

数据库管理

NSD DB 基础

DAY02

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	约束条件
	10:30 ~ 11:20	修改表结构
	11:30 ~ 12:00	MySQL 键值
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



约束条件

约束条件

- Null 允许为空，默认设置
- NOT NULL 不允许为空
- Key 索引类型
- Default 设置默认值，缺省为 NULL

```
mysql> DESC tea;
```

Field	Type	Null	Key	Default	Extra
name	varchar(4)	YES		NULL	
gender	enum('boy','girl')	YES		NULL	
interest	set('book','film','music','football')	YES		NULL	

```
3 rows in set (0.00 sec)
```



约束条件 (续 1)

```
mysql> CREATE TABLE tea2(
-> name varchar(4) NOT NULL,
-> gender enum("boy","girl") DEFAULT "boy",
-> age int(3) NOT NULL DEFAULT 21,
-> interest set("book","film","music","football")
-> );
```

Query OK, 0 rows affected (0.08 sec)

```
mysql> DESC tea2;
```

Field	Type	Null	Key	Default	Extra
name	varchar(4)	NO		NULL	
gender	enum('boy','girl')	YES		boy	
age	int(3)	NO		21	
interest	set('book','film','music','football')	YES		NULL	

4 rows in set (0.00 sec)



修改表结构

修改表结构

修改表结构

语法结构

添加新字段

修改字段类型

修改字段名

删除字段

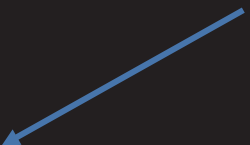
修改表名

修改表结构

语法结构

- 基本用法

- ALTER TABLE 表名 执行动作；



Add	添加字段
Modify	修改字段类型
Change	修改字段名
Drop	删除字段
Rename	修改表名



添加新字段

- 基本用法
 - _ ALTER TABLE 表名
 - _ ADD 字段名 类型(宽度) 约束条件;
可加 AFTER 字段名;
或者 FIRST;



修改字段类型

- 基本用法
 - _ ALTER TABLE 表名
 - _ modify 字段名 类型(宽度) 约束条件;
可加 AFTER 字段名;
或者 FIRST;



修改字段名

- 基本用法
 - _ ALTER TABLE 表名
 - _ change 源字段名 新字段名 类型(宽度) 约束条件;

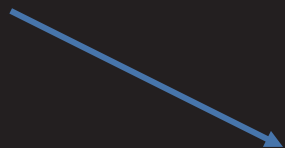


当跟新类型和约束条件时，也可修改字段类型



删除字段

- 基本用法
 - ALTER TABLE 表名
 - drop 字段名 ;



表中有多条记录时，所有列此字段的值都删除了



修改表名

- 基本用法
 - _ ALTER TABLE 表名
 - _ Rename 新表名;



