

Linux运维必会的mysql企业面试题大全

Linux (<http://tn.51cto.com/topic/45>)

MySQL (<http://tn.51cto.com/topic/27>)

Centos (<http://tn.51cto.com/topic/49>)

SQL (<http://tn.51cto.com/topic/30>)

Unix (<http://tn.51cto.com/topic/46>)

1.开启MySQL服务 /etc/init.d/mysqld start service mysqld start systemctl start mysqld 2.检测端口是否运行 lsof -i :3306 netstat -lntup |grep 3306 3.为MySQL设置密码或...

1.开启MySQL服务

```
/etc/init.d/mysqld start
```

```
service mysqld start
```

```
systemctl start mysqld
```

2.检测端口是否运行

```
lsof -i :3306
```

```
netstat -lntup |grep 3306
```

3.为MySQL设置密码或者修改密码

设置密码

```
mysql -uroot -ppassword -e "set password for root = password('password')"
```

```
mysqladmin -uroot password "NEWPASSWORD"
```

更改密码

```
mysqladmin -uroot password oldpassword "NEWPASSWORD"
```

```
use mysql;
```

```
update user set password = PASSWORD('newpassword') where user = 'root';flush privileges;
```

```
mysql 5.7以上版本修改默认密码命令
```

```
alter user 'root'@'localhost' identified by 'root'
```

4.登陆MySQL数据库。

```
mysql -uroot -ppassword
```

5.查看当前数据库的字符集

```
show create database DB_NAME;
```

6.查看当前数据库版本

```
mysql -V
```

```
mysql -uroot -ppassword -e "use mysql;select version();"
```

7.查看当前登录的用户。

```
select user();
```

8.创建GBK字符集的数据库mingongge，并查看已建库完整语句

```
create database mingongge DEFAULT CHARSET GBK COLLATE gbk_chinese_ci;
```

9.创建用户mingongge，使之可以管理数据库mingongge

```
grant all on mingongge.* to 'mingongge'@'localhost' identified by 'mingongge';
```

10.查看创建的用户mingongge拥有哪些权限

```
show grants for mingongge@localhost
```

11.查看当前数据库里有哪些用户

```
select user from mysql.user;
```

12.进入mingongge数据库

```
use mingongge
```

13.创建—innoDB GBK表test，字段id int(4)和name varchar(16)

```
create table test (  
id int(4),  
name varchar(16)  
)ENGINE=innodb DEFAULT CHARSET=gbk;
```

14.查看建表结构及表结构的SQL语句

```
desc test;  
show create table test\G
```

15.插入一条数据 “1,mingongge”

```
insert into test values('1','mingongge');
```

16.再批量插入2行数据 “2,民工哥” , “3,mingonggeedu”

```
insert into test values('2','民工哥'),('3','mingonggeedu');
```

17.查询名字为mingongge的记录

```
select * from test where name = 'mingongge';
```

18.把数据id等于1的名字mingongge更改为mgg

```
update test set name = 'mgg' where id = '1';
```

19.在字段name前插入age字段，类型tinyint(2)

```
alter table test add age tinyint(2) after id;
```

20.不退出数据库,完成备份mingongge数据库

```
system mysqldump -uroot -pMgg123.0. -B mingongge >/root/mingongge_bak.sql
```

21.删除test表中的所有数据，并查看

```
delete from test;  
select * from test;
```

22.删除表test和mingongge数据库并查看

```
drop table test;  
show tables;  
drop database mingongge;  
show databases;
```

23.不退出数据库恢复以上删除的数据

```
system mysql -uroot -pMgg123.0. </root/mingongge_bak.sql
```

24.把库表的GBK字符集修改为UTF8。

```
alter database mingongge default character set utf8;  
alter table test default character set utf8;
```

25.把id列设置为主键，在Name字段上创建普通索引。

```
alter table test add primary key(id);  
create index mggindex on test(name(16));
```

26.在字段name后插入手机号字段(shouji)，类型char(11)。

```
alter table test add shouji char(11);  
#默认就是在最后一列后面插入新增列
```

27.所有字段上插入2条记录（自行设定数据）



```
insert into test values('4','23','li','13700000001'),('5','26','zhao','13710000001');
```

