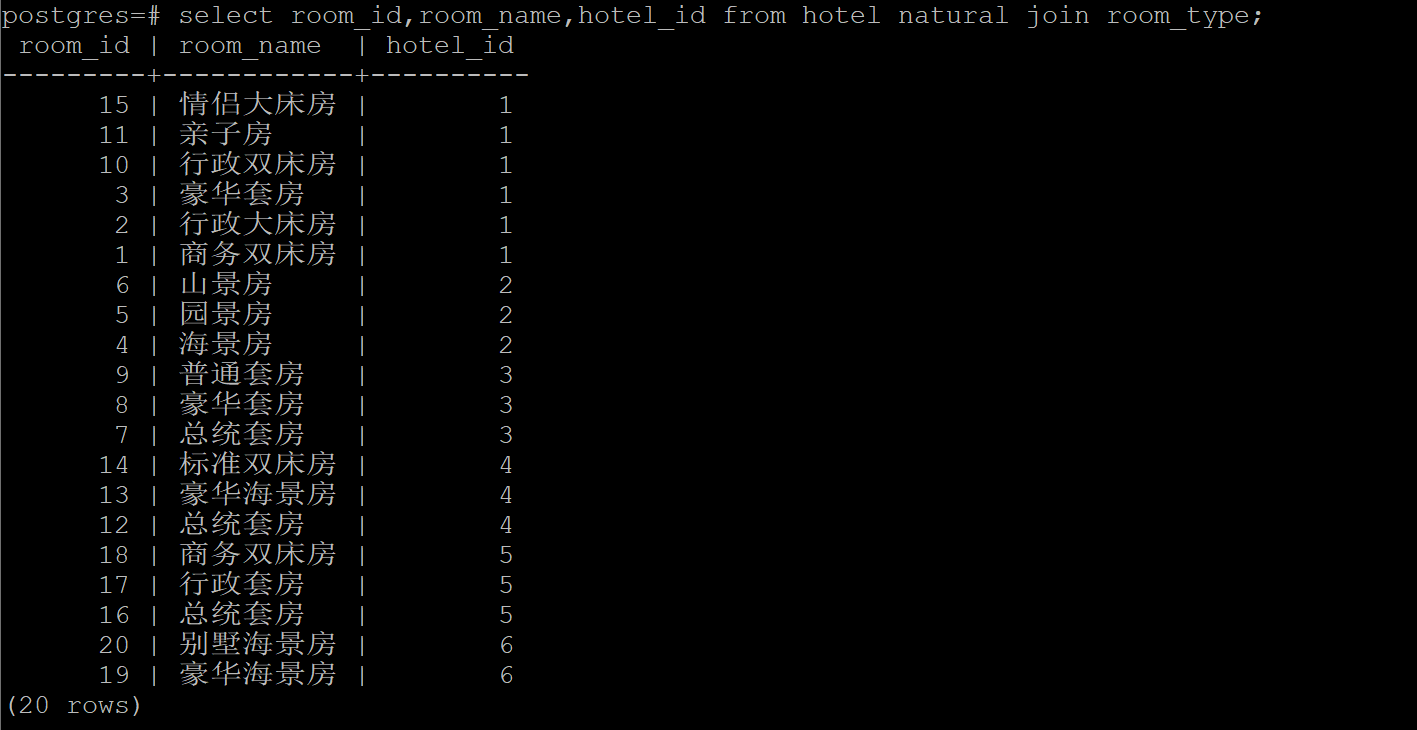
实验报告

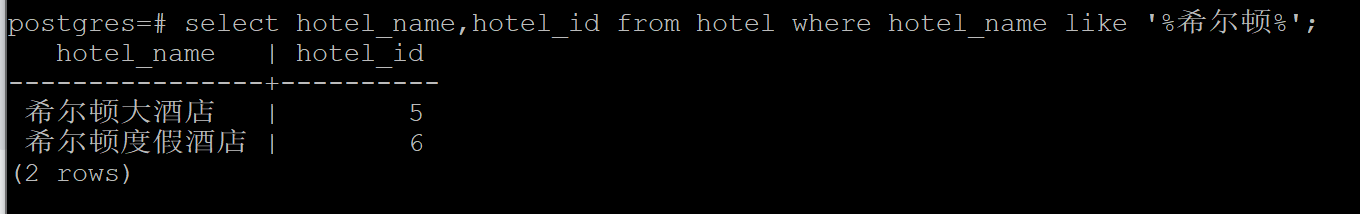
1.

select room\_id,room\_name,hotel\_id from hotel natural join room\_type;



