

PROJECT REPORT

EMPLOYEE PAYROLL MANAGEMENT SYSTEM



Department: (18CSC303J)
Database Management System

Team Members:

RA1811031010053 – Kanwar Devaansh Singh

RA1811031010030 – Rana Karn Singh

RA1811031010029 – Vandit Shankar Sharma

Table of Contents

1. Abstract
2. Requirements Specification
3. E-R Diagram
4. Table Design with Integrity Constraints
5. DDL Queries
6. DML Queries
7. Sub queries
8. SQL Functions

9. Views

10. Conclusion

1. Abstract

1. Employee Information

Employee data is very essential in order to maintain a proper record of the employees and their personal information for various purposes like contacting them for inviting for certain summit, feedback of the company from the employee data

2. Maintaining Salary

Very important to keep this data which will help not only the managers and the HR to keep a track of the employee salaries but also help the company or its board to analyze what amount they are spending on a particular employee of a particular company

3. Work Location

It is very much important for an organization small or big to have a record of all the work locations they operate from to see how they can develop in that particular region and also increase the hiring in that region so that the organization can increase their Market Outreach in that area.

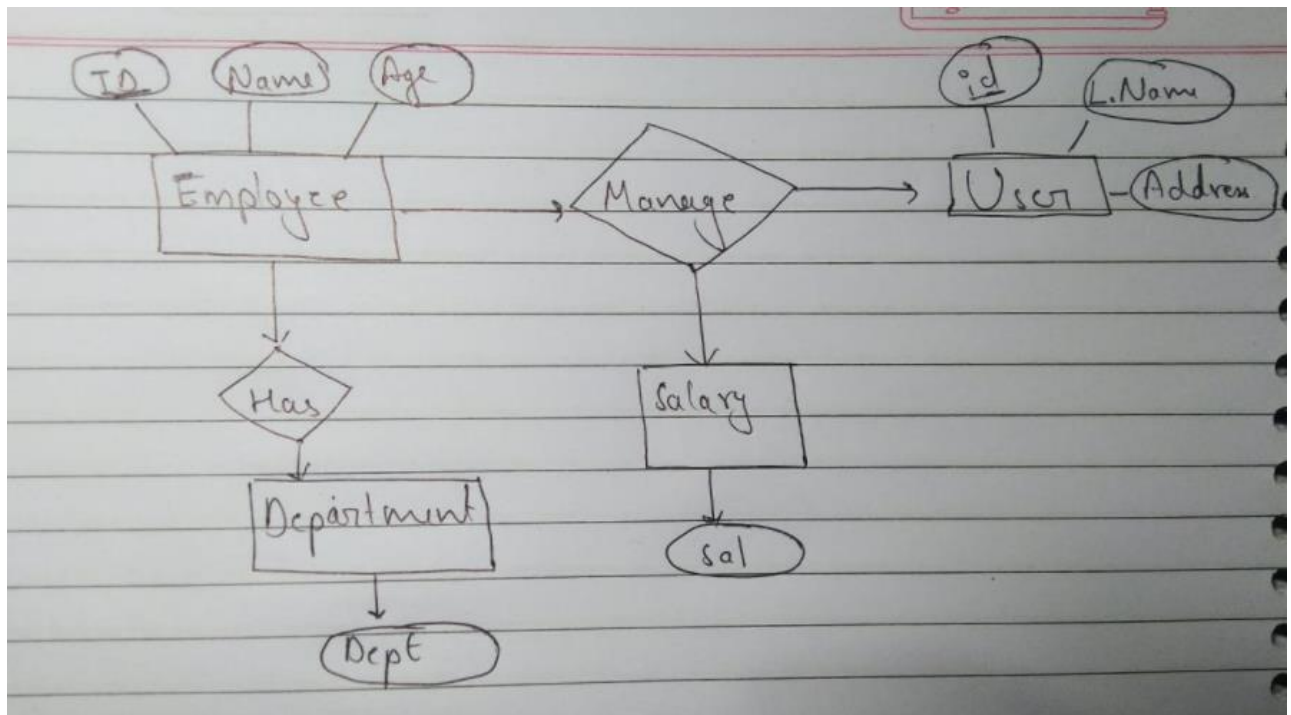
4. Projects

In order to be successful company should be involved in various projects, so they also need to maintain the record of the salaries each employee is being paid for a particular type of project he/she is working on

