# SQL QUERIES

1. Give branchwise Case Clearance Rate for the year 2024.

CREATE VIEW branch\_cases AS

SELECT branch\_id, case\_id, Status

FROM personnel NATURAL JOIN assigned\_case NATURAL JOIN cases;

SELECT r1.branch\_id, (100\*r1.closed\_cases::FLOAT/r2.total\_cases)::NUMERIC(4,2) AS Case\_Clearance\_Rate

FROM

(SELECT branch\_id, COUNT(case\_id) AS closed\_cases

FROM branch\_cases WHERE status = 'Closed' AND case\_id LIKE '%2024%'

GROUP BY branch\_id) AS r1

NATURAL JOIN

(SELECT branch\_id, COUNT(case\_id) AS total\_cases

FROM branch\_cases WHERE case\_id LIKE '%2024%'

GROUP BY branch\_id) AS r2;

1. Give efficiency of personnel till year 2023. We define efficiency as total number of solved cases by personnel divided by the total number of cases.

WITH

closed\_cases AS (

SELECT ac.personnel\_id, COUNT(c.case\_id) AS number\_of\_solved\_cases

FROM assigned\_case AS ac

NATURAL JOIN cases AS c

WHERE c.status = 'Closed' AND

EXTRACT(YEAR FROM c.reporting\_time) <> 2024

GROUP BY ac.personnel\_id),

open\_cases AS (

SELECT ac.personnel\_id, COUNT(c.case\_id) AS total\_number\_of\_cases

FROM assigned\_case AS ac

NATURAL JOIN cases AS c

WHERE EXTRACT(YEAR FROM c.reporting\_time) <> 2024

GROUP BY ac.personnel\_id)

SELECT personnel\_id, 100 \* (closed\_cases.number\_of\_solved\_cases::FLOAT / open\_cases.total\_number\_of\_cases)::NUMERIC(5,2) AS personnel\_efficiency

FROM closed\_cases NATURAL JOIN open\_cases;

3) Calculate branch-wise excess budget.

WITH r1 AS (

SELECT inv.branch\_id, SUM(inv.stock \* it.Cost) AS Cost

FROM (inventory AS inv NATURAL JOIN items AS it)

GROUP BY inv.branch\_id

ORDER BY Cost DESC),

r2 AS

(SELECT branch\_id, SUM(salary) AS total\_salary

FROM personnel

GROUP BY branch\_id),

r3 AS

(SELECT branch\_id, SUM(r1.cost + r2.total\_salary) AS total\_cost

FROM r1 NATURAL JOIN r2

GROUP BY branch\_id)

SELECT branch\_id, (b.budget - r3.total\_cost) AS extra\_budget

FROM branch AS b NATURAL JOIN r3

ORDER BY extra\_budget DESC;

4) Calculate the average case resolution time for each branch.

SELECT r1.branch\_id,

AVG(AGE(r2.case\_end\_date, r2.reporting\_date)) AS avg\_case\_resolve\_time

FROM

(SELECT branch\_id, case\_id

FROM personnel

NATURAL JOIN assigned\_case) AS r1

JOIN

(SELECT case\_id, DATE(reporting\_time) AS reporting\_date, case\_end\_date

FROM cases

NATURAL JOIN verdict) AS r2

ON r1.case\_id = r2.case\_id

GROUP BY branch\_id;

5) Unit Coordinators with the number of cases they have solved during their years of service

SELECT r.personnel\_id, r.name, r.location, r.unit\_name, count(r.case\_id) as number\_of\_solved\_cases

FROM

(branch as b

NATURAL JOIN associated\_unit as au

NATURAL JOIN unit as u

JOIN personnel as p

ON au.coordinator\_id = p.personnel\_id

NATURAL JOIN assigned\_case as ac

NATURAL JOIN cases as c) AS r

WHERE r.status='Closed'

GROUP BY (r.personnel\_id, r.name, r.location, r.unit\_name)

ORDER BY number\_of\_solved\_cases DESC;

6) List all the cases' case\_id, case\_title, associated\_unit for the case that are transferred from KW70 to other branch.

SELECT DISTINCT c.case\_id, c.case\_title, p.branch\_id, p.unit\_id

FROM cases AS c

NATURAL JOIN assigned\_case AS ac

NATURAL JOIN personnel p

WHERE (crime\_location LIKE '%Kolkata%') AND branch\_id <> 'KW70';

7) Branch Heads with the number of cases they have solved during their years of service.

SELECT p.personnel\_id, p.name, b.location, COUNT(c.case\_id) AS number\_of\_cases

FROM

(branch AS b

JOIN Personnel AS p

ON b.head\_id = p.personnel\_id

NATURAL JOIN assigned\_case AS ac

NATURAL JOIN cases AS c)

WHERE c.status='Closed'

GROUP BY (p.personnel\_id, p.name, b.location)

ORDER BY number\_of\_cases DESC;

8) Check if any of the suspects of a given case is already present in the Criminal Record

SELECT r.case\_id, r.criminal\_id, r.name

FROM

(SELECT v.case\_id, cr.criminal\_id, cr.name

FROM criminal\_record AS cr

NATURAL JOIN verdict AS v) AS r

JOIN suspect AS s

ON r.case\_id = s.case\_id

WHERE r.name = s.name

9) List cases in which criminals are Non-Indians.

