-- MySQL dump 10.13 Distrib 5.7.17, for Win64 (x86\_64)

--

-- Host: 35.197.140.185 Database: nexat\_uat

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

-- Server version 5.7.23

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `attendance`

--

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping routines for database 'nexat\_uat'

--

/\*!50003 DROP FUNCTION IF EXISTS `stringSplit` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` FUNCTION `stringSplit`(

x TEXT,

delim VARCHAR(12),

pos INT

) RETURNS varchar(255) CHARSET latin1

RETURN REPLACE(SUBSTRING(SUBSTRING\_INDEX(x, delim, pos),

LENGTH(SUBSTRING\_INDEX(x, delim, pos -1)) + 1), delim, '') ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP FUNCTION IF EXISTS `substrCount` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` FUNCTION `substrCount`(s TEXT, ss TEXT) RETURNS int(5) unsigned

READS SQL DATA

BEGIN

DECLARE count INT(5) UNSIGNED;

DECLARE offset INT(5) UNSIGNED;

DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET s = NULL;

SET count = 0;

SET offset = 1;

REPEAT

IF NOT ISNULL(s) AND offset > 0 THEN

SET offset = LOCATE(ss, s, offset);

IF offset > 0 THEN

SET count = count + 1;

SET offset = offset + 1;

END IF;

END IF;

UNTIL ISNULL(s) OR offset = 0 END REPEAT;

RETURN count;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `new\_Show\_Notifications` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `new\_Show\_Notifications`(

IN pUserID int(11),

IN pDeptID text,

IN pOnDate date

)

BEGIN

DECLARE DeptMngUserID,SupervisorApprovalCount,EarlycheckoutApprovalCount,VisitorcheckoutApprovalCount,

PendingLeaveCount,AppliedCmofCount,OdCount INT;

SELECT group\_concat(DM.DeptManagerID) INTO DeptMngUserID from deptmanager DM where DM.DeptID IN (pDeptID);

/\*Get all hierarchy sub department ids\*/

CALL nex\_GetDeptTree(pDeptID,@AllHrcyDept);

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

select count(SAPV.SupervisorApprvID) INTO SupervisorApprovalCount

from supervisorapproval SAPV

LEFT JOIN userpersonalinfo UP ON SAPV.UserID = UP.UserID

where SAPV.RequestStatus = 'Applied' and SAPV.SupervisorID = pUserID;

select count(ECOA.EarlyCheckoutApprvID) INTO EarlycheckoutApprovalCount

from earlycheckoutapproval ECOA

LEFT JOIN attendance A ON ECOA.AttendanceID = A.AttendanceID

LEFT JOIN userpersonalinfo UP ON ECOA.UserID = UP.UserID

where ECOA.SupervisorUserID = pUserID and ECOA.RequestStatus = 'Applied';

select count(VA.VisitorAttendanceID) INTO VisitorcheckoutApprovalCount

from visitorattendance VA

where VA.MeetToUserID = pUserID and VA.VisitAttenStatus IN ('CheckedIn','CheckedOut-NotVarified');

SELECT count(LA.LeavAppID) INTO PendingLeaveCount

FROM leaveapplied LA

LEFT JOIN leaveappliedstatuslog LLOG ON LLOG.LeavAppID = LA.LeavAppID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

WHERE LA.LeaveStatusID = LLOG.LeaveStatusID AND LA.StatusUpdateDate = LLOG.StatusAppliedDate

AND U.DeptID IN (select \* from temp\_AllDept)

AND LA.UserID NOT IN (DeptMngUserID)

AND LS.LeaveMasterStatusID = 1;

SELECT count(CMOF.CompOffID) INTO AppliedCmofCount

FROM compoffapplied CMOF

LEFT JOIN userpersonalinfo U ON CMOF.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

WHERE CMOF.RequestStatus = 'Applied'

AND U.DeptID IN (select \* from temp\_AllDept)

AND CMOF.UserID NOT IN (DeptMngUserID);

SELECT count(OD.OdID) INTO OdCount

FROM odapplied OD

LEFT JOIN userpersonalinfo U ON OD.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

where OD.RequestStatus = 'Applied'

AND U.DeptID IN (select \* from temp\_AllDept)

AND OD.UserID NOT IN (DeptMngUserID);

select SupervisorApprovalCount,EarlycheckoutApprovalCount,VisitorcheckoutApprovalCount,

PendingLeaveCount,AppliedCmofCount,OdCount;

DROP TEMPORARY TABLE temp\_AllDept;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_addAttendance\_log` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_addAttendance\_log`(

in pAttendanceLogID bigint(20),

in pOrgID int(11),

in pAttendanceLog TEXT,

in pLogTime DATETIME,

in pIsProcessed char(1),

in pProcessedTime DATETIME,

in pTypeFlg int(1)

)

BEGIN

if pTypeFlg=1 then

INSERT INTO attendance\_log

(

AttendanceLog,

OrgID,

LogTime,

IsProcessed,

ProcessedTime

)

VALUES

(

pAttendanceLog,

pOrgID,

pLogTime,

pIsProcessed,

NULL

);

elseif pTypeFlg=2 then

UPDATE attendance\_log AL SET

AL.IsProcessed= pIsProcessed,

AL.ProcessedTime= pProcessedTime

WHERE AL.AttendanceLogID= pAttendanceLogID;

elseif pTypeFlg=3 then

select AL.AttendanceLogID,AL.AttendanceLog

from attendance\_log AL

where AL.OrgID = pOrgID and AL.IsProcessed = 'N' order by AL.LogTime asc limit 0,30;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_addOrganization` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_addOrganization`(

IN pOrgName VARCHAR(15) ,

IN pContactEmail VARCHAR(15) ,

IN pWorkPhone VARCHAR(10) ,

IN pCellPhone VARCHAR(10) ,

IN pAddress VARCHAR(20) ,

IN pReferenceName VARCHAR(20)

)

BEGIN

DECLARE pCreatedDate datetime;

SET pCreatedDate=Now();

INSERT INTO nexorganization

(

OrgName ,

ContactEmail ,

WorkPhone ,

CellPhone ,

Address ,

ReferenceName

)

VALUES

(

pOrgName ,

pContactEmail ,

pWorkPhone ,

pCellPhone ,

pAddress ,

pReferenceName

) ;

SELECT \* from nexorganization where CreatedDate =pCreatedDate and OrgName=pOrgName and ContactEmail=pContactEmail and WorkPhone=pWorkPhone and CellPhone= pCellPhone and Address=pAddress;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_addOrganizationRole` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_addOrganizationRole`(

in `@pRoleName` varchar(50),

in `@OrgID` int(11)

)

BEGIN

INSERT INTO userrole(RoleName,OrigId)

VALUES(pRoleName,OrgID);

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_addOrganizationSignup` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_addOrganizationSignup`(

IN pFName VARCHAR(30),

IN pLName VARCHAR(30),

IN pOrgName VARCHAR(500),

IN pContactEmail VARCHAR(150),

IN pWorkPhone VARCHAR(50),

IN pCellPhone VARCHAR(50),

IN pAddress VARCHAR(200),

IN pReferenceName VARCHAR(200),

IN pPassword VARCHAR(50)

)

BEGIN

DECLARE pCreatedDate datetime;

SET pCreatedDate=Now();

INSERT INTO nexorganization

(

OrgName ,

ContactEmail ,

WorkPhone ,

CellPhone ,

Address ,

ReferenceName

)

VALUES

(

pOrgName ,

pContactEmail ,

pWorkPhone ,

pCellPhone ,

pAddress ,

pReferenceName

) ;

SELECT OrgID INTO @lpOrgID from nexorganization where CreatedDate =pCreatedDate and OrgName=pOrgName and ContactEmail=pContactEmail;

INSERT INTO userpersonalinfo (OrgID,FName,LName,Address,Email,WorkPhone,CellNo,Pwd,CreatedByUserID,PwdResetRequired)

VALUES(@lpOrgID,pFName,pLName,pAddress,pContactEmail,pWorkPhone,pCellPhone,pPassword,0,1);

Select UserID Into @lpUserID From userpersonalinfo where OrgID = @lpOrgID and Email = pContactEmail;

INSERT Into userrole (RoleName, OrgID) values ('ADMIN', @lpOrgID);

Select RoleID into @lpRoleID from userrole where RoleName = 'ADMIN' and OrgID = @lpOrgID;

INSERT Into UserworkInfo(UserID, RoleID, Designation) values (@lpUserID, @lpRoleID, 'CEO');

/\*Additional Queries\*/

INSERT INTO department (DeptPrtID, DepartmentName, OrgID) VALUES ('0', pOrgName, @lpOrgID);

SET @lpDeptID = LAST\_INSERT\_ID();

INSERT INTO deptmanager (DeptID, DeptManagerID) VALUES (@lpDeptID, @lpUserID);

UPDATE userpersonalinfo SET DeptID = @lpDeptID WHERE UserID = @lpUserID;

Select Org.OrgID,Org.OrgName,Org.ContactEmail,Org.WorkPhone,Org.CellPhone,Org.Address, Org.ReferenceName,pi.FName,pi.LName, pi.Pwd as AdminPass from

nexorganization as Org JOIN userpersonalinfo pi on Org.OrgID=pi.OrgID where Org.OrgID=@lpOrgID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShowMaterial` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShowMaterial`(

IN pMaterialID int(11),

IN pOrgID int(11),

IN pMaterialType char(4),

IN pDescription text,

IN pQuantity float(11,2),

IN pChallanNo varchar(255),

IN pSenderName varchar(255),

IN pTransporterType char(4),

IN pVehicleNumber varchar(55),

IN pDriverName varchar(55),

IN pTypeFlg varchar(45)

)

BEGIN

DECLARE LASTMATERIALID int(11);

if pTypeFlg= 'ADD' Then

IF pMaterialType= 'I' THEN

INSERT INTO material

(

OrgID,

MaterialType,

Description,

Quantity,

ChallanNo,

SenderName,

GatePassNo,

TransporterType,

VehicleNumber,

DriverName

)

VALUES

(

pOrgID,

pMaterialType,

pDescription,

pQuantity,

pChallanNo,

pSenderName,

'',

pTransporterType,

pVehicleNumber,

pDriverName

);

ELSE

INSERT INTO material

(

OrgID,

MaterialType,

Description,

Quantity,

ChallanNo,

GatePassNo,

SenderName,

TransporterType,

VehicleNumber,

DriverName

)

VALUES

(

pOrgID,

pMaterialType,

pDescription,

pQuantity,

pChallanNo,

pSenderName,

'',

pTransporterType,

pVehicleNumber,

pDriverName

);

END IF;

SET LASTMATERIALID = LAST\_INSERT\_ID();

select LASTMATERIALID as 'MATERIALID';

elseif pTypeFlg='FETCH' Then

SELECT M.MaterialType,M.ChallanNo,M.SenderName,M.GatePassNo,M.Modified FROM material M WHERE M.OrgID = pOrgID AND M.MaterialID = pMaterialID;

elseif pTypeFlg='FETCHALL\_BY\_ORG' Then

SELECT m.\* FROM material m

WHERE m.OrgID = pOrgID

ORDER BY m.Modified DESC,m.SenderName ASC;

elseif pTypeFlg='FETCH\_FIRST\_GOODS' Then

SELECT MaterialImageID FROM materialimages

WHERE MaterialID = pMaterialID AND ImageType='G' ORDER BY MaterialImageID ASC limit 0,1;

END IF;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShowMaterialImage` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShowMaterialImage`(

IN pMaterialImageID int(11),

IN pOrgID int(11),

IN pMaterialID int(11),

IN pImageType char(4),

IN pImageContent longblob,

IN pTypeFlg varchar(50)

)

BEGIN

DECLARE EXIT HANDLER FOR 1452 SELECT 'InvalidMaterialID';

if pTypeFlg= 'ADD' THEN

INSERT INTO materialimages

(

OrgID,

MaterialID,

ImageType,

ImageContent

)

VALUES

(

pOrgID,

pMaterialID,

pImageType,

pImageContent

);

SELECT LAST\_INSERT\_ID() AS 'MATERIAL\_IMAGE\_ID';

elseif pTypeFlg='FETCH\_BY\_MATERIALID' THEN

SELECT M.\*,MI.MaterialImageID,MI.ImageType,MI.Created FROM material M

LEFT JOIN materialimages MI ON M.MaterialID=MI.MaterialID

WHERE M.OrgID = pOrgID AND

M.MaterialID = pMaterialID

ORDER BY MI.ImageType ASC,MI.Created DESC;

elseif pTypeFlg='FETCH\_BY\_MATERIAL\_IMAGE\_ID' THEN

SELECT MI.MaterialImageID,MI.ImageContent FROM materialimages MI

WHERE MI.MaterialImageID = pMaterialImageID ;

elseif pTypeFlg='DEL' THEN

DELETE FROM materialimages WHERE MaterialImageID=pMaterialImageID and OrgID=pOrgID;

END IF;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShowMaterialNotifiedUser` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShowMaterialNotifiedUser`(

IN pMaterialNotifiedUserID int(11),

IN pOrgID int(11),

IN pMaterialID int(11),

IN pUserID text,

IN pCreatedOn datetime,

IN pTypeFlg varchar(30)

)

BEGIN

DECLARE i,NOOFUSER int;

IF pTypeFlg= 'ADD' THEN

call splitter(pUserID,',');

select count(SplitValue) INTO NOOFUSER from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFUSER DO

INSERT INTO materialnotifieduser

(

OrgID,

MaterialID,

UserID,

CreatedOn

)

VALUES

(

pOrgID,

pMaterialID,

(select SplitValue from splitResultsTmpTable where TmpID = i),

pCreatedOn

);

SET i = i + 1;

END WHILE;

ELSEIF pTypeFlg= 'FETCH\_NOTIFIED' THEN

SELECT UPI.UserID,UPI.FName,UPI.LName,UPI.Email,UPI.EmployeeID,UPI.DeptID

FROM materialnotifieduser MNU LEFT JOIN userpersonalinfo UPI ON MNU.UserID=UPI.UserID

