***RESOURCES MANAGEMENT PROJECT***

**Version: RES-001**

|  |  |
| --- | --- |
|  |  |
| Created Date | 21/12/2019 |
| Author | Tien Nguyen |
| Title | Senior Software Engineer |

Table of Contents

1. **DOCUMENT HISTORY** 2
2. **INTRODUCTION PROJECT**4
   1. Purpose4
   2. Scope4
   3. Overview 4
3. **DATABASE**6
   1. Entity Relationship Designer6
   2. Tables7

3.2.1. ACCOUNT 8

3.2.2. ACCOUNT\_ROLES 9

3.2.3. SECURITY\_ROLES 10

3.2.4. PROFILE 11

3.2.5. TIMESHEET 12

3.2.6. TEAM 13

3.2.7. PROJECT 14

3.2.8. DEPARTMENT 15

3.2.9. BENEFIT 16

3.2.10. BILLABLE 17

3.2.11. SALARY 18

3.2.12. EQUIPMENTS 19

3.2.13. BANK 20

3.2.14. APPLICATION\_LOG 21

3.2.15. DATA\_MIGRATION 22

* 1. Workshop Services 23
     1. Account Service 24
     2. Profile Service 25
     3. TimeSheet Service 26
     4. Project Service 27
     5. Accountant Service 28

1. **TECHNICAL SERVICE** 29
   1. Spring Framework 29
   2. Micro Services 30
   3. Docker 31
2. **Extra Info** 32

1. **DOCUMENT HISTORY:**

|  |  |
| --- | --- |
| Document Location | |
| File Name | Location |
| RMS-Document\_V1.0.docx | Ho Chi Minh City, Viet Nam |
| RMS-Document\_V1.1.docx | Ho Chi Minh City, Viet Nam |
| RMS-Document\_V1.2.docx | Ho Chi Minh City, Viet Nam |

|  |  |  |  |
| --- | --- | --- | --- |
| Document Version History | | | |
| Version | Edited By | Date | Description |
| 1.0 | Tien Nguyen | 21/12/2019 | First publish document |
| 1.1 | Tien Nguyen | 6/1/2020 | Update all workflows of table relationship and descriptions |
| 1.2 | Tien Nguyen | 29/12/2020 | Update Information about system workflow |
| *This document is first created for this project, any update will increase version 0.1 – 0.5, then will change to version 2.0* | | | |

**Approvers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Approvers List | | | | |
| Name | Role | Signature | Date | Version |
| Tien Nguyen | Business Analyst  Database Analyst Designer | Signed | 21/12/2019 | 1.0 |
|  |  |  |  |  |

*PREFACE*

First thing, I would like to thank you for reading this document. I am first member of my team created this document, based on everything I learnt and knowledge to build this project. Thank you to my Developer have good ideas to build this project, using everything modern technical and gained everything experience was worked in old company. The Resources Management Project is Project using new technical: Spring Cloud (Neflix, …) + Micro Service and push data into database PostgreSQL. But not only just 1 database PostgreSQL, we will have more database for each service, it will synchronize together and all of them will be manage on 1 HUB: Docker.

1. **INTRODUCTION PROJECT:**
   1. **Purpose:**

*Resources Management Project* is the System will help owner who is hiring can manage easier about huge company:

Main Services for all companies:

Manage members / profile / account / timesheet / salary of employee /

skill, technical, HR, Admin, IT, Accountant, Projects, Team, Contact client and timesheet…

With using modern latest technical tool & language, combine with using many databases at once time and only 1 server can manage (Docker).

* 1. **Scope**:

This project will have first 5 services in 1 package:

