# DDL

# DML-数据操作语言

是指对表中数据的增,删,改

## 插入-insert

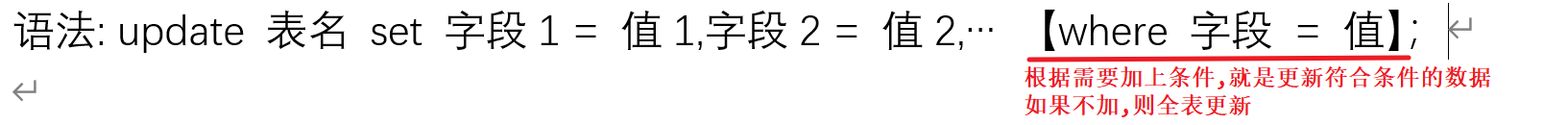
语法: insert into 表名 (列名1,列名2,…) values (值1,值2,…);

注意:

1. 列名顺序与原表无关,但是列要在表中存在
2. 要插入的列,不需要是全表列.可只列出需要插入的字段名
3. 值与列的顺序,个数,类型一致
4. 字符串值,日期值需要使用单引号’’,例如’张三’ ‘2021-01-07’
5. 可以插入空值null(前提是列允许null值)
6. 插入全表时,可以省略列名,值的赋值是按照原表列顺序赋值

|  |
| --- |
| -- 注释  /\*  多行注释  \*/  -- 插入  insert into student (id,username,birthday) values (2,'李四','2021-01-07');  insert into student (id,username,birthday) values ('3','李四','2021-01-07');  insert into student (id,username,birthday) values (4,4,'2021-01-07');  insert into student (id,username,birthday) values (5,'王五',null);  -- 全表  INSERT INTO student (  id,  username,  age,  sex,  birthday,  score  )  VALUES  (  6,  '赵六',  19,  '男',  '2021-01-7',  99.0  );  -- 省略列名,要与原表列顺序一致  insert into student values (7,'赵六7',19,'男','2021-01-07',98.0);  -- 数据长度要与创建时一致  insert into student values (8,'赵六8',20,'男','2021-01-07',999.99);  -- insert into student values (8,'赵六8',20,'男','2021-01-07',9999.99); |

## 更新-update



|  |
| --- |
| -- 更新  -- 将学号为1的学生年龄改为20,成绩改为100  update student set age = 20,score = 100 where id = 1;  -- 将所有女生的年龄改为18  update student set age = 18 where sex = '女'  -- 将学号为7的学生的成绩加2分  update student set score = score + 2 where id = 7;  -- 不加条件,全表更新  update student set birthday = '2021-01-06'; |

## 删除-delete



|  |
| --- |
| -- 删除学生表姓名为张三的数据  delete from student where username = '张三';  delete from student where username = '李四';  -- 如果不加条件,则会删除全表  delete from student; |

ps：

工作中，不能真的删除数据，数据很重要，后期可能会做数据收集，处理，分析等等工作。那么，如何实现“删除”？

设计表时，给表设计多设计一个字段：状态字段status，

一般定义状态码： 1正常 2不正常 或者

1普通用户 2普通会员 3黄金会员 7已注销

在点击按钮删除时，其实是更新字段的状态

# 约束

约束：对插入数据的约束。

一般在创建表时指定约束，然后在插入值时按照约束赋值。

create table 表名(

字段 数据类型(长度) 约束,

字段 数据类型(长度) 约束1 约束2,

字段 数据类型(长度)

);

约束类型:

主键约束

自增约束

唯一约束

数据类型

非空约束

默认值约束

引用完整性约束🡪外键

## 主键约束+自增约束

primary key 主键

auto\_increment 自动增长

|  |
| --- |
| /\*  主键约束:  1 主键列值不能重复,即唯一  2 主键值不能为空  3 主键是索引(index)  一张表中可以设置两个或多个主键,它们是一个整体,叫联合主键  联合主键的所有列值重复,才算重复  ---------------------------  一般主键都会设置为自增.  如果不给主键设置值,则会根据上条数据主键值,自增1  \*/  create table stu1(  id int(11) primary key,  name varchar (10)  );  create table stu2(  id int(11),  age int(11),  primary key(id,age) -- 联合主键  );  create table stu3(  id int primary key auto**\_**increment,  name varchar(10)  );  insert into stu1 (id) values (1);  -- 没有主键的列值,可以重复  insert into student (id) values (1);  insert into student (id) values (1);  -- 主键不能为空  -- insert into stu1 (name) values ('张三');  -- 主键自增,不插入值,会自动增长  insert into stu3 (name) values ('张三');  insert into stu3 (name) values ('李四');  -- 删除一行数据,自增的数据会按照删除之前继续自增  delete from stu3 where id = 2;  insert into stu3 (name) values ('王五'); |

## 唯一约束

|  |
| --- |
| /\*  唯一约束: unique  使该列的值不能重复  \*/  create table stu4(  id int ,  name varchar(10) **unique**  ); |

## 数据类型约束

数据类型约束即,限制该列存储的数据类型,不符合不能存入!

## 非空约束

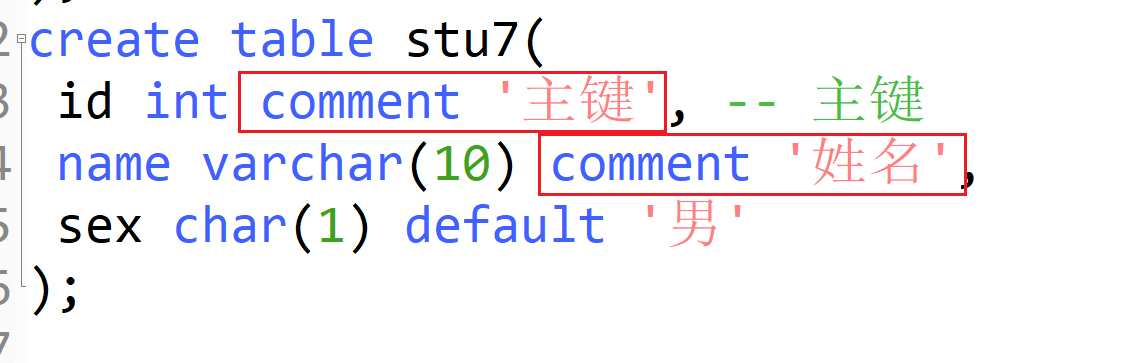
|  |
| --- |
| /\*  非空约束:not null  在插入时该列值不能为空  \*/  create table stu5(  id int,  name varchar(10) not null  ); |

## 默认值约束

DBMS如果创建时不指定约束,那么每个列的值默认为null

|  |
| --- |
| /\*  默认值 default xx  当该列不赋值时,默认赋值为xx.  \*/  create table stu6(  id int,  name varchar(10),  sex char(1) default '男'  ); |

ps: 建表时在字段后 使用comment ‘注释’,可以对该列注释



## 引用完整性约束🡪外键

创建学生表stu,字段sid,sname,age

创建课程表course 字段cid,cname,sid

假设一个学生可以有多个课程,所以在课程表中设计了stu表的主键列sid

create table stu(

sid int primary key auto\_increment comment ‘学生主键’,

sname varchar(10) comment ‘学生姓名’,

age int comment ‘学生年龄’

);

