**NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI-15**

****

**Department of Computer Applications**

TASKHUB

(DBMS PROJECT)

*Submitted By*

**SUKHLAL AHIRWAR – 205118074**

*Under the guidance of*

**Dr. R Eswari**

*Submitted in fulfillment of the project in PL/SQL*

**NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI - 15**

****

**CERTIFICATE**

*This is to certify that* ***SUKHLAL AHIRWAR,*** *student of 2nd semester MCA (batch 2018-2021) of National Institute of Technology, Tiruchirappalli has successfully completed the project* ***’TASKHUB’*** *PL/SQL**under the guidance of* ***Dr. R Eswari.***

Signature

(Dr. R Eswari)

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Description** | **Page No.** |
| 1 | Problem Description and Analysis | 4 |
| 2 | Schema and Tables Descriptions | 5 |
| 3 | Creating tables | 7 |
| 4 | Initializing values | 10 |
| 5 | Packages | 11 |
| 6 | Procedures, view and triggers | 12 |
| 7 | Enhancements & conclusion | 19 |
| 8 | References & Bibliography | 20 |

**Problem Description and Analysis**

The Project titled “TASKHUB” is a database management system for Task provided by Keen and Solved by Wizard. Problem statement will be given by keen in the form of task, a task can be any of the business operation or execution which is commercially legal and a task may be a Query or Drought regarding any of the Procedure and any kind of enthusiasm.

**SCHEMA AND TABLE DESCRIPTION**

**TABLE: KEEN**

Name Null? Type

----------------------------------- -------- ------------------------

KEENID NOT NULL VARCHAR2(10)

PASSWORD NOT NULL VARCHAR2(20)

NAME NOT NULL VARCHAR2(20)

GENDER NOT NULL CHAR(1)

DOB NOT NULL DATE

MOBILE NOT NULL NUMBER(10)

EMAIL VARCHAR2(30)

ADDRESS VARCHAR2(30)

JOIN NOT NULL DATE

TASKASK NOT NULL NUMBER(5)

**TABLE: WIZARD**

Name Null? Type

----------------------------------- -------- ------------------------

WIZARDID NOT NULL VARCHAR2(10)

PASSWORD NOT NULL VARCHAR2(20)

NAME NOT NULL VARCHAR2(20)

GENDER NOT NULL CHAR(1)

DOB NOT NULL DATE

MOBILENO NOT NULL NUMBER(10)

EMAIL VARCHAR2(30)

ADDRESS VARCHAR2(20)

WJOIN NOT NULL DATE

SOLVED NOT NULL NUMBER(5)

TABLE: **TASK**

Name Null? Type

----------------------------------- -------- ------------------------

TASKID NOT NULL VARCHAR2(10)

DES NOT NULL VARCHAR2(100)

TIME NOT NULL DATE

KEENID NOT NULL VARCHAR2(10)

ISSOLVE NOT NULL CHAR(1)

TABLE: **PROPOSE**

Name Null? Type

----------------------------------- -------- ------------------------

PID NOT NULL VARCHAR2(10)

TASKID NOT NULL VARCHAR2(10)

WIZID NOT NULL VARCHAR2(10)

DURATION NOT NULL NUMBER(5)

TIME NOT NULL DATE

ISACCEPT NOT NULL CHAR(1)

TABLE: **SOLUTION**

Name Null? Type

----------------------------------- -------- ------------------------

SID NOT NULL VARCHAR2(10)

PID NOT NULL VARCHAR2(10)

SOL NOT NULL VARCHAR2(100)

TAKETIME NOT NULL NUMBER(5)

**CREATING TABLES**

**CREATE TABLE TASK**

( "TASKID" VARCHAR2(10 BYTE) NOT NULL ENABLE,

"DES" VARCHAR2(100 BYTE) NOT NULL ENABLE,

"TIME" DATE NOT NULL ENABLE,

"KEENID" VARCHAR2(10 BYTE) NOT NULL ENABLE,

"ISSOLVE" CHAR(1 BYTE) NOT NULL ENABLE,

PRIMARY KEY ("TASKID"),

CONSTRAINT "SYSTASKID" CHECK (TASKID LIKE 'T%') ENABLE

) ;