3 Website separate as user who can access or can not access : Web – PMS, Web – TMS, Web – AMS

5 Services as base for system:

**Account Service**

**Profile Service**

**Project Service**

**Timesheet Service**

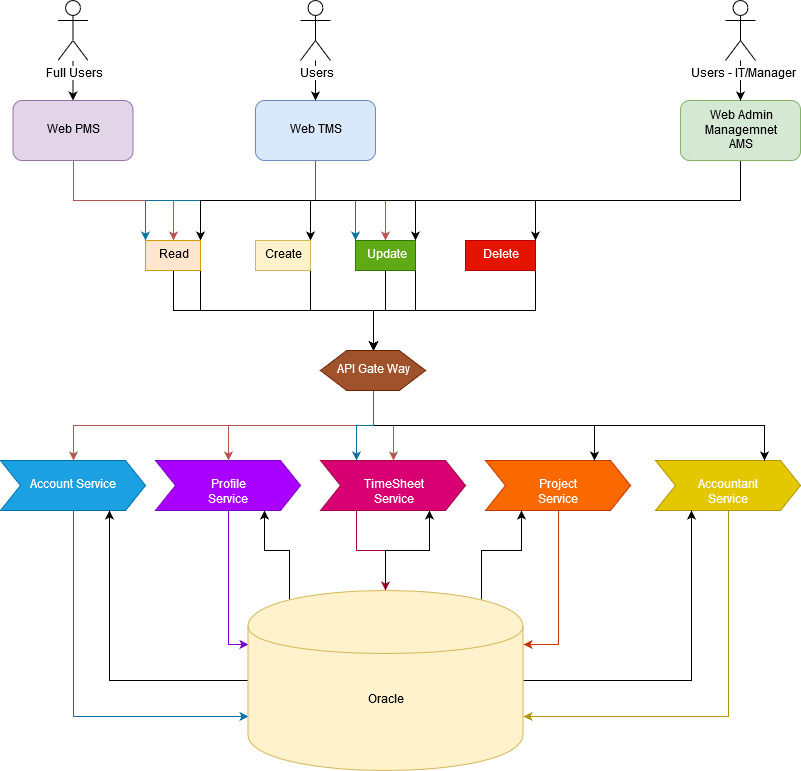
**Accountant Service**

Using Database : MySQL and push all of them into Docker as Big Server

* 1. **Overview**:

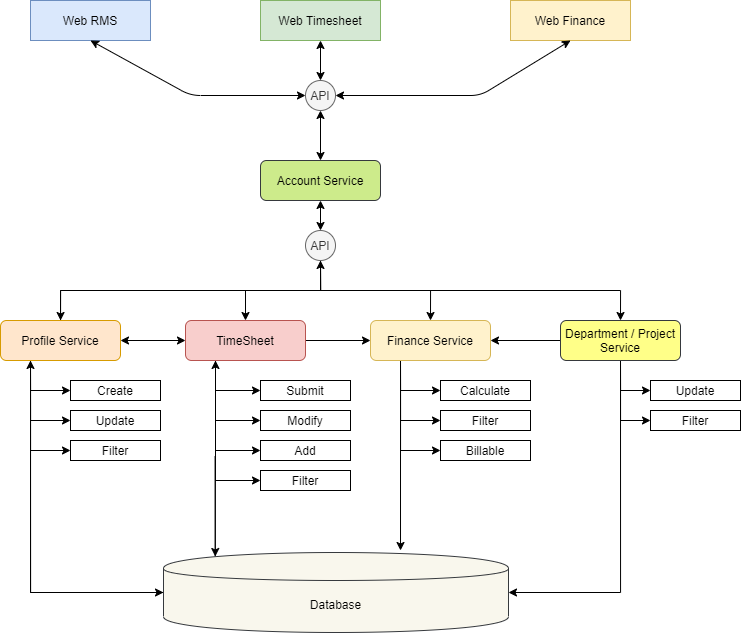
This system will follow with this document provided. If anything that want to change or extend will be new *change request*.

