Virtual Observations 2016

SDSS: 20 queries
(J.Gray, D.Slutz, A.S.Szalay, A.R.Thakar,
J. van den Berg, P.Z.Kunszt, C. Stoughton
MSR-TR-2002-1)

Q1: all galaxies without saturated pixels within 1'

```
declare @saturated bigint;                  -- initialized “saturated” flag
set    @saturated = dbo.fPhotoFlags('saturated'); -- avoids SQL2K optimizer problem
select    G.objID, GN.distance                   -- return Galaxy Object ID and
into ##results              -- angular distance (arc minutes)
from      Galaxy                  as G           -- join Galaxies with
join fGetNearbyObjEq(185,-0.5, 1) as GN         -- objects within 1’ of ra=185 & dec=-.5
on G.objID = GN.objID    -- connects G and GN
where (G.flags & @saturated) = 0               -- not saturated
order by distance                    -- sorted nearest first
```

Q2: all galaxies within magnitude & coordinate range

```
select objID                  -- Get the object identifier
into ##results
from Galaxy                   -- of all the galaxies that have
where  ra between 170 and 190 -- designated ra/dec
                                        -- (need galactic coordinates)
 and dec < 0                  -- declination less than zero.
 and g+rho between 23 and 25  -- g = blue magnitude,
                                        -- rho= 5*ln(r)
```

Q3: all galaxies brighter than some magnitude with cut-off by extinction

```
select objID                    -- find the object IDs
    into ##results
    from Galaxy                         -- join Galaxies with Extinction table
        where    r < 22                     -- where brighter than 22 magnitude
        and  reddening_r> 0.175     -- extinction more than 0.175
```

Q4: find “large” galaxies

```
select ObjID                -- put the qualifying galaxies in a table
into ##results
from Galaxy                 -- select galaxies
where r + rho < 24              -- brighter than magnitude 24 in the red spectral band

   and  isoA_r between 30 and 60  -- major axis between 30" and 60"

   and  (power(q_r,2) + power(u_r,2)) > 0.25 -- square of ellipticity is > 0.5 squared.
```

Q5: find elliptical galaxies

```
declare @binned         bigint;                        -- initialized “binned” literal
set     @binned =       dbo.fPhotoFlags('BINNED1') +   -- avoids SQL2K optimizer problem

                        dbo.fPhotoFlags('BINNED2') +

                        dbo.fPhotoFlags('BINNED4') ;
declare @blended        bigint;                        -- initialized “blended” literal
set     @blended =      dbo.fPhotoFlags('BLENDED');    -- avoids SQL2K optimizer problem
declare @noDeBlend      bigint;                        -- initialized “noDeBlend” literal
set     @noDeBlend =    dbo.fPhotoFlags('NODEBLEND'); -- avoids SQL2K optimizer problem
declare @child          bigint;                        -- initialized “child” literal
set     @child =    dbo.fPhotoFlags('CHILD');          -- avoids SQL2K optimizer problem
declare @edge           bigint;                        -- initialized “edge” literal
set     @edge =     dbo.fPhotoFlags('EDGE');   -- avoids SQL2K optimizer problem
declare @saturated      bigint;                        -- initialized “saturated” literal
set     @saturated =    dbo.fPhotoFlags('SATURATED'); -- avoids SQL2K optimizer problem
select objID
into ##results
from Galaxy as G                  -- count galaxies
where    lDev_r > 1.1 * lExp_r    -- red DeVaucouleurs fit likelihood greater than disk fit

   and  lExp_r > 0                -- exponential disk fit likelihood in red band > 0

   -- Color cut for an elliptical galaxy courtesy of James Annis of Fermilab

   and (G.flags & @binned) > 0  and (G.flags & ( @blended + @noDeBlend + @child)) != @blended and (G.flags & (@edge + @saturated)) = 0  and (G.petroMag_i > 17.5)

   and (G.petroMag_r > 15.5 OR G.petroR50_r > 2)  and (G.petroMag_r < 30 and G.g < 30 and G.r < 30 and G.i < 30)

   and ((G.petroMag_r-G.reddening_r) < 19.2)

   and (   (    ((G.petroMag_r - G.reddening_r) < (13.1 +  -- deRed_r < 13.1 +

                                          (7/3)*(G.g - G.r) +      -- 0.7 / 0.3 * deRed_gr

                                    4 *(G.r - G.i) -4 * 0.18 )) -- 1.2 / 0.3 * deRed_ri

               and (( G.r - G.i - (G.g - G.r)/4 - 0.18) BETWEEN -0.2 AND  0.2 )

              )

            or

            (    (( G.petroMag_r - G.reddening_r) < 19.5 )      -- deRed_r < 19.5 +

               and (( G.r - G.i -(G.g - G.r)/4 -.18) >          -- cperp = deRed_ri

                          (0.45 - 4*( G.g - G.r)))                    -- 0.45 - deRed_gr/0.25

               and ((G.g - G.r) > ( 1.35 + 0.25 *(G.r - G.i)))

         )   )
```

