复习

更改记录

UPDATE emp SET salary=5000,sex=0 WHERE eid=5;

删除记录

DELETE FROM emp WHERE eid=6;

中文乱码

Latin-1

Unicode，utf8

sql脚本文件，客户端设置连接使用的编码，服务器存储数据使用的编码

列类型

数值型

TINYINT(-128~127) SMALLINT(-32768~32767) INT(-2147483648~2147483647) BIGINT FLOAT DOUBLE DECIMAL(6,2) 9998.99 BOOL

字符串型

VARCHAR(M) 65535

CHAR(M) 255

TEXT

日期时间型

DATE TIME DATETIME

列约束

主键约束 PRIMARY KEY

今日目标

列约束

简单查询

1.列约束

(1)主键约束——PRIMARY KEY

(2)唯一约束——UNIQUE

声明了唯一约束的列上不能插入重复的值，但可以插入NULL，而且可以插入多个NULL

(3)非空约束——NOT NULL

声明非空约束的列上不能插入NULL值，如果插入的是NULL也不会报错，会显示空内容。

(4)默认值约束——DEFAULT

可以使用DEFAULT关键字为列声明默认值；有以下两种用法：

INSERT INTO xz\_laptop VALUES(1,'小米air', DEFAULT...);

INSERT INTO xz\_laptop(lid,title) VALUES(2,'苹果pro');

(5)检查约束

检查约束可以对插入的数据范围进行验证

CREATE TABLE student(

age TINYINT CHECK(age>18 AND age<60)

);

MySQL不支持，会降低插入和更新等操作效率。

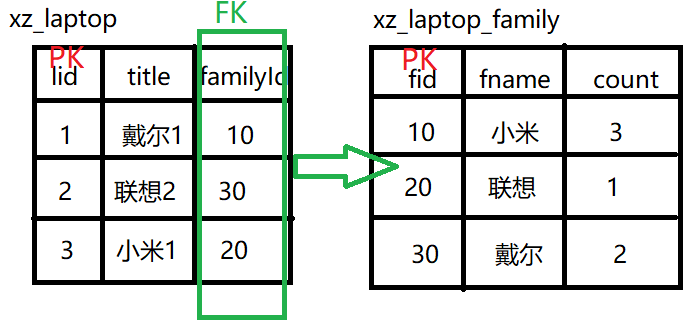
(6)外键约束——FOREIGN KEY

声明了外键约束的列，取值必须在另一个表的主键上出现过

取值可以是NULL;

**注意：**外键的列类型要和另外一个表主键的列类型一致。

FOREIGN KEY(familyId) REFERENCES xz\_laptop\_family(fid)



|  |
| --- |
| 所有的列约束中，主键约束推荐使用——可以提高查询的效率；其它的列约束是否使用根据项目而定，可以不用——会影响数据的插入和更新效率。 |

2.MySQL中的自增列

AUTO\_INCREMENT: 自动增加，加入一个列声明了自增列，无需手动赋值，直接指定为NULL，会自动获取当前的最大值，新纪录会执行+1然后插入。

注意：

允许手动赋值

只用于整数型的主键列上

练习：使用列约束；

创建脚本文件02\_tedu.sql；创建数据库tedu

创建部门表 dept，包含did，dname部门名称；

10 研发部 20 市场部 30 运营部 40 测试部

创建员工表 emp，包含eid，ename，sex，birthday生日，salary工资，deptId隶属的部门编号

插入14条记录，分属于10/20/30部门，其中最后一个员工不属于任何一个部门。

3.项目中如何存储日期时间数据

2018-11-25

2018年11月25日

11-25-2018

25/11/2018

**存储的是距离计算机元年的毫秒数，使用BIGINT类型。**

计算机元年 1970-1-1 0:0:0

1秒钟=1000毫秒

数据库存储1000 1970-1-1 0:0:1

1000\*60 1970-1-1 0:1:0

1000\*60\*60 1970-1-1 1:0:0

2018年距离计算机元年的毫秒：48\*365\*24\*60\*60\*1000

男 女

男孩 女孩

男人 女人

man woman

girl boy

male female

1 0

4.简单查询

**(1)查询所有的列**

SELECT \* FROM emp;

\*代表所有的列

**(2)查询特定的列**

示例：查询所有员工的姓名，生日，工资。

SELECT ename,birthday,salary FROM emp;

练习：查询所有员工的编号，姓名，性别，工资

SELECT eid,ename,sex,salary FROM emp;

**(3)给列取别名**

示例：查询所有员工的姓名和工资，列名使用汉字。

SELECT ename AS 姓名,salary AS 工资 FROM emp;

练习：查询所有员工的编号，姓名，性别，生日；全部使用中文别名。

SELECT eid AS 编号,ename AS 姓名,sex AS 性别,birthday AS 生日 FROM emp;

练习：查询所有员工的编号，姓名，性别，生日；全部使用1个英文字母作为别名。

SELECT eid a,ename b,sex c,birthday d FROM emp;

|  |
| --- |
| AS关键字可以省略 |

**(4)只显示不同的值/合并相同的项**

练习：查询公司都有哪些性别的员工

SELECT DISTINCT sex FROM emp;

练习：查询公司员工都分布在哪些部门

SELECT DISTINCT deptId FROM emp;

**(5)查询时执行运算**

示例：计算5+3-7/2\*9

SELECT 5+3-7/2\*9;

练习：查询所有员工的姓名及其年薪，并且起别名。

SELECT ename AS 姓名,salary\*12 AS 年薪 FROM emp;

练习：假设所有员工工资加500，年终奖是5000，查询出姓名和年薪，并且起别名。

SELECT ename AS 姓名,(salary+500)\*12+5000 AS 年薪 FROM emp;

