（1）PRIMARY KEY 主键约束的建立

1) 建立表时加主键约束

CREATE TABLE ST(

SNO VARCHAR (10) PRIMARY KEY ,

SNAME VARCHAR(20) NOT NULL,

DEPT VARCHAR(20),

AGE INT,

SEX CHAR(4))

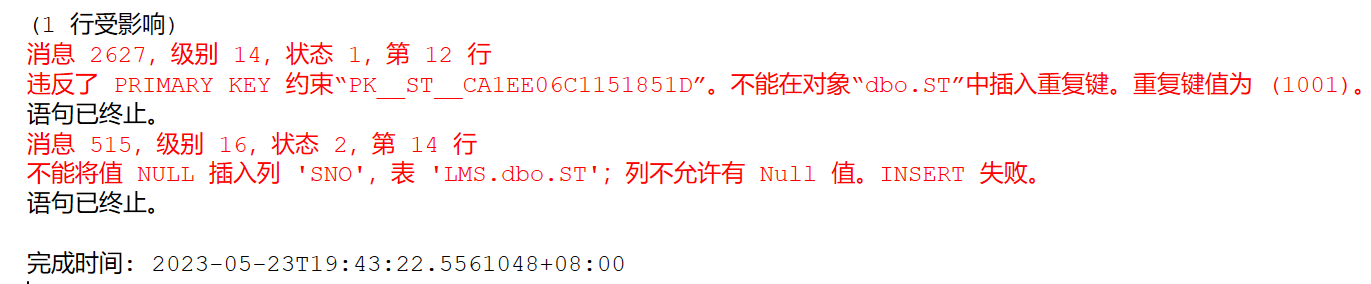
GO

INSERT INTO ST (SNO,SNAME,DEPT) VALUES('1001','ZXX',NULL)

GO

INSERT INTO ST (SNO,SNAME) VALUES('1001','MID')

GO

INSERT INTO ST (SNO,SNAME) VALUES(NULL,'ZXX')

2) 在已有的表上添加约束

CREATE TABLE ST2 (

SNO VARCHAR (10) NOT NULL ,

SNAME VARCHAR(20) NOT NULL,

DEPT VARCHAR(20),

AGE INT,

SEX CHAR(4))

GO

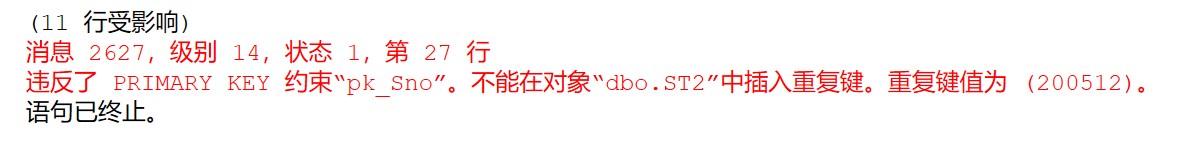
INSERT INTO ST2 SELECT \* FROM S

GO

ALTER TABLE ST2 ADD CONSTRAINT pk\_Sno PRIMARY KEY (SNO)

INSERT INTO ST2 (SNO,SNAME,DEPT) VALUES('200512','Beibei',NULL)

运行后出现什么结果？分析原因。



在执行 INSERT INTO 语句时违反了主键约束，因为插入的值已经存在于主键列中，造成了重复键值。

（2）DEFAULT 约束

CREATE TABLE CUST(

NO VARCHAR(5) PRIMARY KEY,

WEIGHT INT DEFAULT(10))

GO

INSERT INTO CUST(NO) VALUES('ZY01')

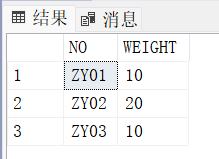
INSERT INTO CUST(NO) VALUES('ZY03')

INSERT INTO CUST VALUES('ZY02',20)

