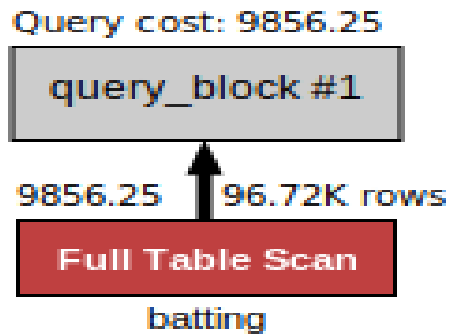


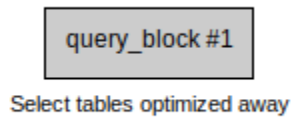
Query 1 select max(H) from batting

Query cost before optimizing. Cost - 9856.25



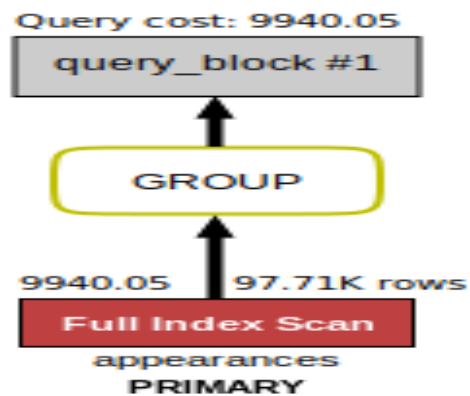
Query Cost after optimizing by creating index on batting table.

create index index_batting_H on batting(H);



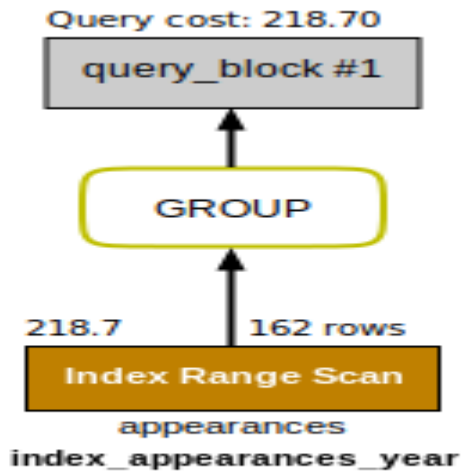
Query 2 SELECT yearID,max(G_1B)FROM appearancesGROUP BY yearID

Query cost before optimizing is 9940.5



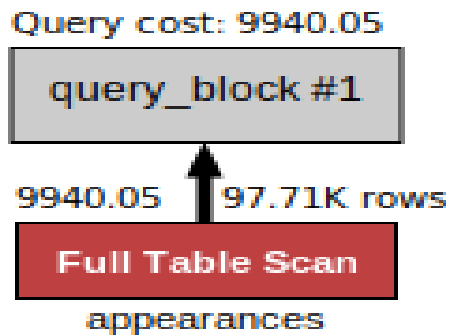
Query Cost after optimizing is 218.70 creating index on appearances table, columns yearID, G_1B

create index index_appearances_year on appearances(yearID, G_1B)



Query 3 SELECT yearid, g_pFROM appearances

Query cost before optimizing is 9940.05



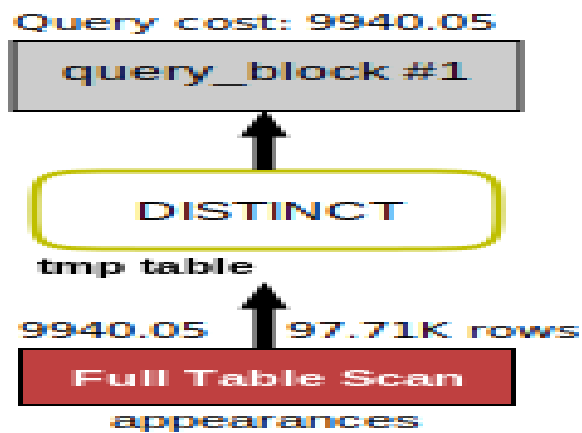
After Optimizing Query with Index

CREATE INDEX index_on_yearid ON appearances(yearid);

