PAYROLL MANAGEMENT SYSTEM

DATABASE DESIGN

Aastha Dixit (axd155030)

Keerthi Bala Sundram (kxb152730)

TABLE OF CONTENTS

1 GENERAL REQUIREMENTS OF THE SYSTEM........ 2

2 ENTITY RELATIONSHIP DIAGRAM........ 4

3 MAPPING ER TO RELATIONAL SCHEMA........ 6

4 FUNCTIONAL DEPENDENCIES........ 9

5 NORMALISATION........ 9

6 SQL STATEMENTS........ 9

6.1 CREATE STATEMENTS........ 9

6.2 INSERT STATEMENTS........ 14

6.3 OTHER STATEMENTS........ 18

7 PL/SQL SECTION........ 20

1 GENERAL REQUIREMENTS OF THE SYSTEM

1. The Payroll system processes salary for all the employees in a company. The Payroll will be run each month for all the employees and calculates the gross salary, net salary, tax, and pays it to the employee with a unique transaction number (payment reference number).
2. An employee will follow a holiday calendar which will be based out of his/her job location (country). A holiday calendar will be same for all the employees of the same country.
3. Each employee works for a department. Each department controls many projects in which multiple employees may be assigned to.
4. The Clock in and Clock out information of the employee is captured for all the working days.
5. Each employee is entitled for two types of leaves, Earned leave (EL) and Casual Leave (CL). For every start of the quarter, an employee is entitled for 4 Earned leaves, and for every start of the year, an employee is entitled for 12 Casual leaves. Any absence/leave beyond the entitled EL, and CL is treated as Loss of Pay, and proportionate salary should be deducted.
6. Each employee can avail an insurance plan. Employee can choose any ONE from the existing plans available. The calculated premium is added to the salary deduction component every month.
7. Each employee has a stage code based on the experience in the company. The designation of the employee and the stage code determines the basic pay of the employee. When an employee joins the company, the stage code is set to 1. For each completion year of service or promotion, the employee moves to the next stage code of the basic pay.

Designation	**Salary Code**	**Basic Pay (in USD)**
Associate	1,2,3	2500-500
Software Engineer	1,2,3	3100-550
Senior Software Engineer	1,2	4200-700
Technical Lead	1,2	5450-850
Project Manager	1,2,3	6000-900
Senior Project Lead	1,2,3	6950-1050
Associate Director	1,2	7590-1300
Director	1,2	8510-1550
Associate Vice President	1,2	9520-1900
Vice President	1,2	10550-2350
Chief Executive Officer	1	12700-3000

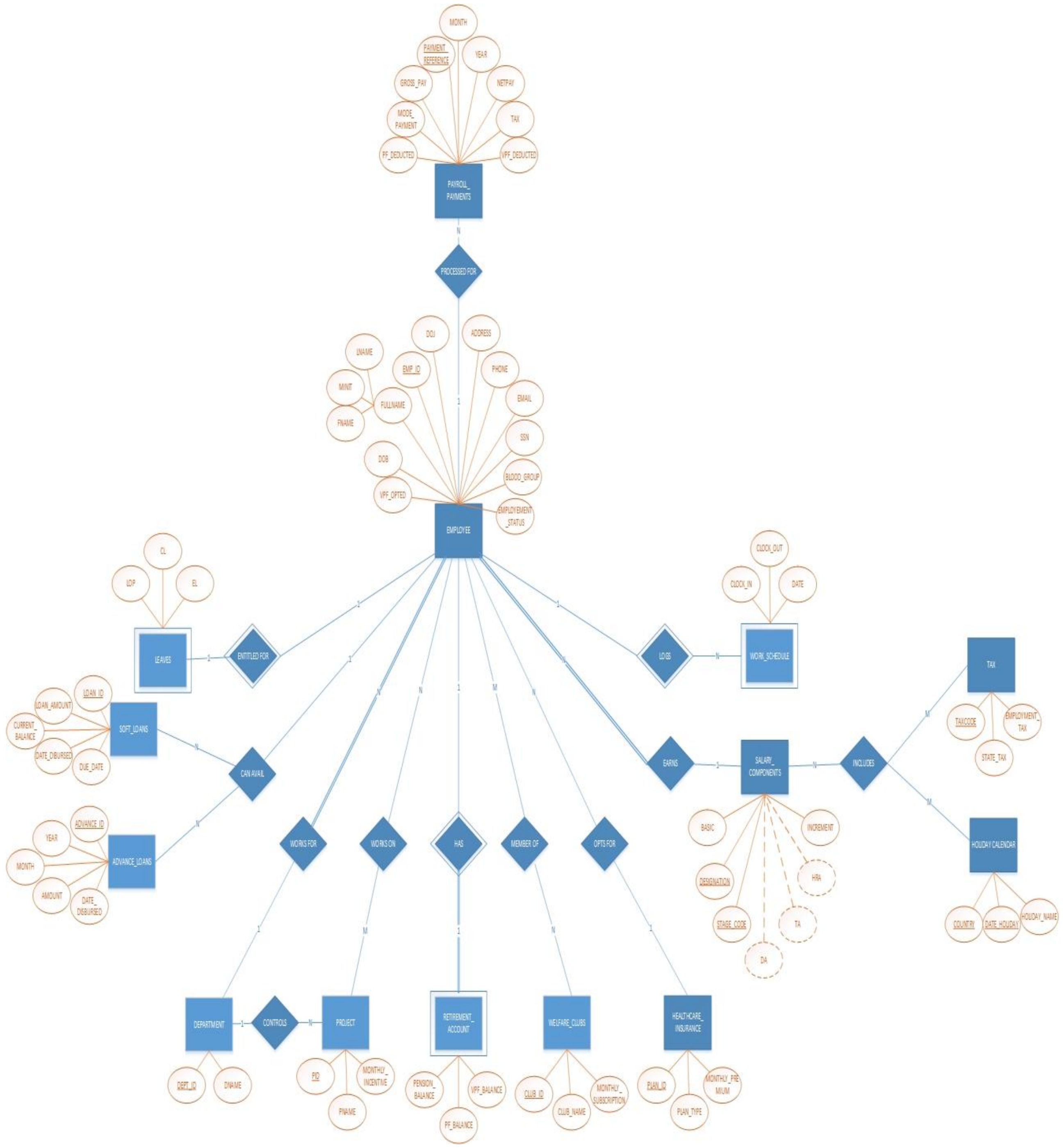
8. The company provides two types of salary loans – Soft Loan, and Salary Advance. An employee can avail only one loan of each type at a time. The maximum amount that an employee can avail in a Soft loan type is 6 times the Basic Pay entitled. An Employee can take an amount that is equal to the last month's net salary as Salary Advance. The employee can pay the soft loan in 24 equated installments.
9. Tax is calculated for an employee each month during payroll processing. The tax deducted is based on the tax code the employee belongs to. The location of the employee and the basic pay determines the tax code. (for the purpose of project it is assumed that company operates in three countries)

Country	**Basic Pay range (normalized in USD)**	**Statutory Tax %**	**Employment Tax %**
USA	Below $3000	10%	2%
USA	Below $6000	20%	5%
USA	Below $10000	30%	7%
USA	Above $10000	35%	11%
INDIA	Below $2000	10%	1%
INDIA	Below $5000	20%	4%
INDIA	Above $5000	30%	8%
CHINA	Below $2500	5%	3%
CHINA	Below $6000	15%	7%
CHINA	Below $9000	20%	11%
CHINA	Above $9000	20%	15%

10. An employee can be a member of multiple welfare organizations within the company. The employee pays a monthly subscription amount through payroll for each membership that is subscribed.
11. Each employee is subjected to a fixed statutory deductions that includes Provident Fund, Pension Fund, and a Voluntary Provident Fund. The system maintains the balances of these funds after every payroll run. 10% of Basic Pay is contributed from Employees salary to PF Account, an equal amount is deposited by the company in Pension Account of that employee. An employee may contribute to a VPF scheme with monthly contribution not exceeding (90% of Cumulative Annual Basic/12).

12. Basic Pay, Dearness Allowance, House Rent Allowance, Transportation Allowance, Monthly project incentive are the Earnings that every employee receives in a month. Based upon his retirement plan, health insurance, tax, memberships, loans availed, loss of pay – the deductions are carried out in the payroll period.
13. The total salary payable to the employee incorporating the gross payable through earnings, net payable after deductions, and taxations, monthly retirement savings, Payment reference details can be taken as payslips by the administrator.

2 ENTITY RELATIONSHIP DIAGRAM



Explanation of the ER Diagram:

Every employee works for a department, which controls multiple projects. Employee can be in multiple projects. Each project has a monthly incentive to its employees.

The Salary Components entity contains all the basic pay details as per the stage code and designation. Each employee also follows a particular holiday calendar which makes him entitled to leaves only of that particular calendar. The work schedule entity stores the IN and OUT time of the employee.

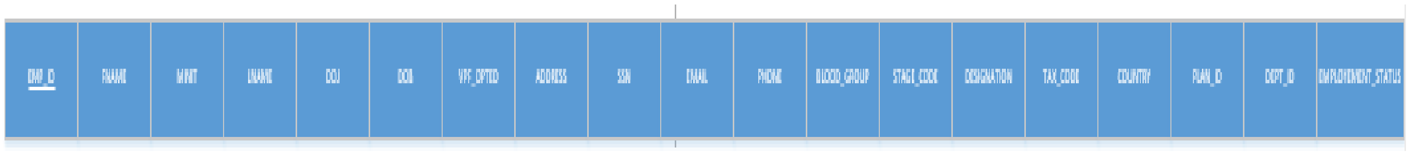
Each employee can also opt for Healthcare Insurance plans.

The Payroll Payments will be processed for all the employees. Each employee is entitled to avail leaves up to his credit The Leave Update PL SQL block will be used by the administrator at the start of every quarter to update the leave balance as per the requirements.

Every employee is entitled to take soft loans or salary advance from the company. Soft Loan can be a maximum of six month last drawn basic salary. Advance can be taken by the employee before the payroll runs for that month. For the purpose of calculation, the employees last month disbursed salary is taken for reference.

3 MAPPING ER TO RELATIONAL SCHEMA

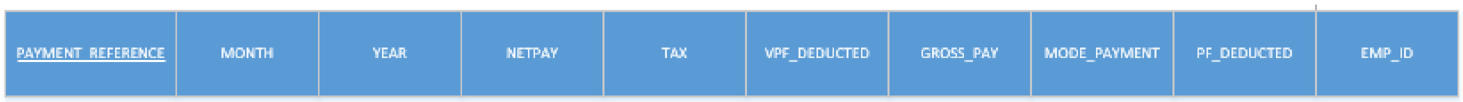
EMPLOYEE:



EMP_ID	FNAME	MINIT	LNAME	DOJ	DOB	VPF_OPTED	ADDRESS	SSN	EMAIL	PHONE	BLOOD_GROUP	STAGE_CODE	DESIGNATION	TAX_CODE	COUNTRY	PLAN_ID	DEPT_ID	EMPLOYEMENT_STATUS

Primary Key for EMPLOYEE: EMP_ID
Foreign Keys in EMPLOYEE : PLAN_ID refers HEALTHCARE_INSURANCE(PLAN_ID)
DEPT_ID refers DEPARTMENT(DEPT_ID)
TAXCODE refers TAX(TAXCODE)
STAGE_CODE AND DESIGNATION refers SALARY_COMPONENTS(STAGE_CODE, DESIGNATION)

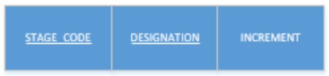
PAYMENT_PAYROLL



PAYMENT_REFERENCE	MONTH	YEAR	NETPAY	TAX	VPF_DEDUCTED	GROSS_PAY	MODE_PAYMENT	PF_DEDUCTED	EMP_ID

Primary Key for PAYMENT_PAYROLL : PAYMENT_REFERENCE
Foreign Keys in PAYMENT : EMP_ID refers EMPLOYEE(EMP_ID)

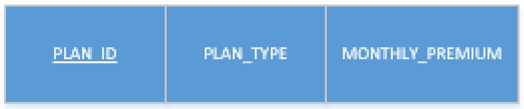
SALARY_COMPONENTS



STAGE_CODE	DESIGNATION	INCREMENT

Primary Key for SALARY_COMPONENTS: STAGE_CODE and DESIGNATION

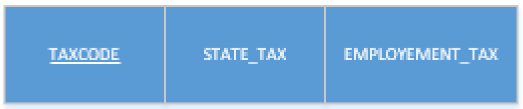
HEALTHCARE_INSURANCE



PLAN_ID	PLAN_TYPE	MONTHLY_PREMIUM

Primary Key for HEALTHCARE_INSURANCE : PLAN_ID

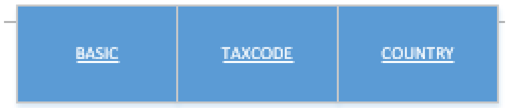
TAX



TAXCODE	STATE_TAX	EMPLOYEMENT_TAX

Primary Key for TAX : TAXCODE

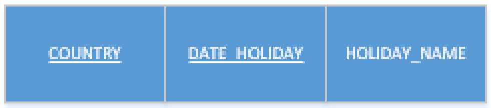
TAX_SLABS



BASIC	TAXCODE	COUNTRY

Primary Key for TAX_SLABS : BASIC, TAXCODE, COUNTRY

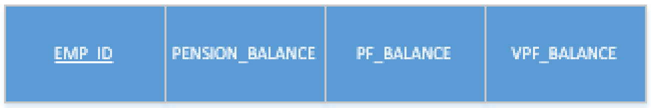
HOLIDAY_CALENDAR



COUNTRY	DATE_HOLIDAY	HOLIDAY_NAME

Primary Key for HOLIDAY_CALENDAR : COUNTRY, DATE_HOLIDAY

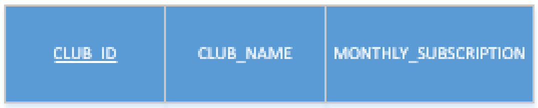
RETIREMENT_ACCOUNT



EMP_ID	PENSION_BALANCE	PF_BALANCE	VPF_BALANCE

Primary Key for RETIREMENT_ACCOUNT: EMP_ID which refers to the EMPLOYEE (EMP_ID)

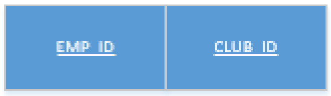
WELFARE_CLUB



CLUB_ID	CLUB_NAME	MONTHLY_SUBSCRIPTION

Primary Key for WELFARE_CLUB : CLUB_ID

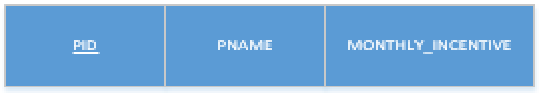
MEMBERSHIP



EMP_ID	CLUB_ID

Primary Key for WELFARE_MEMBERSHIP: EMP_ID which refers EMPLOYEE (EMP_ID) and WELFARE_CLUB (CLUB_ID)

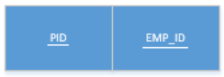
PROJECT



PID	PNAME	MONTHLY_INCENTIVE

Primary Key for PROJECT : PID

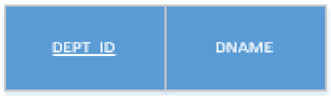
PROJECT_DETAILS



PID	EMP_ID

Primary Key: PID which refers to PROJECT (PID), EMP_ID which refers to EMPLOYEE (EMP_ID)