Q6: deblend galaxies blended with a star

```
select G.ObjID, G.u, G.g, G.r, G.i, G.z      -- output galaxy and magnitudes.
into   ##results
from     galaxy G, star S                 -- for each galaxy
where    G.parentID > 0                   -- galaxy has a “parent”

    and  G.parentID = S.parentID              -- star has the same parent
```

Q7: find “exotic” stars by colors

```
select cast(round((u-g),0) as int) as UG,
    cast(round((g-r),0) as int) as GR,
    cast(round((r-i),0) as int) as RI,
    cast(round((i-z),0) as int) as IZ,
    count(*)                as pop
into  ##results
from  star
where (u+g+r+i+z) < 150   -- exclude bogus magnitudes (== 999)
group by    cast(round((u-g),0) as int), cast(round((g-r),0) as int),

    cast(round((r-i),0) as int), cast(round((i-z),0) as int)
order by count(*)

delete ##results where pop > 500
```

Q8: find objects with unclassified spectra

```
declare @unknown bigint;                  -- initialized “binned” literal

set   @unknown = dbo.fSpecClass('UNKNOWN')

select specObjID

into   ##results

from   SpecObj

             where  SpecClass = @unknown
```

Q9: find QSO with cut by line width and redshift

```
declare         @qso     int;
set         @qso = dbo.fSpecClass('QSO') ;
declare         @hiZ_qso int;
set         @hiZ_qso =dbo.fSpecClass('HIZ-QSO');
select        s.specObjID,                            -- object id

            max(l.sigma*300000.0/l.wave) as veldisp,      -- velocity dispersion

            avg(s.z) as z                             -- redshift
into    ##results
from      SpecObj s, specLine l          -- from the spectrum table and lines
where   s.specObjID=l.specObjID              -- line belongs to spectrum of this obj

        and ( (s.specClass = @qso) or                 -- quasar

                   (s.specClass = @hiZ_qso))               -- or hiZ_qso.

        and  s.z between 2.5 and 2.7                  -- redshift of 2.5 to 2.7

        and  l.sigma*300000.0/l.wave >2000.0   -- convert sigma to km/s

        and  s.zConf > 0.9                       -- high confidence on redshift estimate
group by s.specObjID
```

Q10-1: find galaxies with cut-out by spectral line

```
select G.ObjID          -- return qualifying galaxies

 into  ##results

 from    Galaxy    as G,          -- G is the galaxy
     SpecObj    as S,          -- S is the spectra of galaxy G

     SpecLine   as L,          -- L is a line of S
     specLineNames as LN -- the names of the lines

 where G.ObjID = S.ObjID   -- connect the galaxy to the spectrum

      and S.SpecObjID = L.SpecObjID      -- L is a line of S.

      and L.LineId = LN.value    -- L is the H alpha line

      and LN.name =  'Ha_6565'

      and L.ew > 40
```