WHERE MNU.MaterialID=pMaterialID AND MNU.OrgID=pOrgID ORDER BY MaterialNotifiedUserID ASC;

ELSEIF pTypeFlg= 'FETCH\_MY\_MATERIAL' THEN

SELECT M.\* FROM material M JOIN materialnotifieduser MNU ON M.MaterialID=MNU.MaterialID

WHERE MNU.UserID=pUserID ORDER BY MNU.CreatedOn DESC;

END IF;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShowMaterialVisitor` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShowMaterialVisitor`(

in pMaterialVisitorID int(11),

in pOrgID int(11),

in pMaterialID int(11),

in pVisitorAttendanceID int(11),

in pCreatedOn datetime,

in pTypeFlg varchar(10)

)

BEGIN

if pTypeFlg= 'ADD' Then

start transaction;

INSERT INTO materialvisitor

(

OrgID,

MaterialID,

VisitorAttendanceID,

CreatedOn

)

VALUES

(

pOrgID,

pMaterialID,

pVisitorAttendanceID,

pCreatedOn

);

commit;

elseif pTypeFlg='FETCH' Then

select V.VisitorID,V.FullName,V.CompanyName,V.Address,V.MobileNo,V.Email,V.Photo,

VA.MeetingID,VA.VisitorType,VA.MeetToUserID,VA.CheckInTime,VA.CheckInLocation,VA.CheckOutTime,VA.CheckOutLocation

from materialvisitor MV

LEFT JOIN visitorattendance VA ON MV.VisitorAttendanceID = VA.VisitorAttendanceID

LEFT JOIN visitor V ON V.VisitorID = VA.VisitorID

where MV.MaterialID = pMaterialID and MV.OrgID = pOrgID

order by MV.CreatedOn desc;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShowPolicyNoticePeriod` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`%` PROCEDURE `nex\_AddUpdateShowPolicyNoticePeriod`(

IN `pPolicyNPID` INT(11),

IN `pOrgID` INT(11),

IN `pNoticeDay` INT(11),

IN `pIsNormalOrBusiness` CHAR(1),

IN `pIsActive` char(1),

IN `pTypeFlg` INT(11)

)

BEGIN

DECLARE EXIT HANDLER FOR 1452 SELECT 'InvalidOrganizationID';

if pTypeFlg= 1 Then

select \* from policy\_notice\_period where OrgID=pOrgID;

elseif pTypeFlg= 2 Then

INSERT INTO policy\_notice\_period

(

OrgID,

NoticeDay,

IsNormalOrBusiness,

IsActive

)

VALUES

(

pOrgID,

pNoticeDay,

pIsNormalOrBusiness,

pIsActive

);

select 'inserted' as result;

elseif pTypeFlg= 3 Then

if(pIsActive='Y') Then

Update policy\_notice\_period SET

OrgID = pOrgID,

NoticeDay= pNoticeDay,

IsNormalOrBusiness=pIsNormalOrBusiness,

IsActive=pIsActive

Where PolicyNPID = pPolicyNPID;

else

Update policy\_notice\_period SET

IsActive='N'

Where PolicyNPID = pPolicyNPID;

end if;

select 'updated' as result;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShowPolicyPayrollClosing` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`%` PROCEDURE `nex\_AddUpdateShowPolicyPayrollClosing`(

IN `pPolicyPCID` INT(11),

IN `pOrgID` INT(11),

IN `pClosingOption` CHAR(3),

IN `pNoOfDays` INT(11),

IN `pIsActive` CHAR(1),

IN `pModified` DATETIME,

IN `pTypeFlg` INT(11)

)

BEGIN

if pTypeFlg= 1 Then

select \* from policy\_payroll\_closing where OrgID=pOrgID;

elseif pTypeFlg= 2 Then

INSERT INTO policy\_payroll\_closing

(

OrgID,

ClosingOption,

NoOfDays,

IsActive,

Modified

)

VALUES

(

pOrgID,

pClosingOption,

pNoOfDays,

pIsActive,

pModified

);

elseif pTypeFlg= 3 Then

Update policy\_payroll\_closing SET

OrgID = pOrgID,

ClosingOption= pClosingOption,

NoOfDays=pNoOfDays,

IsActive= pIsActive,

Modified = pModified

Where PolicyPCID = pPolicyPCID;

elseif pTypeFlg= 4 Then

Update policy\_payroll\_closing SET

IsActive= pIsActive,

Modified = pModified

Where OrgID = pOrgID;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShowPolicyPayrollDuration` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShowPolicyPayrollDuration`(

IN `pPolicyPDID` INT(11),

IN `pOrgID` INT(11),

IN `pForYear` INT(11),

IN `pMonth1Start` DATE,

IN `pMonth1End` DATE,

IN `pMonth2Start` DATE,

IN `pMonth2End` DATE,

IN `pMonth3Start` DATE,

IN `pMonth3End` DATE,

IN `pMonth4Start` DATE,

IN `pMonth4End` DATE,

IN `pMonth5Start` DATE,

IN `pMonth5End` DATE,

IN `pMonth6Start` DATE,

IN `pMonth6End` DATE,

IN `pMonth7Start` DATE,

IN `pMonth7End` DATE,

IN `pMonth8Start` DATE,

IN `pMonth8End` DATE,

IN `pMonth9Start` DATE,

IN `pMonth9End` DATE,

IN `pMonth10Start` DATE,

IN `pMonth10End` DATE,

IN `pMonth11Start` DATE,

IN `pMonth11End` DATE,

IN `pMonth12Start` DATE,

IN `pMonth12End` DATE,

IN `pIsActive` CHAR(1),

IN `pTypeFlg` INT(11)

)

BEGIN

declare CURMONTH,CURYEAR,CURRENTCOUNT int(11);

declare startMonthField,endMonthField varchar(50);

DECLARE varEndDate,resultEndDate,finalEndDate DATE DEFAULT NULL;

DECLARE EXIT HANDLER FOR 1452 SELECT 'InvalidOrganizationID';

if pTypeFlg= 1 Then

select \* from policy\_payroll\_duration where OrgID=pOrgID AND ForYear = pForYear;

elseif pTypeFlg= 2 Then

INSERT INTO policy\_payroll\_duration

(

OrgID,

ForYear,

Month1Start,

Month1End,

Month2Start,

Month2End,

Month3Start,

Month3End,

Month4Start,

Month4End,

Month5Start,

Month5End,

Month6Start,

Month6End,

Month7Start,

Month7End,

Month8Start,

Month8End,

Month9Start,

Month9End,

Month10Start,

Month10End,

Month11Start,

Month11End,

Month12Start,

Month12End,

IsActive

)

VALUES

(

pOrgID,

pForYear,

pMonth1Start,

pMonth1End,

pMonth2Start,

pMonth2End,

pMonth3Start,

pMonth3End,

pMonth4Start,

pMonth4End,

pMonth5Start,

pMonth5End,

pMonth6Start,

pMonth6End,

pMonth7Start,

pMonth7End,

pMonth8Start,

pMonth8End,

pMonth9Start,

pMonth9End,

pMonth10Start,

pMonth10End,

pMonth11Start,

pMonth11End,

pMonth12Start,

pMonth12End,

pIsActive

);

select 'inserted' as 'result';

elseif pTypeFlg= 3 Then

Update policy\_payroll\_duration SET

OrgID = pOrgID,

ForYear = pForYear,

Month1Start = pMonth1Start,

Month1End = pMonth1End,

Month2Start = pMonth2Start,

Month2End = pMonth2End,

Month3Start = pMonth3Start,

Month3End = pMonth3End,

Month4Start = pMonth4Start,

Month4End = pMonth4End,

Month5Start = pMonth5Start,

Month5End = pMonth5End,

Month6Start = pMonth6Start,

Month6End = pMonth6End,

Month7Start = pMonth7Start,

Month7End = pMonth7End,

Month8Start = pMonth8Start,

Month8End = pMonth8End,

Month9Start = pMonth9Start,

Month9End = pMonth9End,

Month10Start = pMonth10Start,

Month10End = pMonth10End,

Month11Start = pMonth11Start,

Month11End = pMonth11End,

Month12Start = pMonth12Start,

Month12End = pMonth12End,

IsActive = pIsActive

Where PolicyPDID = pPolicyPDID;

select 'updated' as 'result';

/\* 4 for for only current month duration \*/

elseif pTypeFlg= 4 Then

SELECT MONTH(CONVERT\_TZ(NOW(),'+00:00','+05:30')) INTO CURMONTH;

SELECT YEAR(CONVERT\_TZ(NOW(),'+00:00','+05:30')) INTO CURYEAR;

SET startMonthField= (concat("Month",CURMONTH,"Start"));

SET endMonthField = (concat("Month",CURMONTH,"End"));

/\* SET @stmtCurentChk = concat("SELECT COUNT(PolicyPDID) INTO @CURRENTCOUNT\_TMP FROM policy\_payroll\_duration

WHERE DATE(CONVERT\_TZ(NOW(),'+00:00','+05:30')) BETWEEN ",startMonthField," AND ",endMonthField," AND OrgID=",pOrgID);

PREPARE currentDurationExistStmt FROM @stmtCurentChk;

EXECUTE currentDurationExistStmt;

SET CURRENTCOUNT = @CURRENTCOUNT\_TMP;

IF CURRENTCOUNT = 1 THEN\*/

SET @stmt = CONCAT('SELECT OrgID,',startMonthField," as `Start`, ",endMonthField," as `End`, IsActive FROM policy\_payroll\_duration WHERE

OrgID=",pOrgID,' AND ForYear=',CURYEAR);

PREPARE currentDurationStmt FROM @stmt;

EXECUTE currentDurationStmt;

/\* select @stmt;\*/

elseif pTypeFlg= 5 Then

SET CURMONTH=1;

myWhileLoop: WHILE (CURMONTH <= 12) DO

SET startMonthField= (concat("Month",CURMONTH,"Start"));

SET endMonthField = (concat("Month",CURMONTH,"End"));

SET @stmt2 = CONCAT("SET @varEndDate=","(SELECT `",endMonthField,"` "," FROM policy\_payroll\_duration

WHERE OrgID='",pOrgID,"' AND '",pMonth1Start,"' BETWEEN ",startMonthField," AND ",endMonthField," ORDER BY ForYear DESC limit 0,1)");

/\*select @stmt2;\*/

PREPARE endDateStmt FROM @stmt2;

EXECUTE endDateStmt;

DEALLOCATE PREPARE endDateStmt;

SET resultEndDate = @varEndDate;

IF(resultEndDate IS NOT NULL) THEN

SELECT resultEndDate INTO finalEndDate;

LEAVE myWhileLoop;

END IF;

SET CURMONTH = CURMONTH + 1;

END WHILE;

IF(finalEndDate IS NOT NULL) THEN

SELECT finalEndDate AS EndDate;

ELSE SELECT '' AS EndDate;

END IF;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShowPolicySandwich` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShowPolicySandwich`(

IN `pPolicySWID` INT(11),

IN `pOrgID` INT(11),

IN `pApplicableFor` VARCHAR(50),

IN `pIsActive` CHAR(1),

IN `pCreated` DATETIME,

IN `pModified` DATETIME,

IN `pTypeFlg` INT(11)

)

BEGIN

DECLARE EXIT HANDLER FOR 1452 SELECT 'InvalidOrganizationID';

IF pTypeFlg= 1 THEN

SELECT \* from policy\_leave\_sandwich where OrgID=pOrgID;

ELSEIF pTypeFlg= 2 THEN

INSERT INTO policy\_leave\_sandwich

(

OrgID,

ApplicableFor,

IsActive,

Created,

Modified

)

VALUES

(

pOrgID,

pApplicableFor,

pIsActive,

pCreated,

pModified

);

select 'inserted' as 'result';

ELSEIF pTypeFlg= 3 THEN

Update policy\_leave\_sandwich SET

OrgID = pOrgID,

ApplicableFor = pApplicableFor,

IsActive= pIsActive,

Modified= pModified

Where PolicySWID = pPolicySWID;

select 'updated' as 'result';

END IF;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShowPolicyShiftDuration` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`%` PROCEDURE `nex\_AddUpdateShowPolicyShiftDuration`(

IN `pPolicyShiftID` INT(11),

IN `pOrgID` INT(11),

IN `pHasDoubleShift` CHAR(1),

in `pDblShiftDuration` varchar(10),

IN `pShiftDuration` varchar(10),

IN `pModified` DATETIME,

IN `pIsActive` CHAR(1),

IN `pTypeFlg` INT(11)

)

BEGIN

DECLARE EXIT HANDLER FOR 1452 SELECT 'InvalidOrganizationID';

if pTypeFlg= 1 Then

select \* from policy\_shift\_duration where OrgID=pOrgID;

elseif pTypeFlg= 2 Then

INSERT INTO policy\_shift\_duration

(

OrgID,

ShiftDuration,

Modified,

IsActive

)

VALUES

(

pOrgID,

pShiftDuration,

pModified,

pIsActive

);

select 'inserted' as 'result';

elseif pTypeFlg= 3 Then

Update policy\_shift\_duration SET

OrgID = pOrgID,

ShiftDuration= pShiftDuration,

Modified= pModified,

IsActive = pIsActive

Where PolicyShiftID = pPolicyShiftID;

select 'updated' as 'result';

/\* 4 for All fields of Dbl Shift \*/

elseif pTypeFlg= 4 Then

Update policy\_shift\_duration SET

DblShiftDuration= pDblShiftDuration,

HasDoubleShift = pHasDoubleShift,

Modified= pModified

Where PolicyShiftID = pPolicyShiftID;

select 'updated' as 'result';

elseif pTypeFlg= 5 Then

Update policy\_shift\_duration SET

HasDoubleShift = pHasDoubleShift,

Modified= pModified

Where PolicyShiftID = pPolicyShiftID;

select 'updated' as 'result';

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShowVisitorSettings` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShowVisitorSettings`(

IN `pVisitorSettingID` INT(11),

IN `pOrgID` INT(11) ,

IN `pDynamicQR` char(1),

IN `pStaticQR` char(1),

IN `pTypeFlag` varchar(30)

)

BEGIN

CASE pTypeFlag

WHEN 'GET' THEN

SELECT \* FROM visitorsetting WHERE OrgID=pOrgID;

WHEN 'INSERT' THEN

INSERT INTO visitorsetting (OrgID,DynamicQR,StaticQR) VALUES (pOrgID, pDynamicQR, pStaticQR);

WHEN 'UPDATE' THEN

UPDATE visitorsetting SET DynamicQR=pDynamicQR, StaticQR=pStaticQR WHERE VisitorSettingID=pVisitorSettingID;

END CASE;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShowVisitorStaticQrOption` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShowVisitorStaticQrOption`(

IN `pVisitorStaticQrOptionID` INT(11),

IN `pOrgID` INT(11) ,

IN `pNoOfQrCard` char(10),

IN `pCardNoStart` char(10),

IN `pTypeFlag` varchar(30)

)

BEGIN

CASE pTypeFlag

WHEN 'INSERT' THEN

INSERT INTO visitorstaticqroption (OrgID,NoOfQrCard,CardNoStart) VALUES (pOrgID,pNoOfQrCard,pCardNoStart);

WHEN 'GET' THEN

SELECT \* FROM visitorstaticqroption WHERE OrgID=pOrgID;

WHEN 'CHK\_STC\_CARD\_IN\_USE' THEN

Select VA.StaticQrCardNo,VA.VisitAttenStatus

FROM visitorattendance VA

LEFT JOIN visitor V ON VA.VisitorID = V.VisitorID

WHERE VA.StaticQrCardNo = pCardNoStart AND V.OrgID=pOrgID

AND VA.VisitAttenStatus NOT IN('CheckedOut-NotVarified','CheckedOut','Cancelled');

WHEN 'UPDATE\_QR\_CARDNO' THEN

UPDATE visitorattendance SET StaticQrCardNo=pCardNoStart

WHERE VisitorAttendanceID=pVisitorStaticQrOptionID;

WHEN 'UPDATE' THEN

UPDATE visitorstaticqroption SET NoOfQrCard=pNoOfQrCard, CardNoStart=pCardNoStart WHERE VisitorStaticQrOptionID=pVisitorStaticQrOptionID;

END CASE;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_calendar` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_calendar`(

In pCalendarID int(11),

in pCalendarName varchar(255),

IN pCalendarDesc text,

IN pOrgID int(11),

IN pHolidayID varchar(255),

IN pWeekendID varchar(255),

IN pUserID text,

in pEffectedFrom datetime,

in pAddedOn datetime,

in pTypeFlg int(11)

)

BEGIN

DECLARE i,NOOFHLDY,HLDY,NOOFHWEND,WEND,NOOFUSERCAL,EXTCALCOUNT int;

DECLARE USERCAL text;

/\*Insert into calendar table\*/

if pTypeFlg= 1 Then

INSERT INTO calendar

(

CalendarName,

CalendarDesc,

OrgID

)

VALUES

(

pCalendarName,

pCalendarDesc,

pOrgID

);

select LAST\_INSERT\_ID() as CalendarID;

/\*Update calendar table\*/

elseif pTypeFlg= 2 Then

UPDATE calendar C SET

C.CalendarName = pCalendarName,

C.CalendarDesc = pCalendarDesc

WHERE C.CalendarID = pCalendarID;

/\*Delete calender\*/

elseif pTypeFlg = 3 Then

delete from calendarholiday where CalendarID = pCalendarID;

delete from calendarweekends where CalendarID = pCalendarID;

delete from usercalendar where CalendarID = pCalendarID;

delete from calendar where CalendarID = pCalendarID;

/\*View calender by CalendarID\*/

elseif pTypeFlg = 4 Then

select C.CalendarID,C.CalendarName,C.CalendarDesc from calendar C

where C.CalendarID = pCalendarID;

/\*Search by calender by CalendarName\*/

elseif pTypeFlg = 5 Then

select \* from calendar where CalendarName = pCalendarName and OrgID = pOrgID;

/\*List of all calender\*/

elseif pTypeFlg = 6 Then

select \* from calendar where OrgID = pOrgID order by CalendarID desc;

/\*List of all Holidays of a calender\*/

elseif pTypeFlg = 7 Then

select H.HolidayID,H.HolidayName,H.StartDate from calendarholiday CH

LEFT JOIN holiday H ON CH.HolidayID = H.HolidayID

where CH.CalendarID = pCalendarID;

/\*List of all Weekends of a calender\*/

elseif pTypeFlg = 8 Then

select W.WeekendsID,W.DayName from calendarweekends CW

LEFT JOIN weekends W ON CW.WeekendID = W.WeekendsID

where CW.CalendarID = pCalendarID;

/\*List of all Users of a calender\*/

elseif pTypeFlg = 9 Then

select U.UserID,concat(U.FName,' ',U.LName) UName, U.EmployeeID,UC.EffectedFrom

from usercalendar UC

LEFT JOIN userpersonalinfo U ON UC.UserID = U.UserID

where UC.CalendarID = pCalendarID;

/\*Insert into related tables\*/

elseif pTypeFlg = 10 Then

start transaction;

/\*Delete from all related tables\*/

delete from calendarholiday where CalendarID = pCalendarID;

delete from calendarweekends where CalendarID = pCalendarID;

delete from usercalendar where CalendarID = pCalendarID;

/\*Insert into calendarholiday table\*/

if pHolidayID != '' then

call splitter(pHolidayID,',');

select count(SplitValue) INTO NOOFHLDY from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFHLDY DO

select SplitValue INTO HLDY from splitResultsTmpTable where TmpID = i;

INSERT INTO calendarholiday

(

CalendarID,

HolidayID

)

VALUES

(

pCalendarID,

HLDY

);

SET i = i + 1;

END WHILE;

end if;

if pWeekendID != '' then

/\*Insert into calendarweekends table\*/

call splitter(pWeekendID,',');

select count(SplitValue) INTO NOOFHWEND from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFHWEND DO

select SplitValue INTO WEND from splitResultsTmpTable where TmpID = i;

INSERT INTO calendarweekends

(

CalendarID,

WeekendID

)

VALUES

(

pCalendarID,

WEND

);

SET i = i + 1;

END WHILE;

end if;

/\*Insert into usercalendar table\*/

if pUserID!='' then

call splitter(pUserID,',');

select count(SplitValue) INTO NOOFUSERCAL from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFUSERCAL DO

select SplitValue INTO USERCAL from splitResultsTmpTable where TmpID = i;

INSERT INTO usercalendar

(

CalendarID,

UserID,

EffectedFrom,

AddedOn

)

VALUES

(

pCalendarID,

stringSplit(USERCAL,'~',1),

stringSplit(USERCAL,'~',2),

pAddedOn

);

SET i = i + 1;

END WHILE;

end if;

commit;

elseif pTypeFlg = 11 Then

if pCalendarID != 0 then

select if(count(UC.CalendarID)>0,'Y','N') as isCalendar,concat(U.FName,' ',U.LName) FullName,U.EmployeeID

from usercalendar UC LEFT JOIN userpersonalinfo U ON UC.UserID = U.UserID

where UC.UserID = pUserID and UC.EffectedFrom = pEffectedFrom and UC.CalendarID != pCalendarID;

else

select if(count(UC.CalendarID)>0,'Y','N') as isCalendar,concat(U.FName,' ',U.LName) FullName,U.EmployeeID

from usercalendar UC LEFT JOIN userpersonalinfo U ON UC.UserID = U.UserID

where UC.UserID = pUserID and UC.EffectedFrom = pEffectedFrom;

end if;

elseif pTypeFlg = 12 Then

call splitter(pUserID,',');

select count(SplitValue) INTO NOOFUSERCAL from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFUSERCAL DO

select SplitValue INTO USERCAL from splitResultsTmpTable where TmpID = i;

select count(UserCalendarID) INTO EXTCALCOUNT from usercalendar UC where UC.CalendarID=pCalendarID

and UC.UserID=stringSplit(USERCAL,'~',1) and date(UC.EffectedFrom)=date(stringSplit(USERCAL,'~',2));

if EXTCALCOUNT = 0 then

INSERT INTO usercalendar

(

CalendarID,

UserID,

EffectedFrom,

AddedOn

)

VALUES

(

pCalendarID,

stringSplit(USERCAL,'~',1),

stringSplit(USERCAL,'~',2),

pAddedOn

);

end if;

SET i = i + 1;

END WHILE;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_calendarholiday` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_calendarholiday`(

IN pCalHolidayID int(11),

IN pCalendarID int(11),

IN pHolidayID int(11),

IN pTypeFlg int(11)

)

BEGIN

DECLARE CALHOLIDAYID int;

if pTypeFlg= 1 Then

INSERT INTO calendarholiday

(

CalendarID,

HolidayID

)

VALUES

(

pCalendarID,

pHolidayID

);

SET CALHOLIDAYID = LAST\_INSERT\_ID();

select CALHOLIDAYID;

elseif pTypeFlg= 2 Then

delete from calendarholiday where CalHolidayID= pCalHolidayID;

elseif pTypeFlg= 3 Then

select count(CalHolidayID) as NoOfCalendar from calendarholiday where CalendarID = pCalendarID and HolidayID = pHolidayID;

elseif pTypeFlg= 4 Then

delete from calendarholiday where CalendarID= pCalendarID;

elseif pTypeFlg= 5 Then

select CalendarID from calendarholiday where HolidayID = pHolidayID;

End if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_calendarweekends` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_calendarweekends`(

IN pCalWeekendID int(11),

IN pCalendarID int(11),

IN pWeekendID int(11),

IN pTypeFlg int(11)

)

BEGIN

DECLARE CALWEEKENDID int;

if pTypeFlg= 1 Then

INSERT INTO calendarweekends

(

CalendarID,

WeekendID

)

VALUES

(

pCalendarID,

pWeekendID

);

SET CALWEEKENDID = LAST\_INSERT\_ID();

select CALWEEKENDID;

elseif pTypeFlg= 2 Then

delete from calendarweekends where CalWeekendID= pCalWeekendID;

elseif pTypeFlg= 3 Then

select count(CalWeekendID) from calendarweekends where CalendarID = pCalendarID and WeekendID = pWeekendID;

elseif pTypeFlg= 4 Then

delete from calendarweekends where CalendarID= pCalendarID;

End if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_compoffapplied` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_compoffapplied`(

in pCompOffID text,

in pUserID int(11),

in pCompOffDate date,

in pRequestType varchar(6),

in pCompOffReason text,

in pAppliedDate datetime,

in pRequestStatus varchar(10),

in pStatusUpdateDate datetime,

in pStatusUpdteBy int(11),

in pTypeFlg smallint(1)

)

BEGIN

DECLARE COMPOFID int;

DECLARE i,NOOFCOMP,SELCOMPOFF int;

start transaction;

if pTypeFlg= 1 Then

if pRequestType = 'Payout' then

INSERT INTO compoffapplied

(

UserID,

CompOffDate,

RequestType,

CompOffReason,

AppliedDate,

RequestStatus,

StatusUpdateDate,

StatusUpdteBy

)

VALUES

(

pUserID,

pCompOffDate,

pRequestType,

pCompOffReason,

pAppliedDate,

pRequestStatus,

pStatusUpdateDate,

pStatusUpdteBy

);

/\*SET COMPOFID = LAST\_INSERT\_ID();\*/

elseif pRequestType = 'Leave' then

select CompOffID INTO SELCOMPOFF from compoffapplied CMOF where CMOF.UserID=pUserID AND CMOF.CompOffDate =pCompOffDate;

if SELCOMPOFF > 0 then

/\*If this is approved Off-day(Payout)\*/

INSERT INTO compoffappliedlog

(

CompOffID,

RequestType,

AppliedDate,

RequestStatus,

StatusUpdateDate,

StatusUpdteBy

)

select CompOffID,'Payout',AppliedDate,RequestStatus,StatusUpdateDate,StatusUpdteBy from compoffapplied where CompOffID=SELCOMPOFF;

UPDATE compoffapplied SET

RequestType='Leave',

AppliedDate=pAppliedDate,

RequestStatus=pRequestStatus,

StatusUpdateDate=pStatusUpdateDate,

StatusUpdteBy=pStatusUpdteBy

WHERE CompOffID=SELCOMPOFF;

else

/\*If this is approved OD\*/

INSERT INTO compoffapplied

(

UserID,

CompOffDate,

RequestType,

CompOffReason,

AppliedDate,

RequestStatus,

StatusUpdateDate,

StatusUpdteBy

)

VALUES

(

pUserID,

pCompOffDate,

pRequestType,

pCompOffReason,

pAppliedDate,

pRequestStatus,

pStatusUpdateDate,

pStatusUpdteBy

);

end if;

end if;

elseif pTypeFlg= 2 Then

call splitter(pCompOffID,',');

select count(SplitValue) INTO NOOFCOMP from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFCOMP DO

Update compoffapplied Set

RequestStatus= pRequestStatus,

StatusUpdateDate= pStatusUpdateDate,

StatusUpdteBy= pStatusUpdteBy

Where CompOffID= (select SplitValue from splitResultsTmpTable where TmpID = i);

SET i = i + 1;

END WHILE;

elseif pTypeFlg= 3 Then

Select CMOF.CompOffID,CMOF.UserID,CMOF.CompOffDate,CMOF.RequestType as OrigRequestType,if(CMOF.RequestType='Leave','CO Request','Off Day Present') as RequestType,CMOF.CompOffReason,CMOF.AppliedDate,

CMOF.RequestStatus,CMOF.StatusUpdateDate,CMOF.StatusUpdteBy

from compoffapplied CMOF where CMOF.UserID = pUserID order by CMOF.CompOffDate desc,CMOF.CompOffID desc;

Else

call splitter(pCompOffID,',');

Select CMOF.CompOffID,CMOF.UserID,CMOF.CompOffDate,CMOF.RequestType as OrigRequestType,if(CMOF.RequestType='Leave','CO Request','Off Day Present') as RequestType,CMOF.CompOffReason,CMOF.AppliedDate,