28.在手机字段上对前8个字符创建普通索引。

```
create index SJ on test(shouji(8));
```

29.查看创建的索引及索引类型等信息。

```
show index from test;
```

```
show create table test\G
```

#下面的命令也可以查看索引类型

```
show keys from test\G
```

30.删除Name，shouji列的索引。

```
drop index SJ on test;
```

```
drop index mggindex on test;
```

31.对Name列的前6个字符以及手机列的前8个字符组建联合索引。

```
create index lianhe on test(name(6),shouji(8));
```

32.查询手机号以137开头的，名字为zhao的记录（提前插入）

```
select * from test where shouji like '137%' and name = 'zhao';
```

33.查询上述语句的执行计划（是否使用联合索引等）

```
explain select * from test where name = 'zhao' and shouji like '137%'\G
```

34.把test表的引擎改成MyISAM。

```
alter table test engine=MyISAM;
```

35.收回mingongge用户的select权限

```
revoke select on mingongge.* from mingongge@localhost;
```

36.删除mingongge用户

```
drop user migongge@localhost;
```

37.删除mingongge数据库。

```
drop database mingongge
```

38.使用mysqladmin关闭数据库

```
mysqladmin -uroot -pMgg123.0. shutdown
```

```
ls -l :3306
```

39.MySQL密码丢了，请找回？

```
mysqld_safe --skip-grant-tables & #启动数据库服务
```

```
mysql -uroot -ppassowrd -e "use mysql;update user set passowrd = PASSWORD('newpassword') where user = 'root';flush privileges;"
```

MySQL运维基础知识面试题

面试题001：请解释关系型数据库概念及主要特点？

关系型数据库模型是把复杂的数据结构归结为简单的二元关系，对数据的操作都是建立一个或多个关系表格上，最大的特点就是二维的表格，通过SQL结构查询语句存取数据，保持数据一致性方面很强大

面试题002：请说出关系型数据库的典型产品、特点及应用场景？

mysql 互联网企业常用

oracle 大型传统企业应用软件

如数据备份、复杂连接查询、一致性数据存储等，还是使用MySQL或者其他传统的关系型数据库最合适

面试题003：请解释非关系型数据库概念及主要特点？

非关系型数据库也被称为NoSQL数据库，数据存储不需有特固定的表结构

特点：高性能、高并发、简单易安装

面试题004：请说出非关系型数据库的典型产品、特点及应用场景？

memcached 纯内存

redis 持久化缓存

mongodb 面向文档

如果需要短时间响应的查询操作，没有良好模式定义的数据存储，或者模式更改频繁的数据存储还是用NoSQL

面试题005：请详细描述SQL语句分类及对应代表性关键字。

sql语句分类如下

DDL 数据定义语言，用来定义数据库对象：库、表、列

代表性关键字：create alter drop

DML 数据操作语言，用来定义数据库记录

代表性关键字:insert delete update

DCL 数据控制语言，用来定义访问权限和安全级别

代表性关键字:grant deny revoke

DQL 数据查询语言，用来查询记录数据

代表性关键字:select

面试题006：请详细描述char(4)和varchar(4)的差别

char长度是固定不可变的，varchar长度是可变的（在设定内）

比如同样写入cn字符，char类型对应的长度是4(cn+两个空格),但varchar类型对应长度是2

面试题007：如何创建一个utf8字符集的数据库mingongge？

create database mingongge default character utf8 collate utf8_general_ci;

面试题008：如何授权mingongge用户从172.16.1.0/24访问数据库。

grant all on *.* to mingongge@'172.16.1.0/24' identified by '123456';

面试题009：什么是MySQL多实例，如何配置MySQL多实例？

mysql多实例就是在同一台服务器上启用多个mysql服务，它们监听不同的端口，运行多个服务进程

它们相互独立，互不影响的对外提供服务，便于节约服务器资源与后期架构扩展

多实例的配置方法有两种：

- 1、一个实例一个配置文件，不同端口
- 2、同一配置文件(my.cnf)下配置不同实例，基于mysqld_multi工具

面试题010：如何加强MySQL安全，请给出可行的具体措施？

删除数据库不使用的默认用户

配置相应的权限（包括远程连接）

不可在命令行界面下输入数据库的密码

定期修改密码与加强密码的复杂度

面试题011：MySQL root密码忘了如何找回？

面试题012：delete和truncate删除数据的区别？

前者删除数据可以恢复，它是逐条删除速度慢

后者是物理删除，不可恢复，它是整体删除速度快

面试题013：MySQL Sleep线程过多如何解决？

1、可以杀掉sleep进程，kill PID

2、修改配置，重启服务

[mysqld]

wait_timeout = 600

interactive_timeout=30

如果生产服务器不可随便重启可以使用下面的方法解决

set global wait_timeout=600

set global interactive_timeout=30;

