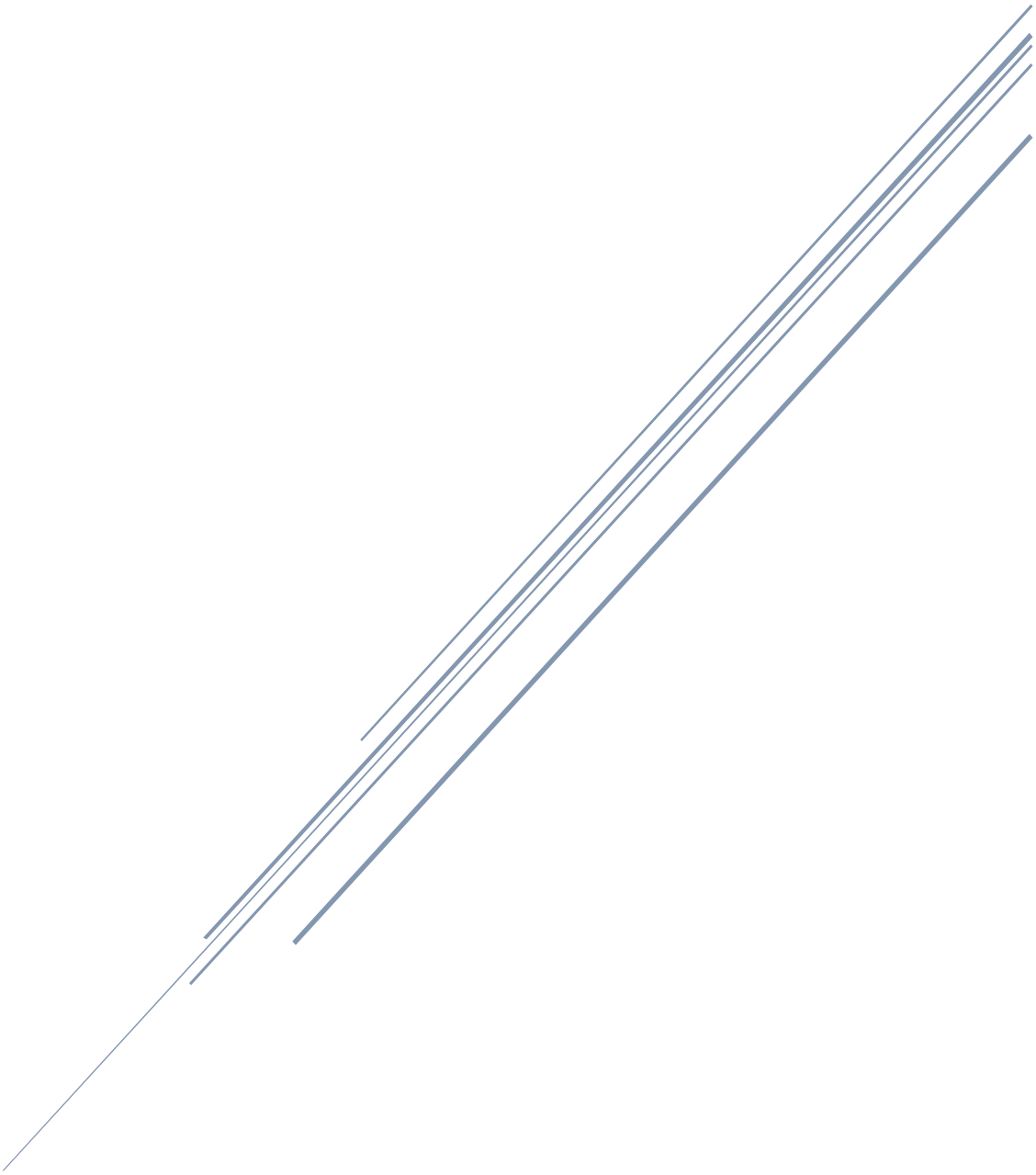
DATABASE PRINCIPLE PROJECT

Advertising Business Company Submission Director (part1)



TSUNG-TSE TU
S4780187

INDEX

Introduction

With the whole industry striving for customer satisfaction nowadays, advertising is considered a must-have strategy for product marketing undoubtedly. Not only can an advertising agency provide an outside point of view to the effort of selling the client's product or service, but also handle the overall marketing and branding strategies for clients. Studies show that pleasant collaborations between clients and advertising agencies often lead to success via a shared purpose. Without a proper approach for ad company to handle their clients' requests, it would be hard for them to offer what the best they got.