SELECT c.case\_id, c.case\_title

FROM

(SELECT v.case\_id

FROM verdict AS v

NATURAL JOIN criminal\_record AS cr

WHERE cr.nationality <> 'Indian') AS r

JOIN cases AS C ON r.case\_id = c.case\_id;

10) Criminal involved in most cases

SELECT cr.criminal\_id, cr.name, COUNT(v.case\_id) AS number\_of\_cases

FROM criminal\_record as cr

NATURAL JOIN verdict as v

GROUP BY cr.criminal\_id, cr.name

ORDER BY number\_of\_cases DESC LIMIT 1;

11) Check if any of the witness of a given case is already present in the Criminal Record

SELECT r.case\_id, r.criminal\_id, r.name

FROM

(SELECT v.case\_id, cr.criminal\_id, cr.name

FROM criminal\_record AS cr

NATURAL JOIN verdict AS v) AS r

JOIN witness AS w

ON r.case\_id = w.case\_id

WHERE r.name = w.name

12) List the criminal(s), victim(s), suspect(s) and witnesses for a given case

SELECT \* FROM criminal\_record NATURAL JOIN verdict AS v WHERE v.case\_id = 'RC-02/2021/ACE/HYD'

SELECT \* FROM Victim WHERE Case\_ID = 'RC-02/2021/ACE/HYD';

SELECT \* FROM Suspect WHERE Case\_ID = 'RC-02/2021/ACE/HYD';

SELECT \* FROM Witness WHERE Case\_ID = 'RC-02/2021/ACE/HYD';

13) Branch that has solved maximum number of cases

SELECT b.location FROM

(SELECT branch\_id, count(case\_id) as number\_of\_solved

FROM assigned\_case as ac

NATURAL JOIN personnel

GROUP BY (branch\_id)

ORDER BY number\_of\_solved DESC LIMIT 1) as r

NATURAL JOIN branch AS b;

14) List Criminals who are convicted under IPC 120B for Bribery

SELECT cr.criminal\_id, cr.name

FROM criminal\_record AS cr

NATURAL JOIN verdict AS v

NATURAL JOIN cases AS c

WHERE v.court\_verdict LIKE '%IPC 120B%'

15) List all the cases that are unsolved from 2 years.

SELECT \*

FROM Cases

WHERE Status = 'Open' AND

CURRENT\_DATE - Reporting\_Time > INTERVAL '2 years';

16) Branch using highest amount of money in inventory

SELECT inv.branch\_id, SUM(inv.stock \* it.cost) AS Cost

FROM inventory AS inv

NATURAL JOIN items AS it

GROUP BY inv.branch\_id

ORDER BY Cost DESC LIMIT 1;

17) List all cases along with the court verdict details for branch 'HT50'

SELECT c.case\_id, c.case\_title, v.court\_verdict

FROM verdict AS v

NATURAL JOIN cases AS c

NATURAL JOIN assigned\_case

NATURAL JOIN personnel

WHERE branch\_id = 'HT50';

18) Find branch that has maximum number of Units

SELECT b.location, COUNT(au.unit\_id) as number\_of\_units

FROM branch as b

NATURAL JOIN associated\_unit as au

GROUP BY b.location

ORDER BY number\_of\_units DESC LIMIT 1;

19) Determine the number of officers per branch.

SELECT branch\_id, unit\_id, COUNT(personnel\_id)

FROM personnel

GROUP BY branch\_id, unit\_id;

20) Branches where the stock of inventory items is less than equal to 10 units

SELECT b.location, it.item\_name

FROM

(branch as b

NATURAL JOIN inventory as inv

NATURAL JOIN items as it)

WHERE inv.stock <= 10;

21) List the evidences so far found for a given case

SELECT \* FROM evidence WHERE case\_id = 'RC-06/2024/ACE/CHE';

22) Give personnel id of personnel with highest salary

SELECT personnel\_id, name, salary from personnel ORDER BY salary DESC LIMIT 1;

23) List unit-wise average salary of personnels

SELECT p.unit\_id, u.unit\_name, AVG(salary)::NUMERIC(8,2) AS average\_salary

FROM personnel AS p

NATURAL JOIN unit AS u

GROUP BY p.unit\_id, u.unit\_name

ORDER BY average\_salary DESC;

24) Average no. of cases per year for every branch (Consider the data of past 4 years).

SELECT p.branch\_id, COUNT(c.case\_id)/4 :: FLOAT AS average\_number\_of\_cases

FROM cases AS c

NATURAL JOIN assigned\_case AS ac

NATURAL JOIN personnel AS p

WHERE EXTRACT(YEAR FROM reporting\_time) <> 2024

GROUP BY p.branch\_id;

25) Give the list of retired or past personnels.

SELECT personnel\_id, name FROM personnel WHERE service\_status = ‘Inactive’;