

第二次数据库作业

19335286 郑有为

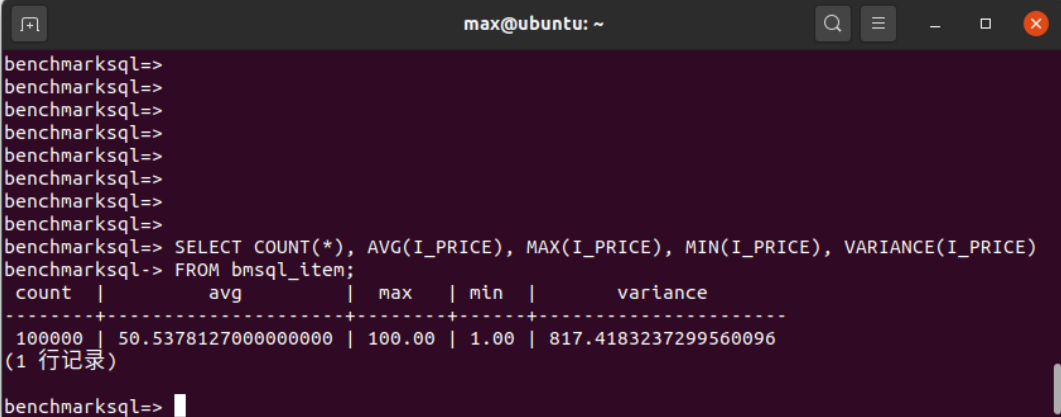
一、统计函数的使用：

1. 在表item中计算所有商品的数量，价格平均值，价格最大值，价格最小值，价格方差。

查询代码：

```
1 SELECT COUNT(*), AVG(I_PRICE), MAX(I_PRICE), MIN(I_PRICE),  
2    VARIANCE(I_PRICE)  
3 FROM bmsql_item;
```

查询结果：



```
max@ubuntu: ~  
benchmarksql=>  
benchmarksql=>  
benchmarksql=>  
benchmarksql=>  
benchmarksql=>  
benchmarksql=>  
benchmarksql=>  
benchmarksql=>  
benchmarksql=> SELECT COUNT(*), AVG(I_PRICE), MAX(I_PRICE), MIN(I_PRICE), VARIANCE(I_PRICE)  
benchmarksql-> FROM bmsql_item;  
count |      avg      | max | min |      variance        
-----+-----+-----+-----+-----  
100000 | 50.5378127000000000 | 100.00 | 1.00 | 817.4183237299560096  
(1 行记录)  
benchmarksql=>
```

2. 在表stock中统计每个仓库保存的商品数量平均值，输出列为w_id, avg。

查询代码：

```
1 SELECT S_W_ID AS w_id, AVG(S_QUANTITY) AS avg  
2 FROM bmsql_stock  
3 GROUP BY S_W_ID;
```

查询结果：

```
max@ubuntu: ~/Desktop
benchmarksql=>
benchmarksql=>
benchmarksql=>
benchmarksql=>
benchmarksql=>
benchmarksql=>
benchmarksql=> SELECT S_W_ID AS w_id, AVG(S_QUANTITY) AS avg
benchmarksql-> FROM bmsql_stock
benchmarksql-> GROUP BY S_W_ID;
w_id |          avg
-----+-----
 1 | 54.9943200000000000
 2 | 54.9252700000000000
 3 | 55.0740700000000000
 4 | 54.9528100000000000
 5 | 55.0172800000000000
 6 | 54.8447900000000000
 7 | 55.1507100000000000
 8 | 55.0558000000000000
 9 | 55.0977900000000000
10 | 55.0273900000000000
11 | 55.0335700000000000
12 | 54.9475500000000000
13 | 55.0674000000000000
14 | 54.9487600000000000
15 | 55.0087200000000000
16 | 54.9077400000000000
17 | 54.9587900000000000
18 | 55.0747400000000000
19 | 54.9785600000000000
20 | 55.0716700000000000
21 | 55.0311200000000000
22 | 55.2672300000000000
23 | 54.9141100000000000
24 | 55.1145700000000000
25 | 54.9243600000000000
26 | 55.0632500000000000
27 | 54.9765800000000000
28 | 54.9660900000000000
29 | 54.8561200000000000
30 | 55.0150600000000000
31 | 55.0053600000000000
32 | 54.9958200000000000
33 | 55.0424700000000000
34 | 54.9715900000000000
35 | 54.9378600000000000
36 | 54.8978700000000000
37 | 55.0074800000000000
38 | 54.9611400000000000
39 | 54.9138800000000000
40 | 55.1259800000000000
(40 行记录)
benchmarksql=>
```

