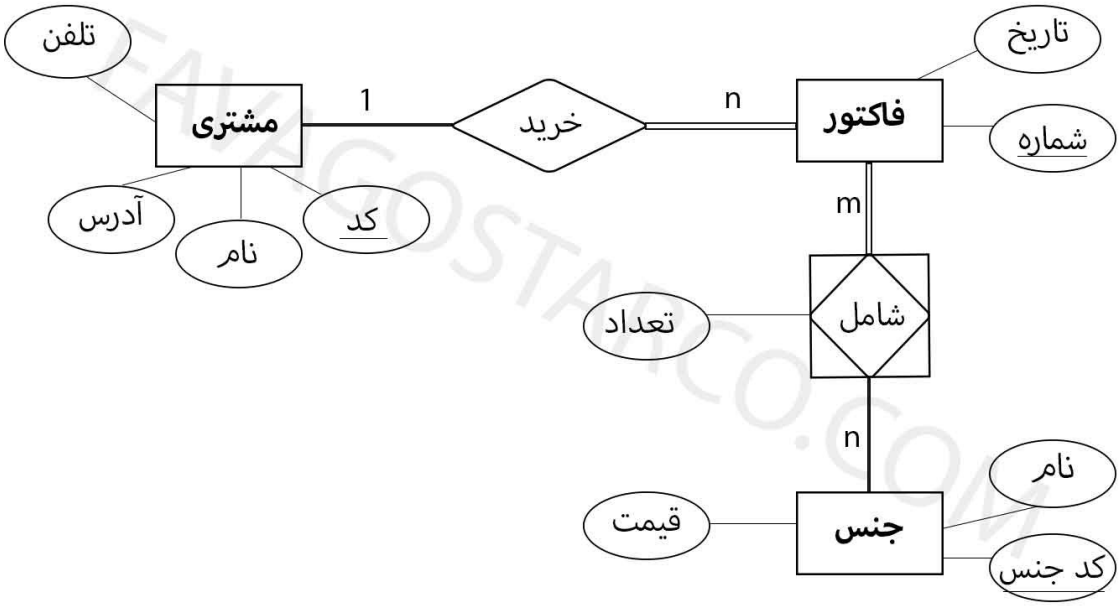
سروش رستم نژاد، پرهام شیروانیون
دانشگاه گیلان
درس آز پایگاه داده
ترم ۱۴۰۱۰۲
استاد یوسفی
تمرین ۱

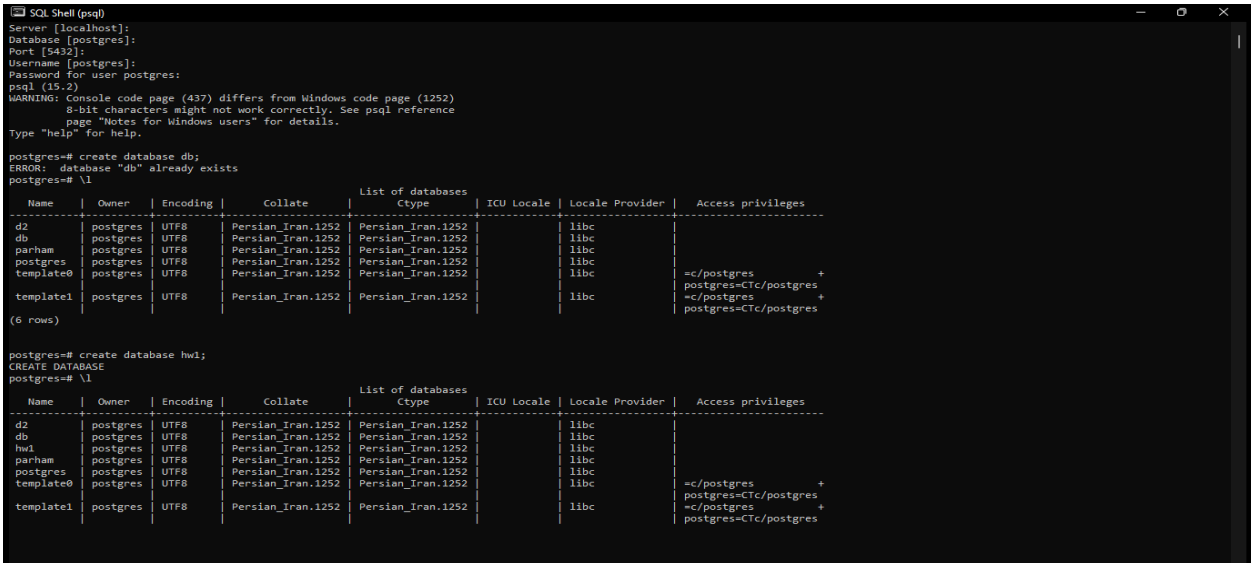
الف)