To make the process more accessible for clients to submit their demands and agencies to carry out the results the clients claimed for, we need a system to connect the two characters, where clients can create an account to submit their ideal project online presenting a quick insight for agency, and after receiving requests, they can further plan actions on what to do next, such as make appointments with client; If the project was already comprehensive enough then proceed to organize a blueprint and keep clients updated of the progress.

Requirements:

Programming language: Python and MySQL

Software: phpMyAdmin and Anaconda (might change during part two)

FUNCTIONALITIES

There will be a login passage for either staffs or clients to enter the project submitting system.

Staffs:

1. Use SSN as username for the first login, can change it afterward if you desire.
2. They can view and update the status of the request. After careful consideration, viable requests will later be created to a new project.

Clients:

1. Create your own account using our submitting system initially.
2. Clients can submit request through our application, and check the request status whenever they want.

ER DIAGRAM

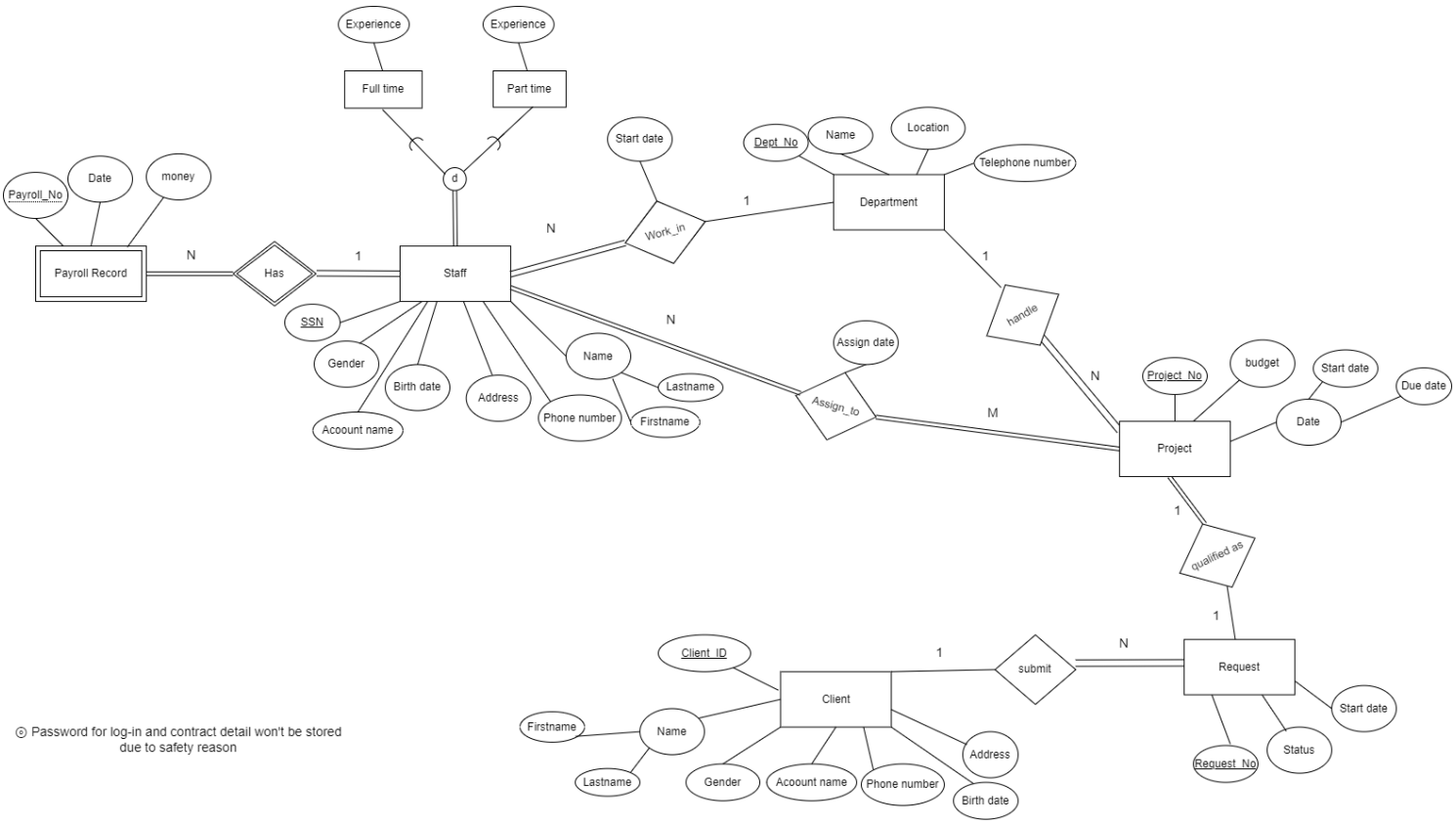


Fig 1: ER Diagram – Advertising Business Company Submission Director

RELATIONAL SCHEMA

