

DBA基础

NSD DBA1

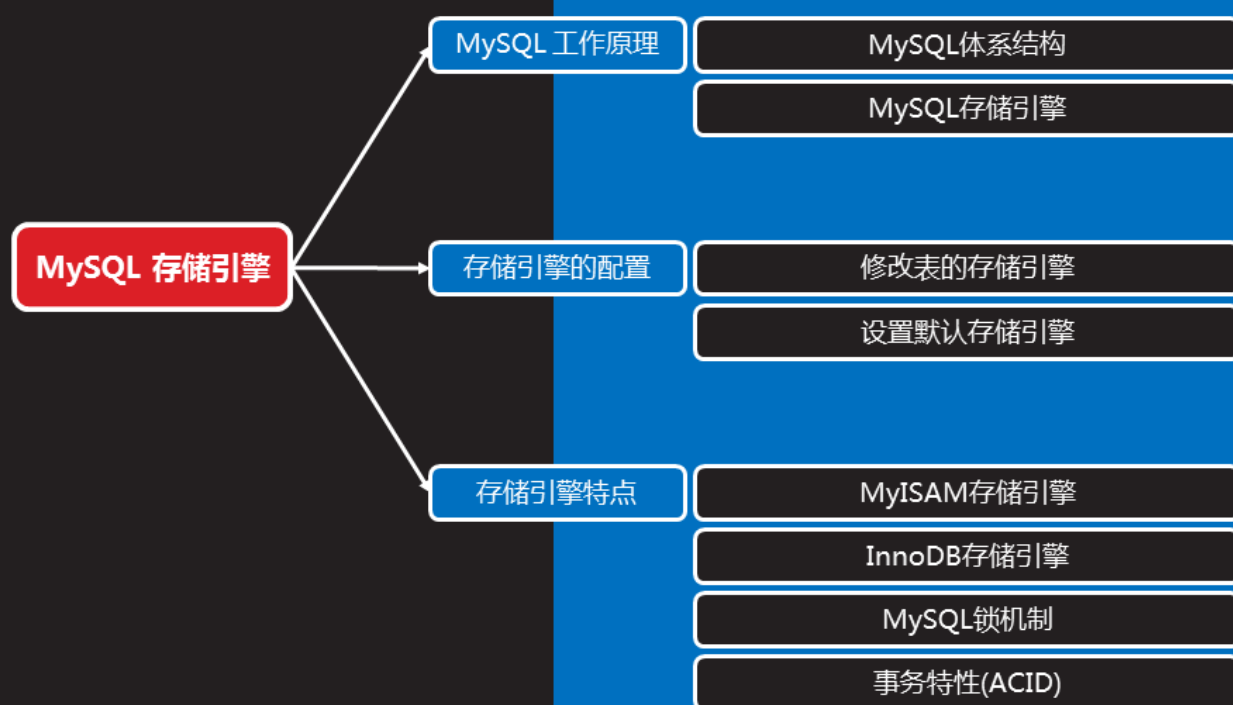
DAY03

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	MySQL存储引擎
	10:30 ~ 11:20	
	11:30 ~ 12:00	数据导入导出
下午	14:00 ~ 14:50	管理表记录
	15:00 ~ 15:50	匹配条件
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