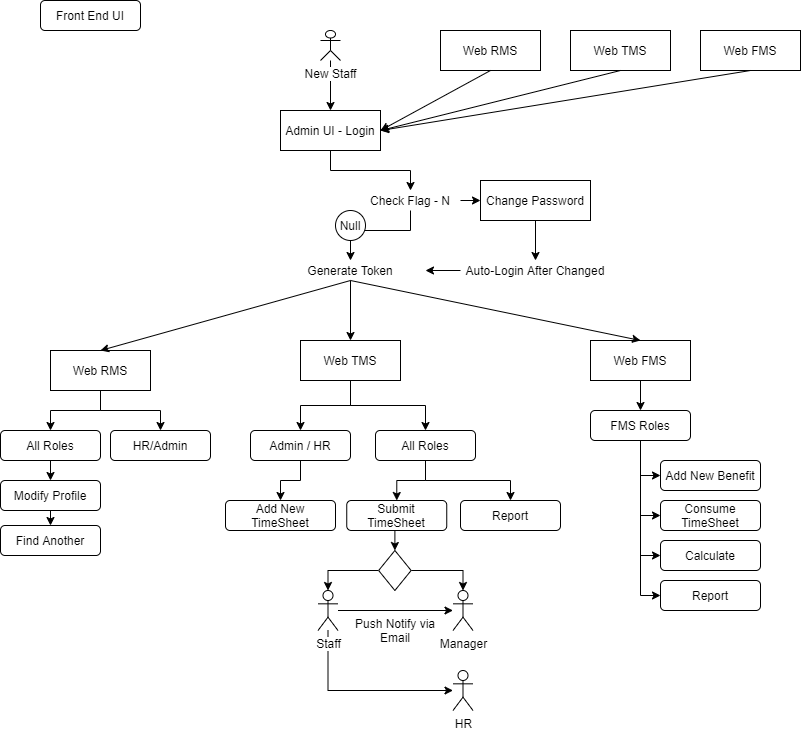
Old Version

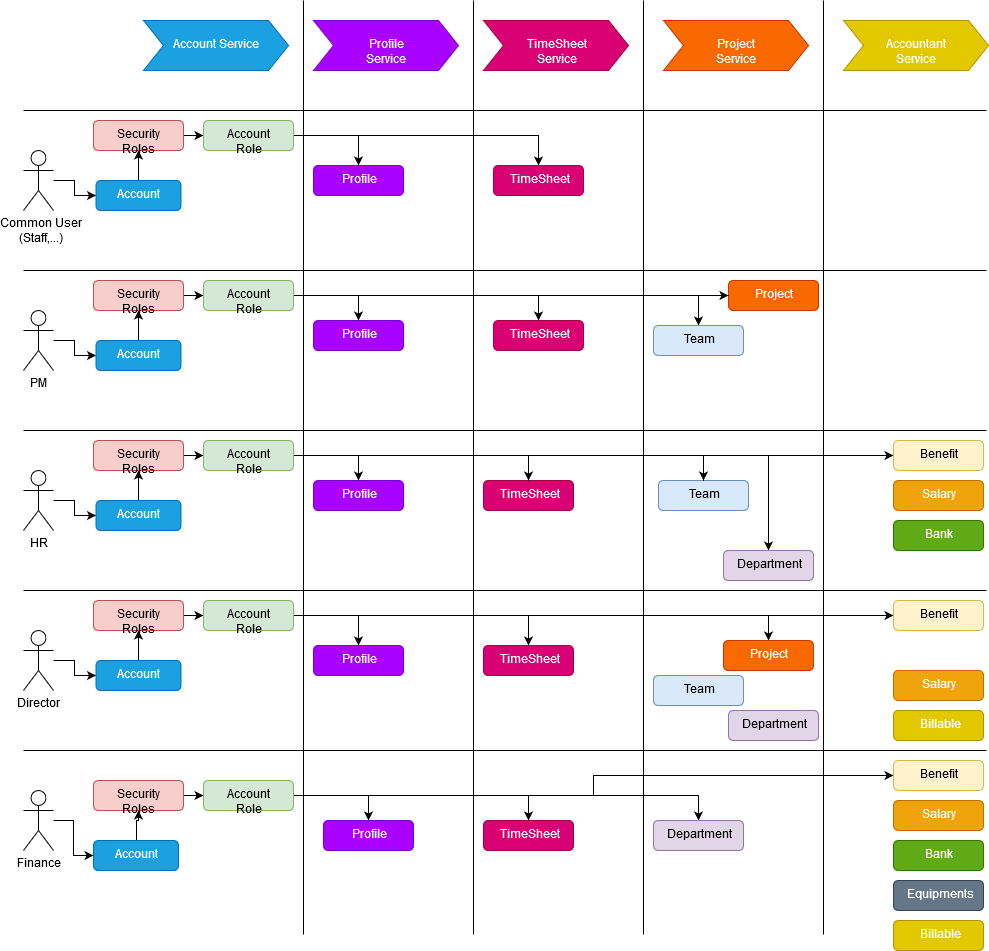


*Image 1 : simple Services and databases to take good performance and reduce resources for server.*

New Version:

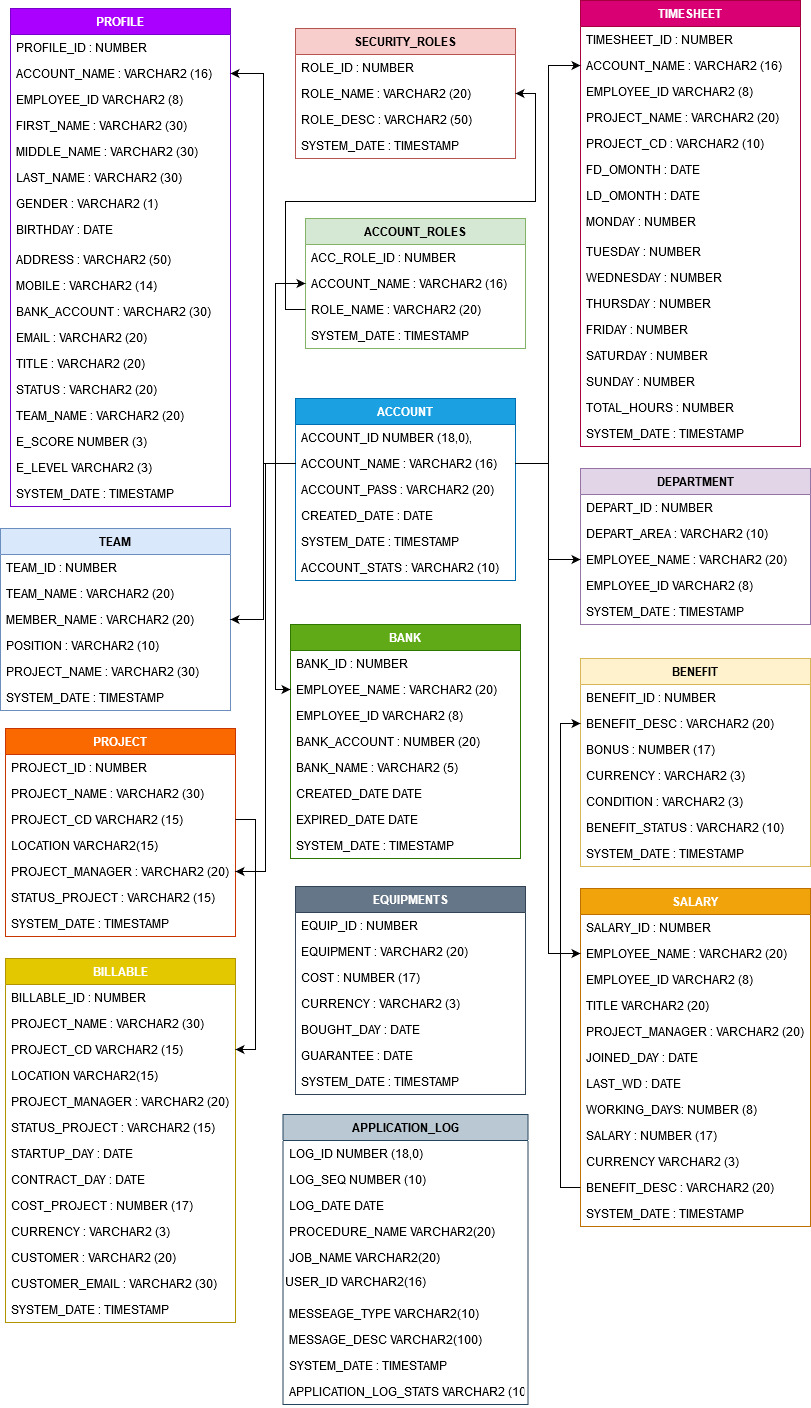




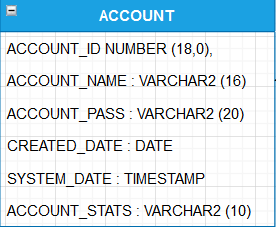


*Image 2: Simple workflows with security roles.*

1. **DATABASE**
   1. **Entity Relationship Design:**



* 1. **Tables:**
     1. **ACCOUNT**



|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Type | Primary Key | Description |
| ACCOUNT\_ID | NUMBER (18,0) | PK | Generate sequence number via trigger before insert for each row |
| ACCOUNT\_NAME | VARCHAR2 (16) |  | Account Name for user login into website |
| ACCOUNT\_PASS | VARCHAR2 (20) |  | Account pass for user login into website |
| CREATED\_DATE | DATE |  | The first date create account for user |
| SYSTEM\_DATE | TIMESTAMP |  | Tracking follow current time (DD-MMM-YYYY HH24:mm:ss.hhhhh |
| ACCOUNT\_STATS | VARCHAR2 (10) |  | Status of user account : active or inactive |

This table will handle user account & password / status of each account. If account status display “active” mean that user is still working. If not, mean that user already resigned.

*Sequence & Trigger of table account:*

CREATE SEQUENCE VM1DTA.SEQ\_ACCOUNT START WITH 1 MAXVALUE 100000 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

CREATE OR REPLACE TRIGGER VM1DTA.TR\_ACCOUNT BEFORE INSERT ON VM1DTA.ACCOUNT FOR EACH ROW

DECLARE

V\_PKVALUE NUMBER;

BEGIN

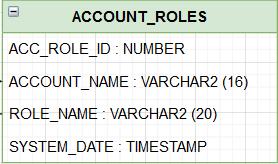
SELECT VM1DTA.SEQ\_ACCOUNT.NEXTVAL INTO V\_PKVALUE FROM DUAL;

:NEW.ACCOUNT\_ID := V\_PKVALUE;

END TR\_ACCOUNT;