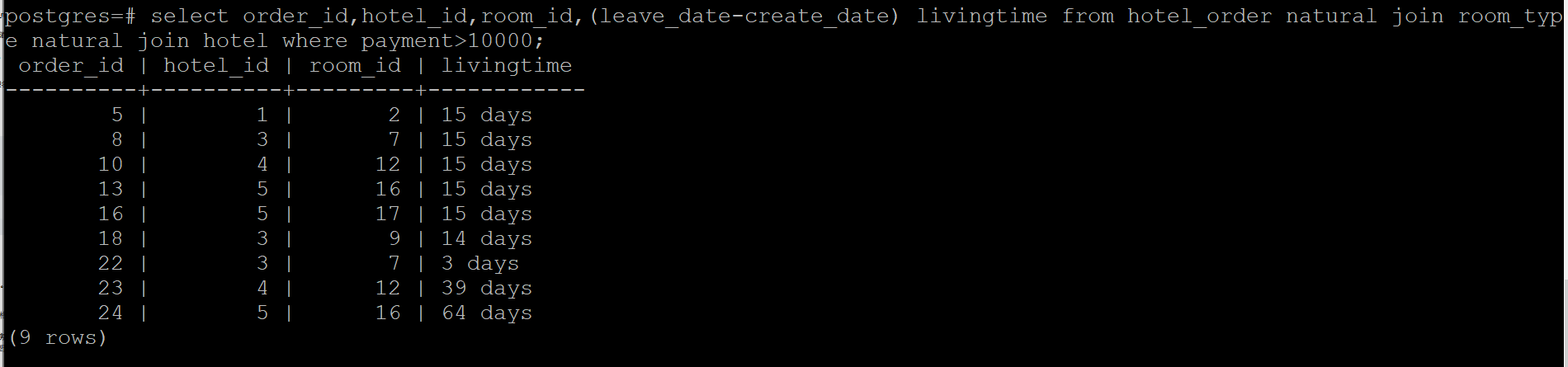
2.

select hotel\_name,hotel\_id from hotel where hotel\_name like '%希尔顿%';



3.

Select order\_id,hotel\_id,room\_id,(leave\_date-create\_date) livingtime from hotel\_order natural join hotel where payment>10000;



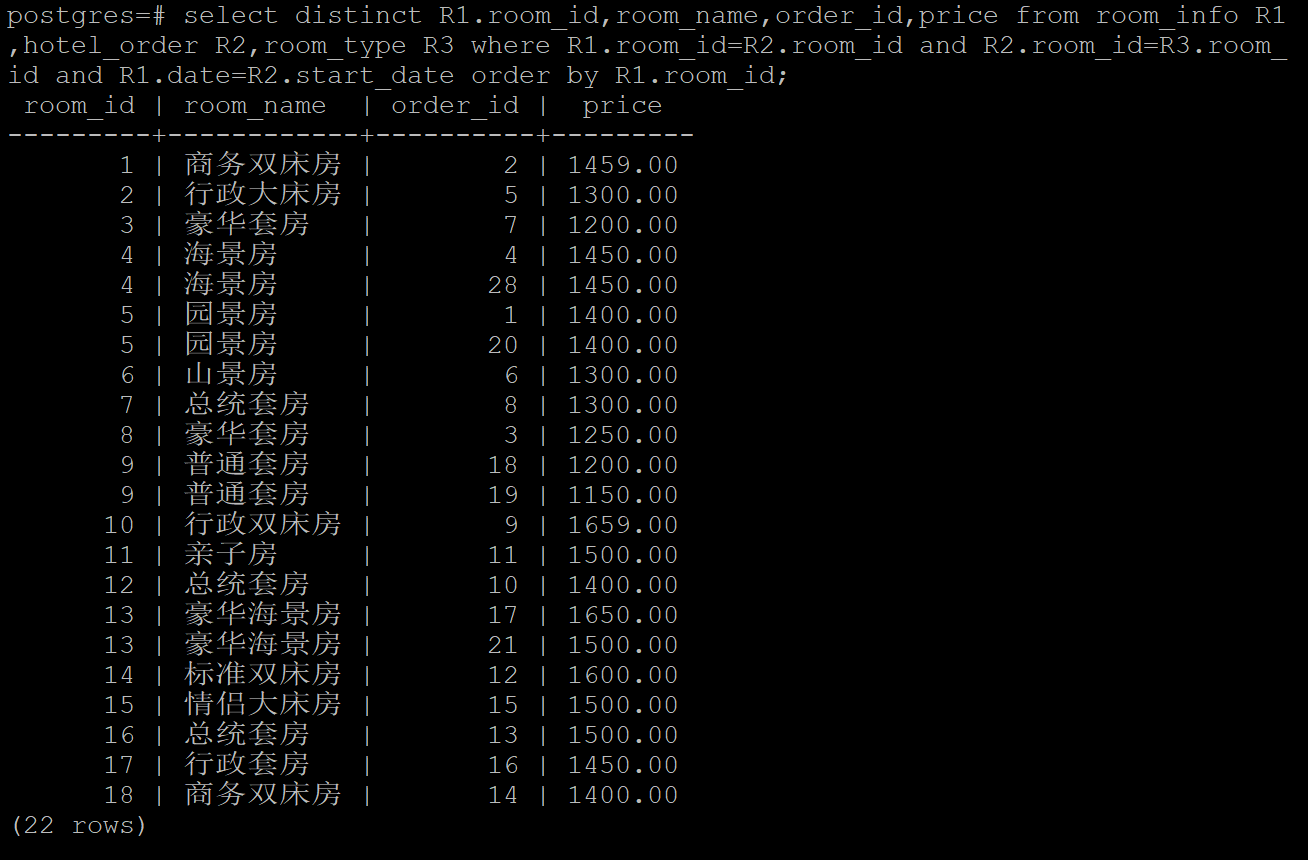
4.