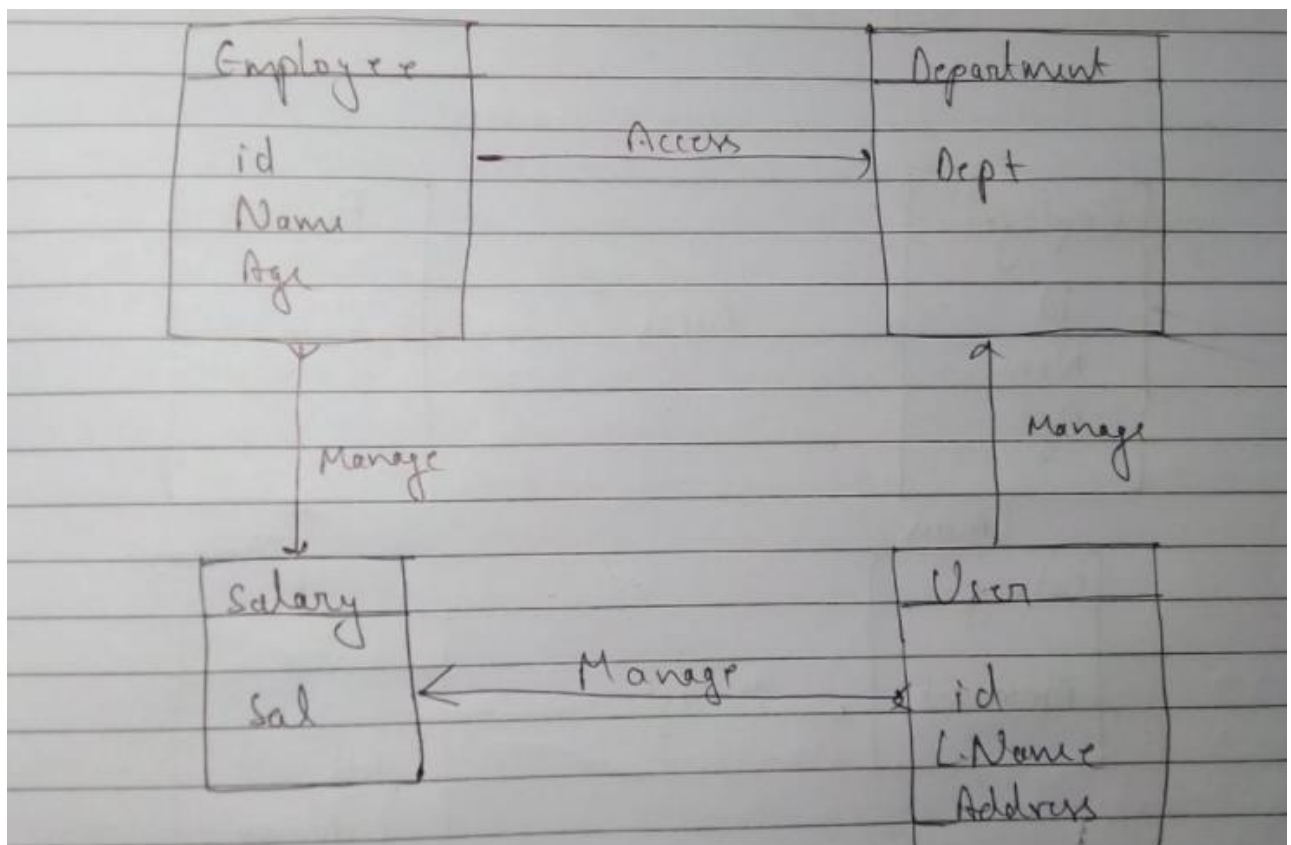
2. Requirements Specification

- Database for MNC is very essential to keep track of Employee personal as well as official data.
- Maintaining salary, hierarchy, contact details is essential part of organization.
- Different Projects running in company and under which manager the project is running is also the thing which we need to keep track of.
- Maintaining coordination among the employees can be achieved using proper database of whole organization.

