面的into可以省略

6.1.3 插入否则更新

由于 **主键** 或者 **唯一键** 对应的值已经存在而导致插入失败

```
-- 主键冲突
INSERT INTO students (id, sn, name) VALUES (100, 10010, '唐大师');
ERROR 1062 (23000): Duplicate entry '100' for key 'PRIMARY'

-- 唯一键冲突
INSERT INTO students (sn, name) VALUES (20001, '曹阿瞒');
ERROR 1062 (23000): Duplicate entry '20001' for key 'sn'
```

可以选择性的进行同步更新操作 语法:

```
INSERT ... ON DUPLICATE KEY UPDATE
    column = value [, column = value] ...
```

如果插入的时候有冲突
就把东西改掉

```
MySQL@<lesson5> $ select * from students;
+----+-------+---------------+-------------+
| id | sn    | name          | qq          |
+----+-------+---------------+-------------+
|  1 |   123 | zhangfei      | 123@qq.com  |
|  2 |  1234 | liubei        | NULL        |
|  3 |  1235 | sunquan       | 1235@qq.com |
|  4 |  1241 | sunshangxiang | 2412@qq.com |
|  5 |  4321 | yuanshao      | 4321@qq.com |
|  6 |   468 | liubiao       | 783@qq.com  |
|  7 | 23424 | zhouyu        | 534@qq.com  |
+----+-------+---------------+-------------+
7 rows in set (0.00 sec)

MySQL@<lesson5> $ insert into students(id,sn,name) values(7 ,890,'tangsanzang');
ERROR 1062 (23000): Duplicate entry '7' for key 'PRIMARY'
MySQL@<lesson5> $ insert into students(id,sn,name) values(7 ,890,'tangsanzang') on dup
licate key update sn=890,name='tangsanzang';
Query OK, 2 rows affected (0.00 sec)

MySQL@<lesson5> $ select * from students;
+----+------+---------------+-------------+
| id | sn   | name          | qq          |
+----+------+---------------+-------------+
|  1 |  123 | zhangfei      | 123@qq.com  |
|  2 | 1234 | liubei        | NULL        |
|  3 | 1235 | sunquan       | 1235@qq.com |
|  4 | 1241 | sunshangxiang | 2412@qq.com |
|  5 | 4321 | yuanshao      | 4321@qq.com |
|  6 |  468 | liubiao       | 783@qq.com  |
|  7 |  890 | tangsanzang   | 534@qq.com  |
+----+------+---------------+-------------+
7 rows in set (0.00 sec)

MySQL@<lesson5> $
```

如果插入的时候发生冲突了
直接把冲突的改掉

```
MySQL@<lesson5> $ select row_count();
+-------------+
| row_count() |
+-------------+
|          -1 |
+-------------+
1 row in set (0.00 sec)
```

select row_count();
可以查看最近一条语句有几行受影响

```
MySQL@<lesson5> $ ^C
MySQL@<lesson5> $ replace into sutdents (sn, name,qq) values(999, 'sunwukong','8984@qq
.com');
ERROR 1146 (42S02): Table 'lesson5.sutdents' doesn't exist
MySQL@<lesson5> $ replace into students (sn, name,qq) values(999, 'sunwukong','8984@qq
.com');
Query OK, 1 row affected (0.00 sec)

MySQL@<lesson5> $ replace into students (sn, name,qq) values(890, 'sunwukong','8984@qq
.com');
Query OK, 3 rows affected (0.00 sec)

MySQL@<lesson5> $ select * from students;
+----+------+--------------+------------+
| id | sn   | name         | qq         |
+----+------+--------------+------------+
|  1 |  123 | zhangfei     | 123@qq.com |
|  2 | 1234 | liubei       | NULL       |
|  3 | 1235 | sunquan      | 1235@qq.com |
|  4 | 1241 | sunshangxiang | 2412@qq.com |
|  5 | 4321 | yuanshao     | 4321@qq.com |
|  6 |  468 | liubiao      | 783@qq.com |
|  9 |  890 | sunwukong    | 8984@qq.com |
+----+------+--------------+------------+
7 rows in set (0.00 sec)

MySQL@<lesson5> $
```

如果唯一键冲突

```
-- 主键 或者 唯一键 没有冲突，则直接插入；
-- 主键 或者 唯一键 如果冲突，则删除后再插入

REPLACE INTO students (sn, name) VALUES (20001, '曹阿瞒');
Query OK, 2 rows affected (0.00 sec)

-- 1 row affected:      表中没有冲突数据，数据被插入
-- 2 row affected:      表中有冲突数据，删除后重新插入
```

6.2 Retrieve

语法：

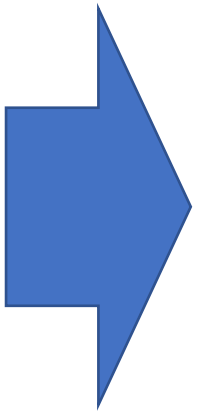
```
SELECT
    [DISTINCT] {* | {column [, column] ...}
    [FROM table_name]
    [WHERE ...]
    [ORDER BY column [ASC | DESC], ...]
    LIMIT ...
```

```
MySQL@<lesson5> $ select * exam result;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corres
ponds to your MySQL server version for the right syntax to use near 'exam_result' at l
ine 1
MySQL@<lesson5> $ select * from exam_result'
    '> ;
    '> ^C
MySQL@<lesson5> $ select * from exam_result;
Empty set (0.00 sec)

MySQL@<lesson5> $ desc exam_result;
+---------+------------------+------+-----+---------+----------------+
| Field   | Type             | Null | Key | Default | Extra          |
+---------+------------------+------+-----+---------+----------------+
| id      | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| name    | varchar(20)      | NO   |     | NULL    |                |
| chinese | float            | YES  |     | 0       |                |
| math    | float            | YES  |     | 0       |                |
| english | float            | YES  |     | 0       |                |
+---------+------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

```
MySQL@<lesson5> $ INSERT INTO exam_result (name, chinese, math, english) VALUES
    -> ('唐三藏', 67, 98, 56), ('孙悟空', 87, 78, 77), ('猪悟能', 88, 98, 90), ('曹孟
德', 82, 84, 67), ('刘玄德', 55, 85, 45), ('孙权', 70, 73, 78), ('宋公明', 75, 65, 30)
;
Query OK, 7 rows affected (0.00 sec)
Records: 7  Duplicates: 0  Warnings: 0

MySQL@<lesson5> $ select * from exam_result;
+----+-----------+---------+------+---------+
| id | name      | chinese | math | english |
+----+-----------+---------+------+---------+
|  1 | 唐三藏    |      67 |   98 |      56 |
|  2 | 孙悟空    |      87 |   78 |      77 |
|  3 | 猪悟能    |      88 |   98 |      90 |
|  4 | 曹孟德    |      82 |   84 |      67 |
|  5 | 刘玄德    |      55 |   85 |      45 |
|  6 | 孙权      |      70 |   73 |      78 |
|  7 | 宋公明    |      75 |   65 |      30 |
+----+-----------+---------+------+---------+
7 rows in set (0.00 sec)
```



准备内容已经弄好，下节课
我们再详细讲怎么查