二、正则表达式的使用：

1. 找出所有以'NB'为名字开头或者以'VT'为结尾的商品的所有信息；

查询代码：

```
1 | SELECT *
2 | FROM bmsql_item
3 | WHERE I_NAME ~ '^NB|VT$';
```

查询结果：共58条记录。

i_id	i_name	i_price	i_data	i_tm_id
5464	UPfnlY5BoiowL6KtbQFm3VT	43.79	P66ea1jVwRnPBjPjURjsBLTsysks8Tbkv	340
5801	NB1umKV0PDysyakHvIk4	90.47	SRSxytgLVAZic1sVlpQDWUGGsAG8PRNQ2	8414
6908	NB3j5jPfuIGKYoJfASrtd	13.80	s6eS8PpFr0iCbyFrwLnTuh3wP8LaZz1eLXW0rfhy2	4724
8632	NBKVvB6UhZVEbYs35INx	94.08	rGr2Hf0j87M9hyoUG3KAJKeMmJl	5704
8635	NB2WHD2f4pQv4mEv43	40.59	ZyMsNdZwBpq8djEWxeTzoORIGINALIkLt4kr9ydmYKl0	9738
9079	A8bdQ0KEd5cyfwSNFPkZhsVT	33.92	geIjocIsyBq42KsszaXSry4akjD8E3Nxe2lQsRL9	8776
9858	xcQTOWPMvtsUH6Xp86C4MVT	38.72	KksxFRCa0mC5P10HZXrak8mcRC1DHx	3454
10108	NBIdY38zG4txkY	63.54	JHZGwXL5Y9y8Bm1DTYLdcrH9l009rhIARceMDMKZpYRX0	3216
12523	NBcA1h7G046sSrzmZZyn2V	5.44	QURjyqu34a8rFN7xJEmUCb8HpjsMNMJC6JQSDb7DrNgbSMAAQ	5156
13619	skxmC07NBMenkPApBwrGmdVT	9.14	gF6ST4AnkChap80Jgn0C0gzHupZauQD8ddDmdMl6ncf2KMPMcw	7055
14893	ZHNIquFGwAIEHE1kVT	5.55	DnwXLN4uHa8gydyon7uSdzFkb4vWQ4UV7STXJ40b	3825
15490	NBstfOPbQHR8J2	99.82	TkrG5vm04lpJL2ZE66h0PQpf9zQcFgl5P7V5Bb0	7631
16546	cRrrrJVJvFegwmpKU8VT	83.72	UdHKRaU252KYmwCU025yBu58LCd2b25Zi6	3893
17282	NB37wZXvRT7nlfxoS89gzEW	99.84	cRyJQd13gTCVxPWoIctQaXs8hVnIwvSRHP6QJ4Le1iAQ0E2EET	5679
22165	t0EGQ5vpFSfVX9ZxVT	60.83	GSSsHr39PZr00nL6vyudDhc7zuRGd0jL2be8rmXaMTi	8544
25402	NBY0dYH4FLb4b2nBPg	84.23	SGQqK0wEgu9srLEsmF4qpEHOEte	1203
29097	NBctmNiQTVRrbt	39.47	o3SIfFBEUmn252425sQoRBdk7NLOGTWyESUWFZ	7206
29588	RJ0i4A6d4Ufz3xzQZVT	13.25	UH7jsUebcVJCRSyd5IavtrNU95q7rG3JH0kW3Zp	4132
30997	NBSx7leV8rFkTb1NXd6jh	86.72	X66Sakj57DzDJ03yp2ikJDeGserd76aeAznqb2	9704
32709	NBrrNPezKRCNZafPXHNyd	47.10	TeTTh9PhFJRfwxjdyLiI7e2m55p1q2ldXz8Wa7A56J3p	9820
34698	NBHN2QbY7i0fQ52vU	56.38	cAzy8uL2tGihhRBGffdlTGukJUmD	502
38026	NBC0wArMn9VSG5	26.73	L3aorPrTU7Y7aRHaH7V0dQgJlyGJzTL0	7000
38796	NBYUQukyzV0pqqQfgGUXb	71.67	vJUXbb0TKhfakEp0SFY6QUtxX2NbbLkDQDrr7	8848
40151	YkKB8z72UGzLWytLNdScSVT	72.39	Eu8dm0W9lEzQQ17G8j2Zc2fZcpCgY8adTx4j60IvxAp2GEyI	7287
40948	z4wXBfAGovJu27myQVT	57.61	mguaa7RyAQf8t54V1JIBxbLHrzVvhlz17F8FIkeGy0Xs4icGgC	2555
42389	NBZNUeAmmTTG0G	87.93	xJ4LCLZLlTSRPSY7nuF3Uv6BMBbn2WjyGiPNjI3tEeDmVr	844
43231	SkWqFdA2msa0inKvW8Box0VT	15.09	JVwXRWA0lq58ZaL8N8pvd1P9h8e	4113
45897	UQkqLHiNFnmQVVT	32.09	LrQXQORIGINALwLG5zc16cL4pdq03aog4Bk	2834
49011	NB5sGcwnEQnsFoz5rgfy	11.76	psIJhLMyXsmXLMW6v6iz4zspVscGBD4Cd2kW0c	5049
53235	NBzfUBFAhZcehVnqS3	60.45	vyMXt6PHBTWMeHwCt62dWYL777f0ZCdMeUh	1632
53649	yuk2SjJpI1PdzhKSeVlVT	75.00	dERGYMFV75XVv4P1mZwh9NfBLK	1571
54481	I63zXyLA1t9Sb1ND8VT	29.67	j9BiH3g2ybIaxkeJwjp1fEKQ59XFKTiX35YdgZAZ61k	5426
55096	zSj4fELsAHVTV2kwvVT	9.40	EqFCkH05oCtbUP2VBgn00yKHORIGINALNG84gEhjvSzeHa	2188
55854	pEqd6sN4IR2ncnsnIqnMVT	88.64	LigvFRdx2Hgh69WawxiCgFi20Y4tvo0o54Xr	862
58503	E5omXQ5bJuv77f0qXVT	18.75	fVFFP0XpEs0bxE1rcFyK0e0xQeKVkJumcJImbUSRADAK	6736
58866	nMd3e7P6K3UaHECXU0VT	75.23	yTyu0hojTmSV9ctowQB0S8ogZMCYfZACqP7vXITYm2eLiFL	2775
63971	TiP9rsE3Hrx7VT	16.24	jJPjoXmQJpfKTzB3lWtK0egTgeuSntbm	475
66083	CoexSww5Hev0B1OVT	35.22	oh4jocowq8PKeuBqW22akdiM8mGneBosFNdXvPwaxWX	6937
67300	NBGZgLzUQSVF4J22kyjALTu	33.82	X9xAk00541SXj31aND0o6AqjT6tyYhJdh	7800
67607	NBToaxiQ5bctk2lAIXub	42.04	y8ZvmGhH0vx7BBEv9njZ15NP6piUuaIcGM3VTK7LA	882
67682	NBeIsiVNLi0SL3dt52xEDh	91.78	H0B0fSL4L8ke5Z0KLlW7Ytaf4QX	4128
69786	DAGV4QPkBPMYAVNL7iVT	9.27	sU4xiM5ax7A9zm8Ifr3XIqKCwDdxksy0n2	9710
70677	YoxmJMXghYEWdVT	41.62	Fkfqbv0xUjviI4P9fdi9wLurnTcgLgE70e7rdQfRGQMIdqH9r	3246
73156	dV6MKIKJcIWNPKmApVT	80.16	GLCXXJzZKguK1IEPIa4558Mzr06phjU0gdr2082	7683
74972	NB0lc8BkpQSDw8cgZY	5.07	oxPNg82BBuNLlgdn1DNZpvSVVG2zjydfpzz3pL8	2519
76319	NBfxDEi4EVYhqmN	69.92	e7820P0jE8VfHfT86m2NI2OM2wy6rcKkw7kk1KuhNIXh1	8725
76803	Y69VwpckxYvH1f7ozVvVT	86.31	stLIvP9XmTpfWmjekrPVns1zg	3062
77879	BGLGbE3GdG0dAU0enA20VT	5.08	ORIGINALN7uUN7vG03YoTK50aub	9574
78144	NB1TLGIARoyjmZuLJWB	21.25	bauus8wvBE1d0jvTjHycTnyYTJB4Lq6MxxUp2q	1147
85041	TSJpHeIrZVBEOdVT	39.18	Kc4zon4kEHeEVgWAbMyr8CyYPUGKS7xfLORIGINALmPU8u	7267
78144	NB1TLGIARoyjmZuLJWB	21.25	bauus8wvBE1d0jvTjHycTnyYTJB4Lq6MxxUp2q	1147
85041	TSJpHeIrZVBEOdVT	39.18	Kc4zon4kEHeEVgWAbMyr8CyYPUGKS7xfLORIGINALmPU8u	7267
87117	NBP8otzPuz3cdNGC30WZ	26.92	g3nVbIOFX3bLH40xp7qyQds77GRa3TSVvJrwpY7FAeauEL	6067
87693	NB6Ei1L73EoK8CnVFsD6aPl	23.51	pYA83Cbrdxzri0RBCLcLcsK7MIQEG4EHpMrwvo6JCP	8103
88909	L1yL5z3YeXzPoUVT	95.54	xCJx1XyFyThRoHk5WIo0CKRe6beuPYyzzp	8127
91741	KW86LNG01TiIEOVT	75.29	gA29aBy79v008JKHFipwL4FTNQ00y00w	8815
93646	NBn5YGak2Q0QVYmLLHI3nthN	85.23	r9I0QAcz7bF9eF8J42iaehZwZh6jMm8V2E	552
95620	LyWzrYeIeqpk8nVT	92.89	NaL4Zg16j53u8KHjP4F244AaQDMosnwZCorGAJwZT	1682
96215	NBfDecGGBMLHV16	44.40	BgJpoHCJycCVfH72BSZM9aLQuzjGg2svAoCsxGpby00L4Yy	8348
97174	NBDGG6DBtg87GibYUXJ6b	30.59	X7g009F8Izpf7xtcVQRn8IdnoMWrFqZ3g	832

2. 统计以名字开头字母在h-m之间的商品数量，以及平均价格。

```

1 SELECT COUNT(*), AVG(I_PRICE)
2 FROM bmsql_item
3 WHERE I_NAME ~ '^h-m';

