

数据库管理

NSD DATABASE

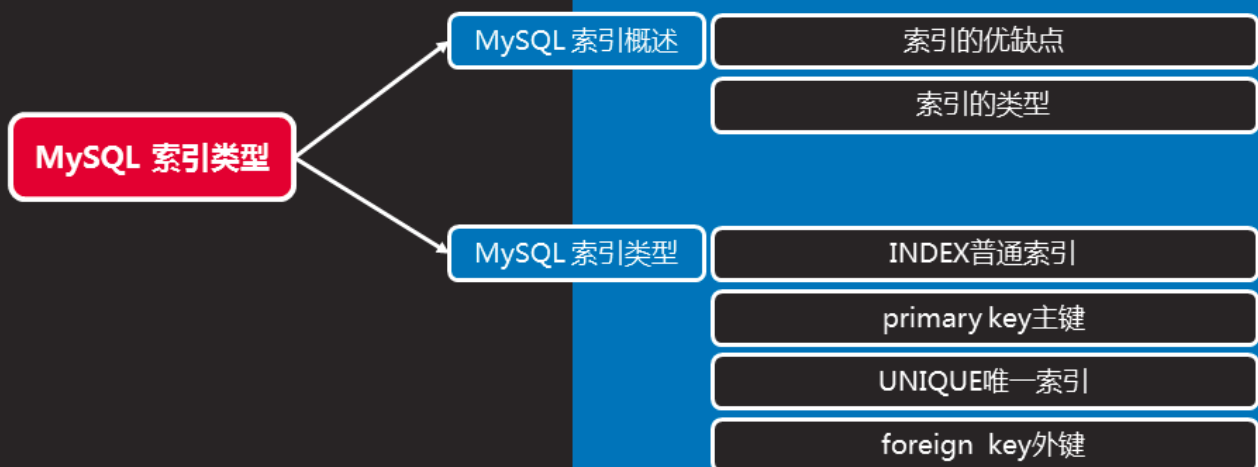
DAY02

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	MySQL 索引类型
	10:30 ~ 11:20	
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	MySQL 存储引擎
	15:00 ~ 15:50	
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



MySQL 索引类型



MySQL 索引概述

索引的优缺点

- 索引的优点
 - 通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性
 - 可以加快数据的检索速度
- 索引的缺点
 - 当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，降低了数据的维护速度
 - 索引需要占物理空间

索引的类型

知识讲解

- INDEX：普通索引
- UNIQUE：唯一索引
- FULLTEXT：全文索引
- PRIMARY KEY：主键
- FOREIGN KEY：外键



MySQL 索引类型

INDEX普通索引

知识讲解

- 使用说明
 - 一个表中可以有多个INDEX字段
 - 字段的值允许有重复，可以赋NULL值
 - 经常把做查询条件的字段设置为INDEX字段
 - INDEX字段的KEY标志是MUL



INDEX普通索引（续1）

- 建表的时候指定索引字段
 - INDEX(字段1), INDEX(字段2)

知识讲解

```
mysql> CREATE TABLE tea4(  
-> id char(6) NOT NULL,  
-> name varchar(4) NOT NULL,  
-> age int(3) NOT NULL,  
-> gender ENUM("boy","girl") DEFAULT "boy",  
-> INDEX(id), INDEX(name)  
-> );
```

Query OK,

```
mysql> DESC tea4;
```

Field	Type	Null	Key	Default	Extra
id	char(6)	NO	MUL	NULL	
name	varchar(4)	NO	MUL	NULL	
age	int(3)	NO		NULL	
gender	enum('boy','girl')	YES		boy	



INDEX普通索引（续2）

知识讲解

- 在已有的表中设置INDEX字段
 - CREATE INDEX 索引名 ON 表名(字段名);
- 删除指定表的索引字段
 - DROP INDEX 索引名 ON 表名;

```
mysql> DROP INDEX name ON tea4;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE INDEX nianling ON tea4(age);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0
```