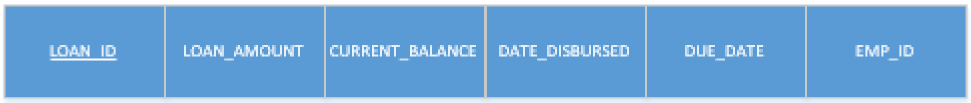
DEPARTMENT



DEPT_ID	DNAME

Primary Key: DEPT_ID

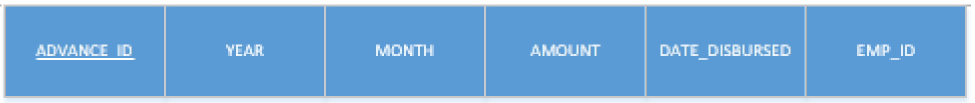
SOFT_LOANS



LOAN_ID	LOAN_AMOUNT	CURRENT_BALANCE	DATE_DISBURSED	DUE_DATE	EMP_ID

Primary Key: LOAN_ID
Foreign Key: EMP_ID refers to EMPLOYEE (EMP_ID)

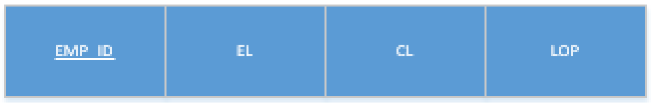
ADVANCE_LOANS



ADVANCE_ID	YEAR	MONTH	AMOUNT	DATE_DISBURSED	EMP_ID

Primary Key: ADVANCE_ID
Foreign Key: EMP_ID refers to EMPLOYEE (EMP_ID)

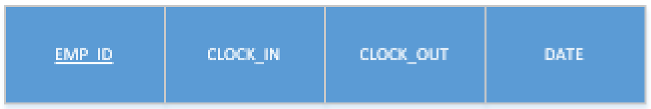
LEAVES



EMP_ID	EL	CL	LOP

Primary Key: EMP_ID which refers EMPLOYEE (EMP_ID)

WORK_SCHEDULE



EMP_ID	CLOCK_IN	CLOCK_OUT	DATE

Primary Key: EMP_ID which refers EMPLOYEE (EMP_ID)

4 FUNCTIONAL DEPENDENCIES

There are no partial dependencies and transitive dependencies related to this project. All the non-prime attributes are determined only by the primary key for that respective table**.**

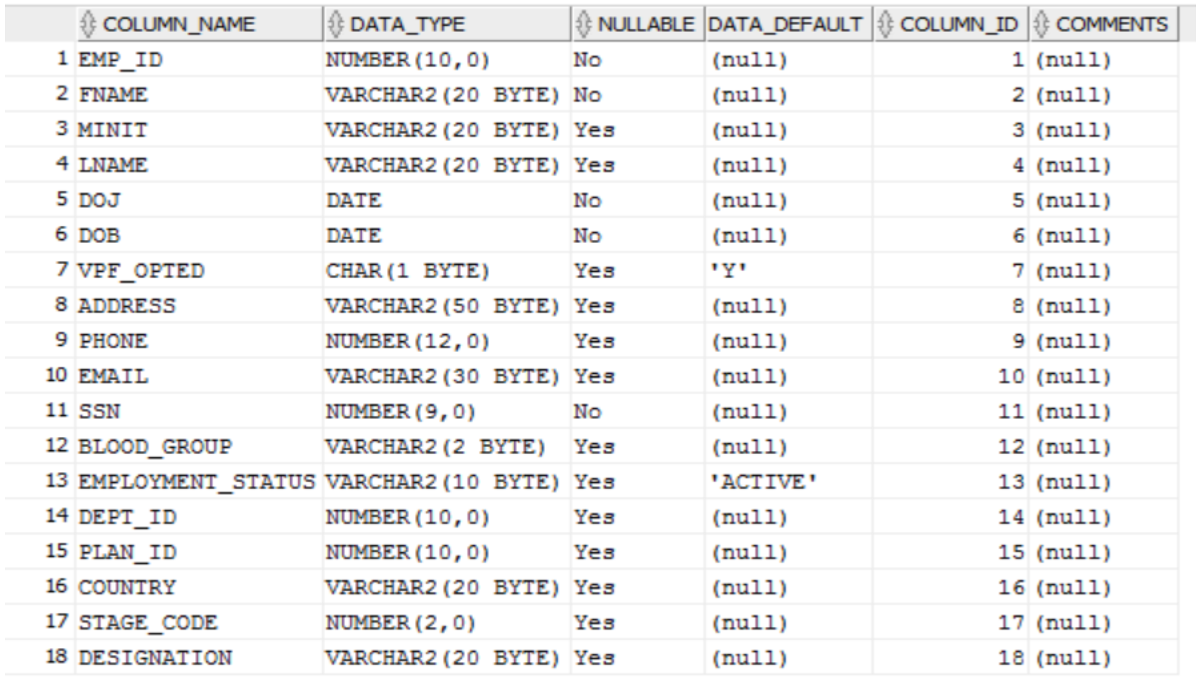
5 NORMALISATION

The Tables are in normalized form. Our design phase of the schema resulted in a relation where the attributes depend on nothing but the key. Hence the above given relations are normalized.

6 SQL STATEMENTS

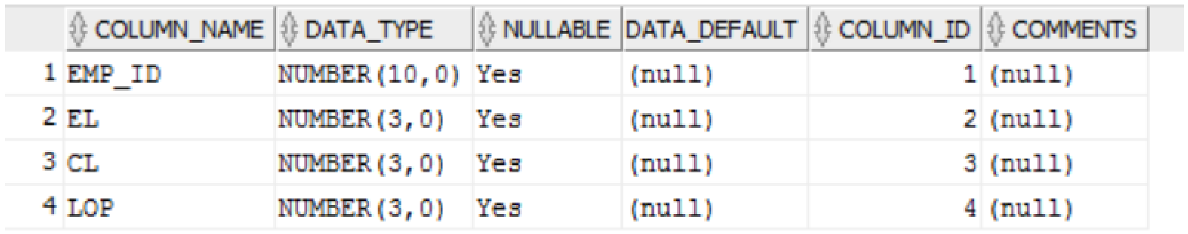
6.1 CREATE STATEMENTS

```
DROP TABLE EMPLOYEE;
DROP TABLE LEAVES;
DROP TABLE SOFT_LOANS;
DROP TABLE ADVANCE_LOANS;
DROP TABLE DEPARTMENT;
DROP TABLE PROJECT;
DROP TABLE WORKS_ON;
DROP TABLE RETIREMENT_ACCOUNT;
DROP TABLE MEMBERSHIP;
DROP TABLE WELFARE_CLUBS;
DROP TABLE HEALTHCARE_INSURANCE;
DROP TABLE HOLIDAY_CALENDAR;
DROP TABLE TAX;
DROP TABLE TAX_SLAB;
DROP TABLE SALARY_COMPONENTS;
DROP TABLE PAYROLL_PAYMENTS;

CREATE TABLE EMPLOYEE(EMP_ID NUMBER(10) CONSTRAINT EMP_PK PRIMARY KEY,FNAME VARCHAR2(20) CONSTRAINT
FNAME_NN NOT NULL,MINIT VARCHAR2(20),LNAME VARCHAR2(20),DOJ DATE CONSTRAINT DOJ_NN NOT NULL,DOB DATE
CONSTRAINT DOB_NN NOT NULL,VPF_OPTED CHAR DEFAULT 'Y',ADDRESS VARCHAR2(50),PHONE NUMBER(12),EMAIL
VARCHAR2(30),SSN NUMBER(9) CONSTRAINT SSN_NN NOT NULL, BLOOD_GROUP VARCHAR2(2),EMPLOYMENT_STATUS
VARCHAR2(10) DEFAULT 'ACTIVE',DEPT_ID NUMBER(10),PLAN_ID NUMBER(10),COUNTRY VARCHAR2(20) ,CONSTRAINT
EMP_HEA_FK FOREIGN KEY (PLAN_ID) REFERENCES HEALTHCARE_INSURANCE(PLAN_ID) ON DELETE CASCADE, CONSTRAINT
EMP_DEP_FK FOREIGN KEY (DEPT_ID) REFERENCES DEPARTMENT(DEPT_ID) ON DELETE CASCADE,STAGE_CODE
NUMBER(2),DESIGNATION VARCHAR2(20),CONSTRAINT EMP_SAL_FK FOREIGN KEY (STAGE_CODE,DESIGNATION)
REFERENCES SALARY_COMPONENTS(STAGE_CODE,DESIGNATION) ON DELETE CASCADE);
```



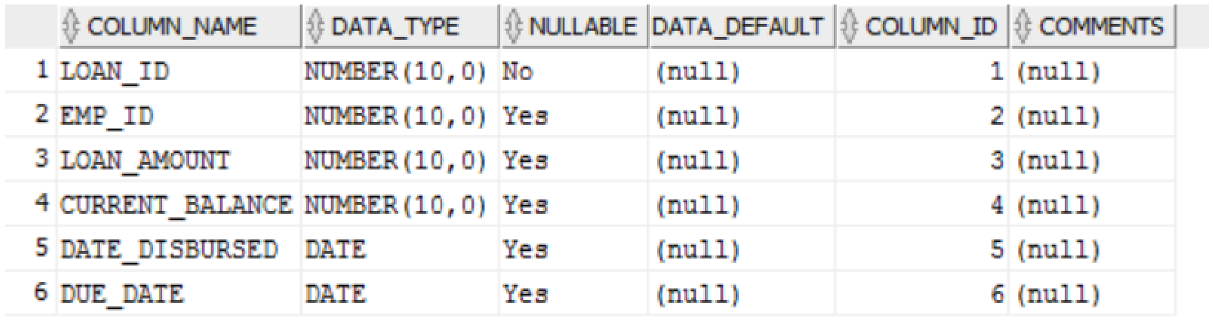
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMP_ID	NUMBER(10,0)	No	(null)	1	(null)
2	FNAME	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	MINIT	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	LNAME	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)
5	DOJ	DATE	No	(null)	5	(null)
6	DOB	DATE	No	(null)	6	(null)
7	VPF_OPTED	CHAR(1 BYTE)	Yes	'Y'	7	(null)
8	ADDRESS	VARCHAR2(50 BYTE)	Yes	(null)	8	(null)
9	PHONE	NUMBER(12,0)	Yes	(null)	9	(null)
10	EMAIL	VARCHAR2(30 BYTE)	Yes	(null)	10	(null)
11	SSN	NUMBER(9,0)	No	(null)	11	(null)
12	BLOOD_GROUP	VARCHAR2(2 BYTE)	Yes	(null)	12	(null)
13	EMPLOYMENT_STATUS	VARCHAR2(10 BYTE)	Yes	'ACTIVE'	13	(null)
14	DEPT_ID	NUMBER(10,0)	Yes	(null)	14	(null)
15	PLAN_ID	NUMBER(10,0)	Yes	(null)	15	(null)
16	COUNTRY	VARCHAR2(20 BYTE)	Yes	(null)	16	(null)
17	STAGE_CODE	NUMBER(2,0)	Yes	(null)	17	(null)
18	DESIGNATION	VARCHAR2(20 BYTE)	Yes	(null)	18	(null)

CREATE TABLE LEAVES(EMP_ID NUMBER(10),EL NUMBER(3),CL NUMBER(3),LOP NUMBER(3),CONSTRAINT LEA_FK FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID) ON DELETE CASCADE);



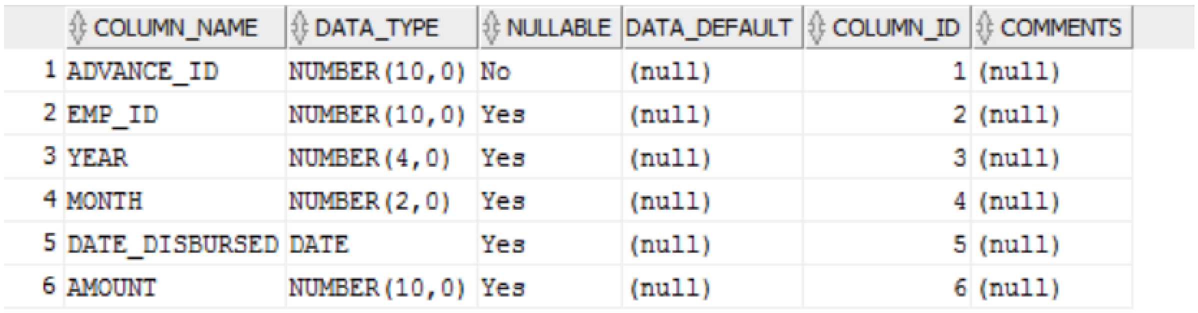
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMP_ID	NUMBER(10,0)	Yes	(null)	1	(null)
2	EL	NUMBER(3,0)	Yes	(null)	2	(null)
3	CL	NUMBER(3,0)	Yes	(null)	3	(null)
4	LOP	NUMBER(3,0)	Yes	(null)	4	(null)

CREATE TABLE SOFT_LOANS(LOAN_ID NUMBER(10) CONSTRAINT SOF_PK PRIMARY KEY,EMP_ID NUMBER(10), LOAN_AMOUNT NUMBER(10), CURRENT_BALANCE NUMBER(10), DATE_DISBURSED DATE, DUE_DATE DATE,CONSTRAINT SOF_FK FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID) ON DELETE CASCADE);



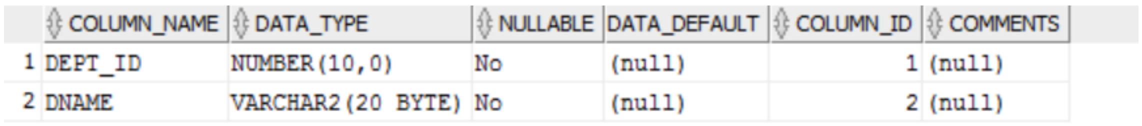
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	LOAN_ID	NUMBER(10,0)	No	(null)	1	(null)
2	EMP_ID	NUMBER(10,0)	Yes	(null)	2	(null)
3	LOAN_AMOUNT	NUMBER(10,0)	Yes	(null)	3	(null)
4	CURRENT_BALANCE	NUMBER(10,0)	Yes	(null)	4	(null)
5	DATE_DISBURSED	DATE	Yes	(null)	5	(null)
6	DUE_DATE	DATE	Yes	(null)	6	(null)

CREATE TABLE ADVANCE_LOANS(ADVANCE_ID NUMBER(10) CONSTRAINT ADV_PK PRIMARY KEY,EMP_ID NUMBER(10), YEAR NUMBER(4), MONTH NUMBER(2), DATE_DISBURSED DATE, AMOUNT NUMBER(10),CONSTRAINT ADV_FK FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID) ON DELETE CASCADE);



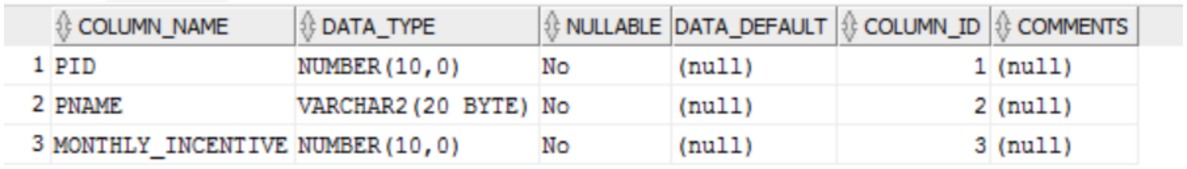
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ADVANCE_ID	NUMBER(10,0)	No	(null)	1	(null)
2	EMP_ID	NUMBER(10,0)	Yes	(null)	2	(null)
3	YEAR	NUMBER(4,0)	Yes	(null)	3	(null)
4	MONTH	NUMBER(2,0)	Yes	(null)	4	(null)
5	DATE_DISBURSED	DATE	Yes	(null)	5	(null)
6	AMOUNT	NUMBER(10,0)	Yes	(null)	6	(null)