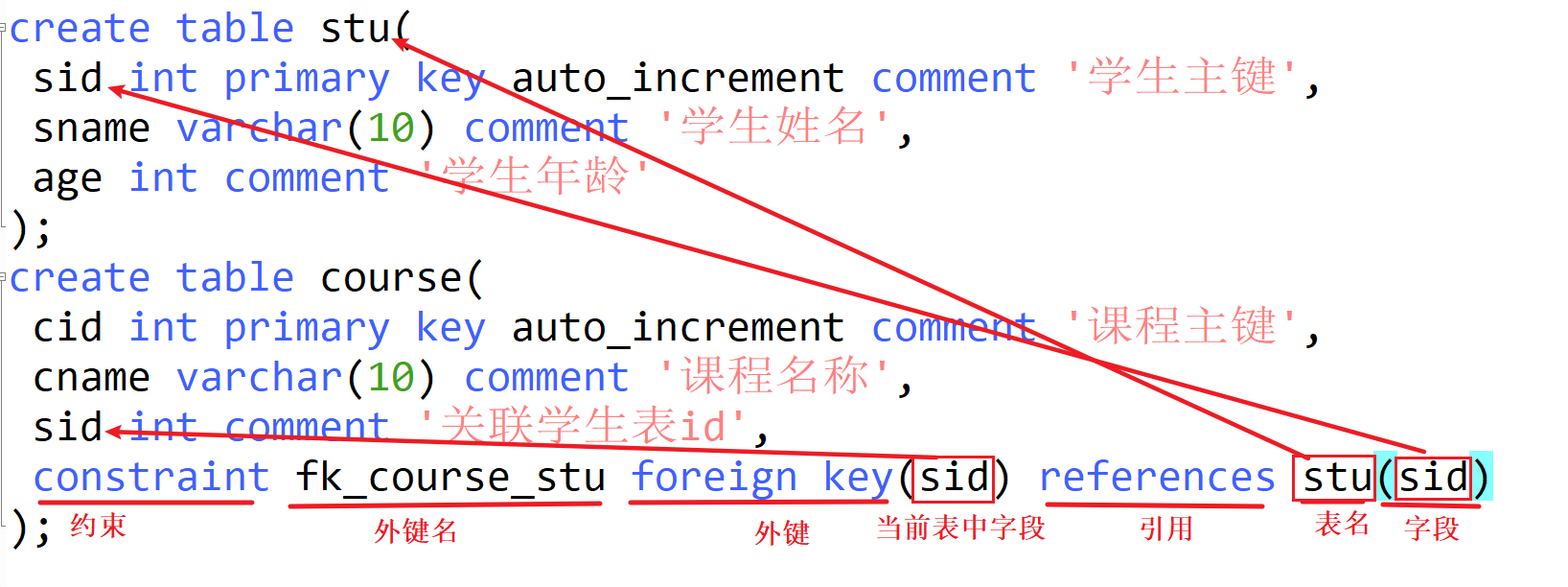
create table course(

cid int primary key auto\_increment comment ‘课程主键’,

cname varchar(10) comment ‘课程名称’,

sid int comment ‘关联学生表id’

)



设计了外键后,课程表中的sid字段就与学生表的sid字段有联系.

父表: stu表 (被引用的表)

子表: course表 (引用表)

插入时:

学生表可以随意插入值

课程表插入值时,sid字段只能插入学生表中有的sid值

删除或更新时:

RESTRICT: 如果想要删除/更新父表的记录,子表中有关联该父表记录的,则不允许删除父表记录

NO ACTION: 同restrict一样,删除时先检查外键.如果有引用,不能删除

CASCADE: 级联,父表删除/更新时,如果有子表有关联,那么子表父表数据一起删除/更新

SET NULL: 置空,父表删除/更新时,如果子表有关联,那么子表该条记录置为null

|  |
| --- |
| /\*  引用完整性约束-外键(foreign key)  创建外键的方式  1 在建表时直接指定  在引用了别的表字段的表中添加外键约束  constraint 外键名 foreign key(当前表字段) references 其他表(字段)  2 建好表后再指定  alter table 表名 add constraint 外键名 foreign key (字段) references 其他表(字段)  \*/  create table stu(  sid int primary key auto\_increment comment '学生主键',  sname varchar(10) comment '学生姓名',  age int comment '学生年龄'  );  create table course(  cid int primary key auto\_increment comment '课程主键',  cname varchar(10) comment '课程名称',  sid int comment '关联学生表id',  constraint fk\_course\_stu foreign key(sid) references stu(sid)  );  alter table course add constraint fk\_course\_stu foreign key (sid) references stu(sid); |

# DQL

DQL是指查询语言,从数据表中通过一系列条件将符合条件查出.