```

查询结果：共1条查询记录。

```
max@ubuntu: ~/Desktop
benchmarksql=>
benchmarksql=>
benchmarksql=>
benchmarksql=>
benchmarksql=>
benchmarksql=> SELECT COUNT(*), AVG(I_PRICE)
benchmarksql-> FROM bmsql_item
benchmarksql-> WHERE I_NAME ~ '^[h-m]';
count |          avg
-----+-----
11547 | 50.6911881874079848
(1 行记录)

benchmarksql=> 
```

三、all/any的使用：

我们作以下定义：

如果商品a在仓库b中的数量大于10，则称为仓库b有储备商品a；

如果数量大于50，则称仓库b有充分储备商品a；

如果数量大于95，则称仓库b主要储备商品a。

1. 对于二.1中的商品，找出在所有仓库中都有储备的商品，输出商品的所有信息。

查询代码：

```
1 SELECT *
2 FROM bmsql_item
3 WHERE I_NAME ~ '^ANB|VT$' AND 10 < ALL (
4     SELECT S_QUANTITY
5     FROM bmsql_stock
6     WHERE S_I_ID = I_ID
7 );
```

查询结果：共31条记录。

i_id	i_name	i_price	i_data	i_in_id
5464	UPfnlY580i1owL6KtbQFm3VT	43.79	P66ea1jVwRnPBjpURjsBltSysiksS8TbkV	340
5801	NB1umKV0PD9yeakHvik4	90.47	SRSZxytgLVAZic1sVlpQDHWGgsAG8PRNQ2	8414
9858	xcQT0wPMvtsUH6Xp86C4wVT	38.72	KksxFRCaOwC5P10HZxrak8mcRc1DHx	3454
12523	NBCA1h7G046sSrzmZZyn2V	5.44	QUrjyqu34a8rFN7xjEmUCb8HpsMNMJC6JQ5Db7DrNgbSMAAQ	5156
13619	skxnc07N8MenkPapBwrGMdVT	9.14	gF6ST4ANkcHap80Jgn0C0gzHupZaUQD8ddDmdML6Ncf2KMPMCw	7055
14893	zHNIquFgWAIEHE1kVT	5.55	DnwXLN4uHa8gYdyon7u5dZfkb4vWQ4UV75TXJ40b	3825
15490	NBstf0PBQHR8J2	99.82	TkrG5vm04ipJL2ZE66h0PQpf9zQcfigl5P7V58b0	7631
17282	NB37wZXvrt7nlfXo589gzEW	99.84	cRyJQdigTCVp9WoIctQaXsBhVniWvSRHP6QJ4Le1lAQOE2EET	5679
22165	t0EGQ5vpFSFVX9ZxVT	60.83	GSSshRX39PZr00nL6vyudHC7ZuRGdojL2be8rmXaMxTl	8544
25402	NBY0dYH4fLb4b2nBPg	84.23	SGQqK0wEgu9srLEsmf4qpEHOete	1203
29097	NBCtmNiQtVRrbt	39.47	o35IffBEUMn252425sQoRBdk7NL0GTWYESUWFZ	7206
29588	RJ0l4A6d4Ufz3xzQZVT	13.25	UH7jsUebcVJCRSyd5IavtrNU95q7rGJ3H0kw3Zp	4132
32709	NBrrNPezkRCNZafPXHNyd	47.10	TeTTh9PhFwJRfwmjYdiI7e2m5Sp1q2LdXz8Wa7A56Jp3	9820
34698	NBHN2QbY7ioFq52vU	56.38	cAzy8uL5TGihhRbGffdTlGUKJUmd	502
38026	NBC0waRmN9VGS5	26.73	L3aorPrTUY7zaRHaH7V0dgQJlyGJzTL0	7000
40151	YkkB8z72UGzLWYTLNnDscSVT	72.39	Eu8dm0W9lLeZQ017G8j2zC2fZcpCgY8adTx4j60IvxAp2GEyI	7287
40948	z4wXbZg0v3u2myQVT	57.61	mguaa7RyAQF8t54V1JIBxbiHrzVvhlzL7f8FIkeGy0Xs4icGgC	2555
42389	NBZNuEAmmTTG0G	87.93	xj4LCZlLtSRpSY7nuF3UV68MBbn2WjyGiPNji3tEednVr	844
43231	SKwqfDA2nsa0inkvW8Box0VT	15.09	JVwXRMA0lq58ZaL8N8pvd1P9h8e	4113
45897	UQkqLhlnFNhmQVVT	32.09	lrQXQ0RIGINALwLg5zc16cl4pdq03aog4Bk	2834
55096	Z5j4fELsAhvTV2kwVVT	9.40	EgFCKW0SoCtbUP2vBgn00yKHORIGINALNG84gEhjvSzeha	2188
55854	pEqd6sN4IR2ncnsnIqnMVT	88.64	LlgvFRdx2Hgh69WawxlcGfi20Y4tvo0o54Xr	862
58503	E5onXZAHEV77F0qXVT	18.75	fVfFP0XpEsObxE1rcFyK0e0xQeKVKJuncJImbUSRADAK	6736
63971	TLp9rsE3Hrx7VT	16.24	jJPjoXnQJpfKTzb3lWtK0egTgeu5ntbm	475
70677	YoxmJMXgHyEWDVT	41.62	FkfQbv0xUvjI14P9fdi9wlurNTcgLgE70e7rdQfRGmQIMdqH9r	3246
74972	NBoLc8bKp5Dw8cgZY	5.07	oxPNg82bBuNLIgdn1DNZpSVVG2zjydfpzz3pL8	2519
76319	NBfxDEl4EVVhqmn	69.92	e7820P0je8VfHTB6m2NI0M2wy6rcKkw7kk1KuhNIXh1	8725
76803	Y69VwpcKxYVh1f0zVVT	86.31	stLIVp9XqMtpFwMjkrPVns1zg	3862
85041	TSJpHeIrcZVBEOdVT	39.18	Kc4zon4kEHeEVgWabMyr8CyYUGKS7xfLORIGINALmPU8u	7267
91741	KW86LNG01TYiEOVT	75.29	ga29aBy79v008lKHFIpwl4FTNQ800y00w	8815
97174	NBDGG6DBt87GlyUXJ6b	30.59	X7gz009F8IzpF7xtcYQRn8ldnoHwrfqZ3g	832

- 找出至少有一个仓库主要储备且该仓库销售税（w_tax）大于0.16的商品（使用any），输出这些商品的所有信息。

查询代码：

```

1  SELECT *
2  FROM bmsql_item
3  WHERE 95 < ANY (
4      SELECT S_QUANTITY
5      FROM bmsql_stock
6      WHERE S_I_ID = I_ID AND S_W_ID IN (
7          SELECT W_ID
8          FROM bmsql_warehouse
9          WHERE W_TAX > 0.16
10     )
11 );

