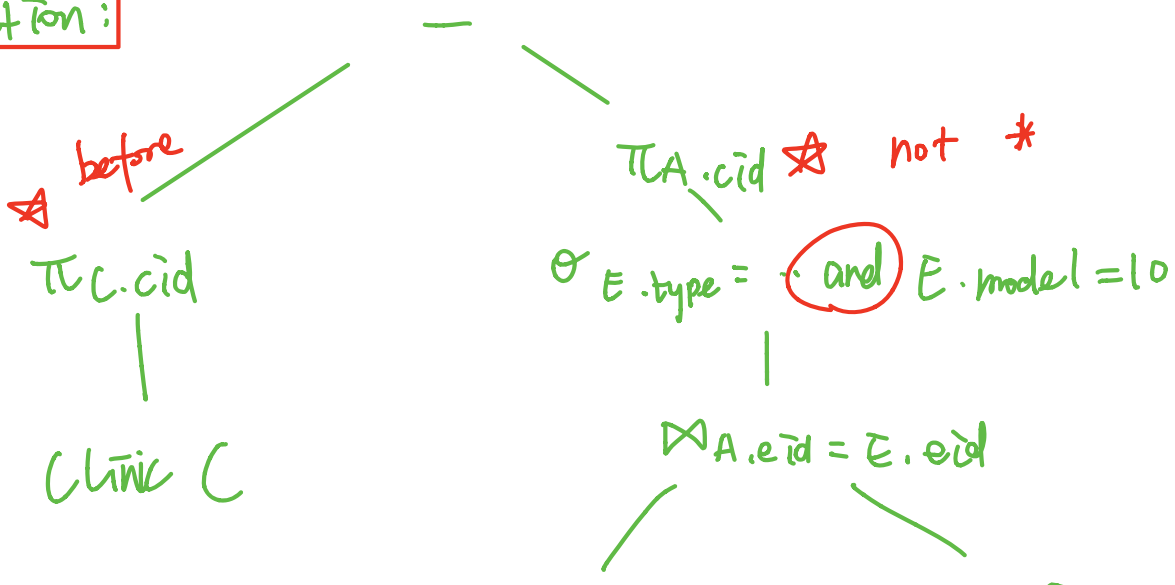
From, Where, Group By, Having, Order By, Select

- Sqlite only support LEFT OUTER JOIN.
- FULL OUTER JOIN <=>
  (SELECT * FROM A LEFT OUTER JOIN B
  ON A.a = B.b) UNION (SELECT * FROM
  B LEFT OUTER JOIN A ON A.a = B.b);
  UNION ALL is a bag that keeps duplicate; union doesn't.

- RA: 先select 再 union, intersect, difference
  EXist 的条件相当于 operation 前的 select.

Ie.: SELECT C.cid
FROM Clinic C
WHERE NOT EXISTS (SELECT * FROM Assi A, Equi E
WHERE C.cid = A.cid AND A.eid = E.eid
AND E.type = 'Frid' AND E.model = 10);

Solution:



Assi A Equi E

- GROUP BY 的时候 SELECT 里面的 attributes 都要和 Primary Key 一起写.

i.e.:
```
SELECT  I.username , I.fname , I.lname
FROM    ...  I ,  ... T
WHERE   I.username = T.username
GROUP BY  I.username , I.fname , I.lname
HAVING COUNT(*) > 1
```

- How many instructors teach in department(s) w/ the most instructors ?

- 找 element 数最多的那个 group, 和 element 数量.

```
WITH  DeptInstr AS (
      SELECT  dept , count(*) as cnt
      FROM  Teaches
      GROUP BY  dept
), MaxInstrCnt AS (
 SELECT MAX(cnt) as maxcnt
 from DeptInstr
)
 SELECT DI.dept , DI.cnt
 FROM  DeptInstr DI, MaxInstrCnt MIC
 WHERE DI.cnt = MIC.maxcnt ;
```

Nested Loop Semantics

```
SELECT x_1.a_1, …, x_n.a_n
FROM x_1, …, x_n
WHERE <cond>

for each tuple in x_1:
    …
        for each tuple in x_n:
            if <cond>(x_1, …, x_n):
                output(x_1.a_1, …, x_n.a_n)
```

- 题目一但提到 for each sth，count(*)，一般存在 count 为 0 的情况，解题用 sth LEFT OUTER JOIN 以保留 count = 0 的 row.

- 写 RA γ 时上面 σ & π 要用的 attributes 都要写

$\pi_{s,t}$ (简化)

γ Sum(xxxxx) → s, Total(xxxxx) → t

Join 时不要弄列，要写成正规的格式.

NOT:

BUT

- Natural Join 取 A 和 B 的交集

如果 $n \leq m$，max cardinality 是 n

distinct foreign key

key

i.e.: set: R S

size: m n

foreign key → key

max card: 至多 m.

|Set(Join btwn key & foreign key)| ≤ |Set(foreign key)|

- Functional dependencies

3. **(18AU Final)**
Find all functional dependencies in the following table. You only need to write a minimal set of dependencies that logically imply others.

A	B	C	D	E
0	0	0	0	0
1	1	1	1	1
2	2	2	2	0
3	3	3	0	1
4	4	0	1	0
5	5	1	2	1
6	0	2	0	0
7	1	3	1	1
8	2	0	2	0
9	3	1	0	1
10	4	2	1	0
11	5	3	2	1
12	0	0	0	0

Solution: $A \to BCDE$, $C \to E$, $B \to DE$, $DE \to B$, because:

$B = (A \mod 6)$ $C = (A \mod 4)$ $D = (A \mod 3)$ $E = (A \mod 2)$

no points off for missing $DE \to B$
1 point partial credit for writing just $A \to BCDE$.

- 有 x.count = MAX(x.count) 的属于 witnessing problem.
或者 max(count(*)) 都是 Invalid.

witnessing problem:

```
WITH A ( SELECT w.year, w.pid, COUNT(w.pid) as cnt
         FROM   WorkOn w      -- get each count for year&pid
         GROUP BY  w.year , w.pid ),
```

```
B ( SELECT A.year, MAX( A.cnt ) as max
    FROM A          -- get max of each pid count in each year
    GROUP BY A.year )
SELECT A.year, A.pid
FROM A, B
WHERE A.year = B.year AND
      A.cnt = B.max ;
```

- 大概就 HAVING 里有的 attribute, Group By 的时候也要写

```
SELECT D.did, D.name
FROM Projects P, WorksOn w, Developer D
WHERE w.year >= p.startYear AND
      w.year <= 2015 AND
      w.did = D.did AND w.pid = D.pid AND
      p.name = 'Sys X'
GROUP BY D.did, D.name, P.startYear
HAVING COUNT(*) = 2015 - p.startYear + 1 ;
```