

2019.10.22

22 October 2019 09:02

最近Java成神之路应该继续看了，另外就是继续熟悉

Golang，继续学习Golang。

还有就是Spring相关。

另外开始做自己的小项目啊，先做一个多人聊天室出来看看

第X天，我们来扫个尾巴哈哈哈哈哈 😊

联合外键，关联到另一个表的联合主键，话说为什么会有联合主键的存在呢？有什么具体的案例么？

联合主键。。。会应用到什么场景中去？迫不及待想查一下了。

身份证号可以确定一个人的身份，但是不能用来把他作为主键。

假设有一张人员表，还有100张表关联到这张表。假如用身份证做主键，前几年升级身份证，15位变18

位，那么你有101个字段需要重新设置，没修改一个人员的身份证，就要附带更新100张表。

而所谓的无关，就是用自增字段或者UUID，或者类似Oracle rowid这样不包含任何业务信息，只代表这条记录在这张表里面位置的信息作为主键。无论业务如何变化，这个字段都不需要做修改（顶多因为业务量变大，修改字段长度，但不需要修改数据本身）。

哈哈，不知道你懂没有。

额，看看老师现在在干嘛。。好像是在讲check()这个

check型约束。。

据说学完Oracle以后有一个考试。。。

行吧

约束学了五种，你还记得有哪些吗

那些Oracle会帮你创建索引？unique和primary key

接下来要根据E-R图写实例图（才不）

其实是根据实例图写sql

与其说是“图”，不如说是“表”吧，就是一眼能看出来的。就像DataGrip建表时用的那种，只不过人家都帮我们写好DDL了，现在我们是自己写。。。

数据库这些基础性的知识还是蛮简单的，主要是深入的一些知识，明明面试可能考到（？），但是他也不讲，只能自己去网上学习咯，要是你没有这个意识那就是吃大亏了/

现在老师还在写建表语句 😊

其实本来以为来这里学更深入的知识，结果搞了半天是全复习了一遍，有什么用呢？我觉得还是学一些自己没有学过，但是面试、工作又会使用到的知识比较好。比如深入的学习一下JVM、并发多线程之类，还有SpringBoot、SpringCloud，明明这些课程都学过了，还要装没学过似的再学一遍，完全是浪费时间。我也不是说不学这些，我是说这些完全可以自己复习到。而那些更深入的知识，我还是希望能有一个人带我学，而不是一个人干啃📖。

emmmm。。。可是现在也只能靠自己了。

well, never give up.

现在说到哪儿了？用已有表建新表？

```
Create table emp41(name, salary, did)
```

As

```
select last_name,salary,dept_id
```

```
from s emp
```

where dept_id = 41

从别的表复制过来的，约束不是很全。只拷见了not null，剩下的约束都不见了呀哈哈哈哈哈哈哈哈哈哈

[illegible]

😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈
😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈
哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈
😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈
😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈
😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈😄哈哈

、如果我只想拷贝表的结构，不想拷贝表的数据咋办呢？哈哈，你猜怎么着？在where后面写的是where 1=2，哈哈哈哈哈，是不是很假呀，以后还能见到1=1呢哈哈哈哈哈

红烧牛肉饭哈哈哈哈哈，还天魔超市哈哈哈哈哈

过不了春节po要负责任哈哈哈哈哈
哈哈哈哈哈

DML和TCL语句

咦，昨天没有学🐶？



(◎_◎)这咋不动吖

emmmm，现在在讲insert语句哦，一般格式我都回了，有啥特殊用法吗？



E看来可以插入一些特殊值，比如USER \NULL\DATE啥的，不过Date要转特定格式吗？这个我还真不知道。

有主外键关系，先往父表中插入数据，然后再往子表中插入有关联关系的数据。。这个应该挺清楚的了。

记得当年做课设写程序的时候，设计表还挺累的。

！！insert也可以用select语句啊，

今天仍然要喝5杯水啊，2500ml要达标的噢

看个例子

```
Insert into some-table(col1,col2)
```

```
Select col1, col2
```

```
From another-table
```

```
Where some = rules
```

发现了没？我们这时候没有用values这个关键字啊

然后我们来看UPDATE，唉，这个字体这怎么总不对。。不管了 😞



王的无奈

```
Update student
```

```
Set coln = value, colx = value
```

```
Where id = 1
```

Emmm,果然，接下来就是和主外键有关的错误了，其实编程中千锤百炼，已经知道这些简单的错误了 ✕

😏 windows 这个 emoji 还挺好看的，比安卓。。。

修改外键列中的值时候，一定要确保其在父表中由对应值，

删除 DELETE，delete + 表名 + [Where 语句]，如果没有 Where 语句，那这个表的所有数据就都删除了，但是这个表还在的哦。如果是想删除表的话，应该用 DROP 这个关键字。