```

查询结果：共32689条记录。

i_id	i_name	i_price	i_data	i_im_id
4	lAzAKhfff7o88Es2jvRqa6C	14.68	nHk4tBSAD67tLWNaZNTg4aLVX8Mzy0TASFCg3yMC50jcvF	9876
5	w4PJ3VvwPMugKTDJDHmkhkBog	60.13	g9j06i11Pw9CTgLa9xttL2ZAVjVFrN5kBMry	8437
10	L1YmTAPc8fVhLc	40.97	kkt81XTydkRBvHUxx1Iw0cuCovQFXmrRhgQzn3I4iLzt	7935
12	nI9H9d0otVKIJWTctY9	48.12	sk0NoAVUhh7MPoRKM6Vap2ELGLoem5fypmORIGINALxpVz1	8646
16	JpLcQGXKZ8mQHM	99.60	C1FC0iM0Rhn0DzQbPSy5HkCqEF0aJzhrxc2z2qq1l3A	6213
20	QCfmGbd6ZHD19gvJ40	72.76	uSVIXaNe163R5MmPTfNMVpB60D5XRJX	4842
27	LU7MxUsq0rTtyT	12.10	Pp5NmnBXLi4vqLpz5oN2ktzh0tEh63rIkKJzqNkOZiS73Zmz	3182
28	UGTYd96scC0gqggY	73.89	cXGMeh2E0iEHORNJ2Seq46eFJZKXuQ58fdf73csGNl9U7n	8969
29	n08AyQmLemFuLR0EtCTK	19.79	xi258TdoSiYKoh3BEJKCB5Fg7FEFok4EFy	5405
32	Gh2ZvpuMRGHmftT3	99.10	miZ9eSUKxkVKuPaYbRCMZJqWZahPvpq0cdBn0cejFjJz	1547
37	QdKIpuUHVNWwFZzI	25.50	idvKLaUGHS0G0FPJYvYLyXWeocoJ5ur5zuR	9925
38	NSp0NgM8wNFUQKv1fLYhesq	8.65	KoNnvQPB8bp1f9dVUIYPq01pj7HHAGThFpfy	5995
40	p9ewdu3U9v7Qh8s0rGyus	29.98	rMyRlNfWiZrPsjVI5JNjRjXMHxMSUzt0SkEZ2g	3951
43	ZTojcNmgyd7MwFcZ8x	11.40	LzYGJtsPLAVHTV3t5j3gWUX4wyUxMSVE39FFqm	4776
45	tzutM3iKcbv7BsWfjNJCTRSD	26.36	IrfXGyRZRUgovZjCjtcYR88KKBSbz8Num0HYltgQt8zrjq0	7716
50	IH3mTEA2xQyZ8zKkDOXN	15.84	DmgbjbTikj85foU0tD23xW1tB021wY	6501
52	SLt1Zlq00qET5Dp5R20bm	68.43	jLE0aPORIGINALSR33XwrytRE25dGlx2Z0Z27BpWID	2510
53	UZ7yHsc036gNrvMG4Ib	72.47	DG3IL0o2ybvUkl346s61vAXFZgviqq3vEXh5W	6740
57	nFTLImq7G9M6377PxKoBMBHy	78.93	msBA1qjb9ljCD0QSLWNRRx1LMFMiU2	1682
58	B7SrKpppEva3sSY1d	95.77	fECWDrgyl3toCQZahYWR5yXD8CUjrz6N2AR	2857
59	q0j0FnCKLEgYWHJt	28.35	a2wmwpXUGxrpUPbK9vdwQLgVwGxMr	1775
68	tkR8JXS08n0AP0VYOjxC	69.29	L1x6Zckv8ufpkNfBr1EBiemXWvORIGINALuyWg7LEk6liqkGI	1344
73	frQLXQpyZV8YXe0	86.74	dTL4o04q7TQQKAA0MNYKZBD14BbacJC	39
74	OdKh5FGTWXctc95c	41.28	xJDBdRdb8mjK5o8ajhBubZf5oBp8pc	9989
75	OMmCbLS7MK3q1LN	6.45	eORIGINALegZcPncnwqTKUuCWJlpZBasslz843U4HX6fq	8670
80	wcHtceupuFcZd3A	3.71	LoIdxySMeap2F1LpUY8Rxo52G1PIElYeeMaAUCJmlyZm0QWx	9471
82	c1w76ifa6vhiEbQX	97.94	rdci3lL8rjTcf0mKKpqKek1y2zBll8mv9lNutBPuGyd9LJIf	3072
95	VjWjLJLVmCA0ISVt	24.20	aR6csLxy16zpqHdfAT5qc0x7NXllef22yJ0f	9578
98	rIjhsY2gmIbcsCw4cRnD1zb	9.42	xTLT5B0QxMC64jEkaGtQWU71HtYirEexTj2	8344
105	nM0210lzeYusjf6ZVF	19.13	FciwykzuguHppGjoxu2pJUmBTc1nJo89S	7883
107	pMi0ezbg7E7F4N4aL	96.76	bsl1LXLZiYiTCqtpw1urCkH4YLaSjpYv	6083
110	au0b15H4yXIIIDc7Lr	89.27	sJl0DYTPR3X0KYMD7ThX6RrK3njuMx0L5	9191
112	K3H8hk14q1iUGewVxNB	77.27	EujI8H3cqsLpB06G0QZq2md0E30GAIQ3119J09e	6866
129	TGXbTlsjG8LxR1U	41.24	ccXUfPTwadTigfb44RMP2wr8gNf32e092uC	2248
130	lu2N01GP1yi5U0zA	62.88	JPAwKufhMpZkBs3pyRAZyruoDVYWNFLORIGINALFwu	6962
131	coo7Icohmpufeaob	4.14	M9rcVeZEULdzjo7kiz0uVpz7U0ahnn7qzjePFsavsQ6	6575
133	hgEh3jKHJNPean	64.19	hVxa0DSEfP8k5RxxYrKjc0szB06ayDztm	5176
134	pxcQ1TQk8hdMmM5ai7KtKJMe	52.82	n5T4hhn99FAttKp94Kmg9F3TaCClWRVvUePGYeb09poUSFU	1691
140	Cs2x7ZmMtoV0hK1UtJaK	20.19	ukeJkPCiD0cSukr0Gu68CgXTY7hrIU	897
144	E40VIDPdlF9DWRNltLrS1	83.59	cSv9pUQAxHT0jhsXRCVtEMgvGGuSwIbI3XDzj5FHYmLWZ0K	3668
145	sLAVnXccLHejHj0gHrUCb9	78.06	aNZC6JhrBNmJuILRBacFY73PHMcGa6tfWUp8b	3962
150	jPGWvSMLXAbiSDVq	68.00	nQYYuVlWfJ6hgTzpSAvyYwdHenjLczKJCtdxP1jUXIw24x	487
154	VTR4aAnitjXPLEV6LK5	8.75	CF2gBTotVtI66tk0iGMTtVvF9MoqdZ8LYaFT00wu6	4671
158	q3lnw3g8bdQheI	82.83	tQGWDw2xPaFXVqIT7yvvj1QRt3qetIfZtQuD	1323
161	bwG3UmXuc75RaXdhZgJLPshx	84.64	WzhpIcjZhVFYn3Dd0eH2LmTdg3Aionc0cE	1566
162	ecb70XhDspCNAAX9oiW2LP	82.00	Zncyujc46K0SaUbrF4ZfnKLwfruyS6YN	5166
164	Znc2Gvo6Biy10wJ8X	43.51	GQ46TI0GBss72VAKFCNb702vf2dMorw	6377
174	P0SzMtaVU5Whuy8LUuxz	61.79	m6V6ugVkt8XgH4048XWayofohPDhXv	789
182	CTDA5zvHujGqDFXoFPsNBD	62.21	Zkeofxj8neVTm8UWN8WYwnfav8QYzu	4971
184	QM8NGs4DVtFhB9Ce	76.45	TpNe0TPESI37xhUaHCpugq6jD70BzNPL	6810

