```
DELIMITER $$

CREATE TRIGGER BeforePropertiesUpdate
AFTER UPDATE ON Properties
FOR EACH ROW
BEGIN
   IF OLD.price <> NEW.price THEN
      INSERT INTO audit_log (tableName, fieldName, oldValue, newValue, timestamp)
      VALUES ('Properties', 'price', OLD.price, NEW.price, NOW());
   END IF;
END $$
DELIMITER;
```

```
DELIMITER //
CREATE TRIGGER after_property_delete
AFTER DELETE ON UserListings
FOR EACH ROW
BEGIN
        DELETE FROM Address WHERE propertyId = OLD.propertyId;
        DELETE FROM UserFavorites WHERE propertyId = OLD.propertyId;
        DELETE FROM Properties WHERE propertyId = OLD.propertyId;

END;
//
DELIMITER ;
```

```
DELIMITER $$

CREATE PROCEDURE GetPropertiesWithinDistance(
   IN targetLat DOUBLE(15, 7),
   IN targetLon DOUBLE(15, 7),
   IN minDistance DECIMAL(10, 2),
   IN minPrice DECIMAL(10, 2),
   IN maxPrice DECIMAL(10, 2),
   IN numBathrooms INT,
   IN numBedrooms INT,
   IN propType VARCHAR(255),
   IN yearBuilt INT
)
BEGIN
```

```
    SELECT p.*, a.latitude, a.longitude
    FROM Properties p
    JOIN Address a ON p.propertyId = a.propertyId
    WHERE EXISTS (
        SELECT 1
        FROM (
            SELECT a.propertyId AS id,
                (
                  6371 * acos(
                    cos(radians(a.latitude)) * cos(radians(targetLat)) *
                    cos(radians(targetLon) - radians(a.longitude)) +
                    sin(radians(a.latitude)) * sin(radians(targetLat))
                  )
                ) AS distance
            FROM Address a
        ) AS distances
        WHERE distances.id = p.propertyId
          AND distances.distance <= minDistance
    )
    AND (minPrice IS NULL OR p.price >= minPrice)
    AND (maxPrice IS NULL OR p.price <= maxPrice)
    AND (numBathrooms IS NULL OR p.numBathrooms = numBathrooms)
    AND (numBedrooms IS NULL OR p.numBedrooms = numBedrooms)
    AND (propType IS NULL OR p.propertyType = propType)
    AND (yearBuilt IS NULL OR p.yearBuilt >= yearBuilt);
END$$

DELIMITER ;
```

```
DELIMITER //

CREATE PROCEDURE ComparePropertyPriceToAverage(
    IN centerLat DECIMAL(10, 8),
    IN centerLng DECIMAL(11, 8)
)
BEGIN
    DECLARE radius DECIMAL(10, 2);
    DECLARE average DECIMAL(10, 2);

    SET radius = 5; -- Radius in miles

    -- Calculate the average price within the radius
```

```
    SELECT AVG(price) INTO average
    FROM Properties
    WHERE ST_Distance_Sphere(
        point(centerLat, centerLng),
        point(Properties.latitude, Properties.longitude)
    ) <= (radius * 1609.34);

    -- Select all properties and compare their price to the average
    SELECT Properties.propertyId, Properties.price,
           CASE
               WHEN Properties.price > (average * 1.1) THEN 'Bad Value'
               WHEN Properties.price < (average * 1.1) THEN 'Good Value'
               ELSE 'Fair Value'
           END AS priceComparison
    FROM Properties;
END //

DELIMITER ;
```