CREATE TABLE DEPARTMENT(DEPT_ID NUMBER(10) CONSTRAINT DEP_PK PRIMARY KEY,DNAME VARCHAR2(20) CONSTRAINT DEPT_NN NOT NULL);



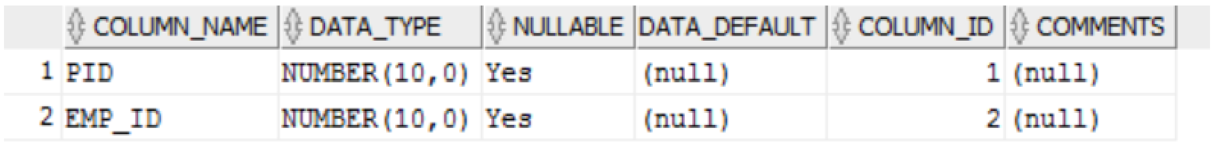
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	DEPT_ID	NUMBER(10,0)	No	(null)	1	(null)
2	DNAME	VARCHAR2(20 BYTE)	No	(null)	2	(null)

CREATE TABLE PROJECT(PID NUMBER(10) CONSTRAINT PRO_PK PRIMARY KEY,PNAME VARCHAR2(20) CONSTRAINT PRO_NN NOT NULL, MONTHLY_INCENTIVE NUMBER(10) CONSTRAINT PROJ_MI_NN NOT NULL);



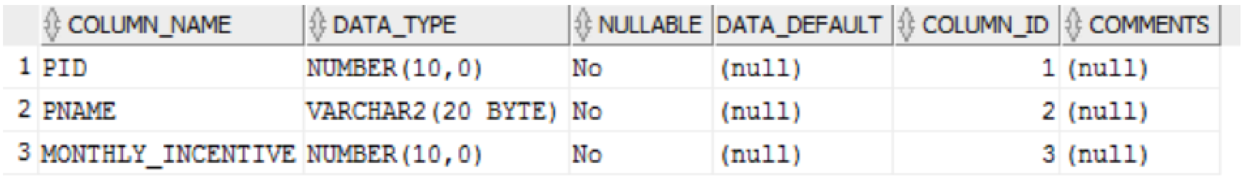
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PID	NUMBER(10,0)	No	(null)	1	(null)
2	PNAME	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	MONTHLY_INCENTIVE	NUMBER(10,0)	No	(null)	3	(null)

CREATE TABLE WORKS_ON(PID NUMBER(10),EMP_ID NUMBER(10),CONSTRAINT WOR_EMP_FK FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID) ON DELETE CASCADE,CONSTRAINT WOR_PRO_FK FOREIGN KEY (PID) REFERENCES PROJECT(PID) ON DELETE CASCADE);



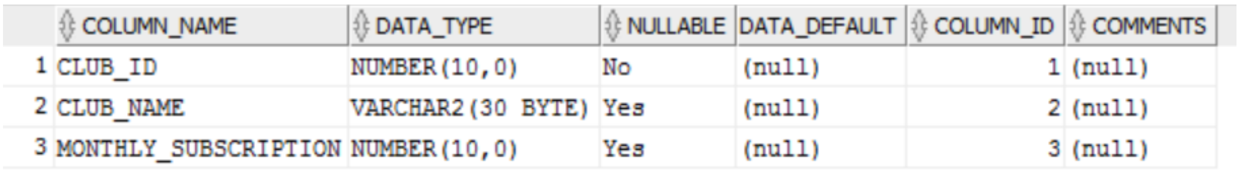
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PID	NUMBER(10,0)	Yes	(null)	1	(null)
2	EMP_ID	NUMBER(10,0)	Yes	(null)	2	(null)

CREATE TABLE RETIREMENT_ACCOUNT(EMP_ID NUMBER(10) CONSTRAINT RET_NN PRIMARY KEY,PENSION_BALANCE NUMBER(10),PF_BALANCE NUMBER(10),VPF_BALANCE NUMBER(10),CONSTRAINT RET_FK FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID) ON DELETE CASCADE);



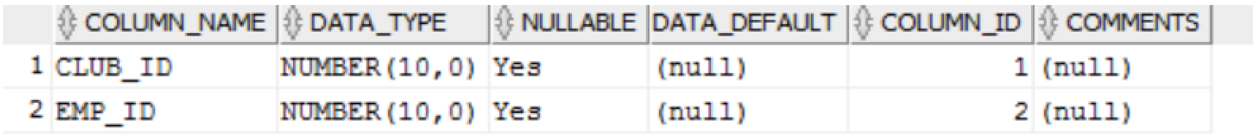
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PID	NUMBER(10,0)	No	(null)	1	(null)
2	PNAME	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	MONTHLY_INCENTIVE	NUMBER(10,0)	No	(null)	3	(null)

CREATE TABLE WELFARE_CLUBS(CLUB_ID NUMBER(10) CONSTRAINT WEL_PK PRIMARY KEY, CLUB_NAME VARCHAR2(30), MONTHLY_SUBSCRIPTION NUMBER(10));



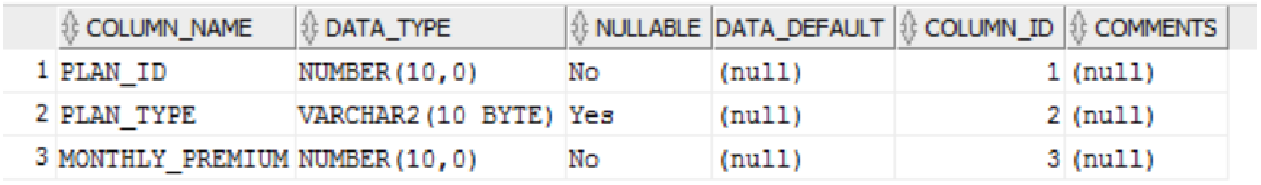
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CLUB_ID	NUMBER(10,0)	No	(null)	1	(null)
2	CLUB_NAME	VARCHAR2(30 BYTE)	Yes	(null)	2	(null)
3	MONTHLY_SUBSCRIPTION	NUMBER(10,0)	Yes	(null)	3	(null)

CREATE TABLE MEMBERSHIP(CLUB_ID NUMBER(10),EMP_ID NUMBER(10),CONSTRAINT MEM_EMP_FK FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID) ON DELETE CASCADE,CONSTRAINT MEM_WEL_FK FOREIGN KEY (CLUB_ID) REFERENCES WELFARE_CLUBS(CLUB_ID) ON DELETE CASCADE);



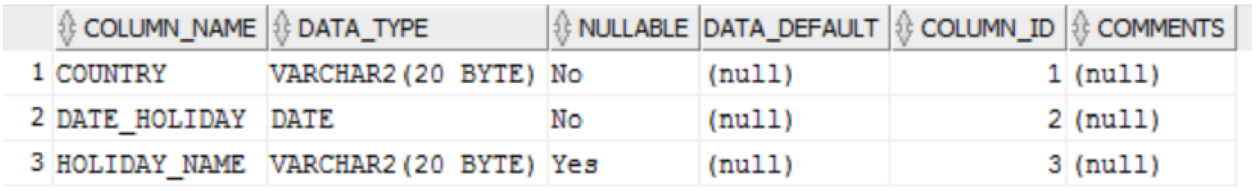
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CLUB_ID	NUMBER(10,0)	Yes	(null)	1	(null)
2	EMP_ID	NUMBER(10,0)	Yes	(null)	2	(null)

CREATE TABLE HEALTHCARE_INSURANCE(PLAN_ID NUMBER(10) CONSTRAINT HEA_PK PRIMARY KEY,PLAN_TYPE VARCHAR2(10), MONTHLY_PREMIUM NUMBER(10) CONSTRAINT HEALTH_NN NOT NULL);



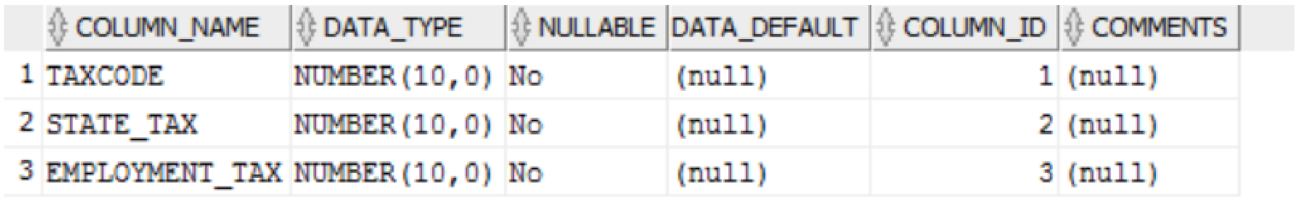
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PLAN_ID	NUMBER(10,0)	No	(null)	1	(null)
2	PLAN_TYPE	VARCHAR2(10 BYTE)	Yes	(null)	2	(null)
3	MONTHLY_PREMIUM	NUMBER(10,0)	No	(null)	3	(null)

CREATE TABLE HOLIDAY_CALENDAR(COUNTRY VARCHAR2(20),DATE_HOLIDAY DATE CONSTRAINT HOL_NN NOT NULL,HOLIDAY_NAME VARCHAR2(20),CONSTRAINT HOL_PK PRIMARY KEY(COUNTRY,DATE_HOLIDAY));



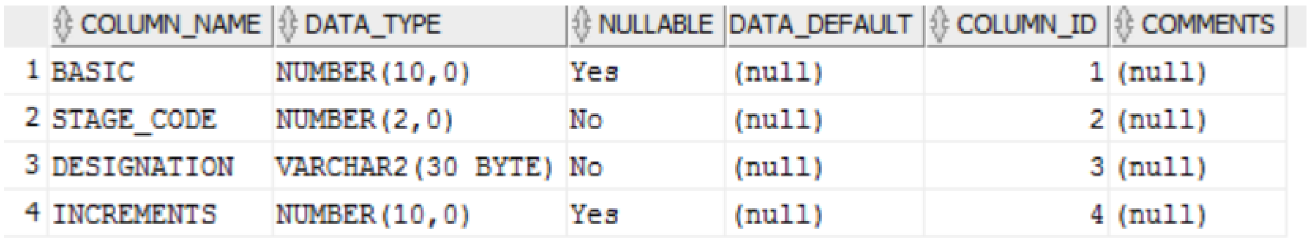
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COUNTRY	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	DATE_HOLIDAY	DATE	No	(null)	2	(null)
3	HOLIDAY_NAME	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)

CREATE TABLE TAX (TAXCODE NUMBER(10) CONSTRAINT TAX_PK PRIMARY KEY, STATE_TAX NUMBER(10) CONSTRAINT TAXS_NN NOT NULL,EMPLOYMENT_TAX NUMBER(10) CONSTRAINT TAXE_NN NOT NULL);



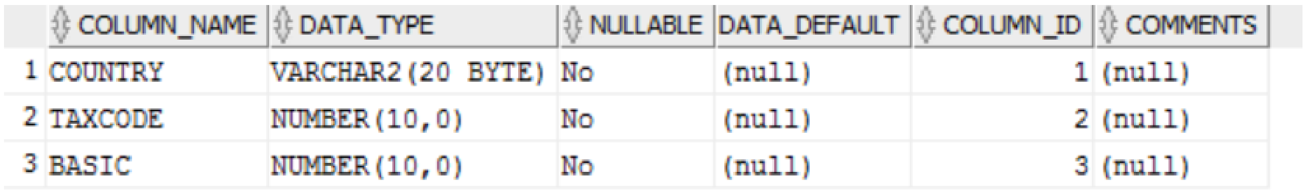
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	TAXCODE	NUMBER(10,0)	No	(null)	1	(null)
2	STATE_TAX	NUMBER(10,0)	No	(null)	2	(null)
3	EMPLOYMENT_TAX	NUMBER(10,0)	No	(null)	3	(null)

CREATE TABLE SALARY_COMPONENTS(BASIC NUMBER(10),STAGE_CODE NUMBER(2),DESIGNATION VARCHAR2(30),INCREMENTS NUMBER(10),CONSTRAINT SAL_PK PRIMARY KEY (STAGE_CODE,DESIGNATION));



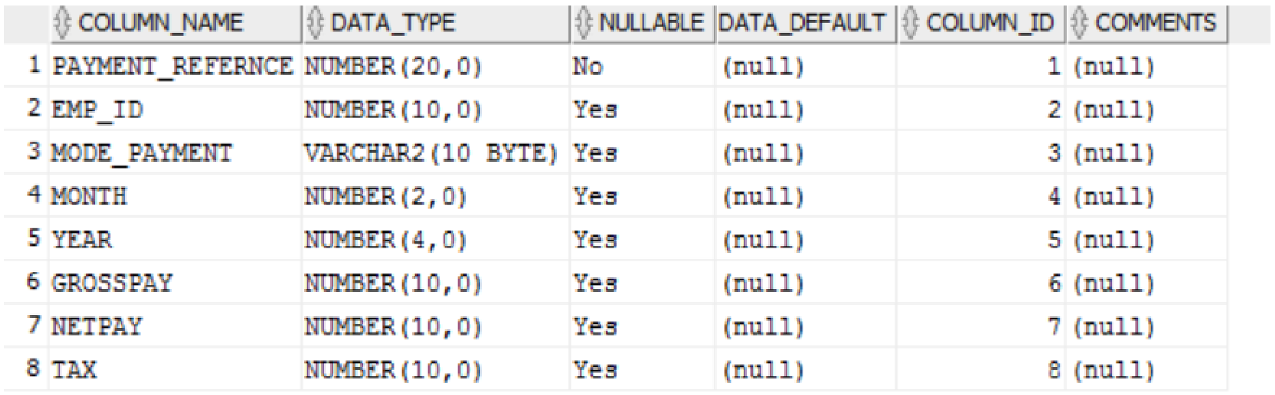
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BASIC	NUMBER(10,0)	Yes	(null)	1	(null)
2	STAGE_CODE	NUMBER(2,0)	No	(null)	2	(null)
3	DESIGNATION	VARCHAR2(30 BYTE)	No	(null)	3	(null)
4	INCREMENTS	NUMBER(10,0)	Yes	(null)	4	(null)

CREATE TABLE TAX_SLAB(COUNTRY VARCHAR(20),TAXCODE NUMBER(10),BASIC NUMBER(10),CONSTRAINT TAXS_PK PRIMARY KEY (TAXCODE,COUNTRY,BASIC),CONSTRAINT TAX_TAXCODE_FK FOREIGN KEY (TAXCODE) REFERENCES TAX(TAXCODE));



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COUNTRY	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	TAXCODE	NUMBER(10,0)	No	(null)	2	(null)
3	BASIC	NUMBER(10,0)	No	(null)	3	(null)