3. E-R Diagram



4. Table Design with Integrity Constraints



5. DDL Queries

1. Create table

```
SQL> create table Employee(ID integer, Name varchar(20), Dept varchar(20), Age integer, Salary integer);  
Table created.
```

```
SQL> create table details(id int , lname varchar(15), address varchar(15));  
Table created.
```

```
SQL> select * from Employee;
```

ID	NAME	DEPT	AGE	SALARY
10	Satyam	CSE	25	20000
30	Shivam	IT	25	22000
35	Sundram	IT	18	35000
37	Hari	ECE	16	18000
40	Murli	Civil\	45	50000
43	Krishna	IT	25	50000
50	Govinda	CSE	28	40000
52	Narayan	Mechanical	17	32000

8 rows selected.

```
SQL> select* from Details;
```

ID	LNAME	ADDRESS
10	Singh	Delhi
30	Manhas	Samba
35	Chib	Gurugram
37	Singh	Ambala
40	Gupta	Chennai
43	Malhotra	Noida
50	Handa	Ghaziabad
52	Kumar	Lucknow

8 rows selected.

2. Alter table

a. Add column

```
SQL> alter table details add DOJ varchar(10);  
Table altered.
```

b. Modify column

```
SQL> alter table details modify DOJ date;  
Table altered.
```

c. Drop column

```
SQL> alter table details drop column DOJ;  
Table altered.
```

3. Drop table

```
SQL> drop table details;  
Table dropped.  
SQL> drop table employee;  
Table dropped.
```

6. DML Queries :

1. Insert :

```
SQL> Insert into Details values(10, 'Singh', 'Delhi');  
1 row created.  
  
SQL> Insert into Details values(30, 'Manhas', 'Samba');  
1 row created.  
  
SQL> Insert into Details values(35, 'Chib', 'Gurugram');  
1 row created.  
  
SQL> Insert into Details values(37, 'Singh', 'Ambala');  
1 row created.  
  
SQL> Insert into Details values(40, 'Gupta', 'Chennai');  
1 row created.  
  
SQL> Insert into Details values(43, 'Malhotra', 'Noida');  
1 row created.  
  
SQL> Insert into Details values(50, 'Handa', 'Ghaziabad');  
1 row created.  
  
SQL> Insert into Details values(52, 'Kumar', 'Lucknow');  
1 row created.
```

```
SQL> Insert into Employee values(10, 'Satyam', 'CSE', 25, 20000);  
1 row created.  
  
SQL> Insert into Employee values(30, 'Shivam', 'IT' , 25, 22000);  
1 row created.  
  
SQL> Insert into Employee values(35, 'Sundram', 'IT', 18, 35000);  
1 row created.  
  
SQL> Insert into Employee values(37, 'Hari', 'ECE', 16, 18000);  
1 row created.  
  
SQL> Insert into Employee values(40, 'Murli', 'Civil', 45, 50000);  
1 row created.  
  
SQL> Insert into Employee values(43, 'Krishna', 'IT', 25, 50000);  
1 row created.  
  
SQL> Insert into Employee values(50, 'Govinda', 'CSE', 28, 40000);  
1 row created.  
  
SQL> Insert into Employee values(52, 'Narayan', 'Mechanical', 17, 32000);  
1 row created.
```

2. Select :

1. Select Distinct

```
SQL> Select Distinct DEPT FROM Employee;
```

```
DEPT
```

```
-----
```

```
CSE
```

```
ECE
```

```
IT
```

```
Civil
```

```
Mechanical
```

1. Arithmetic in select

```
SQL> Select Age*10 from Employee;
```

```
AGE*10
```

```
-----
```

```
250
```

```
250
```

```
180
```

```
160
```

```
450
```

```
250
```

```
280
```

```
170
```

```
8 rows selected.
```

```
SQL> Select Age-5 from Employee;
```

```
AGE-5
```

```
-----
```

```
20
```

```
20
```

```
13
```

```
11
```

```
40
```

```
20
```

```
23
```

```
12
```

```
8 rows selected.
```


1. Conditional clause using <, >, =, and, or, not conditions.

```
SQL> select * from Employee WHERE Age>25;
```

ID	NAME	DEPT	AGE	SALARY
40	Murli	Civil	45	50000
50	Govinda	CSE	28	40000

```
SQL> SELECT * FROM Employee
2 WHERE Dept='IT' OR Dept='CSE';
```

