



Bahria University
Discovering Knowledge

DATABASE MANAGEMENT System

FINAL PROJECT

“HUMAN RESOURCE MANAGEMENT SYSTEM”

GROUP MEMBERS

Name	Enrollment No.
Habib Ur Rehman	02-134192-002
Rehana Riaz	02-134192-014
Muhammad Saad Iqbal	02-134191-018

SUBMITTED to:- SIR

MALIK M.Ali

Lab InCHARGE:-

MISS SABA IMTIAZ

TABLE OF CONTENTS:-

TABLE OF CONTENTS:.....	2
ACKNOWLEDGMENT:	5
ABSTRACT:.....	6
PROJECT IDEA:	7
PROBLEM STATEMENT:.....	8
BENEFITS OF USING DATABASE	8
PHYSICAL STRUCTURE & DETAILS	9
TABLES	9

ACKNOWLEDGEMENT:

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show our greatest appreciation to our course teacher **sir Malik M Ali** and our lab engineer **miss Saba Imtiaz** for their tremendous support and help. We feel motivated and encouraged every time without their encouragement and guidance this project would not have materialized. The guidance and support received from all the members who contributed to this project, was vital for the success of the project.

ABSTRACT:

A project titled “**Human Resource Management System**” is completed successfully, which was implemented in Oracle 12C.

The concepts of database were taught, which is implemented in the project. Our project “Human Resource Management System” is an online application. The main aim of the project is to develop a Human Resource management module for the Intranet Automation of HR Software. A Human Resource Management System (HRMS) refers to the systems and processes at the intersection between human resource management (HRM) and information technology (IT). It merges HRM as a discipline and in particular it's basic HR activities and processes with the information technology field. Which makes this software application both rigid and flexible.

This project mainly aims to maintain the database of the employees working in an organization their salary, payrolls with a facility to edit and update it. The HRMS projects facilitates user as per their hierarchy and grants privilege to them accordingly. The complete project is designed using Oracle 12C

INTRODUCTION:

Human Resource Management Systems is the word increasingly used refers to policies, practices and procedures that are related to the management of the people within an organization. In simple words Human Resource Management Systems (HRMS) is software that allows or connects human resources and information technology using processes or systems. The human resources management mainly deals with management and staffing effective work force for the organization. The main functions of HRMS include HR management, managing employee's payroll, training and recruitment, organization change management, dealing with insurance and policy, observing performance of employee and manager self-services. HRMS is must and required system to perform main HR management responsibilities like mainly in reporting, pay roll management,

recruiting and learning management. Above functions and responsibilities are included in wide selection of tools with many detailed answers for many business requirements or needs on other side when it comes to human resource management and it is best suited for almost all of the companies. The complication of HRM systems is actually depends on the how big the company is, employee numbers, roles and responsibilities the HRM should handle.

PROJECT IDEA:

The idea of this project is to maintain an HR management system of a particular company with proper data of how an HR manager manages the information of employees and other professional belongings of employees for company usage and requirement.

PROBLEM STATEMENT:

Human resource management (HRM) is adopted by many companies because of its benefits. But at the same time, various challenges and issues may emerge in front of managers of human resource department while performing their duties.

Any capable HR manager would work on these issues and challenges to prevent the organizational activities from being obstructed. But they must first identify these issues. Such issues and challenges are described here.

Recruiting new staff

Companies sometimes need to recruit new talent for various reasons such as an increase in project scope, operations. While recruiting, HRM faces major challenges i.e. selecting the best candidate and making the hired candidate familiar with the environment and culture.

HRM has to select such candidates which are not only technically expert but socially too because the company cannot receive expected output from employees who are unable to socialize. This affects the productivity of the company.

Productivity

Maximizing profit and minimizing cost is the essence of productivity. Higher the productivity level, more successful will be the company. HRM should always focus on maintaining high productivity level.