CREATE TABLE PAYROLL_PAYMENTS(PAYMENT_REFERNCE NUMBER(20) CONSTRAINT PAY_PK PRIMARY KEY,EMP_ID NUMBER(10),MODE_PAYMENT VARCHAR2(10),MONTH NUMBER(2),YEAR NUMBER(4),GROSSPAY NUMBER(10),NETPAY NUMBER(10),TAX NUMBER(10),CONSTRAINT PAY_EMP_FK FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID) ON DELETE CASCADE);



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PAYMENT_REFERNCE	NUMBER(20,0)	No	(null)	1	(null)
2	EMP_ID	NUMBER(10,0)	Yes	(null)	2	(null)
3	MODE_PAYMENT	VARCHAR2(10 BYTE)	Yes	(null)	3	(null)
4	MONTH	NUMBER(2,0)	Yes	(null)	4	(null)
5	YEAR	NUMBER(4,0)	Yes	(null)	5	(null)
6	GROSSPAY	NUMBER(10,0)	Yes	(null)	6	(null)
7	NETPAY	NUMBER(10,0)	Yes	(null)	7	(null)
8	TAX	NUMBER(10,0)	Yes	(null)	8	(null)

6.2 INSERT STATEMENTS

```
DESC DEPARTMENT;

INSERT INTO DEPARTMENT VALUES(1,'HR DEPARTMENT');
INSERT INTO DEPARTMENT VALUES(2,'SALES');
INSERT INTO DEPARTMENT VALUES(3,'DEVELOPMENT');
INSERT INTO DEPARTMENT VALUES(4,'TESTING');
INSERT INTO DEPARTMENT VALUES(5,'MAINTENANCE');

DESC PROJECT;

INSERT INTO PROJECT VALUES(1000,'EMPLOYEE RELATIONS',50);
INSERT INTO PROJECT VALUES(1001,'FINNACLE',100);
INSERT INTO PROJECT VALUES(1002,'BANK OF MICHIGAN',175);
INSERT INTO PROJECT VALUES(1003,'NATIONAL OIL CORP',200);
INSERT INTO PROJECT VALUES(1004,'DEV INDUSTRIES',220);
INSERT INTO PROJECT VALUES(1005,'WEBSITE FOR USPS',235);
INSERT INTO PROJECT VALUES(1006,'GENERAL MOTORS',250);
INSERT INTO PROJECT VALUES(1007,'BLUE CROSS ',275);

DESC WELFARE_CLUBS;

INSERT INTO WELFARE_CLUBS VALUES(2001,'Women Empowerment Society',5);
INSERT INTO WELFARE_CLUBS VALUES(2002,'Asian Community',3);
INSERT INTO WELFARE_CLUBS VALUES(2003,'NASCCOM membership',10);
INSERT INTO WELFARE_CLUBS VALUES(2004,'General Thrift Society',10);
INSERT INTO WELFARE_CLUBS VALUES(2005,'General Cooperative Society',15);

DESCRIBE HEALTHCARE_INSURANCE;

INSERT INTO HEALTHCARE_INSURANCE VALUES(9001,'SIMPLE',50);
INSERT INTO HEALTHCARE_INSURANCE VALUES(9002,'SILVER',70);
INSERT INTO HEALTHCARE_INSURANCE VALUES(9003,'GOLD',85);
INSERT INTO HEALTHCARE_INSURANCE VALUES(9004,'PREMIUM',100);
INSERT INTO HEALTHCARE_INSURANCE VALUES(9005,'PLATINUM',125);

DESCRIBE HOLIDAY_CALENDAR;

INSERT INTO HOLIDAY_CALENDAR VALUES('INDIA','14-JAN-2015','HARVESTING FESTIVAL');
INSERT INTO HOLIDAY_CALENDAR VALUES('INDIA','01-MAY-2015','LABOUR DAY');
INSERT INTO HOLIDAY_CALENDAR VALUES('INDIA','26-JAN-2015','REPUBLIC DAY');
INSERT INTO HOLIDAY_CALENDAR VALUES('INDIA','15-AUG-2015','INDEPENDENCE DAY');
INSERT INTO HOLIDAY_CALENDAR VALUES('INDIA','25-DEC-2015','XMAS');
INSERT INTO HOLIDAY_CALENDAR VALUES('USA','01-JAN-2015','NEW YEAR');
INSERT INTO HOLIDAY_CALENDAR VALUES('USA','27-NOV-2015','THANKSGIVING DAY');
INSERT INTO HOLIDAY_CALENDAR VALUES('USA','28-NOV-2015','BLACK FRIDAY');
INSERT INTO HOLIDAY_CALENDAR VALUES('USA','04-JUL-2015','INDEPENDENCE DAY');
INSERT INTO HOLIDAY_CALENDAR VALUES('USA','25-DEC-2015','XMAS');
INSERT INTO HOLIDAY_CALENDAR VALUES('CHINA','01-JAN-2015','NEW YEAR');
INSERT INTO HOLIDAY_CALENDAR VALUES('CHINA','18-FEB-2015','CHINESE NEW YEAR');
INSERT INTO HOLIDAY_CALENDAR VALUES('CHINA','05-APR-2015','QUINGMING FESTIVAL');
INSERT INTO HOLIDAY_CALENDAR VALUES('CHINA','01-MAY-2015','MAY DAY');
INSERT INTO HOLIDAY_CALENDAR VALUES('CHINA','20-JUN-2015','DRAGON BOAT FESTIVAL');

DESC TAX;

INSERT INTO TAX VALUES(10000,10,2);
INSERT INTO TAX VALUES(10001,20,5);
INSERT INTO TAX VALUES(10002,30,7);
INSERT INTO TAX VALUES(10003,35,11);
INSERT INTO TAX VALUES(10004,10,1);
INSERT INTO TAX VALUES(10005,20,4);
INSERT INTO TAX VALUES(10006,30,8);
INSERT INTO TAX VALUES(10007,5,3);
INSERT INTO TAX VALUES(10008,15,7);
INSERT INTO TAX VALUES(10009,20,11);
INSERT INTO TAX VALUES(10010,20,15);

DESCRIBE SALARY_COMPONENTS;

INSERT INTO SALARY_COMPONENTS VALUES(2500,1,'Associate',500);
INSERT INTO SALARY_COMPONENTS VALUES(3000,2,'Associate',500);
INSERT INTO SALARY_COMPONENTS VALUES(3500,3,'Associate',500);
INSERT INTO SALARY_COMPONENTS VALUES(3100,1,'Software Engineer',550);
INSERT INTO SALARY_COMPONENTS VALUES(3650,2,'Software Engineer',550);
INSERT INTO SALARY_COMPONENTS VALUES(4200,3,'Software Engineer',550);
INSERT INTO SALARY_COMPONENTS VALUES(4200,1,'Senior Software Engineer',700);
INSERT INTO SALARY_COMPONENTS VALUES(4900,2,'Senior Software Engineer',700);
INSERT INTO SALARY_COMPONENTS VALUES(5450,1,'Technical Lead',850);
INSERT INTO SALARY_COMPONENTS VALUES(6300,2,'Technical Lead',850);
INSERT INTO SALARY_COMPONENTS VALUES(6000,1,'Project Manager',900);
INSERT INTO SALARY_COMPONENTS VALUES(6900,2,'Project Manager',900);
INSERT INTO SALARY_COMPONENTS VALUES(7800,3,'Project Manager',900);
INSERT INTO SALARY_COMPONENTS VALUES(6950,1,'Senior Project Lead',1050);
INSERT INTO SALARY_COMPONENTS VALUES(8000,2,'Senior Project Lead',1050);
INSERT INTO SALARY_COMPONENTS VALUES(7590,1,'Associate Director',1300);
INSERT INTO SALARY_COMPONENTS VALUES(8640,2,'Associate Director',1300);
INSERT INTO SALARY_COMPONENTS VALUES(8510,1,'Director',1550);
INSERT INTO SALARY_COMPONENTS VALUES(10060,2,'Director',1550);
INSERT INTO SALARY_COMPONENTS VALUES(9520,1,'Associate Vice President',1900);
INSERT INTO SALARY_COMPONENTS VALUES(11420,2,'Associate Vice President',1900);
INSERT INTO SALARY_COMPONENTS VALUES(10550,1,'Vice President',2350);
INSERT INTO SALARY_COMPONENTS VALUES(12900,2,'Vice President',2350);
INSERT INTO SALARY_COMPONENTS VALUES(12700,1,'Chief Executive Officer',2350);


INSERT INTO TAX_SLAB VALUES('USA',10001,6000);
INSERT INTO TAX_SLAB VALUES('USA',10002,10060);
INSERT INTO TAX_SLAB VALUES('INDIA',10005,4200);
INSERT INTO TAX_SLAB VALUES('INDIA',10006,8000);
INSERT INTO TAX_SLAB VALUES('CHINA',10008,6000);
INSERT INTO TAX_SLAB VALUES('CHINA',10010,8510);
INSERT INTO TAX_SLAB VALUES('INDIA',10001,2500);

DESC EMPLOYEE;

INSERT INTO EMPLOYEE VALUES(5001,'Aganya','K','Pari','01-Jan-2014','06-May-1990','Y','Mahas
Avenue',9940058794,'aganya@itco.com',123356789,'B+','Active',3,9002,'India',1,'Project Manager');
INSERT INTO EMPLOYEE VALUES(5002,'Ankita','S','Agarwal','05-Oct-2012','14-Nov-1989','N','AHD
Towns',8763304311,'ankita@itco.com',678912345,'A+','Active',1,9001,'India',2,'Senior Project Lead');
INSERT INTO EMPLOYEE VALUES(5003,'Pooja','S','Junnarkar','11-Aug-2015','07-Feb-
1993','Y','Marquis',9123058344,'pooja@itco.com',222222222,'0-','Active',2,9003,'China',1,'Project Manager');
INSERT INTO EMPLOYEE VALUES(5004,'Mahak','K','Khattar','11-Aug-2011','26-May-
1988','Y','Vrindavan',9982738773,'mahak@itco.com',333333333,'A-','Active',2,9004,'China',1,'Director');
INSERT INTO EMPLOYEE VALUES(5005,'Dinesh','M','Ganesan','10-Oct-2012','13-Jul-1990','Y','VAK
Homes',5940058329,'dinesh@itco.com',444444444,'A+','Active',4,9004,'India',1,'Associate');
INSERT INTO EMPLOYEE VALUES(5006,'Udaya','KB','Shankar','22-Aug-2014','08-Jan-1990','Y','Velvet
Flowers',4440058323,'udaya@itco.com',555555555,'O+','Active',5,9001,'India',3,'Software Engineer');
INSERT INTO EMPLOYEE VALUES(5007,'Santosh','S','Sekar','11-Nov-2013','31-May-
1989','Y','TCSLD',5109333039,'sugar@itco.com',666666666,'B-','Active',5,9003,'USA',2,'Director');
INSERT INTO EMPLOYEE VALUES(5008,'Aastha','D','Dixit','06-Aug-2015','02-Sep-
1989','N','Mccallum',7483901938,'aastha@itco.com',888888888,'B+','Active',1,9002,'USA',2,'Director');
INSERT INTO EMPLOYEE VALUES(5009,'Keerthi','B','Sundram','06-Aug-2015','07-Mar-
1990','N','Parkside',5109233839,'kee@itco.com',777777777,'B+','Active',1,9002,'USA',2,'Director');


INSERT INTO LEAVES VALUES (5001,15,15,30);
INSERT INTO LEAVES VALUES (5002,20,20,40);
INSERT INTO LEAVES VALUES (5003,11,12,21);
INSERT INTO LEAVES VALUES (5004,9,11,23);
INSERT INTO LEAVES VALUES (5005,16,12,36);
INSERT INTO LEAVES VALUES (5006,23,24,36);
INSERT INTO LEAVES VALUES (5007,18,19,37);
INSERT INTO LEAVES VALUES (5008,24,27,40);
INSERT INTO LEAVES VALUES (5009,13,12,18);

INSERT INTO SOFT_LOANS VALUES (1,5001,3100,2900,'01-DEC-2015','01-MAR-2016');
INSERT INTO SOFT_LOANS VALUES (2,5002,3000,2200,'05-DEC-2015','22-FEB-2016');
INSERT INTO SOFT_LOANS VALUES (3,5003,3700,3600,'09-DEC-2015','04-APR-2016');
INSERT INTO SOFT_LOANS VALUES (4,5006,3900,3700,'05-DEC-2015','20-APR-2016');
INSERT INTO SOFT_LOANS VALUES (5,5007,4200,4000,'04-DEC-2015','01-MAY-2016');

INSERT INTO ADVANCE_LOANS VALUES (1,5002,2015,05,'23-MAY-2015',25071);
INSERT INTO ADVANCE_LOANS VALUES (2,5004,2015,10,'18-OCT-2015',30835);
INSERT INTO ADVANCE_LOANS VALUES (3,5008,2015,09,'21-SEP-2015',35730);
INSERT INTO ADVANCE_LOANS VALUES (4,5005,2015,08,'28-AUG-2015',41930);
INSERT INTO ADVANCE_LOANS VALUES (5,5009,2015,11,'07-NOV-2015',41930);

INSERT INTO WORKS_ON VALUES (1000,5001);
INSERT INTO WORKS_ON VALUES (1001,5002);
INSERT INTO WORKS_ON VALUES (1002,5003);
INSERT INTO WORKS_ON VALUES (1003,5004);
INSERT INTO WORKS_ON VALUES (1004,5005);
INSERT INTO WORKS_ON VALUES (1004,5006);
INSERT INTO WORKS_ON VALUES (1005,5007);
INSERT INTO WORKS_ON VALUES (1006,5008);
INSERT INTO WORKS_ON VALUES (1007,5009);

INSERT INTO RETIREMENT_ACCOUNT VALUES (5003, 105002, 179200, 195000);
INSERT INTO RETIREMENT_ACCOUNT VALUES (5004, 250005, 230200, 245000);
INSERT INTO RETIREMENT_ACCOUNT VALUES (5005, 293800, 269200, 274600);
INSERT INTO RETIREMENT_ACCOUNT VALUES (5006, 190820, 208250, 213900);
INSERT INTO RETIREMENT_ACCOUNT VALUES (5007, 310700, 252020, 269800);

INSERT INTO MEMBERSHIP VALUES (2001,5001);
INSERT INTO MEMBERSHIP VALUES (2001,5002);
INSERT INTO MEMBERSHIP VALUES (2002,5003);
INSERT INTO MEMBERSHIP VALUES (2003,5004);
INSERT INTO MEMBERSHIP VALUES (2004,5005);
INSERT INTO MEMBERSHIP VALUES (2005,5006);
INSERT INTO MEMBERSHIP VALUES (2005,5007);
```

Note : The payroll_Payments table will be populated on execution of the stored procedure payroll_RUN