INDEX普通索引（续3）

知识讲解

- 查看表的索引信息
 - SHOW INDEX FROM 表名;

```
mysql> SHOW INDEX FROM tea4\G
***** 1. row *****
      Table: tea4
    Non_unique: 1
      Key_name: id
   Seq_in_index: 1
   Column_name: id
     Collation: A
   Cardinality: 0
      Sub_part: NULL
        Packed: NULL
         Null:
   Index type: BTREE
      Comment:
```

使用B树算法



primary key主键

知识讲解

- 注意事项
 - 一个表中只能有一个primary key字段
 - 对应的字段值不允许有重复，且不允许赋NULL值
 - 如果有多个字段都作为PRIMARY KEY，称为复合主键，必须一起创建。
 - 主键字段的KEY标志是PRI
 - 通常与 AUTO_INCREMENT 连用
 - 经常把表中能够唯一标识记录的字段设置为主键字段 [记录编号字段]



primary key主键（续1）

知识讲解

- 建表的时候指定主键字段
 - PRIMARY KEY(字段名)

```
mysql> CREATE table tea6(
-> id int(4) AUTO INCREMENT,
-> name varchar(4) NOT NULL,
-> age int(2) NOT NULL,
-> PRIMARY KEY(id)
-> );
```

Query OK, 0 rows affected (0.02 sec)

```
mysql> DESC tea6;
```

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	NULL	auto_increment
name	varchar(4)	NO		NULL	
age	int(2)	NO		NULL	



primary key主键（续2）

知识讲解

- 在已有的表中设置PRIMARY KEY字段
 - ALTER TABLE 表名 ADD PRIMARY KEY(字段名);
- 移除表中的PRIMARY KEY字段
 - ALTER TABLE 表名 DROP PRIMARY KEY;

```
mysql> ALTER TABLE tea6 MODIFY id int(4) NOT NULL;  
Query OK, 0 rows affected (0.01 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> ALTER TABLE tea6 DROP PRIMARY KEY;  
Query OK, 0 rows affected (0.06 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

移除主键前，如果有自增属性，必须先去掉



UNIQUE唯一索引

知识讲解

- 使用说明
 - 一个表中可以有多个UNIQUE字段
 - 对应的字段值不允许有重复
 - UNIQUE字段的KEY标志是UNI
 - UNIQUE字段的值允许为NULL，当将其修改为不允许为NULL，则此字段限制与主键相同



UNIQUE唯一索引（续1）

- 建表的时候指定UNIQUE字段
 - UNIQUE(字段1), UNIQUE(字段2).. ..

知识讲解

```
mysql> CREATE TABLE tea5(
  -> id char(6),
  -> name varchar(4) NOT NULL,
  -> age int(3) NOT NULL,
  -> UNIQUE(id), UNIQUE(name), INDEX(age)
  -> );
Query OK,
```

```
mysql> DESC tea5;
```

Field	Type	Null	Key	Default	Extra
id	char(6)	YES	UNI	NULL	
name	varchar(4)	NO	PRI	NULL	
age	int(3)	NO	MUL	NULL	

不允许为NULL时，自动变成主键



UNIQUE唯一索引（续2）

- 在已有的表中设置UNIQUE字段
 - CREATE UNIQUE INDEX xxxx ON 表名(字段名) ;

知识讲解

```
mysql> DROP INDEX name ON tea5;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> CREATE UNIQUE INDEX name ON tea5(name);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

与删除INDEX索引的方法相同



foreign key外键

知识讲解

- 什么是外键？
 - 让当前表字段的值在另一个表中字段值的范围内选择。
- 使用外键的条件
 - 表的存储引擎必须是innodb
 - 字段类型要一致
 - 被参照字段必须要是索引类型的一种(primary key)



foreign key外键 (续1)

知识讲解

- 基本用法
 - FOREIGN KEY(表A的字段名)
References 表B(字段名)
ON UPDATE CASCADE
ON DELETE CASCADE



foreign key外键 (续2)

- 删除外键字段

- ALTER TABLE 表名 DROP FOREIGN KEY 约束名;

