

一对一关系


它是一种特殊的一对多，外键字段是唯一的

国家表

名	类型	长度	小数点	不是 null	虚拟	键	注释
ctr_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	国家编号
ctr_name	varchar	200	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		国家名称

总统表

添加外键字段ctr_id, 并且给它加一个唯一索引

名	类型	长度	小数点	不是 null	虚拟	键	注释
pre_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	总统编号
pre_name	varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		总统姓名
ctr_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		国家ID

字段	索引	外键	触发器	选项	注释	SQL 预览
名	字段	索引类型	索引方法	注释		
ctr_id	`ctr_id`	UNIQUE	BTREE			

一对一关联查询

```
select c.*,p.* from country c, president p where c.ctr_id=p.ctr_id;
```

修改数据

update 表名 set 字段名1=, 字段名2= where

```
update country set ctr_name='美利坚合众国' where ctr_id=1;
update new_student set stu_name='智友',stu_age=20,stu_birthday='2000-6-8' where stu_id=1;
```

删除数据

delete from 表名 where

truncate table 表名 (比delete不带条件删除速度更快)

```
-- 删除数据
delete from new_student where stu_id=8;
delete from new_student where stu_id>=50;

-- 删除全表数据
TRUNCATE table new_student;
```

常用表结构修改语句（DDL）

```
-- DDL（定义表结构的语句），DML（操作数据的语句）
ALTER TABLE emp add emp_salary int not null DEFAULT 0; -- 新增一个字段
alter table emp MODIFY emp_salary varchar(10); -- 修改字段的属性
alter table emp CHANGE emp_salary emp_sal varchar(20); -- 修改字段名字或属性
alter table emp drop COLUMN emp_sal; -- 删除字段
alter table emp rename to emp_new; -- 修改表的名称
alter table president add CONSTRAINT fk_president_country FOREIGN KEY(ctr_id)
    REFERENCES country(ctr_id); -- 添加外键
alter table president drop FOREIGN key fk_president_country; -- 删除外键
```

数据库里的其他对象

视图

```
-- 创建视图
create view v_student as
    select s.s_id,s.s_name,s.s_sex,s.s_age, c.c_name from t_stu s, t_class c
    where s.c_id=c.c_id;
-- 从视图查询数据
select * from v_student;
```

索引

```
-- 创建索引
CREATE index idx_emp_name on emp(emp_name);
```

存储过程

```
-- 存储过程
create PROCEDURE p_getage(in empid int, out empage int)
BEGIN
    select emp_age into empage from emp where emp_id=empid;
END;
-- 调用存储过程
set @id = 3;
call p_getage(@id, @age);
select @age;
```

触发器

事务

完整的业务功能：取钱， 下订单。。。

```
start TRANSACTION; -- 开启事务
update student set stu_age = 20 where stu_id=1;
update student set stu_age = 21 where stu_id=2;
-- commit; -- 提交事务
ROLLBACK; -- 回滚事务
```

事务的特性

四个特性ACID：

- A (Atomicity) 原子性
一个事务内的所有SQL是一个整体，不可再拆分。例如转账事务，必须一个账号减少，另一个账号增加，这两条语句是一个整体，不可分割
- C (Consistency) 一致性
保证数据的一致，完整。例如刚才转账的例子，两个账号之间不管怎么转，总的金额是不变
- I (Isolation) 隔离性
当一个事务还未完成时，另一个事务能否读取前一个事务过程中的数据

	脏读	不可重复读	幻读
Read Uncommitted	V	V	V
Read Committed	X	V	V
Repeatable read	X	X	V
Serializable	X	X	X

数据库有四种隔离级别：

1. 读未提交 (Read uncommitted)

- 事务未提交的数据能被别的事务读取到
- 会产生脏读现象（读取到了事务中间的数据，有可能那个事务被回滚，读到的数据就是脏数据），
- 不可重读（在一个事务中，执行两次同一个查询，读到不一样的值），
- 幻读（可能查询到新的数据，或者数据减少）

2. 读已提交 (Read committed)

只有提交后的事务数据才能被其它事务读取

可以避免脏读。会产生不可重读和幻读现象

3. 可重复读 (Repeatable read)

在一个事务中，多次读取同一个数据，结果是一样的

可以避免脏读和不可重读

会产生幻读现象

4. 串行化 (Serializable)

相当于是同步执行

可以避免脏读，不可重读，幻读

- D (Durability) 持久性

保证所有的数据都能永久保存到硬盘。

log 日志

锁

- 根据锁的类型分为：排他锁和共享锁
- 根据锁的粒度：表级锁，行级锁