Q10-2: find galaxies with cut-out by spectral line

```
select G.ObjID          -- return qualifying galaxies
 into  ##results
 from     Galaxy    as G,          -- G is the galaxy
     SpecObj    as S,         -- S is the spectra of galaxy G
     SpecLine   as L1,        -- L1 is a line of S
     SpecLine   as L2,        -- L2 is a second line of S
     specLineNames as LN1,         -- the names of the lines (Halpha)
     specLineNames as LN2          -- the names of the lines (Hbeta)
where G.ObjID = S.ObjID            -- connect the galaxy to the spectrum
  and S.SpecObjID = L1.SpecObjID      -- L1 is a line of S.
  and S.SpecObjID = L2.SpecObjID      -- L2 is a line of S.  and L1.LineId = LN1.LineId
  and L1.LineId = LN1. value
  and LN1.name =  'Ha_6565'           -- L1 is the H alpha line
  and L2.LineId = LN2.value        -- L2 is the H alpha line
  and LN2.name =  'Hb_4863'        --
  and L1.ew > 200             -- BIG Halpha
  and L2.ew > 10              -- significant Hbeta emission line
  and L2.ew * 20 < L1.ew           -- Hbeta is comparatively small
```

Q11: find galaxies of certain type with spectral anomalies

```
select      distinct G.ObjID        -- return qualifying galaxies
into   ##results
from   Galaxy      as G,            -- G is the galaxy
       SpecObj              as S,          -- S is the spectra of galaxy G

       SpecLine      as L,          -- L is a line of S

       specLineNames as LN,                -- the type of line

       XCRedshift    as XC          -- the template cross-correlation
where G.ObjID = S.ObjID             -- connect galaxy to the spectrum

   and S.SpecObjID = L.SpecObjID           -- L is a line of S

   and S.SpecObjID = XC.SpecObjID          -- CC is a cross-correlation with templates

   and XC.tempNo = 8                -- Template('Elliptical') -- CC says "elliptical"

   and L.LineID = LN.value          -- line type is found

   and LN.Name = 'UNKNOWN'                 --       but not identified

   and L.ew > 10                    -- a prominent (wide) line

   and S.SpecObjID not in (         -- insist that there are no other lines
select S.SpecObjID             -- that are know and are very close to this one
from   SpecLine      as L1,  -- L1 is another line
              specLineNames as LN1
where S.SpecObjID = L1.SpecObjID           -- for this object

   and abs(L.wave - L1.wave) <.01          -- at nearly the same wavelength

   and L1.LineID = LN1.value               -- line found and

   and LN1.Name != 'UNKNOWN'               --       it IS identified
)
```

Q12: create density map for galaxies

```
--- First find the grided galaxy count (with the color cut)
--- In local tangent plane, ra/cos(dec) is a “linear” degree.
declare @LeftShift16 bigint;          -- used to convert 20-deep htmIds to 6-deep IDs
set     @LeftShift16 = power(2,28);
select cast((ra/cos(cast(dec*30 as int)/30.0))*30 as int)/30.0 as raCosDec,
         cast(dec*30 as int)/30.0                                as dec,
         count(*)                                                as pop
into ##GalaxyGrid
from   Galaxy as G ,
         dbo.fHTM_Cover('CONVEX J2000 6 6 175 -5 175 5 185 5 185 -5') as T
where  htmID between T.HTMIDstart*@LeftShift16 and T. HTMIDend*@LeftShift16
   and  ra between 175 and 185
   and  dec between -5 and 5
   and  u-g > 1
   and  r < 21.5
group by  cast((ra/cos(cast(dec*30 as int)/30.0))*30 as int)/30.0,
             cast(dec*30 as int)/30.0

--- now build mask grid.
select cast((ra/cos(cast(dec*30 as int)/30.0))*30 as int)/30.0 as raCosDec,
cast(dec*30 as int)/30.0                                as dec,
count(*)                                                as pop
into ##MaskGrid
from  photoObj as PO,
      dbo.fHTM_Cover('CONVEX J2000 6 6 175 -5 175 5 185 5 185 -5') as T,
      photoType as PT
where htmID between T.HTMIDstart*@LeftShift16 and T. HTMIDend*@LeftShift16
  and ra between 175 and 185
  and dec between -5 and 5
  and PO.type = PT.value
  and PT.name in (‘COSMIC_RAY’, ‘DEFECT’, ‘GHOST’, ‘TRAIL’, ‘UNKNOWN’)
group by  cast((ra/cos(cast(dec*30 as int)/30.0))*30 as int)/30.0,
          cast(dec*30 as int)/30.0
```