customer(<u>code</u>, name, address, tel)
factor(<u>no</u>, date,**costumerCode**[1])
commodity(<u>code</u>, name, price)
include(**factorNo**[2], **commodityCode**[3], number)

[1] (foreign key)
[2] (foreign key)
[3] (foreign key)

ب) ابتدا دیتابیس مورد نظر را می سازیم که با استفاده از دستور: create database database_name است و سپس با \L لیست دیتابیس ها را می بینیم.



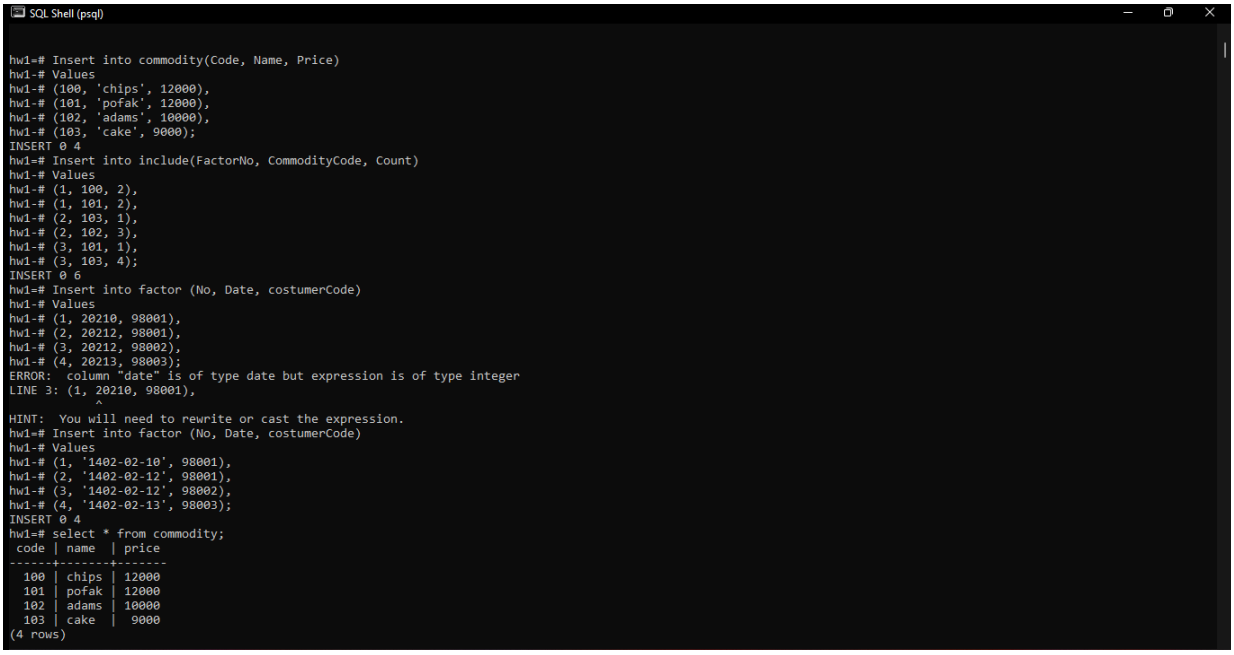
ج)

```
Create table customer(
        Code int primary key,
        Name varchar(15) not null,
        Address varchar(150),
        Tel int);

Create table factor(
        No int primary key,
        Date int,
        costumerCode int not null);


Create table commodity(
        Code int primary key,
        Name varchar(15) not null,
        Price int not null);


Create table include(
        FactorNo int not null,
        CommodityCode int,
        Count int check(Count > 0));
```

