

**Procedure:**

```
DROP PROCEDURE IF EXISTS UpdateChannelRatingsAdvanced2;
```

```
DELIMITER //
```

```
CREATE PROCEDURE UpdateChannelRatingsAdvanced2()
```

```
BEGIN
```

```
    DECLARE done INT DEFAULT FALSE;
```

```
    DECLARE channel_id_something VARCHAR(255);
```

```
    DECLARE avg_likes FLOAT;
```

```
    DECLARE cat_count INT;
```

```
    DECLARE union_likes INT;
```

```
    DECLARE rating_score INT;
```

```
        DECLARE letterGrade VARCHAR(255);
```

```
    DECLARE cur1 CURSOR FOR SELECT channelId FROM Channels;
```

```
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
```

```
    -- Temporary table for average likes
```

```
    DROP TABLE IF EXISTS TempAvgLikes;
```

```
    CREATE TABLE TempAvgLikes (
```

```
        channel_id VARCHAR(255) ,
```

```
        Average_Likes FLOAT,
```

```
        Category_Count INT
```

```
);
```

```
    -- Temporary table for likes from UNION query
```

```
    DROP TABLE IF EXISTS TempUnionLikes;
```

```
    CREATE TABLE TempUnionLikes (
```

```
        channel_id VARCHAR(255),
```

```
        likes INT
```

```
);
```

```
    -- Populate TempAvgLikes
```

```
    INSERT INTO TempAvgLikes (channel_id, Average_Likes, Category_Count)
```

```
    SELECT Videos.channel_id, AVG(likes) AS Average_Likes, COUNT(Categories.categoryId)
```

```
    AS Category_Count
```

```
    FROM Videos
```

```
    JOIN Categories ON Videos.category_id = Categories.categoryId
```

```
    GROUP BY Videos.channel_id;
```

```
    -- Populate TempUnionLikes
```

```
    INSERT INTO TempUnionLikes (channel_id, likes)
```

```
    (
```

```

SELECT Channels.channelId, Videos.likes
FROM Videos
JOIN Channels ON Videos.channel_id = Channels.channelId
WHERE Videos.view_count < 1000
UNION
SELECT Channels.channelId, Videos.likes
FROM Videos
JOIN Channels ON Videos.channel_id = Channels.channelId
WHERE Videos.view_count > 10000
);

OPEN cur1;

read_loop: LOOP
    FETCH cur1 INTO channel_id_something;
    IF done THEN
        LEAVE read_loop;
    END IF;

    -- Retrieve average likes
    SELECT Average_Likes INTO avg_likes
    FROM TempAvgLikes
    WHERE channel_id = channel_id_something;

    SELECT SUM(Category_Count) INTO cat_count
    FROM TempAvgLikes
    WHERE channel_id = channel_id_something;

    -- Retrieve likes from UNION query
    SELECT MAX(likes) INTO union_likes
    FROM TempUnionLikes
    WHERE channel_id = channel_id_something;

    -- Calculate rating score (placeholder logic)
    SET rating_score = ((avg_likes + union_likes)*cat_count) / 2;

    -- Determine letterGrade based on rating_score
    IF rating_score > 100000 THEN
        SET letterGrade = 'A';
    ELSEIF rating_score > 80000 AND rating_score <= 100000 THEN
        SET letterGrade = 'B';
    ELSEIF rating_score IS NULL THEN
        SET letterGrade = 'D';
    ELSE

```

```

        SET letterGrade = 'C';
    END IF;

    -- Update Channels table
    UPDATE Channels
    SET rating_score = rating_score, letterGrade = letterGrade
    WHERE channelId = channel_id_something;
END LOOP;

-- Close cursor
CLOSE cur1;

-- DROP TEMPORARY TABLE IF EXISTS TempAvgLikes;
-- DROP TEMPORARY TABLE IF EXISTS TempUnionLikes;
END //

DELIMITER ;

CALL UpdateChannelRatingsAdvanced2();

```

### Trigger:

```

DROP TRIGGER AssignRating;

DELIMITER $$
CREATE TRIGGER AssignRating
AFTER INSERT ON Login
FOR EACH ROW
BEGIN
    DECLARE numVideos INT;
    DECLARE newscore FLOAT;
    DECLARE newgrade VARCHAR(255);
    DECLARE user_ct VARCHAR(255);
    DECLARE cid VARCHAR(255);
    SET user_ct = NEW.user;

    SELECT channelId into cid FROM Channels where channelTitle LIKE
CONCAT('%', NEW.user, '%');

    SELECT COUNT(*) into numVideos from Videos WHERE channel_id LIKE
CONCAT('%', cid, '%');

```

```
-- -- Count the number of videos for the inserted user's channel
-- SELECT COUNT(video id) INTO numVideos
-- FROM Videos
-- WHERE v.title KE CONCAT('%', 'Polo G', '%');

-- Check if the channel has at least 500 videos
IF numVideos > 20 THEN
    UPDATE Channels c

    SET c.rating_score = 100000, c.letterGrade = "A+"
    WHERE c.channelTitle LIKE CONCAT('%', NEW.user, '%');
END IF;
END$$
```