# 一 数据约束

数据类型决定字段需要存储什么样的数据，而约束告诉我们这些数据具体需要满足的规则

## 1 默认值 (DEFAULT)

设置默认值后，插入数据时，如果没有给该字段插入数据，该字段的值使用设置的默认值，设置非NULL的默认值后，插入数据时，该值也可以设为NULL

CREATE TABLE person (

did int ,

dname varchar(10) DEFAULT ‘hello’

);

## 2 非空 （是否为NULL）

Person表的name字段设置成不为NULL，插入数据时，如果不给name字段赋值，报错

mysql> insert into person(id) values(4);

1364 - Field 'name' doesn't have a default value

修改name字段不能为NULL，插入值时，如果name为NULL，报错

mysql> insert into person(id,name) values(4,NULL);

1048 - Column 'name' cannot be null

## 3 唯一 (Unique)

CREATE TABLE person(

id int UNIQUE,

name VARCHAR(10)

)

字段设置为unique，会自动建立唯一性索引

如果插入两条id值相同数据：

mysql> insert into person(id,name) values(3,'wangwu');

mysql> insert into person(id,name) values(3,'wangwu');

1062 - Duplicate entry '3' for key 'index\_id'

注意：唯一性约束的字段，可以设置为NULL，而且多条记录中，该字段都可以为NULL，且不会报错

## 4 自增(AUTO INCREMENT)

设置为自增字段后，字段自动设置为not null

插入数据时，不用给自增的字段赋值，mysql中记录了表的自增字段的最大值，如果删除数据，再次插入数据时，自增字段的值根据mysql记录的最大值进行处理

INSERT INTO person(name,age) values('zhaosi',12)

当然，也可以在插入语句中给自增的字段直接赋值

INSERT INTO person(id,name,age) values(20,'zhaosi',12)

注：自增属性不单独设置，需要配合主键或唯一索引使用

## 5 主键 (PRIMARY KEY)

主键用来标识一个唯一记录，大部分情况下，每个表都会设置主键；

主键可以有一个，也可以有多个，如果有多个主键，称为联合主键；

主键相当于 非空+唯一

CREATE TABLE person (

id INT PRIMARY KEY,

name VARCHAR(20)

)

INSERT INTO person(id,name,age) values(22,'zhaosi',12)

[Err] 1062 - Duplicate entry '22' for key 'PRIMARY'

很多时候，可以在表中设置某个字段为自增约束(AUTO\_INCREMENT)，并且用该字段作为主键

## 6 外键 (FOREIGN KEY)

思考：有部门表和员工表，员工表中有个字段，用来表示部门表的id。插入数据时，在员工表中的部门id可以随意插入数据，并且部门表中不存在的数据也可以插入。这样做是否合适？如何解决？

主表：通过其他表的外键进行关联，约束其他表的表

副表：有外键的表，被约束的表

(1)建立外键

ALTER TABLE `employee` ADD FOREIGN KEY (`deptId`) REFERENCES `department` (`did`);

