**实验报告四**

|  |  |
| --- | --- |
| 实验序号：4 | 实验名称：表的删除与修改 |
| 学号：2023015509 | 实验日期：2025年5月15日 |
| 姓名：胡林森 | 实验教室：C4 420机房 |

**一、实验目的**

1．掌握基本表的删除与修改；

2．掌握实体完整性、参照完整性和用户定义的完整性的定义、检查和违约处理；

3．掌握视图的定义、查询和更新，了解视图的作用。

**二、实验学时**

2学时

**三、实验内容**

1．利用Query Analyzer完成以下操作：

⑴ 向在实验三中所定义的数据表增加“备注”列，其数据类型为字符型，并查看新增列的值。

⑵ 对上述数据表增加主码约束条件，并观察在数据表中存在数据的情况下主码约束是否创建成功，然后再次执行实验三中实验内容3.2的操作，并观察记录执行结果。

⑶ 删除上述数据表中的数据，然后再删除该数据表，对这两个操作进行比较。

2．利用Query Analyzer完成以下任务：

⑴ 对实验二中所定义的6个数据表增加主码约束条件，并观察在数据表中存在数据的情况下主码约束是否创建成功，然后执行以下2个操作，观察并记录实体完整性的检查和违约处理。

① into 材料费表 ,('zy2018001','wm004',100,10)

② into 材料费表 ,('zy2018002',NULL,200,10)

注：“材料费表”根据自己所命名的表名进行相应的替换。

⑵ 对实验二中所定义的6个数据表增加相应的参照完整性约束，并观察在数据表中存在数据的情况下参照完整性约束是否创建成功，然后执行以下操作，观察并记录参照完整性的检查和违约处理。

① 将（y007 油井 112203002）插入到油水井表。

② into 材料费表 ,('zy2018007','wm006',100,10)

③ 将作业项目编号zy2018001的施工单位修改为“作业公司作业五队”。

④ 将单位代码表中的（112202002 采油二矿二队）删除，查看油水井表和作业项目表中的数据有何变化。

⑤ 将物码表中的（wm004 材料四 袋）修改为（wm04 材料四 袋）。

⑥ 撤销上述成功的更新操作。

注：“材料费表”根据自己所命名的表名进行相应的替换。

⑶ 对实验二中所定义的6个数据表按以下要求增加相应的完整性约束条件，并观察在数据表中存在数据的情况下完整性约束是否创建成功。

① 单位代码表的单位名称不能取空值、且取值唯一。

② 油水井表的井别只允许取“油井”或“水井”，单位代码不能取空值。

③ 物码表的名称规格不能取空值、且取值唯一，计量单位不能取空值。

④ 材料费表的消耗数量不能取空值，单价不能取空值。

⑤ 对作业项目表根据实际应用的要求定义适当的用户定义的完整性约束条件。

3．利用Query Analyzer完成以下操作：

⑴ 定义一个视图，用于保存作业项目表和材料费表的全部列。

⑵ 查询上面定义的视图，可任意组合查询条件，构造出2个查询。

⑶ 定义一个反映作业项目预算状态的视图，并向该视图插入('zy2018008','112202002','y005',10000,'张三', '2018-07-02')，查看作业项目表的数据有何变化。

⑷ 撤销上述成功的更新操作。

**四、实验报告**

目 录

[一、实验步骤 5](#_Toc199842130)

[（一）实验三中添加备注、完整性约束、删除 5](#_Toc199842131)

[（二）实验二中完整性约束 6](#_Toc199842132)

[（三）对实验三所定义表数据插入测试 6](#_Toc199842133)

[（四）对实验二所定义表数据插入测试 7](#_Toc199842134)

[（五）项目材料明细视图创建与操作 8](#_Toc199842135)

[1.创建项目材料明细视图 8](#_Toc199842136)

[2. 视图操作： 9](#_Toc199842137)

[（六）预算状态视图创建与操作 10](#_Toc199842138)