知识讲解

```
mysql> SHOW CREATE TABLE gz\G
***** 1. row *****
      Table: gz
Create Table: CREATE TABLE `gz` (
  `gz_id` int(4) NOT NULL,
  `name` char(16) NOT NULL,
  `gz` float(7,2) NOT NULL DEFAULT '0.00',
  KEY `name` (`name`),
  KEY `gz_id` (`gz_id`),
  CONSTRAINT `gz_ibfk_1` FOREIGN KEY (`gz_id`) REFERENCES `yg` (
`yg_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.00 sec)
```

```
mysql> ALTER TABLE gz DROP FOREIGN KEY gz_ibfk_1;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

外键约束名称



MySQL索引创建与删除

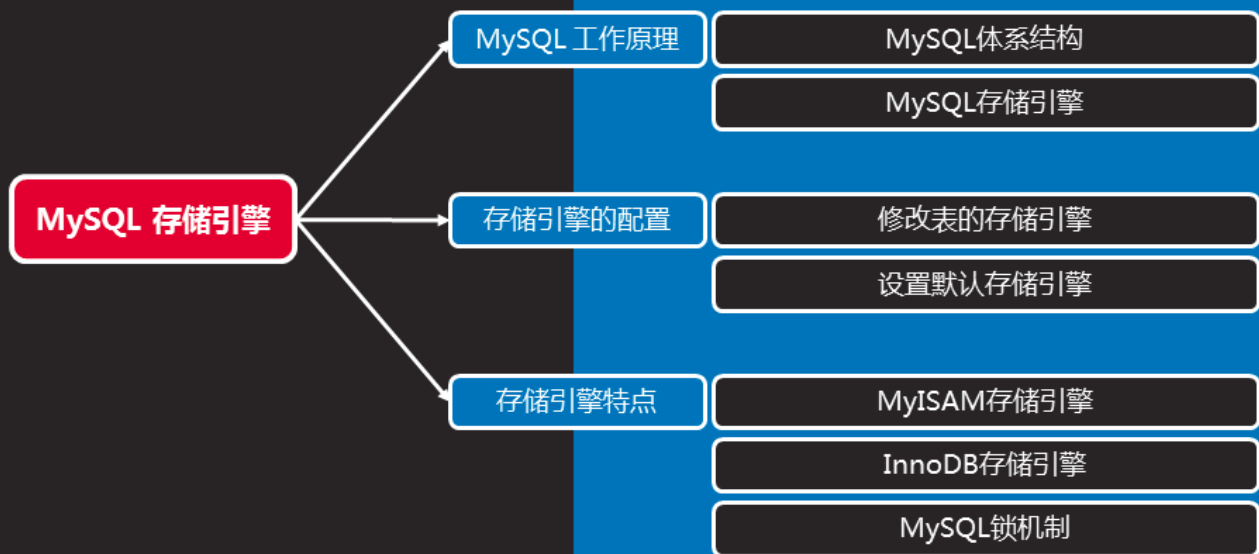
1. 普通索引、唯一索引、主键索引的创建/删除
2. 自增主键索引的创建/删除
3. 建立员工表yg、工资表gz，并设置外键实现同步更新与同步删除

课堂练习

yg表		gz表		
yg_id	name	gz_id	name	gz
1	Jerry	1	Jerry	12000
2	Tom	2	Tom	8000

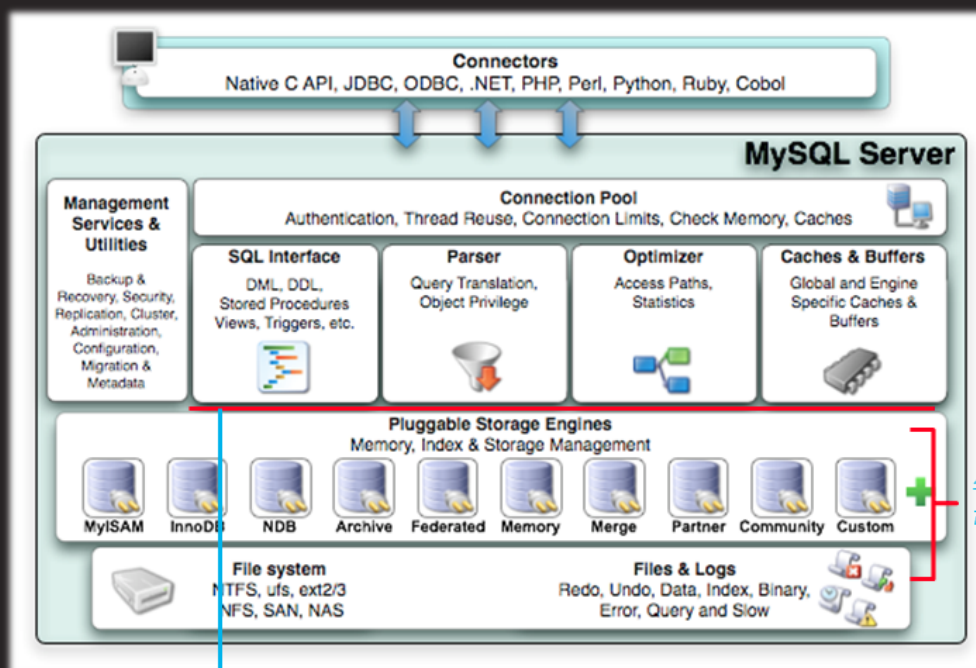


MySQL 存储引擎



MySQL 工作原理

MySQL体系结构



各种存储引擎组件

SQL接口/解析器/优化器/缓存

知识讲解

MySQL存储引擎

- 作为可插拔式的组件提供
 - 负责为数据库执行实际的数据I/O操作，不同的存储引擎，其存储数据的方式也不同
 - 用户通过连接器访问MySQL数据库，根据实际需要选用何种存储引擎，避免大的编码更改
- 默认的存储引擎
 - MySQL 5.0/5.1 ---> MyISAM
 - MySQL 5.5/5.6 ---> InnoDB

知识讲解

MySQL存储引擎(续1)

- 列出可用的存储引擎类型
 - SHOW ENGINES; 或 SHOW ENGINES\G

知识讲解

