

```

CREATE VIEW publication_keyword_detail_view AS
SELECT
    p.id AS publication_id,
    p.title,
    p.venue,
    p.year,
    p.num_citations,
    pk.keyword_id,
    k.name AS keyword,
    pk.score,
    f.id AS faculty_id,
    f.name AS faculty_name,
    k_all.name AS all_keyword
FROM publication p
JOIN Publication_Keyword pk ON p.id = pk.publication_id
JOIN keyword k ON pk.keyword_id = k.id
LEFT JOIN Publication_Keyword pk_all ON p.id = pk_all.publication_id
LEFT JOIN keyword k_all ON pk_all.keyword_id = k_all.id
LEFT JOIN faculty_publication fp ON p.id = fp.publication_id
LEFT JOIN faculty f ON fp.faculty_id = f.id;

```

```

CREATE VIEW faculty_keyword_score_view AS
SELECT
    f.id AS faculty_id,
    f.name AS faculty_name,
    f.position,
    u.name AS university_name,
    k.name AS keyword,
    fk.score
FROM faculty f
JOIN faculty_keyword fk ON f.id = fk.faculty_id
JOIN keyword k ON fk.keyword_id = k.id
JOIN university u ON f.university_id = u.id;

```

```

CREATE VIEW university_keyword_summary_view AS
SELECT
    f.id AS faculty_id,
    f.name AS faculty_name,
    k.name AS keyword,
    fk.score,
    u.name AS university_name,
    u.photo_url
FROM faculty f
JOIN faculty_keyword fk ON f.id = fk.faculty_id
JOIN keyword k ON fk.keyword_id = k.id
JOIN university u ON u.id = f.university_id;

```

```
CREATE TABLE keyword_score_log (  
    log_id INT AUTO_INCREMENT PRIMARY KEY,  
    faculty_id INT,  
    keyword_id INT,  
    old_score FLOAT,  
    new_score FLOAT,  
    changed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
DELIMITER $$
```

```
CREATE TRIGGER log_faculty_keyword_score_update  
BEFORE UPDATE ON faculty_keyword  
FOR EACH ROW  
BEGIN  
    IF OLD.score != NEW.score THEN  
        INSERT INTO keyword_score_log (faculty_id, keyword_id, old_score,  
new_score)  
        VALUES (OLD.faculty_id, OLD.keyword_id, OLD.score, NEW.score);  
    END IF;  
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE TRIGGER log_publication_keyword_score_update  
BEFORE UPDATE ON Publication_Keyword  
FOR EACH ROW  
BEGIN  
    IF OLD.score != NEW.score THEN  
        INSERT INTO keyword_score_log (publication_id, keyword_id, old_score,  
new_score)  
        VALUES (OLD.publication_id, OLD.keyword_id, OLD.score, NEW.score);  
    END IF;  
END$$
```

```
DELIMITER ;
```