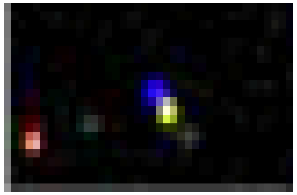
Q13: create density map for HTM with color cut

```
declare @RightShift12 bigint;
set     @RightShift12 = power(2,24);
select (htmID /@RightShift12) as htm_8, -- group by 8-deep HTMID (rshift HTM by 12)
avg(ra)  as ra,
avg(dec) as [dec],
count(*) as pop          -- return center point and count for display

  into  ##results                    -- put the answer in the results set.

  from       Galaxy                         -- only look at galaxies

  where  (0.7*u - 0.5*g - 0.2*i) < 1.25  -- meeting this color cut

      and  r < 21.75                 -- fainter than 21.75 magnitude in red band.

   group by (htmID /@RightShift12)    -- group into 8-deep HTM buckets..HTM buckets
```

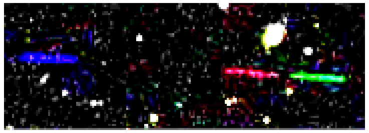
Q14: find stars with magnitude variations

```
declare @star int;                          -- initialized “star” literal
set     @star = dbo.fPhotoType('Star');           -- avoids SQL2K optimizer problem
select s1.objID as ObjID1, s2.objID as ObjID2 -- select object IDs of star and its pair
into  ##results
from   star      as   s1,                   -- the primary star

           photoObj  as   s2,                     -- the second observation of the star

           neighbors as   N                       -- the neighbor record
where s1.objID = N.objID                    -- insist the stars are neighbors

   and s2.objID = N.neighborObjID           -- using precomputed neighbors table

   and distanceMins < 0.5/60                      -- distance is ½ arc second or less

   and s1.run != s2.run                           -- observations are two different runs

   and s2.type = @star                            -- s2 is indeed a star

   and  s1.u between 1 and 27                     -- S1 magnitudes are reasonable

   and  s1.g between 1 and 27

   and  s1.r between 1 and 27

   and  s1.i between 1 and 27

   and  s1.z between 1 and 27

   and  s2.u between 1 and 27                     -- S2 magnitudes are reasonable.

   and  s2.g between 1 and 27

   and  s2.r between 1 and 27

   and  s2.i between 1 and 27

   and  s2.z between 1 and 27

   and (                                    -- and one of the colors is  different.

             abs(S1.u-S2.u) > .1 + (abs(S1.Err_u) + abs(S2.Err_u))

         or abs(S1.g-S2.g) > .1 + (abs(S1.Err_g) + abs(S2.Err_g))

         or abs(S1.r-S2.r) > .1 + (abs(S1.Err_r) + abs(S2.Err_r))

         or abs(S1.i-S2.i) > .1 + (abs(S1.Err_i) + abs(S2.Err_i))

         or abs(S1.z-S2.z) > .1 + (abs(S1.Err_z) + abs(S2.Err_z))

       )
```

Q15: find suspected asteroids



```
-- slow mooving
select  objID,                                                  -- return object ID
        sqrt( power(rowv,2) + power(colv, 2) ) as velocity,  -- velocity
        dbo.fGetUrlExpId(objID) as Url                          -- url of image to examine it.
 into  ##results
 from   PhotoObj                                                -- check each object.
 where (power(rowv,2) + power(colv, 2)) between 50 and 1000 -- square of velocity
   and rowv >= 0 and colv >=0                                -- negative values indicate error
```



```
--- fast mooving
select r.objID as rId, g.objId as gId,
r.run, r.camcol,
r.field as field, g.field as gField,
        r.ra as ra_r, r.dec as dec_r,
g.ra as ra_g, g.dec as dec_g, --(note acos(x) ~ x for x~1)
        sqrt(power(r.cx-g.cx,2)+power(r.cy-g.cy,2)+power(r.cz-g.cz,2)) *
(180*60/PI()) as distance,
       dbo. fGetUrlExpId (r.objID) as rURL,      -- returns URL for image of object
       dbo. fGetUrlExpId (g.objID) as gURL
from   PhotoObj r, PhotoObj g
where  r.run = g.run and r.camcol=g.camcol -- same run and camera column
  and abs(g.field-r.field) <= 1               -- adjacent fields
        -- the red selection criteria
  and ((power(r.q_r,2) + power(r.u_r,2)) > 0.111111 )  -- q/u is ellipticity
  and r.fiberMag_r between 6 and 22
  and r.fiberMag_r < r.fiberMag_u
  and r.fiberMag_r < r.fiberMag_g
  and r.fiberMag_r < r.fiberMag_i
  and r.fiberMag_r < r.fiberMag_z
  and r.parentID=0
  and r.isoA_r/r.isoB_r > 1.5
  and r.isoA_r > 2.0
      -- the green selection criteria
  and ((power(g.q_g,2) + power(g.u_g,2)) > 0.111111 )
  and g.fiberMag_g between 6 and 22
  and g.fiberMag_g < g.fiberMag_u
  and g.fiberMag_g < g.fiberMag_r
  and g.fiberMag_g < g.fiberMag_i
  and g.fiberMag_g < g.fiberMag_z
  and g.parentID=0
  and g.isoA_g/g.isoB_g > 1.5
  and g.isoA_g > 2.0
-- the match-up of the pair --(note acos(x) ~ x for x~1)
  and sqrt(power(r.cx-g.cx,2)+power(r.cy-g.cy,2)+power(r.cz-g.cz,2))*(180*60/pi()) < 4.0
  and abs(r.fiberMag_r-g.fiberMag_g)< 2.0
```