The Query Cost is 2421.6

Query 4 SELECT DISTINCT yearid, g_pFROM appearances

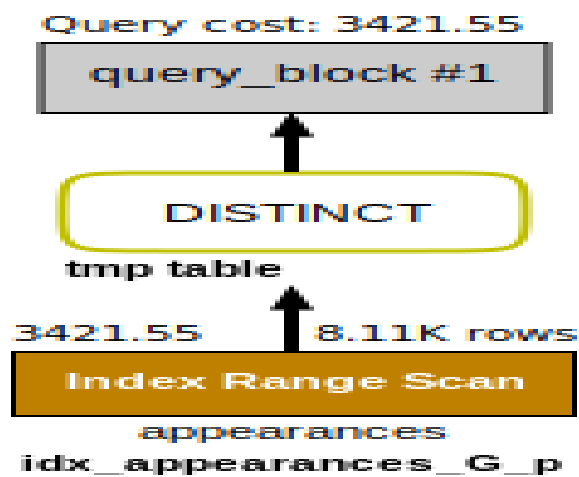
Query Cost before optimization is 9940.05



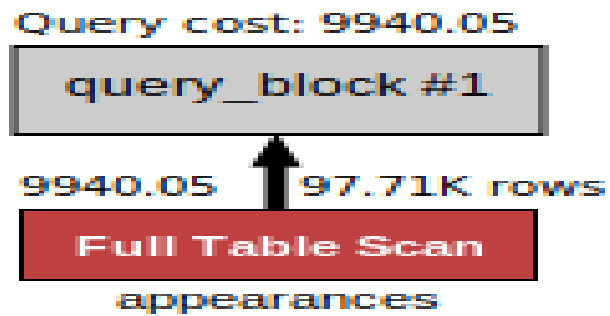
Query Cost after optimization by creating index on column G_P , yearID is a primary key for which index is created by default 3421.55

Query Cost after optimization is

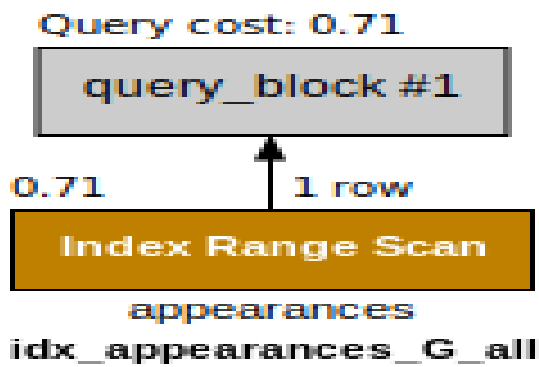
create index idx_appearances_G_P on appearances(G_P)



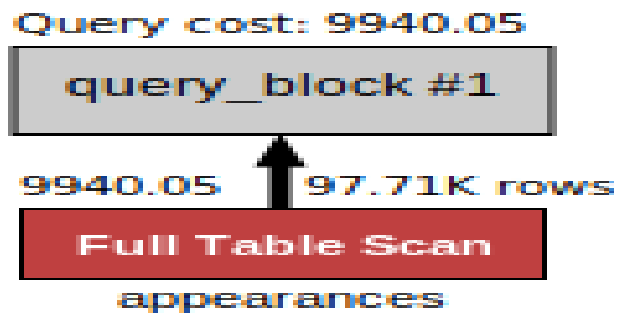
Query 5 SELECT *FROM appearancesWHERE g_all > 190
Query Cost before optimization is 9940.5



Query cost after creating index on column g_all is 0.71
create index idx_appearances_G_all on appearances(g_all)

