CHAPTER 1

INTRODUCTION

Online job portal the main aim of this project is to develop an online search Portal for the people who are looking for jobs and for the companies to publish about the vacancies in it and find the best employee. It is an online website that can be accessed from anywhere, everyone is allowed access to search through once he/she has signed up to the website, with their login credentials, Online job portal makes job search a simple task where you need to just login to the website and you get all information's about the vacancies in the companies, and it made the task of companies easier by providing a free place to advertise about their vacancies

.

Earlier searching for job was a big task where the Jobseeker had to visit the agencies for applying for the jobs, the agency would show the list of jobs available according to his/her qualification, and the jobseeker could apply through agencies.

This project is been planned to be have the view of distributed architecture, with a centralized storage of the database where we store information about the jobseekers, employer, company and available jobs. The storing of the data has been planned using the constructs of PhpMyAdmin, My Sql, and Xamp Server and all the user interfaces have been designed using the HTML, CSS, JavaScript technologies. The front end is connected to the database using the php code.

CHAPTER 2

ANALYSIS AND DESIGN

The chapter tells about the overview of the project is to implement the Job portal Website with a centralized storage satisfaction of the jobseeker and the companies. The normal process which was followed until recently was through agencies or ads, the Company had to hire agency to hire employees for them or publish ads through newspapers or internet about the job vacancies, the job seeker had to visit the agency or get information through other sources, which was a time and consuming task.

2.1 DRAWBACK OF EXISTING SYSTEM

- Existing system are too time consuming, it cannot provide quick solution.
- Cost effective – High service charges by agencies and for advertising
- The process is very lengthy with unwanted formalities.

The purpose of developing the online job portal is to provide the jobseekers a free platform for finding a right and a satisfactory job according to their qualification and skills. It also connects the job seekers with the major companies and it allows the company to post their vacancies and get the best employees.

2.2 Need for new System

- The proposed Job Portal system consists of 4 modules: The Jobseeker, The Employer, The Administrator and Jobs.
- Online Job Portal will provide the fast operation the user can access from anywhere anytime with less formalities,
- Cost Inexpensive – It doesn't charge any commission as the agencies, and cost for advertising is also saved
- Easy job search, and Easy Advertising about the vacancies.

CHAPTER 3

DATABASE CONCEPTS

3.1 DATABASE MANAGEMENT SYSTEM

Database Management System is a collection of programs or **system software** which is used for storing the data, modifying the data and extracting the information from the database.

DATABASE

Database is an **organized format** of storing the data that are entered in the DBMS. In **RDBMS**(**R**elational **D**atabase **M**anagement **S**ystem) the data are organized in the form of rows and columns. By default MYSQL software stores the data in the form of rows and columns.

DATABASE SCHEMA

Schema is a **framework** using which the database for the particular concept is being created. Therefore, Schema is specified during the database design and is not expected to change the data frequently once the data are entered as it leads to ambiguity.

The Schma defines what are the tables, fields and relationships between the fields and tables of the database.

ENTITY

Entity represents a **real world object**. It can be an object with physical independence such as person, car, employee etc or it may be an object with conceptual existence such as company, job etc.

Each Entity can have any number of attributes to define what the entity is.

ATTRIBUTE

The **property of entities** are called as **attributes** i.e.s., the functionality of the entities which defines it.

TYPES OF THE ATTRIBUTE

1) Simple Attribute
2) Composite attribute
3) Single valued attribute
4) Multi valued attribute
5) Derived attribute
6) Null value

1) SIMPLE ATTRIBUTE

Simple attributes are those which cannot be divided into parts.

Example:

Phone number,Ticket number

2) COMPOSITE ATTRIBUTE

Composite attributes are those which can be divided into smaller parts and each part defines an attribute.

Example:

Name which can be divided into **Firstname**, **Middlename** and **Lastname**

3) SINGLE VALUED ATTRIBUTE

Single valued attributes are those where attribute contains single value..

Example:

Gender,Age

4) MULTI VALUED ATTRIBUTE

Multi valued attributes are those which contains more than one values for the attribute.

Example:

Address, phone number

5) DERIVED ATTRIBUTE

Derived attributes are those that does not exist in physical database but the values are derived from other attribute present in the database.

6) NULL VALUE

If any of the attributes are not applicable for the particular entity then that attribute can be given a **NULL** value.

3.2 RELATIONSHIPS

Relationships are those which describes how two tables are related to each other.

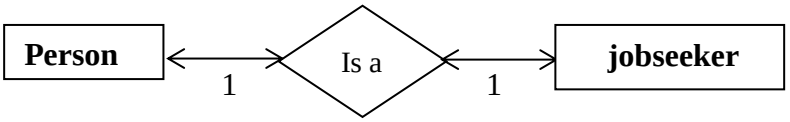
TYPES OF RELATIONSHIPS BASED ON THE RATIO

1) 1 TO 1
2) 1 TO N
3) N TO 1
4) N TO N

1) 1 TO 1

One row in table A can have **only one matching** row in table B.

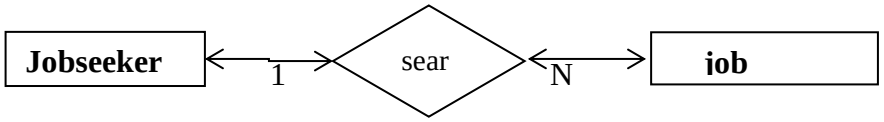
Example:



2) 1 TO N

One row in table A can have **more than one matching** rows in table B.

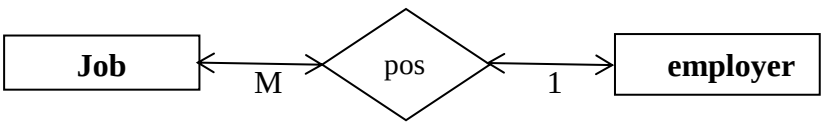
Example:



3) N TO 1

One row in table A can have **many matching** rows in table B but **not vice versa**.

