

实验考试复习

- **第一章 SQL语句**

- 数据定义

- 创建表、属性、数据类型、约束、检查约束、主键、外键、索引
 - 修改：增加新列、修改属性数据类型
 - 删除

- 数据查询

- 单表查询

- 查询前五：select to 5*
 - 按XXX升序/降序排序：order by XXX(desc)
 - 运算：avg(),count(), 输出新属性select 'name', 操作
 - 属于某个集合 in('em1', 'em2')
 - 模糊匹配查询 like '%xxx%'
 - 删除重复项：select **distinct** xxx
 - 计算具有共同属性的xx值：group by
 - 作用于组，从中选择满足条件的组：having

- 连接查询

- 自然连接：自然联接为所有公共属性匹配具有相同值的元组，并且只保留每个公共列的一个副本，对于相同的属性要求其全部相等。
 - from table1 natural join table2 (SQL Server上不行)
 - 笛卡尔积：from table1, table2
 - 等值连接：from table1, table2 where table1.xxx=table2.xxx
 - 自连接，给表起别名：from table x, table y

- 外连接: from table1 join table2 on table1.xxx = table2.xxx
- 嵌套查询
 - 查找xxx最小的yyy: select yyy from table where xxx<= all()
 - = some() == in()
 - exists() 返回值是真假, 判断是否为空集: 是返回假
- 集合运算
 - 比较运算符必须确保子查询返回值是单值!!!
 - 并集===**union**: select() union select()
 - 交集===**intersect**
 - 补集===**except**
- 数据更新
 - 插入
 - insert into table-name(col) values('值')
 - 修改
 - update table-name set col=xxx where
 - 删除
 - 单行: delete table-name where.....
 - 多行:delete from table-name where.....
- 视图
 - **create view** view-name **as** select.....
 - with check option
- 数据控制——权限
 - 授权——grant
 - grant select on courses to public(向所有用户授以可查询权)

- **grant** insert, update **on** students to user-1 with grant option (只能插入和更新, 并获得可以传播权限的能力)
- **grant** update(score): 只可以更新score列
- 回收——revoke

第二章 数据库的完整性控制

- 实体完整性: 主键、外键等
 - 建表时: **constraint name primary key()**
 - 后续添加: **alter table** table-name **add constraint xxx foreign key(x) references** table-name-2(xxx)
 - 创建事务: **set xact_abort on begin transaction T1 commit transaction T1**
- 参照完整性
- 用户自定义完整性
 - 对某一属性创建唯一性约束: **sid char(5) constraint U1 unique; age int constraint U2 check(age < 29)**
 - 取消约束: **alter table xxx drop xxx**
 - 创建规则: **create rule** rule-name **as** @valueA in ('F', 'M')
 - 绑定 **exec sp_bindrule** 'rule-name', 'table-name.[xxx]'
 - 解除绑定: **exec sp_unbindrule** 'table-name.[xxx]'
- 触发器
 - insert 触发器: 先向inserted表中插入, 再检查, 若有效, 则插入
 - update 触发器: 先将原始数据移到deleted表, 然后新的数据插入到inserted表中
 - delete 触发器: 删除的数据放到deleted表
 - 创建: **create trigger T1 on** table-name **for** insert, update **as**
 - 删除: **drop trigger** dbo.T1

- **第三章 数据库的安全性控制**

- 用户标识与鉴别

- 创建登陆账户: `exec sp_addlogin 'myname', 'mypassword', 'mydb', 'ch' go | use mydb go | exec sp_grantdbaccess 'myname'`
- 撤销账户: `use mydb go | exec sp_revokedbaccess 'myname'; exec sp_droplogin 'myname';`

- **第四章 数据库事务**

- 事务的定义

- 创建存储过程: **`create procedure fun @varB char(10) as`**
- 调用存储过程: `exec fun @varb='xxx'`
- `begin tran commit tran rollback transaction`

- 事物与锁

- 事务与隔离级别

- **`set transaction isolation level read uncommitted`**

- 锁冲突与死锁

- 事务与事务日志

- 游标

```

USE school
GO
DECLARE @tid char(10), @salary int          -- 声明变量
DECLARE my_cursor CURSOR FOR                -- 声明游标
    SELECT tid, salary FROM teachers
OPEN my_cursor                              -- 打开游标
FETCH NEXT FROM my_cursor INTO @tid, @salary -- 读取记录并推进游标
WHILE @@fetch_status = 0
BEGIN
    IF @salary < 3000
        UPDATE teachers SET salary = @salary + 300 WHERE tid = @tid
    IF @salary >= 3000 and @salary < 4000
        UPDATE teachers SET salary = @salary + 200 WHERE tid = @tid
    IF @salary >= 4000
        UPDATE teachers SET salary = @salary - 200 WHERE tid = @tid
    FETCH NEXT FROM my_cursor INTO @tid, @salary -- 读取记录并推进游标
END
CLOSE my_cursor                            -- 关闭游标
DEALLOCATE my_cursor                       -- 释放游标

```

- 打印除字符串之外的: print 'string'+convert(varchar, @varA)
- 初始化变量的值用set

以上内容整理于 [幕布文档](#)