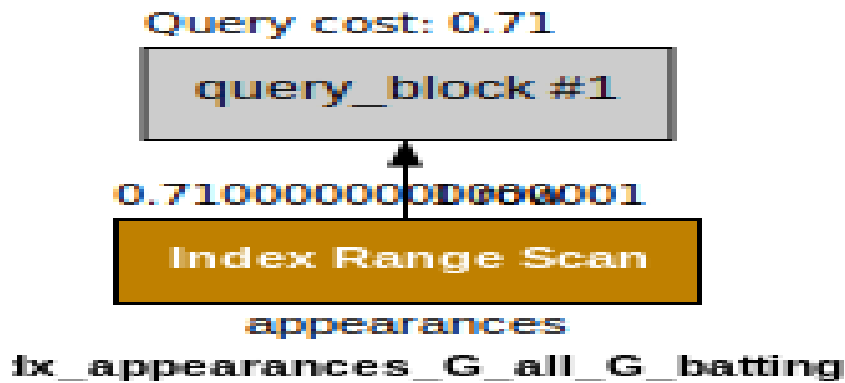


Query 6 SELECT *FROM appearancesWHERE g_all > 190 AND g_batting < 170
Query Cost before optimizing is 9940.05



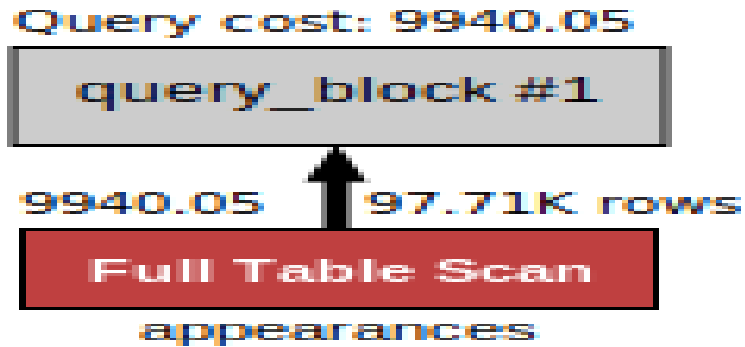
Query Cost after optimization is 0.71

create index idx_appearances_G_all_G_batting on appearances(g_all,g_batting)

