

Exercise 6: Write a query that selects the Hipparcos identifier, the B and V-band apparent magnitudes, and the variability period. Only select sources whose V-band magnitude is greater than 11, and whose variability type is 'P'. Order the results by the variability period.

Solution:

```
SELECT HIP, Bmag, Vmag, Per  
FROM photometry  
WHERE Vmag > 11  
AND Hvar = 'P'  
ORDER BY Per;
```

Returning unique results: SQL DISTINCT

When you sort the results you may realize that our database has duplicate entries. To return only unique rows use the DISTINCT keyword.

for example:

```
SELECT DISTINCT HIP, Bmag, Vmag, Per  
FROM photometry  
WHERE Per != 0  
AND Bmag != 0  
AND Bmag < 10  
ORDER BY Vmag;
```

Counting Returned Values

You can count the number of results returned with the SQL COUNT command

for example, to see how many hipparcos identifiers are returned without using SQL DISTINCT:

```
SELECT COUNT(HIP)
FROM photometry
WHERE Per != 0
AND Bmag != 0
AND Bmag < 10
ORDER BY Vmag;
```

To see how many unique hipparcos identifiers are returned, add a distinct to inside the SQL COUNT (otherwise SQL DISTINCT is applied after SQL COUNT):

```
SELECT COUNT(DISTINCT HIP)
FROM photometry
WHERE Per != 0
AND Bmag != 0
AND Bmag < 10
ORDER BY Vmag;
```

Searching across multiple tables

```
SELECT <column1_name>, <column2_name>, ..., <columnN_name>  
FROM <table1_name>  
JOIN <table2_name> ON <condition>;
```

for example:

```
SELECT data.HIP, pmRA, pmDE, Bmag, Vmag  
FROM data  
JOIN photometry ON data.HIP = photometry.HIP;
```

Note that when you select a column name that appears in both tables you must specify which table you want it returned from

Note that WHERE, AND, and ORDER BY conditions can be applied to the query after the JOIN, for example:

```
SELECT data.HIP, pmRA, pmDE, Bmag, Vmag  
FROM data  
JOIN photometry ON data.HIP = photometry.HIP  
WHERE Vmag > 11  
AND Bmag > 11  
ORDER BY Vmag;
```