د)

```
Insert into customer (Code, Name, Address, Tel)
Values
        (98001, ‘parham’, ‘rasht’, 0133322),
        (98002, ‘soroush’, ‘lahijan’, 0131122),
        (98003, ‘ali’, ‘rasht’, 0133355);
Insert into factor (No, Date, costumerCode)
Values
        (1, '1402-02-10', 98001),
        (2, '1402-02-12', 98001),
        (3, '1402-02-12', 98002),
        (4, '1402-02-13', 98003);
Insert into commodity(Code, Name, Price)
Values
        (100, ‘chips’, 12000),
        (101, ‘pofak’, 12000),
        (102, ‘adams’, 10000),
        (103, ‘cake’, 9000);
Insert into include(FactorNo, CommodityCode, Count)
Values
        (1, 100, 2),
        (1, 101, 2),
        (2, 103, 1),
        (2, 102, 3),
        (3, 101, 1),
        (3, 103, 4);
```

تصاویر مربوطه در صفحه بعد 👇

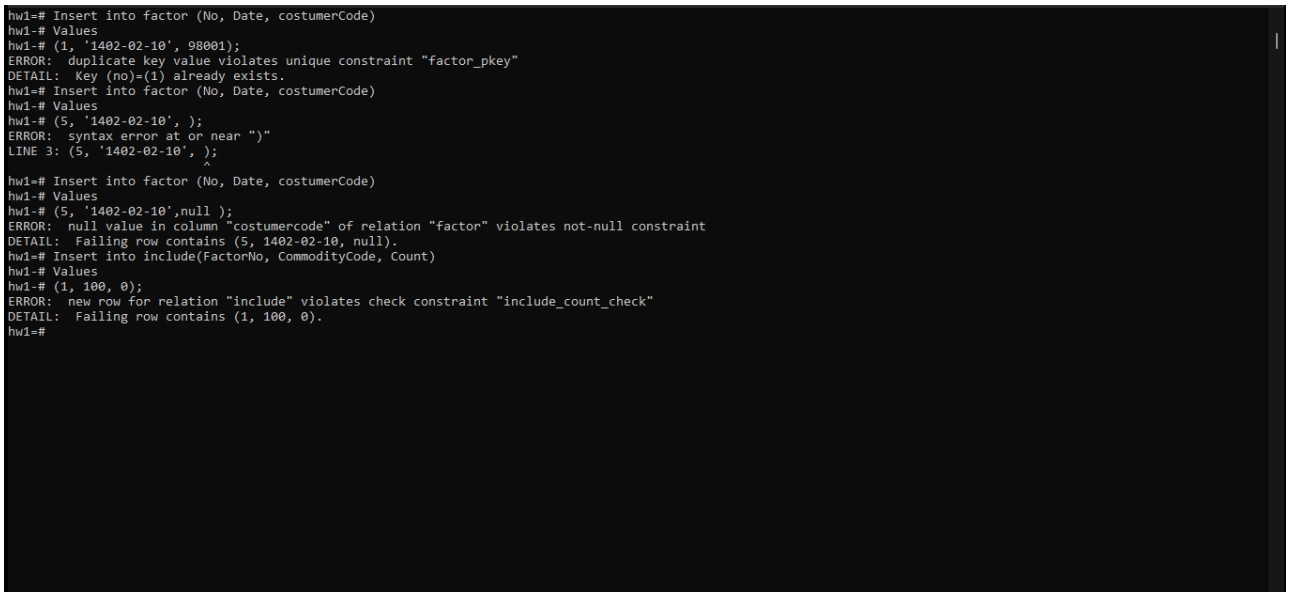

SQL Shell (psql)
hw1=# Insert into commodity(Code, Name, Price)
hw1-# Values
hw1-# (100, 'chips', 12000),
hw1-# (101, 'pofak', 12000),
hw1-# (102, 'adams', 10000),
hw1-# (103, 'cake', 9000);
INSERT 0 4
hw1=# Insert into include(FactorNo, CommodityCode, Count)
hw1-# Values
hw1-# (1, 100, 2),
hw1-# (1, 101, 2),
hw1-# (2, 103, 1),
hw1-# (2, 102, 3),
hw1-# (3, 101, 1),
hw1-# (3, 103, 4);
INSERT 0 6
hw1=# Insert into factor (No, Date, costumerCode)
hw1-# Values
hw1-# (1, 20210, 98001),
