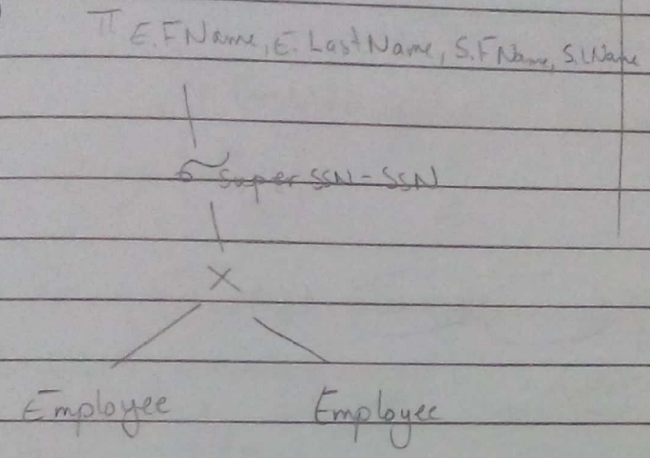
Sec 3 Advanced DB
5/11/2018

Chapter 15 : Query Processing

Sheet

15.13

①



select ( ) from tbl1, tbl2
where (cond1, cond2)

① نعمل Cartesian Product

② بعد كدا اعمل ال cond.

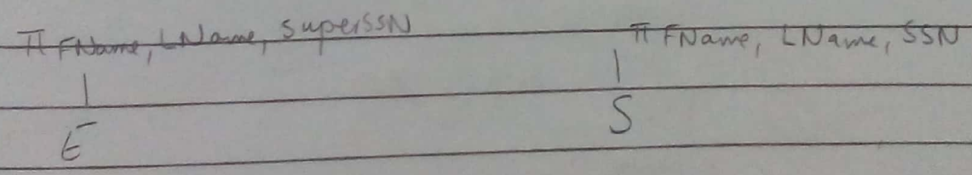
③ بعد كدا اعمل ال select

* دا كدا من غير optimization

* ال tree دي كدا ← optimized عايزين نعملها optimization

* طب ما بلاش اشتغل على كل ال fields في ال Cartesian product ممكن انزل ال select

قبل بحيث اشتغل على fields اقل (اقل blocks ←)



