

## Search for merging galaxy pairs, as per the prescription in Allam et al. 2004

```
SELECT COUNT(*)
/*
g1.objectId AS g1_id,
g1.ra AS g1_ra, g1.dec AS g1_dec,
g1.modelmag_u AS g1_u,
g1.modelmag_g AS g1_g,
g1.modelmag_r AS g1_r,
g1.modelmag_i AS g1_i,
g1.modelmag_z AS g1_z,
g2.objectId AS g2_id,
g2.ra AS g2_ra, g2.dec AS g2_dec,
g2.modelmag_u AS g2_u,
g2.modelmag_g AS g2_g,
g2.modelmag_r AS g2_r,
g2.modelmag_i AS g2_i,
g2.modelmag_z AS g2_z,
g1.petroR50_r AS g1_radius,
g2.petroR50_r AS g2_radius,
n.distance AS separation
*/
FROM Galaxy g1,
JOIN Neighbors n USING (objectId)
JOIN Galaxy g2 ON (g2.objectId = N.NeighborObjID)
WHERE g1.objectId < g2.objectId
      AND N.NeighborType = 3
      AND g1.petroRad_u > 0 AND g2.petroRad_u > 0
      AND g1.petroRad_g > 0 AND g2.petroRad_g > 0
      AND g1.petroRad_r > 0 AND g2.petroRad_r > 0
      AND g1.petroRad_i > 0 AND g2.petroRad_i > 0
      AND g1.petroRad_z > 0 AND g2.petroRad_z > 0
      AND g1.petroRadErr_g > 0 AND g2.petroRadErr_g > 0
      AND g1.petroMag_g BETWEEN 16 AND 21
      AND g2.petroMag_g BETWEEN 16 AND 21
      AND g1.uMag > -9999
      AND g1.gMag > -9999
      AND g1.rMag > -9999
      AND g1.iMag > -9999
      AND g1.zMag > -9999
      AND g1.yMag > -9999
      AND g2.uMag > -9999
      AND g2.gMag > -9999
      AND g2.rMag > -9999
      AND g2.iMag > -9999
      AND g2.zMag > -9999
      AND g2.yMag > -9999
      AND abs(g1.gMag - g2.gMag) > 3
      AND (g1.petroR50_r BETWEEN 0.25*g2.petroR50_r AND 4.0*g2.petroR50_r)
      AND (g2.petroR50_r BETWEEN 0.25*g1.petroR50_r AND 4.0*g1.petroR50_r)
      AND (n.distance <= (g1.petroR50_r + g2.petroR50_r))
```

Note: this query takes more than an hour to run without a top <n> or count(\*).

Todo: some columns missing in LSST schema.

This query originally came from SDSS (<http://cas.sdss.org/dr5/en/help/docs/realquery.asp>)