hw1-# (2, 20212, 98001),
hw1-# (3, 20212, 98002),
hw1-# (4, 20213, 98003);
ERROR: column "date" is of type date but expression is of type integer
LINE 3: (1, 20210, 98001),
^
HINT: You will need to rewrite or cast the expression.
hw1=# Insert into factor (No, Date, costumerCode)
hw1-# Values
hw1-# (1, '1402-02-10', 98001),
hw1-# (2, '1402-02-12', 98001),
hw1-# (3, '1402-02-12', 98002),
hw1-# (4, '1402-02-13', 98003);
INSERT 0 4
hw1=# select * from commodity;
code | name | price
100 | chips | 12000
101 | pofak | 12000
102 | adams | 10000
103 | cake | 9000
(4 rows)


SQL Shell (psql)
code | name | price
100 | chips | 12000
101 | pofak | 12000
102 | adams | 10000
103 | cake | 9000
(4 rows)
hw1=# select * from customer;
code | name | address | tel
98001 | parham | rasht | 133322
98002 | soroush | lahijan | 131122
98003 | ali | rasht | 133355
(3 rows)
hw1=# select * from factor;
no | date | costumercode
1 | 1402-02-10 | 98001
2 | 1402-02-12 | 98001
3 | 1402-02-12 | 98002
4 | 1402-02-13 | 98003
(4 rows)
hw1=# select * from include;
factorno | commoditycode | count
1 | 100 | 2
1 | 101 | 2
2 | 103 | 1
2 | 102 | 3
3 | 101 | 1
3 | 103 | 4
(6 rows)
hw1=#

ه) اگر محدودیت های برنامه را زیر پا بگذاریم اخطار می گیریم که در عکس پایین این اخطار را می بینیم :

شروط را با مثال های زیر نقض میکنیم.

