DBMS Project Report

Employee Management System (EMS)

A Project Report Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Bachelor of Technology

In

Computer Science Department

Submitted by

Indradhar Paka
1800315c203
CSE+CSC-1
2nd year

Submitted To

Dr. Brij Bihari Dubey
Assistant Professor
Computer Science Dept.
brijbihari.dubey@bmu.edu.in

BML MUNJAL UNIVERSITY™

SCHOOL OF ENGINEERING AND TECHNOLOGY

BML MUNJAL UNIVERSITY GURGAON

MAY,2020

TABLE OF CONTENTS

ABSTRACT

Database Management Systems is an essential course in computer science curriculum, which helps students to develop a mental model of how database systems work. The designing of a database and managing systems are often complex, non-deterministic which makes them difficult for students to understand. One such concept is Employee Management System.

In practice, DBMS course involve classroom lectures describing high-level abstractions of the concepts, and students complete programming assignments to apply the material in a more concrete way. Depending on the programming assignments, this approach may leave students with only a theoretical understanding of DBMS ideas, which may be different from the actual way these concepts are implemented in a database. What many students require is a practical knowledge of database system implementation to supplement the high-level presentations of concepts taught in class or presented in a textbook.

My project covers all the key concepts which had been taught to us like: CREATE, INSERT, SELECT, FROM, WHERE, DROP, ALTER, UPDATE, DROP, TRUNCATE, VIEWS, TRIGGERS, etc.

PROBLEM STATEMENT AND MOTIVATION

"Employee Management System using MySQL"

In my project I am trying to achieve an optimal management system for employee details in an organization so that we can efficiently achieve our goal of accessing details of employee details. The system is implemented using My SQL server which was used to create a database.

The motivation of the project came to me while I am searching details of employees of an organization where they don't use this system. Then I hoped for a system which could maintain all the details of employees in a database.

INTRODUCTION AND DESCRIPTION:

Employee Management System is a distributed application, developed to maintain the details of employees working in any organization. It maintains the information about the personal details of their employees, also the details about the payroll system which enable to generate the pay slip.

It is simple to understand and can be used by anyone who is not even familiar with simple employees' system. It is user friendly and just asks the user to follow step by step operations by giving him few options. It is fast and can perform many operations of a company.

It also enables users to create and store Employee Records. The application also provides facilities of a payroll system which enables user to generate Pay slips too. This application is helpful to department of the organization which maintains data of employees related to an organization.

This software package has been developed using the powerful coding tools like My Sql. The software is very user friendly. The package contains different modules like Employee details.

PROBLEM FACED

The main problem that I faced during the initial stage was joining the tables and making primary keys and difficulties faced to draw ER diagrams.

Solution: I came up with a clever way of designing entire system and doing the simulation in My SQL Workbench. So, I learned how to do it in a stable manner and able to create the project in a stable manner.

DETAILED EXPLANATION

Database Management System is a collection of programs which enables users to access database, manipulate data, and represent data.

A technology to store and retrieve data with utmost efficiency along with appropriate security measures.

Now, I am going to create the database as "**Project**" and creating tables in my database.

Creating tables using "create "command

```
mysql> create table employee(fname varchar(15) NOT NULL,lname varchar(15) NOT NULL,mname char NULL,ssn char(9) NOT NULL,
bdate date,adress varchar(35) NOT NULL,sex char NULL,salary decimal(10,2) NULL,super_ssn char(9) NULL, dno int NOT NULL)
;
Query OK, 0 rows affected (0.63 sec)

mysql> desc employee;
+-----------+---------------+------+-----+---------+-------+
| Field     | Type          | Null | Key | Default | Extra |
+-----------+---------------+------+-----+---------+-------+
| fname     | varchar(15)   | NO   |     | NULL    |       |
| lname     | varchar(15)   | NO   |     | NULL    |       |
| mname     | char(1)       | YES  |     | NULL    |       |
| ssn       | char(9)       | NO   |     | NULL    |       |
| bdate     | date          | YES  |     | NULL    |       |
| adress    | varchar(35)   | NO   |     | NULL    |       |
| sex       | char(1)       | YES  |     | NULL    |       |
| salary    | decimal(10,2) | YES  |     | NULL    |       |
| super_ssn | char(9)       | YES  |     | NULL    |       |
| dno       | int           | NO   |     | NULL    |       |
+-----------+---------------+------+-----+---------+-------+
10 rows in set (0.12 sec)
```