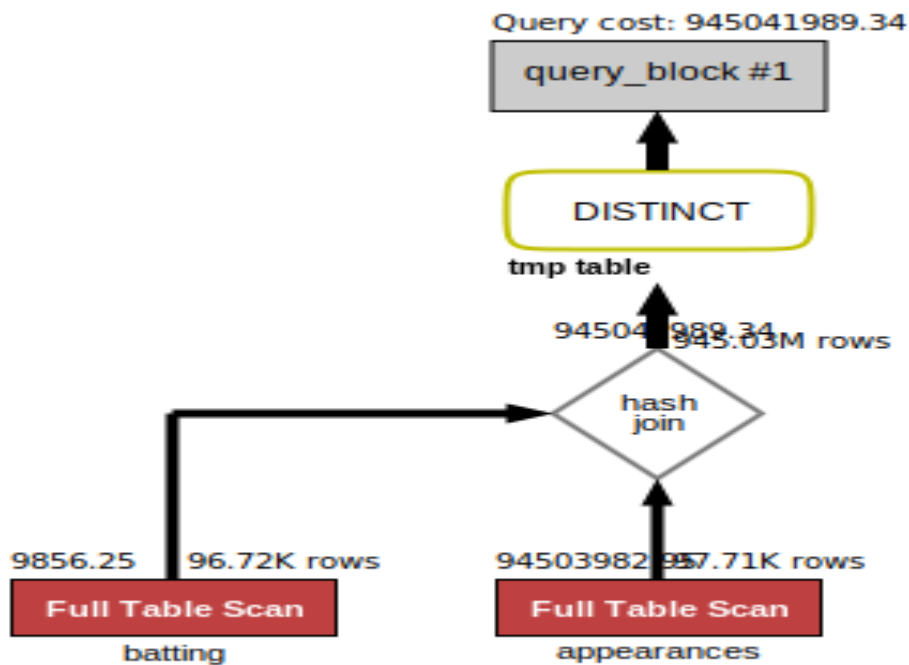


Query 7 SELECT *FROM appearancesWHERE g_all > 190 OR g_batting < 170

Query Cost before optimization is 9940.05



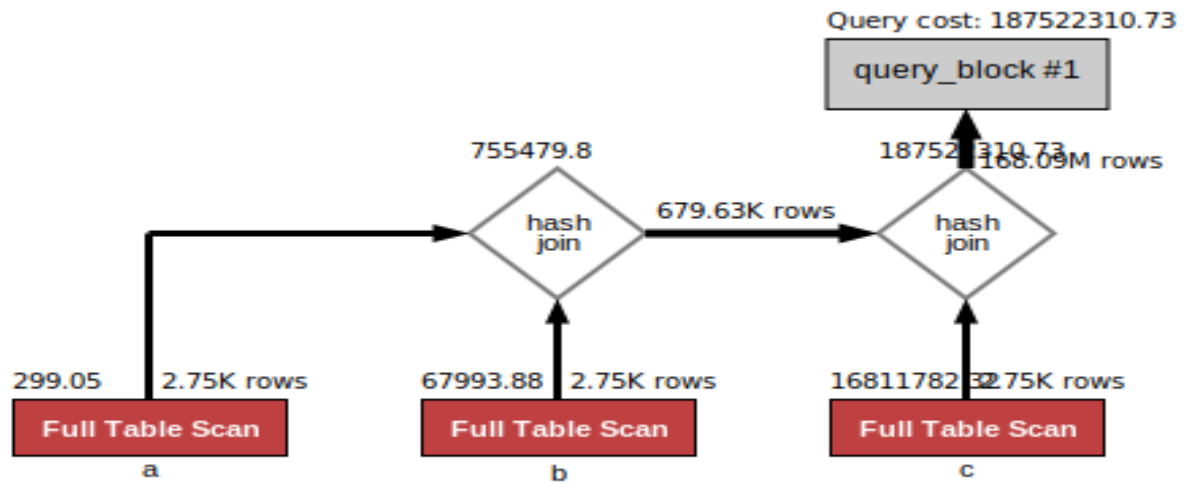
Query 8 SELECT DISTINCT appearances.g_batting, batting.ab FROM appearances, batting WHERE appearances.G_batting = batting.G_batting
Query cost before optimization is 945041989.34



Query Cost after optimization is 2347133.88

create index idx_appearances_G_batting on appearances(G_batting)

Query 9 SELECT a.teamid, a.yearid, b.yearid, c.yearid FROM teams a, teams b, teams c WHERE a.teamid <> b.teamid AND b.teamid <> c.teamid AND a.w = b.w AND b.w = c.w



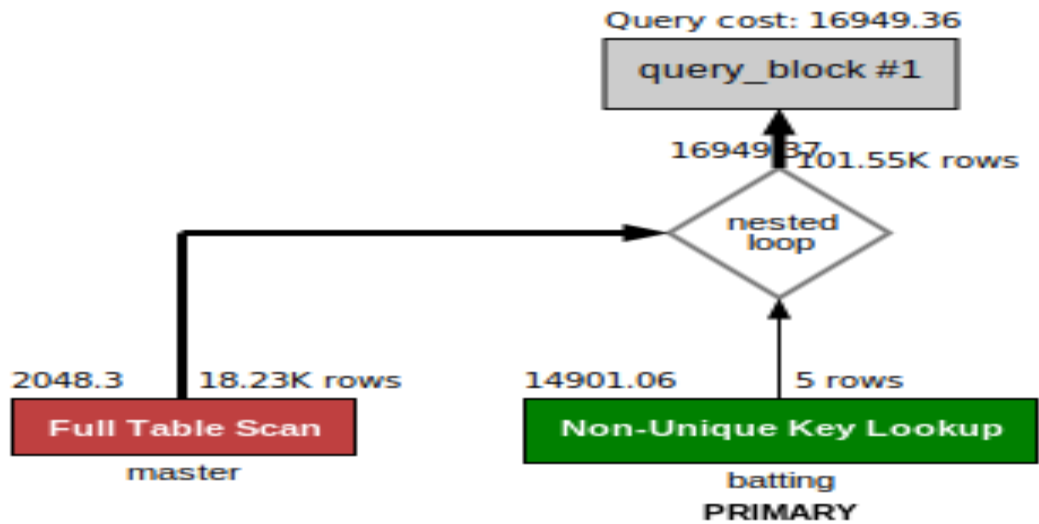
After Optimization Query Cost 100748

CREATE INDEX index_teams_teamid ON teams(teamid) USING btree;