CMOF.RequestStatus,CMOF.StatusUpdateDate,CMOF.StatusUpdteBy

from compoffapplied CMOF

where CMOF.CompOffID IN (select SplitValue from splitResultsTmpTable);

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_department` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_department`(

in pDeptID text,

in pDeptPrtID int(11),

in pDepartmentName varchar(50),

in pDepartmentDesc text,

in pOrgID int(11),

in pDeptManagerID int(11),

in pTypeFlg int(11)

)

BEGIN

DECLARE LASTDEPTID,USERWRKINFO,USERROLEID,OLDUSEPERVISORID,OLDUSEPERVISOROFDEPTS int default 0;

DECLARE OLDUSEPERVISORROLE varchar(10);

DECLARE EXIT HANDLER FOR 1451

SELECT 'FKC-ERROR'; /\*Foreign Key Constraint Error\*/

start transaction;

if pTypeFlg= 1 Then

INSERT INTO department

(

DeptPrtID,

DepartmentName,

DepartmentDesc,

OrgID

)

VALUES

(

pDeptPrtID,

pDepartmentName,

pDepartmentDesc,

pOrgID

);

SET LASTDEPTID = LAST\_INSERT\_ID();

select LASTDEPTID;

elseif pTypeFlg= 2 Then

Update department Set

DepartmentName= pDepartmentName,

DepartmentDesc= pDepartmentDesc

Where DeptID= pDeptID and OrgID= pOrgID;

elseif pTypeFlg= 3 Then

delete from deptmanager where DeptID = pDeptID;

delete from department where DeptID = pDeptID and OrgID= pOrgID;

select 'SUCCESS';

elseif pTypeFlg= 4 Then

select DeptManagerID into OLDUSEPERVISORID from deptmanager where DeptID = pDeptID;

select r.RoleName into OLDUSEPERVISORROLE from userrole r left join userworkinfo uw on r.RoleID = uw.RoleID where uw.UserID = OLDUSEPERVISORID;

select count(DeptID) into OLDUSEPERVISOROFDEPTS from deptmanager where DeptManagerID = OLDUSEPERVISORID;

if (OLDUSEPERVISORROLE = 'SUPERVISOR' and OLDUSEPERVISOROFDEPTS = 1) then

delete from userworkinfo where UserID = OLDUSEPERVISORID;

end if;

delete from deptmanager where DeptID = pDeptID;

INSERT INTO deptmanager

(

DeptID,

DeptManagerID

)

VALUES

(

pDeptID,

pDeptManagerID

);

select count(UserWorkInfoID) into USERWRKINFO from userworkinfo where UserID = pDeptManagerID;

if USERWRKINFO = 0 then

select RoleID into USERROLEID from userrole where RoleName = 'SUPERVISOR' and OrgID = pOrgID;

if USERROLEID = 0 then

INSERT INTO userrole

(

RoleName,

Actions,

OrgID

)

VALUES

(

'SUPERVISOR',

NULL,

pOrgID

);

set USERROLEID = LAST\_INSERT\_ID();

end if;

INSERT INTO userworkinfo

(

UserID,

RoleID,

TeamID,

Designation

)

VALUES

(

pDeptManagerID,

USERROLEID,

0,

'Supervisor'

);

end if;

elseif pTypeFlg= 5 Then

Select \* from department where DepartmentName= pDepartmentName and OrgID= pOrgID;

elseif pTypeFlg= 6 Then

Select DepartmentName as DepartmentName from department where DeptID = pDeptID;

elseif pTypeFlg= 7 Then

Select DeptID, DepartmentName from department where OrgID= pOrgID;

elseif pTypeFlg = 8 Then

select D.DeptID,D.DeptPrtID,D.DepartmentName,D.DepartmentDesc,

DM.DeptManagerID,U.Email,concat(U.FName,' ',U.LName) as UName from department D

LEFT JOIN deptmanager DM ON D.DeptID = DM.DeptID

LEFT JOIN userpersonalinfo U ON U.UserID = DM.DeptManagerID

where D.OrgID = pOrgID and D.DeptPrtID IN (pDeptID);

elseif pTypeFlg = 9 Then

select D.DeptID,D.DeptPrtID,D.DepartmentName,D.DepartmentDesc,D.OrgID,

DM.DeptManagerID,U.Email,concat(U.FName,' ',U.LName) as UName from department D

LEFT JOIN deptmanager DM ON D.DeptID = DM.DeptID

LEFT JOIN userpersonalinfo U ON U.UserID = DM.DeptManagerID

where D.DeptID = pDeptID;

else

Select \* from department where OrgID= pOrgID;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_departmentholiday` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_departmentholiday`(

IN pDeptHolidayID int(11),

IN pDeptID int(11),

IN pHolidayID int(11),

IN pOrgID int(11),

IN pStartDate date,

IN pShowYear varchar(4),

IN pTypeFlg int(11)

)

BEGIN

DECLARE DEPARTHOLIDAYID int;

if pTypeFlg= 1 Then

INSERT INTO departmentholiday

(

OrgID,

DeptID,

HolidayID

)

VALUES

(

pOrgID,

pDeptID,

pHolidayID

);

SET DEPARTHOLIDAYID = LAST\_INSERT\_ID();

select DEPARTHOLIDAYID;

elseif pTypeFlg= 2 Then

Delete from departmentholiday

Where DeptHolidayID= pDeptHolidayID;

elseif pTypeFlg= 3 Then

If pShowYear='' Then

select H.HolidayName,H.Holidaydescription,H.StartDate,H.EndDate,DH.DeptHolidayID,DH.DeptID,DH.OrgID

from departmentholiday DH LEFT JOIN holiday H

ON DH.HolidayID = H.HolidayID

where DH.DeptID = pDeptID AND H.OrgID = pOrgID

order by H.StartDate asc;

Else

select H.\*,DH.\*

from departmentholiday DH LEFT JOIN holiday H

ON DH.HolidayID = H.HolidayID

where DH.DeptID = pDeptID AND H.OrgID = pOrgID AND Year(H.StartDate)= pShowYear

order by H.StartDate desc;

End If;

elseif pTypeFlg= 4 Then

select DeptHolidayID

from departmentholiday

Where DeptID= pDeptID AND HolidayID = pHolidayID;

elseif pTypeFlg= 5 Then

select DH.DeptHolidayID,DH.DeptID, D.DepartmentName

from departmentholiday DH LEFT JOIN Department D

ON DH.DeptID = D.DeptID

Where DH.HolidayID= pHolidayID;

elseif pTypeFlg= 6 Then

select DH.DeptHolidayID,DH.DeptID,H.HolidayID,H.StartDate

from departmentholiday DH LEFT JOIN Holiday H

ON DH.HolidayID = H.HolidayID

Where H.OrgID=pOrgID AND DH.DeptID=pDeptID AND H.StartDate=pStartDate;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_departmentweekends` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_departmentweekends`(

IN pDepartmentWeekendsID int(11),

IN pDeptID int(11),

IN pWeekendsID int(11),

IN pOrgID int(11),

IN pTypeFlg int(11)

)

BEGIN

DECLARE DEPTWEEKENDSID int;

if pTypeFlg= 1 Then

INSERT INTO departmentweekends

(

DeptID,

WeekendsID

)

VALUES

(

pDeptID,

pWeekendsID

);

SET DEPTWEEKENDSID = LAST\_INSERT\_ID();

select DEPTWEEKENDSID;

elseif pTypeFlg= 2 Then

Delete from departmentweekends

Where DepartmentWeekendsID= pDepartmentWeekendsID;

elseif pTypeFlg= 3 Then

select W.\*,WE.\*

from departmentweekends WE LEFT JOIN weekends W

ON WE.WeekendsID = W.WeekendsID

where WE.DeptID = pDeptID AND W.OrgID = pOrgID

order by W.DayName;

elseif pTypeFlg= 4 Then

select WE.DepartmentWeekendsID

from departmentweekends WE

where WE.DeptID = pDeptID AND WE.WeekendsID = pWeekendsID;

elseif pTypeFlg= 5 Then

Update departmentweekends Set

WeekendsID= pWeekendsID

Where DepartmentWeekendsID = pDepartmentWeekendsID AND DeptID= pDeptID;

elseif pTypeFlg= 6 Then

select WE.DepartmentWeekendsID, WE.DeptID, WE.WeekendsID

from departmentweekends WE

where WE.DeptID = pDeptID;

elseif pTypeFlg= 7 Then

select W.DayName, WE.DepartmentWeekendsID, WE.WeekendsID

from weekends W JOIN departmentweekends WE

ON W.WeekendsID = WE.WeekendsID JOIN department d

ON d.DeptID = WE.DeptID

where WE.DeptID = pDeptID;

elseif pTypeFlg= 8 Then

Delete from departmentweekends

Where DeptID= pDeptID;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_department\_user` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_department\_user`(

in pUserID text,

in pDeptID int(11),

in pOrgID int(11),

in pTypeFlg int(11)

)

BEGIN

declare USERDEPTID int default 0;

DECLARE i,NOOFUSER int;

if pTypeFlg= 1 Then

select DeptID into USERDEPTID from userpersonalinfo

where UserID = pUserID and OrgID = pOrgID;

if USERDEPTID > 0 then

select 'EXIST-FALSE';

else

UPDATE userpersonalinfo SET DeptID = pDeptID WHERE UserID= pUserID;

select 'SUCCESS';

end if;

elseif pTypeFlg= 2 Then

UPDATE userpersonalinfo SET DeptID = NULL WHERE UserID= pUserID;

elseif pTypeFlg= 3 Then

call splitter(pUserID,',');

select count(SplitValue) INTO NOOFUSER from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFUSER DO

UPDATE userpersonalinfo SET DeptID = pDeptID WHERE UserID= (select SplitValue from splitResultsTmpTable where TmpID = i);

SET i = i + 1;

END WHILE;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_earlycheckoutapproval` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_earlycheckoutapproval`(

in pEarlyCheckoutApprvID int(11),

in pAttendanceID text,

in pUserID int(11),

in pSupUserID int(11),

in pAppliedDate datetime,

in pRequestStatus varchar(10),

in pStatusNote text,

in pStatusUpdateDate datetime,

in pStatusUpdteBy int(11),

in pOrgID int(11),

in pWorkDuration int(11),

in pLessDuration int(11),

in pTypeFlg smallint(1)

)

BEGIN

DECLARE isAlreadyRequested int;

DECLARE i,NOOFATTN int;

start transaction;

if pTypeFlg= 1 Then

select count(ECOA.EarlyCheckoutApprvID) INTO isAlreadyRequested

from earlycheckoutapproval ECOA where ECOA.AttendanceID = pAttendanceID

AND ECOA.OrgID = pOrgID;

if isAlreadyRequested = 0 then

INSERT INTO earlycheckoutapproval

(

AttendanceID,

UserID,

SupervisorUserID,

AppliedDate,

RequestStatus,

StatusNote,

StatusUpdateDate,

StatusUpdteBy,

OrgID,

WorkDuration,

LessDuration

)

VALUES

(

pAttendanceID,

pUserID,

pSupUserID,

pAppliedDate,

pRequestStatus,

pStatusNote,

pStatusUpdateDate,

pStatusUpdteBy,

pOrgID,

pWorkDuration,

pLessDuration

);

else

Update earlycheckoutapproval Set

AppliedDate= pAppliedDate,

RequestStatus= pRequestStatus

Where AttendanceID= pAttendanceID;

end if;

elseif pTypeFlg= 2 Then

call splitter(pAttendanceID,',');

select count(SplitValue) INTO NOOFATTN from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFATTN DO

Update earlycheckoutapproval Set

RequestStatus= pRequestStatus,

StatusNote= pStatusNote,

StatusUpdateDate= pStatusUpdateDate,

StatusUpdteBy= pStatusUpdteBy

where AttendanceID IN (select SplitValue from splitResultsTmpTable where TmpID = i);

SET i = i + 1;

END WHILE;

elseif pTypeFlg= 3 Then

select ECOA.\*,CONCAT(UP.FName,' ',UP.LName) as UserName,UP.EmployeeID as UserEmployeeID,UP.UserPicture,

Time(A.CheckInTime) as CheckInTime,

Time(A.CheckOutTime) CheckOutTime,

(TIMESTAMPDIFF(Second, A.CheckInTime, A.CheckOutTime)) TotalWorkedHours

from earlycheckoutapproval ECOA

LEFT JOIN attendance A ON ECOA.AttendanceID = A.AttendanceID

LEFT JOIN userpersonalinfo UP ON ECOA.UserID = UP.UserID

where ECOA.SupervisorUserID = pSupUserID and ECOA.RequestStatus = 'Applied';

elseif pTypeFlg= 4 Then

select ECOA.\*,CONCAT(UP.FName,' ',UP.LName) as UserName,UP.EmployeeID as UserEmployeeID,

Time(A.CheckInTime) as CheckInTime,

Time(A.CheckOutTime) CheckOutTime,

(TIMESTAMPDIFF(Hour, A.CheckInTime, A.CheckOutTime) + (TIMESTAMPDIFF(Minute, A.CheckInTime, A.CheckOutTime) - ((TIMESTAMPDIFF(Hour, A.CheckInTime, A.CheckOutTime))\*60))\*.01) TotalWorkedHours

from earlycheckoutapproval ECOA

LEFT JOIN attendance A ON ECOA.AttendanceID = A.AttendanceID

LEFT JOIN userpersonalinfo UP ON ECOA.UserID = UP.UserID

where ECOA.EarlyCheckoutApprvID = pEarlyCheckoutApprvID;

elseif pTypeFlg= 5 Then

select count(ECOA.EarlyCheckoutApprvID) as countPrevious

from earlycheckoutapproval ECOA where ECOA.UserID = pUserID

AND date(ECOA.AppliedDate) = date(pAppliedDate);

elseif pTypeFlg= 6 Then /\* added by Shams \*/

select A.LogTime

from attendance A where A.AttendanceID = pAttendanceID;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_EmpAuth` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_EmpAuth`(

