计算机科学与技术学院<u>数据库系统</u>课程实验报告

实验题目：综合实验 3		学号：202300130183
日期：2024/12/13	班级：23 级人工智能	姓名：宋浩宇
Email：2367651943@qq.com		

实验软件和硬件环境：

实验软件：

系统：Windows 11 家庭中文版 23H2 22631.4317

编辑器：Visual Studio Code

数据库：SQL Server 2022

数据库管理工具：Microsoft SQL Server Management Studio 18

硬件环境：

CPU：13th Gen Intel(R) Core(TM) i9-13980HX 2.20 GHz

内存：32.0 GB (31.6 GB 可用)

磁盘驱动器：NVMe WD_BLACKSN850X2000GB

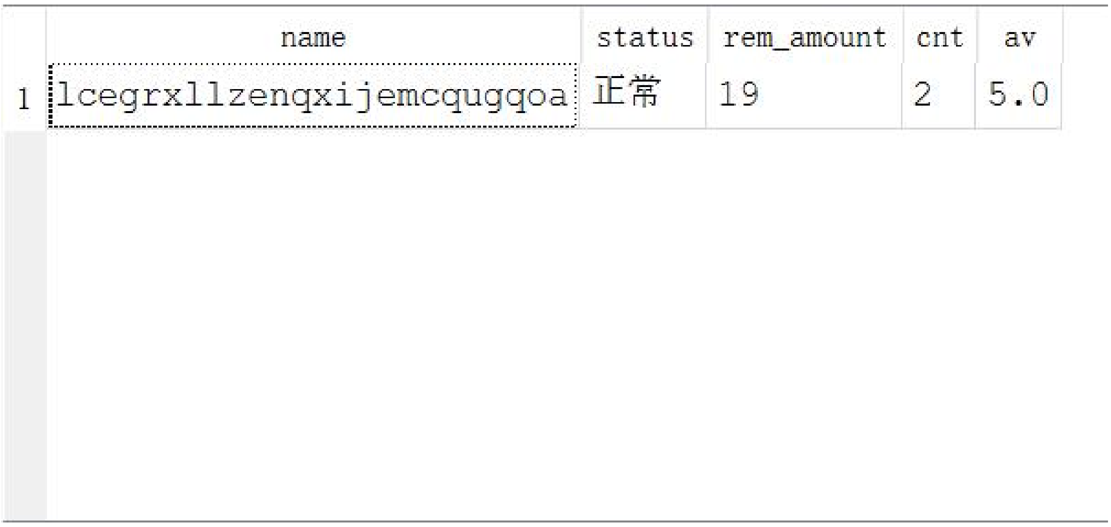
实验步骤：

第一题：

SQL：

```sql
SELECT
    COMMODITY.NAME,
    ONSELL.STATUS,
    ONSELL.REM_AMOUNT,
    CNT,
    AV
FROM
    ONSELL,
    (
        SELECT
            COUNT(*)       AS CNT,
            AVG(BUYERRATE) AS AV
        FROM
            ORDERS
        WHERE
            SID = 251
            AND CID = 669
    )
    JOIN COMMODITY
    ON ONSELL.CID = COMMODITY.CID
WHERE
    ONSELL.SID = 251
    AND ONSELL.CID = 669;
```

结果：



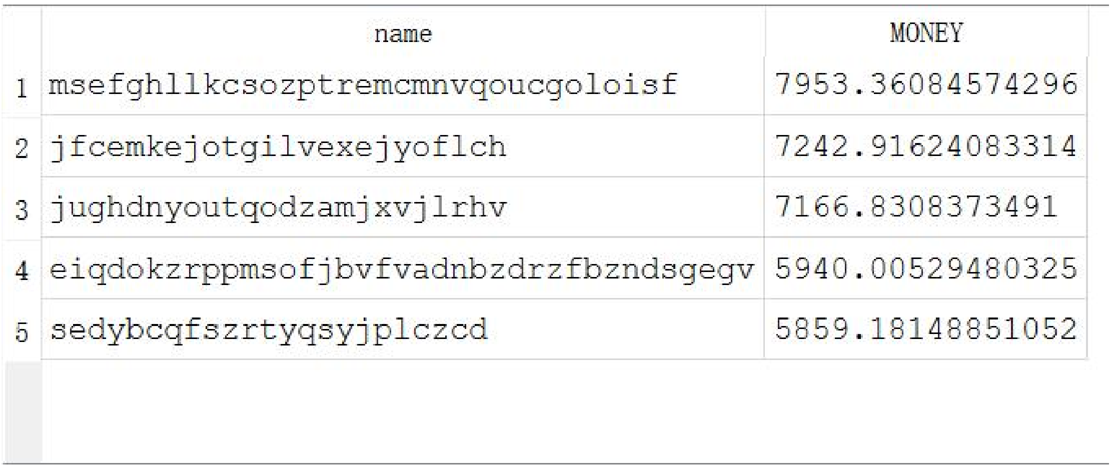
	name	status	rem_amount	cnt	av
1	lcegrxllzenqxijemcqugqoa	正常	19	2	5.0

第二题：

SQL：

```sql
SELECT
    STORES.NAME,
    SUM(ONSELL.PRICE) AS MONEY
FROM
    ORDERS,
    COMMODITY
    JOIN ONSELL
    ON ORDERS.SID = ONSELL.SID
    AND ONSELL.CID = ORDERS.CID
    JOIN BRAND
    ON COMMODITY.BID = BRAND.BID
    JOIN STORES
    ON ORDERS.SID = STORES.SID
WHERE
    COMMODITY.CID = ORDERS.CID
    AND BRAND.NAME = "蒂芙尼"
GROUP BY
    ORDERS.SID
ORDER BY
    SUM(ONSELL.PRICE) DESC LIMIT 5;
```

结果：



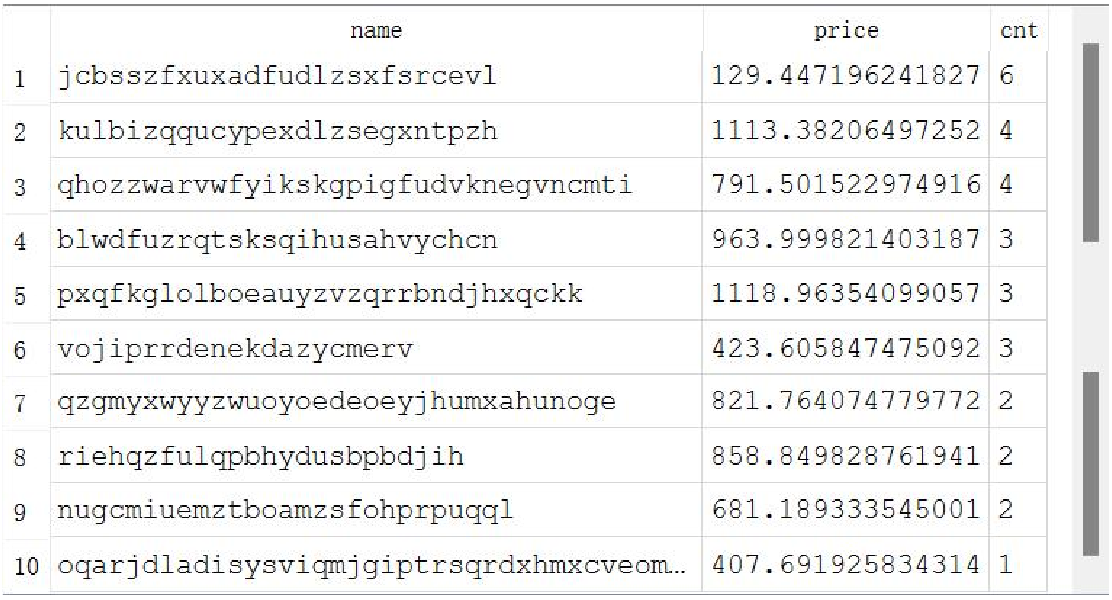
	name	MONEY
1	msefghllkcsozptremcmnvqoucgoloisf	7953.36084574296
2	jfcemkejotgilvexejyoflch	7242.91624083314
3	jughdnyoutqodzamjxvjlrhv	7166.8308373491
4	eiqdokzrppmsofjbvfvadnbzdrzfbzndsgegv	5940.00529480325
5	sedybcqfszrtyqsyjplczcd	5859.18148851052