注：

- 查询的条目太多了，修改查询代码来得到总条数

```

benchmarksql=>
benchmarksql=>
benchmarksql=>
benchmarksql=>
benchmarksql=> SELECT COUNT(*)
FROM bmsql_item
WHERE 95 < ANY (
SELECT S_QUANTITY
FROM bmsql_stock
WHERE S_I_ID = I_ID AND S_W_ID IN (
SELECT W_ID
FROM bmsql_warehouse
WHERE W_TAX > 0.16
)
);
count
-----
32689
(1 行记录)

benchmarksql=>
benchmarksql=>
benchmarksql=>

```

- 这是使用IN代替ANY查询的代码，查询结果与上述查询一致


```

1  SELECT *
2  FROM bmsql_item
3  WHERE I_ID IN (
4      SELECT S_I_ID
5      FROM bmsql_stock
6      WHERE S_QUANTITY > 95 AND S_W_ID IN (
7          SELECT W_ID
8          FROM bmsql_warehouse
9          WHERE W_TAX > 0.16
10     )
11 );

```

四、嵌套查询(in):

1. 找到有商品税大于0.18的仓库主要储备的所有商品，输出它们的所有信息。

查询代码：

```

1  SELECT *
2  FROM bmsql_item
3  WHERE 95 < ANY (
4      SELECT S_QUANTITY
5      FROM bmsql_stock
6      WHERE S_I_ID = I_ID AND S_W_ID IN (
7          SELECT W_ID
8          FROM bmsql_warehouse
9          WHERE W_TAX > 0.18
10     )
11 );

```

注解：

- 内层语句：找到所有商品税大于0.18的仓库的仓库ID；

```

1  SELECT W_ID
2  FROM bmsql_warehouse
3  WHERE W_TAX > 0.18

```

- 中间层语句：从库存表查询所有仓库ID属于内层查询结果，商品ID为外层指定的商品数量；

```

1  SELECT S_QUANTITY
2  FROM bmsql_stock
3  WHERE S_I_ID = I_ID AND S_W_ID IN (...)

```

- 外层语句：根据中间层查询出的商品数目，使用ANY语句，只要该商品满足存在一个仓库该商品库存大于95即满足条件，输出满足ANY条件的商品的全部信息。

```

1  SELECT *
2  FROM bmsql_item
3  WHERE 95 < ANY (...);

```

查询结果：共20182条记录。

i_id	i_name	i_price	i_data	i_tm_id
5	w4PJVVwPMuqKTDJHmkhkBog	60.13	g9j06i11Pw9CTgLa9xtitL2ZAVjVFrN5kBMrY	8437
10	L1YmTAPc8fVhLc	40.97	kkT81XTydkRbVHUXx1Iw0cuCovQFXmrRhqQzn3I4lLzt	7935
12	nI9H9dDotVKIJWTctY9	48.12	sk0NoAVuH7MPoRKM6Vap2ELGloem5fypm0RIGINALxpVz1	8646
16	JplcQGXKZ8mQHm	99.60	C1FG0iM0Rhn0DzQbP5y5HkCqEF0aJzhrcz2z2qq1l3A	6213
27	lU7MxUsg0rTtyT	12.10	Pp5NmnbXli4vqlp25oN2tkzh0tEh63rIkKJzqNkOZis73Zmz	3182
28	UGTYd96sc0gzzgY	73.89	cXGMeh2EQiEH0rNJ2Seq46eFJZXKuQ58df73csGNL9U7n	8969
29	n08AyQmLemFuLR0EtCTK	19.79	xI258TdoS1YKoh3BEJKCB5Fg7FEFok4EFy	5405
32	Gh2ZvpuMRGHmftT3	99.10	miZ9eSUKxkVKuPaYbRCMJqZahPVpq0cDbN0ceFjJz	1547
40	p9ewdu3U9v7Qh8s0rGyus	29.98	rMyRlnfiZrPsjVI5JNjRXMHxMSUzt0SkEZ2g	3951
45	tzuTm3iKcbY7BsWfjNjCTRSD	26.36	IrFXGyZRuGoVZDjctGyRB8KKBYsbz8Num0HYltgQt8zrjq0	7716
53	UZ7yHsc0J6gNrvMG4Ib	72.47	DG3iLo02ybvUkl346s61vAXfZgviqqv3EXh5W	6740
58	B7SrKppEVA3sSY1d	95.77	fECWDrGyI3oTGQzaHYWR5yXD8CUjrz6N2AR	2857
59	q0j0FncKLEgYwHJt	28.35	a2wmwpXUGxrpUPbK9vDwQLgVgwXmR	1775
73	frQlXqpyZV8YXe0	86.74	dTL4o04q7ZQQA0MMyK2bZDi48bacJC	39
74	OdKh5fGTWxctc95c	41.28	xJDBdRdb8mjK5o8ajhBubZf5oBp8pc	9989
82	c1w76ifa6vhiEbQX	97.94	rdC13LBrjTCTf0mKKpqKekiy2zBll8mV9lNutBPuGYd9lJIf	3072
98	rIjhsY2gmIbcsCw4cRnD1zb	9.42	xTLT5B00XkC64jEkaGtQUW1HTYlrEextj2	8344
105	nM0210lzeYusjF6ZVF	19.13	FcIwykzugHppGjoxu2pJumbTC1nJo89S	7083
107	pMi0ezbg7E7F4N4aL	96.76	bsl1LXLZiYiTCqtpw1urCxH4YLASjpyv	6083
129	TGXbtLsjG8LXR1U	41.24	ccXuFPTwAdTigfb44RMP2wr8gNf32e092uc	2248
130	luzN0lGPlyi5U0za	62.88	JPAwkUfhMpZkBs3pyRAzyruoDVYWNFLORIGINALFwu	6962
131	coo7IcohmwpufeaoB	4.14	M9rcVeZEULdzjo7kizOuVpz7U0ahnn7qzjePFsSavksQ6	6575
140	Cs2xF7zmMtoV0hK1UtJaK	20.19	ukeJkPClDoC5ukR0Gu68CgXTY7hrIU	897
145	sLAVnWxcccLHeHj0gHrUCb9	78.06	aNZC6JhrBNmJuILRBaCFY73PMcGa6tfwUp8b	3962
150	jPGWvSMLxAbISDVq	68.00	nQYyUvIWfJ6hgTzPSAvyYwdHEnJLczKJCTdxP1jUXIw24x	487
154	VTR4aNNitjXPLEV6lK5	8.75	CF2gBT0vtI66tk0iGmTtVvF9Moqdz8LyaFT00wu6	4671
158	q3lnw3g8bdQheI	82.83	tQGWDw2xPaFXVqIT7yVVj1QRt3qetIfZtQuD	1323
174	P05zMtaVU5Whuy8LUuxz	61.79	m6V6ugVkt8XgH4048XWayofohPDhXv	789
184	QM8NGs4DVtfhB9Ce	76.45	TpNe0TPESI37xhUaHCPugq6jd70BzNPL	6810
189	yaa0yhwmcFvpXHC	96.90	oZAWcQIYyhdTziZvNr330iP5V7UikUGCKXU0l0iPaJd7NlB1mR	5621
202	TuwGUZU1j1OU43B06	68.43	Zmdoik3MWKItm9PYUoTMyd8hOKrhSRRc0W	9718
204	nosZ6lciB0tK5	46.51	tyuEBscQAe89IQ83dMDXgjc350UUBRhg72M	1700
214	SmjnRreR7eSFIlfYwf4	25.61	wTLKZizFse33a14ukLORIGINALisViYtW	3421
223	g9Prz0IoLtl3qjz	91.13	F0sY2mhQAE6cFcNj9D3QcwVhWacGWP4tFtpyg	9222
231	UHmowpNAuxbDt9iy	3.37	F9Smr0iUyDyqCdduKmxEEJaDFnuGpPqWMrI8bJjbe1vsal6b0Q	3068
233	x0T7kGyq5EzDMC	54.78	x2jbCIB59yxVulveoiyv109Lsr	6801
237	x3t4nG3aRqshTL	49.35	UYJdTwwk3cVXP9xdq1r0yFCk8	2639
238	ZTbS01G1DgogxrC	31.51	vFORIGINALmBqbc2pmXywoFKA2qC520T9	5056
250	uxy72HVppxUgK6S	43.28	gPYqle6AEG0gadZrRbsyU0JLUAXx5MpWceofNjPzcvm2WdmTL	512
252	Dgfk2YPrW0hv1Q	18.29	vHEJ6NfngNZqCg7eqU7U5z0E1zh00e14jVo7	9874
265	AkMj2HHu1Ue88WrJ3	66.05	pfaSU948Zgt2ue0MNF1K3rDELSro9XLhk1l8Y	9075
273	UViilWFA2lH00vkyt3tPzZK5	74.92	SnaT0yd1VvwvzET8xmQ3BAiRPQoQy27bm6AbJ	7938
274	WcWKKWHNJDIEySTevv	5.29	y7qHKMPtkW69Nrcca1qtg8k5MvhatKgdStKoggKk6l	9691