IN pEmpAppAuthID int(11),

IN pOrgID int(11),

IN pByQRCode char(1),

IN pByODTrack char(1),

IN pODTrackUserType char(11),

IN pODTrackSelfieMode char(1),

IN pUserID int(11),

IN pTypeFlg smallint(1)

)

BEGIN

if pTypeFlg = 1 Then

SELECT QRCode, ODTrack, ODTrackUserType, ODTrackSelfieMode FROM empappauth WHERE OrgID=pOrgID;

elseif pTypeFlg = 2 Then

INSERT INTO empappauth

(

OrgID,

QRCode,

ODTrack,

ODTrackUserType,

ODTrackSelfieMode

)

VALUES

(

pOrgID,

pByQRCode,

pByODTrack,

pODTrackUserType,

ODTrackSelfieMode

);

elseif pTypeFlg = 3 Then

INSERT INTO empappauthuser

(

OrgID,

UserID

)

VALUES

(

pOrgID,

pUserID

);

elseif pTypeFlg = 4 Then

DELETE FROM empappauthuser WHERE OrgID=pOrgID;

elseif pTypeFlg = 5 Then

UPDATE empappauth SET QRCode=pByQRCode, ODTrack=pByODTrack, ODTrackUserType=pODTrackUserType, ODTrackSelfieMode=pODTrackSelfieMode WHERE OrgID=pOrgID;

elseif pTypeFlg = 6 Then

SELECT UserID FROM empappauthuser WHERE OrgID=pOrgID;

elseif pTypeFlg = 7 Then

SELECT UserID FROM empappauthuser WHERE UserID=pUserID AND OrgID=pOrgID;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_holiday` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_holiday`(

In pHolidayID int(11),

in pHolidayName varchar(255),

IN pHolidayDescription text,

in pStartDate date,

in pEndDate date,

in pOrgID int(11),

in pShowYear varchar(4),

in pTypeFlg int(11)

)

BEGIN

DECLARE LASTHOLIDAYID int;

if pTypeFlg= 1 Then

INSERT INTO holiday

(

HolidayName,

HolidayDescription,

StartDate,

EndDate,

OrgID

)

VALUES

(

pHolidayName,

pHolidayDescription,

pStartDate,

pEndDate,

pOrgID

);

SET LASTHOLIDAYID = LAST\_INSERT\_ID();

select LASTHOLIDAYID;

elseif pTypeFlg= 2 Then

UPDATE holiday H SET

H.HolidayName = pHolidayName,

H.HolidayDescription = pHolidayDescription,

H.StartDate = pStartDate,

H.EndDate = pEndDate

WHERE H.HolidayID = pHolidayID;

elseif pTypeFlg= 3 Then

if pShowYear='' Then

Select H.HolidayID,H.HolidayName, H.HolidayDescription, H.StartDate, H.EndDate, H.OrgID

from holiday H

where H.HolidayID=pHolidayID AND H.OrgID=pOrgID order by H.StartDate desc;

Else

Select H.\*

from holiday H

where H.HolidayID=pHolidayID AND H.OrgID=pOrgID AND Year(H.StartDate)= pShowYear

order by H.StartDate desc;

End If;

elseif pTypeFlg= 4 Then

Delete from departmentholiday Where HolidayID= pHolidayID;

Delete from holiday Where HolidayID= pHolidayID;

elseif pTypeFlg= 5 Then

Select H.HolidayID,H.HolidayName, H.HolidayDescription, H.StartDate, H.EndDate, H.OrgID

from holiday H

where H.OrgID=pOrgID order by H.StartDate desc;

elseif pTypeFlg= 6 Then

select H.HolidayName,H.HolidayID,H.Holidaydescription

from holiday H

where H.HolidayName = pHolidayName AND H.OrgID = pOrgID;

elseif pTypeFlg= 7 Then

Select H.HolidayID,H.HolidayName, H.HolidayDescription, H.StartDate, H.EndDate, H.OrgID

from holiday H

where H.OrgID=pOrgID AND StartDate=pStartDate order by H.StartDate desc;

Else

if pShowYear='' Then

Select H.HolidayID,H.HolidayName, H.HolidayDescription, H.StartDate, H.EndDate, H.OrgID

from holiday H

where H.OrgID=pOrgID order by H.StartDate desc;

Else

Select H.HolidayID,H.HolidayName, H.HolidayDescription, H.StartDate, H.EndDate, H.OrgID

from holiday H

where H.OrgID=pOrgID AND Year(H.StartDate)= pShowYear

order by H.StartDate desc;

End If;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_holiday2` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_holiday2`(

In pHolidayID int(11),

in pHolidayName varchar(255),

IN pHolidayDescription text,

in pStartDate date,

in pEndDate date,

in pOrgID int(11),

in pIsCloned char(1),

in pClonedParentHolidayID int(11),

in pClonedStatus char(1),

in pShowYear varchar(4),

in pTypeFlg int(11)

)

BEGIN

DECLARE LASTHOLIDAYID int;

DECLARE EXIT HANDLER FOR 1451

SELECT 'FKC-ERROR'; /\*Foreign Key Constraint Error\*/

if pTypeFlg= 1 Then

INSERT INTO holiday

(

HolidayName,

HolidayDescription,

StartDate,

EndDate,

OrgID,

IsCloned,

ClonedParentHolidayID,

ClonedStatus

)

VALUES

(

pHolidayName,

pHolidayDescription,

pStartDate,

pEndDate,

pOrgID,

pIsCloned,

pClonedParentHolidayID,

pClonedStatus

);

SET LASTHOLIDAYID = LAST\_INSERT\_ID();

select LASTHOLIDAYID;

elseif pTypeFlg= 2 Then

UPDATE holiday H SET

H.HolidayName = pHolidayName,

H.HolidayDescription = pHolidayDescription,

H.StartDate = pStartDate,

H.EndDate = pEndDate

WHERE H.HolidayID = pHolidayID;

elseif pTypeFlg= 3 Then

Select H.HolidayID,H.HolidayName, H.HolidayDescription, H.StartDate, H.EndDate, H.OrgID, H.IsCloned,

H.ClonedParentHolidayID, H.ClonedStatus

from holiday H

where H.HolidayID=pHolidayID AND H.OrgID=pOrgID;

elseif pTypeFlg= 4 Then

Delete from holiday Where HolidayID= pHolidayID;

select 'SUCCESS';

elseif pTypeFlg= 5 Then

Select H.HolidayID,H.HolidayName, H.HolidayDescription, H.StartDate, H.EndDate, H.OrgID,

H.IsCloned, H.ClonedParentHolidayID,H.ClonedStatus

from holiday H

where H.OrgID=pOrgID order by H.StartDate desc;

elseif pTypeFlg= 6 Then

select H.HolidayName,H.HolidayID,H.Holidaydescription

from holiday H

where H.HolidayName = pHolidayName

AND H.StartDate = pStartDate AND H.EndDate = pEndDate

AND H.OrgID = pOrgID;

elseif pTypeFlg= 7 Then

select H.HolidayName,H.HolidayID,H.Holidaydescription,H.StartDate,H.EndDate

from holiday H

where H.StartDate BETWEEN pStartDate AND pEndDate AND H.OrgID = pOrgID;

elseif pTypeFlg= 8 Then

select H.HolidayName,H.HolidayID,H.Holidaydescription,H.StartDate,H.EndDate

from holiday H

where year(H.StartDate) = pShowYear AND H.OrgID = pOrgID;

elseif pTypeFlg= 9 Then

UPDATE holiday H SET

H.ClonedStatus = 'P'

WHERE H.HolidayID = pHolidayID;

Else

if pShowYear='' Then

Select H.HolidayID,H.HolidayName, H.HolidayDescription, H.StartDate, H.EndDate, H.OrgID,

H.IsCloned, H.ClonedParentHolidayID, H.ClonedStatus

from holiday H

where H.OrgID=pOrgID order by H.StartDate desc;

Else

Select H.HolidayID,H.HolidayName, H.HolidayDescription, H.StartDate, H.EndDate, H.OrgID,

H.IsCloned, H.ClonedParentHolidayID, H.ClonedStatus

from holiday H

where H.OrgID=pOrgID AND Year(H.StartDate)= pShowYear

order by H.StartDate desc;

End If;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_kiosk` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_kiosk`(

in pKioskID int(11),

in pOrgID int(11),

in pUserID int(11),

in pEmail varchar (50),

in pKioskName varchar (50),

in pUniqueDeviceID varchar (50),

in pDeviceType varchar (50),

in pOTP varchar(10),

in pRegisteredOn datetime,

in pTypeFlg int(11)

)

BEGIN

if pTypeFlg= 1 Then

delete from kiosk where Email = pEmail and OTPSuccess='N' and IsActive='N';

INSERT INTO kiosk

(

OrgID,

UserID,

Email,

KioskName,

UniqueDeviceID,

OTP,

DeviceType

)

VALUES

(

pOrgID,

pUserID,

pEmail,

pKioskName,

pUniqueDeviceID,

pOTP,

pDeviceType

);

elseif pTypeFlg= 2 Then

Update kiosk k Set

k.OTP= pOTP

Where k.KioskID= pKioskID;

elseif pTypeFlg= 3 Then

Select KioskID,UniqueDeviceID from kiosk where Email= pEmail and isActive = 'Y';

elseif pTypeFlg= 4 Then

select KioskID,Email,UniqueDeviceID from kiosk where Email=pEmail and OTP=pOTP and isActive = 'N';

elseif pTypeFlg= 5 Then

Update kiosk k Set

k.OTPSuccess= 'Y',

k.IsActive='Y' ,

k.RegisteredOn=pRegisteredOn

Where k.KioskID= pKioskID;

elseif pTypeFlg= 6 Then

select \* from kiosk where OrgID=pOrgID and OTPSuccess='Y' order by KioskID desc;

elseif pTypeFlg= 7 Then

Update kiosk k Set

k.IsActive='N'

Where k.KioskID= pKioskID;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_KioskAuth` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_KioskAuth`(

in pKioskAuthID int(11),

in pOrgID int(11),

in pByLocation char(1),

in pByTime char(1),

in pTypeFlg int(11)

)

BEGIN

if pTypeFlg = 1 Then

SELECT ByLocation, ByTime, OrgID FROM kioskauth WHERE OrgID=pOrgID;

elseif pTypeFlg = 2 Then

INSERT INTO kioskauth

(

OrgID,

ByLocation,

ByTime

)

VALUES

(

pOrgID,

pByLocation,

pByTime

);

elseif pTypeFlg = 3 Then

UPDATE kioskauth SET ByLocation=pByLocation, ByTime=pByTime WHERE OrgID=pOrgID;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_kioskImage` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_kioskImage`(

in pKioskLogoID int(11),

in pOrgID int(11),

in pDeviceImage varchar(255),

in pDeviceEncImage longblob,

in pVersionID int(11),

in pCreated datetime,

in pModified datetime,

in pIsActive char(1),

in pTypeFlg int(11)

)

BEGIN

if pTypeFlg= 1 Then

select \* from kiosklogo where OrgID=pOrgID;

elseif pTypeFlg= 2 Then

INSERT INTO kiosklogo

(

OrgID,

DeviceImage,

DeviceEncImage,

Created,

Modified,

IsActive

)

VALUES

(

pOrgID,

pDeviceImage,

pDeviceEncImage,

pCreated,

pModified,

pIsActive

);

elseif pTypeFlg= 3 Then

Update kiosklogo Set

DeviceImage= pDeviceImage,

DeviceEncImage= pDeviceEncImage,

VersionID = pVersionID,

Modified= pModified,

IsActive = pIsActive

Where KioskLogoID = pKioskLogoID;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_leaveapplied` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_leaveapplied`(

In pLeavAppID text,

in pUserID int(11),

IN pLeaveTypeID int(11),

in pLeaveStatusID int(11),

in pLeaveStatusNote text,

in pAppliedDate datetime(3),

in pNoOfDays float(3,1),

in pPolicyType varchar(20),

in pStartDate date,

in pEndDate date,

in pIsHalfDay varchar(1),

in pWhichHalf varchar(1),

in pLeaveReason text,

in pStatusUpdateDate datetime(3),

in pStatusUpdteBy int(11),

in pTypeFlg int(11)

)

BEGIN

DECLARE LID int;

DECLARE i,NOOFLEV int;

start transaction;

if pTypeFlg= 1 Then

INSERT INTO leaveapplied

(

UserID,

LeaveTypeID,

LeaveStatusID,

AppliedDate,

NoOfDays,

PolicyType,

StartDate,

EndDate,

IsHalfDay,

WhichHalf,

LeaveReason,

StatusUpdateDate,

StatusUpdteBy

)

VALUES

(

pUserID,

pLeaveTypeID,

pLeaveStatusID,

pAppliedDate,

pNoOfDays,

pPolicyType,

pStartDate,

pEndDate,

pIsHalfDay,

pWhichHalf,

pLeaveReason,

pStatusUpdateDate,

pStatusUpdteBy

);

SET LID = LAST\_INSERT\_ID();

INSERT INTO leaveappliedstatuslog

(

LeavAppID,

LeaveStatusID,

StatusNote,

StatusAppliedDate

)

VALUES

(

LID,

pLeaveStatusID,

pLeaveStatusNote,

pAppliedDate

);

elseif pTypeFlg= 2 Then

call splitter(pLeavAppID,',');

select count(SplitValue) INTO NOOFLEV from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFLEV DO

Update leaveapplied Set

LeaveStatusID= pLeaveStatusID,

StatusUpdateDate= pStatusUpdateDate,

StatusUpdteBy= pStatusUpdteBy

Where LeavAppID = (select SplitValue from splitResultsTmpTable where TmpID = i);

INSERT INTO leaveappliedstatuslog

(

LeavAppID,

LeaveStatusID,

StatusNote,

StatusAppliedDate

)

VALUES

(

(select SplitValue from splitResultsTmpTable where TmpID = i),

pLeaveStatusID,

pLeaveStatusNote,

pStatusUpdateDate

);

SET i = i + 1;

END WHILE;

elseif pTypeFlg= 3 Then

if pLeaveStatusID='' Then

Select LA.LeavAppID,LA.UserID,U.DeptID,LA.LeaveTypeID,LA.LeaveStatusID,LA.AppliedDate,