select distinct R1.room\_id,room\_name,order\_id,price from room\_info R1,hotel\_order R2,room\_type R3 where R1.room\_id=R2.room\_id and R2.room\_id=R3.room\_id and R1.date=R2.start\_date order by R1.room\_id;

（假设连续多天的订单都是按照开房当天的房价支付的）

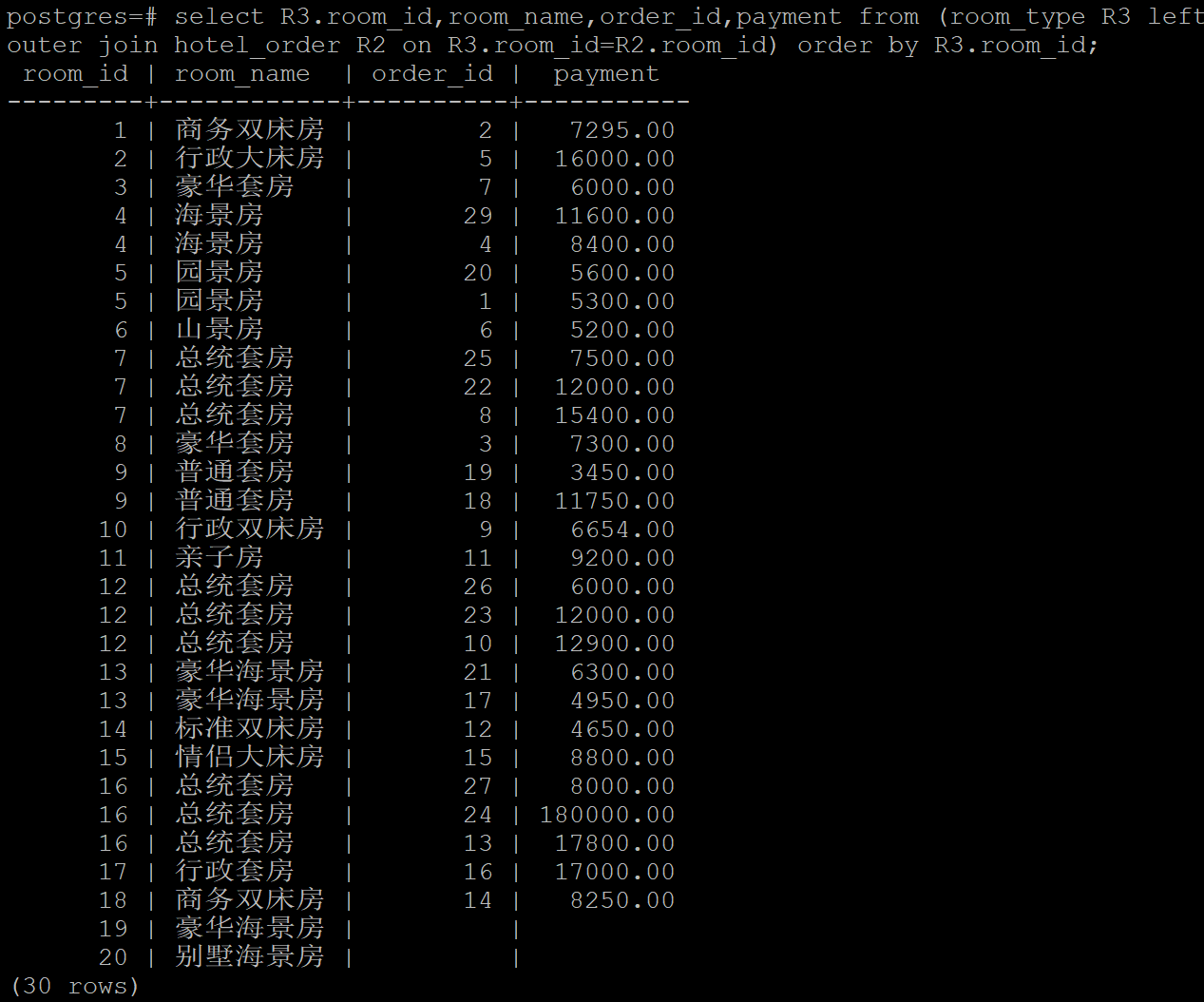
所有房型的订单情况，订单里一定要有所有房型！即时该房型没有出现在订单中

Room\_type left join hotel\_order left join room\_info where



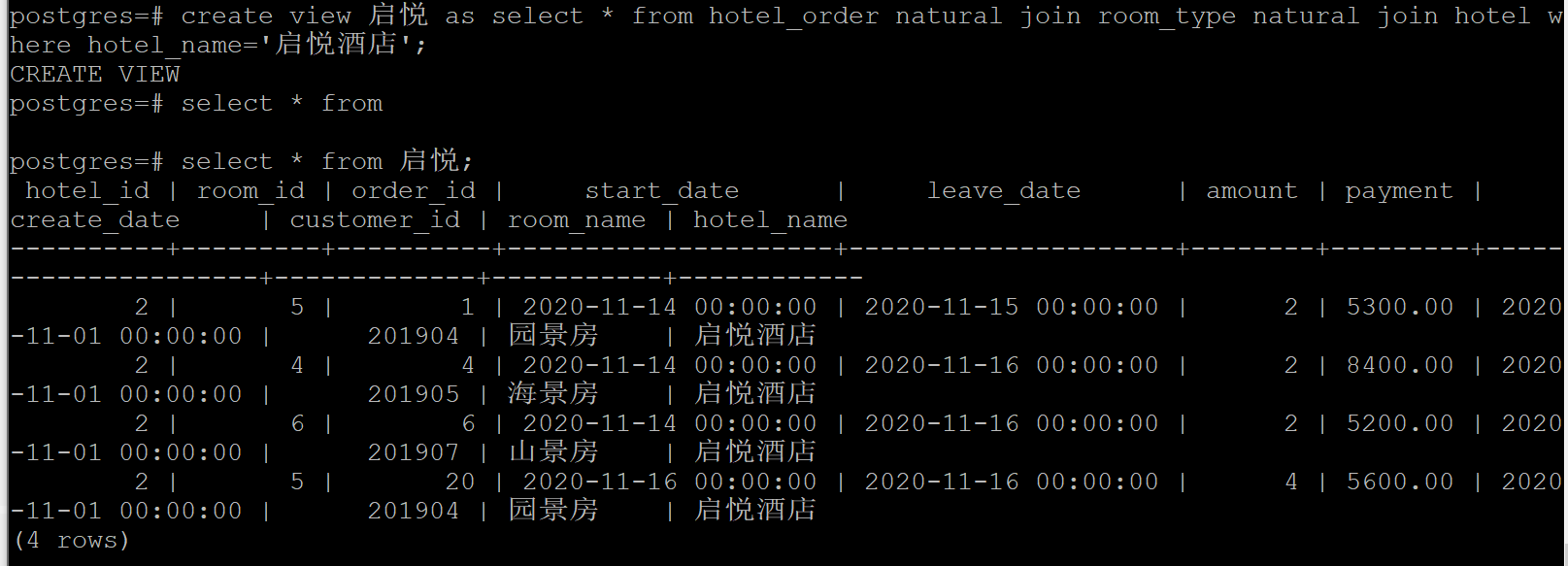