Example:



3.3 ER DIAGRAM

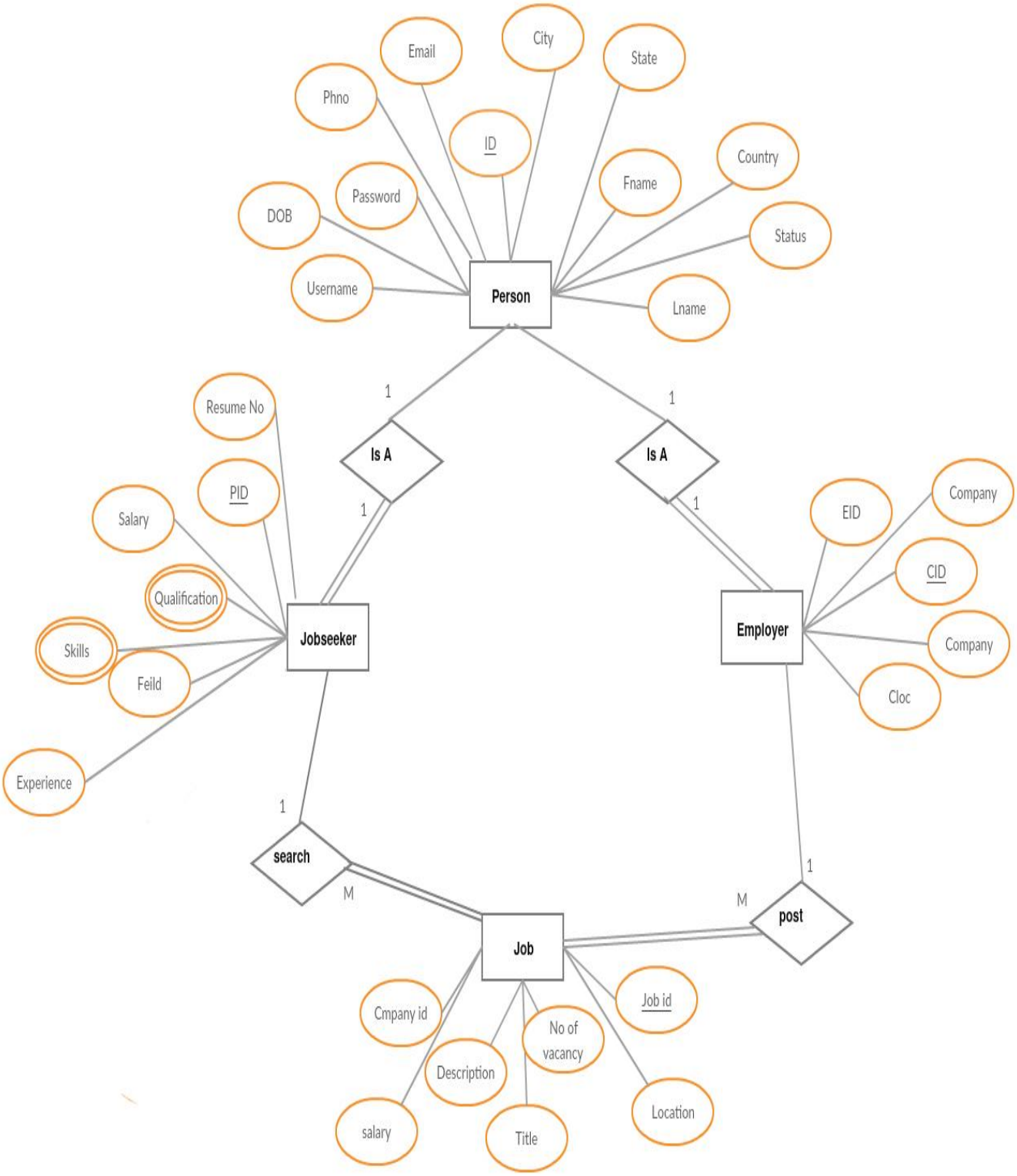


FIGURE 3.3: E-R DIAGRAM

- According to this ER- diagram the tables are created
- The relation between person and jobseeker is 1:1 that means a person can have only one jobseeker account with a unique ID.
- The relation between person and Employer is 1:1 that means a person can have only one Employer account with a unique ID associated to a company.
- The relation between Employer and job is 1:M which means a employer can post many jobs
- The relation between jobseeker and job is 1:M which means a jobseeker can apply to many jobs