LA.StartDate,LA.EndDate,LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,

LA.StatusUpdateDate,LA.StatusUpdteBy,

TRIM(TRAILING '.' FROM TRIM(TRAILING '0' from LA.NoOfDays)) AS NoOfDays,

LT.LeaveTypeText,LS.LeaveStatusText,LAL.StatusNote,LMS.leaveMasterStatus

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavemasterstatus LMS ON LMS.leaveMasterID = LS.LeaveMasterStatusID

LEFT JOIN leaveappliedstatuslog LAL ON LAL.LeavAppID = LA.LeavAppID

where LA.UserID = pUserID AND

(LA.LeaveStatusID = LAL.LeaveStatusID)

order by LA.StartDate desc;

Else

Select LA.LeavAppID,LA.UserID,U.DeptID,LA.LeaveTypeID,LA.LeaveStatusID,LA.AppliedDate,

LA.StartDate,LA.EndDate,LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,

LA.StatusUpdateDate,LA.StatusUpdteBy,

TRIM(TRAILING '.' FROM TRIM(TRAILING '0' from LA.NoOfDays)) AS NoOfDays,

LT.LeaveTypeText,LS.LeaveStatusText,LAL.StatusNote,LMS.leaveMasterStatus

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavemasterstatus LMS ON LMS.leaveMasterID = LS.LeaveMasterStatusID

LEFT JOIN leaveappliedstatuslog LAL ON LAL.LeavAppID = LA.LeavAppID

where LA.UserID = pUserID AND LA.LeaveStatusID = pLeaveStatusID

AND (LA.LeaveStatusID = LAL.LeaveStatusID)

order by LA.StartDate desc;

End If;

elseif pTypeFlg= 5 Then /\* added by shams \*/

Select U.OrgID,LA.LeavAppID,LA.LeaveTypeID,LA.LeaveStatusID,LA.AppliedDate,

LA.StartDate,LA.EndDate,LS.LeaveMasterStatusID

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.LeavAppID = pLeavAppID;

Else

Select U.OrgID,LA.LeavAppID,LA.UserID,U.DeptID,LA.LeaveTypeID,LA.LeaveStatusID,LA.AppliedDate,

LA.StartDate,LA.EndDate,LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,

LA.StatusUpdateDate,LA.StatusUpdteBy,

TRIM(TRAILING '.' FROM TRIM(TRAILING '0' from LA.NoOfDays)) AS NoOfDays,

LT.LeaveTypeText,LS.LeaveStatusText,LAL.StatusNote,LS.LeaveMasterStatusID

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leaveappliedstatuslog LAL ON LAL.LeavAppID = LA.LeavAppID

where LA.LeavAppID IN (pLeavAppID) AND

(LA.LeaveStatusID = LAL.LeaveStatusID);

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_leaveapplied\_BKP` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_leaveapplied\_BKP`(

In pLeavAppID text,

in pUserID int(11),

IN pLeaveTypeID int(11),

in pLeaveStatusID int(11),

in pLeaveStatusNote text,

in pAppliedDate datetime(3),

in pStartDate date,

in pEndDate date,

in pIsHalfDay varchar(1),

in pWhichHalf varchar(1),

in pLeaveReason text,

in pStatusUpdateDate datetime(3),

in pStatusUpdteBy int(11),

in pTypeFlg int(11)

)

BEGIN

DECLARE LID int;

DECLARE i,NOOFLEV int;

start transaction;

if pTypeFlg= 1 Then

INSERT INTO leaveapplied

(

UserID,

LeaveTypeID,

LeaveStatusID,

AppliedDate,

StartDate,

EndDate,

IsHalfDay,

WhichHalf,

LeaveReason,

StatusUpdateDate,

StatusUpdteBy

)

VALUES

(

pUserID,

pLeaveTypeID,

pLeaveStatusID,

pAppliedDate,

pStartDate,

pEndDate,

pIsHalfDay,

pWhichHalf,

pLeaveReason,

pStatusUpdateDate,

pStatusUpdteBy

);

SET LID = LAST\_INSERT\_ID();

INSERT INTO leaveappliedstatuslog

(

LeavAppID,

LeaveStatusID,

StatusNote,

StatusAppliedDate

)

VALUES

(

LID,

pLeaveStatusID,

pLeaveStatusNote,

pAppliedDate

);

elseif pTypeFlg= 2 Then

call splitter(pLeavAppID,',');

select count(SplitValue) INTO NOOFLEV from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFLEV DO

Update leaveapplied Set

LeaveStatusID= pLeaveStatusID,

StatusUpdateDate= pStatusUpdateDate,

StatusUpdteBy= pStatusUpdteBy

Where LeavAppID = (select SplitValue from splitResultsTmpTable where TmpID = i);

INSERT INTO leaveappliedstatuslog

(

LeavAppID,

LeaveStatusID,

StatusNote,

StatusAppliedDate

)

VALUES

(

(select SplitValue from splitResultsTmpTable where TmpID = i),

pLeaveStatusID,

pLeaveStatusNote,

pStatusUpdateDate

);

SET i = i + 1;

END WHILE;

elseif pTypeFlg= 3 Then

if pLeaveStatusID='' Then

Select LA.LeavAppID,LA.UserID,U.DeptID,LA.LeaveTypeID,LA.LeaveStatusID,LA.AppliedDate,

LA.StartDate,LA.EndDate,LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,

LA.StatusUpdateDate,LA.StatusUpdteBy,

IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,

LT.LeaveTypeText,LS.LeaveStatusText,LAL.StatusNote,LMS.leaveMasterStatus

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavemasterstatus LMS ON LMS.leaveMasterID = LS.LeaveMasterStatusID

LEFT JOIN leaveappliedstatuslog LAL ON LAL.LeavAppID = LA.LeavAppID

where LA.UserID = pUserID AND

(LA.LeaveStatusID = LAL.LeaveStatusID)

order by LA.StartDate desc;

Else

Select LA.LeavAppID,LA.UserID,U.DeptID,LA.LeaveTypeID,LA.LeaveStatusID,LA.AppliedDate,

LA.StartDate,LA.EndDate,LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,

LA.StatusUpdateDate,LA.StatusUpdteBy,

IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,

LT.LeaveTypeText,LS.LeaveStatusText,LAL.StatusNote,LMS.leaveMasterStatus

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavemasterstatus LMS ON LMS.leaveMasterID = LS.LeaveMasterStatusID

LEFT JOIN leaveappliedstatuslog LAL ON LAL.LeavAppID = LA.LeavAppID

where LA.UserID = pUserID AND LA.LeaveStatusID = pLeaveStatusID

AND (LA.LeaveStatusID = LAL.LeaveStatusID)

order by LA.StartDate desc;

End If;

elseif pTypeFlg= 5 Then /\* added by shams \*/

Select U.OrgID,LA.LeavAppID,LA.LeaveTypeID,LA.LeaveStatusID,LA.AppliedDate,

LA.StartDate,LA.EndDate,LS.LeaveMasterStatusID

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.LeavAppID = pLeavAppID;

Else

Select U.OrgID,LA.LeavAppID,LA.UserID,U.DeptID,LA.LeaveTypeID,LA.LeaveStatusID,LA.AppliedDate,

LA.StartDate,LA.EndDate,LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,

LA.StatusUpdateDate,LA.StatusUpdteBy,

IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,

LT.LeaveTypeText,LS.LeaveStatusText,LAL.StatusNote,LS.LeaveMasterStatusID

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leaveappliedstatuslog LAL ON LAL.LeavAppID = LA.LeavAppID

where LA.LeavAppID IN (pLeavAppID) AND

(LA.LeaveStatusID = LAL.LeaveStatusID);

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_leavecredit` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_leavecredit`(

IN `pUserLeaveCrdID` INT(11),

IN `pUserID` TEXT,

IN `pLeaveTypeID` TEXT,

IN `pCreditLeave` FLOAT(5,1),

IN `pCreditDate` DATE,

IN `pDeptID` INT(11),

IN `pOrgID` INT(11),

IN `pTypeFlg` INT(11))

BEGIN

DECLARE i,NOOFUSER,LEAVEDAYCOUNT,LID int;

DECLARE NO\_OF\_DAYS\_VAL,EXACT\_HLDWKEND\_CNT float(3,1);

DECLARE POLICYTYPE\_VAL,POLICYTYPE\_VAL\_TMP varchar(20);

DECLARE STARTDATE\_VAL,ENDDATE\_VAL date;

start transaction;

if pTypeFlg= 1 Then

call splitter(pUserID,',');

select count(SplitValue) INTO NOOFUSER from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFUSER DO

INSERT INTO userleavecredit

(

UserID,

LeaveTypeID,

CreditLeave,

CreditDate

)

VALUES

(

(select SplitValue from splitResultsTmpTable where TmpID = i),

pLeaveTypeID,

pCreditLeave,

pCreditDate

);

SET i = i + 1;

END WHILE;

elseif pTypeFlg=2 Then

Update userleavecredit set

CreditLeave=pCreditLeave

where UserLeaveCrdID=pUserLeaveCrdID;

elseif pTypeFlg=3 Then

select U.UserID,U.OrgID,U.EmployeeID,U.DeptID,U.FName,U.LName,U.Email

from userpersonalinfo U

where U.DeptID=pDeptID;

elseif pTypeFlg=4 Then

select U.UserID,U.FName,U.LName,U.EmployeeID,D.DepartmentName,LT.LeaveTypeText,ULC.CreditLeave,ULC.CreditDate,ULC.CreditType

from userleavecredit ULC

LEFT JOIN userpersonalinfo U ON ULC.UserID = U.UseriD

LEFT JOIN department D ON D.DeptID = U.DeptID

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

where ULC.UserID = pUserID order by ULC.CreditDate desc;

elseif pTypeFlg=5 Then

select U.UserID,U.FName,U.LName,U.EmployeeID,D.DepartmentName,LT.LeaveTypeText,ULC.CreditLeave,ULC.CreditDate,ULC.CreditType

from userleavecredit ULC

LEFT JOIN userpersonalinfo U ON ULC.UserID = U.UseriD

LEFT JOIN department D ON D.DeptID = U.DeptID

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

where U.OrgID = pOrgID order by ULC.CreditDate desc;

elseif pTypeFlg=6 Then

UPDATE userleavecredit set IsCountable='Y' WHERE UserLeaveCrdID IN(

select UserLeaveCrdID from (select ULC.UserLeaveCrdID

from userleavecredit ULC

LEFT JOIN userpersonalinfo U ON ULC.UserID = U.UserID

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

where U.OrgID = pOrgID AND ULC.UserID=pUserID AND ULC.CreditType='C' ) ulcid);

UPDATE userleavecredit set IsCountable='N' WHERE UserLeaveCrdID IN(

select UserLeaveCrdID from (select ULC.UserLeaveCrdID

from userleavecredit ULC

LEFT JOIN userpersonalinfo U ON ULC.UserID = U.UserID

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

where U.OrgID = pOrgID AND date(ULC.CreditDate) <=pCreditDate AND ULC.UserID=pUserID

AND ULC.LeaveTypeID=pLeaveTypeID AND ULC.IsCountable='Y') ulcid);

UPDATE userleavecredit set IsCountable='N' WHERE UserLeaveCrdID IN(

select UserLeaveCrdID from (select ULC.UserLeaveCrdID

from userleavecredit ULC

LEFT JOIN userpersonalinfo U ON ULC.UserID = U.UserID

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

where U.OrgID = pOrgID AND date(ULC.CreditDate)>pCreditDate AND ULC.UserID=pUserID

AND ULC.LeaveTypeID=pLeaveTypeID AND ULC.IsCountable='Y' AND ULC.CreditType='U' ) ulcid);

INSERT INTO userleavecredit

(

UserID,

LeaveTypeID,

CreditLeave,

CreditDate,

CreditType

)

VALUES

(

pUserID,

pLeaveTypeID,

pCreditLeave,

pCreditDate,

'U'

);

/\* Get the count of leave taken \*/

SET LEAVEDAYCOUNT= (SELECT DATEDIFF(LA.EndDate, LA.StartDate) from leaveapplied

LA LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

WHERE

U.OrgID = pOrgID

AND LA.LeaveTypeID=pLeaveTypeID

AND LA.UserID=pUserID

AND LA.StartDate <=pCreditDate

AND pCreditDate<LA.EndDate

AND LS.LeaveMasterStatusID IN('1','2')

);

IF(LEAVEDAYCOUNT IS NULL) THEN

set LEAVEDAYCOUNT=0;

END IF;

/\* Split the leave into two entries if LEAVEDAYCOUNT>0 i.e; Taken leave greater than or equal to two \*/

IF LEAVEDAYCOUNT>0 Then

/\* Taking backup of current record starts \*/

drop table if exists temp\_LeaveApplied;

drop table if exists temp\_LeaveApplied2;

drop table if exists temp\_LeaveApplied3;

create TEMPORARY TABLE temp\_LeaveApplied like leaveapplied;

INSERT INTO temp\_LeaveApplied SELECT \* from leaveapplied where LeavAppID IN

(SELECT LeavAppID from((SELECT LA.LeavAppID from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

WHERE

U.OrgID = pOrgID

AND LA.LeaveTypeID=pLeaveTypeID

AND LA.UserID=pUserID

AND LA.StartDate <=pCreditDate

AND pCreditDate<LA.EndDate

AND LS.LeaveMasterStatusID IN('1','2')

) LAPID2));

/\* Taking backup of current record ends \*/

/\*

UPDATE leaveapplied SET EndDate=DATE\_SUB(pCreditDate, INTERVAL 1 DAY),CreateType='S' WHERE LeavAppID IN

(

SELECT LeavAppID from((SELECT LA.LeavAppID from leaveapplied

LA LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where U.OrgID = pOrgID

AND LA.LeaveTypeID=pLeaveTypeID

AND LA.UserID=pUserID

AND LA.StartDate <=pCreditDate

AND pCreditDate<LA.EndDate

AND LS.LeaveMasterStatusID IN('1','2')

) LAPID1));