Query 10 SELECT nameLast

FROM master, batting

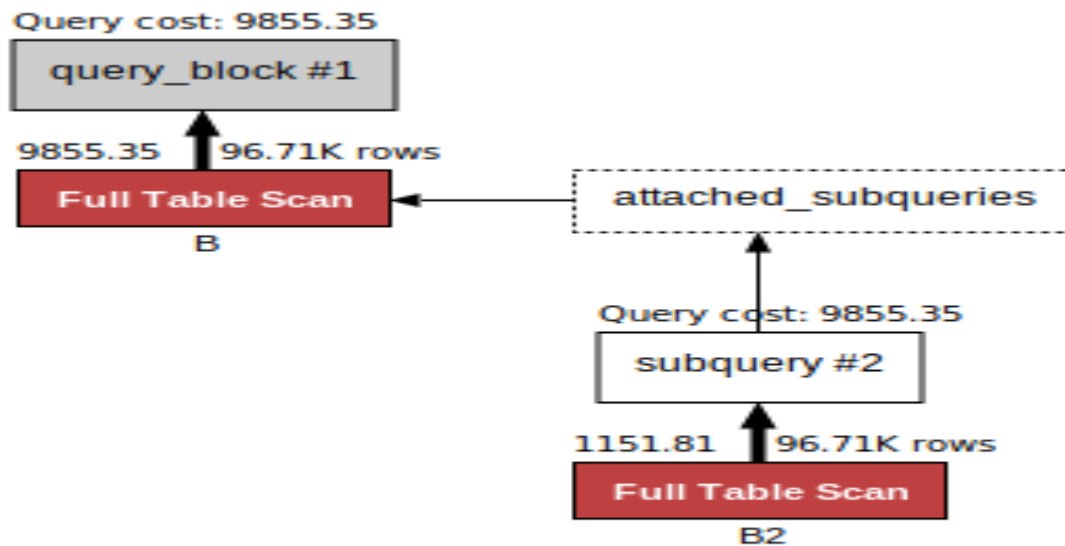
WHERE master.masterid = batting.masterid



After Optimization Query Cost 814

CREATE INDEX index_hash_masterid ON master(masterid) USING btree;

Query 11 SELECT masterid, yearId, HR FROM batting B WHERE HR = (SELECT MAX(HR) FROM batting B2 WHERE B.yearID = B2.yearID)



After Optimization Query Cost 9855

```
CREATE INDEX btree_batting_masterid_yearid_hr  
ON batting (masterid, yearid, hr) USING btree ;
```