注：查询的条目太多了，修改查询代码来得到总条数。

```

max@ubuntu: ~
benchmarksq1=>
benchmarksq1=> SELECT COUNT(*)
benchmarksq1-> FROM bmsql_item
benchmarksq1-> WHERE 95 < ANY (
benchmarksq1(> SELECT S_QUANTITY
benchmarksq1(> FROM bmsql_stock
benchmarksq1(> WHERE S_I_ID = I_ID AND S_W_ID IN (
benchmarksq1(>
benchmarksq1(> SELECT W_ID
benchmarksq1(>
benchmarksq1(> FROM bmsql_warehouse
benchmarksq1(>
benchmarksq1(> WHERE W_TAX > 0.18
benchmarksq1(> )
benchmarksq1(> );
count
-----
20182
(1 行记录)
benchmarksq1=>

```

2. 找到主要贮备有以'SP'为开头的商品的仓库，输出仓库的所有信息。

查询代码：


```

1  SELECT *
2  FROM bmsql_warehouse
3  WHERE W_ID IN (
4      SELECT S_W_ID
5      FROM bmsql_stock
6      WHERE S_QUANTITY > 95 AND S_I_ID IN (
7          SELECT I_ID
8          FROM bmsql_item
9          WHERE I_NAME ~ '^ASP'
10     )
11 );

```

注释:

- 最内层语句：找到所有以“SP”开头的商品的ID；

```

1  SELECT I_ID
2  FROM bmsql_item
3  WHERE I_NAME ~ '^ASP'

```

- 中间层语句：从库存表中查找主要存有（即商品库存数目大于95）内层所查到的商品的仓库ID；

```

1  SELECT S_W_ID
2  FROM bmsql_stock
3  WHERE S_QUANTITY > 95 AND S_I_ID IN (...)

```

- 外层语句：根据中间层查询出的仓库ID，从仓库表输出对应仓库的全部信息。

```

1  SELECT *
2  FROM bmsql_warehouse
3  WHERE W_ID IN (...)

```

查询结果:

```

max@ubuntu: ~/Desktop
benchmarksql=> SELECT *
FROM bmsql_warehouse
WHERE W_ID IN (
SELECT S_W_ID
FROM bmsql_stock
WHERE S_QUANTITY > 95 AND S_I_ID IN (
SELECT I_ID
FROM bmsql_item
WHERE I_NAME ~ '^SP'
)
);

```

w_id	w_ytd	w_tax	w_name	w_street_1	w_street_2	w_city	w_state	w_zip
2	300000.00	0.0467	xcYtgI6jS	ABu48I4d03DcrnlH8f	cA31jXJkmgRfURang3wn	MTPXdaslFfc	GB	226411111
1	300000.00	0.1968	cDiNgvO	Ek8zsakfkbVz60o4h	fuufabxRMZylGB2	O3dnX9nEagP7kcp	GK	165211111
3	300000.00	0.0484	QPQpsHU	xGbt7kU1akPHirPld	BUVepaQMZwn68uAZ	weG9Lyyvjh5	HS	502411111
4	300000.00	0.1045	uFHJ0Xf	xn9SXnp4Ql	HZAu2EY97ZB8Ce	Wf4k4ylx0Fk1kv	XU	657011111
5	300000.00	0.0916	FrXVVzQ	jH0hITB2y6vaWg7cD	VRaoqG1C9ZjWDWbL1	iUmr9CdMuSxRlzs5	AL	764111111
6	300000.00	0.1221	OdccLwtgK	fjUTAJWZovMI2Jt6jHU	ylolIHldxfo	nx129y82Ra31YfANQVb0	UU	696411111
9	300000.00	0.0700	CFETeuZj9G	BIX0sQn6QLW7H1wG5lis	V2H7yQzCKELoL	cFi0a3R0H3zDg	MM	660511111
10	300000.00	0.0656	YU78vml	M88HzenijH	y1D63KE57xhkWP1LF	apJWVDC71M	ZH	230411111
12	300000.00	0.0538	vX9uEDP8I	W4W8mRocARQC21jBgo	HvxqHoagQJvHysR	eSg4UPgG6ZpMyER	ML	201611111
13	300000.00	0.1624	Rprpr3	CPfwtttdpiZK0uaZGxb0	VdfE4o55WuTiCyVr	iJPgvsWlH5ivs9	KX	778011111
14	300000.00	0.0753	gEAPZ4	tqCTCttY2zCY92	qyeCQ00s1P9Kvym1M	ocBPgEAUWdhFvNKGd9	GQ	166511111
16	300000.00	0.1226	SgmSetepNe	iSMNHawxtJA	X2dpj0Xl9g1Mzoq0wlp0	fxer1C1hCdvw0kAw5s0	FL	223811111
17	300000.00	0.0315	fkBR01pUKa	FujbaYfLHdsJmbX	wedPpBRdWpBgQPsTn	ZTe6n3eErJrKnsf7s	OU	341011111
19	300000.00	0.0026	iVxZr3hnt	Urm4dizowxY5BmA	zggRyP7C6xo2ypXxMtP	ZlonpdIjpDraL	HZ	822711111
21	300000.00	0.0336	pvJjHlk	ShzvanurEyrnc8jDeVF	iq8kpYnTPi0tie	rmDlQcNH0UWkop0r	WX	080711111
22	300000.00	0.1123	NH6yqlaupS	JnzyVHdtcvf	aGhyH1C3I5n22	sfb1qFC7NIuhnkXIdz	YR	531811111
23	300000.00	0.1069	DB51Yp	HVOTNBMK3BgtG	xVVRsccgp3VM7y	YzvBeLwRwdNm1vCv0	VS	819611111
25	300000.00	0.1768	TbPZQ5	hUj52drpri2Ru5vqgK	FIfcy5mb58343g	NyyczAIPWGWREja15	PS	805911111
26	300000.00	0.0543	prJHzwNYHL	NMfi1pHHde	Of1fmKdqm1c	UuehxZoakAHw4sk0	XU	210211111
27	300000.00	0.0866	oRSfgM	GCJ28BrtB1ron4G27PM	cd0uQl0uLqq15zqStc	V9AHT9myLnoIhnD9S0Lu	UK	289511111
28	300000.00	0.0192	HVYg10	PZHKU1H1wbG	stcgV8nMspNDRr7DHC	qxR7k4CxGN3qLzJ55	SQ	633211111
29	300000.00	0.1823	Z9M5Jnnt	uZ67QAYRII	B2mQEL9g040h96	YBurkRwDs00rgCX6B4	NW	863911111
30	300000.00	0.1579	La6GMDKD	aQbgb5L3NcQVz	TLLjTm8nZ1k	o1NsfduaN97	NG	219311111
31	300000.00	0.1245	VEgefQZP9S	TE1Gqk41VbCA	Dqc5XLMWhlk	Q8XGfSsqanZ	HH	965011111
32	300000.00	0.1176	JbehRdMIqk	SsbEArvzdRLC	IpbV2cTGSCBe1VnH2Cw	IOX7Ykqyqn	MX	839211111
34	300000.00	0.0013	LVFM140duD	QkBhrCkhMVV25Zt	vxdqdtZzuqw3	ItIxmSUXxq1nE	PX	736711111
35	300000.00	0.0555	rmwzJgv	D5g8S1C4DBPMT3	HTRx7GoJFUG4xCQA	wcdLwIsje9MhWdaKVH	RC	381411111
36	300000.00	0.1685	mdqAc4RnL	oeTtJVK7vG18gV7JW	o2EIoISBgIuQNmN6m	ou0fkQ8WK2sUdL	FZ	735411111
37	300000.00	0.0638	eZqfLppYw	A9NLBq0d1ReaiHioUUQk	yhyP7WuXl08gzP2	l009E1H5wLkGyR	XJ	163311111
38	300000.00	0.1158	JGV2LVhGw	WNAw4GpyPatZf3WZHH	Igis1m8PG2qwsX30V9s9	XAO52Mf0zKC8	BJ	684711111
39	300000.00	0.1951	DVHffsVbt	RYvE915jHV515	JYfQgouov1TzoyvPsX	zP6OwxVkvQ00QtwF	RM	193411111
40	300000.00	0.1108	xD4V3Z	o2UTZ9LwoGpXp	tkRyW4FH1EDXo9l	cRoAnsNuGRBFzyrx3	EF	551911111

```

(32 行记录)
benchmarksql=>