Similarly, I have created tables for Employee, department, dept_loc, project, works_on, dependent.

```
mysql> create table department(Dname varchar(15)not null,Dnumber int not null,mgr_ssn char(9) not null,mgr_start_date da
te null,primary key (Dnumber));
Query OK, 0 rows affected (0.91 sec)

mysql> create table dept_loc(Dnumber int not null,Dloc varchar(20)not null,primary key(Dnumber,Dloc));
Query OK, 0 rows affected (0.78 sec)

mysql> create table project(pname varchar(15) not null,pnumber int not null,ploc varchar(30),Dnum int not null,primary k
ey(pnumber),unique(pname));
Query OK, 0 rows affected (0.58 sec)

mysql> create table works_on(Essn char(9) not null,pno int not null,hours decimal(3,1)not null,primary key(Essn,pno));
Query OK, 0 rows affected (0.66 sec)

mysql> create table dependent(Essn char(9) not null,dependent_name varchar(15)not null,sex char,Bdate date,relationship
varchar(10),primary key(Essn,dependent_name));
Query OK, 0 rows affected (1.01 sec)
```

Now, lets see the tables created and describe each table.

MySQL 8.0 Command Line Client

```
mysql> show tables;
+--------------------+
| Tables_in_project  |
+--------------------+
| department         |
| dependent          |
| dept_loc           |
| employee           |
| project            |
| works_on           |
+--------------------+
6 rows in set (0.12 sec)

mysql> desc dependent;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| Essn            | char(9)     | NO   | PRI | NULL    |       |
| dependent_name  | varchar(15) | NO   | PRI | NULL    |       |
| sex             | char(1)     | YES  |     | NULL    |       |
| Bdate           | date        | YES  |     | NULL    |       |
| relationship    | varchar(10) | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
5 rows in set (0.11 sec)

mysql> desc dept_loc;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Dnumber | int         | NO   | PRI | NULL    |       |
| Dloc    | varchar(20) | NO   | PRI | NULL    |       |
+---------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

```
mysql> desc employee;
+-----------+---------------+------+-----+---------+-------+
| Field     | Type          | Null | Key | Default | Extra |
+-----------+---------------+------+-----+---------+-------+
| fname     | varchar(15)   | NO   |     | NULL    |       |
| lname     | varchar(15)   | NO   |     | NULL    |       |
| mname     | char(1)       | YES  |     | NULL    |       |
| ssn       | char(9)       | NO   | PRI | NULL    |       |
| bdate     | date          | YES  |     | NULL    |       |
| adress    | varchar(35)   | NO   |     | NULL    |       |
| sex       | char(1)       | YES  |     | NULL    |       |
| salary    | decimal(10,2) | YES  |     | NULL    |       |
| super_ssn | char(9)       | YES  |     | NULL    |       |
| dno       | int           | NO   |     | NULL    |       |
+-----------+---------------+------+-----+---------+-------+
10 rows in set (0.00 sec)

mysql> desc project;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| pname   | varchar(15) | NO   | UNI | NULL    |       |
| pnumber | int         | NO   | PRI | NULL    |       |
| ploc    | varchar(30) | YES  |     | NULL    |       |
| Dnum    | int         | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> desc works_on;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| Essn  | char(9)     | NO   | PRI | NULL    |       |
| pno   | int         | NO   | PRI | NULL    |       |
| hours | decimal(3,1)| NO   |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql>
```

The tables are:

- Employee: fname, lname, mname, ssn, bdate, address, sex, salary, superssn, dno
- Department: Dname, Dnumber, mgr_ssn, mgr_start_date
- Dependent: Essn, dependent_name, sex, Bdate, relationship
- Department location: Dnumber, Dloc
- Project: pname, pnumber, ploc, Dnum
- Works on: Essn, pno, hours

After creating tables now inserting values into the tables using "**insert**" command.

Example: insert into employee values('JC2','IP2','b','112233445','1955-01-09','BML NH,Delhi','M','50000','123456789','2');