Query 12 SELECT masterid

FROM master

WHERE masterID IN

(SELECT DISTINCT masterID

FROM pitching

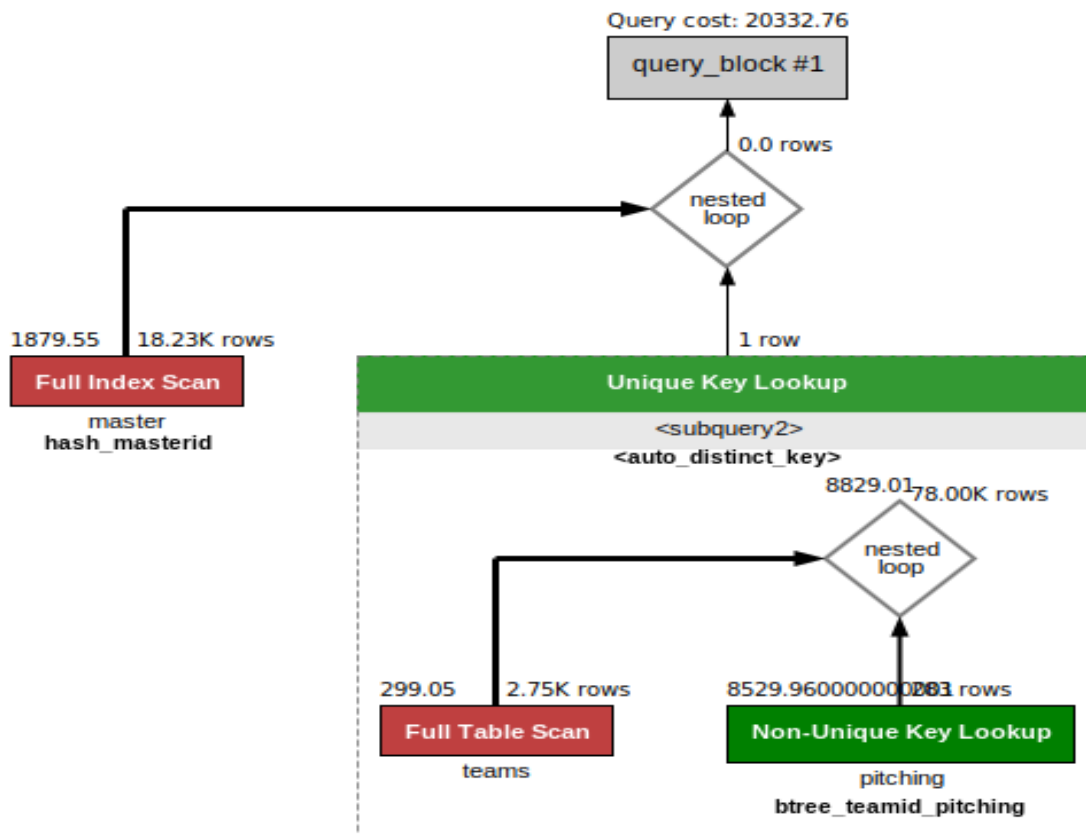
WHERE teamID IN

(SELECT DISTINCT teamID

FROM teams

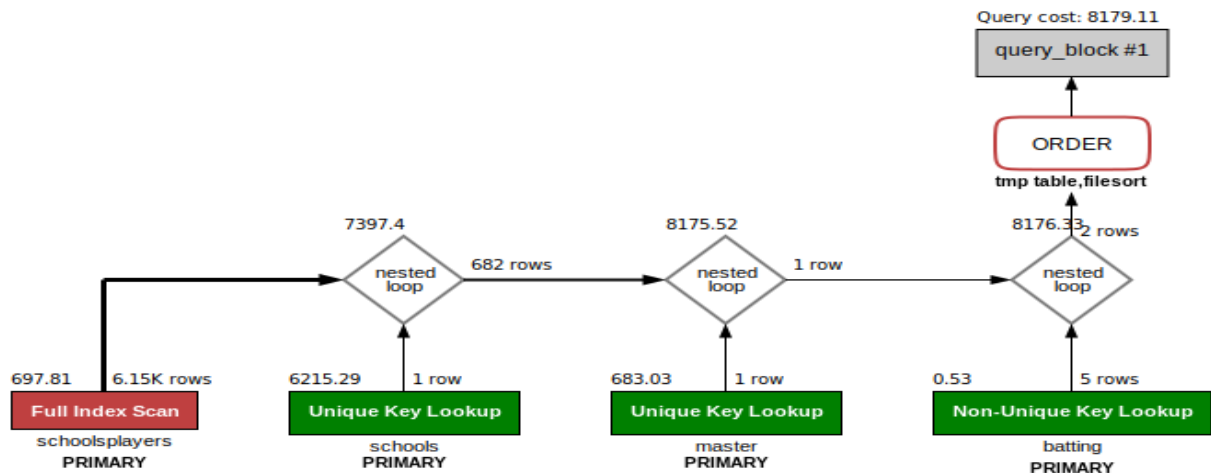
WHERE name = "Boston Red Sox"))

Query cost before optimizing is 1283064.25



Query 13 SELECT h / ab as Average, h as "Hits", ab as "At Bats", nameFirst as "First Name", nameLast as "Last Name", batting.yearID as Year FROM batting, master WHERE ab is not null and batting.masterID = master.masterID AND master.masterID IN (SELECT masterID FROM schoolsplayers WHERE schoolID in (SELECT schoolID FROM schools WHERE schoolName like "%Utah State%")) order by year

Query cost before optimization is "query_cost": "8179.11"