GO

SELECT \* FROM CUST

运行后出现什么结果？分析原因。



NO 列被定义为主键，并且 WEIGHT 列使用默认值为 10。

（3）CHECK 约束

CREATE TABLE CUSTOMER(CUSTNO CHAR(4) NOT NULL CHECK (CUSTNO LIKE

'[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]'), CUSTNAME VARCHAR(20))

GO

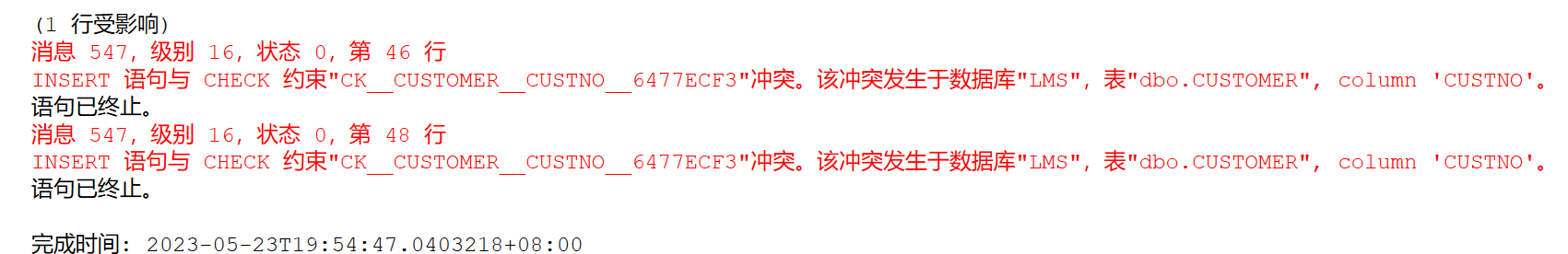
INSERT INTO CUSTOMER VALUES('BJ01','Beijing Grid Corp.')

GO

INSERT INTO CUSTOMER VALUES('BJ1','Beijing Grid Corp.')

GO

INSERT INTO CUSTOMER VALUES('BJ\*1','Beijing Grid Corp.')



运行后出现什么结果？分析原因。这里的[A-Z0-9]是代表啥意义？

这段代码创建了一个名为 "CUSTOMER" 的表，其中 CUSTNO 列被定义为 CHAR 类型，长度为 4，而且不能为空。此外，对 CUSTNO 列设置了一个 CHECK 约束，约束条件是它必须满足一个正则表达式模式，即以字母和数字组成的 4 位字符串。

（4）UNIQUE 唯一性约束的建立

CREATE TABLE CUST1(

CUSTNO CHAR(4)PRIMARY KEY,

CUSTNAME VARCHAR(20) UNIQUE,

COUNTRY VARCHAR(10))

GO

INSERT INTO CUST1(CUSTNO) VALUES('BJ01')

GO

INSERT INTO CUST1(CUSTNO) VALUES ('BJ02')

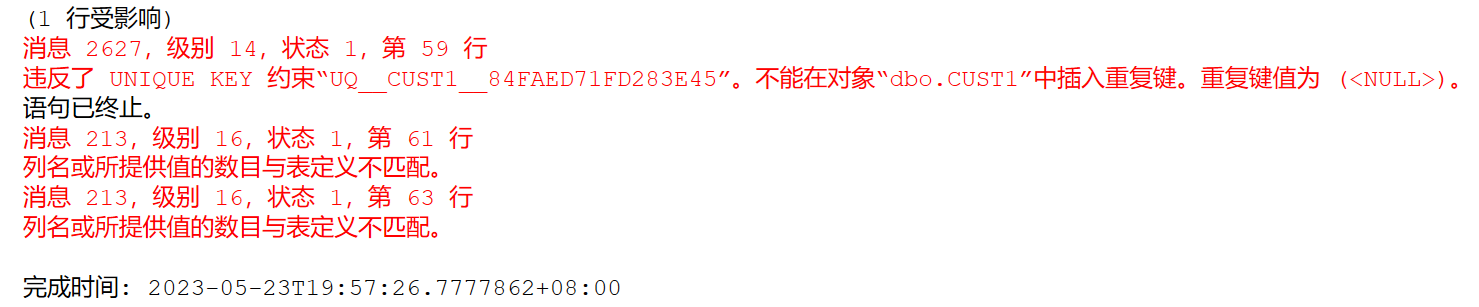
GO

INSERT INTO CUST1 VALUES ('BJ03','Beijing Grid Corp.')