面试题014：sort_buffer_size参数作用？如何在线修改生效？

在每个connection(session)第一次连接时需要使用到，来提访问性能

set global sort_buffer_size = 2M

面试题015：如何在线正确清理MySQL binlog？

MySQL中的binlog日志记录了数据中的数据变动，便于对数据的基于时间点和基于位置的恢复

但日志文件的大小会越来越大，占用大量的磁盘空间，因此需要定时清理一部分日志信息

手工删除：

首先查看主从库正在使用的binlog文件名称

show master(slave) status\G

删除之前一定要备份

purge master logs before '2017-09-01 00:00:00';

#删除指定时间前的日志

purge master logs to 'mysql-bin.000001';

#删除指定的日志文件

自动删除：

通过设置binlog的过期时间让系统自动删除日志

show variables like 'expire_logs_days';

set global expire_logs_days = 30;

#查看过期时间与设置过期时间

面试题016：Binlog工作模式有哪些？各有什么特点，企业如何选择？

1.Row(行模式)；

日志中会记录成每一行数据被修改的形式，然后在slave端再对相同的数据进行修改

2.Statement(语句模式)

每一条修改的数据都会完整的记录到主库master的binlog里面，在slave上完整执行在master执行的sql语句

3.mixed(混合模式)

结合前面的两种模式，如果在工作中有使用函数 或者触发器等特殊功能需求的时候，使用混合模式

数据量达到比较高时候，它就会选择 statement模式，而不会选择Row Level行模式

面试题017：误操作执行了一个drop库SQL语句，如何完整恢复？

1、停止主从复制，在主库上执行锁表并刷新binlog操作，接着恢复之前的全备文件（比如0点的全备）

2、将0点时的binlog文件与全备到故障期间的binlog文件合并导出成sql语句

mysqlbinlog --no-defaults mysql-bin.000011 mysql-bin.000012 > bin.sql

3、将导出的sql语句中drop语句删除，恢复到数据库中

mysql -uroot -pmysql123 < bin.sql

面试题018：mysqldump备份使用了-A -B参数，如何实现恢复单表？

-A 此参数作用是备份所有数据库（相当于--all-databases）

-B databasename 备份指定数据（单库备份使用）

面试题019：详述MySQL主从复制原理及配置主从的完整步骤

主从复制的原理如下：

主库开启binlog功能并授权从库连接主库，从库通过change master得到主库的相关同步信息

然后连接主库进行验证，主库IO线程根据从库slave线程的请求，从master.info开始记录的位置点

向下开始取信息，同时把取到的位置点和最新的位置与binlog信息一同发给从库IO线程

从库将相关的sql语句存放在relay-log里面，最终从库的sql线程将relay-log里的sql语句应用

到从库上，至此整个同步过程完成，之后将是无限重复上述过程

完整步骤如下：

- 1、主库开启binlog功能，并进行全备，将全备文件推送到从库服务器上
- 2、show master status\G 记录下当前的位置信息及二进制文件名
- 3、登陆从库恢复全备文件
- 4、执行change master to 语句
- 5、执行start slave and show slave status\G

面试题020：如何开启从库的binlog功能？

修改配置文件加上下面的配置

log_bin=slave-bin

log_bin_index=slave-bin.index

需要重启服务

面试题021：MySQL如何实现双向互为主从复制，并说明应用场景？

双向同步主要应用于解决单一主库写的压力，具体配置如下

主库配置

[mysqld]

auto_increment_increment = 2 #起始ID

auto_increment_offset = 1 #ID自增间隔

log-slave-updates

从库配置

[mysqld]