[1.创建视图 10](#_Toc199842139)

[2.插入数据 10](#_Toc199842140)

[二、实验总结 11](#_Toc199842141)

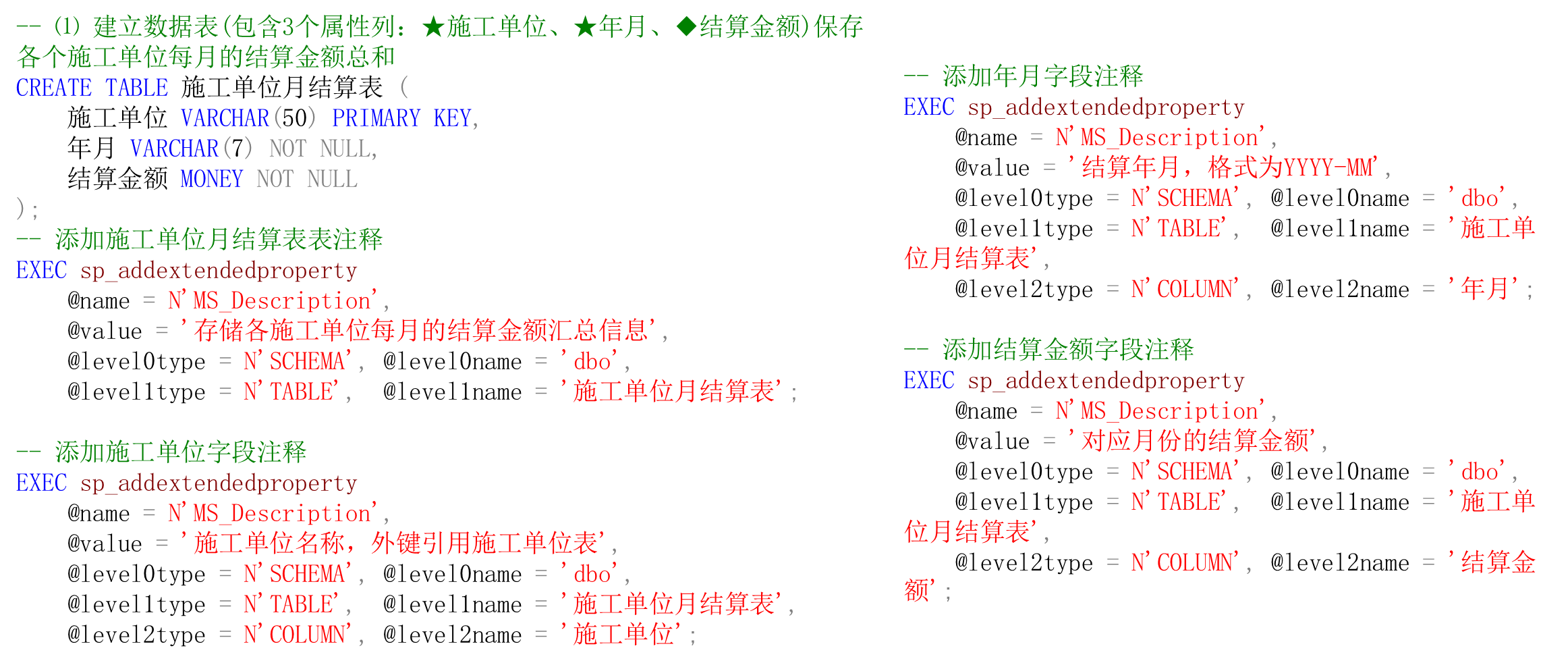
[（一）实验收获 11](#_Toc199842142)

[（二）改进建议 12](#_Toc199842143)

[（三）未来学习方向 12](#_Toc199842144)

# 一、实验步骤

## （一）实验三中添加备注、完整性约束、删除



删除表中的数据：

使用DELETE语句删除所有数据（逐行删除，可以回滚）：

DELETE FROM 施工单位月结算表;