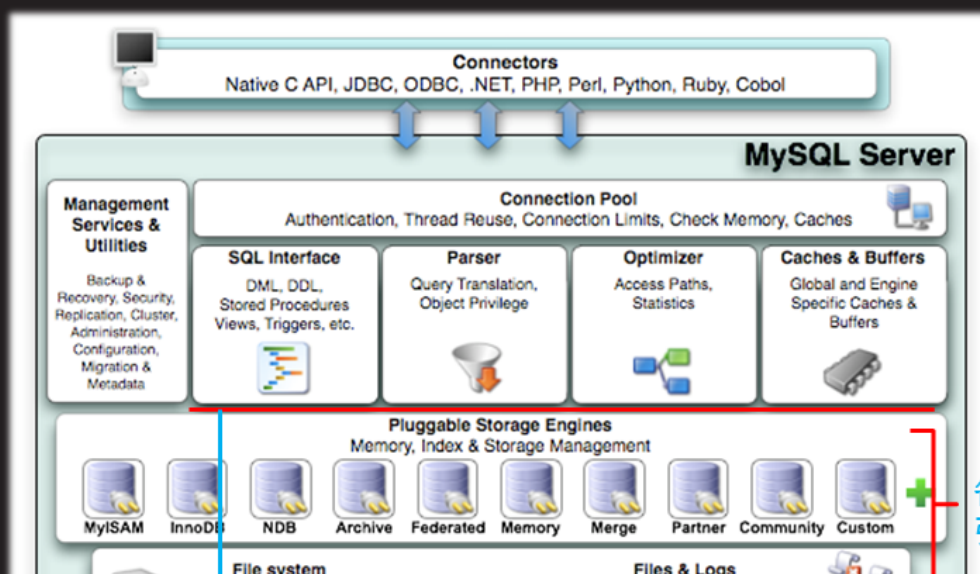
MySQL 存储引擎



MySQL 工作原理

MySQL 体系结构

知识讲解

各种存储
引擎组件

MySQL存储引擎

知识讲解

- 作为可插拔式的组件提供
 - MySQL服务软件自带的功能程序，处理表的处理器
 - 不同的存储引擎有不同的功能和数据存储方式
- 默认的存储引擎
 - MySQL 5.0/5.1 ---> MyISAM
 - MySQL 5.5/5.6 ---> InnoDB



存储引擎的配置



设置默认存储引擎

知识讲解

- 修改/etc/my.cnf配置文件
 - default-storage-engine=存储引擎名称

```
[root@dbsvr1 ~]# vim /etc/my.cnf
```

```
[mysqld]
```

```
...
```

```
default-storage-engine=myisam
```

```
[root@dbsvr1 ~]# systemctl restart mysqld
```

```
Shutting down MySQL....
```

[确定]

```
Starting MySQL.....
```

[确定]



存储引擎特点

myisam存储引擎

知识讲解

- 主要特点
 - 支持表级锁
 - 不支持事务、事务回滚、外键
- 相关的表文件
 - 表名.frm、
 - 表名.MYI
 - 表名.MYD



MySQL锁机制

知识讲解

- 锁粒度
 - 表级锁：一次直接对整张表进行加锁
 - 行级锁：只锁定某一行
 - 页级锁：对整个页面（MySQL管理数据的基本存储单位）进行加锁
- 锁类型
 - 读锁（共享锁）：支持并发读
 - 写锁（互斥锁、排它锁）：是独占锁，上锁期间其他线程不能读表或写表



MySQL锁机制（续1）

- 查看当前的锁状态
 - 检查Table_lock开头的变量，% 作通配符

知识讲解

```
mysql> SHOW STATUS LIKE 'Table_lock%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Table_locks_immediate | 70 |
| Table_locks_waited | 0 |
+-----+-----+
2 rows in set (0.00 sec)
```



事务特性（ACID）

- Atomic：原子性
 - 事务的整个操作是一个整体，不可分割，要么全部成功，要么全部失败。
- Consistency：一致性
 - 事务操作的前后，表中的记录没有变化。
- Isolation：隔离性
 - 事务操作是相互隔离不受影响的。
- Durability：持久性
 - 数据一旦提交，不可改变，永久改变表数据