```
mysql> insert into employee values('JC2','IP2','b','112233445','1955-01-09','BML NH
,Delhi','M','50000','123456789','2');
Query OK, 1 row affected (0.34 sec)

mysql> insert into employee values('JC3','IP3','c','112233254','1956-01-09','H,Delh
i','M','50000','123456780','3');
Query OK, 1 row affected (0.18 sec)

mysql> insert into employee values('JC4','IP4','d','112233245','1957-01-09','H,MP',
'F','500001','123456770','4');
Query OK, 1 row affected (0.09 sec)

mysql> insert into employee values('JC5','IP5','e','116233245','1958-01-09','H,LK',
'F','508001','123458770','5');
Query OK, 1 row affected (0.17 sec)
```

Example: insert into department values('Research','5','112223333','1988-05-22');

```
mysql> insert into department values('administration','1','110253373','1989-05-22')
;
Query OK, 1 row affected (0.19 sec)

mysql> insert into department values('trainer','2','110254573','1990-05-22');
Query OK, 1 row affected (0.19 sec)

mysql> insert into department values('intern','3','110457573','1991-05-22');
Query OK, 1 row affected (0.17 sec)

mysql> insert into department values('managing','4','110455573','1992-05-22');
Query OK, 1 row affected (0.07 sec)

mysql> insert into dependent values('123456897','JCN','M','1983-10-25','son');
Query OK, 1 row affected (0.11 sec)
```

Example: insert into dependent values('123456897','JCN','M','1983-10-25','son');

```
mysql> insert into dependent values('123451234','JCF','F','1984-10-25','daughter');

Query OK, 1 row affected (0.13 sec)

mysql> insert into dependent values('001151234','JCM','F','1970-10-25','mother');
Query OK, 1 row affected (0.15 sec)

mysql> insert into dependent values('001151456','JCF','M','1969-10-25','father');
Query OK, 1 row affected (0.10 sec)

mysql> insert into dependent values('001158879','JCU','M','1985-10-25','uncle');
Query OK, 1 row affected (0.16 sec)
```

Example: insert into dept_loc values('1','JCH');

```
mysql> insert into dept_loc values('2','US');
Query OK, 1 row affected (0.18 sec)

mysql> insert into dept_loc values('3','Ts');
Query OK, 1 row affected (0.17 sec)

mysql> insert into dept_loc values('4','AP');
Query OK, 1 row affected (0.08 sec)

mysql> insert into dept_loc values('5','BML');
Query OK, 1 row affected (0.40 sec)
```

Example: insert into project values('product A','1','JCB','5');

```
mysql> insert into project values('product b','2','TS','2');
Query OK, 1 row affected (0.25 sec)

mysql> insert into project values('product c','3','AP','3');
Query OK, 1 row affected (0.15 sec)

mysql> insert into project values('product d','4','BML','4');
Query OK, 1 row affected (0.12 sec)
```

Example: insert into works_on values('112233556','1','7.5');

```
mysql> insert into works_on values('123456789','2','9');
Query OK, 1 row affected (0.08 sec)

mysql> insert into works_on values('123400889','3','10');
Query OK, 1 row affected (0.07 sec)

mysql> insert into works_on values('987654321','4','11');
Query OK, 1 row affected (0.17 sec)

mysql> insert into works_on values('987012345','5','5');
Query OK, 1 row affected (0.21 sec)
```

Now let's select all the tables to see the data using "**select**" command.

- Select * from employee;
- Select * from works_on;
- Select * from department;
- Select * from dependent;
- Select * from project;