第三题：

SQL：

```
SELECT
    COMMODITY.NAME,
    ONSELL.PRICE,
    COUNT(*) AS CNT
FROM
    STORES,
    ONSELL
    JOIN ORDERS
    ON STORES.SID = ORDERS.SID
    AND ONSELL.CID = ORDERS.CID
    JOIN COMMODITY
    ON ONSELL.CID = COMMODITY.CID
WHERE
    STORES.SID = 1
    AND ONSELL.SID = STORES.SID
GROUP BY
    ONSELL.CID
ORDER BY
    COUNT(*) DESC LIMIT 10;
```

结果：



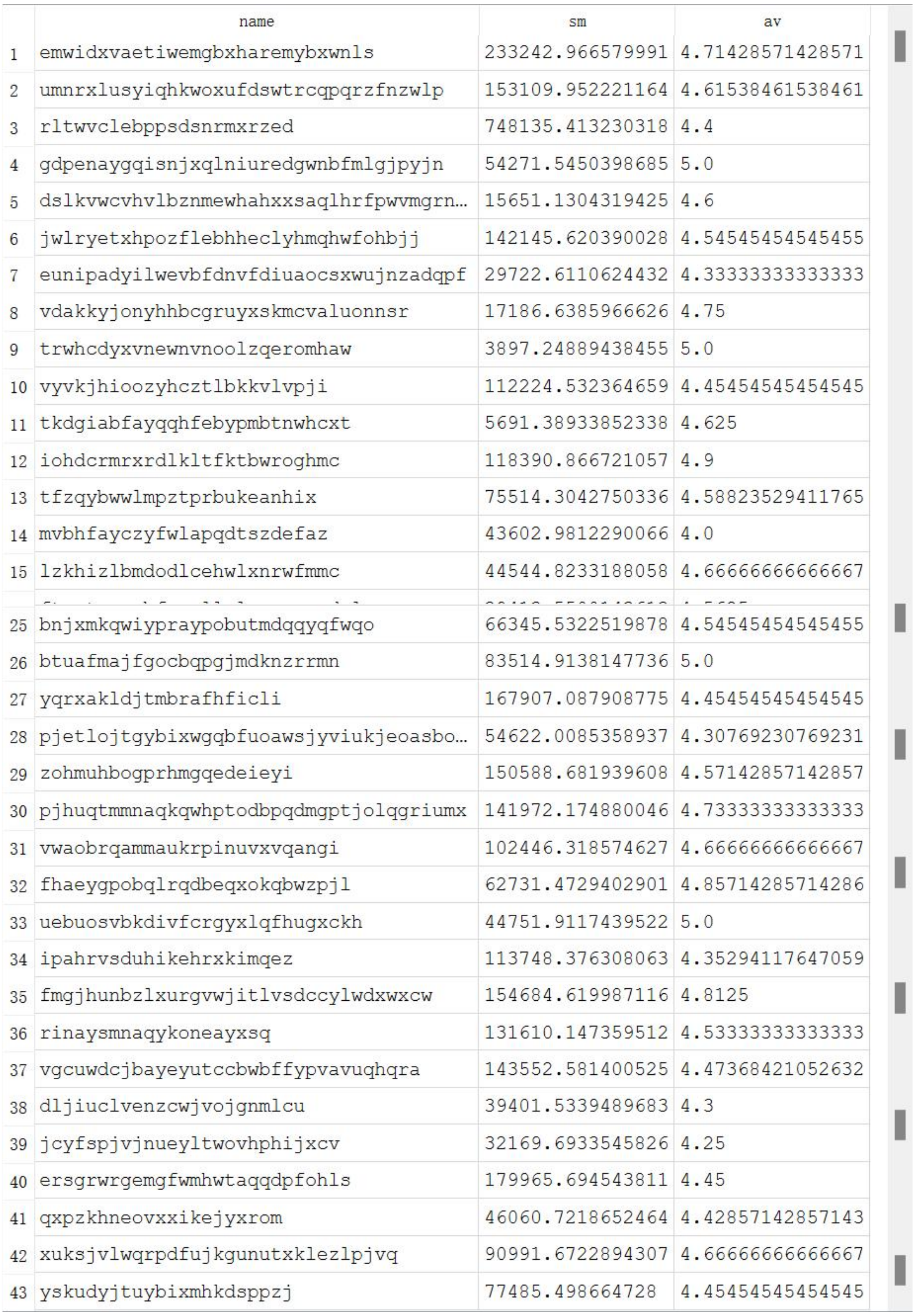
	name	price	cnt
1	jcbsszfxuxadfudlzsxfsrcevl	129.447196241827	6
2	kulbizqqucypexdlzsegxntpzh	1113.38206497252	4
3	qhozzwarvwfyikskgpigfudvknegvncmti	791.50152297 4916	4
4	blwdfuzrqtsksqihusahvychcn	963.999821403187	3
5	pxqfkglolboeauyzvzqrrbndjhxqckk	1118.96354099057	3
6	vojiprrdenekdazycmerv	423.605847475092	3
7	qzgmyxwyyzwuoyoedeoeyjhumxahunoge	821.764074779772	2
8	riehqzfulqpbhydusbpbdjih	858.849828761941	2
9	nugcmiuemztboamzsfohprpuqql	681.189333545001	2
10	oqarjdladisysviqmjgiptrsqrdxhmxcveom…	407.691925834314	1

第四题：

SQL：

```
SELECT
    COMMODITY.NAME,
    SUM(ONSELL.PRICE)     AS SM,
    AVG(ORDERS.BUYERRATE) AS AV
FROM
    BRAND,
    ONSELL
    JOIN COMMODITY
    ON BRAND.BID = COMMODITY.BID
    AND COMMODITY.CID = ONSELL.CID
    JOIN ORDERS
    ON ORDERS.CID = COMMODITY.CID
WHERE
    BRAND.BID = 1
    AND ORDERS.STATUS = "已完成"
GROUP BY
    ORDERS.CID;
```