知识讲解

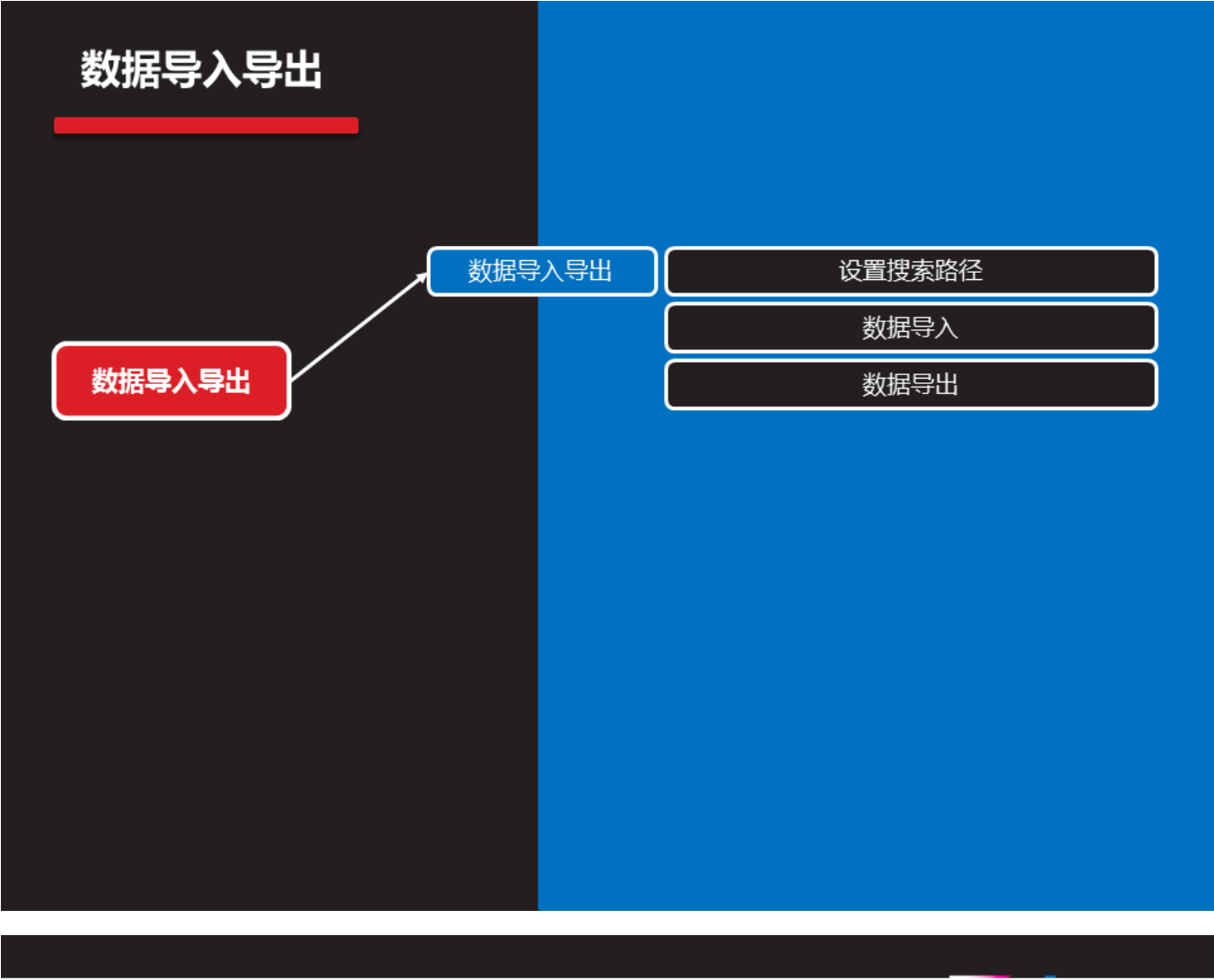


案例1：MySQL存储引擎的配置

1. 查看服务支持的存储引擎
2. 查看默认存储类型
3. 更改表的存储引擎
4. 设置数据库服务默认使用的存储引擎

课堂练习





设置搜索路径

- 查看默认使用目录及目录是否存在

```
mysql> show variables like "secure_file_priv";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| secure_file_priv | /var/lib/mysql-files/ |
+-----+-----+
```

```
[root@localhost ~]# ls -ld /var/lib/mysql-files/
drwxr-x---. 2 mysql mysql 31 4月 19 14:15 /var/lib/mysql-files/
```

知识讲解



设置搜索路径（续1）

- 修改目录及查看修改结果

```
[root@localhost ~]# mkdir /myload ; chown mysql /myload
[root@localhost ~]# vim /etc/my.cnf
[mysqld]
secure_file_priv="/myload"
[root@localhost ~]# systemctl restart mysqld
```

```
mysql> show variables like "secure_file_priv";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| secure_file_priv | /myload/ |
+-----+-----+
```

知识讲解



数据导入

知识讲解

- 基本用法
 - `LOAD DATA INFILE "目录名/文件名"`
`INTO TABLE 表名`
`FIELDS TERMINATED BY "分隔符"`
`LINES TERMINATED BY "\n";`
- 注意事项
 - 字段分隔符要与文件内的一致
 - 指定导入文件的绝对路径
 - 导入数据的表字段类型要与文件字段匹配
 - 禁用Selinux保护机制



数据导出

知识讲解

- 基本用法
 - `SQL查询 into outfile "目录名/文件名"`
`fields terminated by "分隔符"`
`lines terminated by "\n";`
- 注意事项
 - 导出的内容由SQL查询语句决定
 - 导出的是表中的记录，不包括字段名
 - 禁用SElinux



案例2：数据导入/导出

使用SQL语句完成下列导出、导入操作：

- 1) 将/etc/passwd文件导入userdb库user表并给每条记录加编号
- 2) 将userdb库user表中uid小于100的前10条记录导出，存为/myload/user2.txt文件

课堂练习



增加表记录

语法格式

知识讲解

- 格式1：添加1条记录，给所有字段赋值
 - insert into 表名 values (字段值列表)；
- 格式2：添加N条记录，给所有字段赋值
 - insert into 表名
values

(字段值列表),

第1条表记录

(字段值列表),

第2条表记录

(字段值列表);

第3条表记录



语法格式（续1）

知识讲解

- 格式3：添加1条记录，给指定字段赋值
 - insert into 表名 (字段名列表) values (字段值列表);
- 格式4：添加N条记录，给指定字段赋值
 - insert into 表名 (字段名列表)
values

(字段值列表),

第1条表记录

(字段值列表),

第2条表记录

(字段值列表);