factor.no(pkey)=duplicate
factor.costumerCode=null
Include.count=0



ن) برای بازیابی سطر های جدول در postgresql می توان از دستور

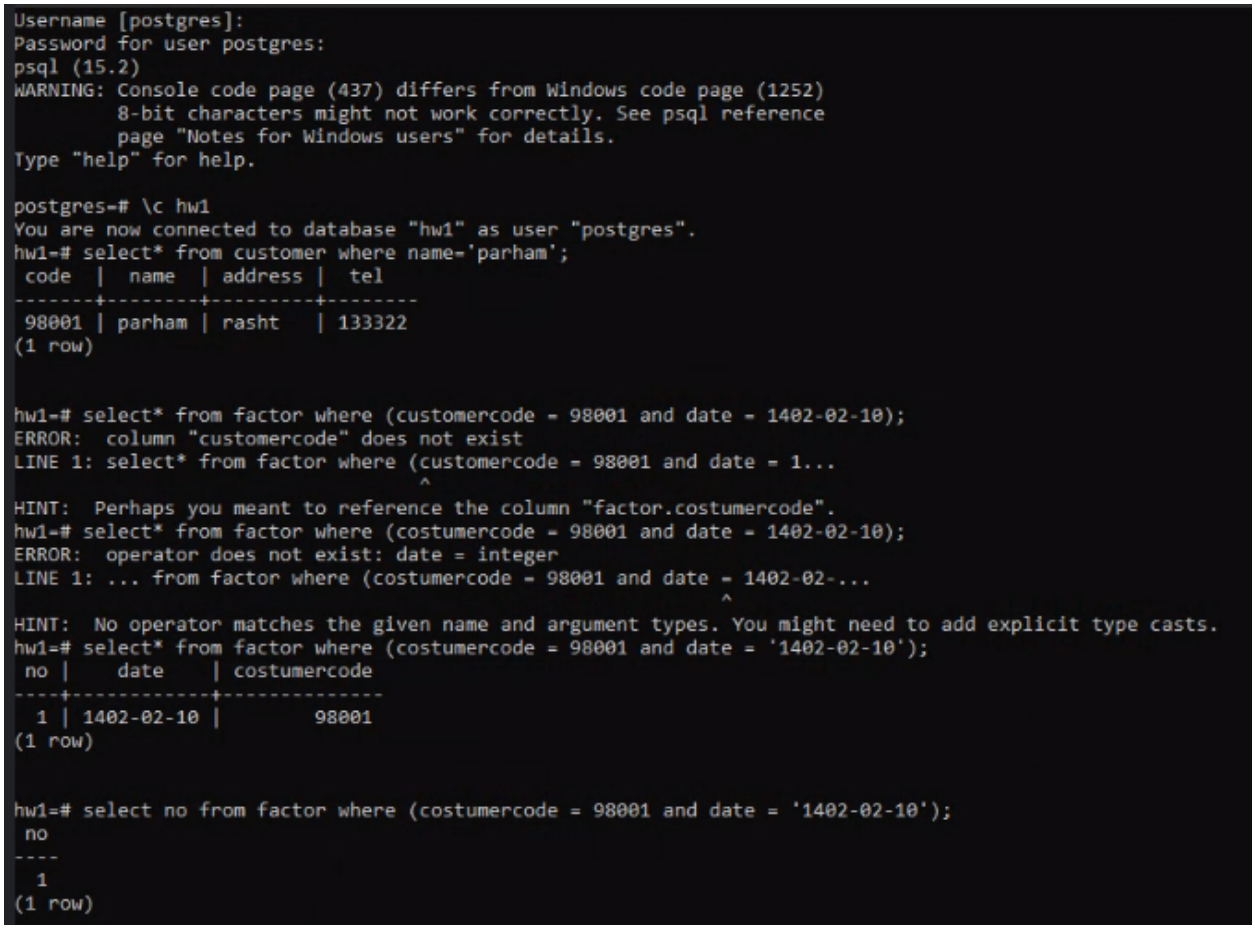
SELECT * FROM db-name;

استفاده کرد که تمامی سطر های جدول را مشاهده کرد. برای بازیابی بخشی خاص از جدول می توانیم به جای * از صفت مورد نیاز استفاده کنیم.

در تصاویر قسمت (د) انجام شده است.

و) در این قسمت از تمرین ما باید با استفاده از دستور select شرط های پیچیده تری را امتحان کنیم. برای مثال :

```
SELECT * FROM table_name WHERE (condition1 AND condition2 AND condition3) OR (condition4 AND condition5);
```



```
Username [postgres]:
Password for user postgres:
psql (15.2)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \c hw1
You are now connected to database "hw1" as user "postgres".
hw1=# select* from customer where name='parham';
 code  |  name  | address |  tel
-------+--------+---------+--------
 98001 | parham | rasht   | 133322
(1 row)


hw1=# select* from factor where (customercode = 98001 and date = 1402-02-10);
ERROR:  column "customercode" does not exist
LINE 1: select* from factor where (customercode = 98001 and date = 1...
                                   ^
HINT:  Perhaps you meant to reference the column "factor.costumercode".
hw1=# select* from factor where (costumercode = 98001 and date = 1402-02-10);
ERROR:  operator does not exist: date = integer
LINE 1: ... from factor where (costumercode = 98001 and date = 1402-02-...
                                                             ^
HINT:  No operator matches the given name and argument types. You might need to add explicit type casts.
hw1=# select* from factor where (costumercode = 98001 and date = '1402-02-10');
 no |    date    | costumercode
----+------------+--------------
  1 | 1402-02-10 |        98001
(1 row)


hw1=# select no from factor where (costumercode = 98001 and date = '1402-02-10');
 no
----
  1
(1 row)
```