结果：

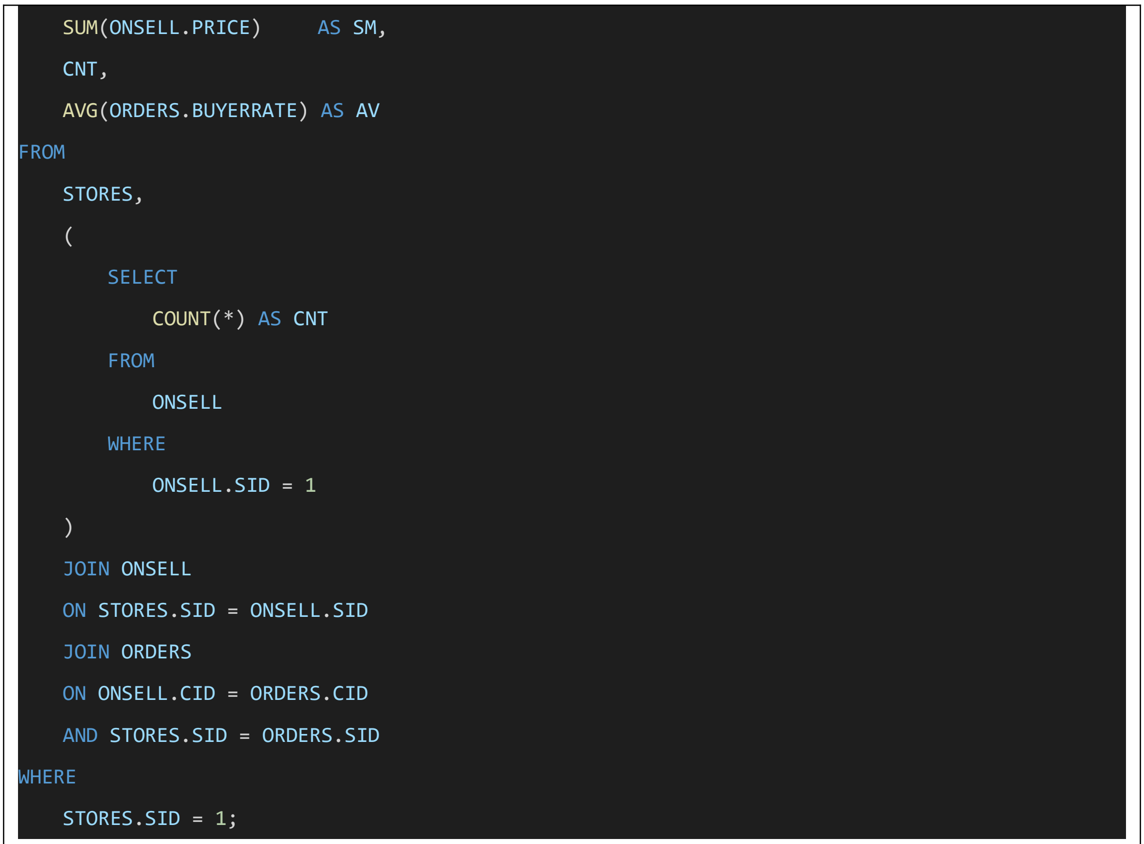


	name	sm	av
1	emwidxvaetiwemgbxharemybxwnls	233242.966579991	4.71428571428571
2	umnrxlusyiqhkwoxufdswtrcqpqrzfnzwlp	153109.952221164	4.61538461538461
3	rltwvclebppsdsnrmxrzed	748135.413230318	4.4
4	gdpenaygqisnjxqlniuredgwnbfmlgjpyjn	54271.5450398685	5.0
5	dslkvwcvhvlbznmewhahxxsaqlhrfpwvmgrn…	15651.1304319425	4.6
6	jwlryetxhpozflebhheclyhmqhwfohbjj	142145.620390028	4.54545454545455
7	eunipadyilwevbfdnvfdiuaocsxwujnzadqpf	29722.6110624432	4.33333333333333
8	vdakkyjonyhhbcgruyxskmcvaluonnsr	17186.6385966626	4.75
9	trwhcdyxvnewnvnoolzqeromhaw	3897.24889438455	5.0
10	vyvkjhioozyhcztlbkkvlvpji	112224.532364659	4.45454545454545
11	tkdgiabfayqqhfebypmbtnwhcxt	5691.38933852338	4.625
12	iohdcrmrxrdlkltfktbwroghmc	118390.866721057	4.9
13	tfzqybwwlmpztprbukeanhix	75514.3042750336	4.58823529411765
14	mvbhfayczyfwlapqdtszdefaz	43602.9812290066	4.0
15	lzkhizlbmdodlcehwlxnrwfmmc	44544.8233188058	4.66666666666667
25	bnjxmkqwiypraypobutmdqqyqfwqo	66345.5322519878	4.54545454545455
26	btuafmajfgocbqpgjmdknzrrmn	83514.9138147736	5.0
27	yqrxakldjtmbrafhficli	167907.087908775	4.45454545454545
28	pjetlojtgybixwgqbfuoawsjyviukjeoasbo…	54622.0085358937	4.30769230769231
29	zohmuhbogprhmgqedeieyi	150588.681939608	4.57142857142857
30	pjhuqtmmnaqkqwhptodbpqdmgptjolqgriumx	141972.174880046	4.73333333333333
31	vwaobrqammaukrpinuvxvqangi	102446.318574627	4.66666666666667
32	fhaeygpobqlrqdbeqxokqbwzpjl	62731.4729402901	4.85714285714286
33	uebuosvbkdivfcrgyxlqfhugxckh	44751.9117439522	5.0
34	ipahrvsduhikehrxkimqez	113748.376308063	4.35294117647059
35	fmgjhunbzlxurgvwjitlvsdccylwdxwxcw	154684.619987116	4.8125
36	rinaysmnaqykoneayxsq	131610.147359512	4.53333333333333
37	vgcuwdcjbayeyutccbwbffypvavuqhqra	143552.581400525	4.47368421052632
38	dljiuclvenzcwjvojgnmlcu	39401.5339489683	4.3
39	jcyfspjvjnueyltwovhphijxcv	32169.6933545826	4.25
40	ersgrwrgemgfwmhwtaqqdpfohls	179965.694543811	4.45
41	qxpzkhneovxxikejyxrom	46060.7218652464	4.42857142857143
42	xuksjvlwqrpdfujkgunutxklezlpjvq	90991.6722894307	4.66666666666667
43	yskudyjtuybixmhkdsppzj	77485.498664728	4.45454545454545

第五题：

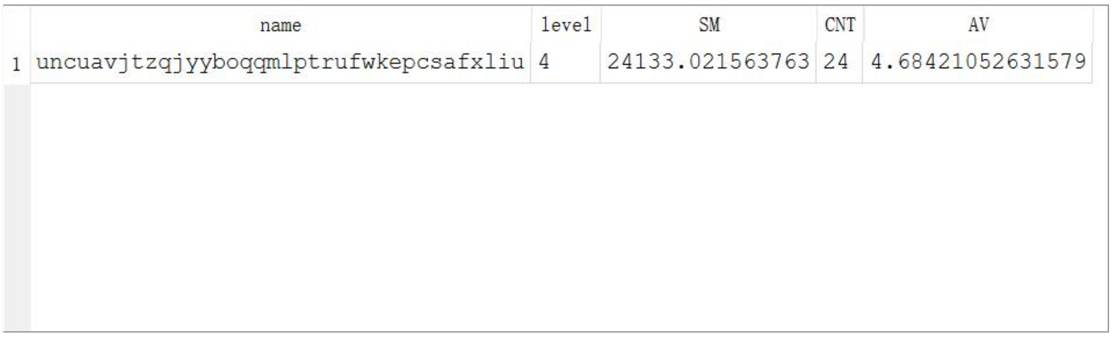
SQL：

```
SELECT
    STORES.NAME,
    STORES.LEVEL,
```



```
    SUM(ONSELL.PRICE)     AS SM,
    CNT,
    AVG(ORDERS.BUYERRATE) AS AV
FROM
    STORES,
    (
        SELECT
            COUNT(*) AS CNT
        FROM
            ONSELL
        WHERE
            ONSELL.SID = 1
    )
    JOIN ONSELL
    ON STORES.SID = ONSELL.SID
    JOIN ORDERS
    ON ONSELL.CID = ORDERS.CID
    AND STORES.SID = ORDERS.SID
WHERE
    STORES.SID = 1;
```