(6)查询的结果排序

示例：查询所有员工的信息，结果按照工资由低到高排序

SELECT \* FROM emp ORDER BY salary ASC;

示例：查询所有员工的信息，结果按照工资由高到低排序

SELECT \* FROM emp ORDER BY salary DESC;

练习：查询所有员工的信息，结果按照年龄由大到小。

SELECT \* FROM emp ORDER BY birthday ASC;

练习：查询所有员工的信息，结果按照姓名升序排序。

SELECT \* FROM emp ORDER BY ename ASC;

练习：查询所有员工的信息，按生日由大到小，若生日相同，再按照姓名升序排列。

SELECT \* FROM emp ORDER BY birthday DESC,ename ASC;

练习：查询所有员工信息，按照工资由大到小排序，要求女员工必须排在男员工的前边。

SELECT \* FROM emp ORDER BY sex,salary DESC;

|  |
| --- |
| asc -> ascend 升序  desc -> descend 降序  如果不加排序规则，默认是按照升序排列（asc）。  ORDER BY可以按照**数值、日期、字符串(**字符的编码)来排序 |

(7)条件查询

示例：查询编号为5的员工工资

SELECT \* FROM emp WHERE eid=5;

|  |
| --- |
| SQL语句中支持的比较运算符：  = != > < >= <=  等于 不等于 大于 小于 大于等于 小于等于 |

练习：查询出工资大于等于6000的员工所有信息。

SELECT \* FROM emp WHERE salary>=6000;

练习：查询出所有女员工的信息；

SELECT \* FROM emp WHERE sex=0;

练习：查询出员工姓名为king的所有信息

SELECT \* FROM emp WHERE ename='king';

练习：查询出不在10号部门的员工的信息。

SELECT \* FROM emp WHERE deptId!=10;

练习：查询出1990年之前出生的员工信息

SELECT \* FROM emp WHERE birthday<'1990-1-1';

练习：查询出1993年之后出生的员工信息

SELECT \* FROM emp WHERE birthday>'1993-12-31';

练习：查询出没有明确部门的员工信息

#SELECT \* FROM emp WHERE deptId=NULL; 错误

SELECT \* FROM emp WHERE deptId IS NULL;

|  |
| --- |
| 注意: NULL不能和任何值进行等于/不等于判定，包括NULL自己。可以使用 IS NULL 和 IS NOT NULL |

练习：查询所有有明确部门的员工信息

SELECT \* FROM emp WHERE deptId IS NOT NULL;

练习：查询出10号部门所有女员工的信息

SELECT \* FROM emp WHERE deptId=10 AND sex=0;

|  |
| --- |
| AND(并且) OR(或者) |

练习：查询出10号和30号部门所有员工信息

SELECT \* FROM emp WHERE deptId=10 OR deptId=30;

练习：查询出工资在6000~8000所有员工信息

SELECT \* FROM emp WHERE salary>=6000 AND salary<=8000;

SELECT \* FROM emp WHERE salary BETWEEN 6000 AND 8000;

|  |
| --- |
| between ... and ... >= 第1个值 <= 第2个值  not between ... and ... |

练习：查询出不在6000~8000所有员工信息。

SELECT \* FROM emp WHERE salary<6000 OR salary>8000;

SELECT \* FROM emp WHERE salary NOT BETWEEN 6000 AND 8000;

练习：查询出1990年出生的员工信息。

SELECT \* FROM emp WHERE birthday>='1990-1-1' AND birthday<='1990-12-31';

SELECT \* FROM emp WHERE birthday BETWEEN '1990-1-1' AND '1990-12-31';

练习：查询10号和20号和30号部门的员工信息。

SELECT \* FROM emp WHERE deptId=10 OR deptId=20 OR deptId=30;

SELECT \* FROM emp WHERE deptId IN(10,20,30);

练习：查询出不在10号和20号部门的员工信息

SELECT \* FROM emp WHERE deptId NOT IN(10,20);

(8)模糊条件查询

示例：查询出姓名中含有字母e的员工信息

SELECT \* FROM emp WHERE ename LIKE '%e%';

练习：查询出姓名以e结尾的员工信息

SELECT \* FROM emp WHERE ename LIKE '%e';

练习：查询出姓名中倒数第2个字符为e的员工信息

SELECT \* FROM emp WHERE ename LIKE '%e\_';

|  |
| --- |
| % 任意多个字符 >=0  \_ 任意一个字符 =1  上述两个符号不能和=连用，必须使用LIKE |

(9)分页查询

分页显示：假如查询的结果集中有太多的记录，一次显示不完，可以分多页显示。

|  |  |  |
| --- | --- | --- |
| 页码 | 开始 | 结束 |
| 第1页 | 0 | 4 |
| 第2页 | 5 | 9 |
| 第3页 | 10 | 14 |
| 第4页 | 15 | 19 |

开始 = (页码-1)\*每页数量

SELECT \* FROM emp LIMIT start,count;

start 开始的值

count 查询的数量

示例：每页大小是5，查询第1页的数据

SELECT \* FROM emp LIMIT 0,5;

练习：每页大小是5，查询第2页的数据

SELECT \* FROM emp LIMIT 5,5;

练习：每页大小是5，查询第3页的数据

SELECT \* FROM emp LIMIT 10,5;

练习：每页大小是5，查询第4页的数据

SELECT \* FROM emp LIMIT 15,5;

**注意：**LIMIT 后边的两个数值不能添加引号。

综合：

SELECT \* FROM emp

WHERE ...

ORDER BY...

LIMIT...

练习：查询出所有的男员工工资最高的前三个人。

课后任务：

(1)复习今天内容

(2)删除所有的代码，保留注释，重新编写SQL语句。