使用TRUNCATE TABLE语句删除所有数据（高效删除，通常不能回滚）：

TRUNCATE TABLE施工单位月结算表;

删除数据表：

DROP TABLE施工单位月结算表;

删除数据表中的数据和删除数据表是两个完全不同的操作。前者是针对表内数据的操作，而后者是针对表本身的操作，对数据库的影响程度和适用场景差异很大。如果只是想清空表中的数据但保留表结构，可以选择 DELETE 或 TRUNCATE；如果是要彻底移除表及其相关的一切定义和数据，则需要使用DROP TABLE。

删除数据操作不会影响表的依赖对象，如基于该表的视图、存储过程等。只要表的结构存在，这些依赖对象就可以继续使用，只是在访问数据时会发现数据为空。

删除表会对依赖于该表的其他数据库对象产生重大影响。例如，如果存在基于施工单位月结算表的视图、存储过程、函数等，这些对象可能会因为失去了所依赖的表而变得不可用或者需要重新定义。

## （二）实验二中完整性约束



## （三）对实验三所定义表数据插入测试

执行以下2个操作，观察并记录实体完整性的检查和违约处理。

① into 材料明细表 ,('zy2018001','wm004',100,10)

② into 材料明细表 ,('zy2018002',NULL,200,10)



对于操作①：可执行。

对于操作②：出现错误，违反完整性约束。

## （四）对实验二所定义表数据插入测试

① 将（y007 油井 112203002）插入到油水井表。

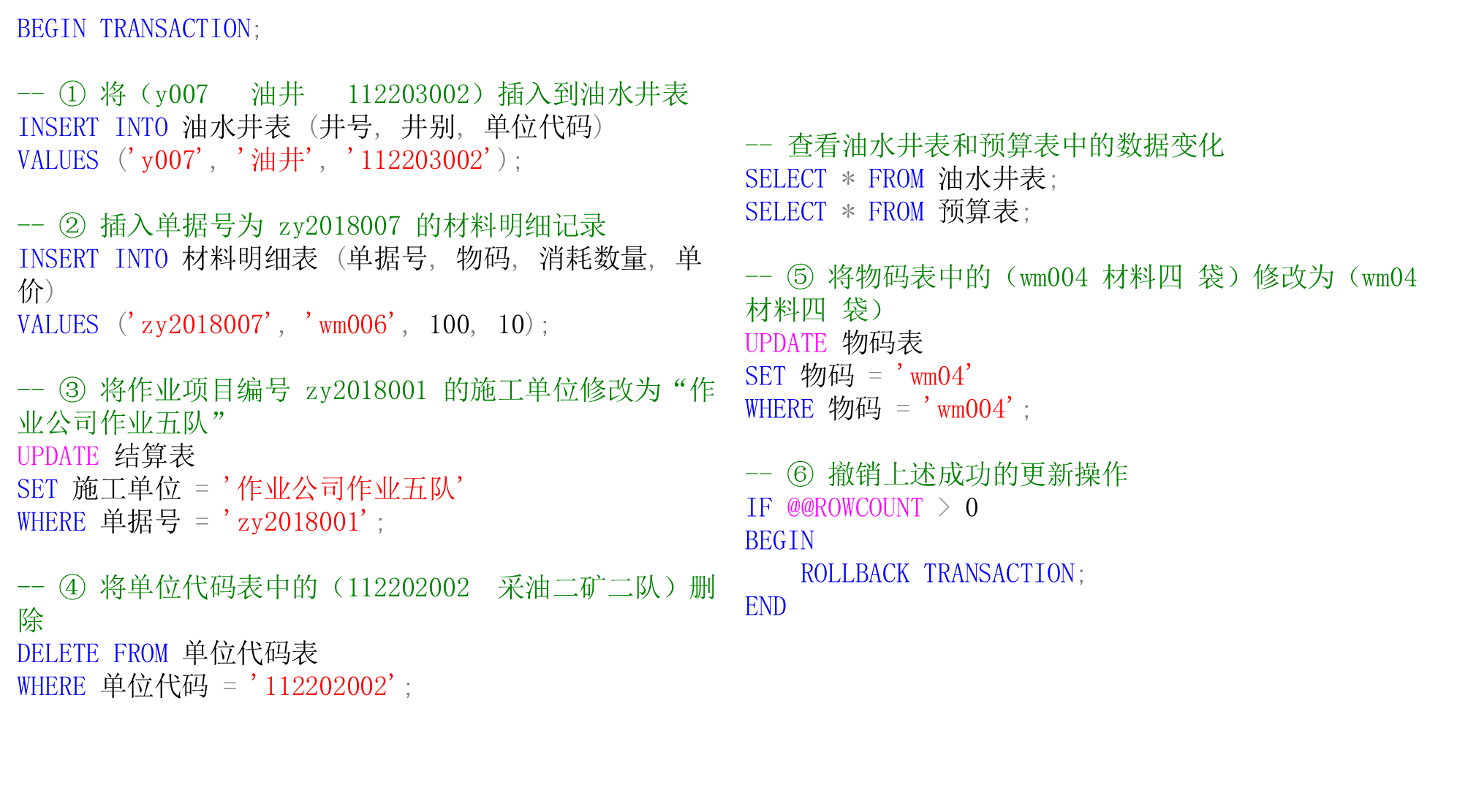
② into 材料费表 ,('zy2018007','wm006',100,10)