After Optimization Query Cost 156.29

CREATE INDEX index_masterid_master

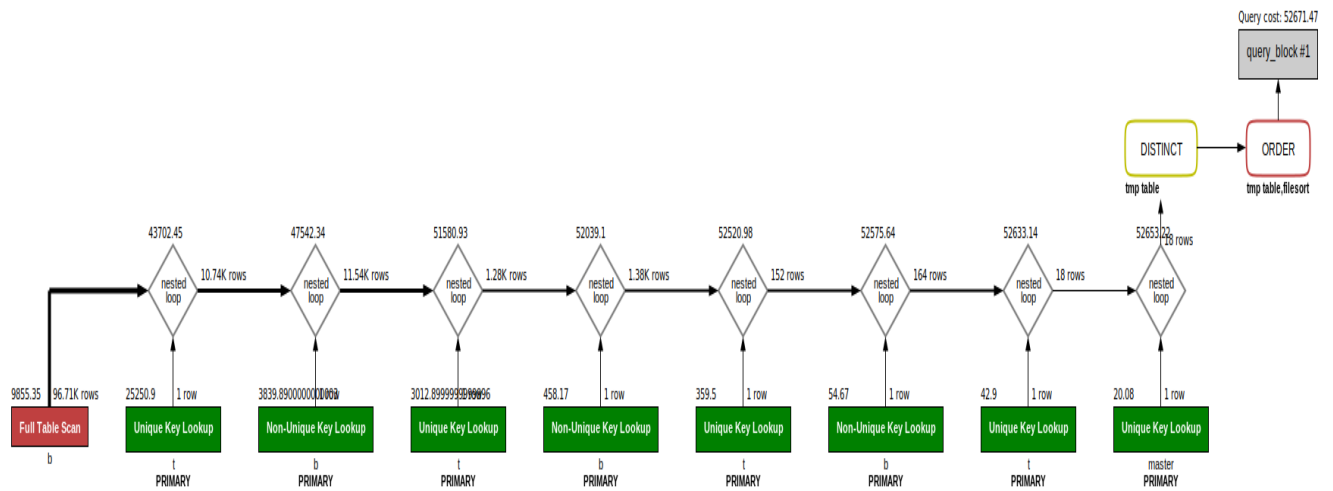
ON master(masterid) USING btree ;

CREATE INDEX index_masterid_batting

ON batting(masterid) USING btree ;

Query 14 SELECT distinct master.nameFirst as "First Name", master.nameLast as "Last Name" FROM (SELECT b.masterID as ID, b.yearID as year FROM batting b, teams t WHERE name like "%New York Yankees%" and b.teamID = t.teamID and b.yearID = t.yearID and t.lgID = b.lgID) y1, (SELECT b.masterID as ID, b.yearID as year FROM batting b, teams t WHERE name like "%New York Yankees%" and b.teamID = t.teamID and b.yearID = t.yearID and t.lgID = b.lgID) y2, (SELECT b.masterID as ID, b.yearID as year FROM batting b, teams t WHERE name like "%New York Yankees%" and b.teamID = t.teamID and b.yearID = t.yearID and t.lgID = b.lgID) y3, (SELECT b.masterID as ID, b.yearID as year FROM batting b, teams t WHERE name like "%New York Yankees%" and b.teamID = t.teamID and b.yearID = t.yearID and t.lgID = b.lgID) y4, master WHERE y1.id = y2.id and y2.id = y3.id and y3.id = y4.id and y1.year + 1 = y2.year and y2.year + 1 = y3.year and y3.year + 1 = y4.year and y4.id = master.masterID ORDER BY master.nameLast, master.nameFirst

Query Cost before optimizing is 52671



After Optimization Query Cost 3488.48

CREATE INDEX btree_teamid_teams

ON teams(teamid) USING btree ;

CREATE INDEX index_team_batting

ON batting(teamid) USING btree ;

CREATE INDEX index_yearid_teams

ON teams(yearid) USING btree ;

CREATE INDEX index_yearid_batting

ON batting(yearid) USING btree ;

CREATE INDEX index_lgid_teams

ON teams(lgid) USING btree ;