

<PostgreSQL - cs421> Script-7

```
/*
 * A stored procedure that ensures that no two employees have the same
 company email.
 *
 * It does this by defining a cursor to the emails of the employees, and
 iterating
 * through them. If two company emails are identical, the second duplicate
 is changed by
 * appending the digit <1> to the end of the email address.
 *
 * For example, if two emails <jeff1@amazon.com> are in the database under
 two employees' emails,
 * then the second will be changed to <jeff11@amazon.com.>
 */
DO $$DECLARE
-- Keep track of previous email you've seen
prev_email VARCHAR(100) := '';
-- A record to insert rows into
rec_emails RECORD;
-- Cursor to fetch emails, in order
cursor_emails CURSOR
FOR
    SELECT p.email
    FROM person p
    INNER JOIN employee e
        ON p.pid = e.pid
    ORDER BY 1;
-- A text variable to store the update statements
update_statement text := '';
BEGIN
    -- Open the cursor
    OPEN cursor_emails;

    LOOP
        -- fetch row into the film
        FETCH cursor_emails INTO rec_emails;
        -- exit when no more row to fetch
        EXIT WHEN NOT FOUND;

        -- build the output
        IF rec_emails.email = prev_email THEN
```

<PostgreSQL - cs421> Script-7

```
-- Build the update statement
update_statement := 'update table person set email = ' ||
    split_part(rec_emails.email, '@', 1) || '1' || '@' ||
split_part(rec_emails.email, '@', 2) ||
    'where email = ' || prev_email;
-- Executer statement
EXECUTE update_statement;
END IF;
-- Update the previous email you've seen
prev_email = split_part(rec_emails.email, '@', 1) || '1' || '@' ||
split_part(rec_emails.email, '@', 2);
END LOOP;

-- Close the cursor
CLOSE cursor_emails;
END$$;
```