Despite HRM's continuous effort, the company might sometimes get an unsatisfactory result. In such case, it should thoroughly analyze the situation and make a proper conclusion i.e. whether it is the result of inefficient employee or inadequate resources. If an inefficient employee is the case, HRM should look forward to train employee, or even recruiting new staffs while it should find alternative resources for sufficient input if inadequate resources are the case.

DISCIPLINE:

Discipline is one of the important issues that HRM needs to handle at present days. Lack of discipline causes various problems which ultimately affect the productivity of the company.

For an instance, when discipline is not maintained, employees neglect their responsibilities and duties. They may procrastinate their tasks and may misbehave with co-workers, leading to a conflict that consumes time as well as energy to resolve. The ultimate result of these activities is decreased productivity of the company.

BENEFITS OF USING DATABASE:

Following are the benefits of using database:

- Reduced data redundancy
- Reduced updating errors and increased consistency
- Greater data integrity and independence from applications programs
- Improved data access to users through use of host and query languages
- Improved data security
- Reduced data entry, storage, and retrieval costs
- Facilitated development of new applications program

PHYSICAL STRUCTURE & DETAILS:

TABLES:

Employee table: shows all information of employee

```
SQL> desc employee;
Name                               Null?      Type
-----
EMP_ID                             NOT NULL   NUMBER(6)
EMP_NAME                           VARCHA2(25)
EMP_ADDRESS                         VARCHA2(45)
EMP_NUMBER                         VARCHA2(20)
EMP_EMAIL                          VARCHA2(20)
SQL>
```

Admin table: shows all the admins of HR management system and they can access the whole

management system.

```
SQL> desc admin;
Name                                     Null?      Type
-----
ADMIN_ID                               NOT NULL   NUMBER(6)
ADMIN_NAME                             VCHAR2(20)
ADMIN_EMAIL                             VCHAR2(20)
ADMIN_PASSWORD                           VCHAR2(15)
```

Department table: department table shows all the department present in company with name of department and location.

```
SQL> desc department;
Name                                     Null?      Type
-----
DEPART_ID                               NOT NULL   NUMBER(4)
DEPART_NAME                             VCHAR2(30)
EMP_ID                                  NUMBER(6)
LOCATION_ID                               NUMBER(4)
```

Location table: location table will show the location and complete information that where it is located.

```
SQL> desc location;
Name                                     Null?      Type
-----
LOCATION_ID                               NOT NULL   NUMBER(4)
STREET_ADDRESS                           VCHAR2(40)
POSTAL_CODE                             VCHAR2(12)
CITY                                     NOT NULL   VCHAR2(30)
STATE_PROVINCE                           VCHAR2(25)
```

Jobs table: this table will show the number of jobs present there for employees and position of employees

```
SQL> desc jobs;
Name                               Null?      Type
-----
JOB_ID                             NOT NULL   VARCHAR2(10)
JOB_TITLE                           NOT NULL   VARCHAR2(35)
MIN_SALARY                          NUMBER(6)
MAX_SALARY                          NUMBER(6)
SQL>
```

Job_history table : this table will show the job history of every employee including their minimum and maximum salary

```
SQL> desc job_history;
Name                               Null?      Type
-----
EMP_ID                             NOT NULL   NUMBER(6)
START_DATE                         NOT NULL   DATE
END_DATE                           NOT NULL   DATE
JOB_ID                             NOT NULL   VARCHAR2(10)
DEPART_ID                          NUMBER(4)
```

Training table: this table will show training id , its title and its description along with its department id

```
SQL> desc training;
Name                                     Null?      Type
-----
TRAINING_ID                             NOT NULL   NUMBER(4)
TRAINING_TITLE                           NOT NULL   VARCHAR2(45)
TRAINING_DESCRIPTION                      VARCHAR2(60)
DEPART_ID                                NUMBER(4)
```