更正：select R1.room\_id,room\_name,order\_id,payment from (room\_type R1 left outer join hotel\_order R2 on R1.room\_id=R2.room\_id) order by R1.room\_id;

联系上下文，说的是订单情况，这个价格指的应该是订单的payment吧



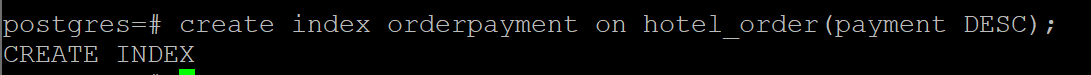
5.

create view 启悦 as select \* from hotel\_order natural join room\_type natural join hotel where hotel\_name='启悦酒店';



6.

create index orderpayment on hotel\_order(payment DESC);

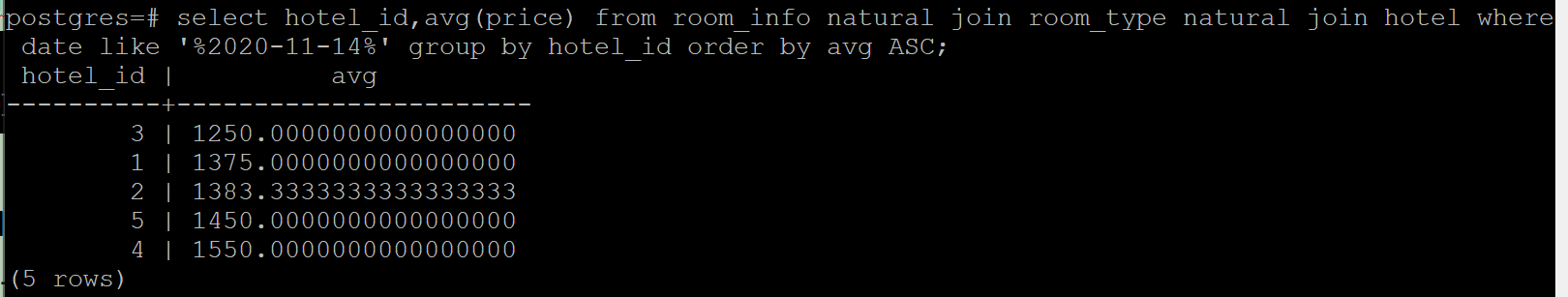


7.