```

五、综合题：

1. 找到以'SP'为开头，且在所有仓库储存的平均数量大于50的商品的全部信息。

查询代码：

```

1 SELECT *
2 FROM bmsql_item
3 WHERE I_NAME ~ '^ASP' AND 50 < (
4     SELECT AVG(S_QUANTITY)
5     FROM bmsql_stock
6     WHERE I_ID = S_I_ID
7 );

```

注解：

- 内层语句：通过查询库存表，计算外层指定的商品在所有仓库中的平均库存；

```

1 SELECT AVG(S_QUANTITY)
2 FROM bmsql_stock
3 WHERE I_ID = S_I_ID

```

- 外层语句：输出所有满足名称以SP开头，平均库存大于50的商品的全部信息。

```

1 SELECT *
2 FROM bmsql_item
3 WHERE I_NAME ~ '^ASP' AND 50 < (...);

```

查询结果：（共26条记录）

```

max@ubuntu: ~/Desktop
benchmarksql=>
benchmarksql=> SELECT *
benchmarksql=> FROM bmsql_item
benchmarksql=> WHERE I_NAME ~ '^SP' AND 50 < (
benchmarksql=> SELECT AVG(S_QUANTITY)
benchmarksql=> FROM bmsql_stock
benchmarksql=> WHERE I_ID = S_I_ID
benchmarksql(> );

```

i_id	i_name	i_price	i_data	i_im_id
733	SPHbekq10CqgjR0nmNCH	31.68	sDIH2a2E2m66rblQmqjKcGdmLcQDpETVr0eaYTqEVM6R	5517
5256	SPTedrytQkj2RpnSawRjVvVb	29.11	y0VcymKVXb5QKCTL3nfiJypT48WxJNjGG4M0jEKAYRQ1nJxip	9074
6592	SPaP6Sk0idfMANn	38.13	xQ1WvHNBc6ItKW8BLsyPvy3bdB4CEeHZ	9
12132	SPjHMuA9cIv7N2o14vXojlD	41.49	YEYDF8mdg4Qz9GYFhP051zzL7Jynz3QIvawEMEsfADw7WW8	331
28167	SPqo5iLErPEu3IZmJgT	8.71	c4JW7D6ozj0C0N0Us1VLwVnFLk8eUwR0jr4F10Cx5FA6iJd8d	1678
30884	SPqLXleejD8gSalFudbej	73.85	ds0yzPVrqdddluwvCvBTKWoGnw2yA	9751
34144	SPSD9CA4KFIEs37	43.64	wgyrePRY9kwxkXVA8aushkYSzqSS	62
35474	SPECMTBBFjynjsDlzsZj5lnE	86.50	WjrSEBgH48e1Vyt4kmICIC7RvleanF	5599
36047	SPUWhG42sf3KVGH9ne	71.99	LPAQtM00uIQcpb5zLSUF548GTAow4uApXtPGJJJoPH7	3148
37079	SPEjPWLOALG44a	28.46	yqSAGSsn0IecXEFvd4Vkcq0dXm57mYFDXV8J1PMhsKlpCYbaz3	5303
45686	SPp5wahD8C8fQadp0u0zJvyz	88.25	Xe3DRLGV4uSNbwU1YA4Jsyt4EBQu4iXFqIRv	4816
48408	SPEv2vLFp7BeuygRc5AB	62.26	qDyXELJbyEdU59tppLsp0TdmR5jRNybcSYsDt6	6783
52170	SP4QnpSHqOdHYUfDj	65.81	og7tS5A1N1QITeJmUYs775gZnVLMHbGHCy8l8hfo	40
52282	SPInrGf0YvT72BkCh8	18.59	x04x07KL8l3e2JKVIIHJTHQuubytmfXd8f6WWddJgJVZZix4	7542
54538	SPucXDTH1p8upbqkADp	26.35	CKL156R4pjA0A80z0VLXY7TQmrwMjwobj7u60PM5	3575
60363	SPRr8nVQOLFumWX67wmyTs	33.71	cN3Y9vNft33xvLzNeXyqagpWlj	5480
64584	SPpKK3jqEKR0j5FYbaarOgp	39.06	J4qjpfcmDUtMnXovpPp6bcdL8T2u	9605
66163	SPTtbvnrZdlwpFyFPY	72.57	OdnrZ7bE54UZmnhrGxUdfhVlth	2785
67123	SP0rpY5KJ04ELYKR6Zsh	18.81	PE5YwZiYQW9l6gPYC5caS9raGbVuH04dmr	2182
67963	SPEyDA6lfht5rVhNa	91.04	v4YrokznZayADD3QJGz5TXCJ41ro	4448
78396	SPkdcX4GP1oJhn5wEyF	99.91	pAdx019HhQWBy6snwGG0w5FKYMBqSmx7b6Vm	623
86850	SP1CRG9VoHxJ28tqZpF1R	48.08	BARuonXLVjCjPw5ezbLZNhsVXnMhFedgYVh3J4wka	7641
87365	SPsmZj0Wf4ET1kBKdy	62.31	cm119bSsywy2nllPaORIGINALFaEkLg	7275
93421	SPtekDHFUpK7zQqeEDNr	7.78	sVGzgKZd5l7h3NQswdjalnXasgATvpccfKrZrF	5567
95913	SPvsMbebkMZK5u0j6Ck78	56.96	Sc1r0dq1Z2LViYux7vxQWkaq5x0Y	3050
97957	SPW5sARsXhQIbg7	72.52	Wy1coE20bQpMW0kNrMVH4YTXt5TgPRWlsgueYoFR8edkR	9083

```

(26 行记录)
benchmarksql=>
benchmarksql=>
benchmarksql=>
benchmarksql=>

