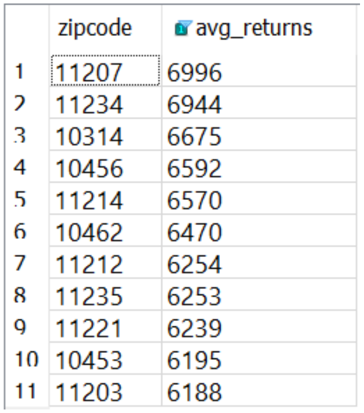
HOMEWORK #5 SOLUTION

Problem 1 (3 points):
Find the 11 zip codes whose annual average tax returns are less than 7000 returns and also greater than 6000 returns. Submit your SQL code along with its resulting output.

Code:

```
SELECT
    zipcode,
    ROUND(AVG(return_count),0) as avg_returns
FROM irs_nyc_tax_returns
GROUP BY
    zipcode
HAVING
    avg_returns < 7000
    AND avg_returns > 6000
ORDER BY
    avg_returns DESC;
```

Result:



	zipcode	avg_returns
1	11207	6996
2	11234	6944
3	10314	6675
4	10456	6592
5	11214	6570
6	10462	6470
7	11212	6254
8	11235	6253
9	11221	6239
10	10453	6195
11	11203	6188

(NOTE: use of ORDER BY and ROUND() are optional)

Problem 2 (4 points):
Create a summarized view of the demographic data available in the US census table by bucketing up the detailed columns into the following supergroups: male, female, age under 19, age over 70, age betw 20 – 70, european ethnicity, african ethincity, asian-pacific ethnicity, hispanic-latino descent, other ethnicities. Order the results by total population for the zip. Limit your result to just the top 5 zip codes. Submit your SQL code along with its resulting output.

Code:

```
SELECT
    cen.zipcode,
/* Gender */
    SUM(cen.Gender_Male) AS gen_male_cnt,
    SUM(cen.Gender_Female) AS gen_female_cnt,
/* ages */
    SUM(cen.Age_under5) + SUM(cen.Age_5to9) + SUM(cen.Age_10to14) + SUM(cen.Age_15to19) AS
age_under19_cnt,
    SUM(cen.Age_over84) + SUM(cen.Age_80to84) + SUM(cen.Age_75to79) + SUM(cen.Age_70to74) AS
age_over70_cnt,
    SUM(Age_all)
        - ( --child count
            SUM(cen.Age_under5) + SUM(cen.Age_5to9) + SUM(cen.Age_10to14) + SUM(cen.Age_15to19)
            )
        - ( --senior citizen count
            SUM(cen.Age_over84) + SUM(cen.Age_80to84) + SUM(cen.Age_75to79) + SUM(cen.Age_70to74)
            )
     AS age_20to70_cnt,
/* ethnicity */
     SUM(cen.Ethnicity_White) AS eth_euro_cnt,
     SUM(cen.Ethnicity_AfricanAmerican) AS eth_african_cnt,
     SUM(cen.Ethnicity_Asian) + SUM(cen.Ethnicity_PacificIslander) AS eth_asiapac_cnt,
     SUM(cen.Ethnicity_NativeAmerican) + SUM(cen.Ethnicity_Other) AS eth_other_cnt,
     SUM(cen.Ethnicity_All_HispancLatino_Descent) AS eth_hislat_descent_cnt,
/* total pop */
```

```
    SUM(cen.Age_all) as total_pop_cnt
FROM
    nyc_census_data cen
GROUP BY
    cen.zipcode
ORDER BY
    total_pop_cnt DESC
LIMIT 5;
```

RESULT:

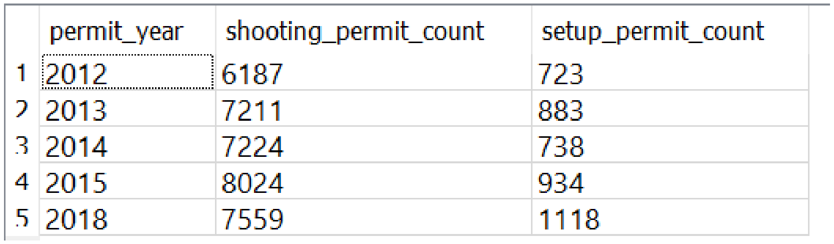
zipcode	gen_male_cnt	gen_female_cnt	age_under18_cnt	age_over70_cnt	age_20to70_cnt	eth_euro_cnt	eth_african_cnt	eth_asiapac_cnt	eth_other_cnt	eth_hislat_descent_cnt	total_pop_cnt
11368	58421	51510	30657	5469	73805	40202	14296	12176	50269	81093	109931
11226	45453	56119	27107	6301	68164	11406	79207	4002	10496	17409	101572
11373	51175	49645	22566	6904	71350	29962	2676	49546	23424	42775	100820
11220	50308	49290	26791	6152	66655	33150	3950	42570	23813	40653	99598
11385	48510	50082	25174	6844	66574	67577	4483	7399	23276	43166	98592

Problem 3 (3 points):
Find the annual count of distinct filming permits for shooting versus setup/non-shooting event types. Display each permit type grouping as its own column in the result (HINT: you should have 3 columns: year, shooting permit counts, and setup/non-shooting permit counts). Submit your SQL code along with its resulting output.

Code:

```
SELECT
    STRFTIME('%Y',StartDateTime) as
permit_year,
    COUNT(DISTINCT CASE
            WHEN EventType IN ("Shooting
Permit","DCAS Prep/Shoot/Wrap Permit") THEN
EventID
            ELSE NULL
        END) AS shooting_permit_count,
    COUNT(DISTINCT CASE
            WHEN EventType NOT IN ("Shooting
Permit","DCAS Prep/Shoot/Wrap Permit") THEN
EventID
            ELSE NULL
        END) AS setup_permit_count
FROM nyc_film_permits
GROUP BY
    permit_year;
```

Result:



	permit_year	shooting_permit_count	setup_permit_count
1	2012	6187	723
2	2013	7211	883
3	2014	7224	738
4	2015	8024	934
5	2018	7559	1118