3.4 RELATIONAL SCHEMA

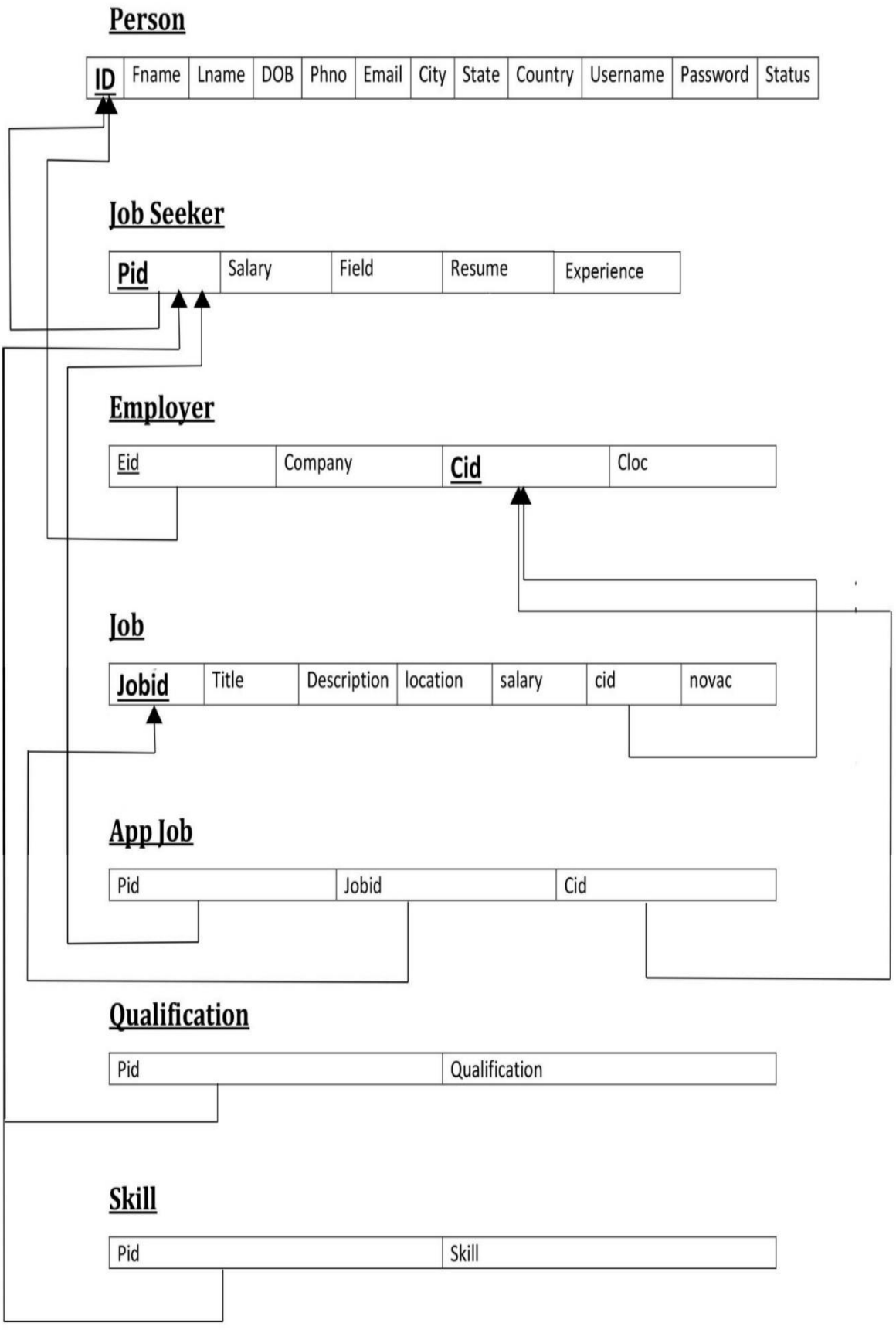


FIGURE 3.4: RELATIONAL SCHEMA

The database schema of the job portal is designed from the ER diagram, it is structure described in a formal language supported by database management system (DBMS). It is the blue print of the database how it is constructed. Basically it shows how the tables has to be created, its attributes, and relationships between tables.

All the entities are connected to the respective tables.

- When there is a N:1 relationship then the table at N side should have the primary key of the table at 1 side.
- When there is a M:N relationship then s new with the primary keys of the both tables and an attributes is created.
- The tables in the database are created according to the designing schema.

CHAPTER 4

IMPLEMENTATION

4.1 TABLE CREATION:

PERSON

CREATE TABLE person (fname varchar(20) DEFAULT NULL, lname varchar(20) DEFAULT NULL,DOB date DEFAULT NULL, id int(5) PRIMARY KEY, phno int(10) NOT NULL, email varchar(20) NOT NULL, city varchar(20) NOT NULL, state varchar(20) NOT NULL, country varchar(20) DEFAULT 'INDIA, usname varchar(20) DEFAULT NULL, Password varchar(20) DEFAULT NULL, status varchar(10) DEFAULT NULL);

JOBSEEKER

CREATE TABLE jobseeker (pid int(5) NOT NULL PRIMARY KEY, Esalary int(10) DEFAULT NULL feild varchar(10) DEFAULT NULL);

ALTER TABLE jobseeker ADD CONSTRAINT jobseeker FOREIGN KEY (pid) REFERENCES person (id);

EMPLOYER

CREATE TABLE employer (eid int(5) DEFAULT NULL, company varchar(20) DEFAULT NULL, cid int(5) PRIMARY KEY, cloc varchar(20) DEFAULT NULL);

ALTER TABLE employer ADD CONSTRAINT employer FOREIGN KEY (eid) REFERENCES person (id);

JOBS

CREATE TABLE `job` (jobid int(5) PRIMARY KEY, title varchar(20) NOT NULL, description varchar(50) NOT NULL, location varchar(20) NOT NULL, salary int(10) DEFAULT NULL, cid int(5) NOT NULL, novac int(5) DEFAULT NULL);

ALTER TABLE job ADD CONSTRAINT FOREIGN KEY (cid) REFERENCES employer (cid);

SKILLS

CREATE TABLE skills(pid int(5) NOT NULL, skills varchar(20) NOT NULL);

ALTER TABLE skill ADD CONSTRAINT FOREIGN KEY (pid) REFERENCES jobseeker (pid);

QUALIFICATIONS

CREATE TABLE qualifications(pid int(5) NOT NULL, qualifications varchar(20) NOT NULL);

ALTER TABLE qualifications ADD CONSTRAINT FOREIGN KEY (pid) REFERENCES jobseeker (pid);

