

搭建MSSQL Server 环境

mssqlserver+docker部署

1.具备条件

服务器需要大于2G内存。如果不够则可能无法正常启动，查看日志报如下错误：

This program requires a machine with at least 2000 megabytes of memory

2.已部署好docker 环境

3.仓库中搜索相关mssql数据库信息

```
docker search mssql
```

```
[root@node1 ~]# docker search mssql
INDEX          NAME
DESCRIPTION                                STARS    OFFICIAL  AUTOMATED
docker.io      docker.io/microsoft/mssql-server-linux
Deprecated SQL Server on Linux Container R...  1215
docker.io      docker.io/microsoft/mssql-server-windows-developer
Official Microsoft SQL Server Developer Ed...  439
docker.io      docker.io/microsoft/mssql-server-windows-express
Official Microsoft SQL Server Express Edit...  368
docker.io      docker.io/microsoft/mssql-tools
Official images for Microsoft SQL Server C...  51
docker.io      docker.io/rsmoorthy/mssql
Database (version SQL2000)                    12      [OK]      MSSQL
```

4. 下载镜像 mssql-server-linux

```
docker pull docker.io/microsoft/mssql-server-linux
```

5. 查看刚才下载镜像信息

```
docker image ls
```

REPOSITORY	CREATED	SIZE	TAG	IMAGE ID
docker.io/microsoft/mssql-server-linux			latest	
314918ddaedf	22 months ago	1.35 GB		

6. 运行mssql

```
docker run -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=sa123SA?' -p 33062:1433 --name
sqlserver -d microsoft/mssql-server-linux
```

指定参数和密码

容器名: sqlserver

镜像名: docker.io/microsoft/mssql-server-linux

容器内端口1433, 对外暴露端口33061

7. 查看运行的mysql容器

```
[root@node1 ~]# docker ps -l
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
af092620c2b6	docker.io/microsoft/mssql-server-linux	"/opt/mssql/bin/sq..."	22 seconds ago	Exited (1)	21 seconds ago	sqlserver

8.查看运行的mysql容器,mssql 容器内端口1433, 对外暴露端口33061

```
[root@node1 ~]# docker ps -l
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
af092620c2b6	docker.io/microsoft/mssql-server-linux	"/opt/mssql/bin/sq..."	9 minutes ago	Up 14 seconds	0.0.0.0:33061->1433/tcp	sqlserver

12.登录 sqlserver 容器

```
[root@node1 ~]# docker exec -it sqlserver bash
root@29af0884be94:/#
```

13.链接 sqlserver

```
/opt/mssql-tools/bin/sqlcmd -S localhost -U SA -P sa123SA?
```

14.执行mssql,查看有那些数据库最后加GO执行

```
1> SELECT Name from sys.Databases
2> GO
Name
-----
master
tempdb
model
msdb
```

15.创建数据库

```
create database mytest
GO
```

16.退出mssql

```
quit
```

数据检索--查询

语法: *代表所有字段

select */字段名称列表 from 表列表

```
select StudentNo,StudentName,Sex,[Address] from Student
--可以为标题设置 别名, 别名可以是中文别名
select StudentNo as 学号,StudentName 姓名,性别=Sex,[Address] from Student
--添加常量列
select StudentNo as 学号,StudentName 姓名,性别=Sex,[Address] ,国籍='中华人民共和国'
from Student
```

```
--select的作用
--1.查询
--2.输出
select 1+1
--+是运算符,系统会自动为你做类型转换
select 1+'1'
select '1'+1
--如果+两边都是字符串,那么它就是一字符串连接符
select '1'+'1'
select 'a'+1
--可以输出多列值
select 1,2,34,3,545,67,567,6,7
--Top、Distinct
select * from Student
--top可以获取指定的记录数,值可以大于总记录数,但是不能是负值
select top 100 * from Student
--百分比是取ceiling()
select top 10 percent * from Student

--重复记录与原始的数据表数据无关,只与你查询的结果集有关系 distinct可以去除结果集中的重复记录--
结果集中每一列的值都一样
select distinct LoginPwd,Sex,Email from Student
select distinct Sex from Student
```