Q16: find all objects similar in color to QSO

```
select count(*)                    as 'total',
sum( case when (type=3) then 1 else 0 end)        as 'Galaxies',
sum( case when (type=6) then 1 else 0 end)        as 'Stars',
sum( case when (type not in (3,6)) then 1 else 0 end) as 'Other'
from    PhotoPrimary              -- for each object
 where (( u - g > 2.0) or (u > 22.3) )    -- apply the quasar color cut.
   and ( i between 0 and 19 )
   and ( g - r > 1.0 )
   and ( (r - i < 0.08 + 0.42 * (g - r - 0.96)) or (g - r > 2.26 ) )
   and ( i - z < 0.25 )
```

Q17: find binaries with possible WD

```
declare @star int;                   -- initialized “star” literal
set     @star = dbo.fPhotoType('Star');   -- avoids SQL2K optimizer problem
select   s1.objID as s1, s2.objID as s2   -- return star pairs
into   ##results
from Star       S1,                  -- S1 is the white dwarf

         Neighbors  N,               -- N is the precomputed neighbors links

         Star       S2               -- S2 is the second star

    where S1.objID = N. objID             -- S1 and S2 are neighbors-within 30 arc sec

         and S2.objID = N.NeighborObjID

         and N.NeighborObjType = @star        -- and S2 is a star

         and N.DistanceMins < .05             -- the 3 arcsecond test

         and (S1.u - S1.g) < 0.4          -- and S1 meets Paul Szkody’s color cut for

         and (S1.g - S1.r) < 0.7              -- white dwarfs.

         and (S1.r - S1.i) > 0.4

         and (S1.i - S1.z) > 0.4
```

Q18: find candidates for strong gravitational lensing

```
select  distinct P.ObjID              -- count distinct cases (will get min objid)

into ##results                        -- oid compare gets minimum object

  From   photoPrimary   P,                -- P is the primary object
         Neighbors      N,            -- N is the neighbor link
         photoPrimary   L             -- L is the lens candidate of P

where P.ObjID = N.ObjID                   -- N is a neighbor record

    and L.ObjID = N.NeighborObjID             -- L is a neighbor of P

    and P.ObjID < L.ObjID                 -- avoid duplicates

    and abs((P.u-P.g)-(L.u-L.g))<0.05         -- L and P have similar spectra.

    and abs((P.g-P.r)-(L.g-L.r))<0.05

    and abs((P.r-P.i)-(L.r-L.i))<0.05

    and abs((P.i-P.z)-(L.i-L.z))<0.05
```

