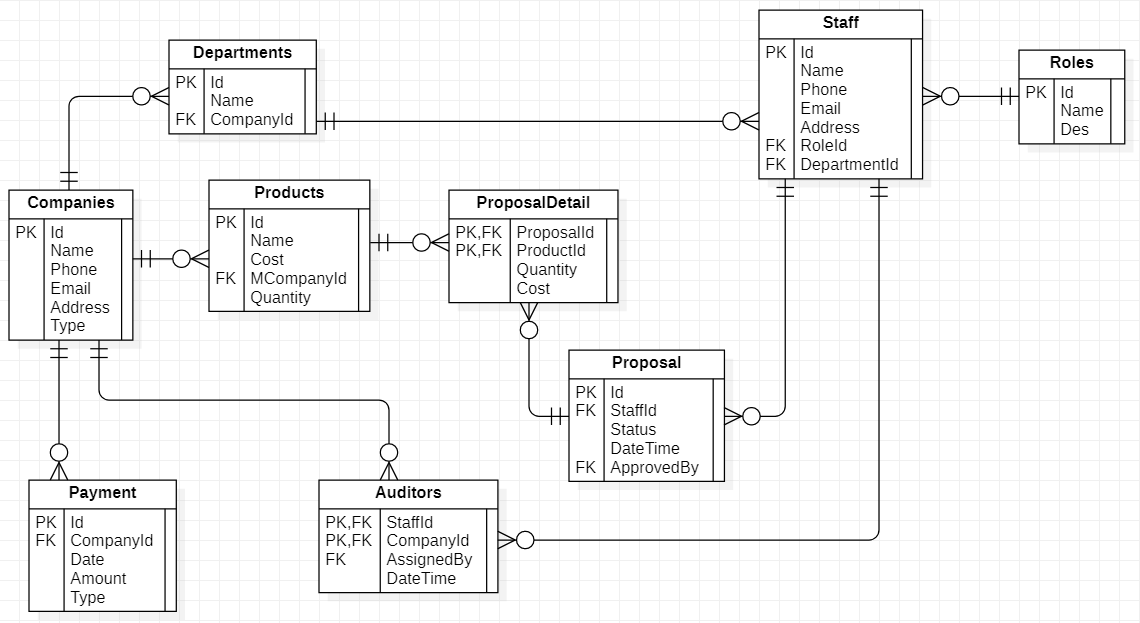
Nguyễn Thị Phi Thương

NodeJS Intern

1. Database schema design



Explanation:

1. Companies

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích | Ghi chú |
| Id | String | Id của công ty | Khoá chính |
| Name | String | Tên công ty |  |
| Phone | String | Số điện thoại công ty |  |
| Email | String | Email công ty |  |
| Address | String | Địa chỉ công ty |  |
| Type | String | Kiểu công ty  Type = “merchant” OR Type = “audit” |  |

1. Departments

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích | Ghi chú |
| Id | String | Id của phòng ban | Khoá chính |
| Name | String | Tên phòng ban |  |
| CompanyId | String | Id của công ty  Mỗi công ty có một hoặc nhiều phòng ban | Khoá phụ |

1. Roles

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích | Ghi chú |
| Id | String | Id chức vụ | Khoá chính |
| Name | String | Tên chức vụ (staff, company managers, department managers,…) |  |
| Des | String | Mô tả về chức vụ |  |

1. Staff

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích | Ghi chú |
| Id | String | Id của nhân viên | Khoá chính |
| Name | String | Tên nhân viên |  |
| Phone | String | Số điện thoại nhân viên |  |
| Email | String | Email nhân viên |  |
| Address | String | Địa chỉ nhân viên |  |
| RoleId | String | Chức vụ của nhân viên | Khoá phụ |
| DepartmentId | String | Nhân viên thuộc phòng ban nào | Khoá phụ |

1. Products

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích | Ghi chú |
| Id | String | Id của sản phẩm | Khoá chính |
| Name | String | Tên sản phẩm |  |
| Cost | Long | Giá tiền mỗi sản phẩm  Cost > 0 |  |
| MCompanyId | String | Id công ty sở hữu sản phẩm (Type = “merchant”)  Mỗi công ty có thể không có hoặc có nhiều sản phẩm | Khoá phụ |
| Quantity | Long | Số lượng sản phẩm  Quantity >= 0 |  |

1. Proposal

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích | Ghi chú |
| Id | String | Id yêu cầu | Khoá chính |
| StaffId | String | Nhân viên yêu cầu (với role là staff và company type = “merchant”)  Mỗi nhân viên có thể không có hoặc có nhiều yêu cầu | Khoá phụ |
| Status | Long | Trạng thái của yêu cầu (“approved”, “not approved” or “waiting”. “waiting” là trạng thái mặc định khi yêu cầu được tạo ra) |  |
| DateTime | DateTime | Thời gian cập nhật trạng thái |  |
| ApprovedBy | String | Id người duyệt yêu cầu  Yêu cầu được duyệt bởi company manager của công ty hoặc trưởng phòng của nhân viên tạo yêu cầu | Khoá phụ |

1. ProposalDetail

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích | Ghi chú |
| ProposalId | String | Id yêu cầu | Khoá chính, khoá phụ |
| ProductId | String | Id sản phẩm được yêu cầu | Khoá chính, khoá phụ |
| Quantity | Long | Số lượng yêu cầu  Quantity > 0 và Quantity < Products.Quantity |  |
| Cost | Long | Giá tiền mỗi sản phẩm  Cost > 0 |  |

1. Payment

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích | Ghi chú |
| Id | String | Id thanh toán | Khoá chính |
| CompanyId | String | Id công ty thực hiện việc thanh toán (type = “merchant”)  Mỗi công ty có thể thực hiện nhiều giao dịch thanh toán hoặc không có giao dịch nào | Khoá phụ |
| Date | DateTime | Ngày thanh toán |  |
| Amout | Long | Tổng số tiền |  |
| Type | String | Loại giao dịch |  |

1. Auditors

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Giải thích | Ghi chú |
| StaffId | String | Id nhân viên kiểm toán  Một nhân viên có thể tham gia kiểm toán nhiều công ty Nhân viên phải thuộc công ty có type = “audit” | Khoá chính, khoá phụ |
| CompanyId | String | Id công ty được kiểm toán  Một công ty có thể được kiểm toán bởi nhiều nhân viên | Khoá chính, khoá phụ |
| AssignedBy | DateTime | Id người phê duyệt cho nhân viên kiểm toán (là company manager công ty của nhân viên) | Khoá phụ |
| DateTime | Long | Thời gian chỉ định |  |

1. Algorithm

Code:

const readline = require('readline');

const rl = readline.createInterface({

  input: process.stdin,

  output: process.stdout

});

const longestSubstringLength = (*str*) => {

    let longestSubstring = "";

    let curSub = "";

    let start = 0, end = 0;

    while (start < *str*.length && end < *str*.length) {

        while (start >= 0 && start < *str*.length && end < *str*.length) {

            if (curSub.indexOf(*str*.charAt(end)) == -1) {

                curSub = *str*.slice(start, end + 1);

                end++;

            } else {

                if (curSub.length > longestSubstring.length) {

                    longestSubstring = curSub;

                    curSub = "";

                }

                start = end;

                end = start + 1;

                break;

            }

        }

    }

    return longestSubstring.length;

}

rl.question('Type your string: ', function (*str*) {

    console.log(`The longest substring of ${*str*} is: ${longestSubstringLength(*str*)}`);

    rl.close();

});

rl.on('close', function () {

  console.log('\nProgram end.');

  process.exit(0);

});

Result:

