Mysql\_learning —day01

===============================

SELECT name,age from student3;

SELECT address from student3;

select DISTINCT address from student3;

drop table student3;

CREATE TABLE student3 ( id int, -- 编号

name varchar(20), -- 姓名

age int, -- 年龄

sex varchar(5), -- 性别

address varchar(100), -- 地址

math int, -- 数学

english int -- 英语

);

INSERT INTO student3(id,NAME,age,sex,address,math,english) VALUES (1,'马云',55,'男',' 杭州',66,78),(2,'马化腾',45,'女','深圳',98,87),(3,'马景涛',55,'男','香港',56,77),(4,'柳岩 ',20,'女','湖南',76,65),(5,'柳青',20,'男','湖南',86,NULL),(6,'刘德华',57,'男','香港',99,99),(7,'马德',22,'女','香港',99,99),(8,'德玛西亚',18,'男','南京',56,65);

SELECT \* FROM student3;

SELECT name, math 数学, english 英语 , math + ifnull(english,0) as 总分 from student3;

select \* FROM student3 where age > 20;

select \* from student3 where age = 20;

select \* from student3 where age != 20;

select \* from student3 where age <> 20;

SELECT \* from student3 where age >= 20 and age <= 30;

SELECT \* from student3 where age >= 20 && age <= 30;

SELECT \* from student3 WHERE age BETWEEN 20 and 30;

SELECT \* from student3 WHERE age = 22 or age = 18 or age = 25;

SELECT \* from student3 WHERE age in (22,18,25);

SELECT \* FROM student3 WHERE english is null;

SELECT \* FROM student3 WHERE english is not NULL;

SELECT \* FROM student3 WHERE NAME LIKE '马%';

SELECT \* from student3 WHERE NAME like '\_化%';

SELECT \* from student3 WHERE NAME like '\_\_\_';

SELECT \* FROM student3 WHERE NAME like '%德%';

SELECT \* From student3 ORDER BY math ;

SELECT \* from student3 ORDER BY math ASC;

SELECT \* from student3 ORDER BY math desc;

SELECT \* from student3 ORDER BY math DESC, english DESC;

SELECT COUNT(english) from student3;

SELECT COUNT(IFNULL(english,0)) from student3;

SELECT COUNT(\*) from student3;

SELECT MAX(math) from student3;

SELECT MIN(english) from student3;

SELECT SUM(math) FROM student3;

SELECT AVG(math) from student3;

SELECT sex, avg(math) from student3 GROUP BY sex;

select sex, AVG(math), COUNT(id) from student3 GROUP BY sex;

SELECT sex, AVG(math), COUNT(id) from student3 where math > 70 GROUP BY sex;

SELECT sex, AVG(math), COUNT(id) from student3 WHERE math > 70 GROUP BY sex HAVING COUNT(id) > 2;

SELECT \* from student3 LIMIT 0,3;

SELECT \* from student3 LIMIT 3,3;

SELECT \* from student3 LIMIT 6,3;

CREATE TABLE stu(

id INT,

NAME VARCHAR(20) NOT NULL );

INSERT into stu( id, NAME) VALUES(1, 'zwq');

SELECT \* FROM stu;

INSERT INTO stu(id , NAME) VALUES(2,NULL);--erro, because the name column is not null.

ALTER TABLE stu MODIFY NAME VARCHAR(20);

INSERT INTO stu(id , NAME) VALUES(3,NULL);--right, because there exists no 'not null' in name column.COLUMN

ALTER TABLE stu MODIFY NAME VARCHAR(20) NOT NULL;

DELETE FROM stu WHERE id IN (2,3);

DESC stu;

CREATE TABLE student1( id INT, phone\_number INT UNIQUE);

SELECT \* FROM student1;

INSERT INTO student1(id ,phone\_number) values(1,111);

INSERT INTO student1(id ,phone\_number) values(1,111);--error

ALTER TABLE student1 MODIFY phone\_number INT;--not work

INSERT INTO student1(id ,phone\_number) values(1,111);

ALTER TABLE student1 DROP INDEX phone\_number;

CREATE TABLE student2(

id int primary key,

NAME VARCHAR(20));

SELECT \* FROM student2;

INSERT into student2( id , name) VALUES( 1, 'zwq');

ALTER TABLE student2 DROP PRIMARY KEY;