GO

INSERT INTO CUST1 VALUES ('BJ04','Beijing Grid Corp.')

运行后出现什么结果？分析原因。



这个错误表示在插入数据时违反了名为 "UQ\_\_CUST1\_\_84FAED71FD283E45" 的唯一键约束。具体错误信息是重复键值为 <NULL>，即插入的数据在 CUSTNAME 列上与已有数据冲突。

在这行插入语句中，只指定了 CUSTNO 列的值，而 CUSTNAME 列的值未提供，因此默认为 NULL。由于 CUSTNAME 列被定义为唯一约束，因此 NULL 值会与已有的 NULL 值发生冲突，导致违反了唯一键约束。

（5）FOREIGN KEY 外键约束的建立

先创建如下数据表：

Create table S (SNO VARCHAR (10) NOT NULL ,SNAME VARCHAR(20),DEPT

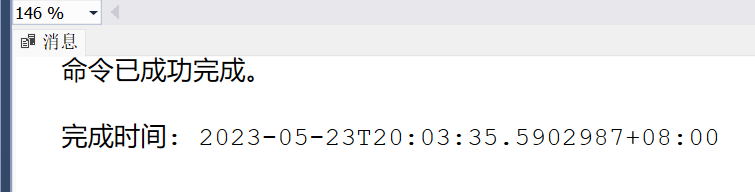
VARCHAR(20), AGE INT, SEX CHAR(4) )

Create table C (CNO VARCHAR(10) NOT NULL, CNAME VARCHAR(20),cpno

varchar(10),credit int)

Create table SC(SNO VARCHAR(10) NOT NULL, CNO VARCHAR(10) NOT NULL, SCORE

DECIMAL)



为了不影响原本的三个表，这里新开了一个数据库用来建立表。

然后执行如下语句：

ALTER TABLE C ADD CONSTRAINT CpriKEY PRIMARY KEY(CNO)

GO

ALTER TABLE SC ADD CONSTRAINT scpriKEY PRIMARY KEY(SNO,CNO)

ALTER TABLE SC ADD CONSTRAINT scforKEY1 FOREIGN KEY (SNO)

REFERENCES ST2(SNO)

ALTER TABLE SC ADD CONSTRAINT scforKEY2 FOREIGN KEY (CNO)

REFERENCES C(CNO)

GO

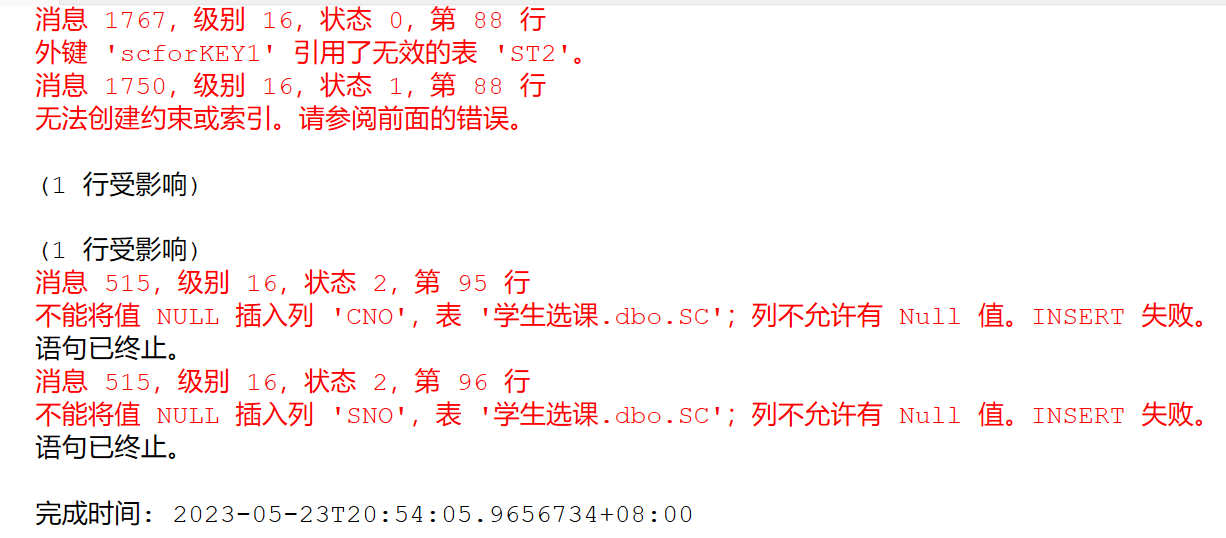
INSERT INTO SC (SNO,CNO) VALUES(‘890’,’C10’)