所有酒店

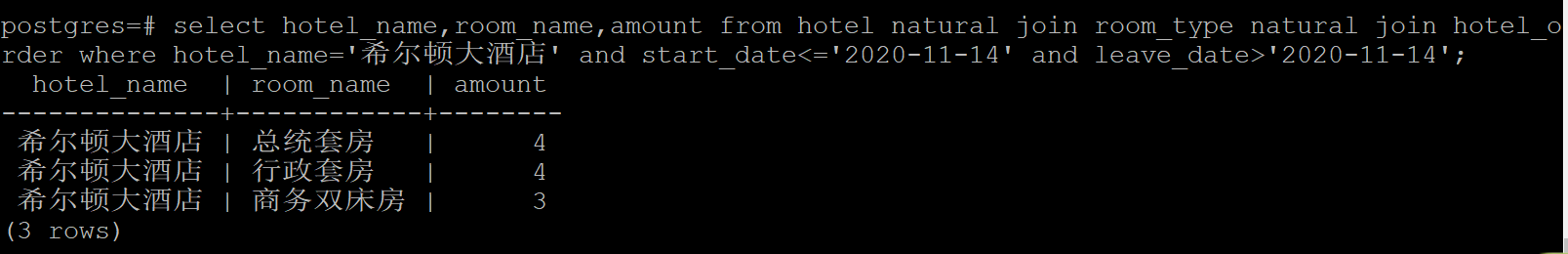
Right outer join hotel where hotel.room\_id=room\_type.room\_id

select hotel\_id,avg(price) from room\_info natural join room\_type natural join hotel where date like '%2020-11-14%' group by hotel\_id order by avg ASC;



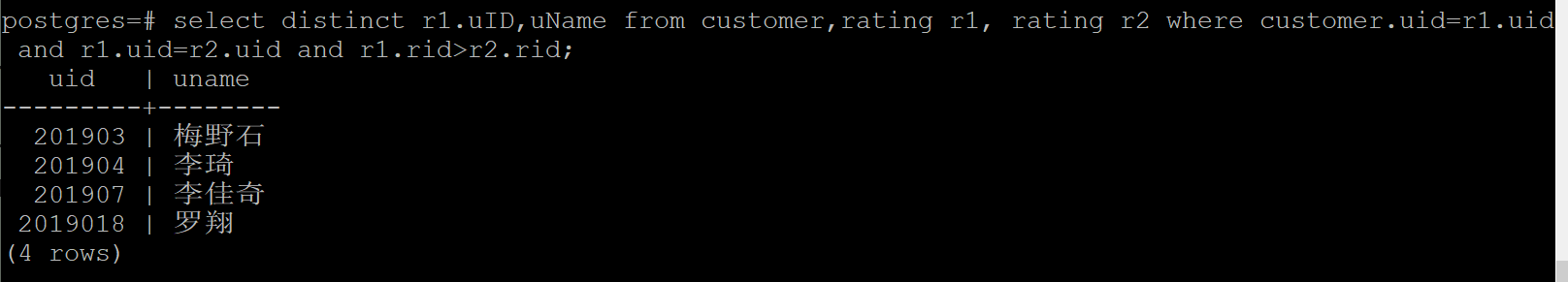
8.

select hotel\_name,room\_name,amount from hotel natural join room\_type natural join hotel\_order where hotel\_name='希尔顿大酒店' and start\_date<='2020-11-14' and leave\_date>='2020-11-14';



9.

select distinct r1.uID,uName from customer,rating r1, rating r2 where customer.uid=r1.uid and r1.uid=r2.uid and r1.rid>r2.rid;



10.