/

* + 1. **ACCOUNT\_ROLES**



|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Type | Primary Key | Description |
| ACC\_ROLE\_ID | NUMBER (18,0) | PK | Generate sequence number via trigger before insert for each row |
| ACCOUNT\_NAME | VARCHAR2 (16) |  | Account Name for user login into website |
| ROLE\_NAME | VARCHAR2 (20) |  | Role Name to define level and position of user can access into the website |
| SYSTEM\_DATE | TIMESTAMP |  | Tracking follow current time (DD-MMM-YYYY HH24:mm:ss.hhhhh |

This table will handle role of each user to define which user can access inside website level. Role Staff will only access website PMS and TMS (refer to image 2: *Simple workflows with security roles.*)

*Sequence & Trigger of table account:*

CREATE SEQUENCE VM1DTA.SEQ\_ACCOUNT\_ROLES START WITH 1 MAXVALUE 100000 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

CREATE OR REPLACE TRIGGER VM1DTA.TR\_ACCOUNT\_ROLES BEFORE INSERT ON VM1DTA.ACCOUNT\_ROLES FOR EACH ROW

DECLARE

V\_PKVALUE NUMBER;

BEGIN

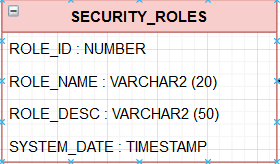
SELECT VM1DTA.SEQ\_ACCOUNT\_ROLES.NEXTVAL INTO V\_PKVALUE FROM DUAL;

:NEW.ACC\_ROLE\_ID := V\_PKVALUE;

END TR\_ACCOUNT\_ROLES;

/

* + 1. **SECURITY\_ROLES**



|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Type | Primary Key | Description |
| ROLE\_ID | NUMBER (18,0) | PK | Generate sequence number via trigger before insert for each row |
| ROLE\_NAME | VARCHAR2 (20) |  | Role Name to define level and position of user can access into the website |
| ROLE\_DESC | VARCHAR2 (50) |  | Describe meaning of each role name |
| SYSTEM\_DATE | TIMESTAMP |  | Tracking follow current time (DD-MMM-YYYY HH24:mm:ss.hhhhh |

This table is describe meaning of each role, combine with table ACCOUNT\_ROLES so that user will have the permission to access website.

*Sequence & Trigger of table account:*

CREATE SEQUENCE VM1DTA.SEQ\_SECURITY\_ROLES START WITH 1 MAXVALUE 100000 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

CREATE OR REPLACE TRIGGER VM1DTA.TR\_SECURITY\_ROLES BEFORE INSERT ON VM1DTA.SECURITY\_ROLES FOR EACH ROW

DECLARE

V\_PKVALUE NUMBER;

BEGIN

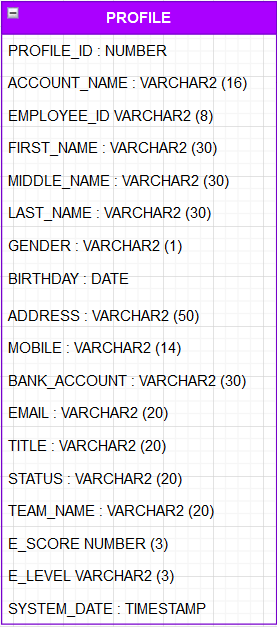
SELECT VM1DTA.SEQ\_SECURITY\_ROLES.NEXTVAL INTO V\_PKVALUE FROM DUAL;

:NEW.ROLE\_ID := V\_PKVALUE;

END TR\_SECURITY\_ROLES;

/

* + 1. **PROFILE**



|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Type | Primary Key | Description |
| PROFILE\_ID | NUMBER (18,0) | PK | Generate sequence number via trigger before insert for each row |
| ACCOUNT\_NAME | VARCHAR2 (20) |  | Account Name for user login into website |
| EMPLOYEE\_ID | VARCHAR2 (8) |  | Number employee id (EX: SD0001) |
| FIRST\_NAME | VARCHAR2 (30) |  | First name of user |
| MIDDLE\_NAME | VARCHAR2 (30) |  | Middle name of user |
| LAST\_NAME | VARCHAR2 (30) |  | Last name of user |
| GENDER | VARCHAR2 (1) |  | Gender of user |
| BIRTHDAY | DATE |  | User date of birth |
| ADDRESS | VARCHAR2 (50) |  | User Address home |
| MOBILE | VARCHAR2 (10) |  | Mobile number to contact |
| BANK\_ACCOUNT | VARCHAR2 (30) |  | User’s bank account (normally, company will supply for user) |
| EMAIL | VARCHAR2 (20) |  | Email of company provide (not personal email) |
| TITLE | VARCHAR2 (20) |  | Title of user, to define position |
| STATUS | VARCHAR2 (10) |  | Status of each user.  Official : mean that user is belong to company  Probation : mean that user is new comer and will work in 2 months |
| TEAM\_NAME | VARCHAR2 (15) |  | User will belong in which team |
| E\_SCORE | NUMBER (3,0) |  | English test score |
| E\_LEVEL | VARCHAR2 (3) |  | English level to define user matured qualified get bonus |
| SYSTEM\_DATE | TIMESTAMP |  | Tracking follow current time (DD-MMM-YYYY HH24:mm:ss.hhhhh |

This table is containing all information of user in 1 company.