**CREATE TABLE KEEN**

( "KEENID" VARCHAR2(10 BYTE) NOT NULL ENABLE,

"PASSWORD" VARCHAR2(20 BYTE) NOT NULL ENABLE,

"NAME" VARCHAR2(20 BYTE) NOT NULL ENABLE,

"GENDER" CHAR(1 BYTE) NOT NULL ENABLE,

"DOB" DATE NOT NULL ENABLE,

"MOBILE" NUMBER(10,0) NOT NULL ENABLE,

"EMAIL" VARCHAR2(30 BYTE),

"ADDRESS" VARCHAR2(30 BYTE),

"JOIN" DATE NOT NULL ENABLE,

"TASKASK" NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,

CONSTRAINT "SYS\_KEENEMAILAT" CHECK (EMAIL LIKE '%@%') ENABLE,

CONSTRAINT "SYSKEENPRIMARY" PRIMARY KEY ("KEENID"),

CONSTRAINT "SYSKEENADDK" CHECK (KEENID LIKE 'K%') ENABLE,

CONSTRAINT "SYSKEENADDMF" CHECK (GENDER IN ('M','F')) ENABLE

) ;

**CREATE TABLE "WIZARD"**

( "WIZARDID" VARCHAR2(10 BYTE) NOT NULL ENABLE,

"PASSWORD" VARCHAR2(20 BYTE) NOT NULL ENABLE,

"NAME" VARCHAR2(20 BYTE) NOT NULL ENABLE,

"GENDER" CHAR(1 BYTE) NOT NULL ENABLE,

"DOB" DATE NOT NULL ENABLE,

"MOBILENO" NUMBER(10,0) NOT NULL ENABLE,

"EMAIL" VARCHAR2(30 BYTE),

"ADDRESS" VARCHAR2(20 BYTE),

"WJOIN" DATE NOT NULL ENABLE,

"SOLVED" NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,

CONSTRAINT "SYSWIZPRI" PRIMARY KEY ("WIZARDID"),

CONSTRAINT "SYSWIZMF" CHECK (GENDER IN ('M','F')) ENABLE,

CONSTRAINT "SYSWIZEMAILAT" CHECK (EMAIL LIKE '%@%') ENABLE,

CONSTRAINT "SYSWIZW" CHECK (WIZARDID LIKE 'W%') ENABLE

) ;

**CREATE TABLE "PROPOSE"**

( "PID" VARCHAR2(10 BYTE) NOT NULL ENABLE,

"TASKID" VARCHAR2(10 BYTE) NOT NULL ENABLE,

"WIZID" VARCHAR2(10 BYTE) NOT NULL ENABLE,

"DURATION" NUMBER(5,0) NOT NULL ENABLE,

"TIME" DATE NOT NULL ENABLE,

"ISACCEPT" CHAR(1 BYTE) NOT NULL ENABLE,

CONSTRAINT "SYSPROISA" CHECK (ISACCEPT IN ('Y','N')) ENABLE,

CONSTRAINT "SYSPROP" CHECK (PID LIKE 'P%') ENABLE,

CONSTRAINT "SYSPROPRI" PRIMARY KEY ("PID") ,

CONSTRAINT "SYSTASKFOR" FOREIGN KEY ("TASKID")

REFERENCES "TASK" ("TASKID") ENABLE,

CONSTRAINT "SYSPROWFOR" FOREIGN KEY ("WIZID")

REFERENCES "WIZARD" ("WIZARDID") ENABLE

) ;

**CREATE TABLE "SOLUTION"**

( "SID" VARCHAR2(10 BYTE) NOT NULL ENABLE,

"PID" VARCHAR2(10 BYTE) NOT NULL ENABLE,

"SOL" VARCHAR2(100 BYTE) NOT NULL ENABLE,

"TAKETIME" NUMBER(5,0) NOT NULL ENABLE,

CONSTRAINT "SYSSOLP" FOREIGN KEY ("PID")

REFERENCES PROPOSE ("PID") ENABLE

) ;

**create table login**

( status char(1) default 'F',

id varchar2(10),

constraint "loginstatus" check(status in ('T','F')) enable

);

**CREATE SEQUENCE seq**

START WITH 1

INCREMENT BY 1

CACHE 10000;