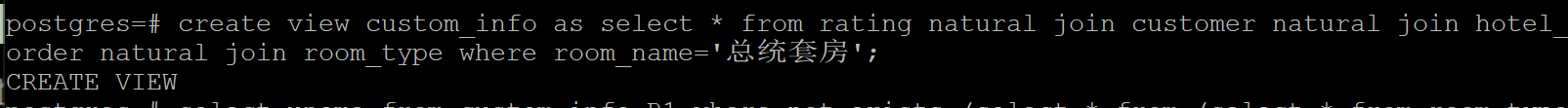
先创建两个视图便于查询：

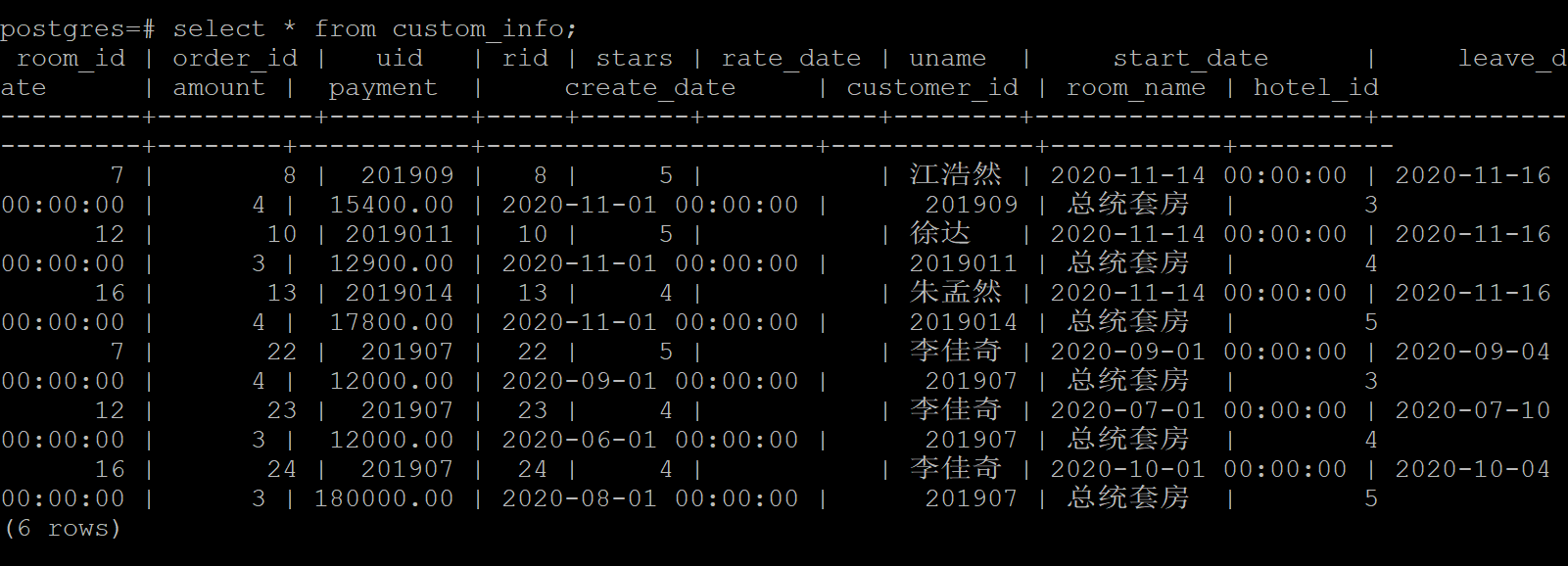
create view hotel\_info as select \* from room\_type where room\_name='总统套房';

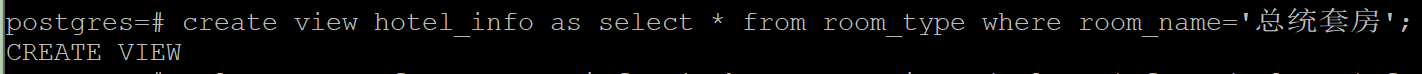
create view custom\_info as select \* from rating natural join customer natural join hotel\_order natural join room\_type where room\_name='总统套房';

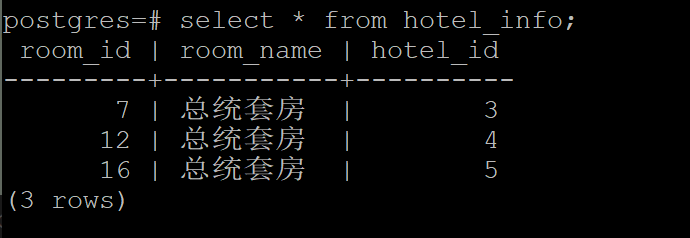
select distinct uname from custom\_info R1 where not exists (select \* from hotel\_info S where S.room\_id not in (select R2.room\_id from custom\_info R2 where R2.uname=R1.uname));