INSERT INTO leaveapplied (UserID, LeaveTypeID,LeaveStatusID,AppliedDate,StartDate,EndDate,

IsHalfDay,WhichHalf,LeaveReason,StatusUpdateDate,StatusUpdteBy,CreateType)

SELECT UserID,LeaveTypeID,LeaveStatusID,AppliedDate,

pCreditDate,EndDate,IsHalfDay,WhichHalf,

LeaveReason,StatusUpdateDate,StatusUpdteBy,'S' from temp\_LeaveApplied;

\*/

SELECT PolicyType,StartDate,EndDate INTO POLICYTYPE\_VAL\_TMP,STARTDATE\_VAL,ENDDATE\_VAL from temp\_LeaveApplied;

CASE POLICYTYPE\_VAL\_TMP

WHEN 'S-PH,WO' THEN

SET POLICYTYPE\_VAL = 'PH,WO';

WHEN 'S-WO,PH' THEN

SET POLICYTYPE\_VAL = 'PH,WO';

WHEN 'S-PH' THEN

SET POLICYTYPE\_VAL = 'PH';

WHEN 'S-WO' THEN

SET POLICYTYPE\_VAL = 'WO';

ELSE SET POLICYTYPE\_VAL = '';

END CASE;

CALL nex\_GetLeaveInHolidayOrWeekend(pUserID,STARTDATE\_VAL,pCreditDate,POLICYTYPE\_VAL,@DaysInHolidayOrWeekend);

SELECT @DaysInHolidayOrWeekend INTO EXACT\_HLDWKEND\_CNT;

SET NO\_OF\_DAYS\_VAL = (DATEDIFF(pCreditDate,STARTDATE\_VAL)+1) - EXACT\_HLDWKEND\_CNT;

UPDATE leaveapplied SET NoOfDays=NO\_OF\_DAYS\_VAL,EndDate=pCreditDate,CreateType='S' WHERE LeavAppID IN

(

SELECT LeavAppID from((SELECT LA.LeavAppID from leaveapplied

LA LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where U.OrgID = pOrgID

AND LA.LeaveTypeID=pLeaveTypeID

AND LA.UserID=pUserID

AND LA.StartDate <=pCreditDate

AND pCreditDate<LA.EndDate

AND LS.LeaveMasterStatusID IN('1','2')

) LAPID1));

SET EXACT\_HLDWKEND\_CNT = 0.0;

CALL nex\_GetLeaveInHolidayOrWeekend(pUserID,DATE\_ADD(pCreditDate, INTERVAL 1 DAY),ENDDATE\_VAL,POLICYTYPE\_VAL,@DaysInHolidayOrWeekend);

SELECT @DaysInHolidayOrWeekend INTO EXACT\_HLDWKEND\_CNT;

SET NO\_OF\_DAYS\_VAL = 0.0;

SET NO\_OF\_DAYS\_VAL = (DATEDIFF(ENDDATE\_VAL,DATE\_ADD(pCreditDate, INTERVAL 1 DAY))+1) - EXACT\_HLDWKEND\_CNT;

INSERT INTO leaveapplied (UserID, LeaveTypeID,LeaveStatusID,AppliedDate,NoOfDays,PolicyType,StartDate,EndDate,

IsHalfDay,WhichHalf,LeaveReason,StatusUpdateDate,StatusUpdteBy,CreateType)

SELECT UserID,LeaveTypeID,LeaveStatusID,AppliedDate,NO\_OF\_DAYS\_VAL,POLICYTYPE\_VAL\_TMP,

DATE\_ADD(pCreditDate, INTERVAL 1 DAY),EndDate,IsHalfDay,WhichHalf,

LeaveReason,StatusUpdateDate,StatusUpdteBy,'S' from temp\_LeaveApplied;

SET LID = LAST\_INSERT\_ID();

create TEMPORARY TABLE temp\_LeaveApplied2 like temp\_LeaveApplied;

insert temp\_LeaveApplied2 select \* from temp\_LeaveApplied;

create TEMPORARY TABLE temp\_LeaveApplied3 like temp\_LeaveApplied;

insert temp\_LeaveApplied3 select \* from temp\_LeaveApplied;

INSERT INTO leaveappliedstatuslog

(

LeavAppID,

LeaveStatusID,

StatusNote,

StatusAppliedDate

)

VALUES

(

LID,

(SELECT LeaveStatusID from temp\_LeaveApplied2),

'',

(SELECT StatusUpdateDate from temp\_LeaveApplied3)

);

drop table if exists temp\_LeaveApplied;

drop table if exists temp\_LeaveApplied2;

drop table if exists temp\_LeaveApplied3;

End if;

/\* RESET ALL LEAVE TO IsCountable='N' \*/

UPDATE leaveapplied SET IsCountable='N' WHERE LeavAppID IN

(

SELECT LeavAppID from ((SELECT LA.LeavAppID from leaveapplied

LA LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

where U.OrgID = pOrgID

AND LA.LeaveTypeID=pLeaveTypeID

AND LA.UserID=pUserID

AND LA.EndDate<=pCreditDate

AND LA.IsCountable='Y'

) LAPID1));

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_leavecredit\_BKP` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_leavecredit\_BKP`(

IN `pUserLeaveCrdID` INT(11),

IN `pUserID` TEXT,

IN `pLeaveTypeID` TEXT,

IN `pCreditLeave` FLOAT(5,1),

IN `pCreditDate` DATE,

IN `pDeptID` INT(11),

IN `pOrgID` INT(11),

IN `pTypeFlg` INT(11))

BEGIN

DECLARE i,NOOFUSER,LEAVEDAYCOUNT,LID int;

start transaction;

if pTypeFlg= 1 Then

call splitter(pUserID,',');

select count(SplitValue) INTO NOOFUSER from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFUSER DO

INSERT INTO userleavecredit

(

UserID,

LeaveTypeID,

CreditLeave,

CreditDate

)

VALUES

(

(select SplitValue from splitResultsTmpTable where TmpID = i),

pLeaveTypeID,

pCreditLeave,

pCreditDate

);

SET i = i + 1;

END WHILE;

elseif pTypeFlg=2 Then

Update userleavecredit set

CreditLeave=pCreditLeave

where UserLeaveCrdID=pUserLeaveCrdID;

elseif pTypeFlg=3 Then

select U.UserID,U.OrgID,U.EmployeeID,U.DeptID,U.FName,U.LName,U.Email

from userpersonalinfo U

where U.DeptID=pDeptID;

elseif pTypeFlg=4 Then

select U.UserID,U.FName,U.LName,U.EmployeeID,D.DepartmentName,LT.LeaveTypeText,ULC.CreditLeave,ULC.CreditDate,ULC.CreditType

from userleavecredit ULC

LEFT JOIN userpersonalinfo U ON ULC.UserID = U.UseriD

LEFT JOIN department D ON D.DeptID = U.DeptID

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

where ULC.UserID = pUserID order by ULC.CreditDate desc;

elseif pTypeFlg=5 Then

select U.UserID,U.FName,U.LName,U.EmployeeID,D.DepartmentName,LT.LeaveTypeText,ULC.CreditLeave,ULC.CreditDate,ULC.CreditType

from userleavecredit ULC

LEFT JOIN userpersonalinfo U ON ULC.UserID = U.UseriD

LEFT JOIN department D ON D.DeptID = U.DeptID

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

where U.OrgID = pOrgID order by ULC.CreditDate desc;

elseif pTypeFlg=6 Then

UPDATE userleavecredit set IsCountable='Y' WHERE UserLeaveCrdID IN(

select UserLeaveCrdID from (select ULC.UserLeaveCrdID

from userleavecredit ULC

LEFT JOIN userpersonalinfo U ON ULC.UserID = U.UserID

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

where U.OrgID = pOrgID AND ULC.UserID=pUserID AND ULC.CreditType='C' ) ulcid);

UPDATE userleavecredit set IsCountable='N' WHERE UserLeaveCrdID IN(

select UserLeaveCrdID from (select ULC.UserLeaveCrdID

from userleavecredit ULC

LEFT JOIN userpersonalinfo U ON ULC.UserID = U.UserID

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

where U.OrgID = pOrgID AND date(ULC.CreditDate) <=pCreditDate AND ULC.UserID=pUserID

AND ULC.LeaveTypeID=pLeaveTypeID AND ULC.IsCountable='Y') ulcid);

UPDATE userleavecredit set IsCountable='N' WHERE UserLeaveCrdID IN(

select UserLeaveCrdID from (select ULC.UserLeaveCrdID

from userleavecredit ULC

LEFT JOIN userpersonalinfo U ON ULC.UserID = U.UserID

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

where U.OrgID = pOrgID AND date(ULC.CreditDate)>pCreditDate AND ULC.UserID=pUserID

AND ULC.LeaveTypeID=pLeaveTypeID AND ULC.IsCountable='Y' AND ULC.CreditType='U' ) ulcid);

INSERT INTO userleavecredit

(

UserID,

LeaveTypeID,

CreditLeave,

CreditDate,

CreditType

)

VALUES

(

pUserID,

pLeaveTypeID,

pCreditLeave,

pCreditDate,

'U'

);

SET LEAVEDAYCOUNT= (SELECT DATEDIFF(LA.EndDate, LA.StartDate) from leaveapplied

LA LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

WHERE

U.OrgID = pOrgID

AND LA.LeaveTypeID=pLeaveTypeID

AND LA.UserID=pUserID

AND LA.StartDate <=pCreditDate

AND pCreditDate<LA.EndDate

AND LS.LeaveMasterStatusID IN('1','2')

);

IF(LEAVEDAYCOUNT IS NULL) THEN

set LEAVEDAYCOUNT=0;

END IF;

IF LEAVEDAYCOUNT>0 Then

drop table if exists temp\_LeaveApplied;

drop table if exists temp\_LeaveApplied2;

drop table if exists temp\_LeaveApplied3;

create TEMPORARY TABLE temp\_LeaveApplied like leaveapplied;

INSERT INTO temp\_LeaveApplied SELECT \* from leaveapplied where LeavAppID IN

(SELECT LeavAppID from((SELECT LA.LeavAppID from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

WHERE

U.OrgID = pOrgID

AND LA.LeaveTypeID=pLeaveTypeID

AND LA.UserID=pUserID

AND LA.StartDate <=pCreditDate

AND pCreditDate<LA.EndDate

AND LS.LeaveMasterStatusID IN('1','2')

) LAPID2));

/\*

UPDATE leaveapplied SET EndDate=DATE\_SUB(pCreditDate, INTERVAL 1 DAY),CreateType='S' WHERE LeavAppID IN

(

SELECT LeavAppID from((SELECT LA.LeavAppID from leaveapplied

LA LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where U.OrgID = pOrgID

AND LA.LeaveTypeID=pLeaveTypeID

AND LA.UserID=pUserID

AND LA.StartDate <=pCreditDate

AND pCreditDate<LA.EndDate

AND LS.LeaveMasterStatusID IN('1','2')

) LAPID1));

INSERT INTO leaveapplied (UserID, LeaveTypeID,LeaveStatusID,AppliedDate,StartDate,EndDate,

IsHalfDay,WhichHalf,LeaveReason,StatusUpdateDate,StatusUpdteBy,CreateType)

SELECT UserID,LeaveTypeID,LeaveStatusID,AppliedDate,

pCreditDate,EndDate,IsHalfDay,WhichHalf,

LeaveReason,StatusUpdateDate,StatusUpdteBy,'S' from temp\_LeaveApplied;

\*/

UPDATE leaveapplied SET EndDate=pCreditDate,CreateType='S' WHERE LeavAppID IN

(

SELECT LeavAppID from((SELECT LA.LeavAppID from leaveapplied

LA LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where U.OrgID = pOrgID

AND LA.LeaveTypeID=pLeaveTypeID

AND LA.UserID=pUserID

AND LA.StartDate <=pCreditDate

AND pCreditDate<LA.EndDate

AND LS.LeaveMasterStatusID IN('1','2')

) LAPID1));

INSERT INTO leaveapplied (UserID, LeaveTypeID,LeaveStatusID,AppliedDate,StartDate,EndDate,

IsHalfDay,WhichHalf,LeaveReason,StatusUpdateDate,StatusUpdteBy,CreateType)

SELECT UserID,LeaveTypeID,LeaveStatusID,AppliedDate,

DATE\_ADD(pCreditDate, INTERVAL 1 DAY),EndDate,IsHalfDay,WhichHalf,

LeaveReason,StatusUpdateDate,StatusUpdteBy,'S' from temp\_LeaveApplied;

SET LID = LAST\_INSERT\_ID();

create TEMPORARY TABLE temp\_LeaveApplied2 like temp\_LeaveApplied;

insert temp\_LeaveApplied2 select \* from temp\_LeaveApplied;

create TEMPORARY TABLE temp\_LeaveApplied3 like temp\_LeaveApplied;

insert temp\_LeaveApplied3 select \* from temp\_LeaveApplied;

INSERT INTO leaveappliedstatuslog

(

LeavAppID,

LeaveStatusID,

StatusNote,

StatusAppliedDate

)

VALUES

(

LID,

(SELECT LeaveStatusID from temp\_LeaveApplied2),

'',

(SELECT StatusUpdateDate from temp\_LeaveApplied3)

);

drop table if exists temp\_LeaveApplied;

drop table if exists temp\_LeaveApplied2;

drop table if exists temp\_LeaveApplied3;

End if;

/\* RESET ALL LEAVE TO IsCountable='N' \*/

UPDATE leaveapplied SET IsCountable='N' WHERE LeavAppID IN

(

SELECT LeavAppID from ((SELECT LA.LeavAppID from leaveapplied

LA LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

where U.OrgID = pOrgID

AND LA.LeaveTypeID=pLeaveTypeID

AND LA.UserID=pUserID

AND LA.EndDate<=pCreditDate

AND LA.IsCountable='Y'

) LAPID1));

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_leavestatus` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_leavestatus`(

IN pLeaveStatusID int(11),

IN pLeaveMasterStatusID int(11),

IN pLeaveStatusText varchar(50),

in pOrgID int(11),

in pTypeFlg int(11)