**INITIALIZING VALUES**

SET SERVEROUTPUT ON;

SET LINESIZE 200;

SET PAGESIZE 200;

**modules**

**PACKAGE: PROALL**

CREATE OR REPLACE

PACKAGE PROALL AS

procedure logout;

procedure myrights;

procedure proposeaccept(id in varchar2);

procedure proposetask(v in varchar2,t varchar2);

procedure signin(id in varchar2,pass in varchar2);

procedure signup;

procedure solve;

procedure taskhub;

procedure taskask;

END PROALL;

**PROCEDURES, view AND TRIGGER**

**CREATE OR REPLACE PROCEDURE LOGOUT** AS

BEGIN

update login set status='F';

END LOGOUT;

**CREATE OR REPLACE PROCEDURE MYRIGHTS** IS

cursor c1 is select id from login;

cursor c2 is select status from login;

id varchar2(10);

t char(1);

BEGIN

open c1;

open c2;

fetch c2 into t;

if t='T' then

fetch c1 into id;

if id like 'W%' then

DBMS\_OUTPUT.PUT\_LINE('WIZARD:');

DBMS\_OUTPUT.PUT\_LINE('1.PROPOSE A TASK :propose(taskid)');

DBMS\_OUTPUT.PUT\_LINE('2.PROVIDE SOLUTION :solve(proposeid,solution)');

DBMS\_OUTPUT.PUT\_LINE('3.SEARCH TASK :task');

DBMS\_OUTPUT.PUT\_LINE('4.SEARCH MY Proposals :totalproposed');

DBMS\_OUTPUT.PUT\_LINE('5.SEARCH proposed accepted :searchaccepted');

DBMS\_OUTPUT.PUT\_LINE('6.LOGOUT : logout');

end if;

if id like 'K%' then

DBMS\_OUTPUT.PUT\_LINE('KEEN:');

DBMS\_OUTPUT.PUT\_LINE('1.TASK : taskask(desc)');

DBMS\_OUTPUT.PUT\_LINE('2.SEARCH SOLUTION :searchsolve()');

DBMS\_OUTPUT.PUT\_LINE('3.ASKED TASK :asktask');

DBMS\_OUTPUT.PUT\_LINE('3.SEARCH PROPOSED :task');

DBMS\_OUTPUT.PUT\_LINE('4.PROPOSE ACCEPT:proposeaccept(pid)');

DBMS\_OUTPUT.PUT\_LINE('5.LOGOUT : logout');

end if;

else

DBMS\_OUTPUT.PUT\_LINE(' Login Please !!!!!!!!!!!!');

end if;

close c1;

close c2;

END MYRIGHTS;

**CREATE OR REPLACE PROCEDURE PROPOSEACCEPT**(id in varchar2)AS

BEGIN

update propose set isaccept='Y' where pid=id;

END PROPOSEACCEPT;

create or replace PROCEDURE SIGNIN(idd in varchar2,pass in varchar2) is

k KEEN.KEENID%type;

w WIZARD.WIZARDID%type;

d varchar2(20);

p varchar2(20);

cursor c3 is select password from keen where keenid=idd;

cursor c4 is select password from wizard where wizardid=idd;

BEGIN

open c3;

open c4;

delete from login;

fetch c3 into p;

fetch c4 into d;

if pass = p then

DBMS\_OUTPUT.PUT\_LINE('KEEN LOGGED IN');

else if pass = d then

DBMS\_OUTPUT.PUT\_LINE('WIZARD LOGGED IN');

else

DBMS\_OUTPUT.PUT\_LINE('Wrong ID password');

end if;

end if;

close c3;

close c4;

insert into login values('T',idd);

END SIGNIN;

**create or replace PROCEDURE PROPOSETASK**(id in varchar2,dur in number ) is

pid varchar2(10) ;

w varchar2(10);

BEGIN

pid := concat('P',seq.nextval);

select id into w from login;

insert into propose (pid,taskid,wizid,duration,time,isaccept) values(pid,id,w,dur,sysdate,'N');

END PROPOSETASK;

/

**CREATE OR REPLACE PROCEDURE SIGNUP**(kw in varchar2,n in varchar2,p in varchar2,g in char,d in date,m in number,e in varchar2,a in varchar2)

is

id varchar2(10);