```
mysql> SHOW ENGINES;
```

Engine	Support	Comment
PERFORMANCE_SCHEMA	YES	Performance Schema
CSV	YES	CSV storage engine
MRG_MYISAM	YES	Collection of identical MyISAM tables
BLACKHOLE	YES	/dev/null storage engine (anything you write to it disappears)
MyISAM	YES	MyISAM storage engine
MEMORY	YES	Hash based, stored in memory, useful for temporary tables
ARCHIVE	YES	Archive storage engine
InnoDB	DEFAULT	Supports transactions, row-level locking, and foreign keys
FEDERATED	NO	Federated MySQL storage engine

9 rows in set (0.00 sec)



存储引擎的配置

修改表的存储引擎

知识讲解

- 建表时手动指定
 - 未指定时，使用默认存储引擎
 - SHOW CREATE TABLE xxx\G ; 可确认

```
mysql> USE test
Database changed
mysql> CREATE TABLE intab(
    -> id int(4)
    -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.06 sec)

mysql> SHOW CREATE TABLE intab\G
***** 1. row *****
      Table: intab
Create Table: CREATE TABLE `intab` (
  `id` int(4) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```



修改表的存储引擎（续1）

知识讲解

- 更改现有表的存储引擎
 - 使用 ALTER TABLE 改变引擎设置

```
mysql> ALTER TABLE intab ENGINE=MyISAM;
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> SHOW CREATE TABLE intab\G
***** 1. row *****
      Table: intab
Create Table: CREATE TABLE `intab` (
  `id` int(4) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```



设置默认存储引擎

知识讲解

- 修改/etc/my.cnf配置文件
 - default-storage-engine=xxxx

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

```
.. ..
```

```
default-storage-engine=InnoDB
```

```
[root@dbsvr1 ~]# service mysql restart
```

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

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