第3条表记录



语法格式（续2）

知识讲解

- 注意事项
 - 字段值要与字段类型相匹配
 - 对于字符类型的字段，要用双或单引号括起来
 - 依次给所有字段赋值时，字段名可以省略
 - 只给一部分字段赋值时，必须明确写出对应的字段名称



查询表记录

语法格式

知识讲解

- 格式1
 - SELECT 字段1, ..., 字段N FROM 表名;
- 格式2
 - SELECT 字段1, ..., 字段N FROM 表名
 - WHERE 条件表达式;
- 注意事项
 - 使用 * 可匹配所有字段
 - 指定表名时, 可采用 库名.表名 的形式



更新表记录

语法格式

知识讲解

- 格式1，更新表内的所有记录
 - UPDATE 表名
 - SET 字段1=字段1值,字段2=字段2值,字段N=字段N值;



语法格式（续1）

知识讲解

- 格式2，只更新符合条件的部分记录
 - UPDATE 表名
 - SET 字段1=字段1值,字段2=字段2值,字段N=字段N值
 - WHERE 条件表达式;



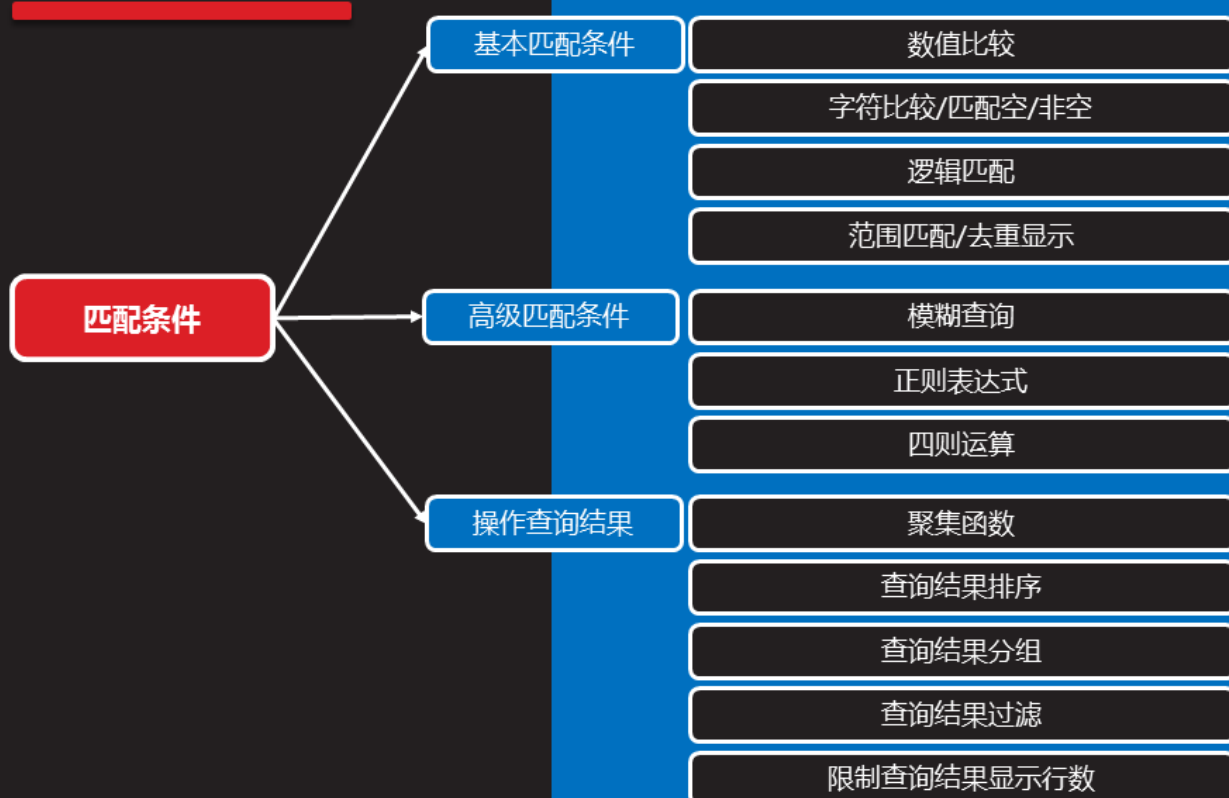
案例3：操作表记录

课堂练习

- 练习表记录的操作
 - 表记录的插入
 - 表记录的更新
 - 表记录的查询
 - 表记录的删除



匹配条件



基本匹配条件

数值比较

- 字段类型必须数据数值类型

类 型	用 途
=	等于
>、>=	大于、大于或等于
<、<=	小于、小于或等于
!=	不等于