DQL查询语句只是查看数据,不会对原表有任何影响.

查询的结果是一张虚拟表.

查询的关键字: select

语法: select 字段1,字段2,… from 表名【where… group by… having… order by… limit…】

|  |
| --- |
| 语法:  select  selection\_list --要查询的列  from  table\_name -- 要查询的表名  where condition -- 过滤行条件  group by grouping\_clumns -- 对结果按照列进行分组  having condition -- 分组后再过滤  order by sort\_column -- 排序  limit offset,row\_count -- 对结果限制 |

## 基础查询

|  |
| --- |
| -- ========= 基本查询 =========  -- 查询所有列,将所有列名写出即可  select sname,sex,sid,score,age,cid,groupLeaderId from stu;  -- 查询所有列,可以使用\* 代替所有列名  select \* from stu;  -- 查询指定 列  -- 查询学生学号,姓名,年龄  select sid,sname,age from stu;  select sname from stu; |

## 条件查询

|  |
| --- |
| -- ============== 条件查询 =================  /\*  条件查询就是查询时在基础查询的后面使用  where语句,where后使用运算符将符合条件的数据保留  = 相等  != <> 不等  <小于 >大于 <= >=  and 和/与 两边的条件同时成立才行  or 或 两边的条件有一个成立即可  in(set) 范围 在set集合范围内  between 值1 and 值2 在..之间.. (包含临界值,从小到大)  not in 取反  \*/  -- 查询学号为1001的学生信息  select \* from stu where sid = 1001  -- 查询学生成绩大于60的学生id 姓名,成绩  select sid,sname,score from stu where score > 60;  -- 查询学生性别为女，并且年龄小于50的记录  select \* from stu where sex = '女' and age < 50  -- 查询学生学号为1001，或者姓名为李四的记录  select \* from stu where sid = 1001 or sname = '李四'  -- 查询学号为1001，1002，1003的记录  -- select \* from stu where sid = 1001 or sid = 1002 or sid = 1003  select \* from stu where sid in (1001,1002,1003)  -- 查询学号不是1001,1002,1003的记录  select \* from stu where sid not in (1001,1002,1003)  select \* from stu where sid != 1001 and sid != 1002 and sid != 1003  -- 查询学生年龄在20到40之间的学生记录  select \* from stu where age >= 20 and age <= 40;  select \* from stu where age between 20 and 40;  -- 查询性别非男的学生记录  select \* from stu where sex != '男'  -- select \* from stu where sex = '女' |

## 模糊查询

|  |
| --- |
| -- ============ 模糊查询 ===========  /\*  模糊查询需要放在where后使用like关键字  例如 where 字段 like '%关键字'  例如 where 字段 like '\_关键字'  通配符:  % ,可以匹配任何个数任意字符  \_ ,可以匹配一个任意字符  \*/  -- 查询姓名以“张”开头的学生记录  select \* from stu where sname like '张%';  select \* from stu where sname like '张\_';  select \* from stu where sname like '张\_\_';  -- 查询姓名中包含“三”的学生记录  select \* from stu where sname like '%三%' |

## 排序查询

|  |
| --- |
| -- ================= 排序查询 ==============  /\*  排序查询:将查询到的的结果按照一定的顺序排序  order by 字段 [desc|asc] ==> desc 降序,asc升序  例如: order by 字段 desc  例如: order by 字段 asc  \*/  -- 查询所有学生记录，按年龄升序排序  select \* from stu order by age asc  -- 查询所有学生记录，按年龄降序排序  select \* from stu order by age desc  -- 不写排序类型,默认是升序  select \* from stu order by age  -- 查询所有学生记录，按年龄升序排序,如果年龄相同时，按编号降序排序  select \* from stu order by age asc,sid desc  -- 查询成绩大于60的学生id,姓名,成绩,并根据成绩降序  select sid,sname,score from stu where score > 60 order by score desc; |