

Stored procedure code:

```
1:
DELIMITER $$

CREATE PROCEDURE Derailment_cause_rate_recent_ten_years()
BEGIN
    WITH accident AS (
        SELECT ACCIDENTS_CAUSE.CAUSE_GROUP AS Cause_Code,
            ACCIDENTS_CAUSE.ADL_CAUSE_SUBGROUP AS Cause_Name,
            COUNT(*) AS derailment
        FROM RR_ACCIDENTS
        JOIN RR_CLASS ON RR_ACCIDENTS.RAILROAD = RR_CLASS.RAILROAD
        JOIN ACCIDENTS_CAUSE ON RR_ACCIDENTS.PRICAUSE =
ACCIDENTS_CAUSE.FRA_CAUSE_CODE
        WHERE RR_CLASS.RRCLASSIFICATION = 1
            AND RR_ACCIDENTS.ACC_TYPE = 1
            AND RR_ACCIDENTS.TRAIN_TYPE = 'F'
            AND RR_ACCIDENTS.TRACK_TYPE = 'MS'
            AND YEAR(RR_ACCIDENTS.`DATE`) BETWEEN 2013 AND 2022
        GROUP BY ACCIDENTS_CAUSE.CAUSE_GROUP,
ACCIDENTS_CAUSE.ADL_CAUSE_SUBGROUP
    ),
    traffic AS (
        SELECT SUM(RR_TRAFFIC.FRTRNMI + RR_TRAFFIC.OTHERMI) AS mile
        FROM RR_TRAFFIC
        JOIN RR_CLASS ON RR_TRAFFIC.RAILROAD = RR_CLASS.RAILROAD
        WHERE RR_CLASS.RRCLASSIFICATION = 1
            AND CONCAT('20', RR_TRAFFIC.IYR) >= '2013'
            AND CONCAT('20', RR_TRAFFIC.IYR) <= '2022'
    )
    SELECT accident.Cause_Code AS Cause_Code,
        accident.Cause_Name AS Cause_Name,
        (accident.derailment / (SELECT mile FROM traffic)) * 1000000 AS derailment_rate
    FROM accident
    ORDER BY derailment_rate DESC
    LIMIT 15;
END$$

DELIMITER ;
```

2:

DELIMITER \$\$

```
CREATE PROCEDURE Top_15_Railroad_traffic_ten_years()
BEGIN
  SELECT
    RR_CLASS.RAILROAD_SUCESSOR,
    RR_CLASS.RRCLASSIFICATION,
    SUM(RR_TRAFFIC.FRTRNMI) AS train_mile
  FROM
    RR_TRAFFIC
  JOIN
    RR_CLASS ON RR_TRAFFIC.RAILROAD = RR_CLASS.RAILROAD
  WHERE
    CONCAT('20', RR_TRAFFIC.IYR) >= '2013'
    AND CONCAT('20', RR_TRAFFIC.IYR) <= '2022'
  GROUP BY
    RR_CLASS.RAILROAD_SUCESSOR, RR_CLASS.RRCLASSIFICATION
  HAVING
    RR_CLASS.RAILROAD_SUCESSOR != ''
  ORDER BY
    train_mile DESC
  LIMIT 15;
END$$
```

DELIMITER ;

Transaction code:

DELIMITER \$\$

```
CREATE PROCEDURE UpdateTrafficData(
  IN p_TRAFFIC_CODE VARCHAR(255),
  IN p_NEW_RAILROAD VARCHAR(255),
  IN p_NEW_IYR INT,
  IN p_NEW_IMO INT
)
BEGIN
  DECLARE v_exists INT;
  DECLARE v_total_mileage DECIMAL(10,2);

  SET SESSION TRANSACTION ISOLATION LEVEL REPEATABLE READ;

  IF p_NEW_IYR < 1996 OR p_NEW_IYR > 2023 OR p_NEW_IMO < 1 OR p_NEW_IMO > 12
  THEN
```

```

        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid year or month';
    END IF;

    START TRANSACTION;

    SELECT COUNT(*) INTO v_exists FROM RR_TRAFFIC WHERE TRAFFIC_CODE =
p_TRAFFIC_CODE;
    IF v_exists = 0 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Traffic code not found';
    END IF;

    UPDATE RR_TRAFFIC
    SET RAILROAD = p_NEW_RAILROAD, IYR = p_NEW_IYR, IMO = p_NEW_IMO
    WHERE TRAFFIC_CODE = p_TRAFFIC_CODE;

    SELECT SUM(FRTRNMI + PASTRNMI + OTHERMI) INTO v_total_mileage
    FROM RR_TRAFFIC
    WHERE TRAFFIC_CODE = p_TRAFFIC_CODE;

    UPDATE RR_TRAFFIC
    SET YSMI = v_total_mileage
    WHERE TRAFFIC_CODE = p_TRAFFIC_CODE;

    COMMIT;
END$$

DELIMITER ;

```

Trigger Code

```

CREATE TRIGGER update_railroad
    BEFORE INSERT ON RR_TRAFFIC
    FOR EACH ROW
        BEGIN -- Check if the inserted RAILROAD is 'CN' and change it to 'CNGT'
            IF NEW.RAILROAD = 'CN' THEN SET NEW.RAILROAD = 'CNGT'
            ELSEIF NEW.RAILROAD = 'CP' THEN SET NEW.RAILROAD = 'CP(US)';
            END IF;
        END;

```

Example update, insert, and delete code:

Insert:

```

INSERT INTO RR_TRAFFIC (

```

```

TRAFFIC_CODE,
RAILROAD,
IYR,
IMO,
STATE,
COUNTY,
YSMI,
FRTRNMI,
PASTRNMI,
OTHERMI
) VALUES (
'TC001',    -- Traffic Code
'CN',       -- Railroad, to be changed to 'CNGT' by the trigger
21,         -- Year
12,         -- Month
48,         -- State code
'Harris',   -- County
100,        -- Yearly Sum Miles (example)
100000000,  -- Freight Train Miles (example)
30,         -- Passenger Train Miles (example)
20         -- Other Miles (example)
);

```

Update:

```

UPDATE RR_TRAFFIC
SET
    FRTRNMI = 75, -- Updated value for Freight Train Miles
    PASTRNMI = 40, -- Updated value for Passenger Train Miles
    OTHERMI = 25  -- Updated value for Other Miles
WHERE
    TRAFFIC_CODE = 'TC001'; -- Identifying the specific record by Traffic Code

```

Delete:

```

DELETE FROM RR_TRAFFIC
WHERE TRAFFIC_CODE = 'TC002'; -- Specify the Traffic Code of the record to delete

```

Other useful code:

```

CALL Top_15_Railroad_traffic_ten_years();
CALL Derailment_cause_rate_recent_ten_years();
SHOW CREATE PROCEDURE procedure_name;
SELECT * FROM INFORMATION_SCHEMA.TRIGGERS WHERE TRIGGER_SCHEMA =
'railroad_db';
CALL UpdateTrafficData();

```

```
SELECT * FROM RR_TRAFFIC WHERE TRAFFIC_CODE = 'TC001';
```