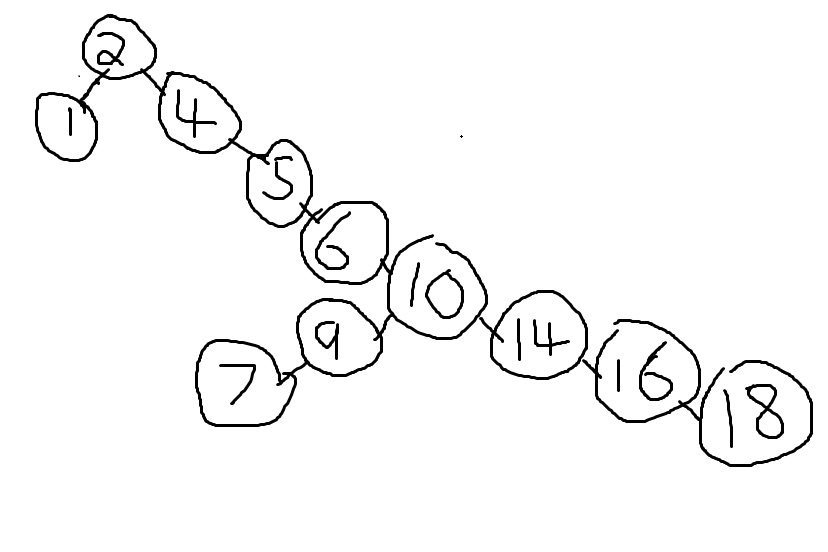
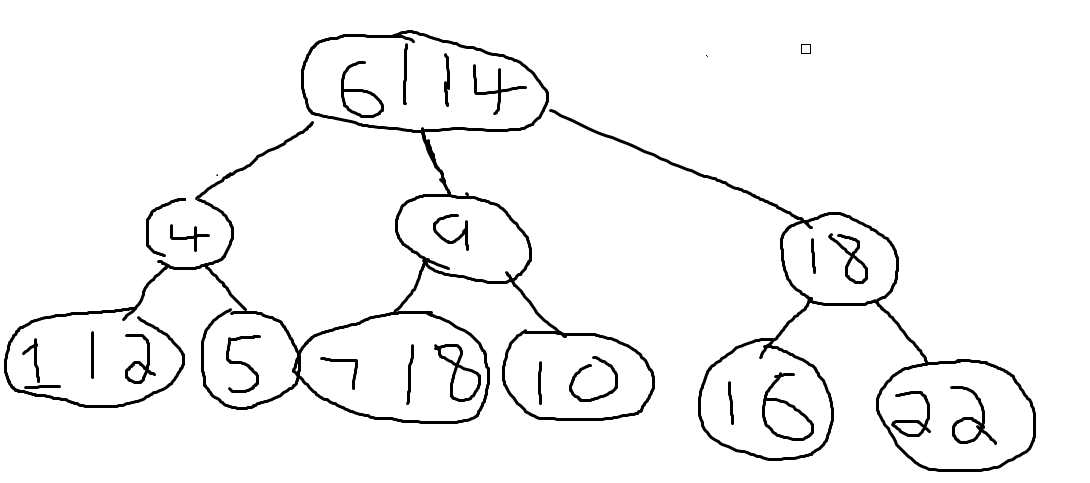
**Raid**

RAID 0: Is implemented by striping blocks. Excellent performance, no redundancy. Using RAID 0 we can use the eight fast disks and disregard the two slow disks. Meaning the speed will be 8000MBs/S

**B-Trees**

Binary Tree B-Tree

This particular binary tree is skewed right (i.e. the vast majority of the nodes are to the right of the first node). Where as with the B-tree there is a move even distribution of nodes.   
  
The main advantage of using a B-tree is that searching is much more efficient in B-trees than in binary trees.

**Contraints and Triggers**

A. Create Or Replace Trigger logTrig  
 After Insert on Teams  
 Begin  
 Insert into logTable values(“insert done”, SYSDATE);  
 END;

B. Check(Competition=”Champions League” Or Competition=”Europa League” or Competition=”Premier League” or Competition=”La Liga”);

C. Check(TeamA\_ID.country= “England” Or TeamA\_ID=”Spain”);  
 Check(TeamB\_ID.country =”England” Or TeamB\_ID=”Spain”);

D. Check(Goal\_A >=0 <= Goal\_B);

E. Create Or Replace Trigger LeagueComp  
 Before Insert Or Update On Matches  
 Begin  
 If Competition = “Premier League”  
 Where TeamA\_ID = Team\_ID   
 Check(TeamA\_ID=”England”);  
 And Where TeamB\_ID=Team\_ID  
 Check(TeamB\_ID=”England”);

Else If Competition = “La Liga”  
 Where TeamA\_ID = Team\_ID   
 Check(TeamA\_ID=”Spain”);  
 And Where TeamB\_ID=Team\_ID  
 Check(TeamB\_ID=”Spain”);  
 END;

F. CREATE OR REPLACE TRIGGER home\_trg

BEFORE INSERT OR UPDATE ON matches

FOR EACH ROW

DECLARE

how\_many NUMBER;

BEGIN

SELECT COUNT(\*) INTO how\_many FROM matches

WHERE TeamA\_ID = :new.TeamA\_ID;

IF how\_many >= 3 then

Raise\_application\_error(-20000,'HomeTeam' ||

:new.TeamA\_ID || 'already has 3 Home Games');

END IF;

END;