视图：

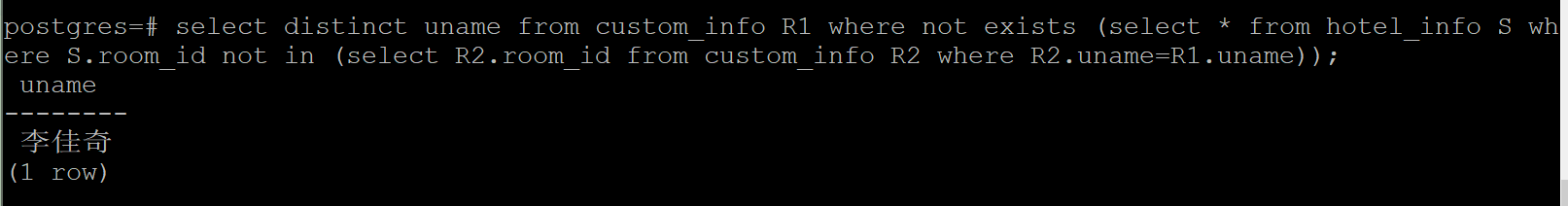






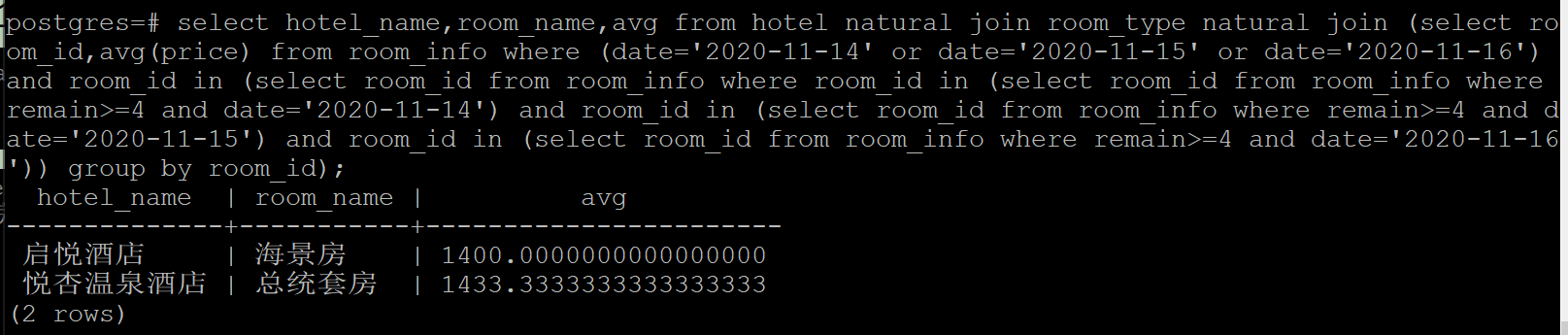


运行结果：



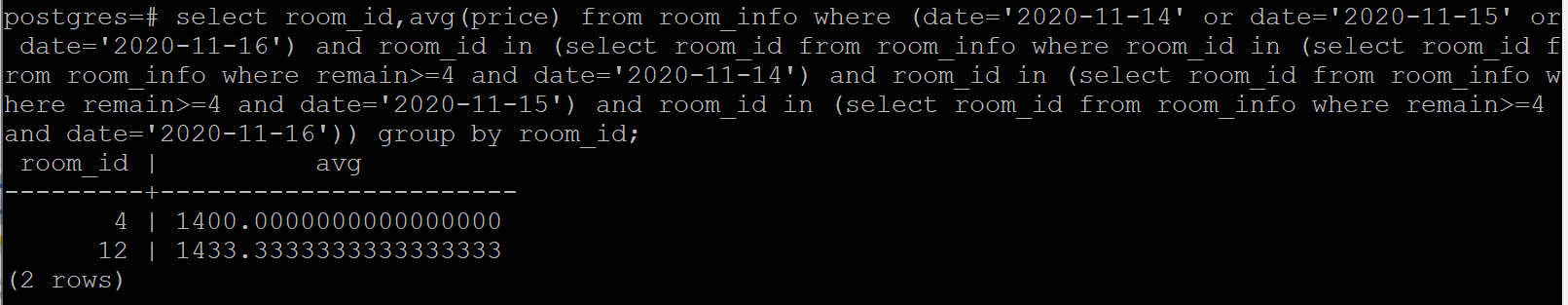
11.

select hotel\_name,room\_name,avg from hotel natural join room\_type natural join (select room\_id,avg(price) from room\_info where (date='2020-11-14' or date='2020-11-15' or date='2020-11-16') and room\_id in (select room\_id from room\_info where room\_id in (select room\_id from room\_info where remain>=4 and date='2020-11-14') and room\_id in (select room\_id from room\_info where remain>=4 and date='2020-11-15') and room\_id in (select room\_id from room\_info where remain>=4 and date='2020-11-16')) group by room\_id);



12.