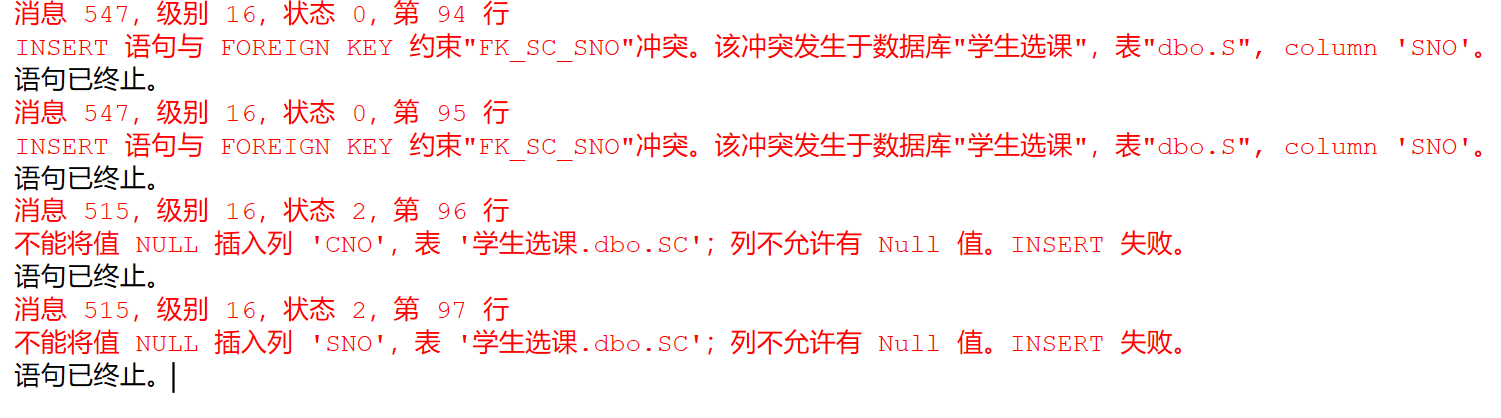
INSERT INTO SC (SNO,CNO) VALUES(‘200518’,’C01’)