ص) با استفاده از سایت postgreSQL داریم :

```
SELECT format('Hello %s', 'World');
Result: Hello World

SELECT format('Testing %s, %s, %s, %%', 'one', 'two', 'three');
Result: Testing one, two, three, %

SELECT format('INSERT INTO %I VALUES(%L)', 'Foo bar', E'O\'Reilly');
Result: INSERT INTO "Foo bar" VALUES('O''Reilly')

SELECT format('INSERT INTO %I VALUES(%L)', 'locations', 'C:\Program Files');
Result: INSERT INTO locations VALUES('C:\Program Files')
```

```
SELECT format('|%10s|', 'foo');
Result: |       foo|

SELECT format('|%-10s|', 'foo');
Result: |foo       |

SELECT format('|%*s|', 10, 'foo');
Result: |       foo|

SELECT format('|%*s|', -10, 'foo');
Result: |foo       |

SELECT format('|%-*s|', 10, 'foo');
Result: |foo       |

SELECT format('|%-*s|', -10, 'foo');
Result: |foo       |
```

```
SELECT format('Testing %3$s, %2$s, %1$s', 'one', 'two', 'three');
Result: Testing three, two, one

SELECT format('|%*2$s|', 'foo', 10, 'bar');
Result: |       bar|

SELECT format('|%1$*2$s|', 'foo', 10, 'bar');
Result: |       foo|
```

```
SELECT format('Testing %3$s, %2$s, %s', 'one', 'two', 'three');
Result: Testing three, two, three
```

```
hw1=# select format('hello %s', 'world');
   format
-------------
 hello world
(1 row)


hw1=# select format('testing %s, %s, %s, %%', 'one', 'two', 'three');
           format
----------------------------
 testing one, two, three, %
(1 row)


hw1=# select format('|%10s|', 'ho ho ho');
    format
--------------
 |  ho ho ho|
(1 row)


hw1=# select format('|%-10s|', 'ho ho ho');
    format
--------------
 |ho ho ho  |
(1 row)
```

ض) در این قسمت ما از تابع رشته ای format در شرط where استفاده کردیم و ستون name را تست کردیم.

```
hw1=# select* from customer where name=format('|%6s|','parham');
 code | name | address | tel
------+------+---------+-----
(0 rows)


hw1=# select* from customer where name=format('parham');
 code  |  name  | address |  tel
-------+--------+---------+--------
 98001 | parham | rasht   | 133322
(1 row)
```

ط) برای دیدن رکورد های جدول توسط دستور select کافی است تا دستور :

```
SELECT * FROM db-name;
```

