Đề bài

Thiết kế database dùng để quản lý công việc các phòng ban:

- Mỗi phòng ban có 1 trưởng phòng và n nhân viên.
- Mỗi nhân viên có thể có một người hướng dẫn.
- Các nhân viên luôn được giao công việc.
- Các trưởng phòng là người lên các kế hoạch công việc sắp tới.

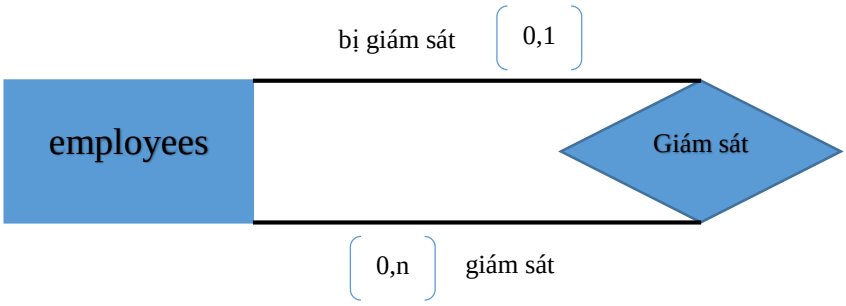
Danh sách kiểu thực thể.

1. **employees**: Nhân viên: lưu các thông tin về nhân viên và cả thông tin đăng nhập.
2. **offices**: Phòng ban.
3. **works**: Công việc cần làm.

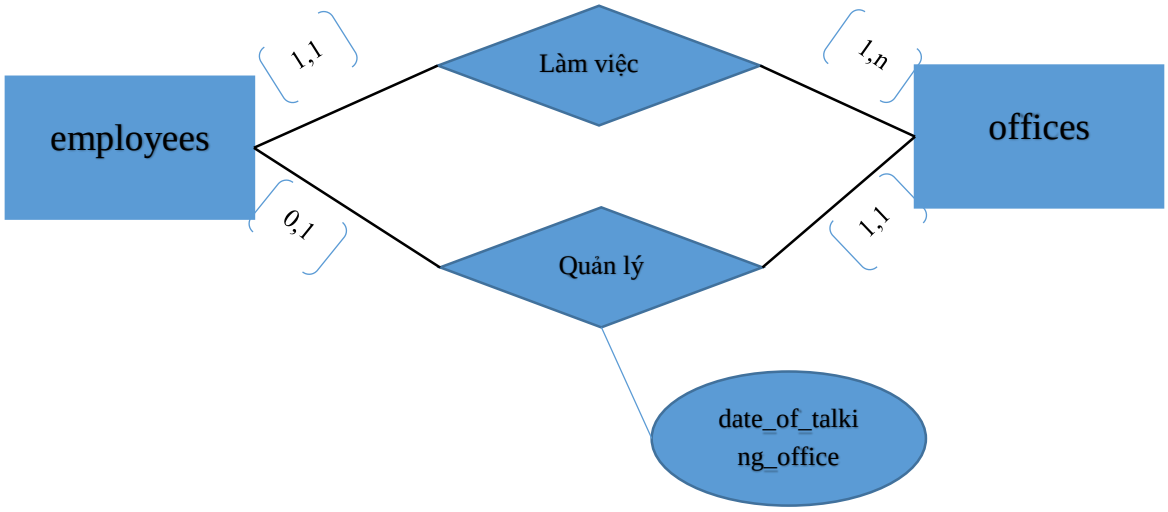
Và 1 kiểu thực thể tuy không có thật sự có mốt quan hệ với những kiểu thực thể khác nhưng đảm nhận vai trò quan sát sự thay đổi đổi trên các bảng khác. Đó là kiểu thực thể **Logs,** *đóng vai trò ghi lại các hành động thêm, xóa, sửa ở các bảng khác.*

Mối quan hệ.

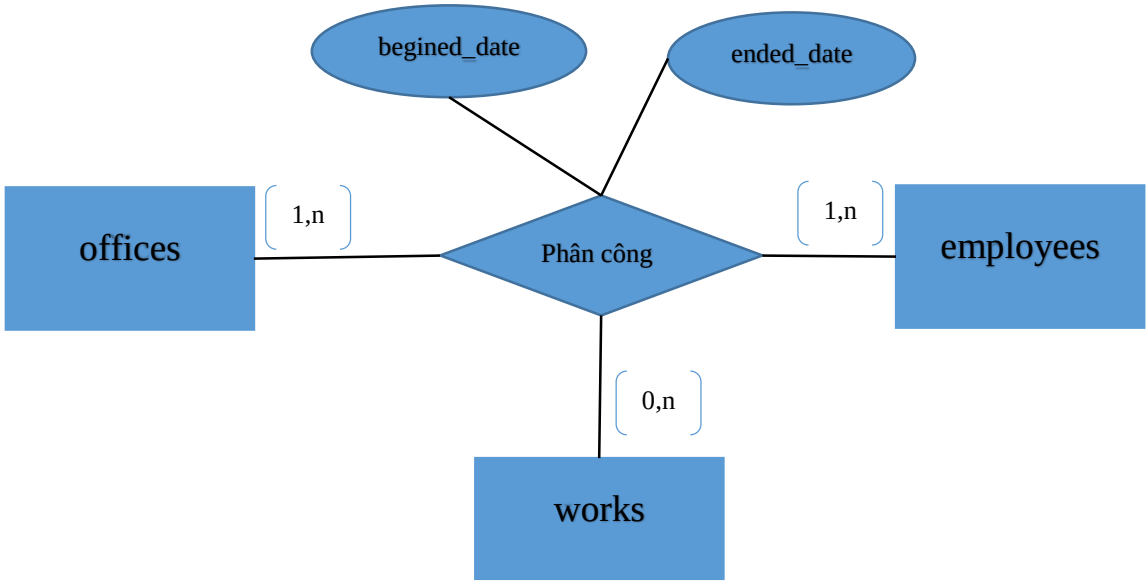
1. employees với employees.