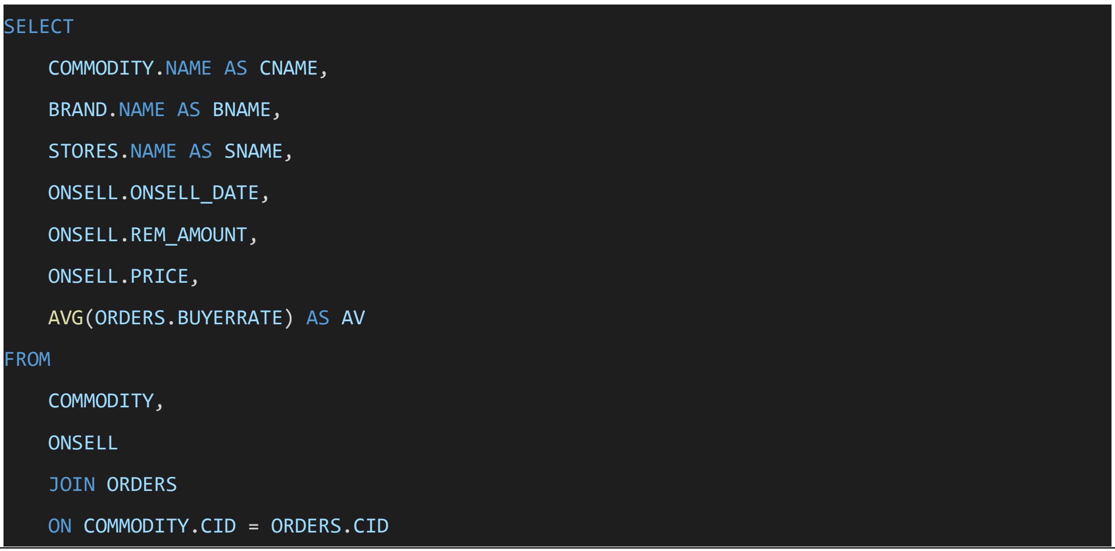
结果：



	name	level	SM	CNT	AV
1	uncuavjtzqjyyboqqmlptrufwkepcsafxliu	4	24133.021563763	24	4.68421052631579

第六题：

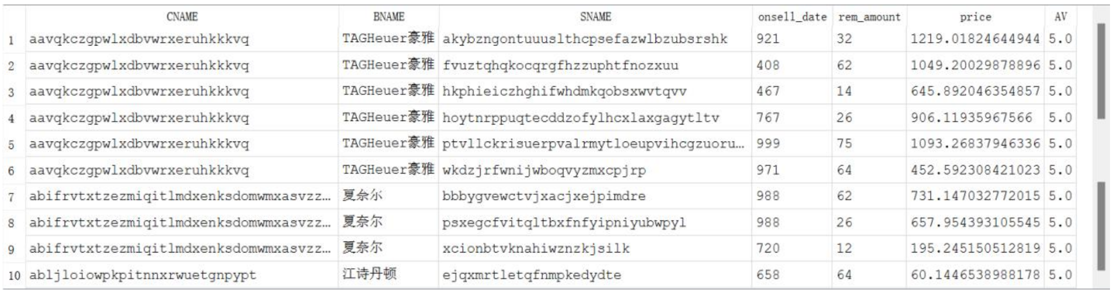
SQL：



```
SELECT
    COMMODITY.NAME AS CNAME,
    BRAND.NAME AS BNAME,
    STORES.NAME AS SNAME,
    ONSELL.ONSELL_DATE,
    ONSELL.REM_AMOUNT,
    ONSELL.PRICE,
    AVG(ORDERS.BUYERRATE) AS AV
FROM
    COMMODITY,
    ONSELL
    JOIN ORDERS
    ON COMMODITY.CID = ORDERS.CID
```

```sql
    AND ONSELL.SID = ORDERS.SID
    JOIN BRAND
    ON COMMODITY.BID = BRAND.BID
    JOIN STORES
    ON STORES.SID = ONSELL.SID
WHERE
    ONSELL.STATUS = "正常"
    AND COMMODITY.TYPE = "女装"
GROUP BY
    COMMODITY.NAME,
    STORES.NAME
ORDER BY
    AVG(ORDERS.BUYERRATE) DESC LIMIT 10;
```

结果：



	CNAME	BNAME	SNAME	onsell_date	rem_amount	price	AV
1	aavqkczgpwlxdbvwrxeruhkkkvq	TAGHeuer豪雅	akybzngontuuuslthcpsefazwlbzubsrshk	921	32	1219.01824644944	5.0
2	aavqkczgpwlxdbvwrxeruhkkkvq	TAGHeuer豪雅	fvuztqhqkocqrgfhzzuphtfnozxuu	408	62	1049.20029878896	5.0
3	aavqkczgpwlxdbvwrxeruhkkkvq	TAGHeuer豪雅	hkphieiczhghifwhdmkqobsxwvtqvv	467	14	645.892046354857	5.0
4	aavqkczgpwlxdbvwrxeruhkkkvq	TAGHeuer豪雅	hoytnrppuqtecddzofylhcxlaxgagytltv	767	26	906.11935967566	5.0
5	aavqkczgpwlxdbvwrxeruhkkkvq	TAGHeuer豪雅	ptvllckrisuerpvalrmytloeupvihcgzuoru…	999	75	1093.26837946336	5.0
6	aavqkczgpwlxdbvwrxeruhkkkvq	TAGHeuer豪雅	wkdzjrfwnijwboqvyzmxcpjrp	971	64	452.592308421023	5.0
7	abifrvtxtzezmiqitlmdxenksdomwmxasvzz…	夏奈尔	bbbygvewctvjxacjxejpimdre	988	62	731.147032772015	5.0
8	abifrvtxtzezmiqitlmdxenksdomwmxasvzz…	夏奈尔	psxegcfvitqltbxfnfyipniyubwpyl	988	26	657.954393105545	5.0
9	abifrvtxtzezmiqitlmdxenksdomwmxasvzz…	夏奈尔	xcionbtvknahiwznzkjsilk	720	12	195.245150512819	5.0
10	abljloiowpkpitnnxrwuetgnpypt	江诗丹顿	ejqxmrtletqfnmpkedydte	658	64	60.1446538988178	5.0

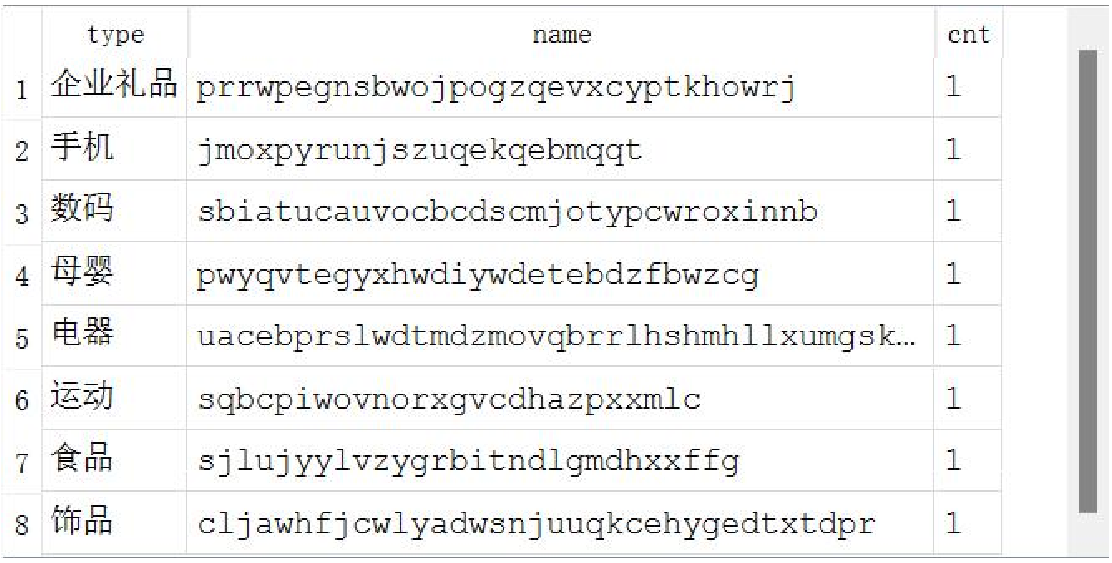
第七题：

SQL：

```sql
WITH RANKED AS (
    SELECT
        C.TYPE,
        C.NAME,
        COUNT(ORDERS.OID)                                            AS CNT,
        ROW_NUMBER() OVER (PARTITION BY C.TYPE ORDER BY COUNT(ORDERS.OID) DESC) AS RANK
    FROM
        ORDERS
        JOIN COMMODITY C
        ON ORDERS.CID = C.CID
    WHERE
        ORDERS.UID = 1
    GROUP BY
        C.TYPE,
        C.NAME
)
SELECT
    TYPE,
    NAME,
    CNT
FROM
```

```
    RANKED
WHERE
    RANK = 1;
```