如果主表和副表间的某些数据有关联，

添加数据时， 先添加主表，再添加副表；

修改数据时，只能修改副表中关联的数据为已有数据，不能修改主表；

删除数据时，只能先删除副表中数据，不能先删除主表数据。如果先删除主表中数据，因为有外键的约束，不能删除，这样就可以避免数据的冗余

1. 级联操作（理解）

如果主表和副表某些数据有关联，主表中数据删除更新时，副表中相关数据也删除更新，这就需要进行级联操作。

RESTRICT 如果有关联的数据，不允许直接删除/更新主表中数据，提示错误

NO ACTION 类似RESTRICT

CASCADE 直接删除/更新主表中数据时，副表中关联的数据一起删除/更新

SET NULL 直接删除/更新主表数据，副表中关联的数据设置为null

# 二 数据库设计

## 1 范式

第一范式：数据表中的字段要保持原子性，即每个字段都是不可再分的最小数据单元

例如：

编号 名字 联系方式

1 zhangsan 123456/1312345678/aa@aa.com

第二范式：满足第一范式的基础上，表中其他字段要和主键字段有关联

例如：

编号 名字 是否下雨

1 张三 否

第三范式：满足第二范式的基础上，表中其他字段都和主键有直接关联

例如：

学生id 学生姓名 选的课程名 课程教材

X001 张三 java java基础入门

如何改？

提供课程表，然后在学生表，提供课程的外键关联

## 2 表之间常见关系

1）一对一

一个人对应一个身份证

2）一对多/多对一

一个部门下有多个员工，多个员工属于一个部门（一个员工只会属于一个部门）

3）多对多

一个学生可以选多门课，一个课程可以多个学生选

学生表、课程表、选课表 体现课程和学生的关系

注意：在数据表设计规范化的同时，要综合考虑数据库的性能

## 3多表查询

查询员工所属的部门名称：

两步走

SELECT deptId from employee WHERE ename='张三'

SELECT dname from department WHERE did=1

(1)内连接

在表中找出符合条件的共有记录

select department.dname, employee.ename

from department,employee

where department.did=employee.deptId

select department.dname

from department,employee

where department.did=employee.deptId and employee.ename='张三'

select \*

from department,employee

where department.did=employee.deptId

//表设置别名，as可以省略

select d.dname,e.ename

from department d,employee e

where d.did=e.deptId

select d.dname,e.ename

from department as d,employee as e

where d.did=e.deptId

**使用inner join进行内连接查询**

select d.dname, e.ename

from department d inner join employee e

on d.did = e.deptId

select d.dname, e.ename

from department d inner join employee e

on d.did = e.deptId and e.ename='张三'

select d.dname, e.ename

from department d inner join employee e

on d.did = e.deptId

where e.ename='张三'

(2)外连接

a.左外连接left join

左表中的数据都会查询出来，如果右表中没匹配的数据，显示null

select d.\*,e.ename

from department d left join employee e

on d.did=e.deptId

select d.\*,e.ename

from employee e left join department d

on d.did=e.deptId

b.右外连接 right join

右表中的数据都会查询出来，如果左表中没有匹配的数据，显示null

select d.\*,e.ename

from employee e right join department d

on d.did=e.deptId

(3)交叉连接

查询语句中，如果不带WHERE条件，它将会返回被连接的两个表的笛卡尔积，即返回结果的行数等于两个表行数的乘积。

通过交叉连接，可以得到所有可能的组合，查询到的记录数 = 表1的行数 X 表2的行数

select \* from department,employee;

(4)自连接

进行连接的表是同一张表

例如：表示菜单项的表

文件

保存

新建

编辑

粘贴

复制

select m1.name, m2.name

from menu m1,menu m2

WHERE m1.id=m2.pid and m1.name='文件'

## 4 嵌套查询 理解

将一个查询块嵌套在另一个查询块的 WHERE 子句或 HAVING 短语的条件中的查询称为嵌套查询，嵌套的查询属于子查询

谓词

1）in表示是否在某个范围内，作用上等效于多个条件使用or连接

select \* from student WHERE tid in (select tid from teacher where age>24)

select \* from student WHERE tid not in (select tid from teacher where age>24)

2）any是表示子查询结果中任意一个

本例中，> any表示只要大于子查询结果中的任一个，表达式就成立，比最小的大

select \* from student WHERE tid > Any (select tid from teacher where age>24)

3）all 表示子查询结果中的所有

select \* from student WHERE tid <> All (select tid from teacher where age>24)

将运算符与any、all结合使用时，关系如下：

IMG_256

5）exists是表示子查询是否返回结果（true或者false），而不管返回的具体内容

使用exists时，对外表的数据逐条查询，遍历每条数据时，都会查看 exists后跟着的语句是否能够返回记录，只要能返回，条件就为真，当前查询的外表的这条记录就会返回，反之，当前查询的外表记录被丢弃

select \* from student WHERE EXISTS (select tid from teacher where age=22)