```
mysql> select * from works_on;
+-----------+-----+-------+
| Essn      | pno | hours |
+-----------+-----+-------+
| 112233556 |   1 |   7.5 |
| 123400889 |   3 |  10.0 |
| 123456789 |   2 |   9.0 |
| 987012345 |   5 |   5.0 |
| 987654321 |   4 |  11.0 |
+-----------+-----+-------+
5 rows in set (0.00 sec)

mysql> select * from project;
+-----------+---------+------+------+
| pname     | pnumber | ploc | Dnum |
+-----------+---------+------+------+
| product A |       1 | JCB  |    5 |
| product b |       2 | TS   |    2 |
| product c |       3 | AP   |    3 |
| product d |       4 | BML  |    4 |
| product e |       5 | US   |    1 |
+-----------+---------+------+------+
5 rows in set (0.00 sec)

mysql> select * from department;
+----------------+---------+-----------+-----------------+
| Dname          | Dnumber | mgr_ssn   | mgr_start_date |
+----------------+---------+-----------+-----------------+
| administration |       1 | 110253373 | 1989-05-22      |
| trainer        |       2 | 110254573 | 1990-05-22      |
| intern         |       3 | 110457573 | 1991-05-22      |
| managing       |       4 | 110455573 | 1992-05-22      |
| Research       |       5 | 112223333 | 1988-05-22      |
+----------------+---------+-----------+-----------------+
5 rows in set (0.11 sec)
```

```
mysql> select * from dependent;
+-----------+----------------+------+------------+-------------+
| Essn      | dependent_name | sex  | Bdate      | relationship |
+-----------+----------------+------+------------+-------------+
| 001151234 | JCM            | F    | 1970-10-25 | mother      |
| 001151456 | JCF            | M    | 1969-10-25 | father      |
| 001158879 | JCU            | M    | 1985-10-25 | uncle       |
| 123451234 | JCF            | F    | 1984-10-25 | daughter    |
| 123456897 | JCN            | M    | 1983-10-25 | son         |
+-----------+----------------+------+------------+-------------+
5 rows in set (0.00 sec)

mysql> select * from employee;
+-------+-------+-------+-----------+------------+-------------+------+-----------+-----------+-----+
| fname | lname | mname | ssn       | bdate      | adress      | sex  | salary    | super_ssn | dno |
+-------+-------+-------+-----------+------------+-------------+------+-----------+-----------+-----+
| JC4   | IP4   | d     | 112233245 | 1957-01-09 | H,MP        | F    | 500001.00 | 123456770 |   4 |
| JC3   | IP3   | c     | 112233254 | 1956-01-09 | H,Delhi     | M    |  50000.00 | 123456780 |   3 |
| JC2   | IP2   | b     | 112233445 | 1955-01-09 | BML NH,Delhi | M    |  50000.00 | 123456789 |   2 |
| JC5   | IP5   | e     | 116233245 | 1958-01-09 | H,LK        | F    | 508001.00 | 123458770 |   5 |
| JC1   | IP1   | A     | 123456789 | 1965-01-09 | 254 AP,TG   | M    |  30000.00 | 111222333 |   1 |
+-------+-------+-------+-----------+------------+-------------+------+-----------+-----------+-----+
5 rows in set (0.11 sec)
```

Now all these tables should be related. So we have to create foreign keys.

Using **ALTER** command:

Example: Alter table ‘dependent’ add foreign key(‘essn’) references employee(‘ssn’);

Similarly, we add foreign keys and make every table related to each other.

Now, we can use UPDATE, DROP, DELETE commands as well.