4.2 SNAPSHOTS

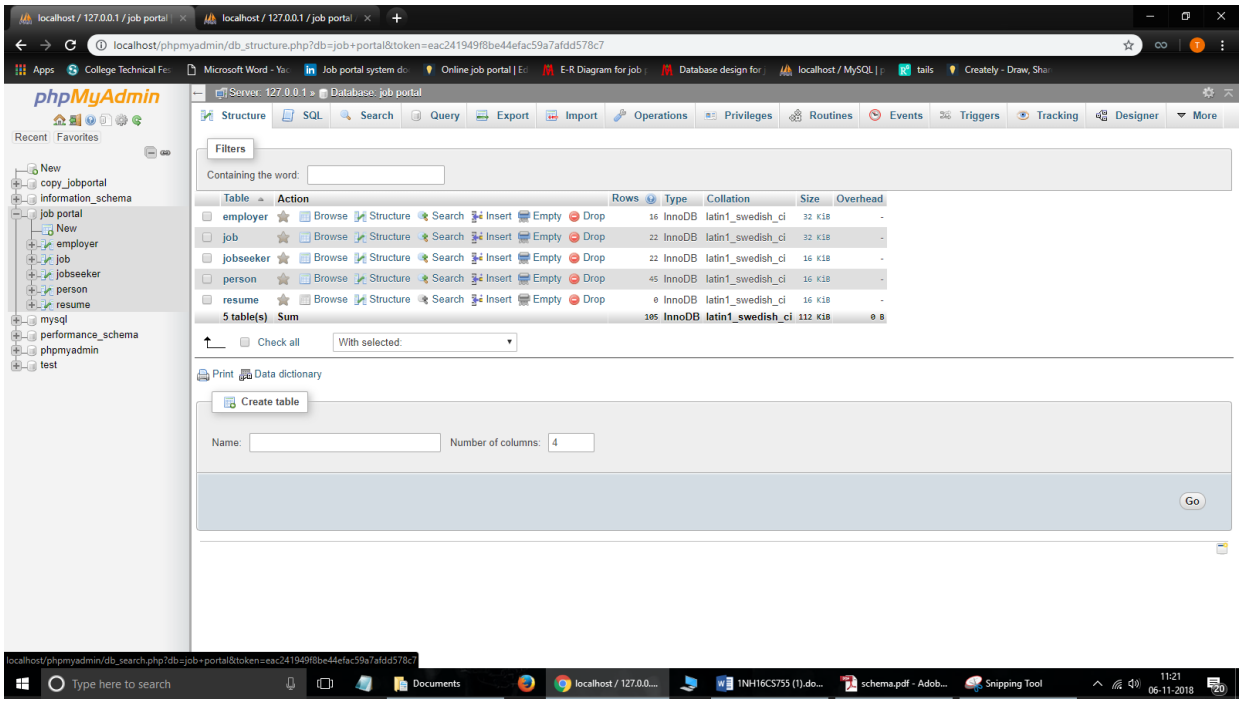


FIGURE 4.2.1: CREATION OF TABLES

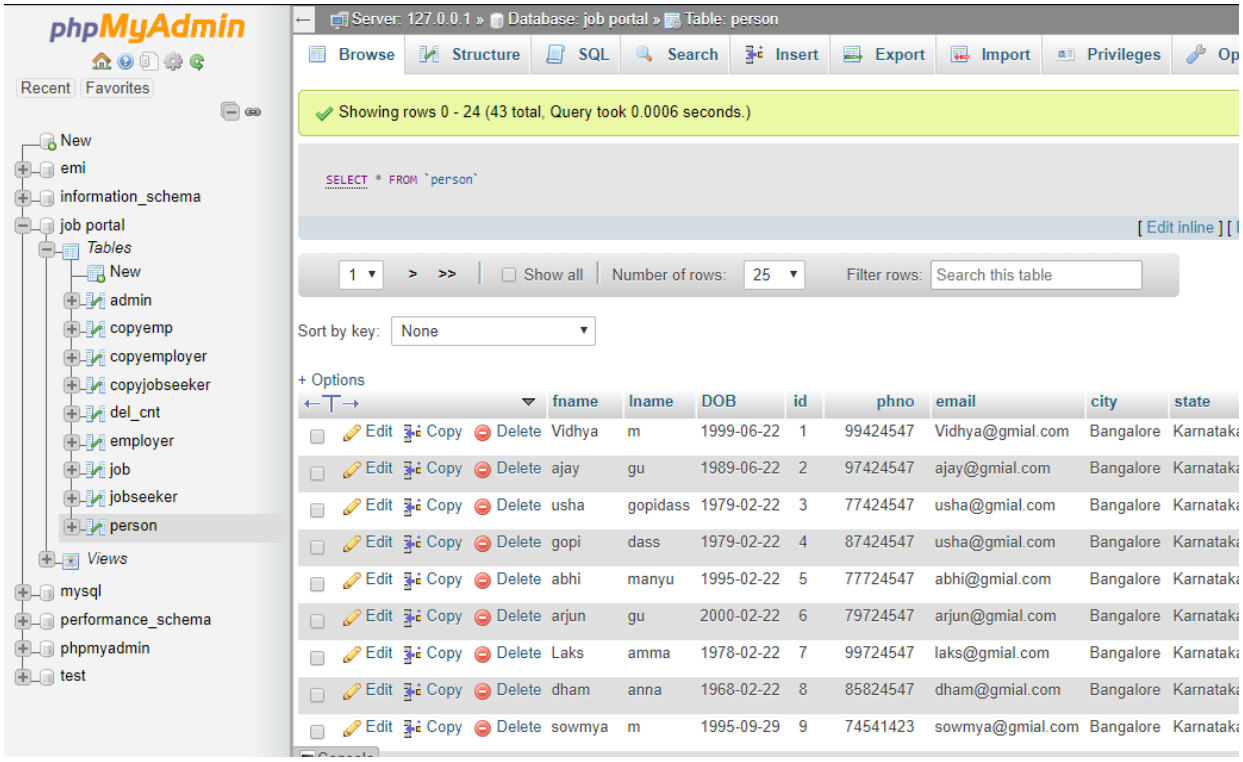


FIGURE 4.2.2: PERSON TABLE

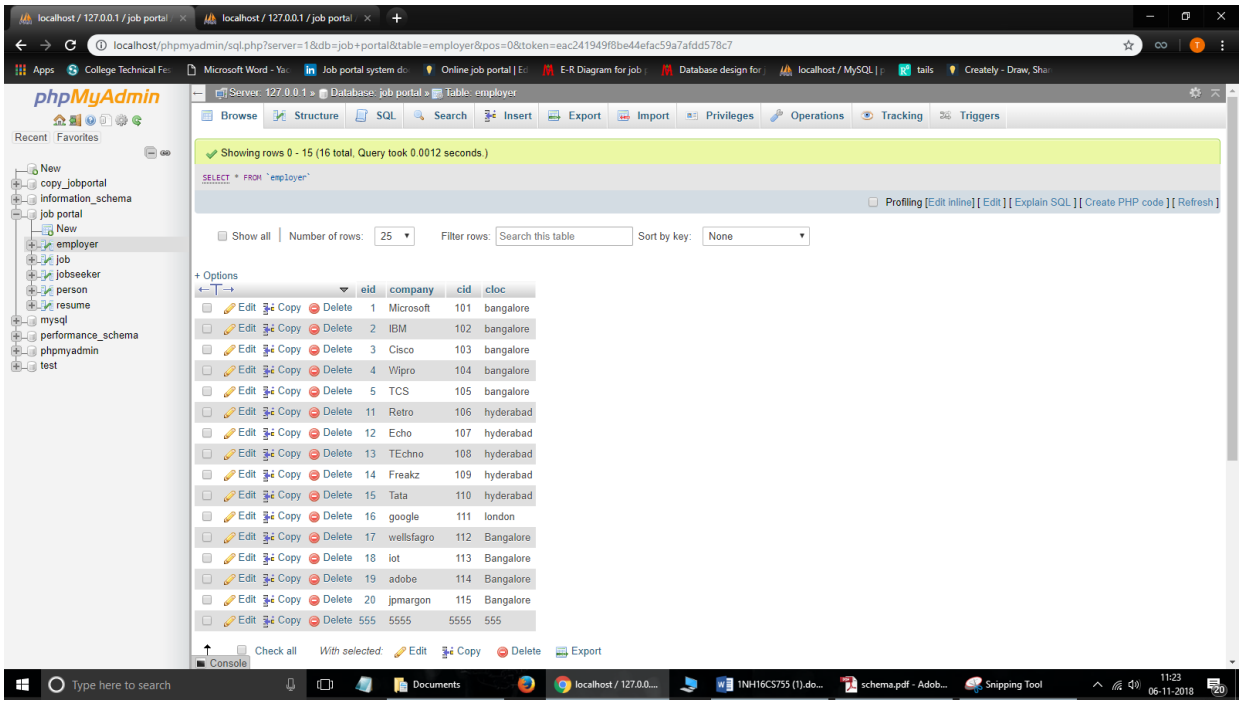


FIGURE 4.2.3: EMPLOYER TABLE

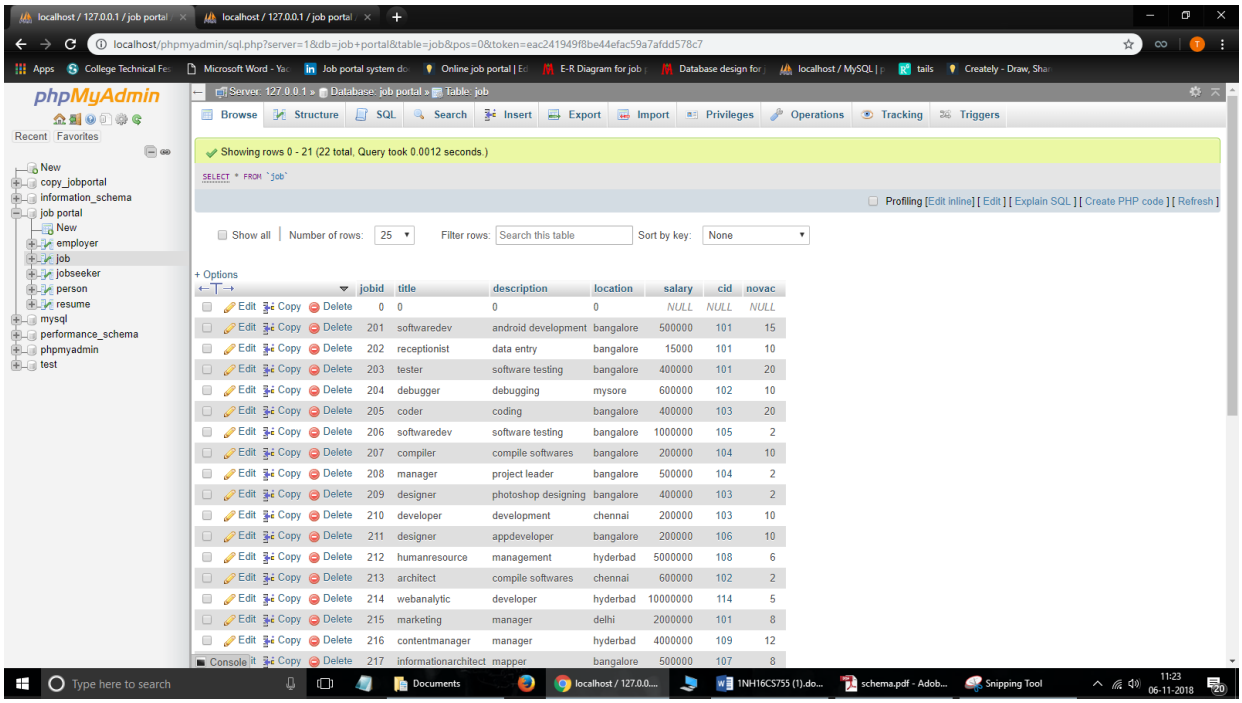


FIGURE 4.2.4: JOB TABLE

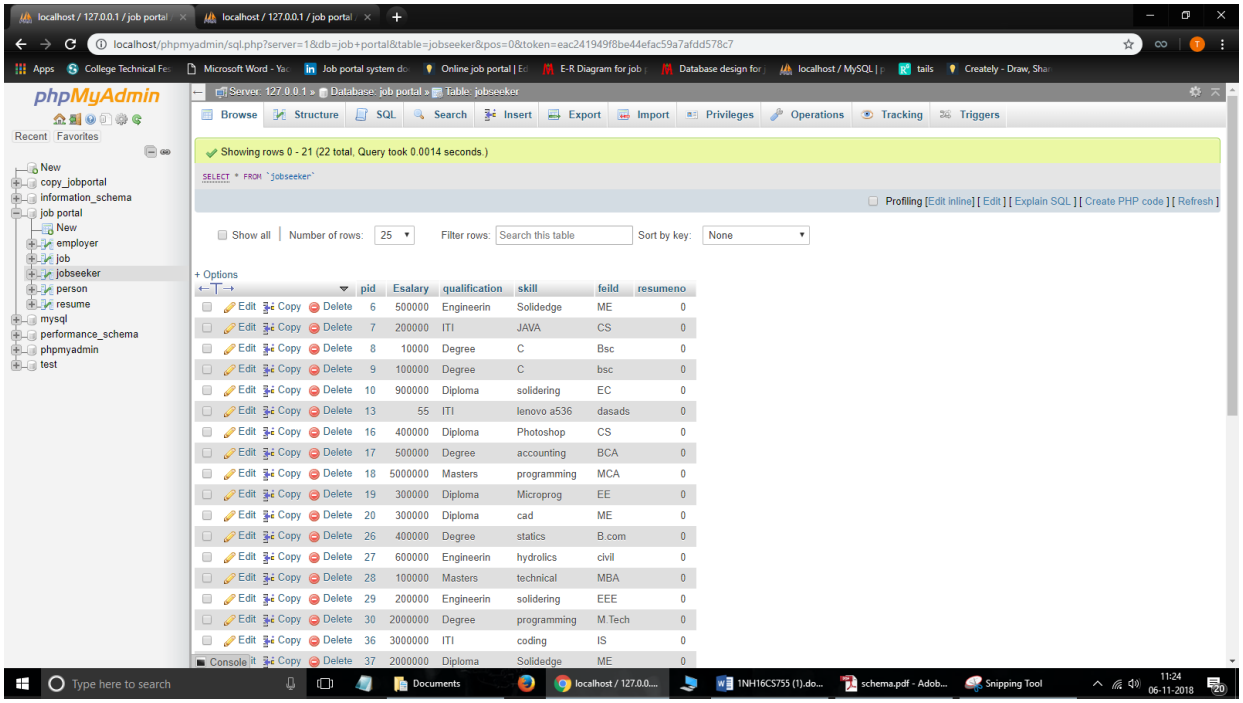


FIGURE 4.2.5: JOBSEEKER TABLE

4.3 FRONT END

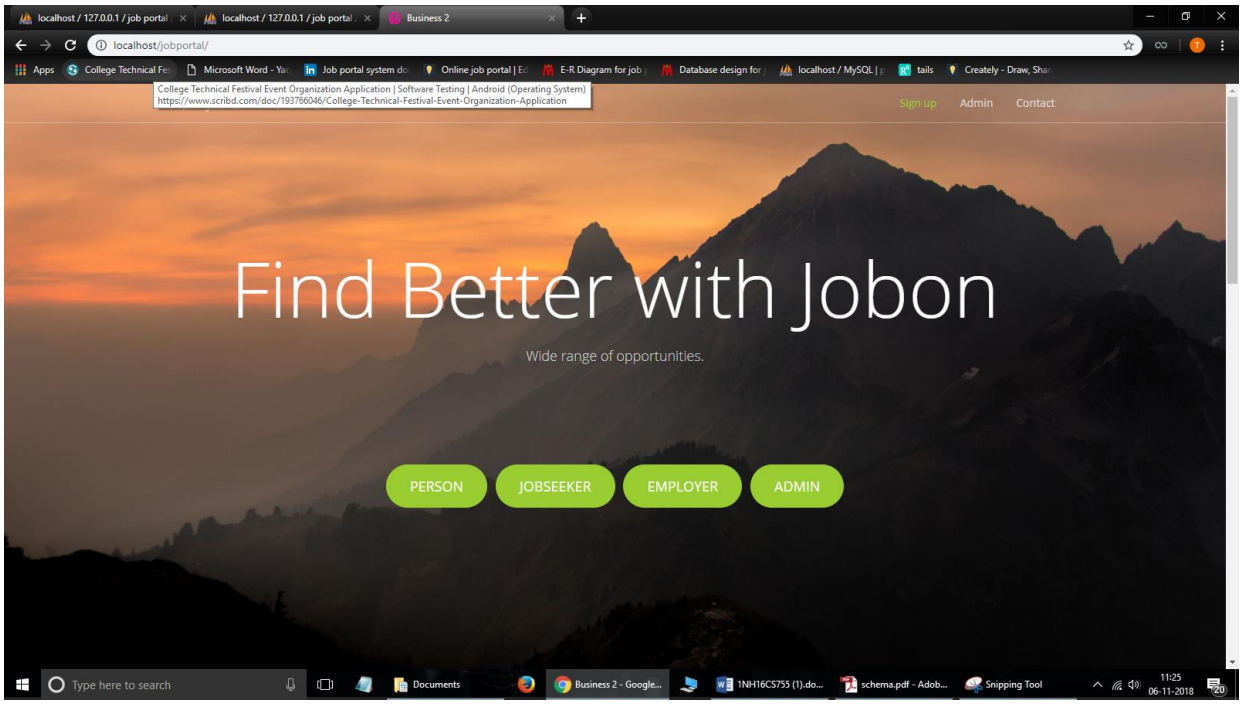


FIGURE 4.3.1: FRONT END

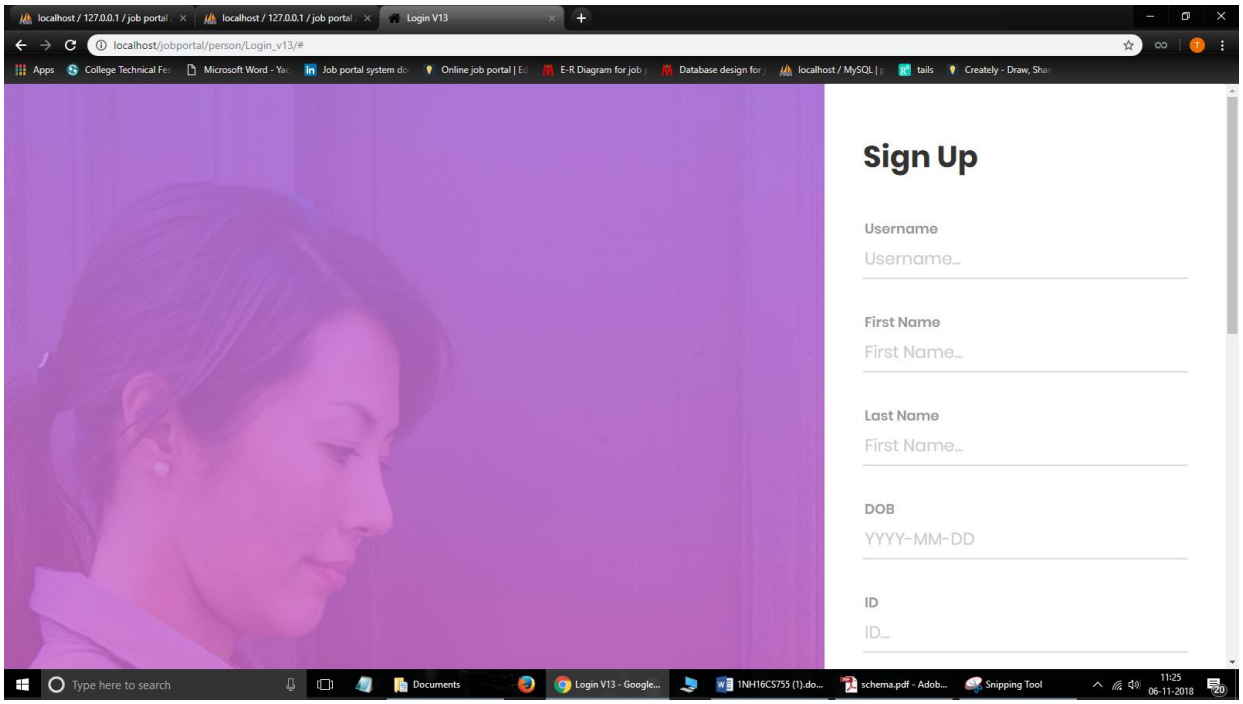


FIGURE 4.3.2: SIGN UP AS PERSON

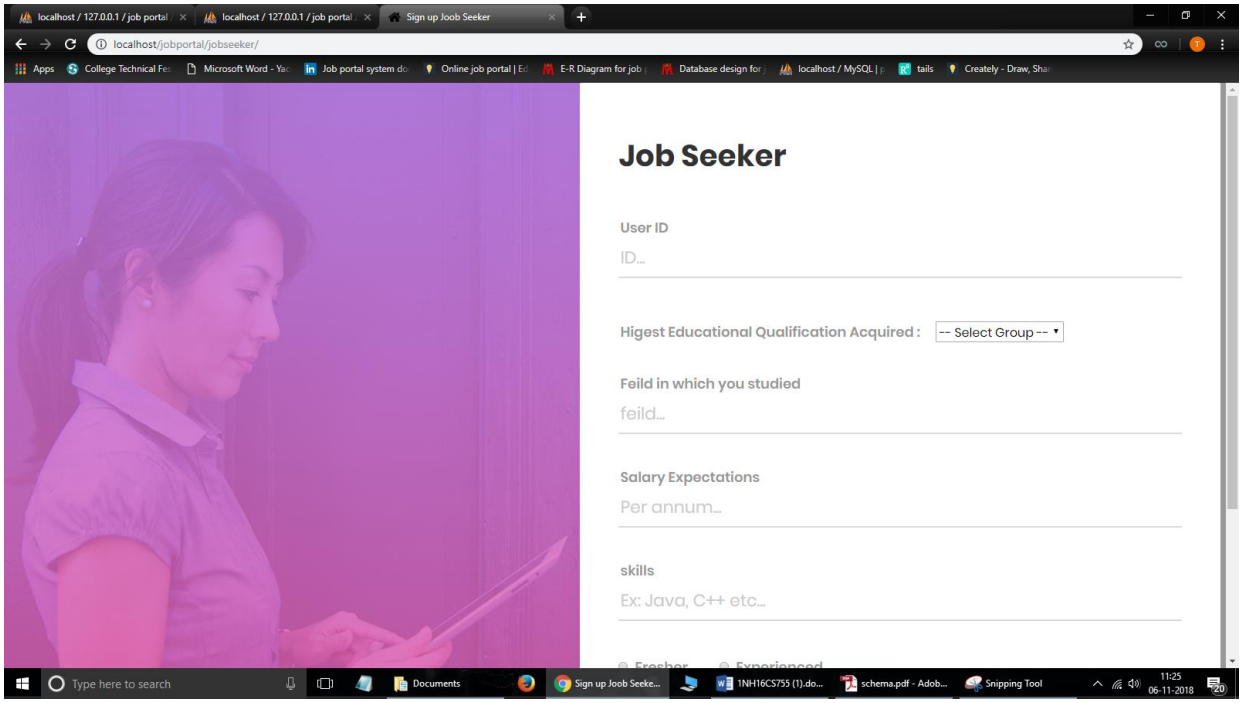


FIGURE 4.3.3: SIGN UP TO JOBSEEKER

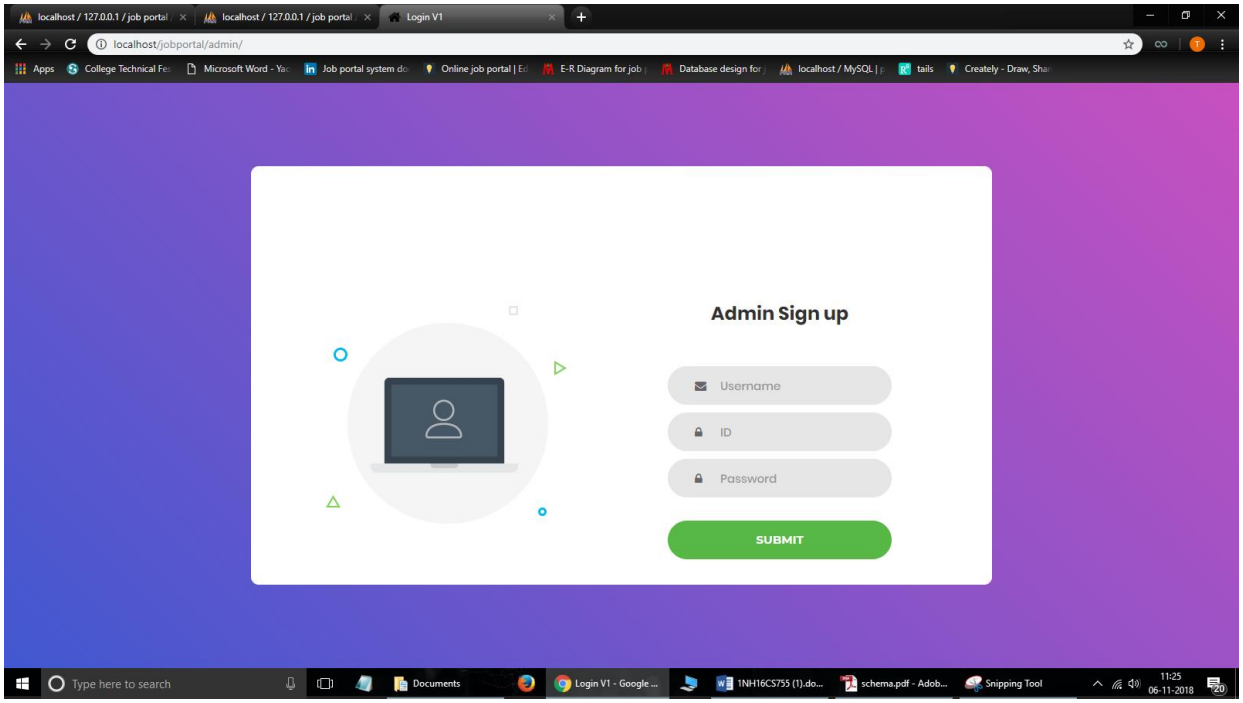


FIGURE 4.3.4: ADMIN LOGIN

CHAPTER 5

QUERIES

5.1 QUERIES ON DDL COMMAND:

1. select * from person;
2. Select pid, field, experience from jobseeker ;
3. Select fname, phno, email from person where place=’bangalore’;
