# 如鹏就业班笔记

数据库

1. 数据库（Datebase）是保存数据的仓库，可以方便的保存并使用数据。数据库管理系统（DBMS）是对数据库进行管理的软件，常用的有MYSQL，Oracle，DB2，MSSQLServer。MYSQL是免费且开源的。只要掌握了一种DBMS使用其他的也就没问题了。

2.MYSQL的部署结构



3. MYSQL的安装：

①将绿色版解压到没有中文和空格的文件夹下面

②每次运行数据库时都要运行bin目录下的mysqld.exe，可以将其设为开机启动项，方法见

<http://jingyan.baidu.com/article/5d368d1e137dae3f61c05756.html>

4. 数据库管理工具：

①数据库管理工具不是DBMS，它是用来管理DBMS的

②MYSQL数据库管理工具一般选用NaviCat Lite

③配置Navi Cat Lite之前要运行mysqld.exe进行服务器的链接

④必须将用户名和密码设置为root，否则出现无法连接1045的错误，解决办法是删除连接并新建连接重新设置

数据库之主键、外键

1. 主键（PrimaryKey）：①主键是数据行的唯一标示，没有重复的列才能作为主键。主键可以没有，但是这样不便于管理，因此最好设定主键。

②主键的选用方式有业务主键和逻辑主键，前者具有业务意义，人能够看懂，但是可能重复；后者没有业务意义，只能由程序读懂，但是能够保证唯一性，所以一般选择逻辑主键。

2. 外键（ForeignKey）：外键类似于函数调用，它将一些重复的数据单独制成另一张表（类似函数的封装），简化主表的内容和日后数据的修改维护。



数据库之建库建表

1、根节点点右键“新建数据库”，数据库名字取得有意义，比如“study1”、字符集建议用"UTF-8"。复习：读写编码一致就不会乱码。

2、在study1下的“表”节点下点右键“新建表”，“栏位”其实指的就是列(Column)/列，翻译的不好。Id(主键，潜规则的名称，int、不允许为空、点右键“主键”)、Name(varchar，长度为10，不允许为空)、Gender(bit，不允许为空)，保存为“T\_Persons”(我习惯的表命名规则T\_名词复数)

3、建表常见错误：列名/表名不要用可能的关键字、不要有空格(包括前后，一不小心输错试试)、不要有特殊字符

双击生成的表，手工插入两条数据，注意点“提交”按钮

4、MYSQL的存储引擎有很多，最常用的是InnoDB和MyISAM，MyISAM效率较高，但是不支持事务、外键约束等特性，因此一般建议用InnoDB，新版本默认也是InnoDB。怎么设：建表时“选项”→“引擎”。建好了如何看引擎：表上点右键“对象信息”。

数据库之常用数据类型

1. 文本：

CHAR(\*)：最多255个字节的定长字符串，它的长度必须在创建时**指定**

VARCHAR(\*)：最多255个字节的可变长度字符串，它的长度必须在创建时**指定**

TEXT：最大长度为64K字符的变长文本

TINYTEXT：最大长度为255字符的变长文本

MEDUIMTEXT：最大长度为16K字符的变长文本

LONGTEXT：最大长度为4GB字符的变长文本

2. 整数(考虑数据取值后选择尽可能小的类型)

tinyint：1字节。有符号值：-128 到127；无符号值：0到255

smallint：2字节。有符号值：-32768 到32767；无符号值：0到65535

mediumint：3字节。

int：4字节

bigint：8字节

3. 小数(需要**指定**长度和小数点，也就是显示宽度和小数位数)：

decimal：精确存储的小数，在内部用字符串存储，适合金额等要求精确的类型。别名：NUMERIC

float：4字节，单精度。会近似存储(\*)，效率比decimal高。

double：8字节，双精度。会近似存储(\*)，效率比decimal高。

4. 日期时间：

DATE：4字节。范围：1000-01-01——9999-12-31

TIME：3字节。范围：-838:59:59——838:59:59

DATETIME：8字节。范围：1000-01-01 00:00:00——9999-12-31 23:59:59

5. 二进制大数据：

TITYBLOB：最大长度为255字节

BLOB：最大长度为64KB

MEDIUMBLOB：最大长度为16MB

LONGBLOB：最大长度为4GB

注意：图片等多媒体文件在数据库中用二进制格式保存（但是开发网站时不适合，因为网站的图片等文件不放在数据库里面，而是放在静态服务器里面，网站服务器只保存相关路径）

数据库之SQL语句入门

1. SQL语句中字符串用单引号

2. SQL语句（包括列名）大小写不敏感

3. NavCat中找到执行SQL语句的地方“查询”→“新建查询”，编写SQL后点击【运行】执行SQL语句。如果只想执行NavCat中编写的一部分代码，而不是执行全部代码，只要选中要执行的代码，点击鼠标右键，选择“运行已经选择的”即可。

该列没有勾选允许控制就不能不填该列

4. 最简单的SQL：查看一个表的全部数据：select \* from t\_students

5. **insert**:

①简单的插入数据的SQL语句：insert into t\_students(Id,Name,Age,Gender) value(5,'Jim',20,1)。Insert语句可以省略列名，但是不推荐这样。

②如果插入的行中有些字段的值不确定，那么Insert的时候不指定那些列即可。不“允许为空”的列在插入时不能省略。

数据库之自动递增列和修改列数据类型

1. 点击设计表在自动递增上面打勾就可以在插入行时不手动设置行数，这样就可以避免字段并发，不用程序员手动用代码控制递增，那样效率太低。

2. 假如将一个列由允许为空改为不允许为空，那么还得确保之前为空的列数据合法。

3. 修改列数据的数据类型时要考虑旧的类型是否能够被转换为新的类型。比如旧的为字符类型而新的为int类型就不行。

数据库之数据更新（Update）

1. 更新一个列：update t\_students set age=30

2. 更新多个列： update t\_students set age=20,name='XBF'

3. 表达式：update t\_students set age=age+5

4. 更新一部分数据： update t\_students set age=20 where name='谢宝发'，用where语句表示只更新Name是'tom'的行，注意SQL中等于判断用单个=，而不是==。

5. Where中还可以使用复杂的逻辑判断update t\_students set age=100 where name='XBF' or age=21，or相当于Java中的||

update t\_students set age=100 where (name='XBF' and age=100) or age=21

6. Where中可以使用的其他逻辑运算符：or、and、not、<、>、>=、<=、!=（或<>）等

数据库之数据删除（Delete）

1. 删除全部数据：delete from t\_students

2. 使用where删除部分数据：delete from t\_students where name='谢宝发' or (name='XBF' and age<>(等价于！=)100)

3. delete只是删除数据，但是表还在，日后还有恢复数据的可能，而使用drop table T\_students，则是将表删除。

数据库之数据初始化和select语法1

1. 检索所有数据：select \* from T\_employees

2. 部分检索：select name,age from t\_employees;

3. 列的别名检索：select name as '姓名',age as '年龄' from t\_employees;

4. 计算列select name,age+1,age from t\_employees;

5. 查看时间：select now();

6. 无意义检索：select 1+1;

7. where条件检索：select \* from T\_employees where age>25;

注意：只运行一行可以不加；，但是运行多行要加；

数据库之聚合函数

1、SQL聚合函数：MAX（最大值）、MIN（最小值）、AVG （平均值）、SUM （和）、COUNT（数量）

2、大于25岁的员工的最高工资 ：select max(salary) from t\_employees where age>25;

3、最低工资和最高工资：select max(salary) as '最高工资',min(salary) as '最低工资' from t\_employees where age<=25;

4. 行数（员工数）：select count(\*) from T\_employees;

select count(salary) from T\_employees;

select count(name) from t\_employees;

4、select count(\*) from T\_employees where age>25;

5、全体员工的工资总和平均工资：select sum(salary),avg(salary) from t\_employees;

数据库之数据排序

1、ORDER BY子句位于SELECT语句的末尾，它允许指定按照一个列或者多个列进行排序，还可以指定排序方式是升序(从小到大排列，ASC)还是降序(从大到小排列，DESC)

2、按照年龄升序排序所有员工信息的列表：select \* from t\_employees order by age asc;

3、按照薪水降序排序所有员工信息的列表：select \*from t\_employees order by salary desc;

4、按照年龄从大到小排序，如果年龄相同则按照工资从大到小排序 ：select \* from t\_employees order by age desc,salary asc;

4、order by子句要放到where子句之后 ：select \*from t\_employees where age>25 order by age desc,salary asc;

数据库之通配符（like）

1、**单字符匹配**的通配符为半角下划线“\_”，它匹配单个出现的字符。以任意字符开头，剩余部分为“erry” ：select \*from t\_employees where name like '\_erry';

2、**多字符匹配**的通配符为半角百分号“%”，它匹配任意次数（零或多个）出现的任意字符。①检索姓名中包含字母“n”的员工信息 ：select \*from t\_employees where name like '%n%';