```
mysql> update employee
    -> set salary='27000'
    -> where ssn='112233245';
Query OK, 1 row affected (0.23 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> delete from works
    ->
    -> ;
ERROR 1146 (42S02): Table 'project.works' doesn't exist
mysql> delete fro
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that cor
responds to your MySQL server version for the right syntax to use near '' at line 1

mysql> delete from works_on
    -> where Essn = '112233556' and pno = '1';
Query OK, 1 row affected (0.22 sec)

mysql>
```

RELATIONAL ALGEBRA AND RELATIONAL CALCULUS:

We can also achieve relational algebra using **select, from, where** clauses.

Example: select fname,lname from employee where dno = ‘4’;

```
mysql> select fname,lname
    -> from employee
    -> where dno='1';
+-------+-------+
| fname | lname |
+-------+-------+
| JC1   | IP1   |
+-------+-------+
1 row in set (0.00 sec)

mysql>
```

```
mysql> select pnumber,dnumber,lname,adress,bdate
    -> from project,department,employee
    -> where Dnum=Dnumber and mgr_ssn=ssn and ploc='JCB';
Empty set (0.11 sec)

mysql>
```

Query: select employees who has immediate supervisors.

```
mysql> select E.fname,E.lname,S.fname,S.lname
    -> from employee as E,employee as S
    -> where E.super_ssn=S.ssn;
+-------+-------+-------+-------+
| fname | lname | fname | lname |
+-------+-------+-------+-------+
| JC2   | IP2   | JC1   | IP1   |
+-------+-------+-------+-------+
1 row in set (0.00 sec)

mysql>
```

Query: project only salary from employee.

```
mysql> select distinct salary
    -> from employee;
+------------+
| salary     |
+------------+
|  27000.00 |
|  50000.00 |
| 508001.00 |
|  30000.00 |
+------------+
4 rows in set (0.09 sec)

mysql>
```

CROSS PRODUCT of employee and department:

```
mysql> select ssn,dname
    -> from employee,department;
+-----------+----------------+
| ssn       | dname          |
+-----------+----------------+
| 112233245 | administration |
| 112233254 | administration |
| 112233445 | administration |
| 116233245 | administration |
| 123456789 | administration |
| 112233245 | trainer        |
| 112233254 | trainer        |
| 112233445 | trainer        |
| 116233245 | trainer        |
| 123456789 | trainer        |
| 112233245 | intern         |
| 112233254 | intern         |
| 112233445 | intern         |
| 116233245 | intern         |
| 123456789 | intern         |
| 112233245 | managing       |
| 112233254 | managing       |
| 112233445 | managing       |
| 116233245 | managing       |
| 123456789 | managing       |
| 112233245 | Research       |
| 112233254 | Research       |
| 112233445 | Research       |
| 116233245 | Research       |
| 123456789 | Research       |
+-----------+----------------+
25 rows in set (0.00 sec)

mysql>
```

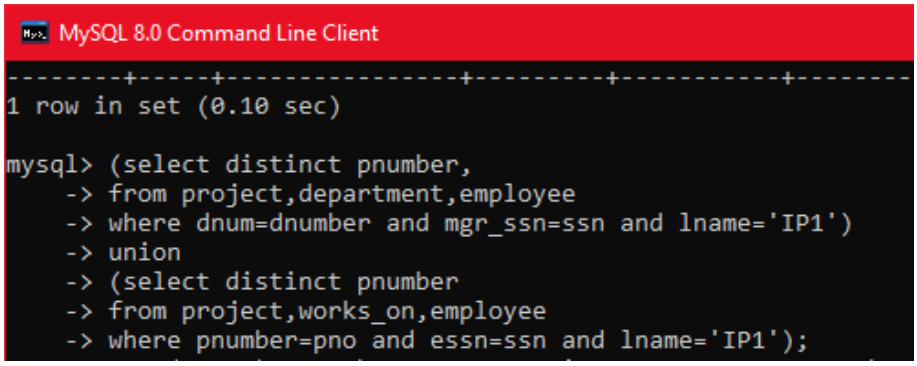
Query: select employees whose department name is administration and department number are same.