③ 将作业项目编号zy2018001的施工单位修改为“作业公司作业五队”。

④ 将单位代码表中的（112202002 采油二矿二队）删除，查看油水井表和作业项目表中的数据有何变化。

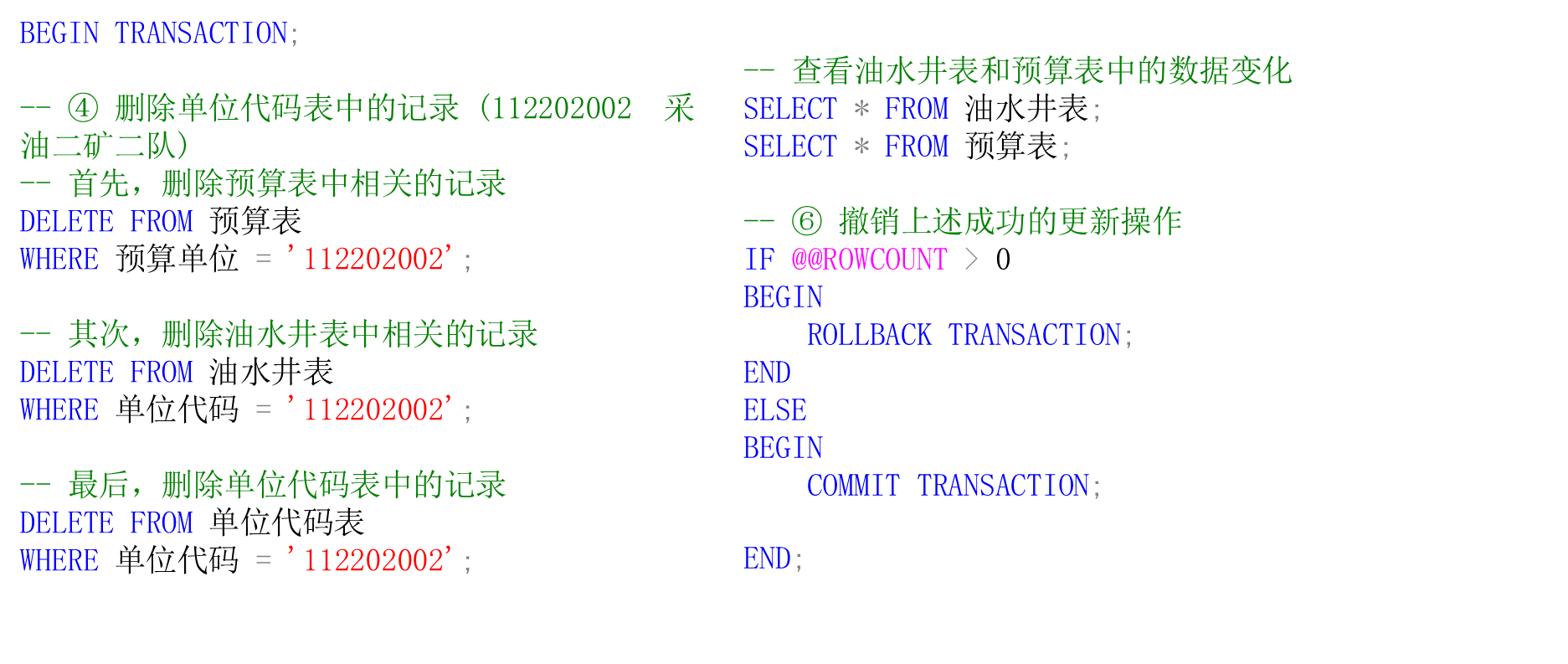
⑤ 将物码表中的（wm004 材料四 袋）修改为（wm04 材料四 袋）。

⑥ 撤销上述成功的更新操作



对于操作④，由于外键约束，对于单位代码表中的（112202002 采油二矿二队）删除执行失败

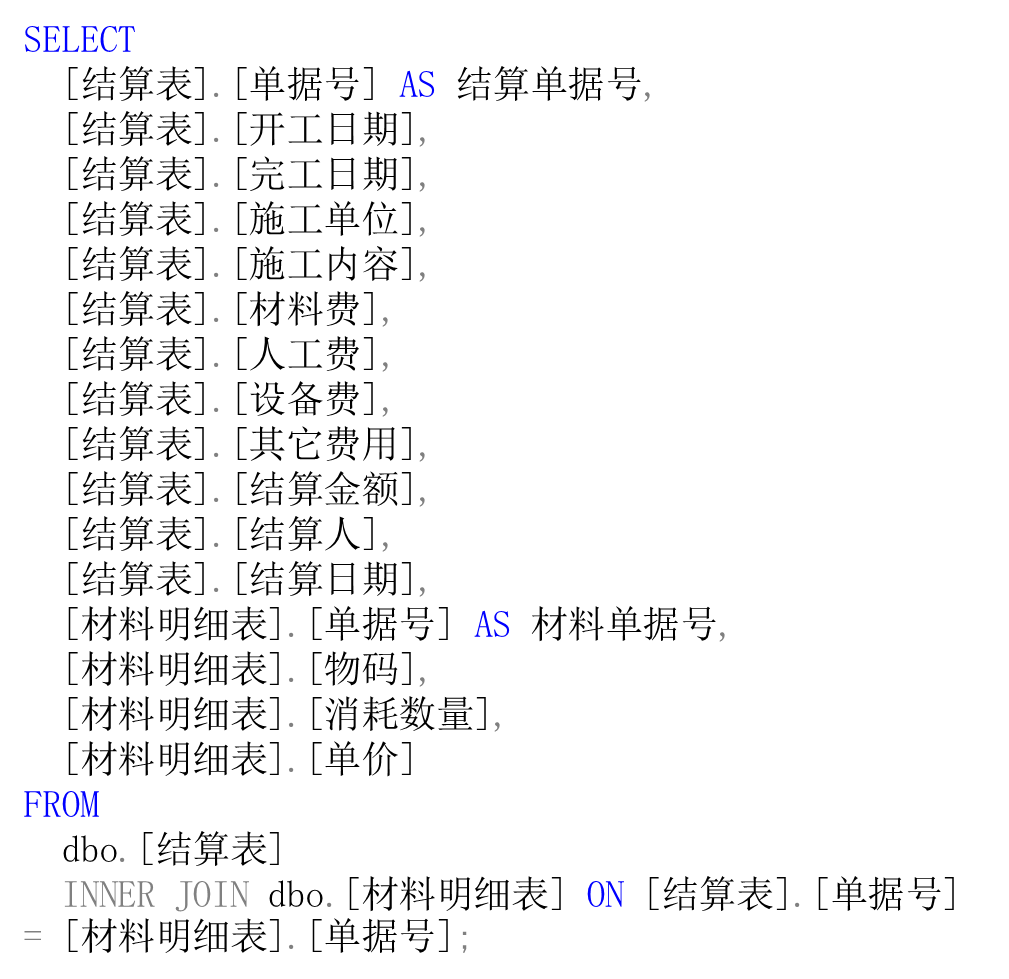
正确操作应该：



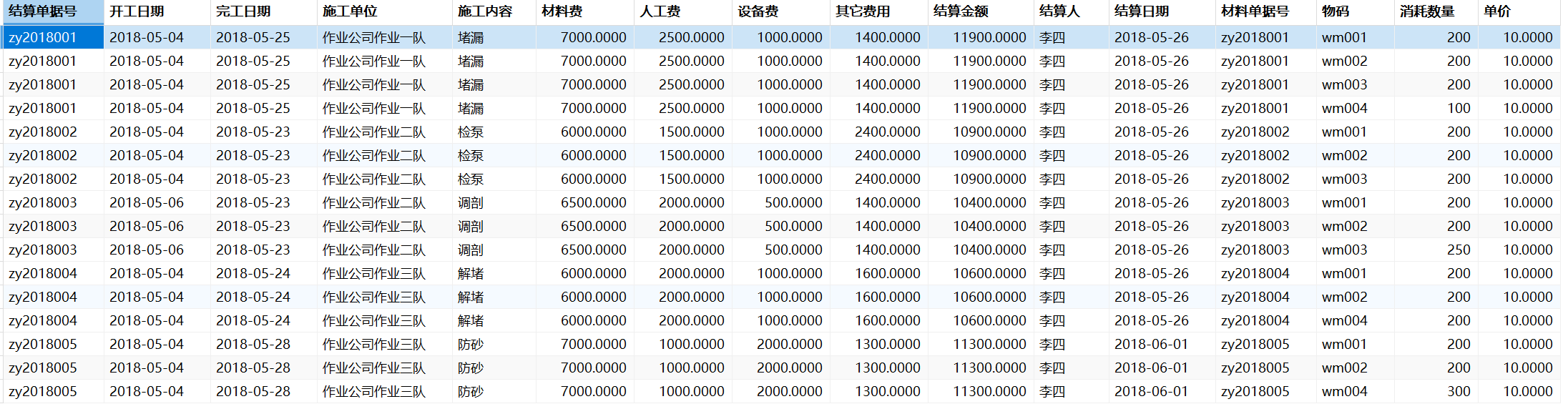
## （五）项目材料明细视图创建与操作

### 1.创建项目材料明细视图

由于计算表与材料明细表中均存在单据号，所以取别名。

