# 关键词

## Left join && Right join

SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo

FROM Persons

LEFT JOIN Orders

ON Persons.Id\_P=Orders.Id\_P

ORDER BY Persons.LastName

左表是Persons，右表是Orders。Left join 会从左表返回所有的行，即使右表中没有匹配的行。

SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo

FROM Persons

RIGHT JOIN Orders

ON Persons.Id\_P=Orders.Id\_P

ORDER BY Persons.LastName

左表是Persons，右表是Orders。Right join 会从右表返回所有的行，即使左表中没有匹配的行。

## Having

一般和group by联合使用，用来过滤group by语句返回的记录值。

# 数据库的DDL操作

## 操作数据库

create database if not exists dbname;/创建数据库

drop databasae if exists dbname;//销毁数据库

Use 数据库名

## 操作表

创建表

CREATE TABLE IF NOT EXISTS `runoob\_tbl`(

`runoob\_id` INT UNSIGNED AUTO\_INCREMENT,

`runoob\_title` VARCHAR(100) NOT NULL,

`runoob\_author` VARCHAR(40) NOT NULL,

`submission\_date` DATE,

PRIMARY KEY ( `runoob\_id` ))ENGINE=InnoDB DEFAULT CHARSET=utf8;

添加字段

ALTER TABLE 表名 ADD 字段名称 字段类型（字段长短-选填） NOT NULL（是否不可为空） DEFAULT 0（默认值） COMMENT '已经跑批给SAS的维度数目之和'（备注）;

修改字段

-- 修改字段属性

-- ALTER TABLE tb\_name MODIFY 字段名称 字段类型 [完整性约束条件]

-- 将email字段 VARCHAR(50)修改成VARCHAR(200)

-- 注意，修改时如果不带完整性约束条件，原有的约束条件将丢失，如果想保留修改时就得带上完整性约束条件

ALTER TABLE user10 MODIFY email VARCHAR(200) NOT NULL DEFAULT 'a@a.com';

-- 将card移到test后面

ALTER TABLE user10 MODIFY card CHAR(10) AFTER test;

-- 将test放到第一个，保留原完整性约束条件

ALTER TABLE user10 MODIFY test CHAR(32) NOT NULL DEFAULT '123' FIRST;

-- 将test字段改为test1

-- ALTER TABLE 表名 CHANGE 原字段名 新字段名 字段类型 约束条件ALTER TABLE user10 CHANGE test test1 CHAR(**32**) NOT NULL DEFAULT '123';

添加主键

ALTER TABLE test12 ADD PRIMARY KEY(id);

-- 添加复合主键

ALTER TABLE test13 ADD PRIMARY KEY(id,card);

五、DDL：数据定义语言

作用：用于描述数据库中要存储的现实世界实体的语言。即创建数据库和表的结构。

常用关键字：CREATE ALTER DROP TRUNCATE

create alter drop truncate

-----------------------------库操作

显示所有的数据库

SHOW DATABASES;

创建一个名称为mydb1的数据库。

CREATE DATABASE mydb1;

查看数据库的创建细节，可以看到使用的字符集。

SHOW CREATE DATABASE mydb1;

创建一个使用gbk字符集的mydb2数据库。

CREATE DATABASE mydb2 CHARACTER SET gbk;

创建一个使用gbk字符集，并带校对规则的mydb3数据库。

CREATE DATABASE mydb3 CHARACTER SET gbk COLLATE gbk\_chinese\_ci;

删除前面创建的mydb3数据库

DROP DATABASE mydb3;

查看服务器中的数据库，并把mydb2的字符集修改为utf8;

ALTER DATABASE mydb2 CHARACTER SET utf8;

-----------------------------表结构操作

显示当前的数据库

SELECT DATABASE();

选择数据库

USE mydb1;

创建一个员工表

CREATE TABLE employee(

id int,

name varchar(100),

gender varchar(10),

birthday date,

entry\_date date,

job varchar(100),

salary float(8,2),

resume text

);

查看库中的所有表格

SHOW TABLES;

查看表的结构

DESC employee;

查看表的创建细节

SHOW CREATE TABLE employee;

在上面员工表的基本上增加一个image列。

ALTER TABLE employee ADD image blob;

修改job列，使其长度为60。

ALTER TABLE employee MODIFY job varchar(60);

删除image列。

ALTER TABLE employee DROP image;

表名改为user。

RENAME TABLE employee TO user;

修改表的字符集为utf8

ALTER TABLE user CHARACTER SET utf8;

列名name修改为username

ALTER TABLE user CHANGE name username varchar(100);

# 三范式

1. 属性不可分割，不能拆分为更小的单位了。
2. 要有主键，比如一张表中存放了学号、姓名和课程号，课程名称，那么一个学号对应多个课程号，要把这两个拆开，学号做主键的一张表，课程号做主键一张表。
3. 消除冗余
4. BCNF消除每一属性对候选键的传递依赖
5. 4NF

# 创建索引

## 主键索引

alter table table\_name add primary key ( `column` )

主键索引，能够唯一标识一条记录。

## 唯一索引

alter table table\_name add unique (`column`)

唯一索引是为了避免某一列中的值重复，主键索引只能有一个，但是唯一索引可以有多个。

## 全文索引

alter table table\_name add fulltext (`column` )

快速定位特定数据，只能用于char，varchar，text数据列类型

## 普通索引

alter table table\_name add index index\_name ( `column` )

快速定位特定数据，

## 聚簇索引

alter table `table\_name` add index index\_name ( `column1`, `column2`, `column3` )

一列中的值重

# 哪些字段适合创建索引

1. 经常搜索的列上创建索引，加快搜索的速度
2. 主键上，
3. 连接列上，主要是一些外键，加快连接速度。
4. 经常需要根据范围进行搜索的列上，因为索引已经排好序，指定的范围是连续的
5. 在经常需要排序的列上，该索引已经排好序
6. 在where子句上，加快判断速度

# 创建索引选择

索引可以增加检索速度，但是修改删除速度就会降低。检索速度和修改速度是相互矛盾的。

# 时间操作

关于时间的函数：<https://www.cnblogs.com/weiqt/articles/1826847.html>

--2019-01-08

SELECT CONVERT (VARCHAR(11), getdate(), 120)

--20190108

select CONVERT (nvarchar(12),GETDATE(),112)

--2019

Select Datename(year,GetDate())

--01

Select Datename(MONTH,GetDate())

--8

Select Datename(DAY,GetDate())

--2019-01

Select Datename(year,GetDate())+'-'+Datename(month,GetDate())

--2019-01-8

Select Datename(year,GetDate())+'-'+Datename(month,GetDate())+'-'+Datename(day,GetDate())

--2019/01/8

select DATENAME(YEAR,GETDATE())+'/'+DATENAME(MONTH,GETDATE())+'/'+DATENAME(DAY,GETDATE())

--2019

select DATENAME(YEAR,GETDATE())

select DATENAME(YY,GETDATE())

select DATENAME(MM,GETDATE()) --月份

select DATENAME(DD,GETDATE()) --日期

select dateName(hh,getdate()) --获取小时

select DATENAME(MI,GETDATE()) --获取分钟

select DATENAME(SECOND,GETDATE()) --获取秒

select DATENAME(WEEK,GETDATE()) --获取当前星期(周)是这一年中的第几个星期(周)--这里查询的是第二周

select DATENAME(WEEKDAY,GETDATE()) --星期几