User table: this table only consist of username and password which shows the those users who have accessibility to this management system.

```
SQL> desc users;
Name                                     Null?      Type
-----
USER_ID                                 NOT NULL   NUMBER(6)
USER_PASSWORD                           VARCHAR2(15)
```

Salary table: this table will show the salary bonus and loans given to every employee.

```
SQL> desc salary;
Name                                     Null?      Type
-----
SALARY_ID                               NOT NULL   NUMBER(4)
SALARY                                  NUMBER(10)
BONUS                                   NUMBER(10)
LOAN                                    NUMBER(10)
LAST_UPDATED                            DATE
EMP_ID                                  NUMBER(6)
```


Country table: this table will show all the details about country in which the HR office is located.

```
SQL> desc country;
Name                                     Null?      Type
-----
COUNTRY_ID                             NOT NULL   CHAR(2)
COUNTRY_NAME                           VARCHAR2(40)
REGION_ID                               NUMBER
```

Region table: this will show the region where HR office is located.

```
SQL> desc region;
Name                                     Null?      Type
-----
REGION_ID                             NOT NULL   NUMBER
REGION_NAME                           VARCHAR2(25)
```

Attendance table: this table will show the attendance record of all the employees through employee id.

```
SQL> desc attendance;
Name                                     Null?      Type
-----
ATTENDANCE_ID                         NOT NULL   NUMBER(4)
ATTENDANCE_EMPLOYEE                   CHAR(4)
ATTENDANCE_DATE                       DATE
```

Following are the implementations:

Views

Admin view:

```
create view v_admin as
  select a.admin_id,a.admin_name,a.admin_password
  from admin a;
```

v_employee10:

```
create view v_employee10 as
  select e.emp_id,e.emp_name,e.emp_number,d.depart_id
  from employee e, department d
  where depart_id=10;
```

v_employee20:

```
create view v_employee20
  as select e.emp_id,e.emp_name,e.emp_number,d.depart_id
  from employee e, department d
  where depart_id=20;
```

v_department:

```
create view v_department as
  select e.emp_id,d.depart_id,e.emp_name,d.depart_name from
  employee e, department d;
```

v_location:

```
create view v_location as
  select location_id,city,state_province
  from location;
```

v_jobs:

```
create view v_jobs as
  select job_id,job_title,min_salary,max_salary
  from jobs;
```

v_jobhistory:

```
create view v_jobhistory as
  select e.emp_id,jh.start_date,jh.end_date,j.job_id,d.depart_id
  from employee e, jobs j, job_history jh,department d;
```

v_training:

```
create view v_training as
  select t.training_id,t.training_title,d.depart_id
  from training t,department d;
```

v_users:

```
create view v_users as
  select user_id, user_password
  from users;
```

v_salary:

```
create view v_salary as
  select e.emp_id,s.salary,s.bonus,s.loan
```

```
from employee e,salary s;
```

v_country:

```
create view v_country as
select c.country_id,c.country_name,r.region_id
from country c, region r;
```

v_region:

```
create view v_region as
select region_id,region_name
from region;
```

v_attendance:

```
create view v_attendance as
select e.emp_id, a.attendance_id,a.attendance_employee,a.attendance_date
from employee e, attendance a;
```

Procedures:

insert_employee:

```
create or replace procedure insert_employee(
e_id employee.emp_id%type,
e_name employee.emp_name%type,
e_address employee.emp_address%type,
e_number employee.emp_number%type,
e_email employee.emp_email%type)
is
begin
insert into employee(emp_id,emp_name,emp_address,emp_number,emp_email)
values(e_id,e_name,e_address,e_number,e_email);
commit;
end;
/
```

Insert_admin:

```
create or replace procedure insert_admin(
a_id admin.admin_id%type,
a_name admin.admin_name%type,
a_email admin.admin_email%type,
a_password admin.admin_password%type)
is
begin
insert into admin (admin_id,admin_name,admin_email,admin_password)
values(a_id,a_name,a_email,a_password);
commit;
end;
/
```