结果：



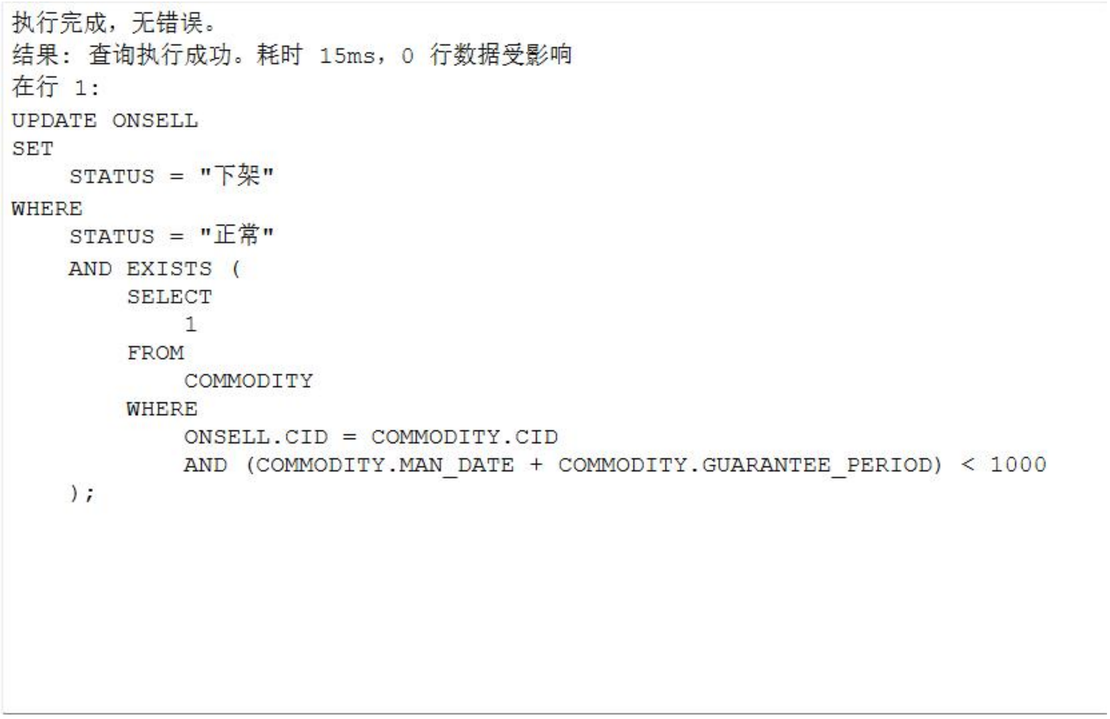
	type	name	cnt
1	企业礼品	prrwpegnsbwojpogzqevxcyptkhowrj	1
2	手机	jmoxpyrunjszuqekqebmqqt	1
3	数码	sbiatucauvocbcdscmjotypcwroxinnb	1
4	母婴	pwyqvtegyxhwdiywdetebdzfbwzcg	1
5	电器	uacebprslwdtmdzmovqbrrlhshmhllxumgsk…	1
6	运动	sqbcpiwovnorxgvcdhazpxxmlc	1
7	食品	sjlujyylvzygrbitndlgmdhxxffg	1
8	饰品	cljawhfjcwlyadwsnjuuqkcehygedtxtdpr	1

第八题：

SQL：

```
UPDATE ONSELL
SET
    STATUS = "下架"
WHERE
    STATUS = "正常"
    AND EXISTS (
        SELECT
            1
        FROM
            COMMODITY
        WHERE
            ONSELL.CID = COMMODITY.CID
            AND (COMMODITY.MAN_DATE + COMMODITY.GUARANTEE_PERIOD) < 1000
    );
```

结果：



```
执行完成，无错误。
结果：查询执行成功。耗时 15ms，0 行数据受影响
在行 1:
UPDATE ONSELL
SET
    STATUS = "下架"
WHERE
    STATUS = "正常"
    AND EXISTS (
        SELECT
            1
        FROM
            COMMODITY
        WHERE
            ONSELL.CID = COMMODITY.CID
            AND (COMMODITY.MAN_DATE + COMMODITY.GUARANTEE_PERIOD) < 1000
    );
```

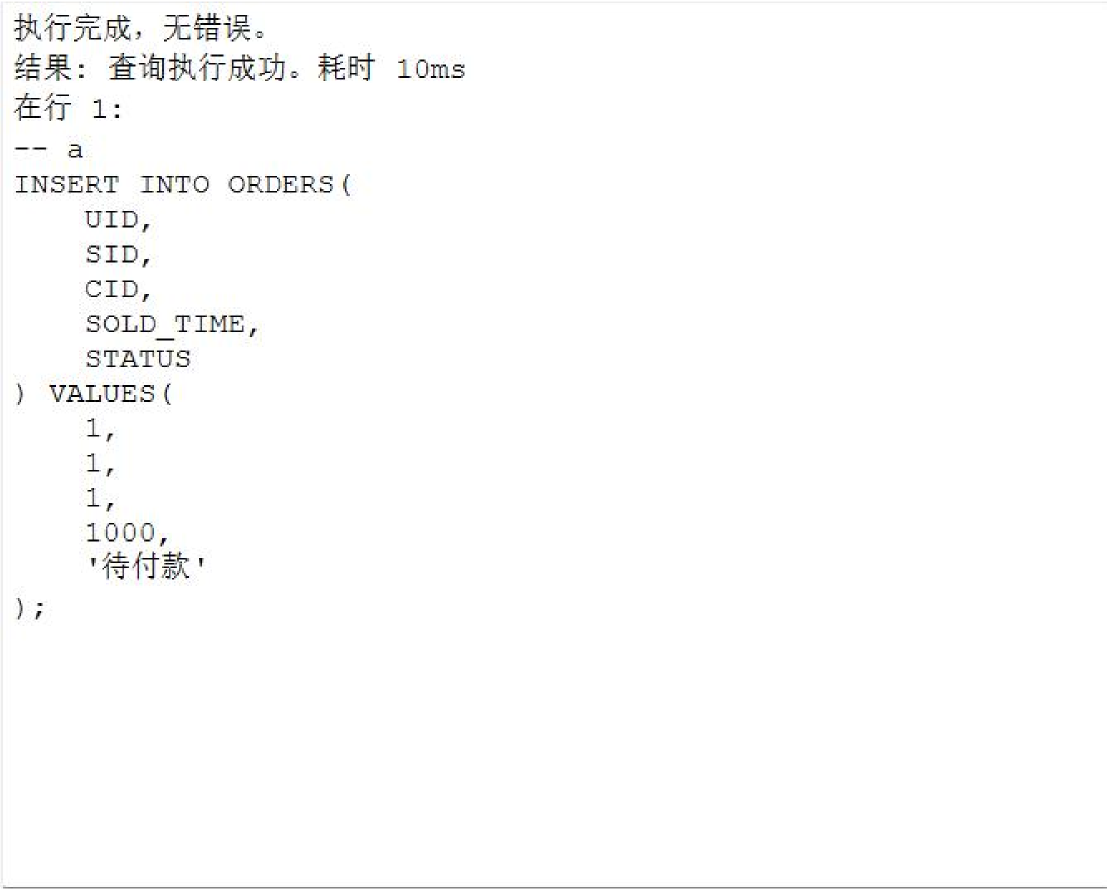
第九题（此为题目原本数据）：

a)