# 三 视图 理解

通过视图，将某个表或多个表的字段提取出来，形成一个新的表

如果不想让用户看到所有数据（字段，记录），只想让用户看到某些的数据时，此时可以使用视图;当需要减化SQL查询语句的编写时，可以使用视图，但不提高查询效率

视图是一种虚表

视图建立在已有表的基础上, 视图赖以建立的这些表称为基表

向视图提供数据内容的语句为 SELECT 语句,可以将视图理解为存储起来的 SELECT 语句

视图没有存储真正的数据，真正的数据还是存储在基表中

程序员虽然操作的是视图，但最终视图还会转成操作基表，一个基表可以有0个或多个视图

Sql语句：

CREATE

VIEW `NewView`AS

SELECT

…

FROM

…

# 四 索引 掌握

## 1 什么是索引

索引是一种快速查询表中内容的机制，使用索引可以提高查询效率，相当于字典的目录

可以将查询过程中经常使用的条件设为索引

## 2 mysql中索引的类型

1）普通索引

这是最基本的索引类型，而且它没有唯一性之类的限制。普通索引可以通过以下几种方式创建：

创建索引，例如CREATE INDEX <索引的名字> ON tablename (列的列表);

修改表，例如ALTER TABLE tablename ADD INDEX [索引的名字] (列的列表);

创建表的时候指定索引，例如CREATE TABLE tablename ( [...], INDEX [索引的名字] (列的列表) );

2）唯一性索引

这种索引和前面的“普通索引”基本相同，但有一个区别：索引列的所有值都只能出现一次，即必须唯一。唯一性索引可以用以下几种方式创建：

创建索引，例如CREATE UNIQUE INDEX <索引的名字> ON tablename (列的列表);

修改表，例如ALTER TABLE tablename ADD UNIQUE [索引的名字] (列的列表);

创建表的时候指定索引，例如CREATE TABLE tablename ( [...], UNIQUE [索引的名字] (列的列表) );

3）主键索引

主键是一种唯一性索引，也可称为聚集索引，但它必须指定为“PRIMARY KEY”。主键一般在创建表的时候指定，例如“CREATE TABLE tablename ( [...], PRIMARY KEY (列的列表) ); ”。但是，我们也可以通过修改表的方式加入主键，例如“ALTER TABLE tablename ADD PRIMARY KEY (列的列表); ”

4）全文索引

全文索引(全文检索)能够利用分词技术等多种算法，按照一定的算法规则智能地筛选出我们想要的搜索结果。

MySQL从3.23.23版开始支持全文索引和全文检索。在MySQL中，全文索引的索引类型为FULLTEXT。全文索引可以在VARCHAR或者TEXT类型的列上创建。它可以通过CREATE TABLE命令创建，也可以通过ALTER TABLE或CREATE INDEX命令创建。对于大规模的数据集，通过ALTER TABLE（或者CREATE INDEX）命令创建全文索引要比把记录插入带有全文索引的空表更快。

需要注意的是，MySQL自带的全文索引只能用于数据库引擎为MyISAM的数据表，如果是其他数据引擎，则全文索引不会生效。

## 3 什么情况下创建索引

1）表经常进行 SELECT 操作

2）表很大(记录超多)，记录内容分布范围很广

3）列名经常在 WHERE 子句或连接条件中出现

注意：符合上述某一条要求，都可创建索引，创建索引是一个优化问题，同样也是一个策略问题

## 4索引的缺点

首先，索引要占用磁盘空间。通常情况下，这个问题不是很突出。但是，如果你创建每一种可能列组合的索引，索引文件体积的增长速度将远远超过数据文件。如果你有一个很大的表，索引文件的大小可能达到操作系统允许的最大文件限制。

第二，对于需要写入数据的操作，比如DELETE、UPDATE以及INSERT操作，索引会降低它们的速度。这是因为MySQL不仅要把改动数据写入数据文件，而且它还要把这些改动写入索引文件。

-- 100万条数据，查询时有无索引，时间差距明显--

select \* from user where username='zz345';