Q19: find QSO-galaxy pairs

```
select Q.ObjID as Quasar_candidate_ID, G.ObjID as Galaxy_ID
into ##results
from  SpecObj          as Q,         -- Q is the specObj of the quasar candidate

           Neighbors   as N,         -- N is the Neighbors list of Q

           Galaxy      as G,         -- G is the nearby galaxy

           SpecClass   as SC,

           SpecLine    as L,         -- L is the broad line we are looking for

           SpecLineNames    as LN
where Q.SpecClass  = SC.class

   and SC.name  in ('QSO', 'HIZ_QSO')    -- Spectrum says "QSO"

   and Q.SpecObjID = L.SpecObjID     -- L is a spectral line of Q.

   and L.LineID = LN.value      -- line found and

   and LN.Name != 'UNKNOWN'          --      not not identified

   and L.ew < -10          -- but its a prominent absorption line

   and Q.ObjID = N.ObjID       -- N is a neighbor record

   and G.ObjID = N.NeighborObjID     -- G is a neighbor of Q

   and N.distanceMins < (10.0/60.0) -- and it is within 10 arcseconds of the Q.
```

Q20: make density map for galaxies around LRGs

```
declare @binned       bigint;                        -- initialized “binned” literal
set     @binned =     dbo.fPhotoFlags('BINNED1') +  -- avoids SQL2K optimizer problem
dbo.fPhotoFlags('BINNED2') +dbo.fPhotoFlags('BINNED4') ;
declare @blended      bigint;                        -- initialized “blended” literal
set     @blended =    dbo.fPhotoFlags('BLENDED');   -- avoids SQL2K optimizer problem
declare @noDeBlend    bigint;                        -- initialized “noDeBlend” literal
set     @noDeBlend =  dbo.fPhotoFlags('NODEBLEND'); -- avoids SQL2K optimizer problem
declare @child        bigint;                        -- initialized “child” literal
set     @child =      dbo.fPhotoFlags('CHILD');     -- avoids SQL2K optimizer problem
declare @edge         bigint;                        -- initialized “edge” literal
set     @edge =       dbo.fPhotoFlags('EDGE');      -- avoids SQL2K optimizer problem
declare @saturated    bigint;                        -- initialized “saturated” literal
set     @saturated=   dbo.fPhotoFlags('SATURATED'); -- avoids SQL2K optimizer problem
select  G.objID, count(*) as pop
into  ##results
from   Galaxy     as G,                      -- first gravitational lens candidate
       Neighbors  as N,                      -- precomputed list of neighbors
       Galaxy     as U,                      -- a neighbor galaxy of G
       PhotoZ     as GpZ,                    -- photoZ of first galaxy
       PhotoZ     as NpZ                     -- photoZ of second galaxy
where  G.objID = N.objID             -- connect G and U via the neighbors table
and U.objID = N.neighborObjID        -- so that we know G and U are within
and N.objID < N.neighborObjID        -- 30 arcseconds of one another.
and G.objID = GpZ.objID              -- join to photoZ of G
and U.objID = NpZ.objID              -- join to photoZ of N
and G.ra between 160 and 170         -- restrict search to a part of the sky
and G.dec between -5 and 5           -- that is in database
and abs(GpZ.Z - NpZ.Z) < 0.05 -- restrict the photoZ differences
-- Color cut for an BCG courtesy of James Annis of Fermilab

and (G.flags & @binned) > 0  and (G.flags & ( @blended + @noDeBlend + @child)) != @blended

and (G.flags & (@edge + @saturated)) = 0  and  G.petroMag_i > 17.5  and (G.petroMag_r > 15.5 or G.petroR50_r > 2)

     and (G.g >0 and G.r >0 and G.i >0)  and ( (   ((G.petroMag_r-G.reddening_r)   < 19.2)

               and ((G.petroMag_r - G.reddening_r)

                              < (12.38 + (7/3)*( G.g-  G.r ) + 4 *( G.r - G.i ) ) )

               and ((abs( G.r - G.i - (G.g - G.r )/4 - 0.18 )) < 0.2)

               and ((G.petroMag_r - G.reddening_r +

                              2.5*Log10(2*pi()*G.petroR50_r* G.petroR50_r )) < 24.2   )

               )

           or (  ((G.petroMag_r - G.reddening_r)       < 19.5                         )

                and ((G.r - G.i - (G.g - G.r)/4 - 0.18 ) > (0.45 - 4*( G.g- G.r ) )     )

                and ((G.g - G.r ) > ( 1.35 + 0.25 *( G.r - G.i ) )                     )

                and ((G.petroMag_r - G.reddening_r  +

                              2.5*Log10(2*pi()*G.petroR50_r* G.petroR50_r )) < 23.3   )

             ) )
group by G.objID
```