BEGIN

if kw='keen' or kw='KEEN' then

id:=concat('K',seq.nextval);

insert into keen values(id,p,n,g,d,m,e,a,sysdate,0);

DBMS\_OUTPUT.PUT\_line(concat('Your ID is:',id));

else if kw='wizard' or kw='WIZARD' then

id:=concat('W',seq.nextval);

insert into wizard values(id,p,n,g,d,m,e,a,sysdate,0);

DBMS\_OUTPUT.PUT\_line(concat('Your ID is:',id));

else

dbms\_output.put\_line('worng input format');

end if;

end if;

END SIGNUP;

**CREATE OR REPLACE PROCEDURE SOLVE**(p in varchar2,de in varchar2) AS

idd varchar2(10);

t number(5);

BEGIN

idd:=concat('S',seq.nextval);

select round(time-sysdate) into t from propose where pid=p;

insert into solution values (idd,p,de,t);

END SOLVE;

**CREATE OR REPLACE PROCEDURE TASKASK**(des in varchar2) is

cursor c1 is select id from login;

id varchar2(10);

tid varchar2(10);

BEGIN

open c1;

fetch c1 into id;

close c1;

tid:=concat('T',seq.nextval);

insert into task values(tid,des,sysdate,id,'F');

dbms\_output.put\_line(tid);

END TASKASK;

**create or replace procedure TaskHub** is

begin

dbms\_output.put\_line('############### TASKHUB ##################');

dbms\_output.put\_line('1. signin(id,password)');

dbms\_output.put\_line('2. signup(keen/wizard,name,password,gender(M/F),dob,monile,email,address)');

end taskhub;

**CREATE OR REPLACE VIEW ASKTASK** AS

SELECT des as TASK

FROM task where

keenid=(select id from login);

**create or replace view searchaccepted** as

select PID from propose where wizid=(select id from login) and isaccept='Y';

**create or replace view searchproposed** as

select propose.PID,propose.TASKID,propose.WIZID,propose.duration from propose,task,keen,login where propose.taskid=task.taskid and task.keenid=keen.keenid and keen.keenid=login.id and task.issolve='F';

**create or replace view searchsolv**e as

select sol from solution , propose , task , login,keen where solution.pid=propose.pid and (propose.taskid=task.taskid) and (task.keenid = keen.keenid) and (keen.keenid = login.id) ;

**create or replace view searchtask** as

select taskid,des,time from task where issolve='F' ;

**create or replace view totalpropose** as

select PID from propose where wizid=(select id from login);

**CREATE OR REPLACE TRIGGER ISSOLVE**

after INSERT ON solution

REFERENCING OLD AS OLD NEW AS NEW

for each row

declare

p varchar2(10);

BEGIN

select task.taskid into p from propose,task where propose.pid=:new.pid and propose.taskid=task.taskid;

update task set issolve='T' where taskid=p;

END;

**CREATE OR REPLACE TRIGGER INCASK**

AFTER INSERT ON task

REFERENCING OLD AS OLD NEW AS NEW

FOR EACH ROW

declare

ta number(5);

BEGIN

select taskask into ta from keen where keenid=:new.keenid;

update keen set taskask=ta+1 where keenid=:new.keenid;

END;

**ENHANCEMENT AND CONCLUSION**

**Enhancement:**

TASKHUB is a best idea for a centralized platform for every kind of business

Where any of the broker is needed. Broker system will be removed from the service provider and customer.

This is the Web Application Project so efficiency, reliability and speed of work will be increased. Reaching to the customers will be very easy for service providers. This will be the central communication hub for selling the services.

**Conclusion:**

My project TASKHUB direct me to learn the working of database concepts and

Working procedure of seller and buyer.

How the database system helps in real world. Dr. R Eswari support me to work and teach me to handle this problem.

**REFERENCES AND BIBLIOGRAPHY**

1. **Koch and Loney –** Oracle 8 : The complete reference
2. **Ivan Bayross -** SQL, PL/SQL : The programming language of Oracle
3. **Korth and Sudershan** **-** Database System Concepts
4. **Oracle sql/plsql tutorial**