Insert_department:

```
create or replace procedure insert_department(
```

```

d_id department.depart_id%type,
d_name department.depart_name%type,
e_id department.emp_id%type,
l_id department.location_id%type)
is
begin
insert into department(depart_id,depart_name,emp_id,location_id)
values(d_id,d_name,e_id,l_id);
commit;
end;
/

```

Insert location:

```

create or replace procedure insert_location(
l_id location.location_id%type,
s_address location.street_address%type,
p_code location.postal_code%type,
city location.city%type,
state_province location.state_province%type)
is
begin
insert into location(location_id,street_address,postal_code,city,state_province)
values(l_id,s_address,p_code,city,state_province);
commit;
end;
/

```

Insert job:

```

create or replace procedure insert_job(
j_id jobs.job_id%type,
j_title jobs.job_title%type,
min_sal jobs.min_salary%type,
max_sal jobs.max_salary%type)
is
begin
insert into jobs(job_id,job_title,min_salary,max_salary)
values(j_id,j_title,min_sal,max_sal);
commit;
end;
/

```

Insert job history:

```

create or replace procedure insert_job_history(
e_id job_history.emp_id%type,
start_date job_history.start_date%type,
end_date job_history.end_date%type,
j_id job_history.job_id%type,
d_id job_history.depart_id%type)
is
begin
insert into job_history(emp_id,start_date,end_date,job_id,depart_id)

```

```
values(e_id,start_date,end_date,j_id,d_id);
commit;
end;
/
```

Insert training:

```
create or replace procedure insert_training(
  t_id training.training_id%type,
  t_title training.training_title%type,
  t_desc training.training_description%type,
  d_id training.depart_id%type)
is
begin
insert into training(training_id,training_title,training_description,depart_id)
values(t_id,t_title,t_desc,d_id);
commit;
end;
/
```

Insert user:

```
create or replace procedure insert_user(
  u_id users.user_id%type,
  u_pass users.user_password%type)
is
begin
insert into users(user_id,user_password)
values(u_id,u_pass);
commit;
end;
/
```

Insert salary:

```
create or replace procedure insert_salary(
  sal_id salary.salary_id%type,
  sal salary.salary%type,
  bonus salary.bonus%type,
  loan salary.loan%type,
  last_updated salary.last_updated%type,
  e_id salary.emp_id%type)
is
begin
insert into salary(salary_id,salary,bonus,loan,last_updated,emp_id)
values(sal_id,sal,bonus,loan,last_updated,e_id);
commit;
end;
/
```

Insert country:

```
create or replace procedure insert_country(
  c_id country.country_id%type,
  c_name country.country_name%type,
  r_id country.region_id%type)
is
```

```
begin
insert into country(country_id,country_name,region_id)
values(c_id,c_name,r_id);
commit;
end;
/
```

Insert region:

```
create or replace procedure insert_region(
  r_id region.region_id%type,
  r_name region.region_name%type)
is
begin
insert into region(region_id,region_name)
values(r_id,r_name);
commit;
end;
/
```

Insert attendance:

```
create or replace procedure insert_attendance(
  atten_id attendance.attendance_id%type,
  atten_emp attendance.attendance_employee%type,
  atten_date attendance.attendance_date%type)
is
begin
insert into attendance(attendance_id,attendance_employee,attendance_date)
values(atten_id,atten_emp,atten_date);
commit;
end;
/
```

Edit employee:

```
create or replace procedure edit_employee(
  e_id in employee.emp_id%type,
  e_name in employee.emp_name%type,
  e_address in employee.emp_address%type,
  e_number in employee.emp_number%type,
  e_email in employee.emp_email%type)
is
begin
update employee set
emp_id=e_id,emp_name=e_name,emp_address=e_address,emp_email=e_email where
emp_id=e_id;
end;
/
```