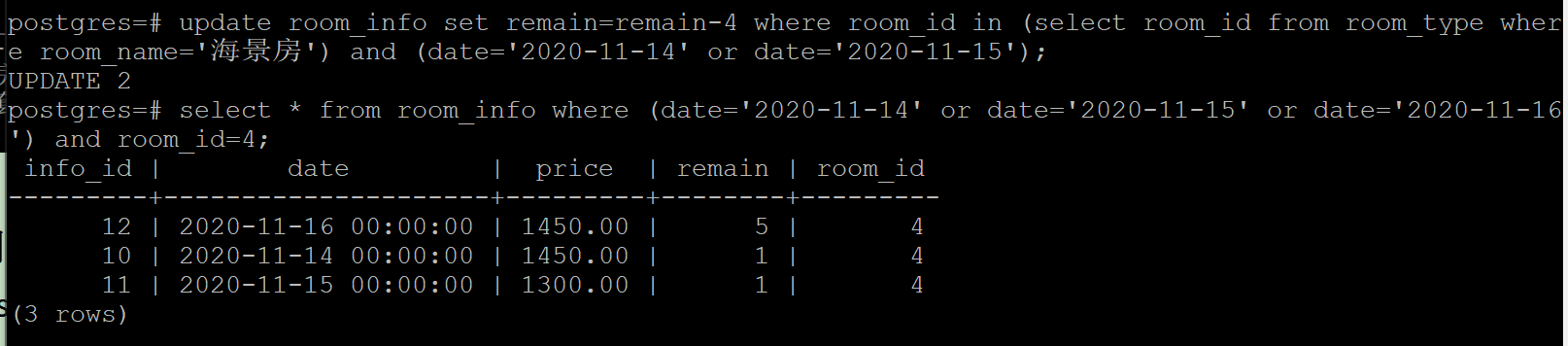
根据上一题的结果可知，能预订的房间有room\_id=4或12



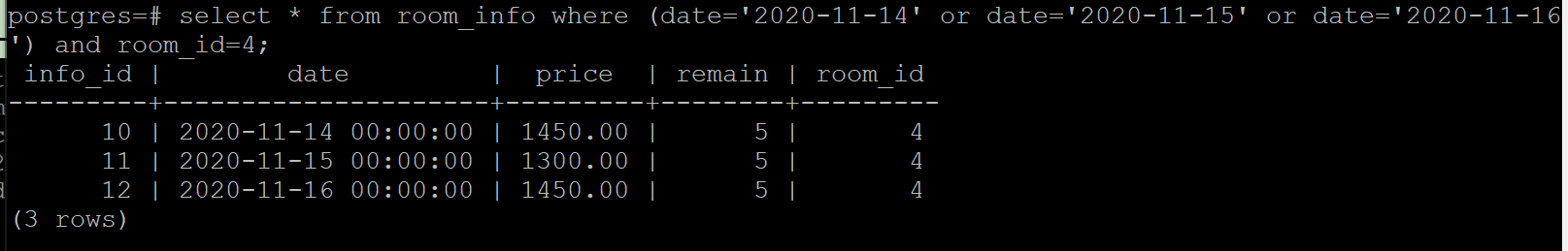
若订单为预订11月14号-15号4号海景房（room\_id=4）4间，客户uid是201908，产生的订单号order\_id是29

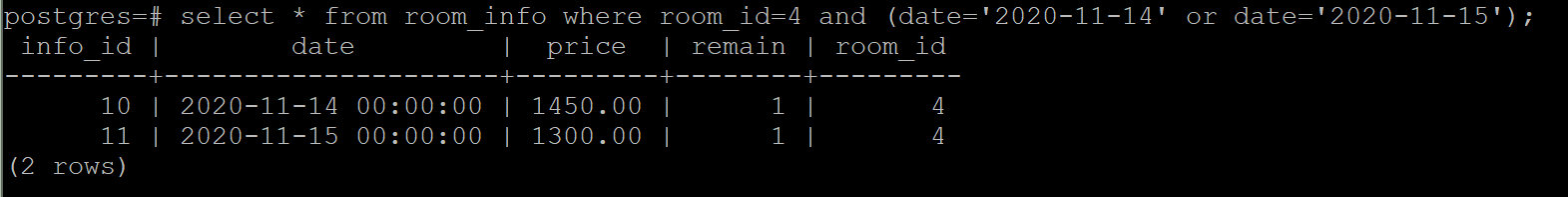
更新房间信息：

update room\_info set remain=remain-4 where room\_id in (select room\_id from room\_type where room\_id=4) and (date='2020-11-14' or date='2020-11-15');



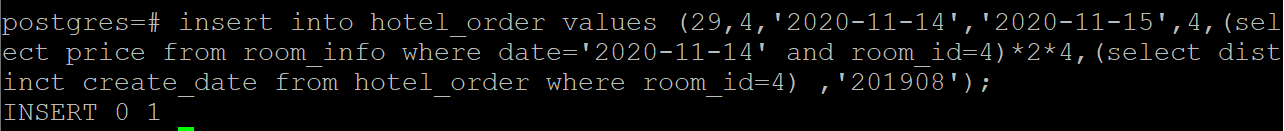
更新前后的数据对比：





插入一条订单信息：

insert into hotel\_order values (29,4,'2020-11-14','2020-11-15',4,(select price from room\_info where date='2020-11-14' and room\_id=4)\*2\*4,(select distinct create\_date from hotel\_order where room\_id=4) ,'201908');



插入的数据：