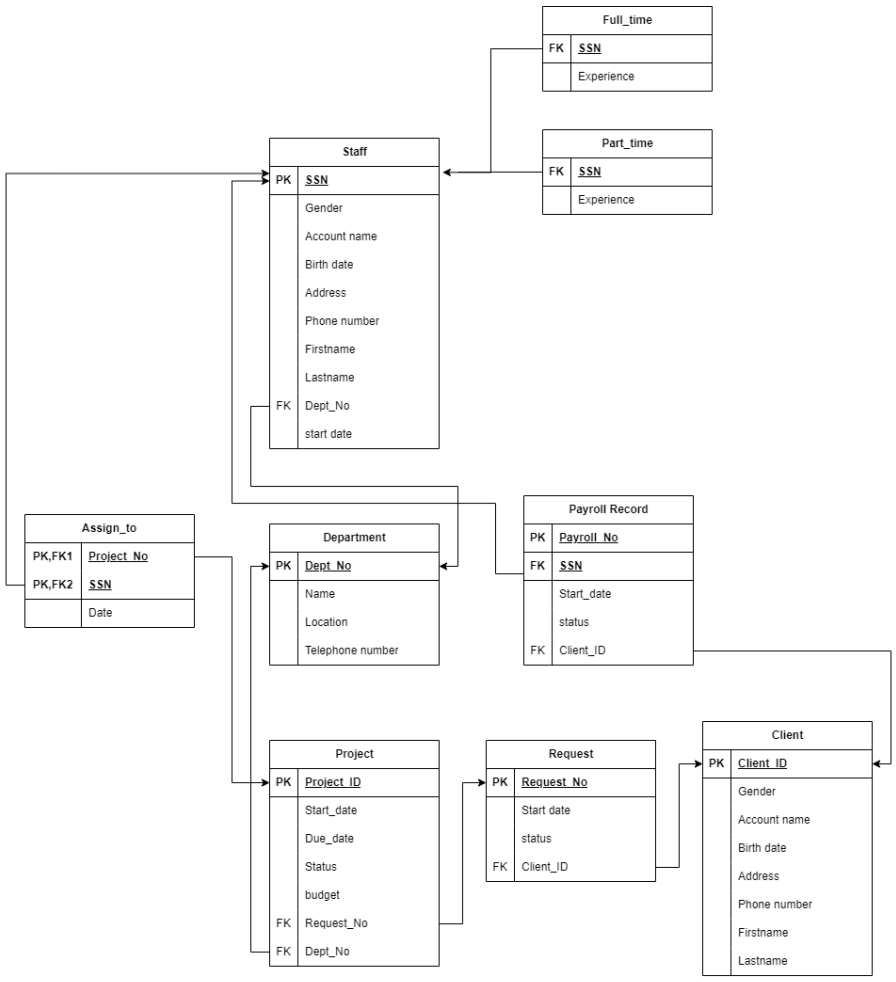


Fig 2: Relational Schema – Advertising Business Company Submission Director

Relations:

1. Staff [SSN: char, Gender: char, Account name: varchar, Birth date: date, Address: varchar, Phone number: integer, Firstname: varchar, Lastname: varchar, Dept_No: integer, start date: date]
2. Department [Dept_No: integer, Name: varchar, Location: integer, Telephone number: integer]
3. Project [Project_No: integer, start date: date, due date: date, status: varchar, budget: integer, Request_No: integer, Dept_No: integer]

4. Client [Client_ID: integer, Firstname: varchar, Lastname: varchar, Gender: char, Account name: varchar, Phone number: integer, Birth date: date, Address: varchar]
5. Request [Request_No: integer, Start date: date, status: varchar, Client_ID: integer]
6. Payroll Record [Payroll_No: integer, SSN: char, date: date, money: integer]
7. Assign_to [Project_No: integer, SSN: char, Assign date: date]
8. Full time [SSN: char, Experience: varchar]
9. Part time [SSN: char, Experience: varchar]

Foreign Key:

1. Payroll Record.SSN → Staff.SSN
2. Project.Request_No → Request.Request_No
3. Request.Client_ID → Client.Client_ID
4. Project.Dept_No → Department.Dept_No
5. Staff.Dept_No → Department.Dept_No
6. Assign_to.Project_No → Project.Project_No
7. Assign_to.SSN → Staff.SSN