Edit admin:

```
create or replace procedure edit_admin(
  a_id in admin.admin_id%type,
  a_name in admin.admin_name%type,
```

```
a_email in admin.admin_email%type,  
a_pass in admin.admin_password%type)  
is  
begin  
update admin set admin_name=a_name,admin_email=a_email,admin_password=a_pass  
where admin_id=a_id;  
end;  
/
```

Edit department:

```
create or replace procedure edit_department(  
d_id in department.depart_id%type,  
d_name in department.depart_name%type,  
e_id in department.emp_id%type,  
l_id in department.location_id%type)  
is  
begin  
update department set depart_name=d_name,emp_id=e_id,location_id=l_id  
where depart_id=d_id;  
end;  
/
```

Edit location:

```
create or replace procedure edit_location(  
l_id in location.location_id%type,  
s_address in location.street_address%type,  
p_code in location.postal_code%type,  
city in location.city%type,  
s_prov in location.state_province%type)  
is  
begin  
update location set  
street_address=s_address,postal_code=p_code,city=city,state_province=s_prov  
where location_id=l_id;  
end;  
/
```

Edit job:

```
create or replace procedure edit_job(  
j_id in jobs.job_id%type,  
j_title in jobs.job_title%type,  
min_sal in jobs.min_salary%type,  
max_sal in jobs.max_salary%type)  
is  
begin  
update jobs set job_title=j_title,min_salary=min_sal,max_salary=max_sal  
where job_id=j_id;  
end;  
/
```

Edit job history:

```

create or replace procedure edit_job_history(
  e_id in job_history.emp_id%type,
  s_date in job_history.start_date%type,
  e_date in job_history.end_date%type,
  j_id in job_history.job_id%type,
  d_id in job_history.depart_id%type)
is
begin
  update job_history set start_date=s_date,end_date=e_date,job_id=j_id,depart_id=d_id where
  emp_id=e_id;
end;
/

```

Edit training:

```

create or replace procedure edit_training(
  t_id in training.training_id%type,
  t_title in training.training_title%type,
  t_des in training.training_description%type,
  d_id in training.depart_id%type)
is
begin
  update training set training_title=t_title,training_description=t_des,
  depart_id=d_id where training_id=t_id;
end;
/

```

Edit users:

```

create or replace procedure edit_users(
  u_id in users.user_id%type,
  u_pass in users.user_password%type)
is
begin
  update users set user_password=u_pass where user_id=u_id;
end;
/

```

Edit salary:

```

create or replace procedure edit_salary(
  s_id in salary.salary_id%type,
  sal in salary.salary%type,
  bonus in salary.bonus%type,
  loan in salary.loan%type,
  last_updated in salary.last_updated%type,
  e_id in salary.emp_id%type)
is
begin
  update salary set salary=sal,bonus=bonus,loan=loan,last_updated=last_updated, emp_id=e_id
  where salary_id=s_id;
end;
/

```

Edit country:


```

create or replace procedure edit_country(
  c_id in country.country_id%type,
  c_name in country.country_name%type,
  r_id in country.region_id%type)
is
begin
  update country set country_name=c_name,region_id=r_id where country_id=c_id;
end;
/

```

Edit region:

```

create or replace procedure edit_region(
  r_id in region.region_id%type,
  r_name in region.region_name%type)
is
begin
  update region set region_name=r_name where region_id=r_id;
end;
/

```

Edit attendance:

```

create or replace procedure edit_attendance(
  atten_id in attendance.attendance_id%type,
  atten_emp in attendance.attendance_employee%type,
  atten_date in attendance.attendance_date%type)
is
begin
  update attendance set attendance_employee=atten_emp,attendance_date=atten_date where
attendance_id=atten_id;
end;
/

```

Delete employee:

```

create or replace procedure delete_employee(e_id in employee.emp_id%type)
is
begin
  delete from employee where emp_id=e_id;
  dbms_output.put_line('*****');
  dbms_output.put_line('employee ID'||e_id||'deleted');
  dbms_output.put_line('*****');
end;
/