ALTER TABLE student2 MODIFY id int PRIMARY KEY auto\_increment;

INSERT into student2( id , name) VALUES( NULL, 'zwq');

ALTER TABLE student2 MODIFY id INT;

create table department(

id int primary key auto\_increment,

dep\_name varchar(20),

dep\_location varchar(20)

);

insert into department values(null, '研发部','广州'),(null, '销售部', '深圳');

select \* from department;

create table employee(

id int primary key auto\_increment, name varchar(20),

age int,

dep\_id int, -- 外键对应主表的主键

CONSTRAINT emp\_dep FOREIGN KEY (dep\_id) REFERENCES department(id) -- 创建外键约束

);

INSERT INTO employee (NAME, age, dep\_id) VALUES ('张三', 20, 1);

INSERT INTO employee (NAME, age, dep\_id) VALUES ('李四', 21, 1);

INSERT INTO employee (NAME, age, dep\_id) VALUES ('王五', 20, 1);

INSERT INTO employee (NAME, age, dep\_id) VALUES ('老王', 20, 2);

INSERT INTO employee (NAME, age, dep\_id) VALUES ('大王', 22, 2);

INSERT INTO employee (NAME, age, dep\_id) VALUES ('小王', 18, 2);

SELECT \* FROM employee;

ALTER TABLE employee DROP FOREIGN KEY emp\_dep;

ALTER TABLE employee ADD CONSTRAINT emp\_dep FOREIGN KEY ( dep\_id ) REFERENCES department ( id);

ALTER TABLE employee ADD CONSTRAINT emp\_dep FOREIGN KEY (dep\_id ) REFERENCES department (id ) ON UPDATE CASCADE ON DELETE CASCADE;

create table tab\_category (

cid int primary key auto\_increment,

cname varchar(100) not null unique

);

insert into tab\_category (cname) values ('周边游'), ('出境游'), ('国内游'), ('港澳游');

select \* from tab\_category;

create table tab\_route(

rid int primary key auto\_increment,

rname varchar(100) not null unique,

price double,

rdate date,

cid int,

foreign key (cid) references tab\_category(cid)

);

INSERT INTO tab\_route VALUES

(NULL, '【厦门+鼓浪屿+南普陀寺+曾厝垵 高铁 3 天 惠贵团】尝味友鸭面线 住 1 晚鼓浪屿', 1499, '2018-01-27', 1),

(NULL, '【浪漫桂林 阳朔西街高铁 3 天纯玩 高级团】城徽象鼻山 兴坪漓江 西山公园', 699, '2018-02-22', 3),

(NULL, '【爆款¥1699 秒杀】泰国 曼谷 芭堤雅 金沙岛 杜拉拉水上市场 双飞六天【含送签费 泰风情 广州 往返 特价团】', 1699, '2018-01-27', 2),

(NULL, '【经典•狮航 ¥2399 秒杀】巴厘岛双飞五天 抵玩【广州往返 特价团】', 2399, '2017-12-23', 2),

(NULL, '香港迪士尼乐园自由行2天【永东跨境巴士广东至迪士尼去程交通+迪士尼一日门票+香港如心海景酒店 暨会议中心标准房1晚住宿】', 799, '2018-04-10', 4);

select \* from tab\_route;

create table tab\_user (

uid int primary key auto\_increment,

username varchar(100) unique not null,

password varchar(30) not null,

name varchar(100),

birthday date,

sex char(1) default '男',

telephone varchar(11),

email varchar(100)

);

-- 添加用户数据

INSERT INTO tab\_user VALUES

(NULL, 'cz110', 123456, '老王', '1977-07-07', '男', '13888888888', '66666@qq.com'), (NULL, 'cz119', 654321, '小王', '1999-09-09', '男', '13999999999', '99999@qq.com');

select \* from tab\_user;

/\*

创建收藏表 tab\_favorite rid 旅游线路 id，外键

date 收藏时间 uid 用户 id，外键

rid 和 uid 不能重复，设置复合主键，同一个用户不能收藏同一个线路两次

\*/

create table tab\_favorite (

rid int,

date datetime,

uid int,

-- 创建复合主键

primary key(rid,uid),

foreign key (rid) references tab\_route(rid),

foreign key(uid) references tab\_user(uid)

);