Project Phase 1

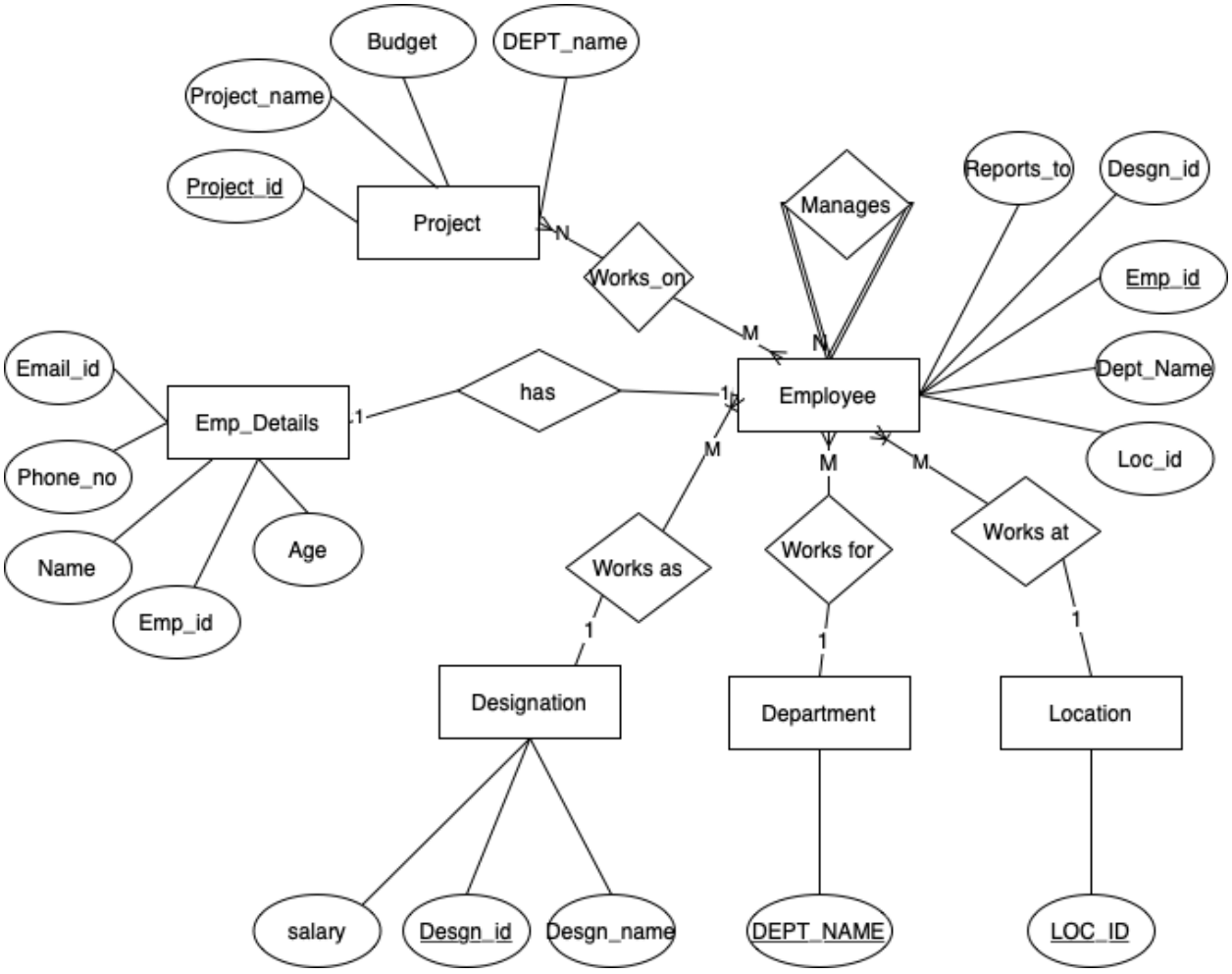
Submitted By: XMEN

- **CH N V B Dattatreya (2020201011)**
- **Chitra Kumari (2020201076)**

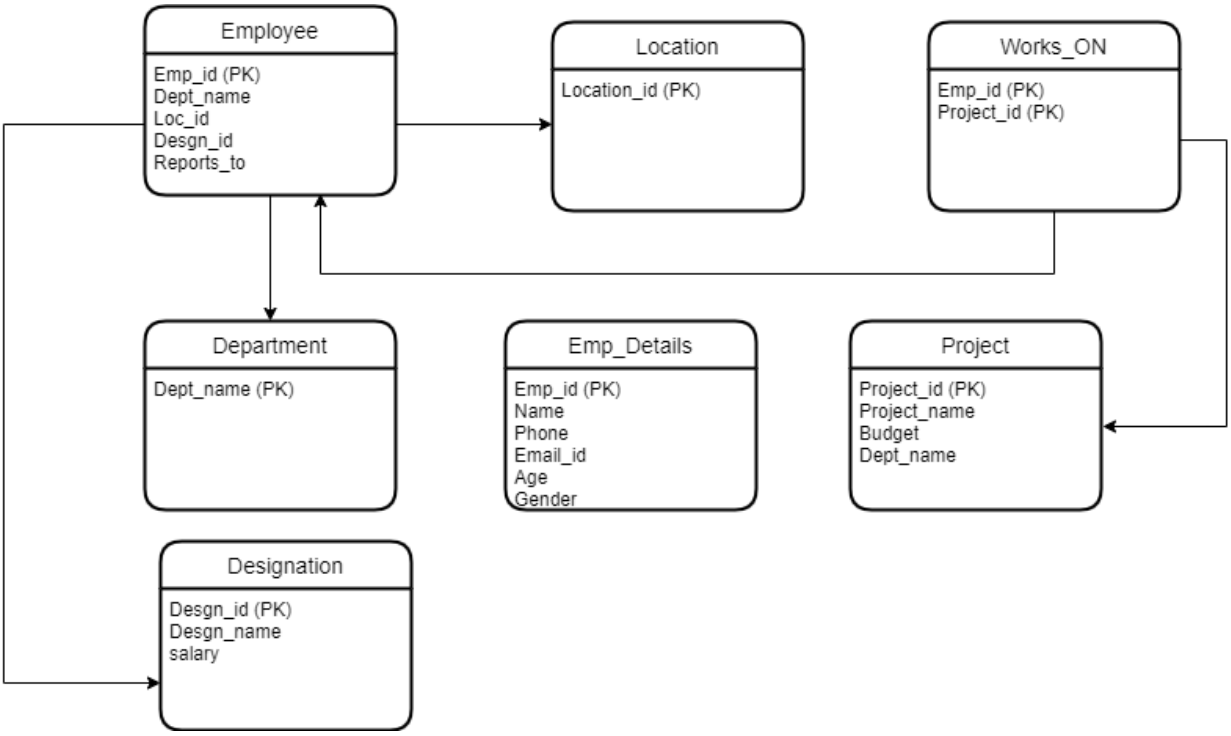
Introduction:

As a part of developing distributed database, we are using Employee database. Aspects related to an employee may be - employee details, department, project, location etc. These details are fragmented and distributed over 4 sites. Fragments and distribution to sites and performed keeping in mind the type of queries querying those details and how frequently they are queried. The motive is to present user with the correct output with minimum latency.

EMPLOYEE - ER DIAGRAM



EMPLOYEE - SCHEMA DIAGRAM



FRAGMENTATION OF EMPLOYEE DATABASE

1. **Fragmentation for Employee Table**
    a. Types of queries:
        i. Emp with dept_id = 3 (select query)
        ii. Emp with certain age (select query)
        iii. Emp with designation (select query)
        iv. How many M/F are there in a dept=hyd. (join query)
    b. Reasoning for fragmentation:
        Since, Employee has 2 broad aspects -
        - Personal Information - name, phone no, gender, email id
        - Official Information - Dept_id, designation, salary

So, Employee table was divided into 2 tables namely - Employee and Employee details.
Since, the type of queries on Employee is based majorly on department and location.
So, Employee table is horizontally divided on dept_id and location_id (4 fragments).
Employee details table is DHF on Employee (4 fragments). These 4 fragments are distributed over the given 4 sites.

	Site1	Site2	Site3	Site4
Employee	HF1	HF2	HF3	HF4

Employee details	DHF1	DHF2	DHF3	DHF4

2. **Fragmentation for Project Table**
    a. Types of queries:
        i. Number of employees in a project (join query)
        ii. Number of projects belonging to dept (group by)
        iii. Budget of a particular project (select query)
    b. Reasoning for fragmentation:
    From the type of queries, we infer that they are of two types - related to budget and related to employees in a project. At a time, queries from either of these 2 sets are asked. So, the project table is Vertically Fragmented as - {project_id, Budget} and {project_id, project_name, dept_name}

    Since, query types have join with either Employee table or Department table, so these fragments are replicated over given 4 sites as:
    - {project_id, Budget} on site3,site1
    - {project_id, project_name, dept_name} on site2, site4