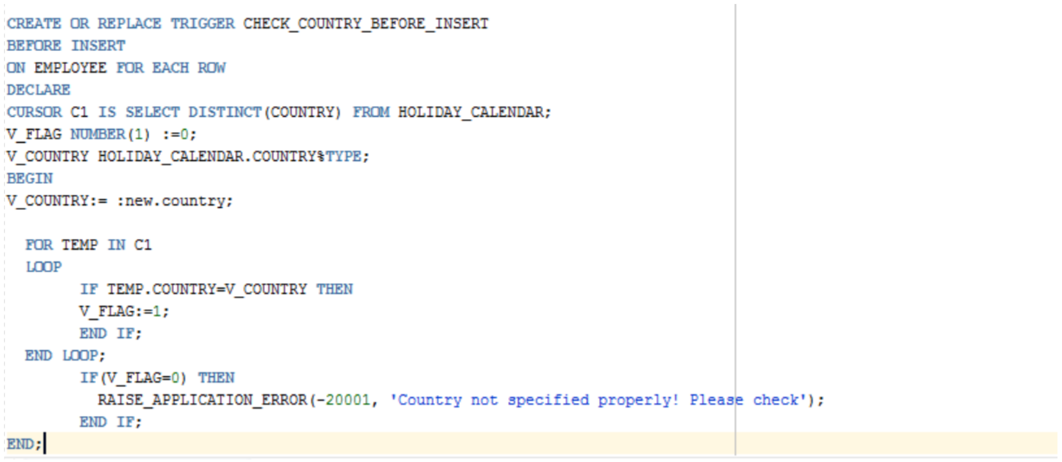
6.3 OTHER STATEMENTS

SEQUENCE:

```
CREATE SEQUENCE LOANID_SEQ
  START WITH 5000
  INCREMENT BY 1
  CACHE 100;

CREATE SEQUENCE LOANID_SEQ
  START WITH 5000
  INCREMENT BY 1
  CACHE 100;

CREATE SEQUENCE ADVANCE_SEQ
  START WITH 15000
  INCREMENT BY 1
  CACHE 100;

CREATE SEQUENCE PAYMENT_SEQ
  START WITH 230000
  INCREMENT BY 5  CACHE 100;
```

TRIGGER:

```
CREATE OR REPLACE TRIGGER CHECK_COUNTRY_BEFORE_INSERT
BEFORE INSERT
ON EMPLOYEE for each row
DECLARE
CURSOR C1 IS SELECT DISTINCT(COUNTRY) FROM HOLIDAY_CALENDAR;
V_FLAG NUMBER(1) :=0;
V_COUNTRY HOLIDAY_CALENDAR.COUNTRY%TYPE;
BEGIN
V_COUNTRY:= :new.country;
FOR TEMP IN C1
LOOP
IF TEMP.COUNTRY=V_COUNTRY THEN
V_FLAG:=1;
END IF;
END LOOP;

IF(V_FLAG=0) THEN
RAISE_APPLICATION_ERROR(-20001, 'Country not specified properly! Please check');
END IF;
END;
```


```
CREATE OR REPLACE TRIGGER CHECK_COUNTRY_BEFORE_INSERT
BEFORE INSERT
ON EMPLOYEE FOR EACH ROW
DECLARE
CURSOR C1 IS SELECT DISTINCT(COUNTRY) FROM HOLIDAY_CALENDAR;
V_FLAG NUMBER(1) :=0;
V_COUNTRY HOLIDAY_CALENDAR.COUNTRY%TYPE;
BEGIN
V_COUNTRY:= :new.country;

  FOR TEMP IN C1
  LOOP
        IF TEMP.COUNTRY=V_COUNTRY THEN
        V_FLAG:=1;
        END IF;
  END LOOP;
        IF(V_FLAG=0) THEN
          RAISE_APPLICATION_ERROR(-20001, 'Country not specified properly! Please check');
        END IF;
END;
```

INDEX:

CREATE INDEX payroll_reports
  ON payroll_payments (gross_pay,netpay);

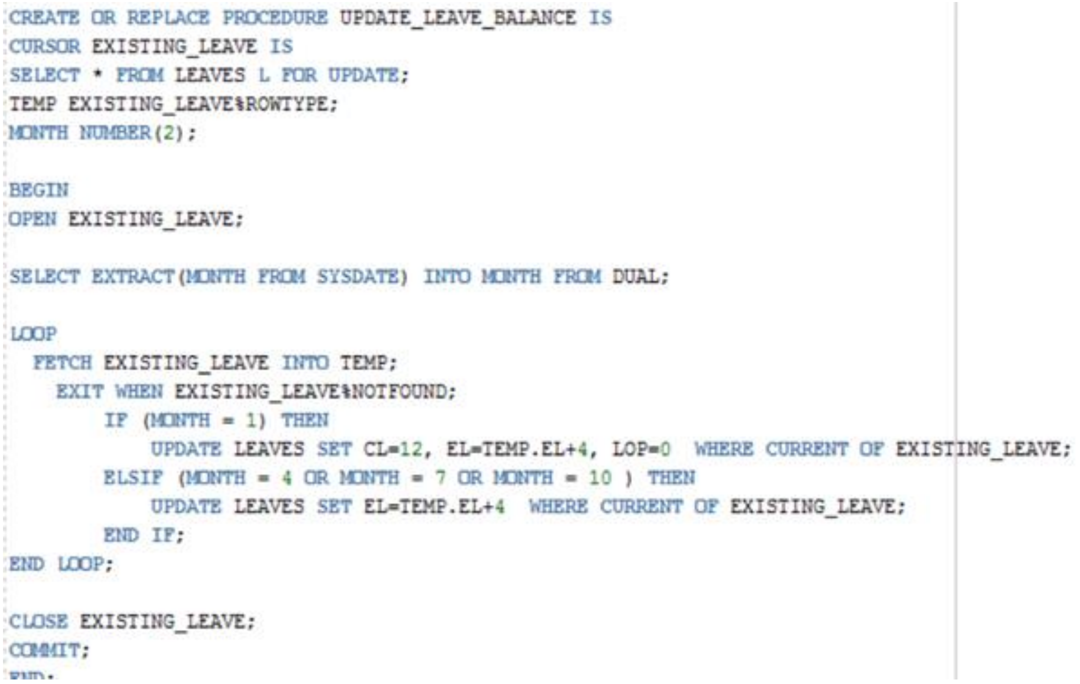
7 PL/SQL SECTION

1. **PROCEDURE NAME:** UPDATE_LEAVE_BALANCE

DESCRIPTION OF THE PROCEDURE: THIS PROCEDURE WILL BE RUN BY THE ADMINSTRATOR DURING THE START OF EVERY QUARTER TO UPDATE THE LEAVE BALANCE OF ALL THE EMPLOYEES ACCORDING TO THE BUSINESS CRITERIA.

RUN FREQUENCY: EVERY QUARTER (MONTHS 1,4,7,10)

```
CREATE OR REPLACE PROCEDURE UPDATE_LEAVE_BALANCE IS
CURSOR EXISTING_LEAVE IS
SELECT * FROM LEAVES L FOR UPDATE;
TEMP EXISTING_LEAVE%ROWTYPE;
MONTH NUMBER(2);

BEGIN
OPEN EXISTING_LEAVE;

SELECT EXTRACT(MONTH FROM SYSDATE) INTO MONTH FROM DUAL;

LOOP
  FETCH EXISTING_LEAVE INTO TEMP;
    EXIT WHEN EXISTING_LEAVE%NOTFOUND;
        IF (MONTH = 1) THEN
            UPDATE LEAVES SET CL=12, EL=TEMP.EL+4, LOP=0  WHERE CURRENT OF EXISTING_LEAVE;
        ELSIF (MONTH = 4 OR MONTH = 7 OR MONTH = 10 ) THEN
            UPDATE LEAVES SET EL=TEMP.EL+4  WHERE CURRENT OF EXISTING_LEAVE;
        END IF;
END LOOP;

CLOSE EXISTING_LEAVE;
COMMIT;
END;
```



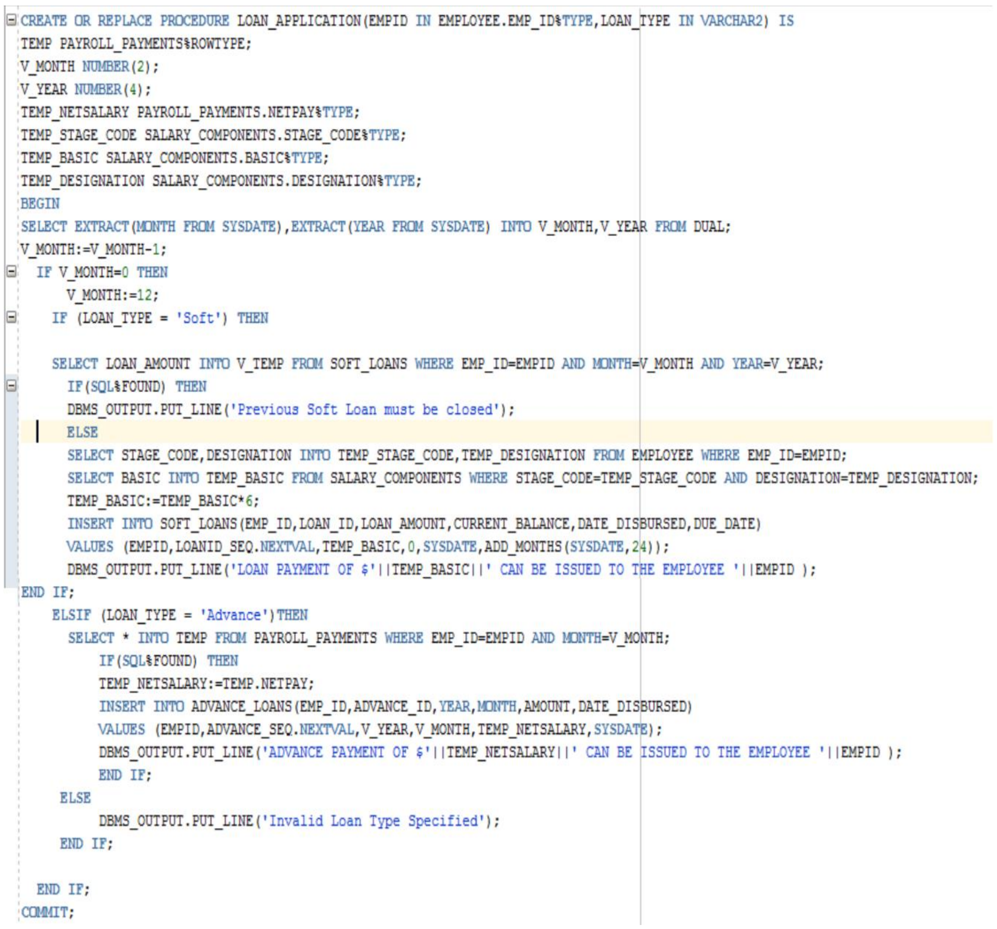
```
CREATE OR REPLACE PROCEDURE UPDATE_LEAVE_BALANCE IS
CURSOR EXISTING_LEAVE IS
SELECT * FROM LEAVES L FOR UPDATE;
TEMP EXISTING_LEAVE%ROWTYPE;
MONTH NUMBER(2);

BEGIN
OPEN EXISTING_LEAVE;

SELECT EXTRACT(MONTH FROM SYSDATE) INTO MONTH FROM DUAL;

LOOP
  FETCH EXISTING_LEAVE INTO TEMP;
    EXIT WHEN EXISTING_LEAVE%NOTFOUND;
        IF (MONTH = 1) THEN
            UPDATE LEAVES SET CL=12, EL=TEMP.EL+4, LOP=0  WHERE CURRENT OF EXISTING_LEAVE;
        ELSIF (MONTH = 4 OR MONTH = 7 OR MONTH = 10 ) THEN
            UPDATE LEAVES SET EL=TEMP.EL+4  WHERE CURRENT OF EXISTING_LEAVE;
        END IF;
END LOOP;

CLOSE EXISTING_LEAVE;
COMMIT;
```

2. **PROCEDURE NAME:** LOAN_APPLICATION

DESCRIPTION OF THE PROCEDURE: THIS PROCEDURE WILL BE RUN BY THE ADMINSTRATOR WHEN AN EMPLOYEE APPLIES FOR A SOFT LOAN OR AN ADVANCE PAYMENT AND THE RULES ARE SET ACCORDING TO THE BUSINESS CRITERIA.

RUN FREQUENCY: AS AN WHEN THE EMPLOYEE REQUESTS

```
CREATE OR REPLACE PROCEDURE LOAN_APPLICATION(EMPID IN EMPLOYEE.EMP_ID%TYPE,LOAN_TYPE IN VARCHAR2) IS
TEMP PAYROLL_PAYMENTS%ROWTYPE;
V_MONTH NUMBER(2);
V_YEAR NUMBER(4);
TEMP_NETSALARY PAYROLL_PAYMENTS.NETPAY%TYPE;
TEMP_STAGE_CODE SALARY_COMPONENTS.STAGE_CODE%TYPE;
TEMP_BASIC SALARY_COMPONENTS.BASIC%TYPE;
TEMP_DESIGNATION SALARY_COMPONENTS.DESIGNATION%TYPE;
BEGIN
SELECT EXTRACT(MONTH FROM SYSDATE),EXTRACT(YEAR FROM SYSDATE) INTO V_MONTH,V_YEAR FROM DUAL;
V_MONTH:=V_MONTH-1;
  IF V_MONTH=0 THEN
     V_MONTH:=12;
    IF (LOAN_TYPE = 'Soft') THEN

    SELECT LOAN_AMOUNT INTO V_TEMP FROM SOFT_LOANS WHERE EMP_ID=EMPID AND MONTH=V_MONTH AND YEAR=V_YEAR;
     IF(SQL%FOUND) THEN
     DBMS_OUTPUT.PUT_LINE('Previous Soft Loan must be closed');
     ELSE
     SELECT STAGE_CODE,DESIGNATION INTO TEMP_STAGE_CODE,TEMP_DESIGNATION FROM EMPLOYEE WHERE EMP_ID=EMPID;
     SELECT BASIC INTO TEMP_BASIC FROM SALARY_COMPONENTS WHERE STAGE_CODE=TEMP_STAGE_CODE AND
DESIGNATION=TEMP_DESIGNATION;
     TEMP_BASIC:=TEMP_BASIC*6;
     INSERT INTO SOFT_LOANS(EMP_ID,LOAN_ID,LOAN_AMOUNT,CURRENT_BALANCE,DATE_DISBURSED,DUE_DATE)
     VALUES (EMPID,LOANID_SEQ.NEXTVAL,TEMP_BASIC,0,SYSDATE,ADD_MONTHS(SYSDATE,24));
     DBMS_OUTPUT.PUT_LINE('LOAN PAYMENT OF $'||TEMP_BASIC||' CAN BE ISSUED TO THE EMPLOYEE '||EMPID );
END IF;
    ELSIF (LOAN_TYPE = 'Advance')THEN
     SELECT * INTO TEMP FROM PAYROLL_PAYMENTS WHERE EMP_ID=EMPID AND MONTH=V_MONTH;
        IF(SQL%FOUND) THEN
        TEMP_NETSALARY:=TEMP.NETPAY;
        INSERT INTO ADVANCE_LOANS(EMP_ID,ADVANCE_ID,YEAR,MONTH,AMOUNT,DATE_DISBURSED)
        VALUES (EMPID,ADVANCE_SEQ.NEXTVAL,V_YEAR,V_MONTH,TEMP_NETSALARY,SYSDATE);
        DBMS_OUTPUT.PUT_LINE('ADVANCE PAYMENT OF $'||TEMP_NETSALARY||' CAN BE ISSUED TO THE EMPLOYEE '||EMPID );
        END IF;
    ELSE
        DBMS_OUTPUT.PUT_LINE('Invalid Loan Type Specified');
    END IF;

  END IF;
COMMIT;
END;
```



```
CREATE OR REPLACE PROCEDURE LOAN_APPLICATION(EMPID IN EMPLOYEE.EMP_ID%TYPE,LOAN_TYPE IN VARCHAR2) IS
TEMP PAYROLL_PAYMENTS%ROWTYPE;
V_MONTH NUMBER(2);
V_YEAR NUMBER(4);
TEMP_NETSALARY PAYROLL_PAYMENTS.NETPAY%TYPE;
TEMP_STAGE_CODE SALARY_COMPONENTS.STAGE_CODE%TYPE;
TEMP_BASIC SALARY_COMPONENTS.BASIC%TYPE;
TEMP_DESIGNATION SALARY_COMPONENTS.DESIGNATION%TYPE;
BEGIN
SELECT EXTRACT(MONTH FROM SYSDATE),EXTRACT(YEAR FROM SYSDATE) INTO V_MONTH,V_YEAR FROM DUAL;
V_MONTH:=V_MONTH-1;
  IF V_MONTH=0 THEN
      V_MONTH:=12;
    IF (LOAN_TYPE = 'Soft') THEN

    SELECT LOAN_AMOUNT INTO V_TEMP FROM SOFT_LOANS WHERE EMP_ID=EMPID AND MONTH=V_MONTH AND YEAR=V_YEAR;
      IF(SQL%FOUND) THEN
      DBMS_OUTPUT.PUT_LINE('Previous Soft Loan must be closed');
      ELSE
      SELECT STAGE_CODE,DESIGNATION INTO TEMP_STAGE_CODE,TEMP_DESIGNATION FROM EMPLOYEE WHERE EMP_ID=EMPID;
      SELECT BASIC INTO TEMP_BASIC FROM SALARY_COMPONENTS WHERE STAGE_CODE=TEMP_STAGE_CODE AND DESIGNATION=TEMP_DESIGNATION;
      TEMP_BASIC:=TEMP_BASIC*6;
      INSERT INTO SOFT_LOANS(EMP_ID,LOAN_ID,LOAN_AMOUNT,CURRENT_BALANCE,DATE_DISBURSED,DUE_DATE)
      VALUES (EMPID,LOANID_SEQ.NEXTVAL,TEMP_BASIC,0,SYSDATE,ADD_MONTHS(SYSDATE,24));
      DBMS_OUTPUT.PUT_LINE('LOAN PAYMENT OF $'||TEMP_BASIC||' CAN BE ISSUED TO THE EMPLOYEE '||EMPID );
END IF;
    ELSIF (LOAN_TYPE = 'Advance')THEN
      SELECT * INTO TEMP FROM PAYROLL_PAYMENTS WHERE EMP_ID=EMPID AND MONTH=V_MONTH;
          IF(SQL%FOUND) THEN
          TEMP_NETSALARY:=TEMP.NETPAY;
          INSERT INTO ADVANCE_LOANS(EMP_ID,ADVANCE_ID,YEAR,MONTH,AMOUNT,DATE_DISBURSED)
          VALUES (EMPID,ADVANCE_SEQ.NEXTVAL,V_YEAR,V_MONTH,TEMP_NETSALARY,SYSDATE);
          DBMS_OUTPUT.PUT_LINE('ADVANCE PAYMENT OF $'||TEMP_NETSALARY||' CAN BE ISSUED TO THE EMPLOYEE '||EMPID );
          END IF;
     ELSE
          DBMS_OUTPUT.PUT_LINE('Invalid Loan Type Specified');
     END IF;

  END IF;
COMMIT;
```