```
mysql> select * from employee,department
   -> where dname='administration'and dno=dnumber;
+-------+-------+-------+----------+------------+----------+-----+---------+-----------+-----+--------------+---------+----------+---------------+
| fname | lname | mname | ssn      | bdate      | adress   | sex | salary  | super_ssn | dno | Dname        | Dnumber | mgr_ssn  | mgr_start_date |
+-------+-------+-------+----------+------------+----------+-----+---------+-----------+-----+--------------+---------+----------+---------------+
| JC1   | IP1   | A     | 123456789 | 1965-01-09 | 254 AP,TG | M   | 30000.00 | 111222333 |   1 | administration |       1 | 110253373 | 1989-05-22    |
+-------+-------+-------+----------+------------+----------+-----+---------+-----------+-----+--------------+---------+----------+---------------+
1 row in set (0.10 sec)

mysql>
```

NESTED QUERIES:

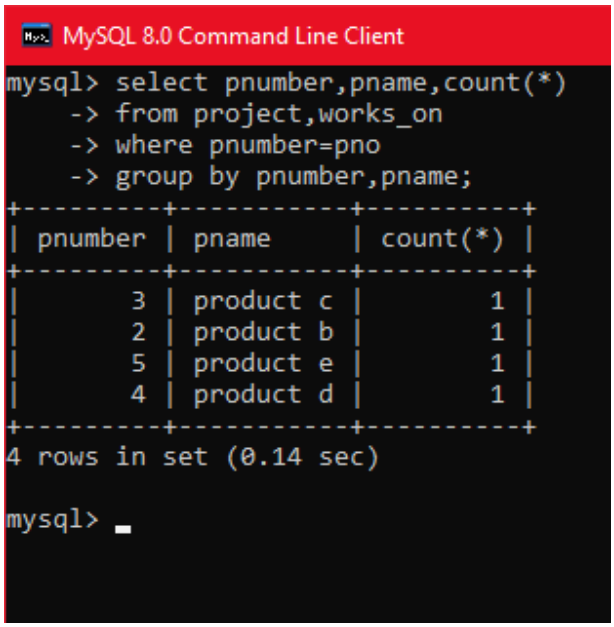
List of all project numbers of project that employee last name is ‘IP1’ either as a worker or a manager of the department that controls project.



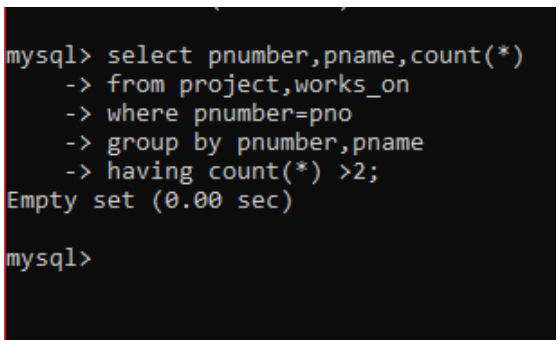
Empty set.

We also use LIKE, %, order by, group by operations also.

Example:



HAVING clause:



Empty set.

JOIN:

```
mysql> select fname,mname,lname,adress
    -> from(employee JOIN department on dno = dnumber)
    -> where dname='intern';
+-------+-------+-------+---------+
| fname | mname | lname | adress  |
+-------+-------+-------+---------+
| JC3   | c     | IP3   | H,Delhi |
+-------+-------+-------+---------+
1 row in set (0.00 sec)

mysql>
```

LEFT OUTER JOIN:

```
mysql> select E.fname as employee_name,
    -> S.lname as Supervisor_name
    -> from(employee as E left outer join employee as S
    -> on E.ssn = S.super_ssn);
+---------------+-----------------+
| employee_name | Supervisor_name |
+---------------+-----------------+
| JC1           | IP2             |
| JC4           | NULL            |
| JC3           | NULL            |
| JC2           | NULL            |
| JC5           | NULL            |
+---------------+-----------------+
5 rows in set (0.00 sec)

mysql>
```

VIEWS:

```
mysql> create view works
    -> as select fname,lname,pname,hours
    -> from employee,project,works_on
    -> where ssn = essn and pno = pnumber;
Query OK, 0 rows affected (0.86 sec)

mysql> seelect * from works;
ERROR 1064 (42000): You have an error in your
responds to your MySQL server version for the
rom works' at line 1
mysql> select * from works;
+-------+-------+-----------+-------+
| fname | lname | pname     | hours |
+-------+-------+-----------+-------+
| JC1   | IP1   | product b |   9.0 |
+-------+-------+-----------+-------+
1 row in set (0.11 sec)

mysql>
```

TRIGGERS:

It can be invoked either before or after the data is been changed by insert, update, delete commands.

- Before insert
- After insert
- Before update
- After update
- Before delete
- After delete

-> create trigger sal_info

-> before insert or update of salary,supervision_ssn on employee

-> for each row

-> when(new.salary > (select salary from employee

-> where ssn = new.supervision_ssn))

-> inform_supervision(new.supervision_ssn,new.ssn);

TIME STAMP:

```
mysql> select timestamp("2017-09-09");
+--------------------------+
| timestamp("2017-09-09") |
+--------------------------+
| 2017-09-09 00:00:00     |
+--------------------------+
1 row in set (0.11 sec)