4. Delete from person where name=’siddu’;
5. Alter table person add caste varchar(20) not null;
6. Drop column caste varchar(20) not null from person;
7. Alter table person rename to user;
8. Alter table person modify city varchar(50) not null;
9. Alter table person rename column city to locality;
10. Select * from jobseeker;

5.2 QUERIES ON AGGREGATION FUNCTION:

1. select count(id) from person;
2. select avg(salary) from jobseeker;
3. select max(salary) from job;
4. select min(salary) from job;
5. select sum(salary) from job;
6. Select * from person where fname like ‘%a’;
7. Select feild,sum(esalary) from jobseeker GROUP BY field
8. Select sum(esalary) from jobseeker GROUP BY feild HAVING sum(esalary) < 15000
9. Select * from person where fname like ‘ ---------- a’;
10. Select * from job where salary between 10000 and 400000;
11. Select * from person where city:’bangalore’ or city:’chennai’;
12. Select distinct(city), fname from jobseeker, person;

5.3 QUERIES ON JOINS:

1. Select p.fname, j.feild , j.pid from person p,jobseeker j where j.pid=p.id;
2. Select e.eid, e.company, j.title, j.salary from employer e, jobseeker j where j.cod=e.cid;
3. Select e.company,max(salary) from employer e, job j where j.cid=e.cid;
4. Select * from person p, jobseeker j where j.pid=p.id;
5. Select * from person p, employer e where e.eid=p.id AND cloc=’bangalore’;

5.4 NESTED AND CORELATED QUERIES:

1) Select * from person where city in(‘chennai’,’bangalore’);
2) Select pid, esalary,feild from jobseeker where pid in(select id from person where city="bangalore");
3) Select * from person where exists in(‘chennai’,’bangalore’);

5.5 VIEWS:

1) Create view job_details as select jopid, description from job;
2) Create view company_info as select fname, company from person p, employer e where p.id=e.eid;