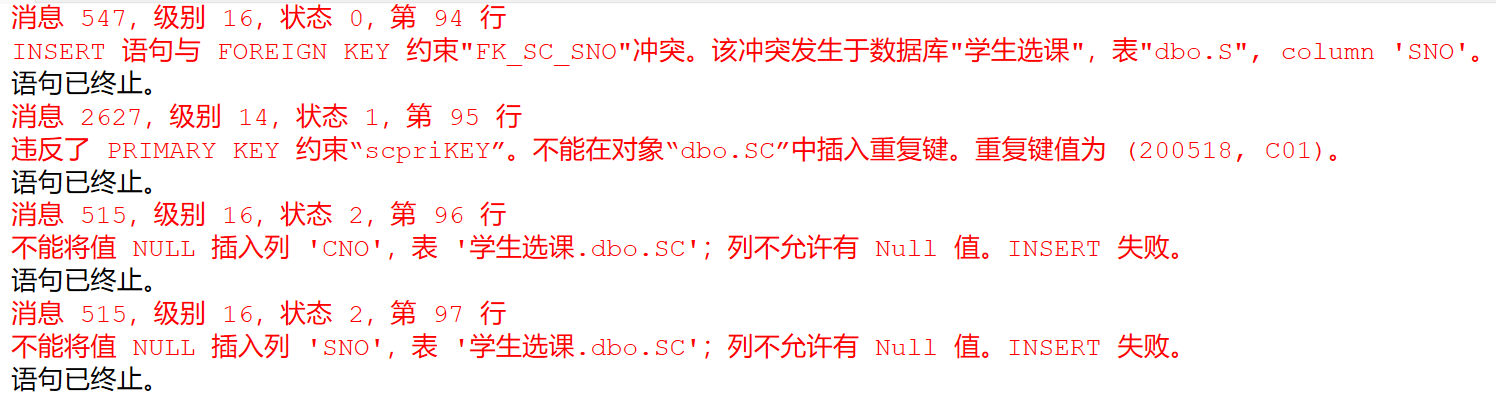
INSERT INTO SC (SNO,CNO) VALUES(‘890’,NULL)

INSERT INTO SC (SNO,CNO) VALUES(NULL,’C10’)

运行后出现什么结果？分析原因。



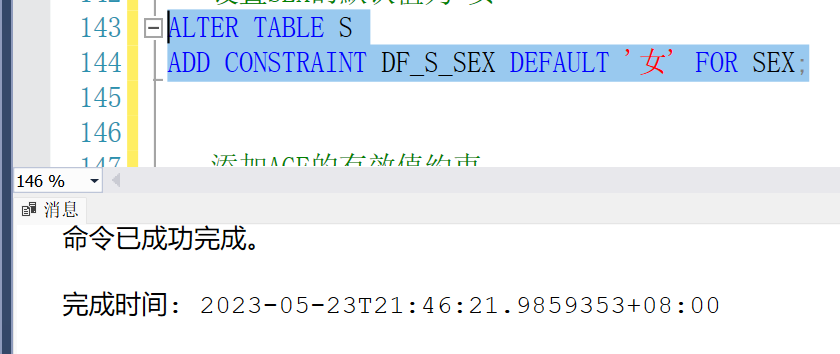


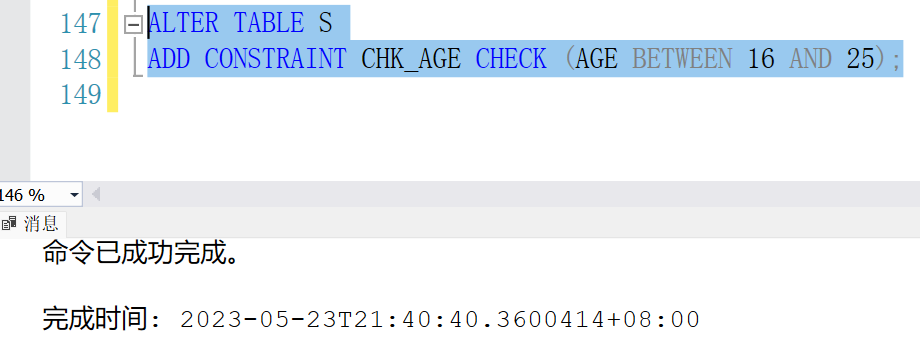


设计性实验

1) 在 S 表中添加完整性约束：SNO 设置为主键，SEX 的却省值为’女’,AGE 的

有效值为 16-25。





2) 创建 1 张与 S 相同的表 S1，在创建的同时将 SNO 设置为主键，SEX 的却省值

为’女’,AGE 的有效值为 16-25 ，并将 S 表中的数据插入到 S1 中，插入不同

的记录来严整设置的完整性。

--创建S1表，并设置字段和约束:

CREATE TABLE S1 (

SNO VARCHAR(10) PRIMARY KEY,

SNAME VARCHAR(20) NOT NULL,

DEPT VARCHAR(20),

AGE INT CHECK (AGE >= 16 AND AGE <= 25),

SEX CHAR(4) DEFAULT '女'

);

--将S表中的数据插入到S1表中:

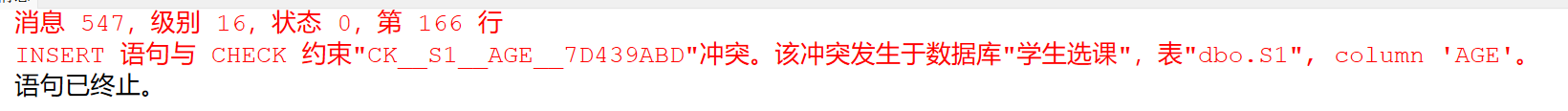
INSERT INTO S1 (SNO, SNAME, DEPT, AGE, SEX)

SELECT SNO, SNAME, DEPT, AGE, SEX FROM S;



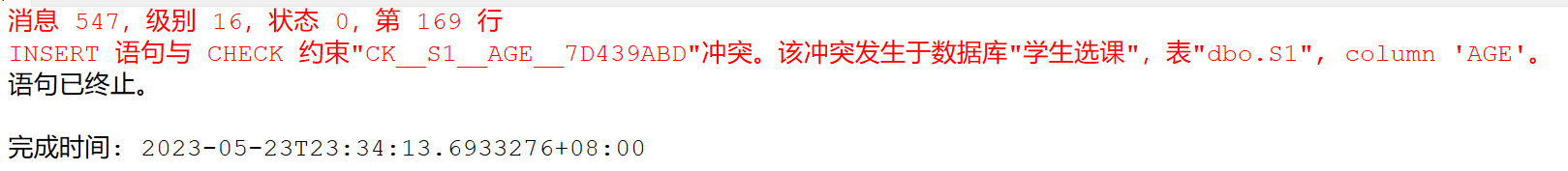
-- 插入一条年龄小于16的记录

INSERT INTO S1 (SNO, SNAME, DEPT, AGE, SEX) VALUES ('200999', '小明', '计算机系', 15, '男');



-- 插入一条年龄大于25的记录

INSERT INTO S1 (SNO, SNAME, DEPT, AGE, SEX) VALUES ('201000', '小红', '数学系', 30, '女');



3) 创建 1 张与 SC 相同的表 SC1，将（SNO，CNO）设置为主键，SNO 和 CNO

设置为外键，并将 SC 表中的数据复制到 SC1 中，插入不同的记录来严整设置的

完整性。

CREATE TABLE SC1 (

SNO VARCHAR(10) NOT NULL,

CNO VARCHAR(10) NOT NULL,

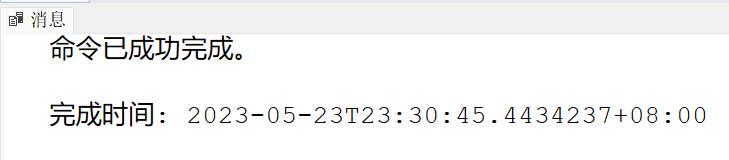
SCORE INT,

PRIMARY KEY (SNO, CNO),

FOREIGN KEY (SNO) REFERENCES S(SNO),

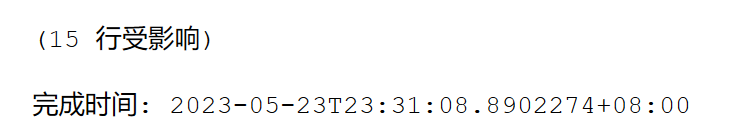
FOREIGN KEY (CNO) REFERENCES C(CNO)

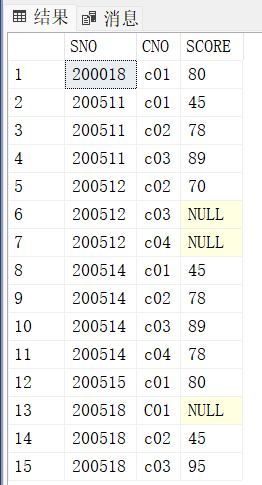
);