SQL：

```
-- a
INSERT INTO ORDERS(
    UID,
    SID,
    CID,
    SOLD_TIME,
    STATUS
) VALUES(
    1,
    1,
    1,
    1000,
    '待付款'
);
```

结果：



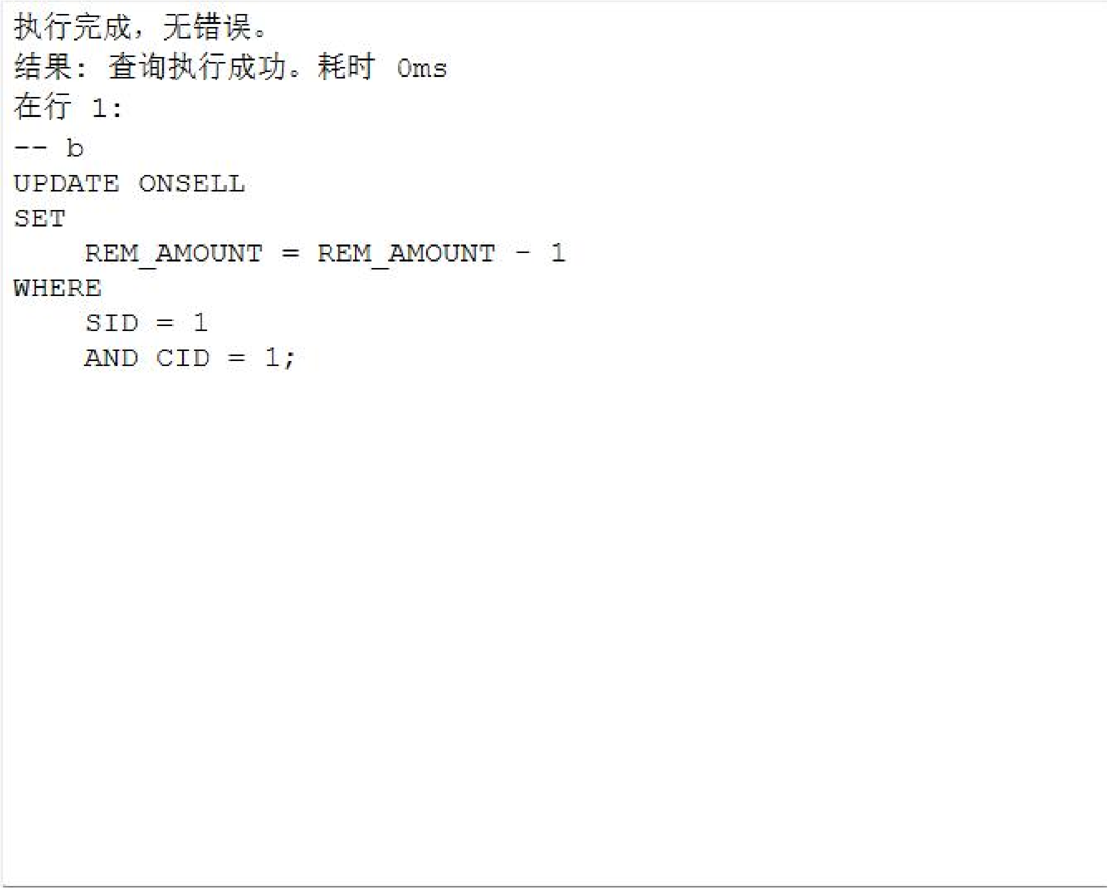
```
执行完成，无错误。
结果： 查询执行成功。耗时 10ms
在行 1:
-- a
INSERT INTO ORDERS(
    UID,
    SID,
    CID,
    SOLD_TIME,
    STATUS
) VALUES(
    1,
    1,
    1,
    1000,
    '待付款'
);
```

b）

SQL：

```
-- b
UPDATE ONSELL
SET
    REM_AMOUNT = REM_AMOUNT - 1
WHERE
    SID = 1
    AND CID = 1;
```

结果：



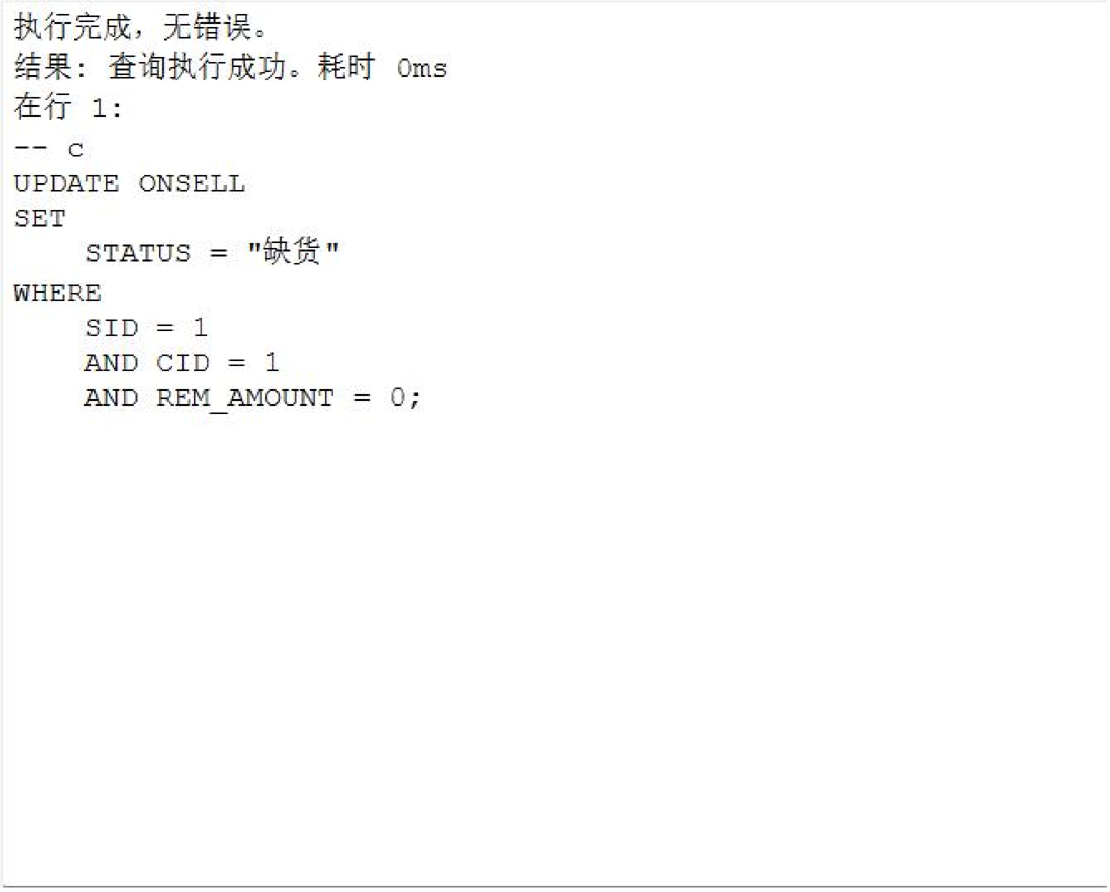
```
执行完成，无错误。
结果: 查询执行成功。耗时 0ms
在行 1:
-- b
UPDATE ONSELL
SET
    REM_AMOUNT = REM_AMOUNT - 1
WHERE
    SID = 1
    AND CID = 1;
```

c)

