S08-P03 Practice

Assignment

Create a new procedure `INS_USER_STAT` to aggregate stats about user logins. The procedure should have 1 input parameter: `user_id`.

Save the following data to `sample_staff`.`user_stat`:

- `user_id`
- `login_date`
- `login_hour`
- `login_count`

...

...

...

SPOILER: Below are the results

...

...

Result

```
DROP PROCEDURE IF EXISTS `INS_USER_STAT`;

DELIMITER //

CREATE PROCEDURE `INS_USER_STAT`(
  in_user_id INT
)
BEGIN

  INSERT INTO `sample_staff`.`user_stat`
    (`user_id`, `date`, `hour`, `login_count`, `insert_dt`,
`insert_process_code`)
```

```
    SELECT
      xTMP.user_id,
      login_date,
      login_hour,
      login_count,
      insert_dt,
      insert_process_code
    FROM (
      SELECT
        `user_login`.`user_id` AS user_id,
        DATE(user_login.login_dt) AS login_date,
        HOUR(user_login.login_dt) AS login_hour,
        COUNT(*) AS login_count,
        NOW() AS insert_dt,
        'INS_USER_STAT' AS insert_process_code
      FROM `sample_staff`.`user_login`
      WHERE 1=1
        AND `user_login`.`user_id` = in_user_id
      GROUP BY
        `user_login`.`user_id`,
        DATE(user_login.login_dt),
        HOUR(user_login.login_dt)
    ) xTMP
  ON DUPLICATE KEY UPDATE
    user_stat.login_count = xTMP.login_count,
    user_stat.update_process_code = CASE
      WHEN user_stat.login_count != xTMP.login_count THEN 'INS_USER_STAT'
      ELSE user_stat.update_process_code
    END,
    user_stat.update_dt = CASE
      WHEN user_stat.login_count != xTMP.login_count THEN NOW()
      ELSE user_stat.update_dt
    END
  ;

END;
//

DELIMITER ;
```

Call the function to try it out:

```
CALL `INS_USER_STAT`(10001);
```