3. **PROCEDURE NAME:** PAYROLL_RUN

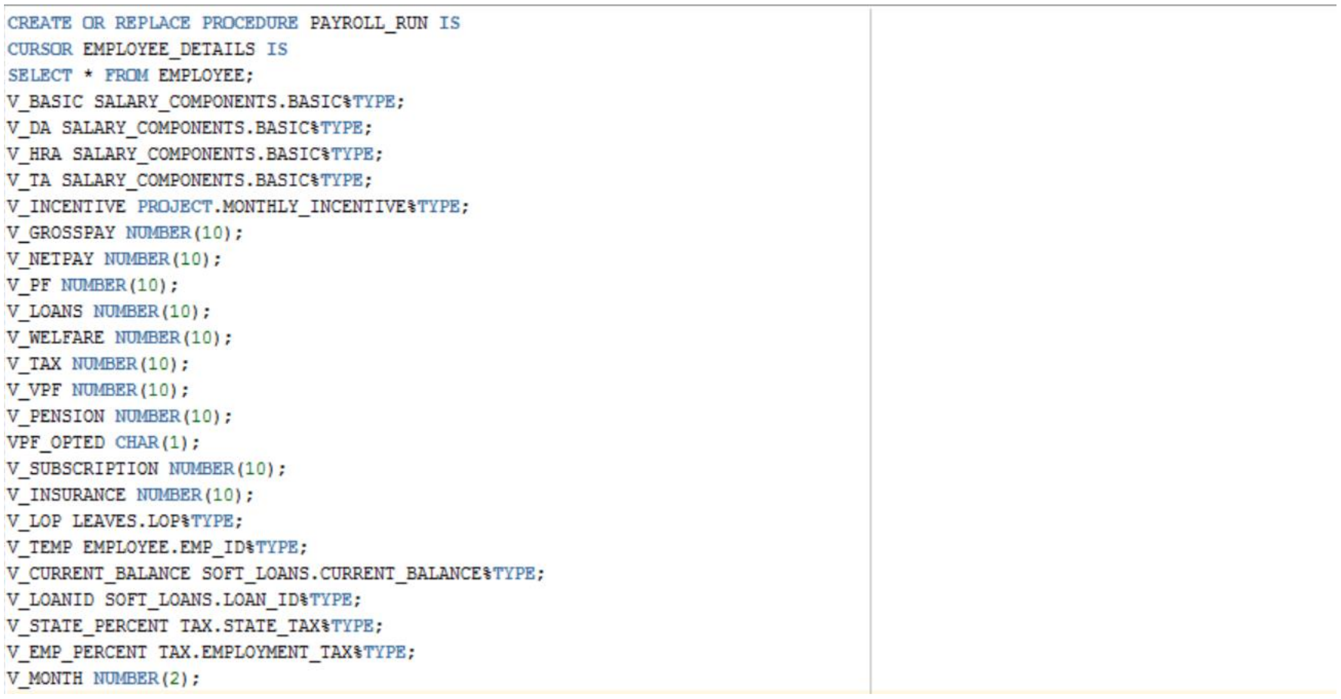
DESCRIPTION OF THE PROCEDURE: THIS PROCEDURE WILL BE RUN BY THE ADMINSTRATOR EVERY MONTH TO PROCESS THE PAYROLL OF ALL THE ACTIVE EMPLOYEES IN THE COMPANY. THE PROCEDURE TAKES CARE OF THE LOANS, WELFARE MEMBERSHIP, TAX, EARNINGS, AND HEALTHCARE INSURANCE. IT COMPUTES THE GROSS PAY, AND NET PAY AFTER DEDUCTIONS.

RUN FREQUENCY: EVERY MONTH END

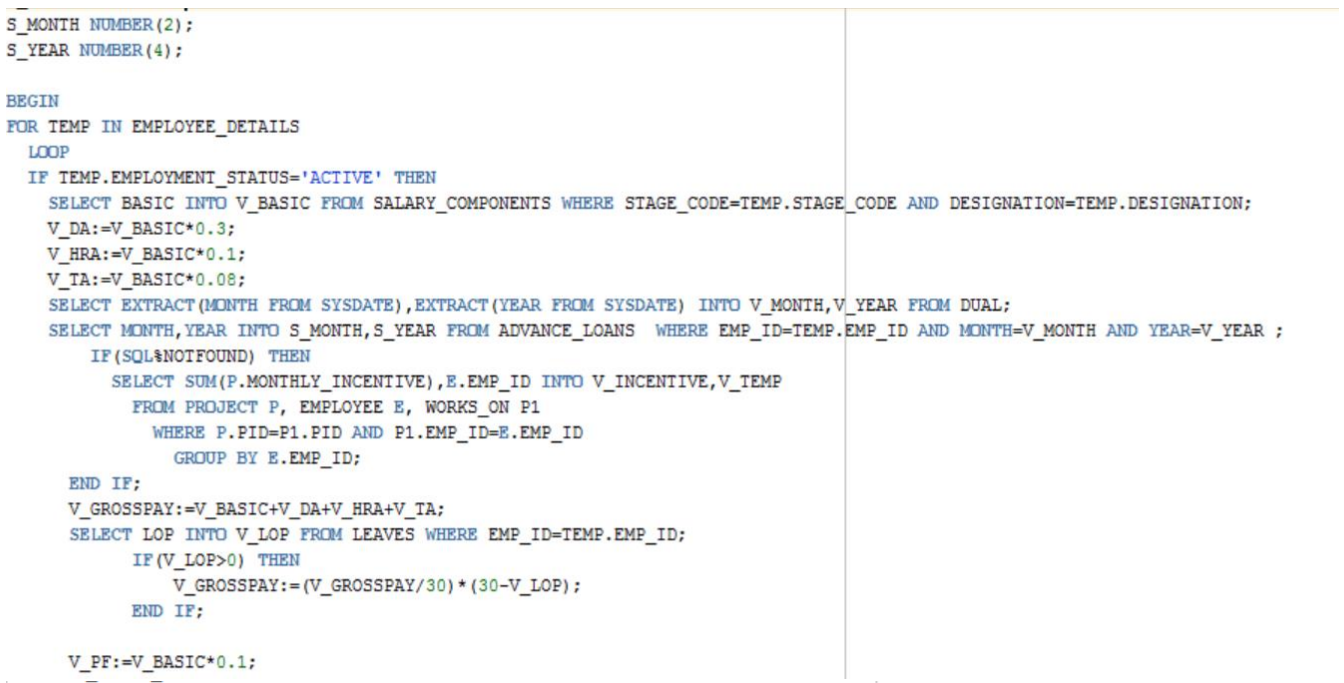
```
CREATE OR REPLACE PROCEDURE PAYROLL_RUN IS
CURSOR EMPLOYEE_DETAILS IS
SELECT * FROM EMPLOYEE;
V_BASIC SALARY_COMPONENTS.BASIC%TYPE;
V_DA SALARY_COMPONENTS.BASIC%TYPE;
V_HRA SALARY_COMPONENTS.BASIC%TYPE;
V_TA SALARY_COMPONENTS.BASIC%TYPE;
V_INCENTIVE PROJECT.MONTHLY_INCENTIVE%TYPE;
V_GROSSPAY NUMBER(10);
V_NETPAY NUMBER(10);
V_PF NUMBER(10);
V_LOANS NUMBER(10);
V_WELFARE NUMBER(10);
V_TAX NUMBER(10);
V_VPF NUMBER(10);
V_PENSION NUMBER(10);
VPF_OPTED CHAR(1);
V_SUBSCRIPTION NUMBER(10);
V_INSURANCE NUMBER(10);
V_LOP LEAVES.LOP%TYPE;
V_TEMP EMPLOYEE.EMP_ID%TYPE;
V_CURRENT_BALANCE SOFT_LOANS.CURRENT_BALANCE%TYPE;
V_LOANID SOFT_LOANS.LOAN_ID%TYPE;
V_STATE_PERCENT TAX.STATE_TAX%TYPE;
V_EMP_PERCENT TAX.EMPLOYMENT_TAX%TYPE;
V_MONTH NUMBER(2);
V_YEAR NUMBER(4);
S_MONTH NUMBER(2);
S_YEAR NUMBER(4);

BEGIN
FOR TEMP IN EMPLOYEE_DETAILS
  LOOP
  IF TEMP.EMPLOYMENT_STATUS='ACTIVE' THEN
    SELECT BASIC INTO V_BASIC FROM SALARY_COMPONENTS WHERE STAGE_CODE=TEMP.STAGE_CODE AND
DESIGNATION=TEMP.DESIGNATION;
    V_DA:=V_BASIC*0.3;
    V_HRA:=V_BASIC*0.1;
    V_TA:=V_BASIC*0.08;
    SELECT EXTRACT(MONTH FROM SYSDATE),EXTRACT(YEAR FROM SYSDATE) INTO V_MONTH,V_YEAR FROM DUAL;
    SELECT MONTH,YEAR INTO S_MONTH,S_YEAR FROM ADVANCE_LOANS  WHERE EMP_ID=TEMP.EMP_ID AND MONTH=V_MONTH AND
YEAR=V_YEAR ;
```

```
    IF(SQL%NOTFOUND) THEN
     SELECT SUM(P.MONTHLY_INCENTIVE),E.EMP_ID INTO V_INCENTIVE,V_TEMP
       FROM PROJECT P, EMPLOYEE E, WORKS_ON P1
         WHERE P.PID=P1.PID AND P1.EMP_ID=E.EMP_ID
           GROUP BY E.EMP_ID;
   END IF;
   V_GROSSPAY:=V_BASIC+V_DA+V_HRA+V_TA;
   SELECT LOP INTO V_LOP FROM LEAVES WHERE EMP_ID=TEMP.EMP_ID;
       IF(V_LOP>0) THEN
           V_GROSSPAY:=(V_GROSSPAY/30)*(30-V_LOP);
       END IF;

   V_PF:=V_BASIC*0.1;
   V_PENSION:=V_PF;

       IF(VPF_OPTED='Y') THEN
           V_VPF:=V_BASIC*0.075;
       ELSE
           V_VPF:=0;
       END IF;

   UPDATE RETIREMENT_ACCOUNT
     SET PF_BALANCE=PF_BALANCE+V_PF,VPF_BALANCE=VPF_BALANCE+V_VPF,PENSION_BALANCE=PENSION_BALANCE+V_PENSION
       WHERE EMP_ID=TEMP.EMP_ID;

   SELECT LOAN_ID,CURRENT_BALANCE,LOAN_AMOUNT INTO V_LOANID,V_CURRENT_BALANCE,V_LOANS
     FROM SOFT_LOANS
       WHERE EMP_ID=TEMP.EMP_ID AND CURRENT_BALANCE!=0;

   IF(SQL%FOUND) THEN
     V_LOANS:=V_LOANS/24;
     V_CURRENT_BALANCE:=V_CURRENT_BALANCE-V_LOANS;
     UPDATE SOFT_LOANS SET CURRENT_BALANCE=V_CURRENT_BALANCE WHERE LOAN_ID=V_LOANID;
       IF(V_CURRENT_BALANCE=0) THEN
         UPDATE SOFT_LOANS SET DUE_DATE=SYSDATE WHERE LOAN_ID=V_LOANID;
       END IF;
END IF;

   SELECT SUM(P.MONTHLY_SUBSCRIPTION),E.EMP_ID INTO V_SUBSCRIPTION,V_TEMP
     FROM WELFARE_CLUBS P, EMPLOYEE E, MEMBERSHIP P1
       WHERE P.CLUB_ID=P1.CLUB_ID AND P1.EMP_ID=E.EMP_ID
         GROUP BY E.EMP_ID;

   SELECT MONTHLY_PREMIUM INTO V_INSURANCE FROM HEALTHCARE_INSURANCE H,EMPLOYEE E
   WHERE H.PLAN_ID=TEMP.PLAN_ID AND E.EMP_ID=TEMP.EMP_ID;
       IF(SQL%NOTFOUND) THEN
         V_INSURANCE:=0;
       END IF;

   SELECT T.STATE_TAX, T.EMPLOYMENT_TAX INTO V_STATE_PERCENT,V_EMP_PERCENT
```