```

2. 找到所有满足条件的仓库的编号(w_id): 该仓库在所有地区的销售税都小于0.15。

查询代码:

```

1 SELECT W_ID
2 FROM bmsql_warehouse
3 WHERE 0.15 > ALL (
4     SELECT D_TAX
5     FROM bmsql_district
6     WHERE D_W_ID = W_ID
7 );

```

注解:

- 内层语句: 从区域表中查询外层指定仓库的售卖街区的销售税;

```

1 SELECT D_TAX
2 FROM bmsql_district
3 WHERE D_W_ID = W_ID

```

- 外层语句: 从仓库表中查询满足其所所有销售街区的销售税都小于0.15的仓库的仓库ID。

```

1 SELECT W_ID
2 FROM bmsql_warehouse
3 WHERE 0.15 > ALL (...);

```

查询结果: (共1条记录)

```
max@ubuntu: ~/Desktop
benchmarksql=>
benchmarksql=> SELECT W_ID
benchmarksql-> FROM bmsql_warehouse
benchmarksql-> WHERE 0.15 > ALL (
benchmarksql(> SELECT D_TAX
benchmarksql(> FROM bmsql_district
benchmarksql(> WHERE D_W_ID = W_ID
benchmarksql(> );
  w_id
  -----
      36
(1 行记录)

benchmarksql=>
benchmarksql=>
```

3. 统计五.2中的仓库主要储备的商品数量，价格平均值，输出列为w_id, number, avg_price。

查询代码：

```
1 SELECT S_W_ID AS w_id, COUNT(I_ID) AS number, AVG(I_PRICE) AS avg_price
2 FROM bmsql_stock, bmsql_item
3 WHERE I_ID = S_I_ID AND S_W_ID IN (
4     SELECT W_ID
5     FROM bmsql_warehouse
6     WHERE 0.15 > ALL (
7         SELECT D_TAX
8         FROM bmsql_district
9         WHERE D_W_ID = W_ID
10    )
11 ) AND S_QUANTITY > 95
12 GROUP BY S_W_ID;
```

注解：

- 内层语句：查询并返回所有中间层指定的仓库的对应销售街区的销售税；

```
1 SELECT D_TAX
2 FROM bmsql_district
3 WHERE D_W_ID = W_ID
```

- 中间层语句：使用 ALL 来筛选出所有销售区域的销售税都小于0.15的仓库，返回仓库的ID；

```
1 SELECT W_ID
2 FROM bmsql_warehouse
3 WHERE 0.15 > ALL (...)
```

- 外层语句：

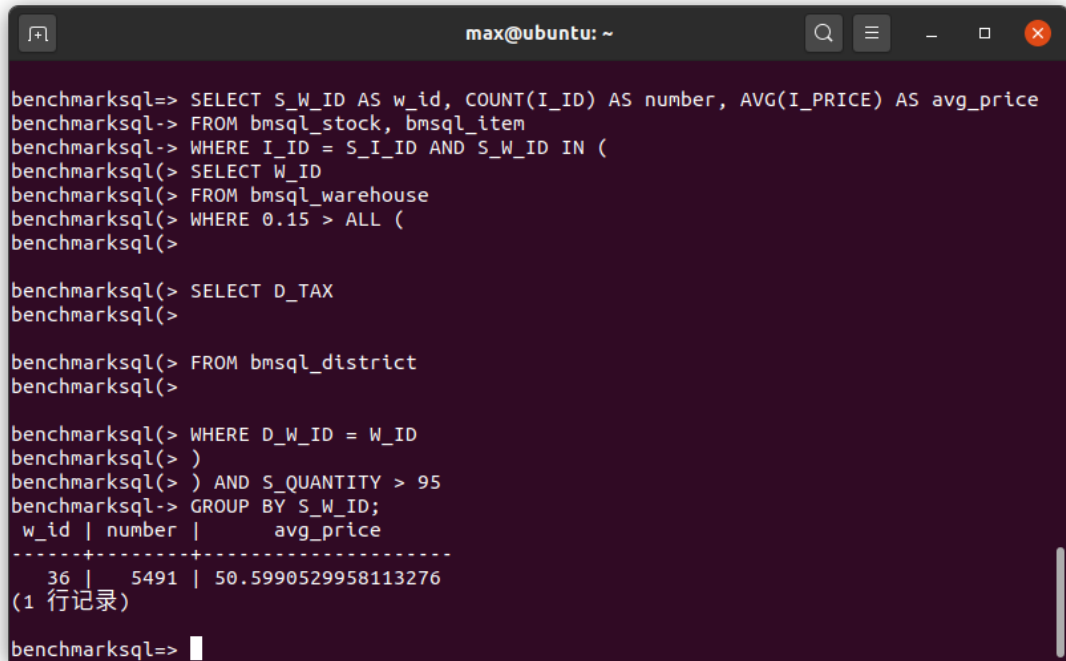
- 将库存表与商品表做条件连接，连接条件有3个：

- I_ID = S_I_ID：商品ID匹配；
- S_W_ID IN (...)：仓库ID为中间层查询结果；
- S_QUANTITY > 95：该商品在对应仓库中的库存大于95（满足主要储备）；

- 分别使用数学函数 `COUNT` 和 `AVG` 来计算商品数目和平均价格，分组（`GROUP BY`）依据为仓库ID；
- 最后使用 `AS` 为各输出列重命名。

```
1 SELECT S_W_ID AS w_id, COUNT(I_ID) AS number, AVG(I_PRICE) AS  
   avg_price  
2 FROM bmsql_stock, bmsql_item  
3 WHERE I_ID = S_I_ID AND S_W_ID IN (...) AND S_QUANTITY > 95  
4 GROUP BY S_W_ID;
```

查询结果：共1条记录



```
max@ubuntu: ~  
benchmarksql=> SELECT S_W_ID AS w_id, COUNT(I_ID) AS number, AVG(I_PRICE) AS avg_price  
benchmarksql-> FROM bmsql_stock, bmsql_item  
benchmarksql-> WHERE I_ID = S_I_ID AND S_W_ID IN (  
benchmarksql-> SELECT W_ID  
benchmarksql-> FROM bmsql_warehouse  
benchmarksql-> WHERE 0.15 > ALL (  
benchmarksql->  
  
benchmarksql-> SELECT D_TAX  
benchmarksql->  
  
benchmarksql-> FROM bmsql_district  
benchmarksql->  
  
benchmarksql-> WHERE D_W_ID = W_ID  
benchmarksql-> )  
benchmarksql-> ) AND S_QUANTITY > 95  
benchmarksql-> GROUP BY S_W_ID;  
w_id | number |      avg_price  
-----+-----+-----  
36 | 5491 | 50.5990529958113276  
(1 行记录)  
benchmarksql=> 
```