* **Slow running query on sylectus2 (the original query executes in 26.9s)**

SELECT a.PABCode, a.PABName, a.AddressLine1, b.City, b.State, a.Zip, b.PCMilerCity, a.TheirCustNum, c.ContactCode, c.FirstName, c.LastName, a.PhoneNumber, a.LocalTZ, a.CODCust, a.ARCreditHold, a.CustomerType1, a.AirportCode, d.PABSecCode, d.TSAApproved, d.CTPATApproved, d.FASTCApproved, d.FASTUApproved, a.SalesmanID, c.Status, a.PhoneIntl, a.SalesmanID2, a.MileageType

FROM PAB a, City b, PABContacts c, PABSecurity d, PABAcctNames e

WHERE (a.City = b.Code) and (a.MABCode = 6050)

and (a.PreferredContact = c.ContactCode)

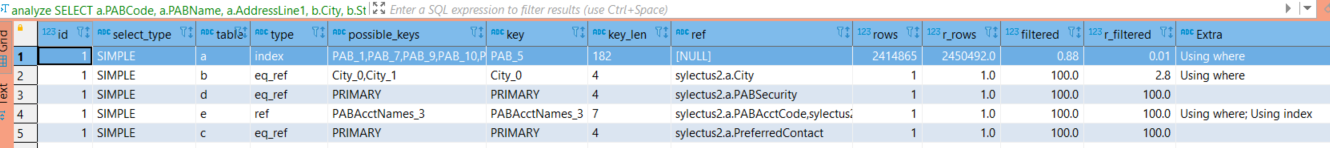
and (a.PABSecurity = d.PABSecCode)

and (a.PABAcctCode = e.PABCode)

and (a.PABAcctVer = e.PABAcctVer)

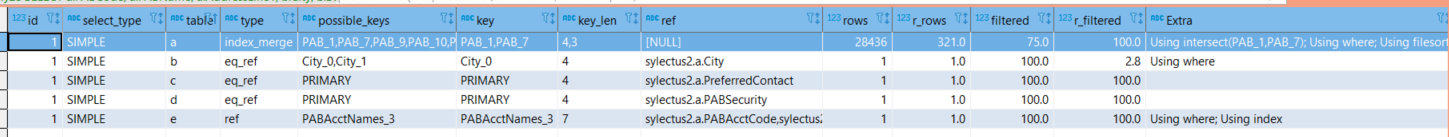
and (a.AddressLine1 LIKE '200%') and (b.State = 'NY') and (a.Active = 'Y')

ORDER BY a.PABName LIMIT 101;



**Explanation**: the index PAB\_5 is used as a driving data because of “Order by clause” => the whole index is scanned first, then filter are applied by joins and other filters.

**Action**: disable index PAB\_5 OR use hint to ignore index PAB\_5 in the query

**Result: after tuning, the query executes in 900ms**

--- If using the hint, the query is added the hint "IGNORE INDEX (PAB\_5)"

SELECT a.PABCode, a.PABName, ....

FROM PAB a IGNORE INDEX (PAB\_5)

, City b, PABContacts c, PABSecurity d, PABAcctNames e

WHERE .....and (a.AddressLine1 LIKE '200%') and (b.State = 'NY') and (a.Active = 'Y')

ORDER BY a.PABName

LIMIT 101;

--- If disable index PAB\_5 is applied (index on column Address, filter with like => this is not effectively)

10ALTER TABLE PAB DROP INDEX PAB\_5;