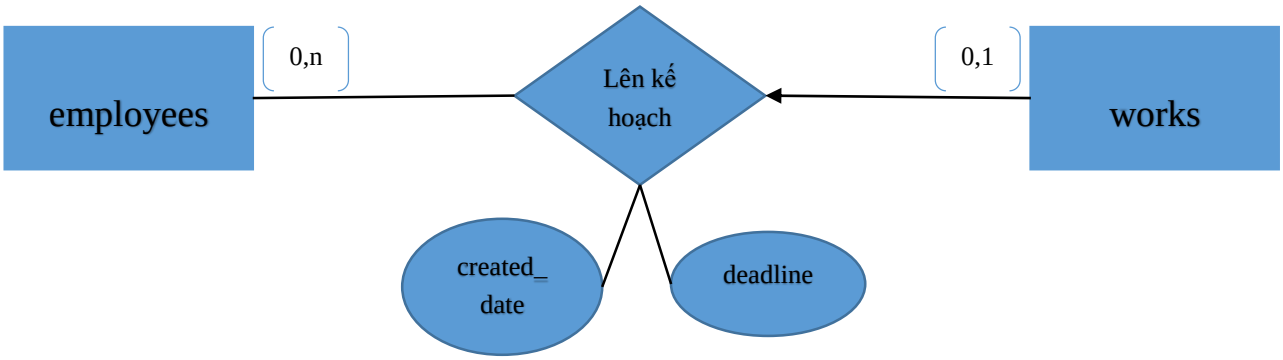
2. employees với offices.



3. Liên kết đa ngôi: employees, offices và works.



4. works với employees.



Danh sách thuộc tính.

1. employees.

- stt
- id **(khóa chính)**
- first_name
- last_name
- gioi_tinh
- country
- city
- district
- villag
- address_extra
- phone
- email
- jobTitle
- reprot_to
- salary
- username
- password
- updated_date.

Lưu ý: id = "FTECH" + %04d(**stt**).

2. offices.

- stt
- id (**Khóa chính**)
- office_name
- description
- updated_date.

Lưu ý: id = "OFTECH" + %04d (**stt**)

3. works.

- stt
- content
- id (**Khóa chính**)
- created_date
- heading
- ended_date

- status
- updated_date.

Lưu ý: id = “WFTECH” + %08d.

4. logs.

- stt **(Khóa chính)**
- table_name
- entity_id
- type
- content.
- edited_employees
- updated_date.

Lý do: Tại sao không chia làm 2 trường: content_old (lưu những trường:giá trị trước khi bị sử đổi) và content_new(lưu những trường:giá trị mới)?

- ⇨ Nếu vậy: khi thêm mới giá trị: content_old = null, content_new = tên trường 1: giá trị trường 1, tên trường 2: giá trị trường 2, …
- ⇨ Khi sửa giá trị: content_old = tên trường 1: giá trị cũ, tên trường 2: giá trị cũ,… và content_new = tên trường 1: giá trị mới, tên trường 2: giá trị mới: …
- ⇨ Khi xóa giá trị: contetn_old: tên trường 1: giá trị, tên trường 2: giá trị,… và content_new = null.
  Như vậy, xảy ra trường hợp 1 dữ liệu được lưu ở nhiều nơi => dư thừa dữ liệu.

Vậy nên, thay vì chia ra làm 2 trường content_old và content_new, ta sẽ chỉ cần 1 trường là content là đủ để so sánh các giá trị thay đổi rồi.

Lưu ý: trường **content** sẽ có các giá trị phụ thuộc vào giá trị trường **type**, cụ thể như sau:

- ➢ INSERT: content = **tên trường 1: giá trị tên trường 1, tên trường 2: giá trị tên trường 2, …**
  Lý do: Tại sao khi thêm mới giá trị vào các bảng, ta ngoài lưu giá trị đó ở các bảng đích, lại còn phải lưu thêm giá trị đó ở bảng Logs?
  - ⇨ Đúng là: việc lưu dữ liệu kiểu này sẽ dẫn đến hiện tượng trùng lặp dữ liệu khi tồn tại đến 2 bản ghi giống nhau ở 2 bảng. Nhưng ta cần phải biết: trước khi dữ liệu bị sửa đổi hay xóa, thì dữ liệu đó có giá trị như thể nào, mà điều này lại không thể lưu ở các bảng đích được vì sau các hành động xóa hoặc sửa, dữ liệu ở bảng đích đã bị thay đổi rồi.

- UPDATE: content = **tên trường 1 chỉnh sửa: giá trị tên trường 1, tên trường 2 chỉnh sửa: giá trị tên trường 2,….**
- DELETE: content = null.

Mô hình dữ liệu.