select的作用

```
--聚合函数:
--1.对null过滤
--2.都需要有一个参数
--3.都是返回一个数值
--sum(): 求和:只能对数值而言,对字符串和日期无效
--avg(): 求平均值
--count(): 计数:得到满足条件的记录数
--max(): 求最大值:可以对任意类型的数据进行聚合,如果是字符串就比较拼音字母进行排序
--min(): 求最小值
--获取学员总人数
```

```

select COUNT(*) from Student
--查询最大年龄值
select MIN(BornDate) from Student
select max(BornDate) from Student

--查询总分
select SUM(StudentResult) from Result where StudentNo=2
--平均分
select avg(StudentResult) from Result where SubjectId=1
--注意细节:
select SUM(StudentName) from Student
select SUM(BornDate) from Student

select min(StudentName) from Student
select max(StudentName) from Student

--查询学号, 姓名, 性别, 年龄, 电话, 地址 ---查询女生
select StudentNo, StudentName, Sex, BornDate, Address from Student where Sex='女' and
BornDate > '1990-1-1' and Address='马哥教育'
--指定区间范围
select StudentNo, StudentName, Sex, BornDate, Address from Student where BornDate
>= '1990-1-1' and BornDate <= '1993-1-1'
--between...and >= <=
select StudentNo, StudentName, Sex, BornDate, Address from Student where BornDate
between '1990-1-1' and '1993-1-1'
--查询班级id 1 3 5 7的学员信息
select * from Student where ClassId=1 or ClassId=3 or ClassId=5 or ClassId=7
--指定具体的取值范围--可以是任意类型的范围.值的类型需要一致--可以相互转换
select * from Student where ClassId in(1,3,'5',7)
select * from Student where ClassId not in(1,3,'5',7)

```

聚合函数

```

--带条件的查询-模糊查询-- 只针对字符串而言

--查询 姓 林 的女生信息
--=是一种精确查询, 需要完全匹配
select * from Student where Sex='女' and StudentName='林'
--通配符--元字符
--%: 任意个任意字段 window:* 正则表达式 : .*
--_:任意的单个字符
--[ ]:代表一个指定的范围, 范围可以是连续也可以是间断的。与正则表达式完全一样[0-9a-zA-Z]. 可以从
这个范围中取一个字符
--[^]:取反值
select * from Student where Sex='女' and StudentName='林%'
--通配符必须在模糊查询关键的中才可以做为通配符使用, 否则就是普通字符
--like 像。。。。一样
select * from Student where Sex='女' and StudentName like '林%'
select * from Student where Sex='女' and StudentName like '林_'
--[ ]的使用 学号在11~15之间的学员信息
select * from Student where StudentNo like '[13579]'

---处理null值
--null:不是地址没有分配, 而是不知道你需要存储什么值 所以null是指不知道。但是=只能匹配具体的
值, 而null根本就不是一个值

```

```

select COUNT(email) from Student where Email !=null
select COUNT(email) from Student where Email is null
select count(email) from Student where Email is not null
--将null值替换为指定的字符串值
select StudentName,ISNULL(Email,'没有填写电子邮箱') from Student where ClassId=2

```

模糊查询

```

--当你看到 每一个, , 各自, 不同, , 分别 需要考虑分组
--查询每一个班级的男生人数
--与聚合函数一起出现在查询中的列, 要么也被聚合, 要么被分组
select classid,Sex,COUNT(*) from Student where Sex='男' group by ClassId,sex
--查询每一个班级的总人数,显示人数>=2的信息
--1.聚合不应出现在 WHERE 子句中--语法错误
select ClassId ,COUNT(*) as num from Student where Email is not null GROUP by
ClassId having COUNT(*)>=2 order by num desc
--完整的sql查询家庭
--5          1          2
3          4
6
--select 字段列表 from 表列表 where 数据源做筛选 group by 分组字段列表 having 分组结果集
做筛选 order by 对结果集做记录重排

select ClassId ,COUNT(*) as num from Student where Email is not null GROUP by
ClassId order by ClassId desc

--关于top的执行顺序 排序之后再取top值
select top 1 ClassId ,COUNT(*) as num from Student GROUP by ClassId order by
num desc

```