auto_increment_increment = 2 #起始ID

auto_increment_offset = 2 #ID自增间隔

log-slave-updates

主从库服务器都需要重启mysql服务

面试题022：MySQL如何实现级联同步，并说明应用场景？

级联同步主要应用在从库需要做为其它数据库的主库

在需要做级联同步的数据库配置文件增加下面的配置即可

log_bin=slave-bin

log_bin_index=slave-bin.index

面试题023：MySQL主从复制故障如何解决？

- 1、登陆从库，执行stop slave;停止主从同步

然后set global sql_slave_skip_counter = 1;跳过一步错误

最后执行 start slave;并查看主从同步状态

2、需要重新进行主从同步操作

进入主库，进行全备数据库并刷新binlog,查看主库此的状态

恢复全备文件到从库，然后执行change master

开启主从同步start slave;并查看主从同步状态

面试题024：如何监控主从复制是否故障？

```
mysql -uroot -ppassword -e "show slave status\G" |grep -E "Slave_IO_Running|Slave_SQL_Running"|awk '{print $2}'|grep -c Yes
```

通过判断Yes的个数来监控主从复制状态，正常情况等于2

面试题025：MySQL数据库如何实现读写分离？

1、通过开发程序实现

2、通过其它工具实现（如mysql-mmm）

面试题026：生产一主多从从库宕机，如何手工恢复？

执行stop slave 或者停止服务

修复好从库数据库

然后重新操作主库同步

面试题027：生产一主多从主库宕机，如何手工恢复？

1、登陆各个从库停止同步，并查看谁的数据最新，将它设置为新主库让其它从库同步其数据

2、修复好主库之后，生新操作主从同步的步骤就可以了

#需要注意的新的主库如果之前是只读，需要关闭此功能让其可写

#需要在新从库创建与之前主库相同的同步的用户与权限

#其它从库执行change master to master_port=新主库的端口，start slave

面试题028：工作中遇到过哪些数据库故障，请描述2个例子？

1、开发使用root用户在从库上写入数据造成主从数据不一致，并且前端没有展示需要修改的内容（仍旧是老数据）

2、内网测试环境服务器突然断电造成主从同步故障

面试题029：MySQL出现复制延迟有哪些原因？如何解决？

1、需要同步的从库数据太多

2、从库的硬件资源较差，需要提升

3、网络问题，需要提升网络带宽

4、主库的数据写入量较大，需要优配置和硬件资源

5、sql语句执行过长导致，需要优化

面试题030：给出企业生产大型MySQL集群架构可行备份方案？

1、双主多从，主从同步的架构，然后实行某个从库专业做为备份服务器

2、编写脚本实行分库分表进行备份，并加入定时任务

3、最终将备份服务推送至内网专业服务器，数据库服务器本地保留一周

4、备份服务器根据实际情况来保留备份数据（一般30天）

面试题031：什么是数据库事务，事务有哪些特性？企业如何选择？

数据库事务是指逻辑上的一组sql语句，组成这组操作的各个语句

执行时要么成功，要么失败

特点：具有原子性、隔离性、持久性、一致性

面试题032：请解释全备、增备、冷备、热备概念及企业实践经验？

全备：数据库所有数据的一次完整备份，也就是备份当前数据库的所有数据

增备：就在上次备份的基础上备份到现在所有新增的数据

冷备：停止服务的基础上进行备份操作

热备：实行在线进行备份操作，不影响数据库的正常运行

全备在企业中基本上是每周或天一次，其它时间是进行增量备份

热备使用的情况是有两台数据库在同时提供服务的情况，针对归档模式的数据库

冷备使用情况有企业初期，数据量不大且服务器数量不多，可能会执行某些库、表结构等重大操作时

面试题033：MySQL的SQL语句如何优化？

建立主键与增加索引

面试题034：企业生产MySQL集群架构如何设计备份方案？

集群架构可采用双主多从的模式，但实际双主只有一主在线提供服务，两台主之间做互备

另外的从可做读的负载均衡，然后将其中一台抽出专业做备份

面试题035：开发有一堆数据发给dba执行，DBA执行需注意什么？

- 1、需要注意语句是否有格式上的错误，执行会出错导致过程中断
- 2、还需要注意语句的执行时间是否过长，是否会对服务器负载产生压力影响实际生产

面试题036：如何调整生产线中MySQL数据库的字符集。

- 1、首先导出库的表结构 -d 只导出表结构，然后批量替换
- 2、导出库中的所有数据（在不产生新数据的前提下）
- 3、然后全局替换set names = xxxxx
- 4、删除原有库与表，并新创建出来，再导入建库与建表语句与所有数据