ID	NAME	DEPT	AGE	SALARY
10	Satyam	CSE	25	20000
30	Shivam	IT	25	22000
35	Sundram	IT	18	35000
43	Krishna	IT	25	50000
50	Govinda	CSE	28	40000

```
SQL> SELECT * FROM Employee
2 WHERE Salary>25000 AND Dept='CSE';
```

ID	NAME	DEPT	AGE	SALARY
50	Govinda	CSE	28	40000

```
SQL> select * from Employee WHERE Age=25;
```

ID	NAME	DEPT	AGE	SALARY
10	Satyam	CSE	25	20000
30	Shivam	IT	25	22000
43	Krishna	IT	25	50000

2. Select from multiple relations

```
SQL> select Employee.name, Employee.Dept, Details.Lname, Details.Address from Employee, Details
2 where Employee.ID=Details.ID;
```

NAME	DEPT	LNAME	ADDRESS
Satyam	CSE	Singh	Delhi
Shivam	IT	Manhas	Samba
Sundram	IT	Chib	Gurugram
Hari	ECE	Singh	Ambala
Murli	Civil	Gupta	Chennai
Krishna	IT	Malhotra	Noida
Govinda	CSE	Handa	Ghaziabad
Narayan	Mechanical	Kumar	Lucknow

8 rows selected.

1. Rename

```
SQL> select age+5 as incr_age from Employee;
```

INCR_AGE
30
30
23
21
50
30
33
22

8 rows selected.

```
SQL> select salary+5000 as incr_salary from Employee;
```

INCR_SALARY
25000
27000
40000
23000
55000
55000
45000
37000

8 rows selected.

3. Update :

```
SQL> UPDATE Employee  
2 SET Dept='CIVIL'  
3 where Dept='Civil';
```

1 row updated.

```
SQL> UPDATE Employee
  2  SET Salary=22500
  3  where Salary > 20000 AND Salary < 25000;

1 row updated.
```

```
SQL> UPDATE Employee
  2  SET Dept='MECH'
  3  where Dept='Mechanical';

1 row updated.
```

4. Delete :

```
SQL> Insert into Employee values(77, 'Puri','CSE',28,40000);

1 row created.

SQL> Insert into Employee values(88, 'Nipun', 'IT',17,32000);

1 row created.
```

```
SQL> Delete from Employee Where Name='Nipun';

1 row deleted.
```

5. Order by clause :

```
SQL> SELECT * FROM Employee
  2  ORDER BY ID ASC;
```

ID	NAME	DEPT	AGE	SALARY
10	Satyam	CSE	25	20000
30	Shivam	IT	25	22500
35	Sundram	IT	18	35000
37	Hari	ECE	16	18000
40	Murli	CIVIL	45	50000
43	Krishna	IT	25	50000
50	Govinda	CSE	28	40000
52	Narayan	MECH	17	32000

```
8 rows selected.
```

```
SQL> SELECT * FROM Employee
2 ORDER BY Salary DESC;
```

ID	NAME	DEPT	AGE	SALARY
40	Murli	CIVIL	45	50000
43	Krishna	IT	25	50000
50	Govinda	CSE	28	40000
35	Sundram	IT	18	35000
52	Narayan	MECH	17	32000
30	Shivam	IT	25	22500
10	Satyam	CSE	25	20000
37	Hari	ECE	16	18000

8 rows selected.

6. Aggregate functions :

1. Sum

```
SQL> select SUM(Salary) from Employee;

SUM(SALARY)
-----
267500
```

1. Count

```
SQL> SELECT COUNT(Dept) from Employee;

COUNT(DEPT)
-----
8
```

1. Max

```
SQL> Select MAX(Age) from Employee;

MAX(AGE)
-----
45
```

7. Set operations :

Union:

```
SQL> Select ID,Name,Dept From Employee
2 Union
3 Select ID,Lname,Address From Details;
```

ID	NAME	DEPT
10	Satyam	CSE
10	Singh	Delhi
30	Manhas	Samba
30	Shivam	IT
35	Chib	Gurugram
35	Sundram	IT
37	Hari	ECE
37	Singh	Ambala
40	Gupta	Chennai
40	Murli	Civil
43	Krishna	IT
43	Malhotra	Noida
50	Govinda	CSE
50	Handa	Ghaziabad
52	Kumar	Lucknow
52	Narayan	Mechanical

16 rows selected.