SQL：

```
-- c
UPDATE ONSELL
SET
    STATUS = "缺货"
WHERE
    SID = 1
    AND CID = 1
    AND REM_AMOUNT = 0;
```

结果：

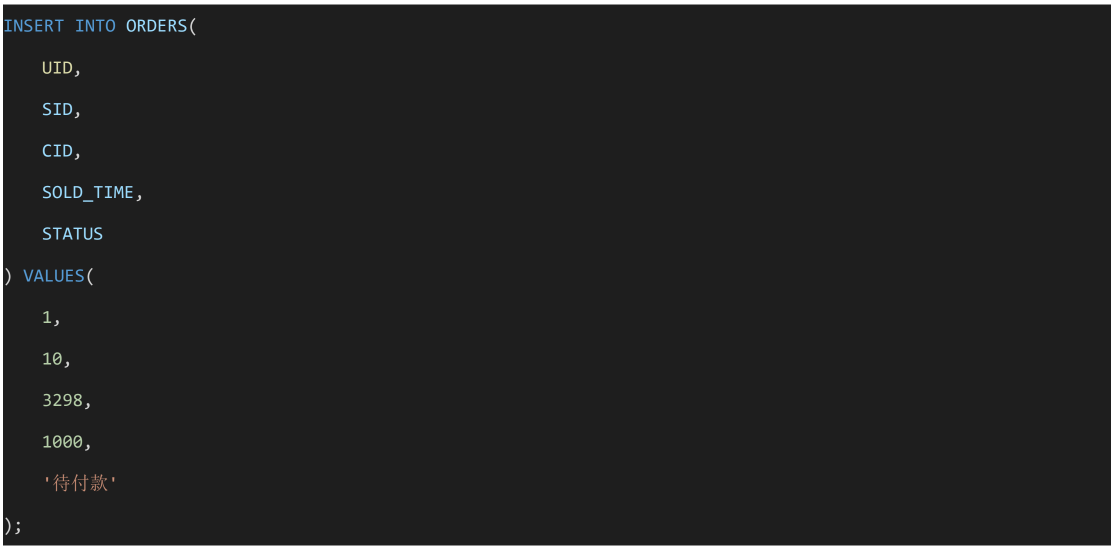


```
执行完成，无错误。
结果: 查询执行成功。耗时 0ms
在行 1:
-- c
UPDATE ONSELL
SET
    STATUS = "缺货"
WHERE
    SID = 1
    AND CID = 1
    AND REM_AMOUNT = 0;
```

第九题（此为改后数据）：

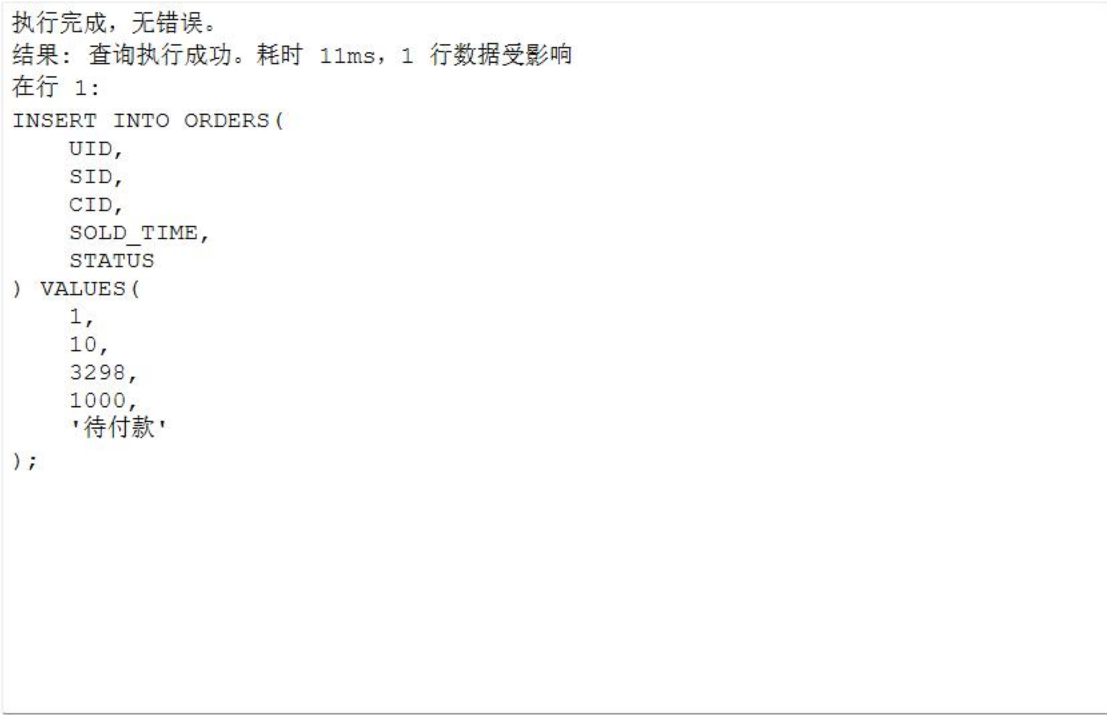
a）

SQL：



```
INSERT INTO ORDERS(
    UID,
    SID,
    CID,
    SOLD_TIME,
    STATUS
) VALUES(
    1,
    10,
    3298,
    1000,
    '待付款'
);
```

结果：



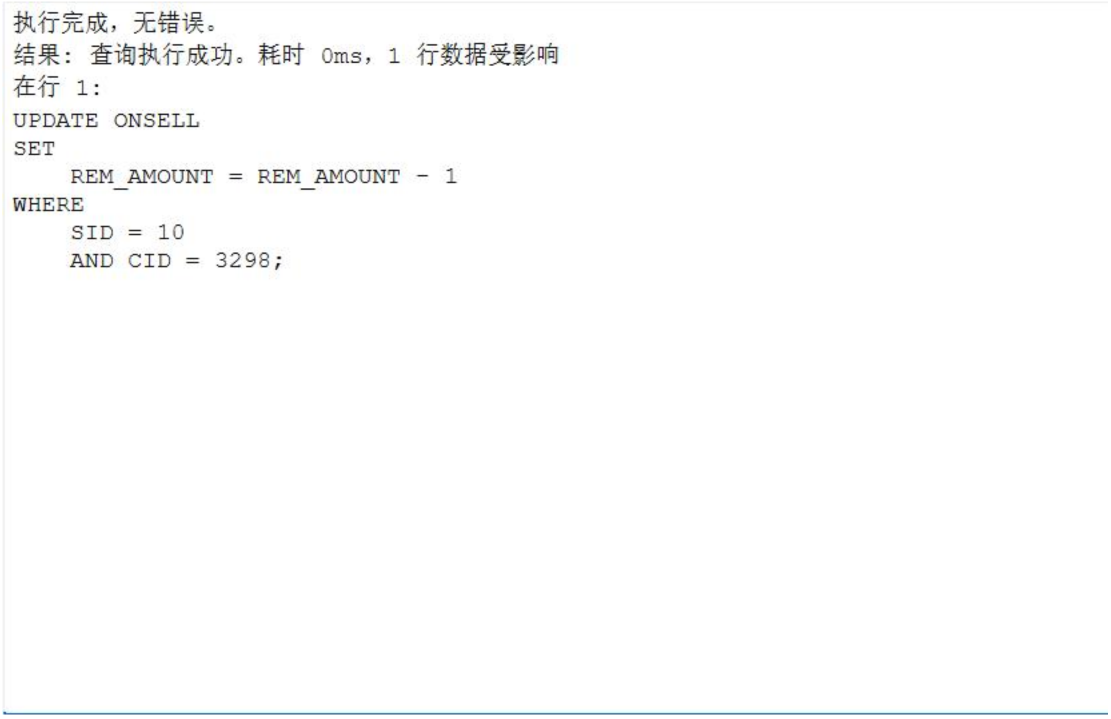
```
执行完成，无错误。
结果： 查询执行成功。耗时 11ms，1 行数据受影响
在行 1:
INSERT INTO ORDERS(
    UID,
    SID,
    CID,
    SOLD_TIME,
    STATUS
) VALUES(
    1,
    10,
    3298,
    1000,
    '待付款'
);
```

