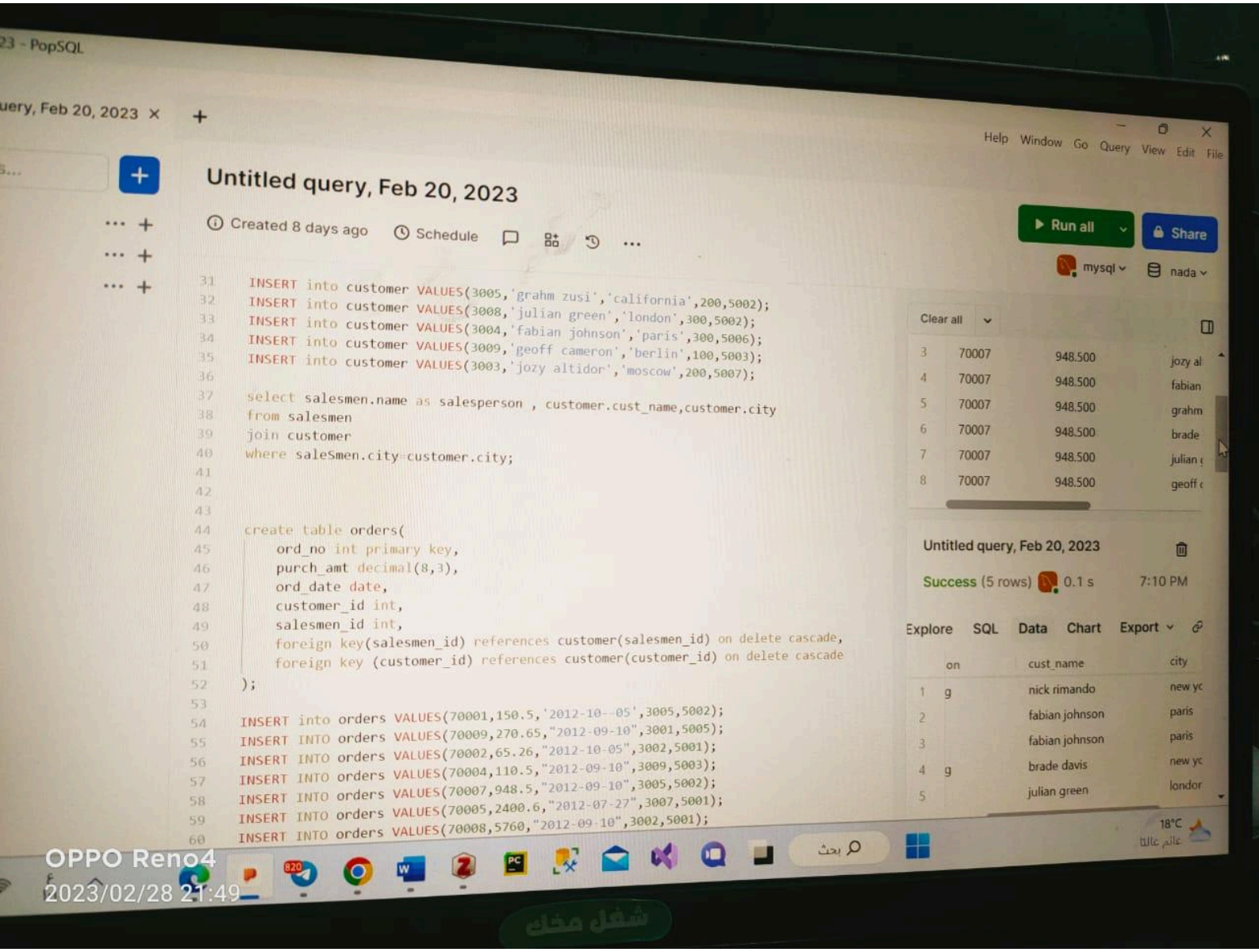
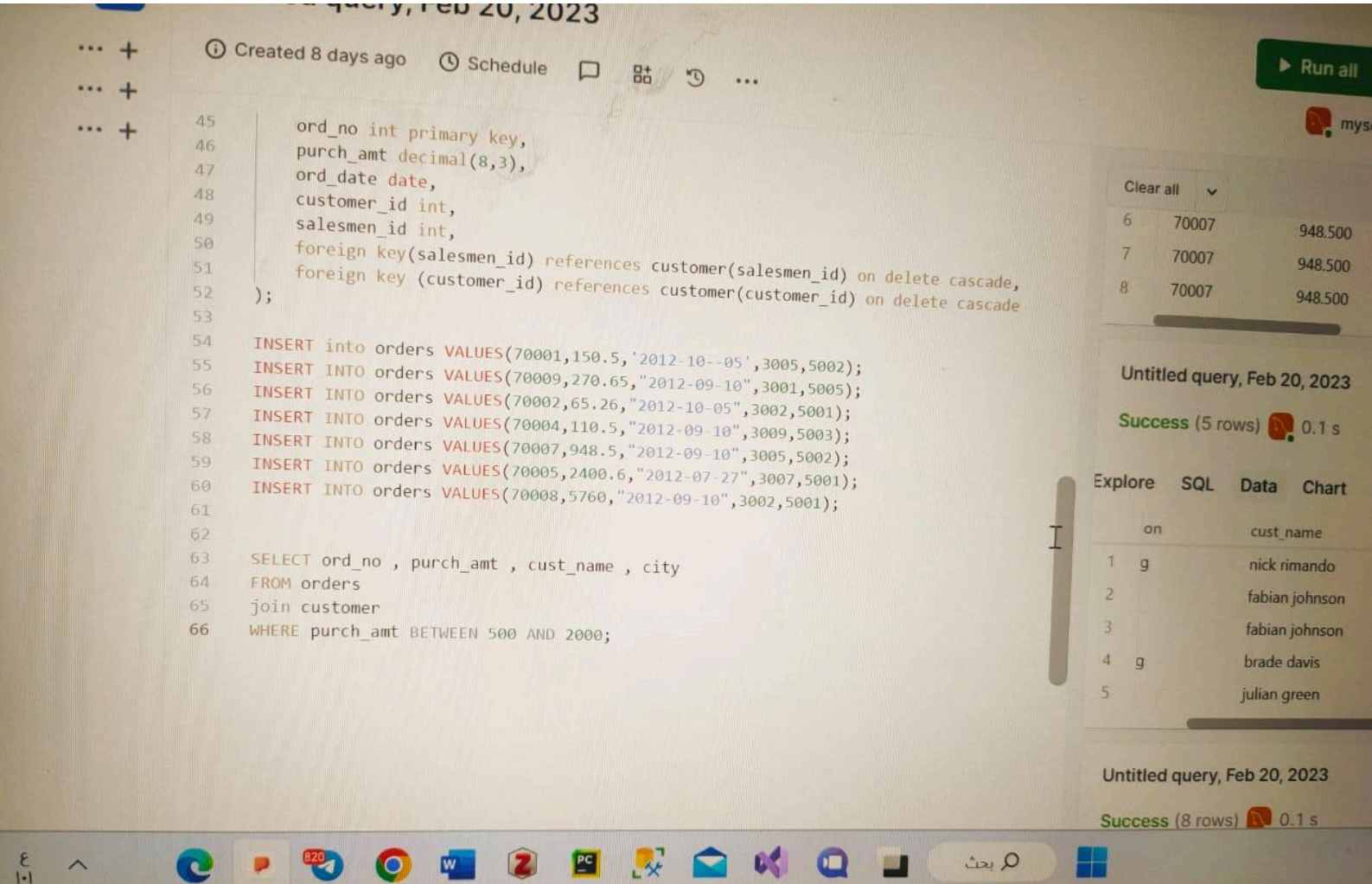