Intersect:

```
SQL> Select id from employee where salary<40000
2 intersect
3 select id from details;
```

ID
10
30
35
37
52

Except:

```
SQL> select * from employee where dept='CSE'
  2  except
  3  select* from details where Lname='Singh';
except
*
ERROR at line 2:
ORA-00933: SQL command not properly ended
```

In:

```
SQL> select name from employee where id in(select id from Details where Lname='Singh');

NAME
-----
Satyam
Hari
```

Not In:

```
SQL> select name from employee where id not in(select id from Details where Lname='Singh');

NAME
-----
Govinda
Narayan
Krishna
Murli
Shivam
Sundram

6 rows selected.
```

8. String operations_:

```
SQL> SELECT * from Employee
  2  WHERE Name LIKE 'M%';
```

ID	NAME	DEPT	AGE	SALARY
40	Murli	CIVIL	45	50000

```
SQL> SELECT * from Employee
  2  WHERE Name LIKE '_a%';
```

ID	NAME	DEPT	AGE	SALARY
10	Satyam	CSE	25	20000
37	Hari	ECE	16	18000
52	Narayan	MECH	17	32000

9. Natural join :


```
SQL> SELECT ID,Name,Dept FROM Employee NATURAL JOIN Details;
```

ID	NAME	DEPT
10	Satyam	CSE
30	Shivam	IT
35	Sundram	IT
37	Hari	ECE
40	Murli	Civil
43	Krishna	IT
50	Govinda	CSE
52	Narayan	Mechanical

8 rows selected.

10. Group by :

```
SQL> Select COUNT(Name) from Employee Group By Dept;
```

COUNT(NAME)
2
1
3
1
1

```
SQL> Select * from Employee;
```

ID	NAME	DEPT	AGE	SALARY
10	Satyam	CSE	25	20000
30	Shivam	IT	25	22000
35	Sundram	IT	18	35000
37	Hari	ECE	16	18000
40	Murli	Civil	45	50000
43	Krishna	IT	25	50000
50	Govinda	CSE	28	40000
52	Narayan	Mechanical	17	32000

8 rows selected.

11. Having clause :

```
SQL> SELECT COUNT(ID), Dept FROM Employee GROUP BY Dept HAVING COUNT(Dept) > 2;
```

COUNT(ID)	DEPT
3	IT

7. Subqueries :

(At Least 10 queries)

```
SQL> Select name from employee where(dept='CSE');
```

```
NAME
```

```
-----
```

```
Satyam
```

```
Govinda
```

```
SQL> select name from employee where id in(select id from details where id>20);
```

```
NAME
```

```
-----
```

```
Shivam
```

```
Sundram
```

```
Hari
```

```
Murli
```

```
Krishna
```

```
Govinda
```

```
Narayan
```

```
7 rows selected.
```

```
SQL> select age from employee where salary>25000 or id in(select id from details where lname='Singh');
```

```
AGE
```

```
-----
```

```
25
```

```
18
```

```
16
```

```
45
```

```
25
```

```
28
```

```
17
```

```
7 rows selected.
```

```
SQL> SELECT COUNT(Name), Dept
```

```
2 FROM Employee
```

```
3 GROUP BY Dept
```

```
4 HAVING COUNT(ID) > 2;
```

```
COUNT(NAME) DEPT
```

```
-----
```

```
3 IT
```



```
SQL> select name from employee where id in(select id from details having address='Noida');
```

NAME

Satyam
Shivam
Sundram
Hari
Murli
Krishna
Govinda
Narayan

8 rows selected.

```
SQL> select lname from details where id in(select id from employee where salary>30000);
```

LNAME

Chib
Gupta
Malhotra
Handa
Kumar

```
SQL> select * from details where id in(select id from employee where name='Murli');
```

ID	LNAME	ADDRESS
40	Gupta	Chennai

```
SQL> select * from employee where id in (select id from details where id<40);
```

ID	NAME	DEPT	AGE	SALARY
10	Satyam	CSE	25	20000
30	Shivam	IT	25	22000
35	Sundram	IT	18	35000
37	Hari	ECE	16	18000

```
SQL> select address from details where id in(select id from employee where salary>40000);
```