INSERT INTO SC1 (SNO, CNO, SCORE)

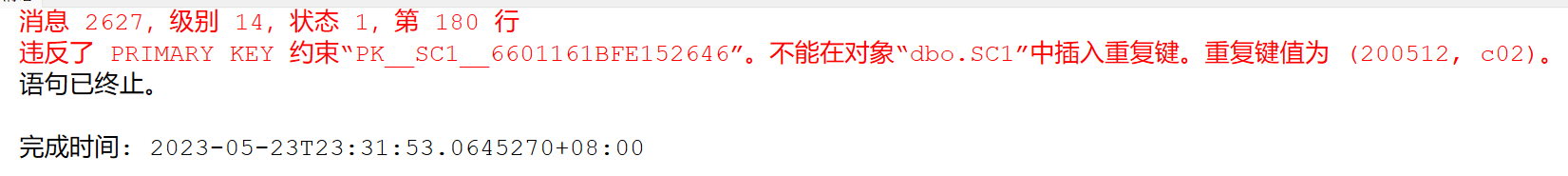
SELECT SNO, CNO, SCORE FROM SC;





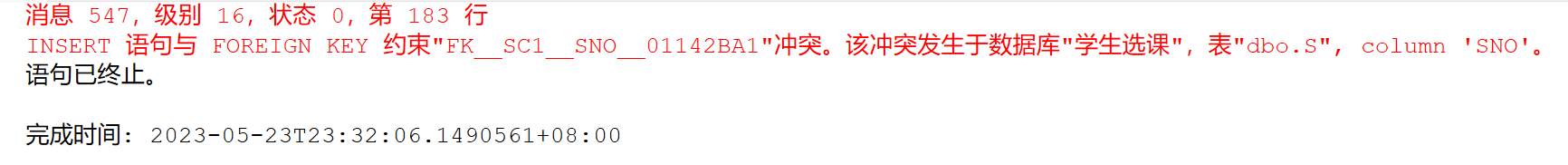
-- 插入一条已存在的记录

INSERT INTO SC1 (SNO, CNO, SCORE) VALUES ('200512', 'c02', 80);



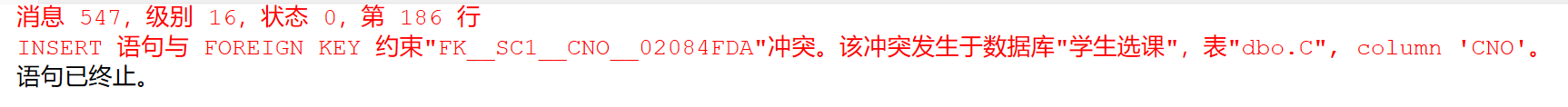
-- 插入一条不存在的学生编号记录

INSERT INTO SC1 (SNO, CNO, SCORE) VALUES ('200999', 'c01', 90);



-- 插入一条不存在的课程编号记录

INSERT INTO SC1 (SNO, CNO, SCORE) VALUES ('200512', 'c99', 85);



4) 创建 1 张与 C 表相同的表 C1，将 C 中数据插入到 C1 后，在 C1 上添加完整性

约束：将（CNO）设置为主键，CPNO 引用 CNO，CREDIT 值为 1-6。

SELECT \* INTO C1 FROM C WHERE 1=0;

INSERT INTO C1

SELECT \* FROM C;



-- 添加主键约束

ALTER TABLE C1

ADD CONSTRAINT PK\_C1 PRIMARY KEY (CNo);

-- 添加外键约束

ALTER TABLE C1

ADD CONSTRAINT FK\_C1\_CPNO\_C1\_CNO FOREIGN KEY (Cpno)

REFERENCES C1 (CNo);

-- 添加 CHECK 约束

ALTER TABLE C1

ADD CONSTRAINT CHK\_C1\_CREDIT CHECK (Credit >= 1 AND Credit <= 6);