*Sequence & Trigger of table account:*

CREATE SEQUENCE VM1DTA.SEQ\_PROFILE START WITH 1 MAXVALUE 100000 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

CREATE OR REPLACE TRIGGER VM1DTA.TR\_PROFILE BEFORE INSERT ON VM1DTA.PROFILE FOR EACH ROW

DECLARE

V\_PKVALUE NUMBER;

BEGIN

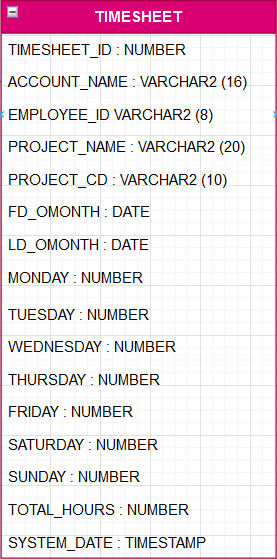
SELECT VM1DTA.SEQ\_PROFILE.NEXTVAL INTO V\_PKVALUE FROM DUAL;

:NEW.PROFILE\_ID := V\_PKVALUE;

END TR\_PROFILE;

/

* + 1. **TIMESHEET**



|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Type | Primary Key | Description |
| TIMESHEET\_ID | NUMBER (18,0) | PK | Generate sequence number via trigger before insert for each row |
| ACCOUNT\_NAME | VARCHAR2 (20) |  | Account Name for user login into website |
| EMPLOYEE\_ID | VARCHAR2 (8) |  | Number employee id (EX: SD0001) |
| PROJECT\_NAME | VARCHAR2 (30) |  | Project name which user is working |
| PROJECT\_CD | VARCHAR2 (10) |  | Project code to define that project has billable and OT |
| FD\_OMONTH | DATE |  | First day of month |
| LD\_OMONTH | DATE |  | Last day of month |
| MONDAY | NUMBER |  | Input time working (maximum : 8) |
| TUESDAY | NUMBER |  | Input time working (maximum : 8) |
| WEDNESDAY | NUMBER |  | Input time working (maximum : 8) |
| THURSDAY | NUMBER |  | Input time working (maximum : 8) |
| FRIDAY | NUMBER |  | Input time working (maximum : 8) |
| SATURDAY | NUMBER |  | Input time working (maximum : 8) |
| SUNDAY | NUMBER |  | Input time working (maximum : 8) |
| TOTAL\_HOUR | NUMBER |  | Total hours that user has worked (included OT) |
| SYSTEM\_DATE | TIMESTAMP |  | Tracking follow current time (DD-MMM-YYYY HH24:mm:ss.hhhhh |

This table will let user know how much time they worked and will help finance team to calculate salary depend on user’s working time.

*Sequence & Trigger of table account:*

CREATE SEQUENCE VM1DTA.SEQ\_TIMESHEET START WITH 1 MAXVALUE 100000 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

PROMPT CREATE TRIGGER NEXT SEQUENCE

CREATE OR REPLACE TRIGGER VM1DTA.TR\_TIMESHEET BEFORE INSERT ON VM1DTA.TIMESHEET FOR EACH ROW

DECLARE

V\_PKVALUE NUMBER;

BEGIN

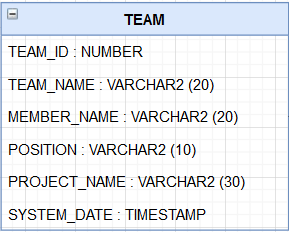
SELECT VM1DTA.SEQ\_TIMESHEET.NEXTVAL INTO V\_PKVALUE FROM DUAL;

:NEW.TIMESHEET\_ID := V\_PKVALUE;

END TR\_TIMESHEET;

/

* + 1. **TEAM**



|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Type | Primary Key | Description |
| TEAM\_ID | NUMBER (18,0) | PK | Generate sequence number via trigger before insert for each row |
| TEAM\_NAME | VARCHAR2 (16) |  | Team name under Project Manager |
| MEMBER\_NAME | VARCHAR2 (20) |  | Name of employee |
| POSITION | DATE |  | Position of each user in 1 team |
| PROJECT\_NAME | VARCHAR2 (30) |  | Project name which user is working |
| SYSTEM\_DATE | TIMESTAMP |  | Tracking follow current time (DD-MMM-YYYY HH24:mm:ss.hhhhh |

This table is handle which user belong to which team so PM (Project manager) can manage them and approve their timesheet.