3) Create view company_info as select fname, company from person p, employer e where p.id=e.eid and company=”Microsoft”;
4) Create view person_info as select fname, Esalary from person p, jobseeker j where p.id=j.pid and Esalary>2500000;
5) Create view person_details as select fname, description, company from person p, employer e, jobseeker j where p.id=e.eid and p.id=j.pid and e.company=”cisco”;

5.6 TRIGGERS:

1) Create trigger t1 before update on job for each row begin set new.location=”new.chennai” end;$$

   Update job set location=’a’ where jobid=201;

2) create trigger t2 before insert on job for each row begin if new.Esalary<25000 or

   age>2500000 then signal sqlstate '5000000' set message_text="restricted error"; end if; end;$$

3) Create trigger `cpyjobseeker` after insert on `jobseeker`

   for each row insert into copyjobseeker values(new.pid, new.Esalary, new.qualification, new.skill, new.feild, new.resumneno) end;$$

4) Create trigger `cpyemployer` after insert on `employer` for each row insert into copyemployer values(new.eid, new.company, new.cid, new.cloc);

CHAPTER 6

CONCLUSION

Finally, I conclude by saying people must do job to fulfill their needs. People are facilitated with more opportunities, they can easily login to website and apply for suitable job.

We offer documentation processing, application for those interested in applying for a job. Online job portal has greater importance and could develop such as efficient existing system which is used by lot of employers and jobseekers.

BIBILOGRAPHY

- Database management system -Bookman
- Fundamentals of database system (seventh edition) -Ramez elmsri and shamkant B.navathe
- http://en.wikibooks.org
- https://en.wikipedia.org/wiki/PhPMyAdmin.