**Assignment 06**

***Place your query after each question***

The city’s Crime Analysis unit has submitted the following data requests. Provide the SQL statements using subqueries to satisfy the requests. Test the statements and show execution results.

1. List the name of each officer who has reported more than the average number of crimes officers have reported.

SELECT co.officer\_id, o.last, o.first

FROM officers o JOIN crime\_officers co

ON o.officer\_id = co.officer\_id

GROUP BY co.officer\_id, o.last, o.first

HAVING COUNT(\*) > (SELECT COUNT(\*) / COUNT(DISTINCT officer\_id)

FROM crime\_officers);

Graphical user interface, application, Word

Description automatically generated

1. List the names of all criminals who have committed less than average number of crimes and aren’t listed as violent offenders.

SELECT cls.criminal\_id, cls.first, cls.last

FROM crimes c JOIN criminals cls

ON cls.criminal\_id = c.criminal\_id

WHERE cls.v\_status = 'N'

GROUP BY cls.criminal\_id, cls.first, cls.last

HAVING COUNT(\*) < (SELECT COUNT(\*) / COUNT(DISTINCT criminal\_id) FROM crimes);

Graphical user interface, application

Description automatically generated

1. List appeal information for each appeal that has a less than average number of days between the filing and hearing dates.

SELECT crime\_id, appeal\_id, filing\_date, hearing\_date, status

FROM appeals

GROUP BY crime\_id, appeal\_id, filing\_date, hearing\_date, status

HAVING AVG(DATEDIFF(DAY, filing\_date, hearing\_date)) <

(SELECT AVG(DATEDIFF(DAY, filing\_date, hearing\_date))

FROM appeals);

Table

Description automatically generated

1. List the names of probation officers who have had a less than average number of criminals assigned.

SELECT p.last, p.first

FROM sentences s JOIN prob\_officers p

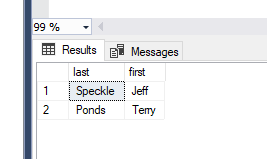
ON s.prob\_id = p.prob\_id

GROUP BY p.last, p.first

HAVING COUNT(p.prob\_id) <

(SELECT COUNT(\*)/COUNT(DISTINCT prob\_id)

FROM sentences);



1. List each crime that has had the highest number of appeals recorded.

SELECT cc.crime\_code, cc.crime\_id, cc.charge\_status, a.filing\_date

FROM appeals a JOIN crime\_charges cc

ON a.crime\_id = cc.crime\_id

WHERE a.crime\_id

IN (SELECT a.crime\_id

FROM

(SELECT MAX(a.appealCount) as maxAppealCount, a.crime\_id

FROM

(SELECT crime\_id, COUNT(appeal\_id) AS appealCount

FROM appeals

GROUP BY crime\_id) a

GROUP BY crime\_id) a );

Table

Description automatically generated

1. Display code description that has crime code from 302 to 305, display both columns.

SELECT crime\_code, code\_description

FROM crime\_codes

WHERE crime\_code IN (302,305);

Graphical user interface, application

Description automatically generated

1. Add 10 days to the pay due date of the crime\_charges table then display the result, name the column New\_Due\_date.

SELECT pay\_due\_date, DATEADD(DAY, 10, pay\_due\_date) AS "New\_Due\_date"

FROM crime\_charges;

Graphical user interface, table

Description automatically generated with medium confidence

1. Display crime id and calculate how many days between filing and hearing dates, name this column “Days to Hearing”.

SELECT crime\_id, DATEDIFF(DAY, filing\_date, hearing\_date) AS "Days to Hearing"

FROM appeals;

Table

Description automatically generated

1. Write a query to display the probation officers’ id and their first and last names separated by a comma.

SELECT officer\_id, CONCAT(first, ', ', last) AS "Officer Name"

FROM officers;

Table

Description automatically generated

1. Display the lowest fine amount charged to a crime. Give the column an alias name “Lowest Fine Amount”.

SELECT MIN(fine\_amount ) AS "Lowest Fine Amount"

FROM crime\_charges

WHERE crime\_code IN

(SELECT crime\_code

FROM crime\_codes);

Graphical user interface, table

Description automatically generated