面试题037：请描述MySQL里中文数据乱码原理，如何防止乱码？

服务器系统、数据库、客户端三方字符集不一致导致，需要统一字符

面试题038：企业生产MySQL如何优化（请多角度描述）？

- 1、提升服务器硬件资源与网络带宽
- 2、优化mysql服务配置文件
- 3、开启慢查询日志然后分析问题所在

面试题039：MySQL高可用方案有哪些，各自特点，企业如何选择？

高可用方案有：主从架构、MMM MHA mysql+haproxy+drbd mysql+proxy+amoeba

面试题040：如何批量更改数据库表的引擎？

通过mysqldump命令备份出一个sql文件，再使用sed命令替换

或者执行下面的脚本进行修改

```
#!/bin/sh

user=root

passwd=123456

cmd="mysql -u$user -p$passwd "

dump="mysqldump -u$user -p$passwd"

for database in ` $cmd -e "show databases;" | sed '1,2d' | egrep -v "mysql|performance_schema" `
do

    for tables in `dump -e "show tables from $databases;" | sed '1d'`
    do

        $cmd "alter table $database.$tables engine = MyISAM;"
```


done

done

面试题041：如何批量更改数据库字符集？

通过mysqldump命令备份出一个sql文件，再使用sed命令替换sed -i 's/GBK/UTF8/g'

面试题042：网站打开慢，请给出排查方法，如是数据库慢导致，如何排查并解决，请分析并举例？

可以使用top free 等命令分析系统性能等方面的问题

如是因为数据库的原因造成的，就需要查看慢查询日志去查找并分析问题所在

发表于 5分钟前 阅读 (62)

0 推荐

收藏



(http://tn.51cto.com/question/1108) 4 回答

你可能感兴趣的文章

Linux三剑客命令之Cron (http://tn.51cto.com/article/366) 1 浏览

Linux三剑客命令之Sed (http://tn.51cto.com/article/366) 1 浏览

Linux命令三剑客之Awk (http://tn.51cto.com/article/365) 2 浏览

%20Tech%20Neo%20%E5%BC%80%E5%8F%91%E8%80%85%E7%A4%BE%E5%8C%BA)

(多图慎入)图解菜鸟到大拿实现财务自由走向人生巅峰之路 (http://tn.51cto.com/article/361) 142 浏览


Launchy - 经典实用的免费键盘快速启动工具 (http://tn.51cto.com/article/358) 9 浏览

相关问题

- 求助centos网络问题 (http://tn.51cto.com/question/1108) 4 回答
- 作为一个初级运维，公司使用的技术不是特别牛逼，该怎么提升呢？ (http://tn.51cto.com/question/1103) 8 回答
- 遇到一个mysql查询的疑问，求知道的朋友解答一下。具体见描述。 (http://tn.51cto.com/question/1098) 3 回答
- 软件实施人的职业发展，如果往技术发展有哪些方向？如果往管理发展有哪些方向？ (http://tn.51cto.com/question/1091) 5 回答
- 对于服务器的选择，如何在Windows和Linux中进行选择？ (http://tn.51cto.com/question/1087) 11 回答
- linux怎么免密码登陆？ (http://tn.51cto.com/question/1066) 2 回答

0 条评论

请先 登录 (http://tn.51cto.com/login) 后评论











民工哥 (http://tn.51cto.com/people/281)

5 篇文章

(http://tn.51cto.com/people/281)

作家榜 » (http://tn.51cto.com/top/articles)

1.		飞侠candy (http://tn.51cto.com/people/120)	84 文章
2.		开始 (http://tn.51cto.com/people/6)	47 文章
3.		床前明月光 (http://tn.51cto.com/people/179)	39 文章
4.		51CTO小官 (http://tn.51cto.com/people/5)	17 文章
5.		wireless_com (http://tn.51cto.com/people/191)	7 文章
6.		小官51CTO (http://tn.51cto.com/people/125)	6 文章
7.		民工哥 (http://tn.51cto.com/people/281)	5 文章
8.		向问天 (http://tn.51cto.com/people/204)	4 文章