mysql>
```

TECHNICAL FEATURES AND ELEMENTTS

Employee management system can be done efficiently on database rather than file management system.

- Ease of management
- Robust transactional support
- Comprehensive application development
- Low time cost of ownership
- High performance
- Secure data protection
- Scalability and flexibility

EXISTING STATE OF THE ART

Employee management system has been developed by many sources and also available in java where My SQL is used as backend and advanced java swig concepts has been used in the front end

To better illustrate the existing state of the art, I designed a table which consists all the information.

S No.	Existing State of the art	Drawbacks in existing state of the art	Overcome
1	Employee management system in java using swings	Has dependency on java and front-end development	To create a database, we don't require any dependencies
2	Employee management system in web development using HTML, JS,CSS	Has dependency on these we development software tools	In order to use a database in an organization additional web development tools are not required

BLOCK DIAGRAM AND FLOW CHART

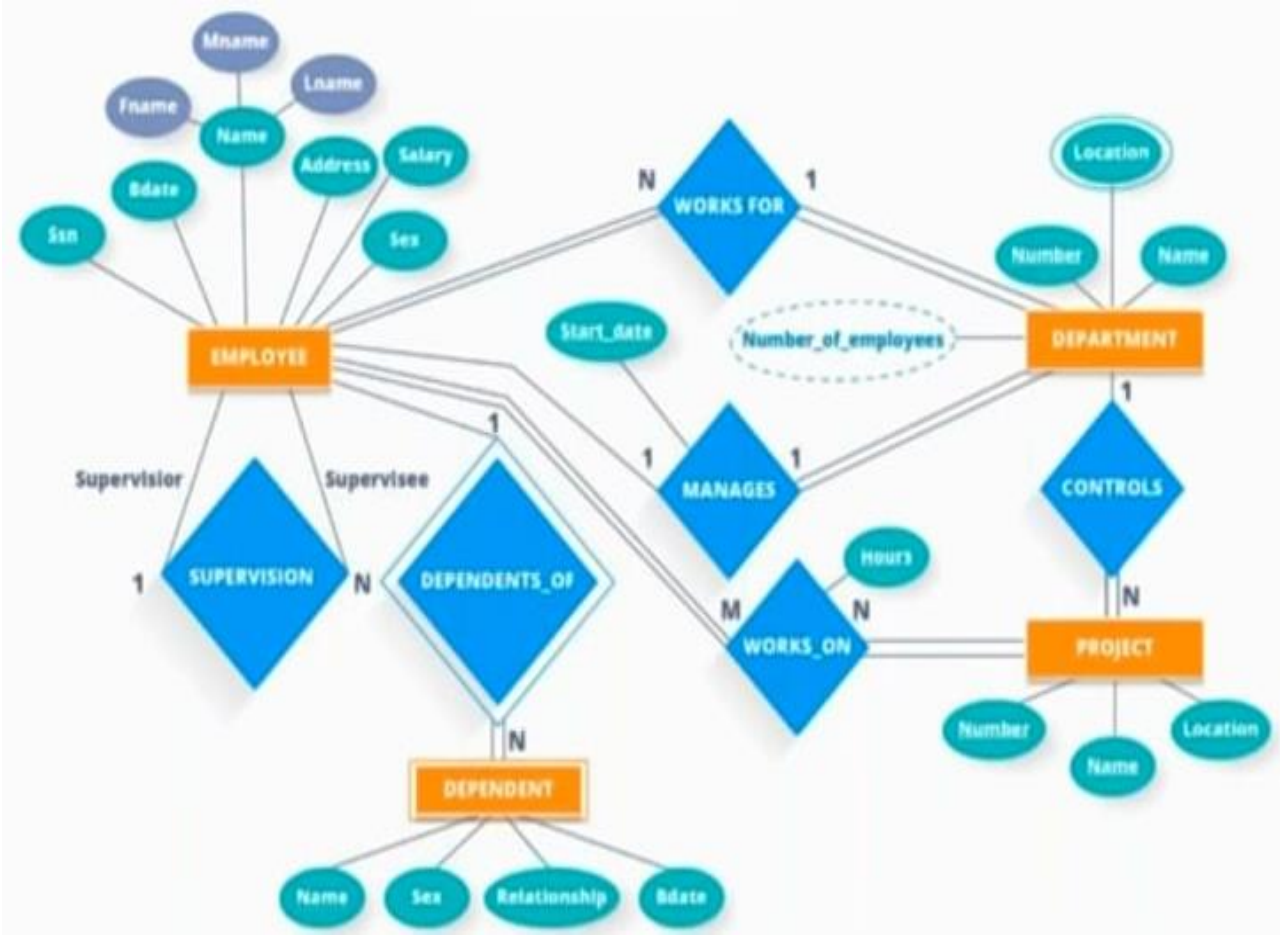


Figure 1 ER diagram for Employee management system

DISTINGUISHABLE FEATURES

The other projects have been created in Unity, Java or C++ but those are dependent on the technologies. My project uses the My SQL database by Oracle to create a database that means that means that anyone having a My SQL Server installed can run the project. Now days since every device has browser and an internet anyone can access the server.

ALTERNATE SOLUTION

There may be many alternate implementations of this project. People may use technologies in which they are comfortable and can implement the project using technologies such as REACT, UNITY, Angular. Even I initially thought of doing this project on java and then connecting it to My SQL server. But it is quite difficult to implement **views, triggers, joins**, etc. using JAVA.

STATUS AND IDEA OF THE PROJECT

The project has been created, tested, implemented on My SQL server database. There is currently no other system that uses this technology to implement this project. I found one website which uses JAVA using SWING concepts to do the same. But it doesn't have functionalities like my project has.

I first came across this idea while using our LMS (lms.bmu.edu.in). I thought that, it would be great if students also get an opportunity to design any management system, as that would help them in academics.

LIST OF COMPONENTS USED

- My SQL command line client
- My SQL server
- My SQL work bench

CONCLUSION

Employee Management system is a major concept in DBMS. Since this project has been designed exclusively as a project, certain complexities that do faced by any real-life manual problem like total no. of employee, address redundancy etc. are considered in this project. But enhancement to the project can easily be made without changing the current design and programming structure.

This project gave also improved the understanding of Database concepts and to implement various functionalities of DBMS concepts.

I thank our faculty mentor for giving such opportunity for making us learn those concepts practically in the form of project work.

REFERENCES:

- www.java.sun.com
- www.smartdraw.com
- https://www.oracle.com/a/ocom/docs/mysql/mysql-database-service-infographic.pdf
- https://www.oracle.com/mysql/
- https://dev.mysql.com/doc/refman/8.0/en/mysql-commands.html