* ولازم اجيب ال π اللي فوق عشان انا محتاج ال SSN
وال ال superSSN

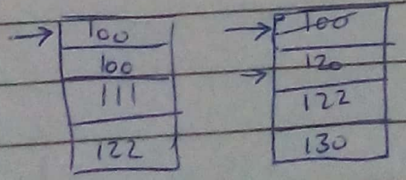
* طب لو ال 2 table ← sorted على ال SSN

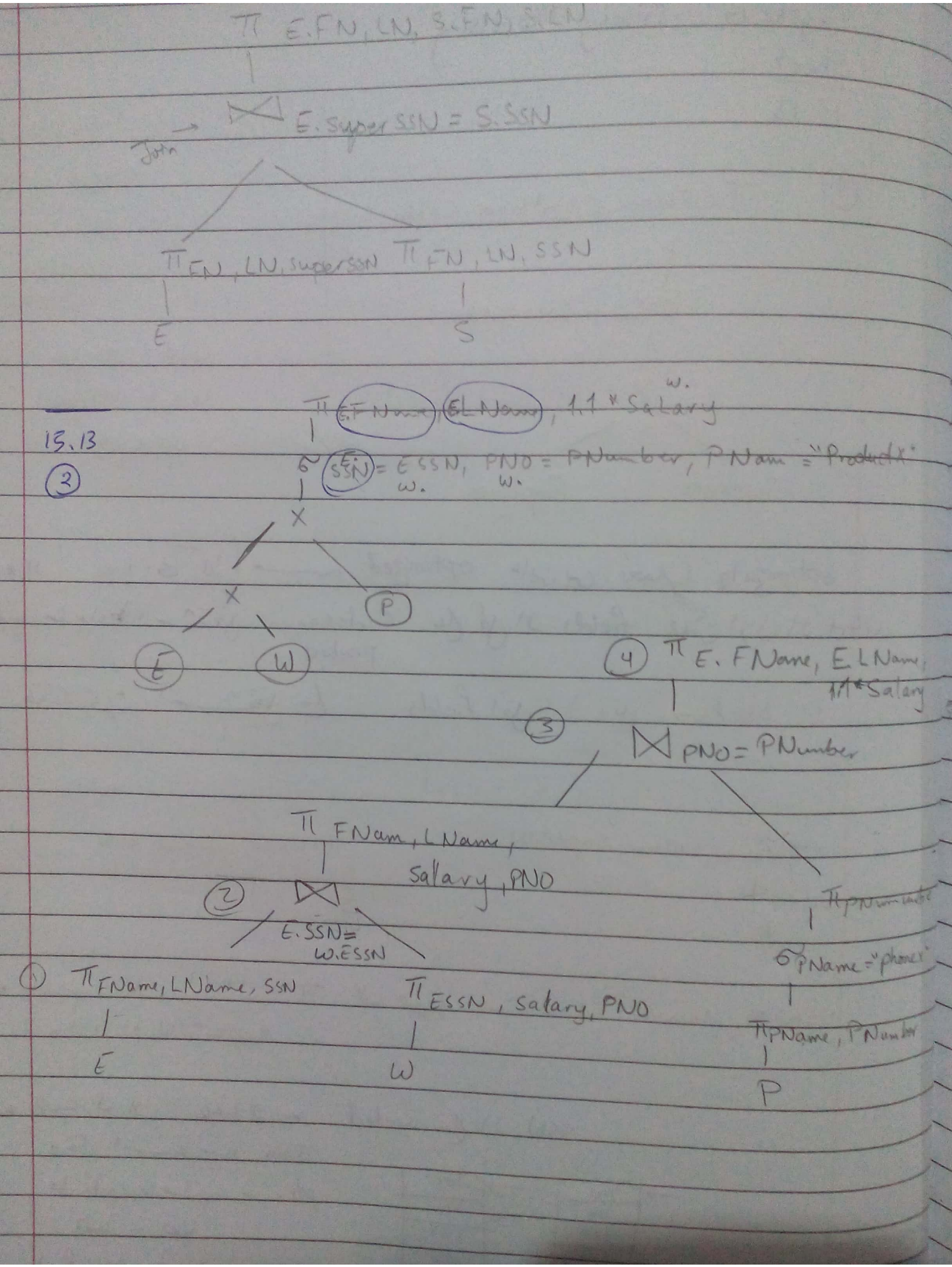
ممكن استخدم ال Join

دا كأني ماشي بـ ptr.

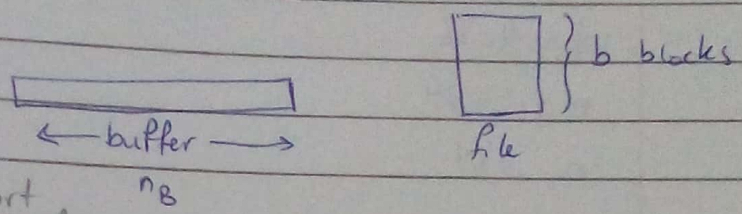
هنا و هنا

لو equal هعمل Join





* Cost of External Cost



n_R Runs to sort the file

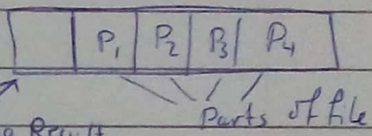
$$n_R = \left\lceil \frac{b}{n_B} \right\rceil$$

$$d_m = \min(n_{B-1}, n_r)$$

$$n_p = \log_{d_m}(n_R)$$

n_p ↓ عدد ال merge

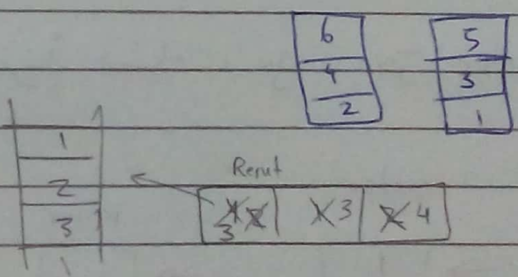
* merge



← بتحط بعدها في ال disk

*ال file أكبر من ال memory

ex



* عندي 8 blocks وال memory ← 3 (يعني p ← تقسم 2 , والثالث للResult)

8	7	6	5	4	3	2	1

⇓

4 blocks needed to be merged

⇓

2

↓

1

n_R 100 → يعني هقسم 10 by 10

11 block in mem

100 ↓ 90 ↓ 80 ↓ ... 1

$= \log_{10} 100$

* يعني انا بقسم على $n_B - 1$ ← (dm)

$$n_p = \log_2 (8)$$

($8 = n_R$, $2 = n_{B-1}$)

أو ال n_R لو هي أقل
يعني أصغر من ال memory
يعني هعمل merge مرة واحدة

$$n_p = \log_{n_R}(n_R) = 1$$

15.14

$b = 4096$

$n_B = 64$

$n_r = 64$

$d_m = \min(64, 63) = 63$

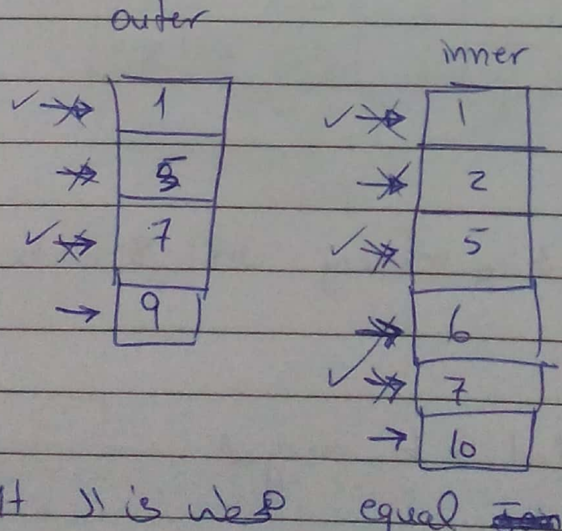
$\lceil \log_{63}(64) \rceil = 2$

15.17 No ,

15.21

* طب ابني ذا ال merge Algorithm تقريباً عشان اعمل outer Join

يعني لو عندي كدا



① اول case لو ~~[illegible]~~ equal هحط ذا ال result
بتاع ال Join و هـ advance الاثنين

② ثاني case لو وصلت equality ، والصغير هو اللي ذا ال inner
زي 5 (outer) , 2 (inner) يعني هـ advance ال ptr بتاع ال inner بس

outer inner

③ ثالث حاجة بقى في الأمر 9,10

الـ outer هو الصغير يبقى كل حاجة في الـ Result

واخد قيمة NULL