

- 1、主键
- primary key
 - 唯一标识表中的每行的一列(或者几列，这个又叫复合主键或者组合主键)，称为主键，也叫主关键字
 - 其中任何列都可以做为主键，但是需要满足以下条件：
 - 1) 任意两行都不具有相同的主键值，即**主键值不重复**
 - 2) 每个行都必须具有一个主键值，即**主键值不能为空(NULL)**
 - 主键用来表示一个特定的行，如果没有主键，更新或者删除表中特定的行会很困难，因为没有安全的方法来保证只操作到涉及相关的行
 - 一张表只能有一个主键，但是这个主键可以是一列，也可以是多列的组合；如果是多列的组合，那么这个组合也必须满足上面两个条件

学号	姓名	年龄	性别	手机号
001	张三			
002	张三			
003				

- 2、外键
- foreign key
 - 外键用于关联两张表，外键为表中的一列，同时它又是另外一张表的主键，定义了两个表之间的关系
 - 外键又叫外关键字
 - 主表与从表
 - 主表，以另外一个表的外键做为主键的表
 - 具有此外键的表被称为从表
 - 一张表可以有多个外键，每个外键分别关联不同主表的主键

供应商编码	供应商名称	地址	联系人	邮箱	产品编码	产品名称	产品价格	供应商编码

供应商表

产品编码	产品名称	产品价格	供应商编码

产品表

学生表				课程表		教师表			
学号	姓名	性别	年龄	课程编号	课程名称	学号	姓名	性别	年龄
1	张三	男	20	101	大学英语	1001	令狐	男	20
2	李四	男	18	102	高数	1002	乔帮主	男	18
3	王五	女	16	103	马列主义	1003	仓老师	女	16

授课表		成绩表		
课程编号	老师编号	学号	课程编号	分数
101	1003	1	101	99
102	1001	1	102	88
103	1002	2	101	59

- 外键的作用
 - 1) 外键用于关联两张表
 - 2) 保持数据一致性，完整性，主要目的是控制在外键表中的数据
 - 3) 存储在从表中的外键值只能引用主表中的主键值或者使用空值
 - 最直观的作用就是防止非法输入，就是说录入外键数据时，必须在主表中有对应的数据，否则录入失败
- 外键的功能由两种方法实现：
 - 阻止执行
 - (1) 从表插入新行，其外键值不是主表的主键值便阻止插入；
 - (2) 从表修改外键值，新值不是主表的主键值便阻止修改；
 - (3) 主表删除行，其主键值在从表里存在便阻止删除(要想删除，必须先删除从表的相关行)；
 - (4) 主表修改主键值，旧值在从表里存在便阻止修改(要想修改，必须先删除从表的相关行)。
 - 级联执行
 - (1) 主表删除行，连带从表的相关行一起删除；
 - (2) 主表修改主键值，连带从表相关行的外键值一起修改。
 - 两种实现方法，通过下在创建外键时指定更新和删除的CASCADE、RESTRICT选项来实现，CASCADE为级联执行、RESTRICT为阻止执行，默认为RESTRICT。

- 3、创建表
- 创建表
 - create table (if not exists) 表名(字段1 数据类型 属性, 字段2 数据类型 属性,...,primary key(主键字段)) engine=引擎;
 - 表名见名知意，不能使用关键字、保留字、函数等
 - 主键添加
 - 三种方式
 - 建表的时候两种
 - alter table的方式
 - 练习创建两个表，两种方式
 - primary key(id)
 - id int primary,
 - 查看建表语句
 - desc 表名;

- 4、外键的添加与删除
- 添加
 - 创建表的时候添加
 - constraint fk_xxx_yyy foreign key(xxx_id) references yyy(yyy_id)
 - alter table的方式添加
 - alter table xxx add constraint fk_xxx_yyy foreign key(xxx_id) references yyy(yyy_id);
 - 删除
 - alter table 表名 drop foreign key fk_xxx_yyy;

- 5、字段的添加、修改、删除
- 添加字段
 - alter table 表名 add 字段名 数据类型 属性;
 - alter table student add girl_friend varchar(20);
 - 修改字段
 - alter table 表名 modify 字段名 新数据类型 新属性;
 - alter table student modify girl_friend char(10) not null;
 - alter table 表名 change 旧列名 新列名 新数据类型 属性;
 - alter table student change girl_friend boy_friend char(10);
 - 删除字段
 - alter table 表名 drop 字段
 - alter table student drop boy_friend;
 - 重命名表名...
 - alter table 旧表名 rename to 新表名
 - alter table student rename to students;

- 6、表数据的增删改操作
- 插入
 - insert into 表名(字段1, 字段2, ..., 字段n) values(值1, 值2, ..., 值n);
 - insert into 表名 values(值1, 值2, ..., 值n);
 - insert into 表名 values (值1, 值2, ..., 值n), (值1, 值2, ..., 值n), (值1, 值2, ..., 值n) ;
 - 插入场景
 - 对所有字段插入值
 - 对指定字段插入值
 - 打乱插入字段的顺序来进行插入值
 - 在一条语句中插入多条语句
 - 修改字段(表数据)
 - update 表名 set 字段= 新值 where 子句;
 - 删除表数据
 - delete
 - delete from 表名 [where 子句];
 - truncate
 - truncate 表名;
 - truncate table 表名;
 - delete和truncate的区别(面试)
 - truncate原理是，删表再重建一样的表，所以速度快
 - truncate不能恢复，但是delete能恢复数据

7、sql语句

```
# sql语句

# 切换数据库
use lanhai;

# 创建商品类型表category
# 字段: 分类id, 分类名称, 小分类数量
create table if not exists category(
    c_id int primary key,
    c_name varchar(20),
    c_count int);

# 创建商品表product
# 字段: 商品id, 商品名称, 商品价格, 商品类型
create table if not exists product(
    pid int,
    pname varchar(20),
    p_price decimal(10,2) not null,
    c_id int,
    f_id int,
    primary key(pid),
    constraint fk_product_category foreign key(c_id)
        references category(c_id)
);

# 创建厂商表factory
# 字段: 厂商id, 厂商名称, 创建时间
create table if not exists factory(
    f_id int primary key comment '厂商id',
    f_name varchar(20) not null comment '厂商名称',
    f_create_time year comment '创建年份'
);

# 使用alter方式
# 修改product表的pname类型为char(100) not null
alter table product modify pname char(100) not null;

# 修改product表的pname列, 改名为新列p_name, 类型改为varchar(200), 可以为空, 并且加个注释产品名称
alter table product change pname p_name varchar(200) comment '产品名称';
```