```

Delete admin:

```

create or replace procedure delete_admin(a_id in admin.admin_id%type)
is
begin
  delete from admin where admin_id=a_id;
  dbms_output.put_line('*****');
  dbms_output.put_line('admin_id'||a_id||'deleted');
  dbms_output.put_line('*****');
end;

```

/

Delete location:

```
create or replace procedure delete_location(loc_id in location.location_id%type)
is
begin
delete from location where location_id=loc_id;
dbms_output.put_line('*****');
dbms_output.put_line('location_id||loc_id||deleted');
dbms_output.put_line('*****');
end;
/
```

Delete department:

```
create or replace procedure delete_department(d_id in department.depart_id%type)
is
begin
delete from department where depart_id=d_id;
dbms_output.put_line('*****');
dbms_output.put_line('depart_id||d_id||deleted');
dbms_output.put_line('*****');
end;
/
```

Delete job:

```
create or replace procedure delete_job(j_id in jobs.job_id%type)
is
begin
delete from jobs where job_id=j_id;
dbms_output.put_line('*****');
dbms_output.put_line('job_id||j_id||deleted');
dbms_output.put_line('*****');
end;
/
```

Delete job history:

```
create or replace procedure delete_job_history(j_id in job_history.job_id%type)
is
begin
delete from job_history where job_id=j_id;
dbms_output.put_line('*****');
dbms_output.put_line('job_id||j_id||deleted');
dbms_output.put_line('*****');
end;
/
```

Delete training:

```
create or replace procedure delete_training(t_id in training.training_id%type)
is
begin
delete from training where training_id=t_id;
dbms_output.put_line('*****');
dbms_output.put_line('training_id||t_id||deleted');
```

```
dbms_output.put_line('*****');
end;
/
```

Delete user:

```
create or replace procedure delete_user(u_id in users.user_id%type)
is
begin
dbms_output.put_line('*****');
dbms_output.put_line('user_id||u_id||deleted');
dbms_output.put_line('*****');
end;
/
```

Delete salary:

```
create or replace procedure delete_salary(s_id in salary.salary_id%type)
is
begin
dbms_output.put_line('*****');
dbms_output.put_line('salary_id||s_id||deleted');
dbms_output.put_line('*****');
end;
/
```

Delete country:

```
create or replace procedure delete_country(c_id in country.country_id%type)
is
begin
dbms_output.put_line('*****');
dbms_output.put_line('country_id||c_id||deleted');
dbms_output.put_line('*****');
end;
/
```

Delete region:

```
create or replace procedure delete_region(r_id in region.region_id%type)
is
begin
dbms_output.put_line('*****');
dbms_output.put_line('region_id||r_id||deleted');
dbms_output.put_line('*****');
end;
/
```

Delete attendance:

```
create or replace procedure delete_attendance(atten_id in attendance.attendance_id%type)
is
begin
dbms_output.put_line('*****');
dbms_output.put_line('attendance_id||atten_id||deleted');
dbms_output.put_line('*****');
end;
```

/

Triggers:

Insert value emp:

```
create or replace trigger insert_value_emp
after insert on employee
for each row
begin
if inserting then
dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
end if;
end;
/
```

Insert value admin:

```
create or replace trigger insert_value_admin
after insert on admin
for each row
begin
if inserting then
dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
end if;
end;
/
```

Insert value department:

```
create or replace trigger insert_value_department
after insert on department
for each row
begin
if inserting then
dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
end if;
end;
/
```

Insert value location:

```
create or replace trigger insert_value_location
after insert on location
for each row
begin
if inserting then
dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
end if;
end;
/
```

Insert value job:

```
create or replace trigger insert_value_job
after insert on jobs
```

```
    for each row
    begin
    if inserting then
    dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
    end if;
    end;
```