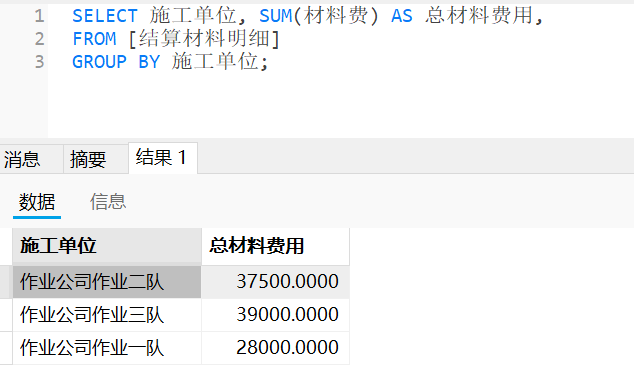


视图明细：



### 2. 视图操作：

(1)查询统计每个施工单位的材料费用总和：

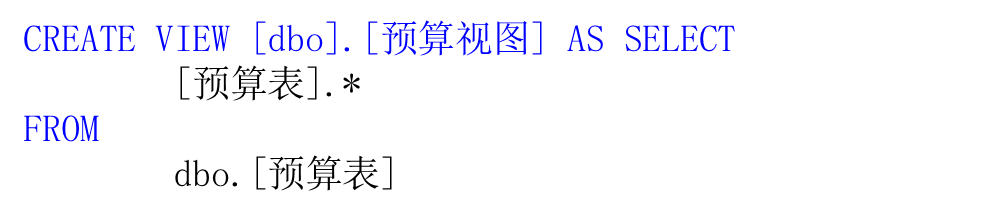


(2)查询特定施工单位的相关信息



## （六）预算状态视图创建与操作

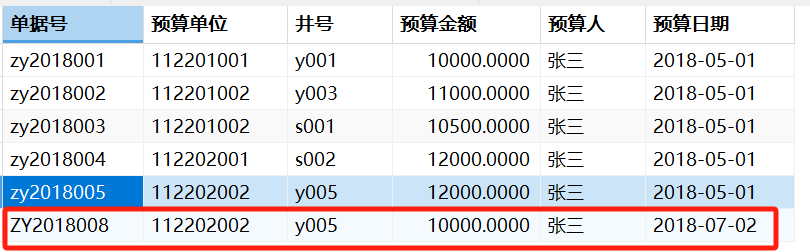
### 1.创建视图





### 2.插入数据

该视图插入('ZY2018008','112202002','y005',10000,'张三', '2018-07-02')；首先视图中数据新增一项。



原基本表中也新增一项。



# 二、实验总结

## （一）实验收获

通过本次实验，深入掌握了数据库表的删除与修改操作，熟练运用ALTER TABLE语句实现列的添加，深刻理解了主码约束在数据唯一性和完整性保障中的核心作用。