```
[确定]
```

```
[确定]
```



存储引擎特点

MyISAM存储引擎

知识讲解

- 主要特点
 - 支持表级锁定，不支持行级锁定
 - 不支持事务
 - 独享表空间
 - 多用在查询、插入操作较多的表
- 相关的表文件
 - xxx.frm、xxx.MYI、xxx.MYD



存放表结构



InnoDB存储引擎

知识讲解

- 主要特点
 - 支持表级/行级锁定
 - 支持事务、事务回滚、支持外键
 - 共享表空间
- 相关的表文件
 - xxx.frm、xxx.ibd
 - ibdata1
 - ib_logfile0、ib_logfile1



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）

知识讲解

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

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



案例2：MySQL存储引擎的配置

课堂练习

1. 可用的存储引擎类型
2. 查看默认存储类型
3. 更改表的存储引擎



总结和答疑

总结和答疑

primary key

问题现象1

故障分析及排除

问题现象2

故障分析及排除

foreign key

问题现象1

故障分析及排除

问题现象2

故障分析及排除

primary key

问题现象1

- 把已有表中的字段设置为主键报错
 - 报错： Duplicate entry '2' for key 'PRIMARY'

知识讲解

```
mysql> select * from t1;
| 2 | jerry |
| 2 | bob   |
+-----+-----+
4 rows in set (0.01 sec)
mysql>
mysql> alter table t1 add primary key(id);
ERROR 1062 (23000): Duplicate entry '2' for key 'PRIMARY'
```



故障分析及排除

知识讲解

- 原因分析
 - 不符合主键使用规则，主键字段的值不允许重复
- 解决办法
 - 修改字段的重复值

```
mysql> update t1 set id=4 where name="jerry";  
...  
mysql> alter table t1 add primary key(id);  
...
```



问题现象2

知识讲解

- 删除表中主键字段报错
 - 报错：ERROR 1075 (42000):

```
mysql> desc t2;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type   | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| id     | int(2) | NO   | PRI | NULL    | auto_increment |  
| name   | char(10)| YES  |     | NULL    |                |  
+-----+-----+-----+-----+-----+-----+
```

```
mysql> alter table t2 drop primary key;  
ERROR 1075 (42000): Incorrect table definition; there can be only  
one auto column and it must be defined as a key
```



故障分析及排除

知识讲解

- 原因分析：
 - 有自动增长属性的字段必须作为主键，不允许删除
- 解决办法：
 - 去掉字段的自动增长属性，再删除主键

```
mysql> alter table t2 modify id int(2) not null;
```

```
.. ..
```

```
mysql> alter table t2 drop primary key;
```

```
.. ..
```



foreign key

问题现象1

- 创建外键失败
 - 报错：ERROR 1215 (HY000): Cannot ...

知识讲解

```
mysql> desc t2 ;
+-----+-----+-----+-----+-----+-----+
| id   | int(2) | NO   |      | NULL   |      |
| name | char(10)| YES  |      | NULL   |      |
+-----+-----+-----+-----+-----+-----+
mysql> create table t3(
    -> id int(2),
    -> pay float(7,5),
    -> foreign key(id) references t2(id)
    -> );
ERROR 1215 (HY000): Cannot add foreign key constraint
```



故障分析及排除

- 原因分析：
 - 不符合外键使用规则。
 - 被参考字段必须是key值中的一种。
- 解决办法：
 - 给被参考字段设置key值

知识讲解

```
mysql> create index id on t2(id);
mysql> create table t3( id int(2), pay float(7,5), foreign key(id)
references t2(id) );
... ..
```



问题现象2

知识讲解

- 删除表中字段失败
 - 报错：Cannot delete or update a parent row

```
mysql> drop table t2;  
ERROR 1217 (23000): Cannot delete or update a parent row: a  
foreign key constraint fails  
  
mysql> alter table t2 drop id;  
ERROR 1553 (HY000): Cannot drop index 'id': needed in a foreign  
key constraint
```



故障分析及排除

知识讲解

- 原因分析
 - 被参考的表、表的字段不允许被删除
- 解决办法
 - 删除其他表对待删除表或字段的参考

```
mysql> alter table t3 drop foreign key t3_ibfk_1;  
..  
mysql> alter table t2 drop id;  
..
```



