|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1、创建数据表**  语法：create table 表名  (  字段名1 字段类型,  字段名2 字段类型,  ……  )  **CREATE TABLE student\_info(**  **学号 char(4) ,**  **姓名 char(8) ,**  **性别 char(2),**  **出生日期 DATE,**  **家庭住址 VARCHAR(50)**  **);**  **2、查看表结构**   * 用desc语句查看表定义   Desc 表名;   * 用show create table 语句查看表详细定义   Show create table 表名;  **3、修改数据表**   * 修改表名   Alter table 旧表名 rename to 新表名;   * 修改字段   1)修改字段的数据类型modify  Alter table 表名 modify 字段名 数据类型;  2)修改字段名change  Alter table 表名 change 旧字段名 新字段名 旧类型|新类型;  若只改字段名，则使用旧数据类型。   * 添加字段   增加单个字段：  **alter table idoxu add age1 int(10);**  **alter table idoxu add age2 int(11);**  **alter table idoxu add age3 varchar(20);**  增加多个字段：  a**lter table idoxu add (age1 int(10),age2 int(11),age3 varchar(20)**);   * 在表的最后一个位置添加字段   Alter table 表名 add 字段名 类型;   * 在表的第一个位置添加字段   Alter table 表名 add 字段名 类型 first;   * 在表的指定字段后添加字段   Alter table 表名 add 字段名 类型 after 字段名;   * 删除字段   语法：alter table 表名 drop 字段名;  **4、添加表约束（可以通过create语句或者alter语句）**   1. 主键约束   **主键约束可以唯一标识数据表中的每条记录，要求主键必须唯一，且其值不能为空。主键可以是单个字段，也可以是多个字段的联合。**   * 单字段主键   **CREATE TABLE student\_info(**  **学号 char(4) PRIMARY KEY ,**  **姓名 char(8) not NULL,**  **性别 char(2),**  **出生日期 DATE,**  **家庭住址 VARCHAR(50)**  **);**  或者：  **CREATE TABLE student\_info(**  **学号 char(4) ,**  **姓名 char(8) not NULL,**  **性别 char(2),**  **出生日期 DATE,**  **家庭住址 VARCHAR(50),**  **PRIMARY KEY(学号),**  **);**   * 多字段主键（联合主键）   **CREATE TABLE grade(**  **学号 char(4) not NULL ,**  **课程编号 char(4) not NULL ,**  **分数 INT,**  **PRIMARY key(学号,课程编号)**  **);**   1. 外键约束(参照完整性、引用完整性)   用来在两个表的数据之间建立连接，可以指定一个字段或者多个字段为外键。外键的作用是保护数据的引用完整性。一个表的外键可以为空值，若不为空值，则必须是另一个表中主键的某个值。  表1 student\_info表结构   |  |  |  |  | | --- | --- | --- | --- | | 列名 | 数据类型 | 允许NULL值 | 主键 | | **学号** | char(4) | 否 | 是 | | 姓名 | char(8) | 否 | 否 | | 性别 | char(2) | 是 | 否 | | 出生日期 | date | 是 | 否 | | 家庭住址 | varchar(50) | 是 | 否 |   表2 curriculum表结构   |  |  |  |  | | --- | --- | --- | --- | | 列名 | 数据类型 | 允许NULL值 | 主键 | | **课程编号** | char(4) | 否 | 是 | | 课程名称 | varchar(50) | 是 | 否 | | 学分 | int | 是 | 否 |   表3 grade表结构   |  |  |  |  | | --- | --- | --- | --- | | 列名 | 数据类型 | 允许NULL值 | 外键 | | **学号** | char(4) | 否 | 是 | | **课程编号** | char(4) | 否 | 是 | | 分数 | int | 是 | 否 |   语法格式：  **CREATE TABLE grade(**  **学号 char(4) not NULL,**  **课程编号 char(4) not NULL,**  **分数 INT,**  **Constraint fk\_curriculum\_grade Foreign key(课程编号) references curriculum(课程编号);**  **);**   1. 非空约束   语法：字段名 数据类型 not null  **CREATE TABLE grade(**  **学号 char(4) not NULL,**  **课程编号 char(4) not NULL,**  **分数 INT**  **);**   1. 唯一约束   语法：字段名 数据类型 unique  **CREATE TABLE student\_info(**  **学号 char(4) not NULL ,**  **姓名 char(8) not NULL,**  **性别 char(2),**  **邮箱地址 char(20) unique,**  **家庭住址 VARCHAR(50)**  **);**   1. 默认约束   语法：字段名 数据类型 default 默认值  **CREATE TABLE student\_info(**  **学号 char(4) not NULL ,**  **姓名 char(8) not NULL,**  **性别 char(2) default ‘男’,**  **邮箱地址 char(20) unique,**  **家庭住址 VARCHAR(50)**  **);**  **5、删除数据表**  语法：drop table 表名;  **6.复制表结构及数据到新表**  1）CREATE TABLE 新表 SELECT \* FROM 旧表  2）只复制表结构到新表  CREATE TABLE 新表 SELECT \* FROM 旧表 WHERE 1=2  或者  CREATE TABLE 新表 LIKE 旧表  3）复制旧表的数据到新表(假设两个表结构一样)  INSERT INTO 新表 SELECT \* FROM 旧表  4.复制旧表的数据到新表(假设两个表结构不一样)  INSERT INTO 新表(字段1,字段2,.......) SELECT 字段1,字段2,...... FROM 旧表 |