②检索姓名中以字母“T”开头的员工信息：select \*from t\_employees where name like 'T%';

③检索姓名中以字母“Y”结尾的员工信息：select \*from t\_employees where name like '%y';

3、Like性能较差，很容易造成全表扫描，谨慎使用。后面会讲数据库优化(索引等)，项目中做搜索用全文检索。

数据库之null

1、null是一个即时的随机对象，不同时刻的null是不知道相同不相同的。

2、数据库中，一个列如果没有指定值，那么值就为null，数据库中的null表示“不知道”，而不是表示没有。因此select null+1结果是null，因为“不知道”加1的结果还是“不知道”。

3、SELECT \* FROM T\_Employees WHERE NAME=null ；

SELECT \* FROM T\_Employees WHERE NAME!=null ；

结果为null，因为不知道name的值是否是null。**注意=和<>是判断值是否相等。**

4、SQL中使用is null、is not null来进行空值判断：

select \*from T\_employees where name is null;

select \*from t\_employees where name is not null;

**注意：is 和is not并不是判断值是不是相等，而是问：是不是不知道？**

数据库之limit

1、LIMIT关键字用来限制返回的结果集， LIMIT放在SELECT语句的最后位置，语法为“LIMIT 首行行号，要返回的结果集的最大数目” 。比如下面的SQL语句将返回Name不为空的、按照工资降序排列的从第二行开始（行号从0开始）的最多五条记录：

select \*from T\_employees where name is not null order by age desc limit 2,3;

2、**limit一定要放到所有的语句的最后**。

数据库之group by

1、数据分组用来将数据分为多个逻辑组，从而可以对每个组进行聚合运算。SQL语句中使用GROUP BY子句进行分组，使用方式为“GROUP BY 分组字段”。分组语句一般和聚合函数一起使用，GROUP BY子句负责将数据分成逻辑组（先），而聚合函数则对每一个组进行统计计算（后）。

2、查看公司员工有哪些年龄段的：

SELECT Age FROM T\_Employees GROUP BY Age

将Age相同的数据行放到一组，分组后的数据可以看作一个临时的结果集，而SELECT Age语句则取出每组的Age字段的值，这样我们就得到上表的员工年龄段表了。

3、如果SELECT语句有WHERE子句，则GROUP BY子句必须放到WHERE语句的之后。

4、分组后就可以对组内的数据采用聚合函数进行统计了：

1）计算每个分组中的员工平均工资

SELECT Age,AVG(Salary) FROM T\_Employees

GROUP BY Age

2）查看每个年龄段的员工的人数：

SELECT Age,COUNT(\*) FROM T\_Employees

GROUP BY Age

数据库之left join

1、查询每张订单的订单号、价格、对应的客户姓名以及客户年龄

SELECT o.Number,o.Price,c.Name,c.Age

FROM T\_Orders o

LEFT JOIN T\_Customers c

ON o.CustomerId=c.Id

2、添加where语句(显示价格>=150元的订单)

SELECT o.Number,o.Price,o.CustomerId,

c.Name,c.Age FROM T\_Orders o

LEFT JOIN T\_Customers c ON o.CustomerId=c.Id

WHERE o.Price>=150

3、可以join多张表：

SELECT o.Number 订单号,o.Price 价格, c.Name 客户姓名,c.Age 客户年龄,t.Name 订单类型 FROM T\_Orders o LEFT JOIN T\_Customers c ON o.CustomerId=c.Id LEFT JOIN T\_OrderTypes t ON o.TypeId=t.Id

数据库之外键约束

1、如果删除/更新T\_Customers一行记录，那么就可能会导致T\_Orders中存在CustomerId为非法值的数据，使得程序逻辑错误。一般不会更新主键Id的值，所以谈外键约束的时候只谈“删除T\_Customers时”。

2、外键约束：当删除T\_Customers中一条数据的时候，如何处理T\_Orders等存在指向T\_Customers外键的行。外键约束建立在外键字段\*\*\*Id的表上(t\_orders)。

3、建外键约束的方法：新建或者修改表的时候“外键”→“添加外键”。名：自动命名即可；栏位名：CustomerId；参考表：t\_customers；外栏位名：Id；删除时、更新时：一般默认RESTRICT(CASCADE:删除T\_Customers一行时把它的订单也删除了；SET NULL:删除T\_Customers一行时把它的订单CustomerId设置为NULL；NO ACTION/RESTRICT：拒绝删除)。

4、有的公司不习惯建外键，而是通过程序进行数据合法性控制，对于初学者先不建议这样，都把外键加上。