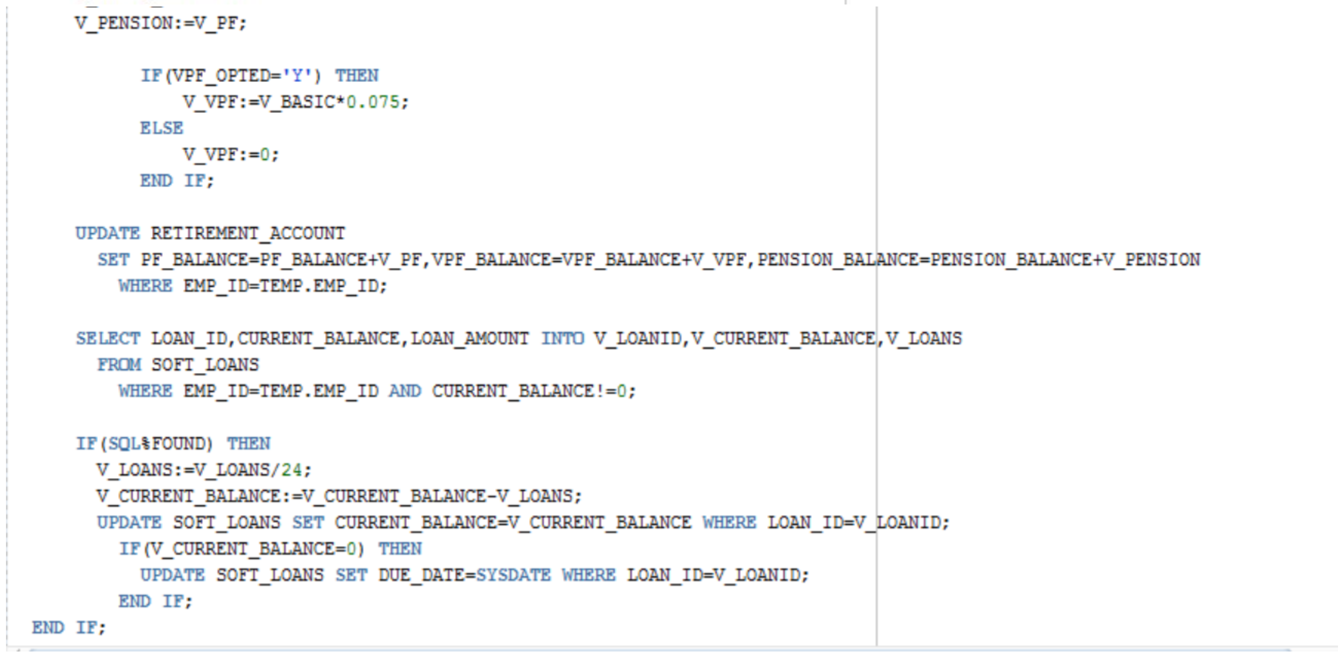
```
    FROM TAX T, SALARY_COMPONENTS S, TAX_SLAB T1, HOLIDAY_CALENDAR C, EMPLOYEE E
      WHERE E.STAGE_CODE=S.STAGE_CODE AND E.DESIGNATION=S.DESIGNATION AND T.TAXCODE=T1.TAXCODE
      AND C.COUNTRY=E.COUNTRY AND E.EMP_ID=TEMP.EMP_ID;

   V_TAX:=(V_STATE_PERCENT/100*V_GROSSPAY)+(V_EMP_PERCENT/100*V_GROSSPAY);

   V_NETPAY:=V_GROSSPAY-V_PF-V_VPF-V_SUBSCRIPTION-V_LOANS-V_TAX-V_INSURANCE;

   IF (V_NETPAY<0) THEN
    V_NETPAY:=0;
   END IF;

   INSERT INTO PAYROLL_PAYMENTS(PAYMENT_REFERNCE,MONTH,YEAR,NETPAY,TAX,GROSSPAY,MODE_PAYMENT,EMP_ID)
    VALUES(PAYMENT_SEQ.NEXTVAL,V_MONTH,V_YEAR,V_NETPAY,V_GROSSPAY,V_TAX,'ONLINE TRANSFER',TEMP.EMP_ID);

   ELSE
    DBMS_OUTPUT.PUT_LINE('INFO: EMPLOYEE '||TEMP.EMP_ID||' HAS AVAILED SALARY ADVANCE ALREADY. CANNOT RELEASE
SALARY');
   END IF;

END LOOP;

COMMIT;
END;
```



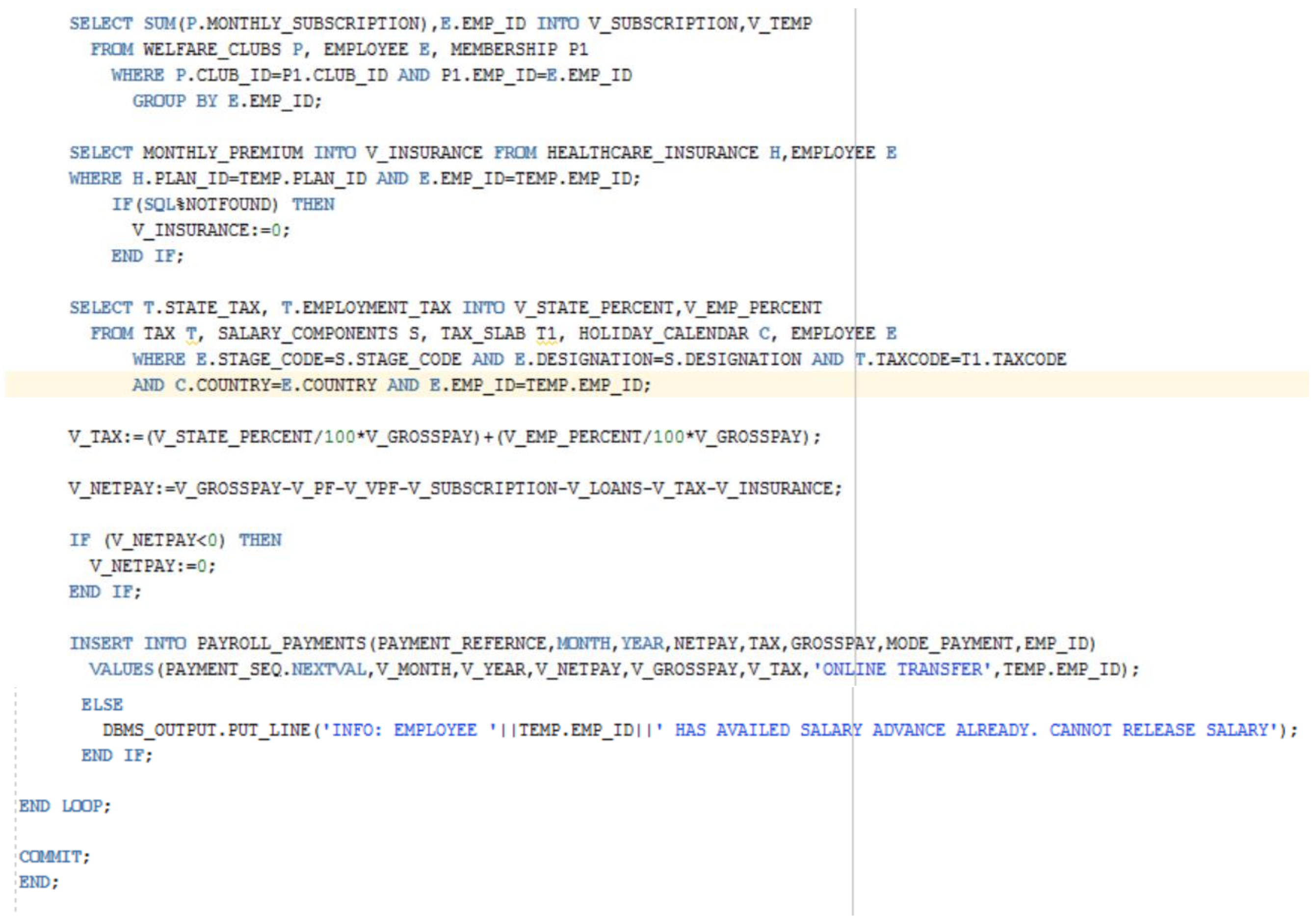
```
CREATE OR REPLACE PROCEDURE PAYROLL_RUN IS
CURSOR EMPLOYEE_DETAILS IS
SELECT * FROM EMPLOYEE;
V_BASIC SALARY_COMPONENTS.BASIC%TYPE;
V_DA SALARY_COMPONENTS.BASIC%TYPE;
V_HRA SALARY_COMPONENTS.BASIC%TYPE;
V_TA SALARY_COMPONENTS.BASIC%TYPE;
V_INCENTIVE PROJECT.MONTHLY_INCENTIVE%TYPE;
V_GROSSPAY NUMBER(10);
V_NETPAY NUMBER(10);
V_PF NUMBER(10);
V_LOANS NUMBER(10);
V_WELFARE NUMBER(10);
V_TAX NUMBER(10);
V_VPF NUMBER(10);
V_PENSION NUMBER(10);
VPF_OPTED CHAR(1);
V_SUBSCRIPTION NUMBER(10);
V_INSURANCE NUMBER(10);
V_LOP LEAVES.LOP%TYPE;
V_TEMP EMPLOYEE.EMP_ID%TYPE;
V_CURRENT_BALANCE SOFT_LOANS.CURRENT_BALANCE%TYPE;
V_LOANID SOFT_LOANS.LOAN_ID%TYPE;
V_STATE_PERCENT TAX.STATE_TAX%TYPE;
V_EMP_PERCENT TAX.EMPLOYMENT_TAX%TYPE;
V_MONTH NUMBER(2);
```



```
S_MONTH NUMBER(2);
S_YEAR NUMBER(4);

BEGIN
FOR TEMP IN EMPLOYEE_DETAILS
  LOOP
  IF TEMP.EMPLOYMENT_STATUS='ACTIVE' THEN
    SELECT BASIC INTO V_BASIC FROM SALARY_COMPONENTS WHERE STAGE_CODE=TEMP.STAGE_CODE AND DESIGNATION=TEMP.DESIGNATION;
    V_DA:=V_BASIC*0.3;
    V_HRA:=V_BASIC*0.1;
    V_TA:=V_BASIC*0.08;
    SELECT EXTRACT(MONTH FROM SYSDATE),EXTRACT(YEAR FROM SYSDATE) INTO V_MONTH,V_YEAR FROM DUAL;
    SELECT MONTH,YEAR INTO S_MONTH,S_YEAR FROM ADVANCE_LOANS  WHERE EMP_ID=TEMP.EMP_ID AND MONTH=V_MONTH AND YEAR=V_YEAR ;
        IF(SQL%NOTFOUND) THEN
          SELECT SUM(P.MONTHLY_INCENTIVE),E.EMP_ID INTO V_INCENTIVE,V_TEMP
            FROM PROJECT P, EMPLOYEE E, WORKS_ON P1
              WHERE P.PID=P1.PID AND P1.EMP_ID=E.EMP_ID
                GROUP BY E.EMP_ID;
      END IF;
      V_GROSSPAY:=V_BASIC+V_DA+V_HRA+V_TA;
      SELECT LOP INTO V_LOP FROM LEAVES WHERE EMP_ID=TEMP.EMP_ID;
            IF(V_LOP>0) THEN
                V_GROSSPAY:=(V_GROSSPAY/30)*(30-V_LOP);
            END IF;

      V_PF:=V_BASIC*0.1;
```



```
      V_PENSION:=V_PF;

            IF(VPF_OPTED='Y') THEN
                V_VPF:=V_BASIC*0.075;
            ELSE
                V_VPF:=0;
            END IF;

      UPDATE RETIREMENT_ACCOUNT
        SET PF_BALANCE=PF_BALANCE+V_PF,VPF_BALANCE=VPF_BALANCE+V_VPF,PENSION_BALANCE=PENSION_BALANCE+V_PENSION
          WHERE EMP_ID=TEMP.EMP_ID;

      SELECT LOAN_ID,CURRENT_BALANCE,LOAN_AMOUNT INTO V_LOANID,V_CURRENT_BALANCE,V_LOANS
        FROM SOFT_LOANS
          WHERE EMP_ID=TEMP.EMP_ID AND CURRENT_BALANCE!=0;

      IF(SQL%FOUND) THEN
        V_LOANS:=V_LOANS/24;
        V_CURRENT_BALANCE:=V_CURRENT_BALANCE-V_LOANS;
        UPDATE SOFT_LOANS SET CURRENT_BALANCE=V_CURRENT_BALANCE WHERE LOAN_ID=V_LOANID;
          IF(V_CURRENT_BALANCE=0) THEN
            UPDATE SOFT_LOANS SET DUE_DATE=SYSDATE WHERE LOAN_ID=V_LOANID;
          END IF;
  END IF;
```



```
    SELECT SUM(P.MONTHLY_SUBSCRIPTION),E.EMP_ID INTO V_SUBSCRIPTION,V_TEMP
      FROM WELFARE_CLUBS P, EMPLOYEE E, MEMBERSHIP P1
        WHERE P.CLUB_ID=P1.CLUB_ID AND P1.EMP_ID=E.EMP_ID
          GROUP BY E.EMP_ID;

    SELECT MONTHLY_PREMIUM INTO V_INSURANCE FROM HEALTHCARE_INSURANCE H,EMPLOYEE E
    WHERE H.PLAN_ID=TEMP.PLAN_ID AND E.EMP_ID=TEMP.EMP_ID;
        IF(SQL%NOTFOUND) THEN
          V_INSURANCE:=0;
        END IF;

    SELECT T.STATE_TAX, T.EMPLOYMENT_TAX INTO V_STATE_PERCENT,V_EMP_PERCENT
      FROM TAX T, SALARY_COMPONENTS S, TAX_SLAB T1, HOLIDAY_CALENDAR C, EMPLOYEE E
          WHERE E.STAGE_CODE=S.STAGE_CODE AND E.DESIGNATION=S.DESIGNATION AND T.TAXCODE=T1.TAXCODE
          AND C.COUNTRY=E.COUNTRY AND E.EMP_ID=TEMP.EMP_ID;

    V_TAX:=(V_STATE_PERCENT/100*V_GROSSPAY)+(V_EMP_PERCENT/100*V_GROSSPAY);

    V_NETPAY:=V_GROSSPAY-V_PF-V_VPF-V_SUBSCRIPTION-V_LOANS-V_TAX-V_INSURANCE;

    IF (V_NETPAY<0) THEN
      V_NETPAY:=0;
    END IF;

    INSERT INTO PAYROLL_PAYMENTS(PAYMENT_REFERNCE,MONTH,YEAR,NETPAY,TAX,GROSSPAY,MODE_PAYMENT,EMP_ID)
      VALUES(PAYMENT_SEQ.NEXTVAL,V_MONTH,V_YEAR,V_NETPAY,V_GROSSPAY,V_TAX,'ONLINE TRANSFER',TEMP.EMP_ID);
   ELSE
     DBMS_OUTPUT.PUT_LINE('INFO: EMPLOYEE '||TEMP.EMP_ID||' HAS AVAILED SALARY ADVANCE ALREADY. CANNOT RELEASE SALARY');
   END IF;

END LOOP;

COMMIT;
END;
```