/

Insert jobhistory:

```
create or replace trigger insert_jobhistory
after insert on job_history
for each row
begin
if inserting then
dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
end if;
end;
```

/

Insert training:

```
create or replace trigger insert_training
after insert on training
for each row
begin
if inserting then
dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
end if;
end;
```

/

Insert users:

```
create or replace trigger insert_users
after insert on users
for each row
begin
if inserting then
dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
end if;
end;
```

/

Insert salary:

```
create or replace trigger insert_salary
after insert on salary
for each row
begin
if inserting then
dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
end if;
end;
```

/

Insert country:

```
create or replace trigger insert_country
after insert on country
for each row
begin
if inserting then
dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
end if;
end;
/
```

Insert region:

```
create or replace trigger insert_region
after insert on region
for each row
begin
if inserting then
dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
end if;
end;
/
```

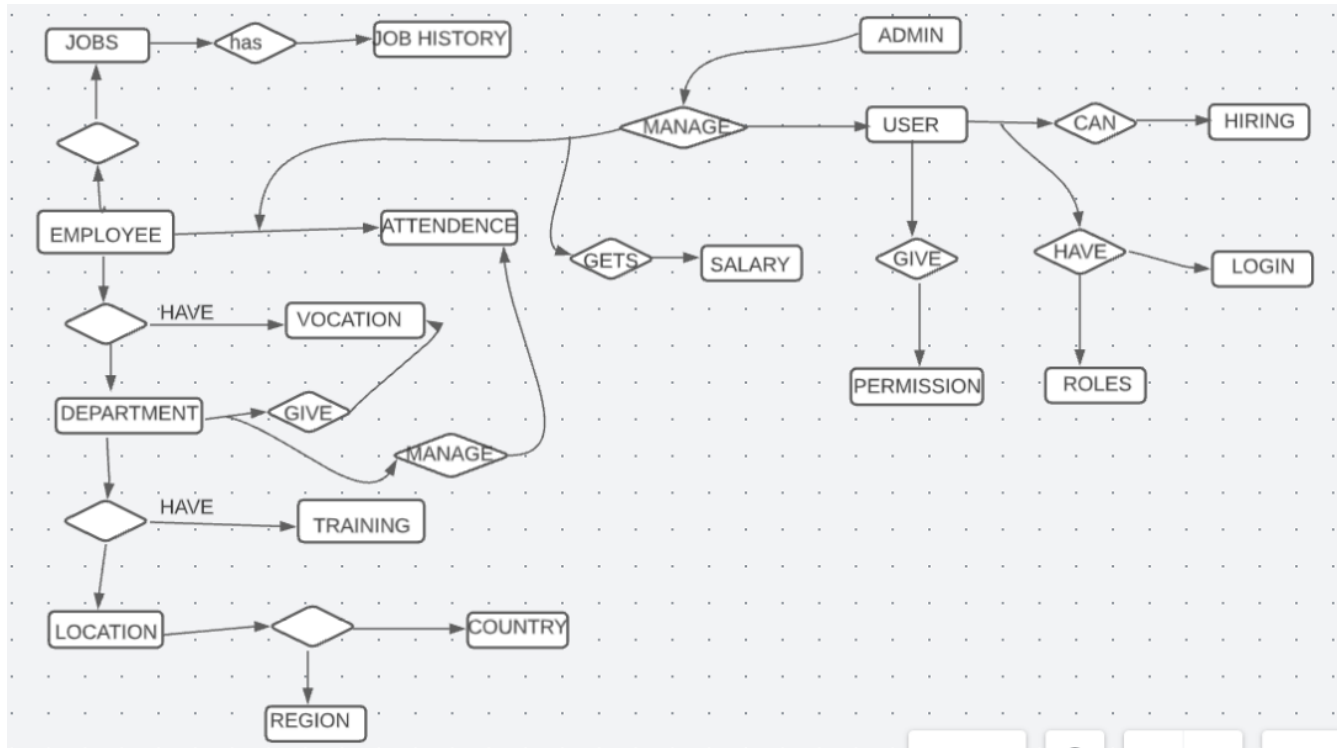
Insert attendance:

```
create or replace trigger insert_attendance
after insert on attendance
for each row
begin
if inserting then
dbms_output.put_line('RECORD INSERTED SUCCESSFULLY'||sysdate());
end if;
end;
/
```