OPPO Reno4
2023/02/28 21:50

* inner join :-

في حالة وجود TwoTables

يقوم innerjoin بأخذ المشترك بينهم

* outer join :- عبارة عن left , Right , full

left :- يقوم بأخذ كل data الموجودة في left وليست موجودة في Right

Right :- يقوم بأخذ كل الموجودة في Right وليس موجودة في left

Full :- يقوم بتجميع كل البيانات الموجودة في الجدولين سواء كان فيه مشترك او لا

التاريخ: / /

الموضوع:

Merg Sort..

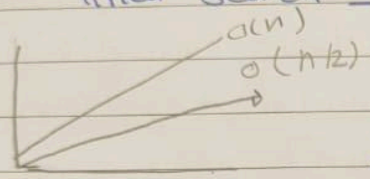
تعتمد على مبدأ التقسيم والتجميع

* تقسم من المنتصف وتضغر Array
وتفضل تقسم Array لحد اصغر حاجة
لحد ما وصل لاصغر شئ اقسمه
بعد كده بدأ نعمل عملية الجمع والدمج
ندمج اعيد بس بالترتيب وندمج صغيرهم
والاصغر يكون الاول
وهكذا لحد ما Array ترجع لشكلها الاول
بس بالترتيب

الخوارزميات:

input → Algorithms → output
مشكلة ... حل

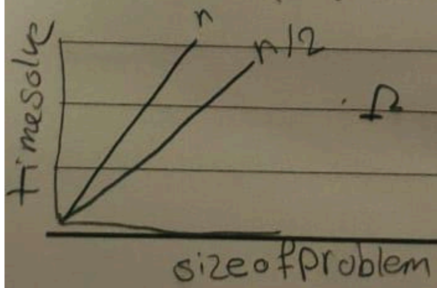
← الكمبيوتر بيشوف index . index
يعني لما يكون واقف عند index ميعرفش حاجة عن الباقي
← عشان نبحث عن افضل خوارزمية بحث
← خوارزمية بتبحث عنصر عنصر → linear search
وتنتج شكل خطي
← أسوء سيناريو O
← أفضل سيناريو Ω



← كود الخوارزمية من كلام نظري لكلام فعلي (Code)
اعمل for loop بتمشي على Array
من البداية للنهاية

```
for i form 0 to n-1
    if number behind doors[i]
        Return true
Return false .
```

← مستخدم Binary search خاصية البحث بالنصف
الي بتبدأ البحث من النصف
أفضل سيناريو (1) Ω
log n لـ
نص نص المشكلة



Binary search:-

if no doors
    Return false
if number behind doors [midde]
    Retun true
Else if number < doors [Middle]
    search doors [0] through [middl-1]
Else if number > doors [middle]
    search doors [middle+1] through doors [n-1]

لازم تكون مرتبة Array خاصية Binary search الـ

selection sort:-

نثبت أنفسك اول index وبعدين نبدأ نتنقل على كل العناصر وبعدين تشوف اصغر index وتعمل swap بينه وبين العنصر اللي هو واقف عليه وهكذا

$n + n-1 + n-2 + n-3 ...$

$O(n^2)$ اسوء سيناريو

=

n^2

hold and Compare

↓ loop ↓ loop

التاريخ: / / الموضوع:

Bubble sort:

تعتمد على مبدأ مقارنة كل

يعني أي Two index ومقارنة بينهم

if number [i] and numbers [i+1] out of order swap them

$O(n^2)$ أسوأ سيناريو

$O(n)$ أفضل سيناريو

Repeat n-1 times

for i from 0 to n-2

if numbers [i] anel numbers [i+1] out of order

swap them

if no swaps

Out

---

recursion: void fun (int x)

{

fun(x);

}