在完整性约束实践中，清晰区分了实体完整性（主码非空唯一）、参照完整性（外键关联约束）和用户定义完整性（自定义CHECK规则），并通过具体操作直观观察到违反约束时的系统反馈，如插入空值或无效外键时的错误提示，强化了对数据库完整性规则的理解与应用能力。

视图操作部分，成功掌握了视图的创建、查询及更新机制。通过创建包含多表字段的视图，体会到视图在简化复杂查询、封装业务逻辑方面的优势；同时发现视图更新受限于底层表的结构，例如基于多表连接的视图通常不支持更新，这加深了对视图特性的认知。此外，明确了DELETE、TRUNCATE与DROP的本质区别：前两者针对数据操作，后者彻底删除表结构，且TRUNCATE不可回滚的特性使其在清空大数据量表时更高效。

## （二）改进建议

实验过程中发现，当表中已有数据违反约束条件时，直接添加会失败在处理参照完整性问题（如删除父表数据导致外键冲突）时，当前需手动先删子表记录再删父表记录，操作繁琐。可探索数据库级联操作（如ON DELETE CASCADE），简化外键关联数据的删除流程，增强数据操作的便捷性。

## （三）未来学习方向

基于本次实验对完整性约束和视图的基础认知，未来计划深入学习数据库高级特性，如存储过程、触发器和索引优化。存储过程可封装复杂业务逻辑，提升数据操作效率；触发器能自动响应数据变更事件，实现数据一致性的自动维护；索引优化则可显著提升大数据量下的查询性能，这些都是数据库开发与管理的核心技能。

进一步探索视图在数据权限控制中的应用，例如通过创建基于特定条件的视图，结合用户权限管理，实现细粒度的数据访问控制，这对多用户环境下的数据库安全至关重要。此外，计划学习数据库设计范式（如第三范式、BC 范式），优化表结构以减少数据冗余，提升数据库设计的规范性和可维护性，为构建大型复杂数据库系统奠定基础。