Relationship	Relation between entities	Relational cardinality	Participation cardinality
Has(weak)	Staff – Payroll Record	1 -> N	Staff (T) – Payroll Record(T)
Work_in	Department - Staff	1 -> N	Department(P) – Staff(T)
Handle	Department - Project	1 -> N	Department(P) – Project(T)
Assign_to	Project - Staff	M -> N	Project(T) – Staff(T)
Qualified as	Request - Project	1 -> 1	Request(P) – Project(T)
Submit	Client - Request	1 -> N	Client(P) – Request(T)

Total participation (T)

Partial participation (P)

FUNCTIONAL DEPENDENCIES

- Staff
    - **SSN** -> Gender, Account name, Birth date, Address, Phone number, Firstname, Lastname
    - Full time:
        - **SSN** -> Experience
    - Part time:
        - **SSN** -> Experience
- Payroll Record
    - **Payroll_No**, **SSN** -> Date, money
- Department
    - **Dept_No** -> Name, Location, Telephone number
- Project
    - **Project_No** -> budget, Start date, Due date
- Request
    - **Request_No** -> Status, Start date
- Client
    - **Client_ID** -> Firstname, Lastname, Gender, Account name, Phone number, Birth date, Address

NORMALIZATION

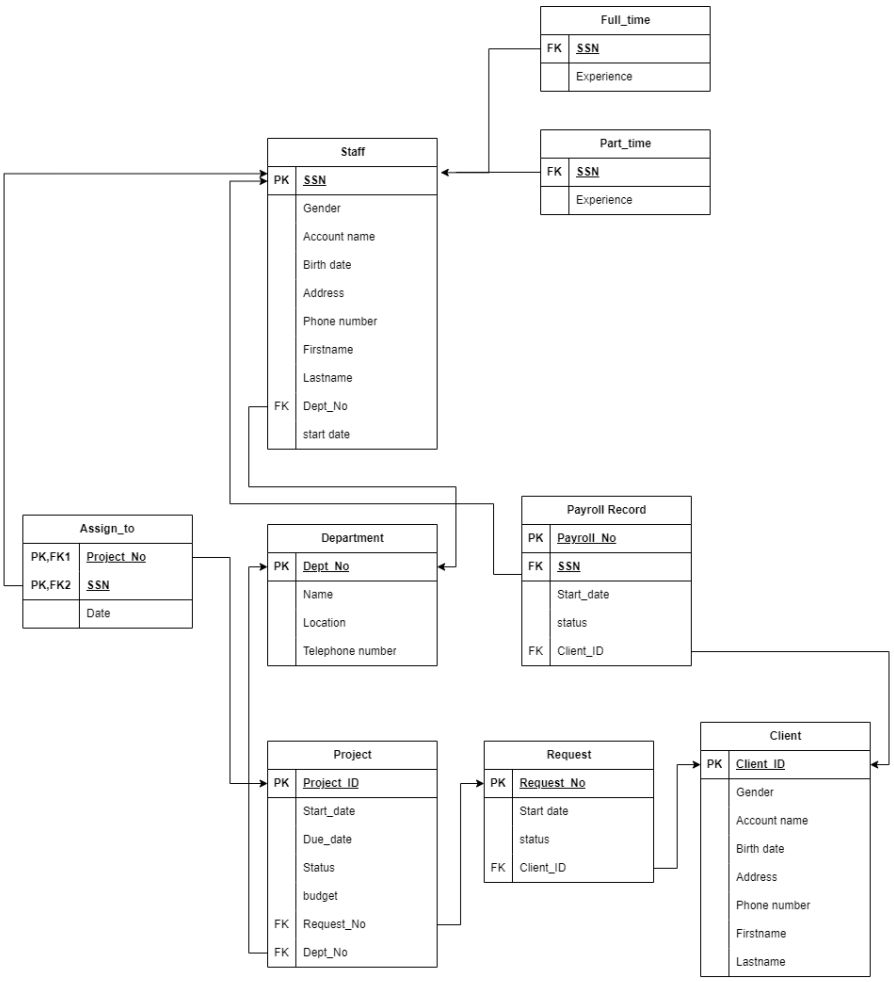


Fig 3: Normalized Schema – Advertising Business Company Submission Director

BCNF check:

- No implicit FD as there is only one FD with only one prime attribute, which is PK.
- As the FD is a superkey, BCNF is not violated.

FD name	No implicit FD	FD is Superkey	BCNF confirm
Staff	Yes	Yes	✔
Payroll Record	Yes	Yes	✔
Department	Yes	Yes	✔
Project	Yes	Yes	✔
Request	Yes	Yes	✔
Client	Yes	Yes	✔

Conclusion:

Relational schema does not violate BCNF.