*Sequence & Trigger of table account:*

CREATE SEQUENCE VM1DTA.SEQ\_TEAM START WITH 1 MAXVALUE 100000 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

CREATE OR REPLACE TRIGGER VM1DTA.TR\_TEAM BEFORE INSERT ON VM1DTA.TEAM FOR EACH ROW

DECLARE

V\_PKVALUE NUMBER;

BEGIN

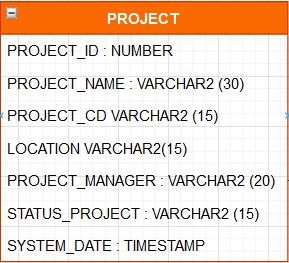
SELECT VM1DTA.SEQ\_TEAM.NEXTVAL INTO V\_PKVALUE FROM DUAL;

:NEW.TEAM\_ID := V\_PKVALUE;

END TR\_TEAM;

/

* + 1. **PROJECT**



|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Type | Primary Key | Description |
| PROJECT\_ID | NUMBER (18,0) | PK | Generate sequence number via trigger before insert for each row |
| PROJECT\_NAME | VARCHAR2 (30) |  | Project name created by confirmed by Customer and Company |
| EMPLOYEE\_ID | VARCHAR2 (8) |  | Employee ID of Project Manager |
| PROJECT\_CD | VARCHAR2 (15) |  | Project code to define that project has billable and OT |
| LOCATION | VARCHAR2 (15) |  | Location of this project was developing |
| PROJECT\_MANAGER | VARCHAR2 (20) |  | Name of Project manager take care |
| STATUS\_PROJECT | VARCHAR2 (15) |  | Status of project which is still working or which already Go live |
| SYSTEM\_DATE | TIMESTAMP |  | Tracking follow current time (DD-MMM-YYYY HH24:mm:ss.hhhhh |

This table is handling information of each project and which PM is taking care with his/her team

*Sequence & Trigger of table account:*

CREATE SEQUENCE VM1DTA.SEQ\_PROJECT START WITH 1 MAXVALUE 100000 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

CREATE OR REPLACE TRIGGER VM1DTA.TR\_PROJECT BEFORE INSERT ON VM1DTA.PROJECT FOR EACH ROW

DECLARE

V\_PKVALUE NUMBER;

BEGIN

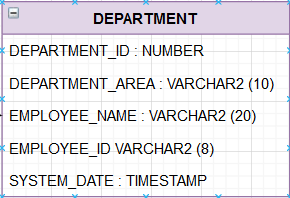
SELECT VM1DTA.SEQ\_PROJECT.NEXTVAL INTO V\_PKVALUE FROM DUAL;

:NEW.PROJECT\_ID := V\_PKVALUE;

END TR\_PROJECT;

/

* + 1. **DEPARTMENT**



|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Type | Primary Key | Description |
| DEPARTMENT\_ID | NUMBER (18,0) | PK | Generate sequence number via trigger before insert for each row |
| DEPARTMENT\_AREA | VARCHAR2 (10) |  | Department area (EX: HR, FINANCE,IT…) |
| EMPLOYEE\_NAME | VARCHAR2 (20) |  | Employee name in which department area |
| EMPLOYEE\_ID | VARCHAR2 (8) |  | Employee ID |
| SYSTEM\_DATE | TIMESTAMP |  | Tracking follow current time (DD-MMM-YYYY HH24:mm:ss.hhhhh |

This table is describing employee was working and belong in which department area.

*Sequence & Trigger of table account:*

CREATE SEQUENCE VM1DTA.SEQ\_DEPARTMENT START WITH 1 MAXVALUE 100000 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

CREATE OR REPLACE TRIGGER VM1DTA.TR\_DEPARTMENT BEFORE INSERT ON VM1DTA.DEPARTMENT FOR EACH ROW

DECLARE

V\_PKVALUE NUMBER;

BEGIN

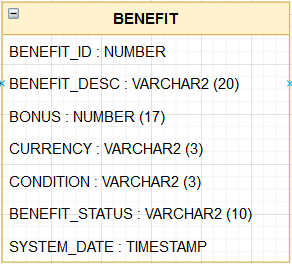
SELECT VM1DTA.SEQ\_DEPARTMENT.NEXTVAL INTO V\_PKVALUE FROM DUAL;

:NEW.DEPARTMENT\_ID := V\_PKVALUE;