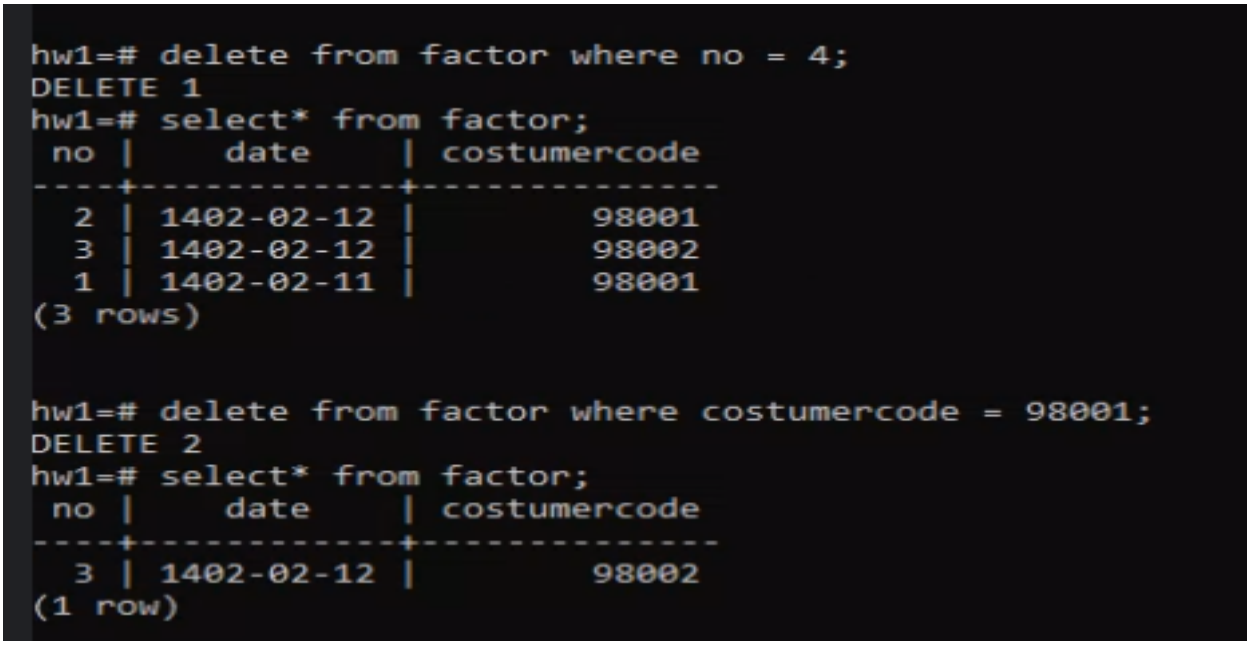
را بزنیم و سپس با استفاده از دستور update:

```
UPDATE table_name
SET column1 = value1
,column2 = value2
...
WHERE condition;
```

```
hw1=# select* from factor;
 no |    date    | costumercode
----+------------+--------------
  1 | 1402-02-10 |        98001
  2 | 1402-02-12 |        98001
  3 | 1402-02-12 |        98002
  4 | 1402-02-13 |        98003
(4 rows)


hw1=# update factor, set no=1, date='1402-02-11', costumercode=98001 where no=1;
ERROR:  syntax error at or near ","
LINE 1: update factor, set no=1, date='1402-02-11', costumercode=980...
                     ^
hw1=# update factor set no=1, date='1402-02-11', costumercode=98001 where no=1;
UPDATE 1
hw1=# select* from factor;
 no |    date    | costumercode
----+------------+--------------
  2 | 1402-02-12 |        98001
  3 | 1402-02-12 |        98002
  4 | 1402-02-13 |        98003
  1 | 1402-02-11 |        98001
(4 rows)


hw1=# update factor, set no=4, date='1402-02-14', costumercode=98001 where no=4;
ERROR:  syntax error at or near ","
LINE 1: update factor, set no=4, date='1402-02-14', costumercode=980...
                     ^
hw1=# update factor set no=4, date='1402-02-14', costumercode=98001 where no=4;
UPDATE 1
hw1=# select* from factor;
 no |    date    | costumercode
----+------------+--------------
  2 | 1402-02-12 |        98001
  3 | 1402-02-12 |        98002
  1 | 1402-02-11 |        98001
  4 | 1402-02-14 |        98001
(4 rows)
```

ظ) برای حذف یک یا چندین سطر از جدول ما از دستور Delete استفاده می کنیم که اینگونه نوشته می شود:

DELETE FROM table_name WHERE condition;



ع) در PostgreSQL، یک tablespace به عنوان نقشه اسم منطقی برای یک موقعیت فیزیکی در دیسک به کار می رود. در واقع یک مکان در دیسک است که تمامی شی های دیتابیس ها مانند index ها و جدول ها را ذخیره میکند. فضاهای جدولی در PostgreSQL به مدیران پایگاه داده اجازه می دهد تا مکان هایی را در سیستم فایل تعریف کنند که در آن فایل های نشان دهنده اشیاء پایگاه داده می توانند ذخیره شوند. پس از ایجاد، فضای جدول می تواند با نام در هنگام ایجاد اشیاء پایگاه داده ارجاع داده شود.