ADDRESS

Chennai
Noida

```
SQL> select * from employee where id in (select id from details where lname like 'S%');
```

ID	NAME	DEPT	AGE	SALARY
10	Satyam	CSE	25	20000
37	Hari	ECE	16	18000

7. SQL Functions

Write queries related to your project using these SQL functions:

i) Concat

```
SQL> Select Name, CONCAT(Dept,ID) from Employee;
```

NAME	CONCAT(DEPT, ID)
Satyam	CSE10
Shivam	IT30
Sundram	IT35
Hari	ECE37
Murli	Civil40
Krishna	IT43
Govinda	CSE50
Narayan	Mechanical52

ii) Lower

```
SQL> select Name, LOWER(Dept) from employee;
```

NAME	LOWER(DEPT)
Satyam	cse
Shivam	it
Sundram	it
Hari	ece
Murli	civil
Krishna	it
Govinda	cse
Narayan	mechanical

8 rows selected.

iii) Reverse

```
SQL> SELECT ID, REVERSE(Name) from Employee;
```

ID	REVERSE(NAME)
10	maytaS
30	mavihS
35	mardnuS
37	iraH
40	ilruM
43	anhsirK
50	adnivoG
52	nayaraN

```
8 rows selected.
```

iv) Substring

```
SQL> SELECT ID, SUBSTR(Dept, 1, 3) from Employee;

      ID SUBSTR(DEPT,
-----
      10 CSE
      30 IT
      35 IT
      37 ECE
      40 Civ
      43 IT
      50 CSE
      52 Mec

8 rows selected.
```

v) Abs

```
SQL> select abs(-10) from dual;

ABS(-10)
-----
      10
```

vi) Ceiling/floor

```
SQL> Select floor(99.9) from dual;

FLOOR(99.9)
-----
      99
```

```
SQL> select ceil(99.9) from dual;

CEIL(99.9)
-----
     100
```

vii) Current_timestamp

```
CURRENT_TIMESTAMP
-----
2021-05-21 06:49:49.149000 PM +05:30
2021-05-21 06:49:49.149000 PM +05:30
2021-05-21 06:49:49.149000 PM +05:30
2021-05-21 06:49:49.149000 PM +05:30
2021-05-21 06:49:49.149000 PM +05:30
```

viii) Dateadd

```
SQL> select ADD_MONTHS('26-JAN-2021', 2) from dual;

ADD_MONTH
-----
26-MAR-21
```

ix) Datediff

```
DATE-I
-----
2021-05-11
2021-05-11
2021-05-11
2021-05-11
2021-05-11
```

x) Getdate

```
SQL> select SYSDATE from dual;
```

```
SYSDATE
```

```
-----
```

```
24-MAY-21
```

8. Views

Create Views for your project database.

- a) Views to check Employees who all have salary greater than 30000.

```
SQL> create view high_salary as
  2 select ID, Name, Dept
  3 From Employee
  4 where Salary > 30000;
```

View created.

```
SQL> select * from high_salary;
```

ID	NAME	DEPT
35	Sundram	IT
40	Murli	Civil
43	Krishna	IT
50	Govinda	CSE
52	Narayan	Mechanical

- b) Views to see Employees Details which have Department IT.

```
SQL> create view SDE as
  2 select ID, Name, Age
  3 From Employee
  4 where Dept='IT';
```

View created.

```
SQL> select * from SDE;
```

ID	NAME	AGE
30	Shivam	25
35	Sundram	18
43	Krishna	25

c) Views to check Eligibility of Employees based on Age.

```
SQL> create view Eligibility as
  2  select ID, Name
  3  From Employee
  4  where Age > 18;
```

View created.

```
SQL> select * from Eligibility;
```

ID	NAME
10	Satyam
30	Shivam
40	Murli
43	Krishna
50	Govinda

9. Conclusion

During our database management course, we have learned about the basics of database design. This project gave us the opportunity to try our new skills in practice

While doing this project we also gained a deeper understanding on database design and how it can be implemented in real life situations.

In conclusion, a database is a far more efficient mechanism to store and organize data than spreadsheets; it allows for a centralized facility that can easily be modified and quickly shared among multiple users.

* * * * * End * * * * *