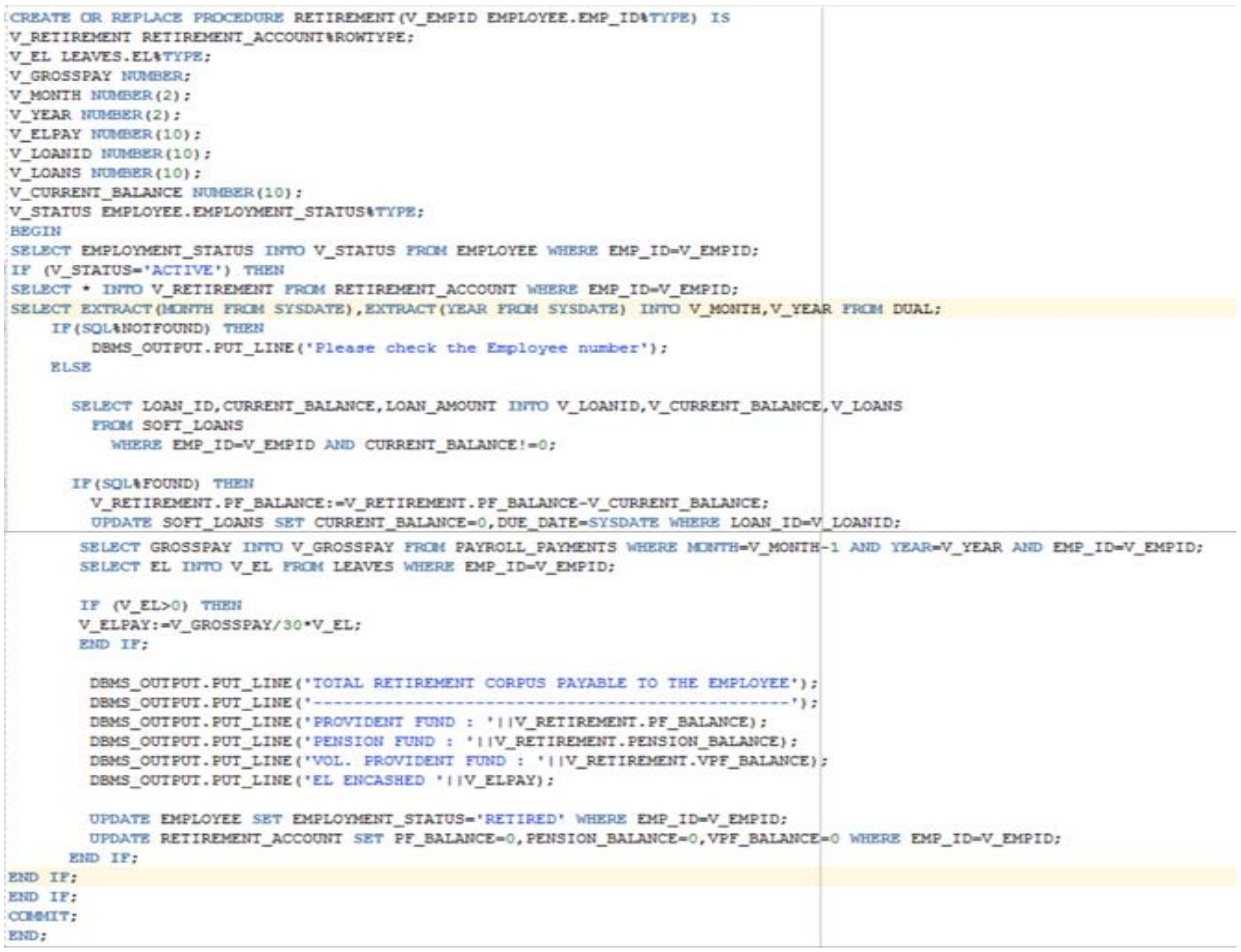
4.
PROCEDURE NAME: RETIREMENT

DESCRIPTION OF THE PROCEDURE: THIS PROCEDURE WILL BE RUN BY THE ADMINSTRATOR WHEN AN EMPLOYEE RETIRES FROM THE COMPANY. THE PROCEDURE CALCULATES THE RETIREMENT BALANCE, ALONG WITH THE LEAVE ENCASHMENT DUE TO THE EMPLOYEE (IF ANY). THE SYSTEM ALSO CLOSES THE EXISTING LOANS TAKEN BY THE EMPLOYEE FROM THE RETIREMENT BENEFITS & GIVES US THE TOTAL AMOUNT TO BE DISBURSED TO THE EMPLOYEE.

RUN FREQUENCY: WHEN AN EMPLOYEE RESIGNS, VOLUTARILY RETIRES, OR SUPERANNUATION.

```
CREATE OR REPLACE PROCEDURE RETIREMENT(V_EMPID EMPLOYEE.EMP_ID%TYPE) IS
V_RETIREMENT RETIREMENT_ACCOUNT%ROWTYPE;
V_EL LEAVES.EL%TYPE;
V_GROSSPAY NUMBER;
V_MONTH NUMBER(2);
V_YEAR NUMBER(2);
V_ELPAY NUMBER(10);
V_LOANID NUMBER(10);
V_LOANS NUMBER(10);
V_CURRENT_BALANCE NUMBER(10);
V_STATUS EMPLOYEE.EMPLOYMENT_STATUS%TYPE;
BEGIN
SELECT EMPLOYMENT_STATUS INTO V_STATUS FROM EMPLOYEE WHERE EMP_ID=V_EMPID;
IF (V_STATUS='ACTIVE') THEN
SELECT * INTO V_RETIREMENT FROM RETIREMENT_ACCOUNT WHERE EMP_ID=V_EMPID;
SELECT EXTRACT(MONTH FROM SYSDATE),EXTRACT(YEAR FROM SYSDATE) INTO V_MONTH,V_YEAR FROM DUAL;
   IF(SQL%NOTFOUND) THEN
      DBMS_OUTPUT.PUT_LINE('Please check the Employee number');
   ELSE

     SELECT LOAN_ID,CURRENT_BALANCE,LOAN_AMOUNT INTO V_LOANID,V_CURRENT_BALANCE,V_LOANS
      FROM SOFT_LOANS
        WHERE EMP_ID=V_EMPID AND CURRENT_BALANCE!=0;

     IF(SQL%FOUND) THEN
      V_RETIREMENT.PF_BALANCE:=V_RETIREMENT.PF_BALANCE-V_CURRENT_BALANCE;
      UPDATE SOFT_LOANS SET CURRENT_BALANCE=0,DUE_DATE=SYSDATE WHERE LOAN_ID=V_LOANID;

      SELECT GROSSPAY INTO V_GROSSPAY FROM PAYROLL_PAYMENTS WHERE MONTH=V_MONTH-1 AND YEAR=V_YEAR AND
EMP_ID=V_EMPID;
      SELECT EL INTO V_EL FROM LEAVES WHERE EMP_ID=V_EMPID;

      IF (V_EL>0) THEN
      V_ELPAY:=V_GROSSPAY/30*V_EL;
      END IF;

      DBMS_OUTPUT.PUT_LINE('TOTAL RETIREMENT CORPUS PAYABLE TO THE EMPLOYEE');
      DBMS_OUTPUT.PUT_LINE('-----------------------------------------------');
      DBMS_OUTPUT.PUT_LINE('PROVIDENT FUND : '||V_RETIREMENT.PF_BALANCE);
      DBMS_OUTPUT.PUT_LINE('PENSION FUND : '||V_RETIREMENT.PENSION_BALANCE);
      DBMS_OUTPUT.PUT_LINE('VOL. PROVIDENT FUND : '||V_RETIREMENT.VPF_BALANCE);
```

```
    DBMS_OUTPUT.PUT_LINE('EL ENCASHED '||V_ELPAY);

    UPDATE EMPLOYEE SET EMPLOYMENT_STATUS='RETIRED' WHERE EMP_ID=V_EMPID;
    UPDATE RETIREMENT_ACCOUNT SET PF_BALANCE=0,PENSION_BALANCE=0,VPF_BALANCE=0 WHERE EMP_ID=V_EMPID;
   END IF;
END IF;
END IF;
COMMIT;
END;
```



```
CREATE OR REPLACE PROCEDURE RETIREMENT(V_EMPID EMPLOYEE.EMP_ID%TYPE) IS
V_RETIREMENT RETIREMENT_ACCOUNT%ROWTYPE;
V_EL LEAVES.EL%TYPE;
V_GROSSPAY NUMBER;
V_MONTH NUMBER(2);
V_YEAR NUMBER(2);
V_ELPAY NUMBER(10);
V_LOANID NUMBER(10);
V_LOANS NUMBER(10);
V_CURRENT_BALANCE NUMBER(10);
V_STATUS EMPLOYEE.EMPLOYMENT_STATUS%TYPE;
BEGIN
SELECT EMPLOYMENT_STATUS INTO V_STATUS FROM EMPLOYEE WHERE EMP_ID=V_EMPID;
IF (V_STATUS='ACTIVE') THEN
SELECT * INTO V_RETIREMENT FROM RETIREMENT_ACCOUNT WHERE EMP_ID=V_EMPID;
SELECT EXTRACT(MONTH FROM SYSDATE),EXTRACT(YEAR FROM SYSDATE) INTO V_MONTH,V_YEAR FROM DUAL;
    IF(SQL%NOTFOUND) THEN
        DBMS_OUTPUT.PUT_LINE('Please check the Employee number');
    ELSE

      SELECT LOAN_ID,CURRENT_BALANCE,LOAN_AMOUNT INTO V_LOANID,V_CURRENT_BALANCE,V_LOANS
        FROM SOFT_LOANS
          WHERE EMP_ID=V_EMPID AND CURRENT_BALANCE!=0;

      IF(SQL%FOUND) THEN
        V_RETIREMENT.PF_BALANCE:=V_RETIREMENT.PF_BALANCE-V_CURRENT_BALANCE;
        UPDATE SOFT_LOANS SET CURRENT_BALANCE=0,DUE_DATE=SYSDATE WHERE LOAN_ID=V_LOANID;
       SELECT GROSSPAY INTO V_GROSSPAY FROM PAYROLL_PAYMENTS WHERE MONTH=V_MONTH-1 AND YEAR=V_YEAR AND EMP_ID=V_EMPID;
       SELECT EL INTO V_EL FROM LEAVES WHERE EMP_ID=V_EMPID;

       IF (V_EL>0) THEN
       V_ELPAY:=V_GROSSPAY/30*V_EL;
       END IF;

        DBMS_OUTPUT.PUT_LINE('TOTAL RETIREMENT CORPUS PAYABLE TO THE EMPLOYEE');
        DBMS_OUTPUT.PUT_LINE('----------------------------------------------');
        DBMS_OUTPUT.PUT_LINE('PROVIDENT FUND : '||V_RETIREMENT.PF_BALANCE);
        DBMS_OUTPUT.PUT_LINE('PENSION FUND : '||V_RETIREMENT.PENSION_BALANCE);
        DBMS_OUTPUT.PUT_LINE('VOL. PROVIDENT FUND : '||V_RETIREMENT.VPF_BALANCE);
        DBMS_OUTPUT.PUT_LINE('EL ENCASHED '||V_ELPAY);

        UPDATE EMPLOYEE SET EMPLOYMENT_STATUS='RETIRED' WHERE EMP_ID=V_EMPID;
        UPDATE RETIREMENT_ACCOUNT SET PF_BALANCE=0,PENSION_BALANCE=0,VPF_BALANCE=0 WHERE EMP_ID=V_EMPID;
      END IF;
END IF;
END IF;
COMMIT;
END;
```

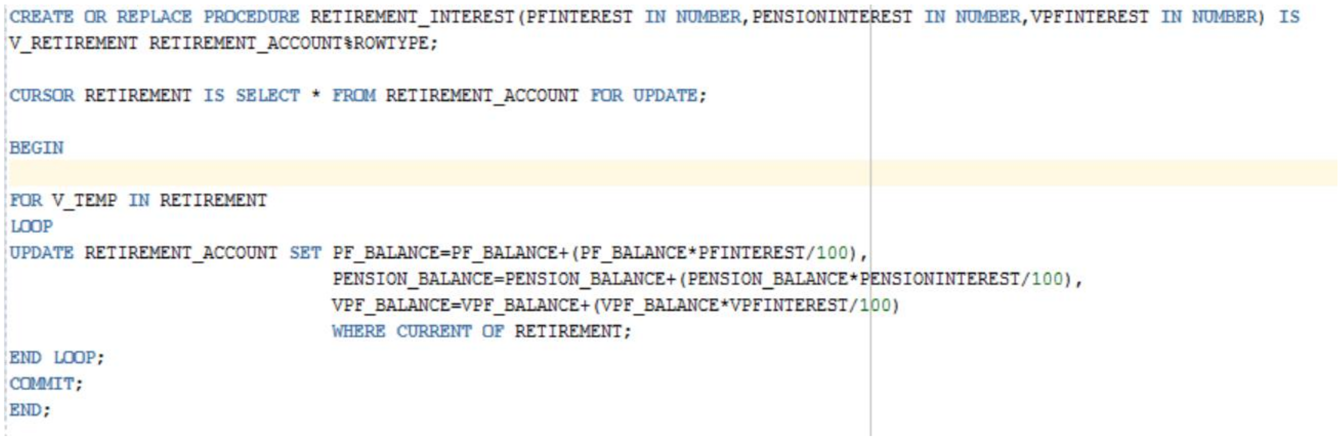
5.

PROCEDURE NAME: RETIREMENT_INTEREST

DESCRIPTION OF THE PROCEDURE: THIS PROCEDURE WILL BE RUN BY THE ADMINSTRATOR EVERY YEAR TO UPDATE THE INTEREST ON THE RETIRMENT BALANCES OF THE EMPLOYEE. TYPICALLY IT IS RUN YEARLY ONCE AFTER DETERMINING THE INTEREST RATE.

RUN FREQUENCY: YEARLY.

```
CREATE OR REPLACE PROCEDURE RETIREMENT_INTEREST(PFINTEREST IN NUMBER,PENSIONINTEREST IN NUMBER,VPFINTEREST IN
NUMBER) IS
V_RETIREMENT RETIREMENT_ACCOUNT%ROWTYPE;

CURSOR RETIREMENT IS SELECT * FROM RETIREMENT_ACCOUNT FOR UPDATE;

BEGIN

FOR V_TEMP IN RETIREMENT
LOOP
UPDATE RETIREMENT_ACCOUNT SET PF_BALANCE=PF_BALANCE+(PF_BALANCE*PFINTEREST/100),
                  PENSION_BALANCE=PENSION_BALANCE+(PENSION_BALANCE*PENSIONINTEREST/100),
                  VPF_BALANCE=VPF_BALANCE+(VPF_BALANCE*VPFINTEREST/100)
                  WHERE CURRENT OF RETIREMENT;
END LOOP;
COMMIT;
END;
```



```
CREATE OR REPLACE PROCEDURE RETIREMENT_INTEREST(PFINTEREST IN NUMBER,PENSIONINTEREST IN NUMBER,VPFINTEREST IN NUMBER) IS
V_RETIREMENT RETIREMENT_ACCOUNT%ROWTYPE;

CURSOR RETIREMENT IS SELECT * FROM RETIREMENT_ACCOUNT FOR UPDATE;

BEGIN

FOR V_TEMP IN RETIREMENT
LOOP
UPDATE RETIREMENT_ACCOUNT SET PF_BALANCE=PF_BALANCE+(PF_BALANCE*PFINTEREST/100),
                              PENSION_BALANCE=PENSION_BALANCE+(PENSION_BALANCE*PENSIONINTEREST/100),
                              VPF_BALANCE=VPF_BALANCE+(VPF_BALANCE*VPFINTEREST/100)
                              WHERE CURRENT OF RETIREMENT;
END LOOP;
COMMIT;
END;
```