表对应的文件名，也会改变



案例 1：修改表结构

- 表的字段修改
 - 添加字段
 - 修改字段名
 - 修改字段类型
 - 删除字段



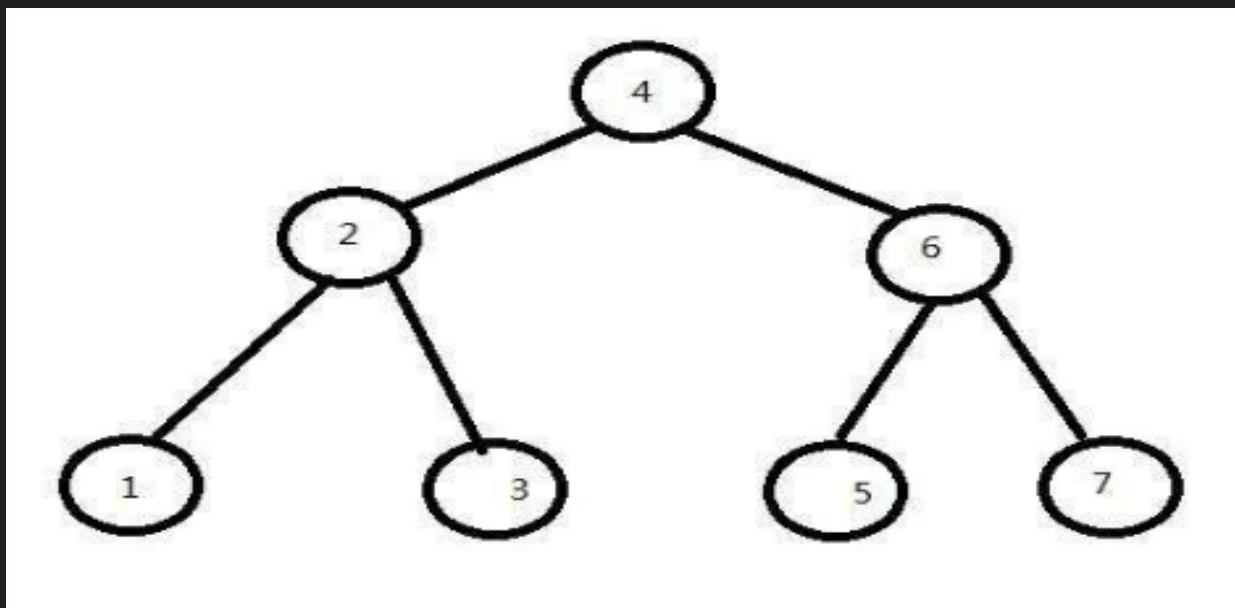
MySQL 键值



MySQL 索引概述

索引介绍

- 索引是什么？
 - 索引是对记录集的多个字段进行排序的方法。
 - 类似于书的目录
 - 索引类型包括 :Btree、 B+tree 、 hash



索引优缺点

- 索引优点
 - 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



键值类型

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



MySQL 键值类型



INDEX 普通索引

- 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，切可以赋 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)

```
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)

- 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)

- 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

- 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)

- 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)

- 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



foreign key 外键

- 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)

- 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;

```
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

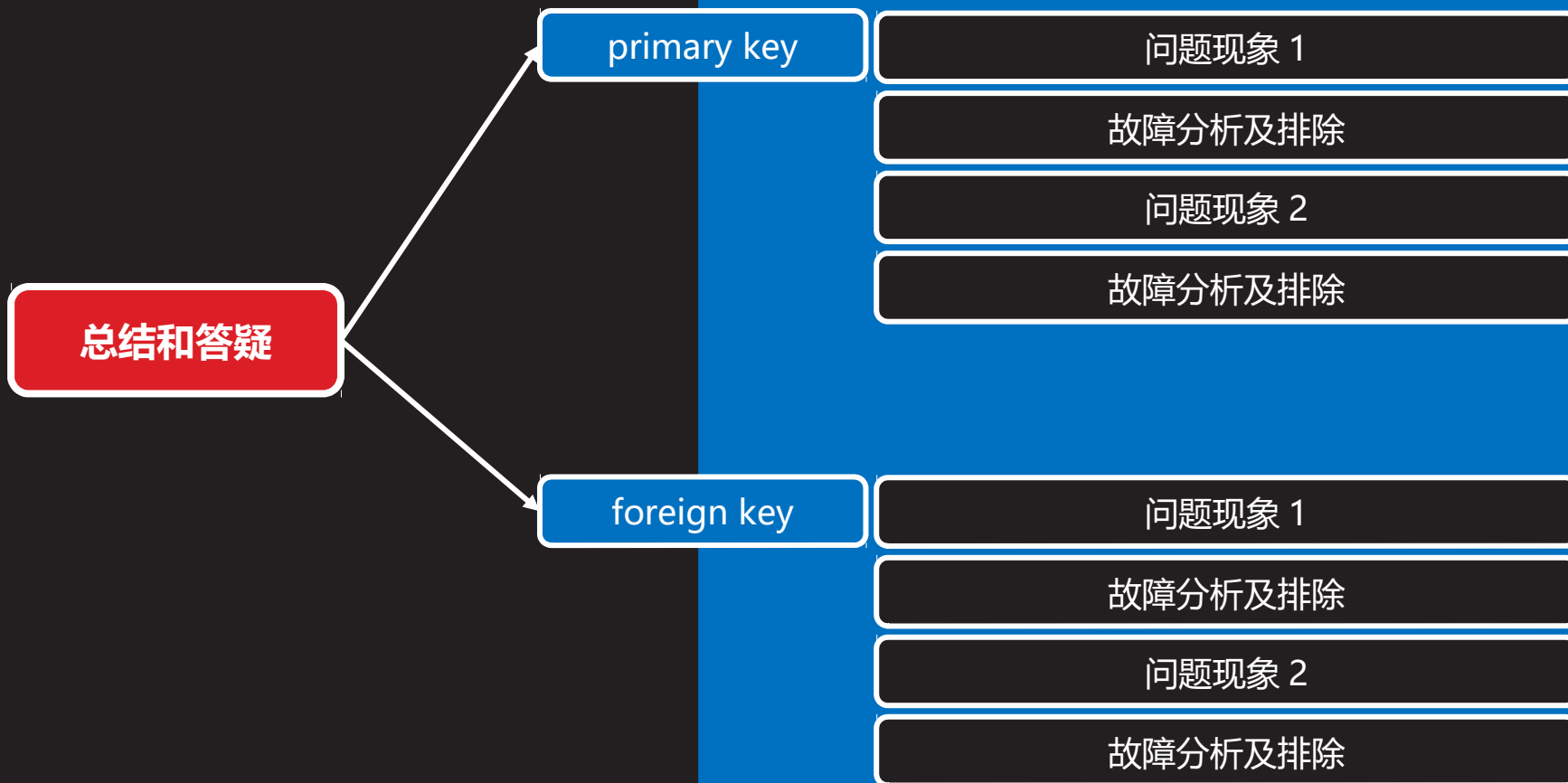
1. 普通索引、唯一索引、主键索引的创建 / 删除
2. 自增主键索引的创建 / 删除
3. 建立员工表 yg、工资表 gz，并设置外键实现同步更新与同步删除

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



总结和答疑



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

- 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

- 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

- 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**

```
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

```
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

- 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

- 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;
```

```
...
```