如果有主外键的关系，先删除子表数据，然后再删除父表中的

以上就是 DML 了，接下来看 TCL

大概有几种：commit、rollback 啥的，继续看吧。。。

ACID

A 原子性

C 一致性

I 隔离性 ☆☆

D持久性

然后就是在讲脏读，是指一个事务读到了另一个事务还没有提交的数据。



oracle默认隔离级别就不支持脏读

	转账事务	取款事务
1		开始事务
2	开始事务	
3		查询账户余额为2000元
4		取款1000元，余额被更改为1000元
5	查询账户余额为1000元 (产生脏读)	
6		取款操作发生未知错误，事务回滚，余额变更为2000元
7	转入2000元，余额被更改为3000元 (脏读的1000+2000)	
8	提交事务	
备注	按照正确逻辑，此时账户余额应该为4000元	

有例子好像就好理解一些 😊

不可重复读是指前后多次读取，造成数据不一致

	事务A	事务B
1	开始事务	
2	第一次查询, 小明的年龄为20岁	
3		开始事务
4	其他操作	
5		更改小明的年龄为30岁
6		提交事务
7	第二次查询, 小明的年龄为30岁	
备注	按照正确逻辑, 事务A前后两次读取到的数据应该一致	

幻读, 前后多次读取, 前后数据总量不一致

	事务A	事务B
1	开始事务	
2	第一次查询, 数据总量为100条	
3		开始事务
4	其他操作	
5		新增100条数据
6		提交事务
7	第二次查询, 数据总量为200条	
备注	按照正确逻辑, 事务A前后两次读取到的数据总量应该一致	

不可重复读和幻读到底有什么区别呢?

(1) 不可重复读是读取了其他事务更改的数据, 针对insert与update操作

解决: 使用行级锁, 锁定该行, 事务A多次读取操作完成后才释放该锁, 这个时候才允许其他事务更改刚才的数据。

(2) 幻读是读取了其他事务新增的数据, 针对insert与delete操作


解决: 使用表级锁, 锁定整张表, 事务A多次读取数据总量之后才释放该锁, 这个时候才

允许其他事务新增数据。

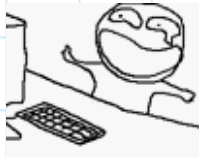


你又懂了?



饿了 好像饿了 想吃肉 又不能吃肉 只能吃菜
和 

感觉这种记笔记的方式似乎很有趣，哈哈哈



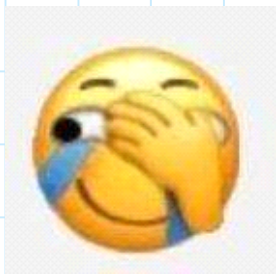
还不下课啊，到时间了吧应该。。。

A 岛上看到一个很有意思的串，哈哈啊哈。

```
insert into student (id, name)
values(2,'jack');
insert into student (id, name)
values(3,'jack3');
```

如果还没有提交commit的话，打开另一个窗口，你查一下发现这两数据还没有被插进去，这就是ORACLE帮你解决了脏读的问题。

事务可以包含多个DML操作，但是DDL、DCL只包含一个，而且exit的时候，前面的操作也会帮你自动提交。



小谢尔顿第三季哈哈

啊，对了，刀剑神域也没看呢。。。



rollback把未提交的事务的操作全部回退，然后提交✓

啥叫个回滚点呢

Savepoint xxx

Rollback xxx

就是和游戏里的存档差不多。

Lesson 10

修改表的操作，要学ALTER啦哈哈😄



好像没钱了，该充钱了今天哦哦o_o

Alter table stu

```
Add(  
    xxxxxxxxx  
)
```

```
Alter table stu  
Modify(  
    xxxxxx  
)
```

```
Alter table stu  
Drop column xxx;
```

```
Alter table stu  
Add constraint stu_xxx_PK primary key(xxx);
```

```
Alter table stu  
Modify(  
    Xxx xxx xxx not null  
)
```

```
Alter table xxx  
Drop constraint xxx
```

```
Alter table sss  
Drop primary key(aaadD) CASCADE
```

CASCADE:删除父表中的主键约束，级联的删除子表中的外键约束。

```
Alter table ccc
```

```
Disable constraint xxx_xxx-pk
```

```
Alter table ccc
```

```
enable constraint xxx_xxx-pk
```

```
Drop table xxx cascade constraint
```

cascade constraint级联地将子表中的约束给删除



困到沉默

重命名表

```
RENAME stu to stuu
```

重名列

```
Alter table stu
```

rename column id to idcard

TRUNCATE TABLE stu

Delete from stu;

都可以删除表中数据，但truncate属于DDL，对于事务来说会自动提交事务，不能回滚，truncate会释放表空间；而delete是DML，不会自动提交，可以删除局部数据，可以回滚，不能释放表空间。

评论，comment。。。

Comment on table stu is 'something'

Desc user_tab_comments 查看注释

Select comments

From user_tab_comments

Where lower(table_name)='stu';

给列加注释：

Comment on column stu.id is 'other thing'

查看注释：

Desc user_col_comments


```
Select colimn_name, comments  
From user_col_comments  
Where lower(table_name)='stu' ;
```