b）

SQL：

```
UPDATE ONSELL
SET
    REM_AMOUNT = REM_AMOUNT - 1
WHERE
    SID = 10
    AND CID = 3298;
```

结果：



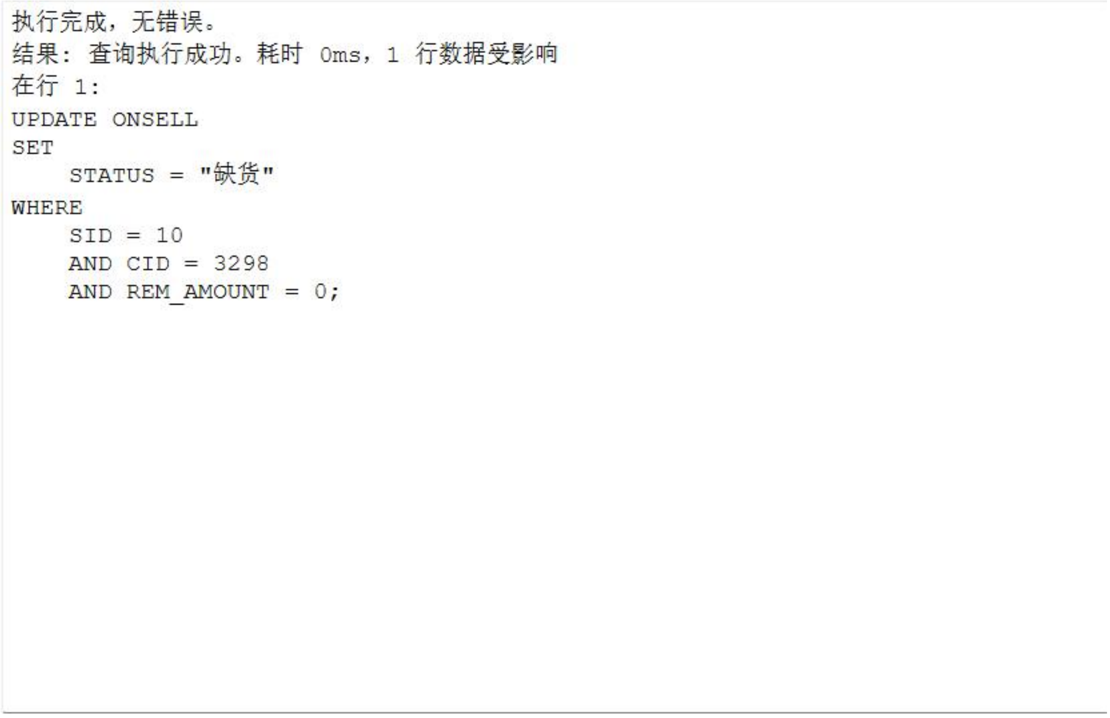
```
执行完成，无错误。
结果: 查询执行成功。耗时 0ms, 1 行数据受影响
在行 1:
UPDATE ONSELL
SET
    REM_AMOUNT = REM_AMOUNT - 1
WHERE
    SID = 10
    AND CID = 3298;
```

c)

SQL：

```sql
UPDATE ONSELL

SET

    STATUS = "缺货"

WHERE

    SID = 10

    AND CID = 3298

    AND REM_AMOUNT = 0;
```

结果：



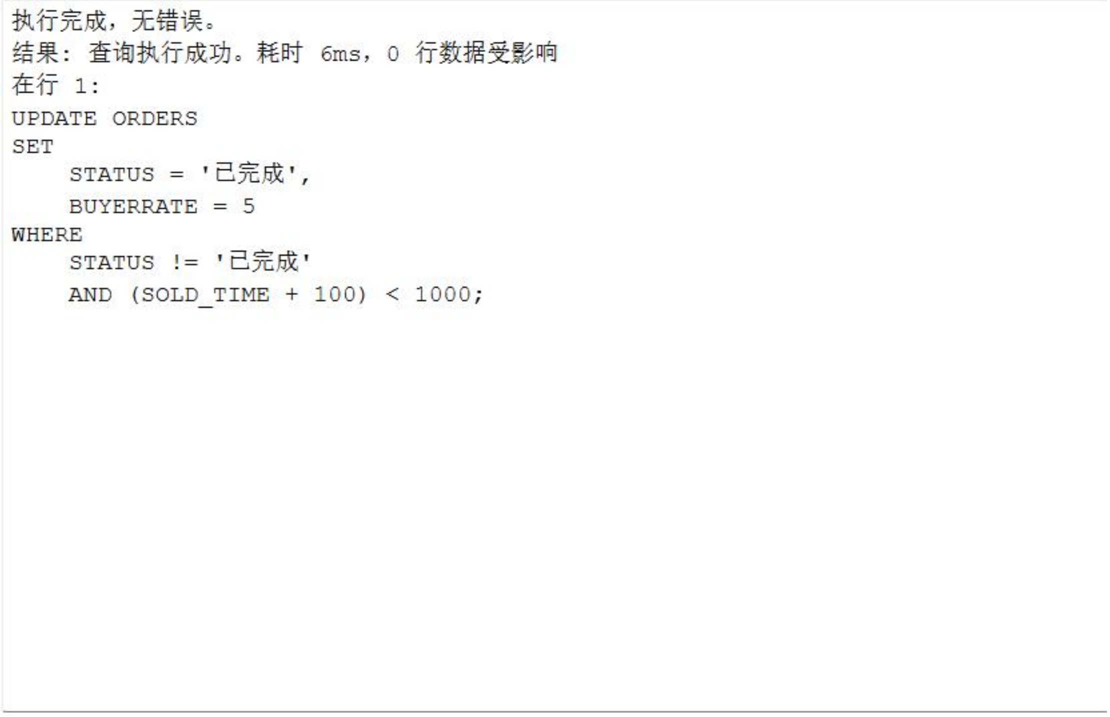
```
执行完成，无错误。
结果：查询执行成功。耗时 0ms，1 行数据受影响
在行 1：
UPDATE ONSELL
SET
    STATUS = "缺货"
WHERE
    SID = 10
    AND CID = 3298
    AND REM_AMOUNT = 0;
```

d)

SQL：

```
UPDATE ORDERS
SET
    STATUS = '已完成',
    BUYERRATE = 5
WHERE
    STATUS != '已完成'
    AND (SOLD_TIME + 100) < 1000;
```

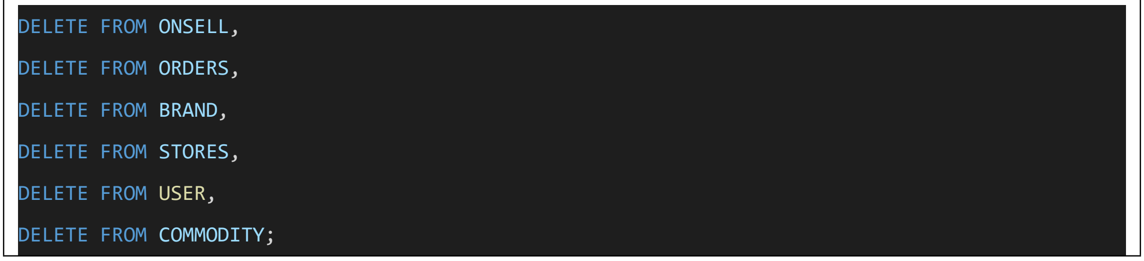
结果：



```
执行完成，无错误。
结果：查询执行成功。耗时 6ms，0 行数据受影响
在行 1:
UPDATE ORDERS
SET
    STATUS = '已完成',
    BUYERRATE = 5
WHERE
    STATUS != '已完成'
    AND (SOLD_TIME + 100) < 1000;
```

第十题：

删除顺序为：



```
DELETE FROM ONSELL,
DELETE FROM ORDERS,
DELETE FROM BRAND,
DELETE FROM STORES,
DELETE FROM USER,
DELETE FROM COMMODITY;
```