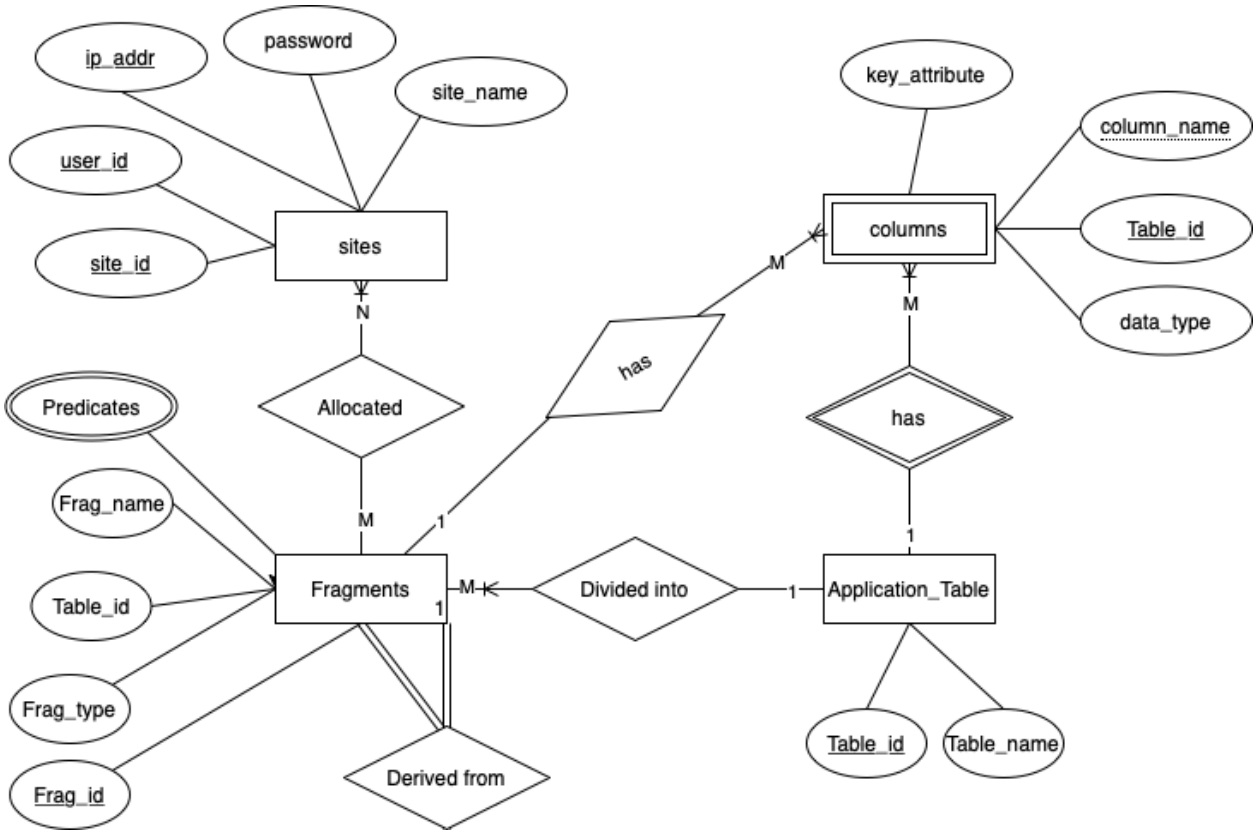
	Site1	Site2	Site3	Site4
Project	VF1	VF2	VF1	VF2

3. **Fragmentation for Works On Table**
    a. Types of queries:
        i. Number of projects by given employee (aggregate)
        ii. Number of employees by given project (aggregate)
    b. Reasoning for fragmentation:
    Since, queries mentioned above related to employee, so this table is Derived Horizontally Fragmented on Employee table. As employee table has 4 tables, so this table also gets fragmented into 4 pieces.

Table	Site1	Site2	Site3	Site4
Employee	H1	H2	H3	H4
Works On	DHF1	DHF2	DHF3	DHF4

4. **Fragments for Department, Designation and Location**
   Since data in all these tables are very less as an organization has limited number of departments and location, so there is no point fragmenting these. If we fragment it, the processing of queries will add to latency unnecessarily.
   - Department - site1
   - Location - site2
   - Designation - site3

SYSTEM CATALOG - ER DIAGRAM



SYSTEM CATALOG - SCHEMA DIAGRAM