Before insert, update or delete from employee:

```
create or replace trigger backup
before insert or update or delete on employee
for each row
begin
insert into employee values
(:old.emp_id,:old.emp_name,:old.emp_address,:old.emp_number,:old.emp_email);
end;
/
```

Entity Relationship Diagram:



NORMALIZATION:

1NF Normalization:

(Emp_ID, EMP_name, Emp_address, Emp_number, Emp_email, **Depart_ID**, Depart_Name, **Location_ID**, **Vacation_ID**, Vacation_Title, FromDate, ToDate, StreetAddress, PostalCode, City, State_Province, **JobID**, Jobtitle, minsalary, maxsalary, startdate, enddate, **TrainingID**, TrainingTitle, Trainingdescription, **AdminID**, AdminPassword, AdminEmail, AdminName, **UserID**, Userpassword, **SalaryID**, Salary, bonus, loan, lastupdate, **CountryID**, CountryName, **RegionID**, RegionName, **AttendanceID**, Attendance_employee, attendance_Date)

All the conditions of First Normal Form of Normalization are satisfied hence 1NF Normalization is achieved.

2NF Normalization:

Now as in aforementioned table, Depart_Name does not directly depends on Emp_ID hence Partial dependency exists, to remove it we make a separate table called Department.

Employee (**Emp_ID**, EMP_name,Emp_address,Emp_number,Emp_email)

Department (**Depart_ID**, Depart_Name)

Similarly we remove Partial Dependency from the main table and add Foreign Keys to each table respectively.

Employee Table:-

Employee (**Emp_ID**, EMP_name, Emp_address,Emp_number,Emp_email)

Department Table:-

Department (**Depart_ID**, Depart_Name,Emp_ID,Location_ID)

Vacation Table:-

Vacation (**Vacation_ID**,Vacation_Title,FromDate,ToDate,Emp_ID)

Location Table:-

Location (**Location_ID**, ,StreetAddress,PostalCode,City,State_Province)

Jobs Table:-

Jobs (**JobID**, Jobtitle,minsalary, maxsalary)

Jobhistory Table:-

Jobhistory(Emp_ID, startdate, enddate,Job_ID,Depart_ID)

Trainings Table:-

Trainings (**TrainingID** ,TrainingTitle, Trainingdescription, Depart_ID)

Admin Table:-

Admin (**AdminID**,AdminPassword,AdminEmail,AdminName)

User Table:-

User (**UserID**,Userpassword)

Salary Table:-

Salary (**SalaryID**,Salary,bonus,loan,lastupdate, Emp_ID)

Country Table:-

Country (**CountryID**,CountryName, RegionID)

Region Table:-

Region (**RegionID**, RegionName)

Attendance Table:-

Attendance (**AttendanceID**, Attendance_employee, attendance_Date)

3NF Normalization:-

The tables are already in Third Normal Form.

CONCLUSION:

In today world, the Human Resource Management plays a very significant role in the daily life. On the one hand, the Soft and Hard Human Resource Management influence on the business and lets them development rapidly. It can improve employee's motivation in a business and pay attention to company's policy and law respectively, which can increase the efficiency of company and get higher profits. On the other hand, trade unions help the employee to achieve negotiation successfully in the early time; it means the employee can negotiate a better wages and a good working condition. However, at the present, the employment law gradually becomes the focus in the world, because it has more restrictive and more favorable to protecting employee's benefit. In the future, the Human Resources Management will continue to play its role in each business.