Question 1

1)

- A(<u>a1</u>, a2)
- B(<u>a1</u>) , a1 refer to A
- C(<u>a1</u>, c1) , a1 refer to A
- D(<u>d1</u>,d2)
- E(<u>e1</u>, e2, bd1), a1 refer to BD
- F(<u>f1</u>,f2, a1) , a1 refer to C
- BD(<u>bd1</u>, a1, d1)

2)

Question 2

NOT ON FINAL

Question 3

1)

```
a)
SELECT c.pids FROM `Current` c JOIN Past p ON (c.tid = p.tid, c.pid = p.pid);

b)
SELECT * FROM
(SELECT tid, pid, `from`, `to` FROM past
UNION
SELECT tid, pid, `from`, current_time FROM current) as union_select
ORDER BY (tid,`from`);

c)
SELECT DISTINCT pids FROM `Current` c JOIN Task t ON c.tid = t.tid WHERE t.difficulty >= 4;

d)
SELECT tids FROM Task t WHERE (
    SELECT count(*) FROM Past p WHERE p.tid = t.tid
) > 10;

e)
SELECT tids FROM Task t WHERE (
    SELECT count(*) FROM Past p WHERE p.tid = t.tid
) + (
    SELECT count(*) FROM Current c WHERE c.tid = t.tid
) > 10;

f)
CREATE TRIGGER pointless
    BEFORE INSERT ON `Current`
    REFERENCING NEW as nrow
    FOR EACH ROW
        INSERT INTO Past VALUES (
            SELECT *, current_time FROM `Current` c WHERE c.tid == nrow.tid
        )
        DELETE FROM `Current` c WHERE c.tid == nrow.tid;
```

2)

a) /recipes[preparation[count(step) >= 2]]

b) /recipes[[@title=' Gingerbread Cookie']/nutrition/@fat = /nutrition/@fat]

Question 4

1)
- i) YES
- ii) Not sure
- iii) NO

2) T1 -> T2, T1 <-> T2.

Question 5

1) First we are going to need to access the root (1) then the intermediate node (2) and finally the leaf page where the entry should be placed (3). Then in the best case we are only going to access 3 index pages. Now for the worst case the leaf page is full so we need to split it and insert back in the parent page (4). In the worst case the intermediate is also full so we need to split again and put in the root node (5)

2)

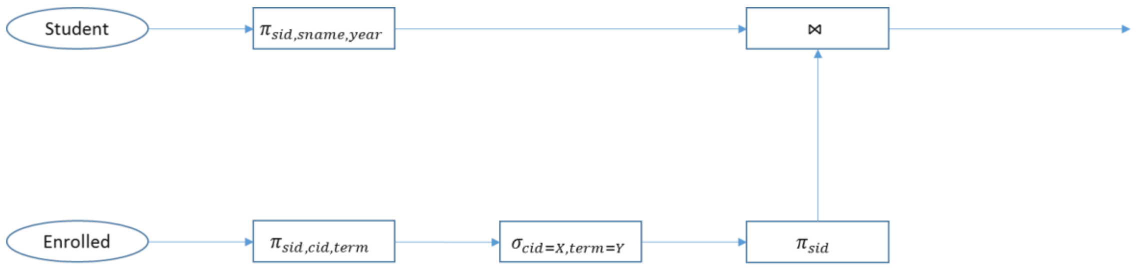
i) This query return all the student that took a given course at a given semester.
We know that we have 400 000 entries and 10 different term in the table enrolled and 5000 courses. Then it means that at a given term there is 400 000/10=40 000 student course mapping. But as we have 5000 courses each courses is taken by only 40 000/5000=8 students on average. Then the query will return 8 tuples on average.

ii) First the cost of the block nested loop is

$$Cost = page(Enrolled) + \frac{page(Enrolled) * page(Students)}{B - 2} = 2000 + \frac{2000 * 250}{48} = 12417$$

Now as the selection is done on the fly it's at no additional cost and same for the projection.

iii)


Student
$\pi_{sid,sname,year}$
$\bowtie$
Enrolled
$\pi_{sid,cid,term}$
$\sigma_{cid=X,term=Y}$
π_{sid}