

select \*from 表名 limit num1, num2; 其中num1表示从第num1行开始查询, num2表示一次查询num2行

一般先分组在排序

数据完整性: 作用: 保证用户输入的数据保存到数据库是正确的

确保数据的完整性 = 在创建表的时候添加约束

完整性的分类:

一, 实体完整性 (行级约束)

实体: 即表中的一行代表一个实体

实体完整性的作用: 标识每一行数据不重复

约束类型:

主键约束 (primary key) 唯一约束 (unique) 自增长列 (auto\_increment)

1, 主键约束:

primary key: 每一个表中有且只有一个主键

特点: 数据唯一, 且不能为null

① create table student2(

classid int,

stuid int,

name varchar(10),

primary key(classid, stuid));

② create table student3(

id int,

name varchar(20));

alter table student3 add primary key(id);

③ create table student1(

id int,

name varchar(10),

primary key(id));

2, 唯一约束 (unique)

数据唯一, 但是可以为null

3, 自增长列 (auto\_increment)

SqlServer数据库: identity

oracle数据库: sequence

## 二，域完整性（列级约束）

域完整性的作用：限制此单元格的数据正确，不对照此列的其他单元格比较

域代表当前单元格

域完整性约束：数据类型 非空约束（not null）默认值约束（default）check约束（mysql不支持）

check（sex = ‘男’ or sex = ‘女’）

1，数据类型（数值类型，日期类型，字符串类型）

2，非空约束（not null）

name varchar(30)not null,

3，默认值约束（default）

## 三，引用完整性（参照完整性）

外键约束：foreign key

有主键才能有外键

## 四，表与表之间的关系

1，一对多（多对一）

2，多对多需要借助第三张表

3，一对一，很少使用，可以建在一张表中

为什么要拆表？

避免数据冗余

## 多表查询：

1，合并结果集：union，union all

①作用：合并结果集就是把俩个select语句的查询结果合并在一起

②合并结果集的两种方式：

a，union：去除重复记录，如：select \*from 表名1 union select \*from 表名2；

（注意：若俩个表中的数据列数不一致，则可以选择查询指定列，使得俩个表查询出来的数据列数一致）

b, union all: 不去除重复记录: 如: select \*from 表名1 union all select \*from 表名2;

批量插入数据: insert into 表名 values (列值1, 列值2....), (列值1, 列值2....), (列值1, 列值2....);

## 2, 连接查询

连接查询会产生笛卡尔积, 可以通过关联关系去除笛卡尔积, 看住主外键当主表中的主键和外表中的外键相等的时候, 进行查询

使用主外键关系作为条件来去除无用信息

如: select \*from 主表, 外表 where 主表. 主键 =外表. 外键;

起别名 表名或列名 (as可以省略) 新表名或新列名

①内连接: **【inner】** join on (默认就是内连接查询, inner可以省略)

select \*from 表1 表1别名 inner join 表2 表2别名 on 查询条件 (主外键关联);

②外链接: outer join on

a, 左外连接: left **【outer】** join on, 参照left左边的表中的数据

如: select \*From 表1 表1别名 left join 表2 表2别名 on 查询条件 (主外键关联);

b, 右外连接: right **【outer】** join on, 参照right右边的表中的数据

如: select \*From 表1 表1别名 right join 表2 表2别名 on 查询条件 (主外键关联);

查询对原表没有关系

主外键关系与原表的数据没有关系

n张表连查需要n-1个条件

保证数据完整性 = 创建表时给表添加约束

c, 全外连接: (mysql不支持) full join

③自然连接: natural join

相当于内连接, 两张连接的表中名称和数据类型完全一致的列作为条件

如: select \*from 表名1 natural join 表2;

### 3, 子查询

#### 注意:

1, 去除重复记录, 需要使用关键字distinct: `select distinct 列名 from 表名`

2, `cls`: 在dos下进行清屏

3, `ifnull`: `ifnull (exp1, exp2)`, 如果exp1不是null, 函数返回exp1, 否则返回exp2, 返回的结果是一个数字或者字符串值

4, `order by`中默认是asc, 即升序, desc表示降序

5, 因为count (列名) 函数中给出的是列名, 所有只统计指定列名不为null的行数

6, 聚合函数和括号之间不能有空格

7, `having`是在分组后对数据进行过滤

`where`是在分组前对数据进行过滤

8, `having`后面可以使用分组函数 (统计函数)

`where`后面不可以使用分组函数

9, `where`是对分组前记录的条件, 如果某行记录满足where子句的条件, 那么这行记录不会参与分

组, 而`having`是对分组后数据的约束

10, 查询语句书写顺序: `select - from - where - group by - having - order by - limit`

查询语句执行顺序: `from - where - group by - having - select - order by - limit`

11, 在mysql中, 需要判断一个属性列为空的话, 需要使用`is null`或者`is not null`进行判断

12, 在mysql中, 要尽量避免使用中文进行命名