END TR\_DEPARTMENT;

/

* + 1. **BENEFIT**



|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Type | Primary Key | Description |
| DEPARTMENT\_ID | NUMBER (18,0) | PK | Generate sequence number via trigger before insert for each row |
| DEPARTMENT\_AREA | VARCHAR2 (10) |  | Department area (EX: HR, FINANCE,IT…) |
| EMPLOYEE\_NAME | VARCHAR2 (20) |  | Employee name in which department area |
| EMPLOYEE\_ID | VARCHAR2 (8) |  | Employee ID |
| SYSTEM\_DATE | TIMESTAMP |  | Tracking follow current time (DD-MMM-YYYY HH24:mm:ss.hhhhh |

This table is containing all bonus and benefit. It will provide when someone gained any certificated.

*Sequence & Trigger of table account:*

CREATE SEQUENCE VM1DTA.SEQ\_BENEFIT START WITH 1 MAXVALUE 100000 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

CREATE OR REPLACE TRIGGER VM1DTA.TR\_BENEFIT BEFORE INSERT ON VM1DTA.BENEFIT FOR EACH ROW

DECLARE

V\_PKVALUE NUMBER;

BEGIN

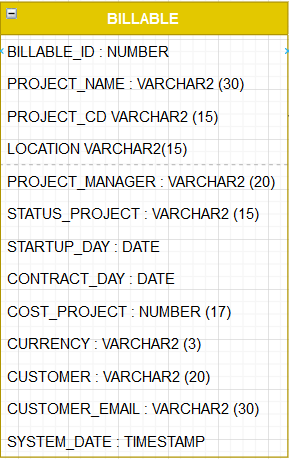
SELECT VM1DTA.SEQ\_BENEFIT.NEXTVAL INTO V\_PKVALUE FROM DUAL;

:NEW.BENEFIT\_ID := V\_PKVALUE;

END TR\_BENEFIT;

/

* + 1. **BILLABLE**



|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Type | Primary Key | Description |
| BILLABLE\_ID | NUMBER (18,0) | PK | Generate sequence number via trigger before insert for each row |
| PROJECT\_NAME | VARCHAR2 (30) |  | Project name created by confirmed by Customer and Company |
| PROJECT\_CD | VARCHAR2 (15) |  | Project code to define that project has billable and OT |
| LOCATION | VARCHAR2 (15) |  | Location of this project was developing |
| PROJECT\_MANAGER | VARCHAR2 (20) |  | Name of Project manager take care |
| STATUS\_PROJECT | VARCHAR2 (15) |  | Status of project which is still working or which already Go live |
| STARTUP\_DAY | DATE |  | The date that Project is starting |
| WORKING\_DATE | NUMBER (17) |  | Number of day that project has spent time |
| CONTRACT\_DAY | DATE |  | The date signed by Customer with company |
| COST\_PROJECT | NUMBER (17) |  | Cost project that customer has to pay (not include new CR) |
| CURRENCY | VARCHAR2 (3) |  | Currency money |
| CUSTOMER | VARCHAR2 (20) |  | Name of Customer to contact |
| CUSTOMER\_EMAIL | VARCHAR2 (30) |  | Customer email |
| SYSTEM\_DATE | TIMESTAMP |  | Tracking follow current time (DD-MMM-YYYY HH24:mm:ss.hhhhh |

This table is containing whole info of all projects in 1 company (include Customer name and their email). Cost to build complete 1 project (will have extra if they want to create new CR).

*Sequence & Trigger of table account:*

CREATE SEQUENCE VM1DTA.SEQ\_BILLABLE START WITH 1 MAXVALUE 100000 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

PROMPT CREATE TRIGGER NEXT SEQUENCE

CREATE OR REPLACE TRIGGER VM1DTA.TR\_BILLABLE BEFORE INSERT ON VM1DTA.BILLABLE FOR EACH ROW

DECLARE

V\_PKVALUE NUMBER;

BEGIN

SELECT VM1DTA.SEQ\_BILLABLE.NEXTVAL INTO V\_PKVALUE FROM DUAL;

:NEW.BILLABLE\_ID := V\_PKVALUE;

END TR\_BILLABLE;

/

* + 1. **SALARY**
    2. **EQUIPMENTS**
    3. **BANK**
  1. **Workshop Services**
     1. **Account Service:**
     2. **Profile Service:**
     3. **TimeSheet Service:**
     4. **Project Service:**
     5. **Accountant Service:**

1. **TECHNICAL LANGUAGE**
   1. **Spring Framework**
   2. **Micro Services**
   3. **Docker**
2. **Extra Info**

<Ending>

THANK YOU FOR YOUR READING

– Tien Nguyen

</Ending>