Transaction

Unset

```
ROLLBACK;

SET SESSION TRANSACTION ISOLATION LEVEL REPEATABLE READ;

USE dwellx;

START TRANSACTION;

SELECT
    AVG(school),
    AVG(safety_level),
    AVG(recreation_proximity),
    AVG(hospital_proximity)
INTO
    @avg_min_school_rating,
    @avg_max_crime_rating,
    @avg_max_recreation_distance_km,
    @avg_max_hospital_distance_km
FROM preferences;

WITH HousingSchoolStats AS (
    SELECT
        h.`index` AS housing_id,
        h.`Property Name` AS property_name,
        h.address,
        h.`Zip Code` AS zip_code,
        AVG(
          CASE s.culture_climate_rating
            WHEN 'WELL ORGANIZED' THEN 5
            WHEN 'ORGANIZED' THEN 4
            WHEN 'NOT YET ORGANIZED' THEN 2
            WHEN 'NOT ENOUGH DATA' THEN 1
            ELSE 0
          END
        ) AS avg_school_rating
    FROM housing h
    LEFT JOIN school_district sd ON h.`index` = sd.housing_id
    LEFT JOIN schools s ON sd.school_id = s.school_id
    GROUP BY h.`index`
),
CrimeStats AS (
    SELECT
```

```
        housing_id,
        COUNT(*) AS nearby_crime_count
    FROM crime_housing
    GROUP BY housing_id
),
HospitalStats AS (
    SELECT
        zip AS zip_code,
        COUNT(*) AS nearby_hospital_count
    FROM hospitals
    GROUP BY zip
)

SELECT
    hs.housing_id,
    hs.property_name,
    hs.address,
    hs.zip_code,
    hs.avg_school_rating,
    c.nearby_crime_count,
    h.nearby_hospital_count
FROM HousingSchoolStats hs
LEFT JOIN CrimeStats c ON hs.housing_id = c.housing_id
LEFT JOIN HospitalStats h ON hs.zip_code = h.zip_code
HAVING hs.avg_school_rating >= @avg_min_school_rating
   AND nearby_crime_count <= @avg_max_crime_rating
ORDER BY avg_school_rating DESC,
         nearby_crime_count ASC,
         nearby_hospital_count DESC
LIMIT 15;

COMMIT;
```

Trigger

Unset

```
USE dwellx;
```

```
DELIMITER //

CREATE TRIGGER create_default_preferences
AFTER INSERT ON users
FOR EACH ROW
BEGIN
    INSERT INTO preferences (user_id, zipcode, school, safety_level,
hospital_proximity, recreation_proximity)
    VALUES (NEW.user_id, 60606, 3, 3, 10, 10);
END//

DELIMITER ;
```

Stored Procedure

Unset

```
DELIMITER $$

CREATE PROCEDURE get_houses_by_preferences(
    IN recreation_distance_km FLOAT,
    IN hospital_distance_km FLOAT,
    IN desired_safety_level INT,
    IN pref_zipcode VARCHAR(10)
)
BEGIN
    -- Query 1: Hospital distance + Crime level
    SELECT
        h.*,
        MIN(hh.distance) AS min_hospital_distance,
        COUNT(c.index) AS crime_count,
        (
            CASE WHEN MIN(hh.distance) > hospital_distance_km THEN 1 ELSE 0 END
+
            CASE WHEN COUNT(c.index) > desired_safety_level THEN 1 ELSE 0 END
        ) AS preference_miss_count_hospital_crime
    FROM housing h
    LEFT JOIN hospitals_houses hh ON h.housing_id = hh.housing_id
    LEFT JOIN crime_housing ch ON h.housing_id = ch.housing_id
    LEFT JOIN crimes c ON c.case_number = ch.case_number
    WHERE (pref_zipcode IS NULL OR h.zip_code = pref_zipcode)
    GROUP BY h.housing_id
    ORDER BY preference_miss_count_hospital_crime ASC;

    -- Query 2: Park distance + School distance
    SELECT
        h.housing_id,
        h.community_area_name,
        MIN(ph.distance) AS min_park_distance,
        MIN(sd.distance) AS min_school_distance,
        (
            CASE WHEN MIN(ph.distance) > recreation_distance_km THEN 1 ELSE 0
END +
            CASE WHEN MIN(sd.distance) > recreation_distance_km THEN 1 ELSE 0
END
        ) AS preference_miss_count_park_school
    FROM housing h
    LEFT JOIN parks_houses ph ON h.housing_id = ph.housing_id
    LEFT JOIN school_district sd ON h.housing_id = sd.housing_id
    WHERE (pref_zipcode IS NULL OR h.zip_code = pref_zipcode)
    GROUP BY h.housing_id
    ORDER BY preference_miss_count_park_school ASC;
END$$

DELIMITER ;



// example: CALL get_houses_by_preferences(5,5,2,'60647');
```