)

BEGIN

DECLARE EXIT HANDLER FOR 1451

SELECT 'FKC-ERROR'; /\*Foreign Key Constraint Error\*/

start transaction;

if pTypeFlg= 1 Then

INSERT INTO leavestatus

(

LeaveStatusText,

LeaveMasterStatusID,

OrgID

)

VALUES

(

pLeaveStatusText,

pLeaveMasterStatusID,

pOrgID

);

elseif pTypeFlg= 2 Then

Update leavestatus Set

LeaveStatusText= pLeaveStatusText,

LeaveMasterStatusID= pLeaveMasterStatusID,

OrgID= pOrgID

Where LeaveStatusID= pLeaveStatusID;

elseif pTypeFlg= 3 Then

Select LS.\*,LMS.leaveMasterStatus from leavestatus LS LEFT JOIN leavemasterstatus LMS

ON LS.LeaveMasterStatusID = LMS.leaveMasterID where LeaveStatusID= pLeaveStatusID;

elseif pTypeFlg= 4 Then

delete from leavestatus where LeaveStatusID= pLeaveStatusID and OrgID = pOrgID;

select 'SUCCESS';

elseif pTypeFlg= 5 Then

select \* from leavestatus where LeaveMasterStatusID = pLeaveMasterStatusID and OrgID = pOrgID;

elseif pTypeFlg= 6 Then

select \* from leavestatus where LeaveStatusText = pLeaveStatusText and OrgID = pOrgID;

else

select LS.\*,LMS.leaveMasterStatus from leavestatus LS LEFT JOIN leavemasterstatus LMS

ON LS.LeaveMasterStatusID = LMS.leaveMasterID where OrgID= pOrgID;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_leavetype` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_leavetype`(

IN pLeaveTypeID int(11),

IN pLeaveMasterTypeID int(11),

IN pLeaveTypeText varchar(50),

IN pLeaveDescription text,

IN pOrgID int(11),

IN pTypeFlg int(11)

)

BEGIN

DECLARE EXIT HANDLER FOR 1451

SELECT 'FKC-ERROR'; /\*Foreign Key Constraint Error\*/

start transaction;

if pTypeFlg= 1 Then

INSERT INTO leavetype

(

LeaveMasterTypeID,

LeaveTypeText,

LeaveTypeDescription,

OrgID

)

VALUES

(

pLeaveMasterTypeID,

pLeaveTypeText,

pLeaveDescription,

pOrgID

);

elseif pTypeFlg= 2 Then

Update leavetype Set

LeaveMasterTypeID= pLeaveMasterTypeID,

LeaveTypeText= pLeaveTypeText,

OrgID= pOrgID,

LeaveTypeDescription= pLeaveDescription

Where LeaveTypeID= pLeaveTypeID;

elseif pTypeFlg= 3 Then

Select LT.\*,LMT.LeaveMasterType

from leavetype LT LEFT JOIN leavemastertype LMT

ON LT.LeaveMasterTypeID = LMT.LeaveMasterTypeID

where LT.OrgID = pOrgID AND LT.LeaveTypeID = pLeaveTypeID;

elseif pTypeFlg= 4 Then

delete from leavetype where LeaveTypeID = pLeaveTypeID and OrgID = pOrgID;

select 'SUCCESS';

elseif pTypeFlg= 5 Then

select \* from leavetype where LeaveMasterTypeID = pLeaveMasterTypeID and OrgID = pOrgID;

elseif pTypeFlg=6 Then

select \* from leavetype where LeaveTypeText= pLeaveTypeText and OrgID= pOrgID;

else

Select LT.\*,LMT.LeaveMasterType

from leavetype LT LEFT JOIN leavemastertype LMT

ON LT.LeaveMasterTypeID = LMT.LeaveMasterTypeID

where LT.OrgID = pOrgID;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_nexapiexceptions` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_nexapiexceptions`(

In pExcptionID int(11),

IN pErrorCode varchar(15),

in pErrMessage varchar(245),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO nexapiexceptions

(

ErrorCode,

ErrMessage

)

VALUES

(

pErrorCode,

pErrMessage

);

elseif pTypeFlg= 2 Then

Update nexapiexceptions Set

ErrorCode= pErrorCode,

ErrMessage= pErrMessage

Where ExcptionID= pExcptionID

;

Else

Select \* from nexapiexceptions;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_nexmodule` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_nexmodule`(

In pModuleID int(11),

IN pModuleName varchar(50),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO nexmodule

(

ModuleName

)

VALUES

(

pModuleName

);

elseif pTypeFlg= 2 Then

Update nexmodule Set

ModuleName= pModuleName

Where ModuleID= pModuleID

;

Else

Select \* from nexmodule;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_nexoperator` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_nexoperator`(

In pOperatorID int(11),

IN pLoginName varchar(150),

in pPassword varchar(50),

in pName varchar(50),

in pEmail varchar(150),

in pEmployeeID varchar(50),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO nexoperator

(

LoginName,

Password,

Name,

Email,

EmployeeID

)

VALUES

(

pLoginName,

pPassword,

pName,

pEmail,

pEmployeeID

);

elseif pTypeFlg= 2 Then

Update nexoperator Set

LoginName= pLoginName,

Password= pPassword,

Name= pName,

Email= pEmail,

EmployeeID= pEmployeeID

Where OperatorID= pOperatorID

;

Else

Select \* from nexoperator;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_nexorganizationpackage` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_nexorganizationpackage`(

In pOrgPkgID int(11),

IN pPackageID Int(11),

in pOrgID int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO nexorganizationpackage

(

PackageID,

OrgID

)

VALUES

(

pPackageID,

pOrgID

);

elseif pTypeFlg= 2 Then

Update nexorganizationpackage Set

PackageID= pPackageID,

OrgID= pOrgID

Where OrgPkgID= pOrgPkgID

;

Else

Select \* from nexorganizationpackage;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_nexpackage` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_nexpackage`(

In pPackageID int(11),

IN pPackageName varchar(50),

in pPackageTypeID int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO nexpackage

(

PackageName,

PackageTypeID

)

VALUES

(

pPackageName,

pPackageTypeID

);

elseif pTypeFlg= 2 Then

Update nexpackage Set

PackageName= pPackageName,

PackageTypeID= pPackageTypeID

Where PackageID= pPackageID

;

Else

Select \* from nexpackage;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_nexpackagemodule` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_nexpackagemodule`(

In pPackageModuleID int(11),

IN pPackageTypeID Int(11),

in pModuleID int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO nexpackagemodule

(

PackageTypeID,

ModuleID

)

VALUES

(

pPackageTypeID,

pModuleID

);

elseif pTypeFlg= 2 Then

Update nexpackagemodule Set

PackageTypeID= pPackageTypeID,

ModuleID= pModuleID

Where PackageModuleID= pPackageModuleID

;

Else

Select \* from nexpackagemodule;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_nexpackagetype` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_nexpackagetype`(

In pPackageTypeID int(11),

IN pTypeName varchar(50),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO nexpackagetype

(

TypeName

)

VALUES

(

pTypeName

);

elseif pTypeFlg= 2 Then

Update nexpackagetype Set

TypeName= pTypeName

Where PackageTypeID= pPackageTypeID

;

Else

Select \* from nexpackagetype;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_notificationtype` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_notificationtype`(

In pNotificationTypeID int(11),

IN pNotificationTypeName varchar(50),

in pOrgID int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO notificationtype

(

NotificationTypeName,

OrgID

)

VALUES

(

pNotificationTypeName,

pOrgID

);

elseif pTypeFlg= 2 Then

Update notificationtype Set

NotificationTypeName= pNotificationTypeName,

OrgID= pOrgID

Where NotificationTypeID= pNotificationTypeID

;

Else

Select \* from notificationtype;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_odapplied` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_odapplied`(

in pOdID text,

in pUserID int(11),

in pOdDate date,

in pOdStartTime datetime,

in pOdEndTime datetime,

in pOdReason text,

in pOdLocationLatLon varchar(255),

in pOdLocation varchar(255),

in pAppliedDate datetime,

in pRequestStatus varchar(10),

in pStatusUpdateDate datetime,

in pStatusUpdteBy int(11),

in pTypeFlg smallint(1)

)

BEGIN

DECLARE COMPOFID int;

DECLARE i,NOOFOD int;

start transaction;

if pTypeFlg= 1 Then

INSERT INTO odapplied

(

UserID,

OdDate,

OdStartTime,

OdEndTime,

OdReason,

OdLocationLatLon,

OdLocation,

AppliedDate,

RequestStatus,

StatusUpdateDate,

StatusUpdteBy

)

VALUES

(

pUserID,

pOdDate,

pOdStartTime,

pOdEndTime,

pOdReason,

pOdLocationLatLon,

pOdLocation,

pAppliedDate,

pRequestStatus,

pStatusUpdateDate,

pStatusUpdteBy

);

/\*SET COMPOFID = LAST\_INSERT\_ID();\*/

elseif pTypeFlg= 2 Then

call splitter(pOdID,',');

select count(SplitValue) INTO NOOFOD from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFOD DO

Update odapplied Set

RequestStatus= pRequestStatus,

StatusUpdateDate= pStatusUpdateDate,

StatusUpdteBy= pStatusUpdteBy

Where OdID= (select SplitValue from splitResultsTmpTable where TmpID = i);

SET i = i + 1;

END WHILE;

elseif pTypeFlg= 3 Then

Select OD.OdID,OD.UserID,OD.OdDate,OD.OdStartTime,OD.OdEndTime,

(TIMESTAMPDIFF(Second,OD.OdStartTime, OD.OdEndTime)) TotalHours,

OD.OdReason,OdLocationLatLon,OD.OdLocation,

OD.OdLocation,OD.AppliedDate,OD.RequestStatus

from odapplied OD where OD.UserID = pUserID order by OD.OdDate desc;

elseif pTypeFlg= 4 Then

Select OD.OdID,OD.UserID,OD.OdDate,OD.OdStartTime,OD.OdEndTime,

(TIMESTAMPDIFF(Second,OD.OdStartTime, OD.OdEndTime)) TotalHours,

OD.OdReason,OdLocationLatLon,OD.OdLocation,OD.AppliedDate,OD.RequestStatus

from odapplied OD

where OD.OdID IN (pOdID);

elseif pTypeFlg= 5 Then

Select OD.OdID,OD.OdDate,OD.OdStartTime,OD.OdEndTime,

(TIMESTAMPDIFF(Second,OD.OdStartTime, OD.OdEndTime)) TotalHours,

OdLocationLatLon,OD.OdLocation,OD.RequestStatus

from odapplied OD

where OD.RequestStatus = 'Approved' and OD.UserID = pUserID

and (OD.OdDate >= date(pOdStartTime) and OD.OdDate <= date(pOdEndTime));

else

select 1;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_shift` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_shift`(

In pShiftID int(11),

in pOrgID int(11),

in pUserID int(11),

in pShiftTitle varchar(255),

in pDescription text,

in pStartTime time,

in pEndTime time,

in pAddedOn datetime,

in pisActive char(1),

in pTypeFlg int(11)

)

BEGIN

DECLARE LASTSHIFTID int;

Declare UserRole varchar(10);

DECLARE EXIT HANDLER FOR 1451

SELECT 'FKC-ERROR'; /\*Foreign Key Constraint Error\*/

if pTypeFlg= 1 Then

INSERT INTO shift

(

OrgID,

UserID,

ShiftTitle,

Description,

StartTime,

EndTime,

AddedOn,

isActive

)

VALUES

(

pOrgID,

pUserID,

pShiftTitle,

pDescription,

pStartTime,

pEndTime,

pAddedOn,

pisActive

);

SET LASTSHIFTID = LAST\_INSERT\_ID();

select LASTSHIFTID;

elseif pTypeFlg= 2 Then

UPDATE shift S SET

S.ShiftTitle = pShiftTitle,

S.Description = pDescription,

S.StartTime = pStartTime,

S.EndTime = pEndTime,

S.isActive = pisActive

WHERE S.ShiftID = pShiftID;

elseif pTypeFlg= 3 Then

Select S.ShiftID,S.ShiftTitle,S.Description,

S.StartTime,S.EndTime,S.AddedOn,S.isActive

from shift S

where S.ShiftID=pShiftID AND S.OrgID=pOrgID;

elseif pTypeFlg= 4 Then

Delete from shift Where ShiftID= pShiftID;

select 'SUCCESS';

elseif pTypeFlg= 5 Then

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pUserID;

if UserRole = 'ADMIN' then

Select S.ShiftID,S.ShiftTitle,S.Description,

S.StartTime,S.EndTime,S.AddedOn,S.isActive

from shift S

where S.OrgID=pOrgID order by S.ShiftID desc;

else

Select S.ShiftID,S.ShiftTitle,S.Description,

S.StartTime,S.EndTime,S.AddedOn,S.isActive

from shift S

where S.OrgID=pOrgID and S.UserID = pUserID order by S.ShiftID desc;

end if;

elseif pTypeFlg= 6 Then

Select S.ShiftID,S.ShiftTitle,S.Description,

S.StartTime,S.EndTime,S.AddedOn,S.isActive

from shift S

where S.OrgID= pOrgID and S.UserID = pUserID and S.ShiftTitle= pShiftTitle;

elseif pTypeFlg= 7 Then

Select ST.ShiftTypeTitle from shifttype ST

where ST.OrgID=pOrgID and ST.isActive = 'Y'

and (pStartTime >= StartTime and pStartTime <= EndTime) limit 0,1;

elseif pTypeFlg= 8 Then

UPDATE shift S SET

S.isActive = pisActive

WHERE S.ShiftID = pShiftID;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_shiftAssignUser` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_shiftAssignUser`(

in pShiftID int(11),

in pOrgID int(11),

in pUserID text,

in pAddedOn datetime,

in pTypeFlg int(11)

)

BEGIN

DECLARE i,NOOFUSERSHFT int;

DECLARE USERSHFT text;

if pTypeFlg= 1 Then

start transaction;

/\*Delete from all related tables\*/

delete from usershift where ShiftID = pShiftID;

/\*Insert into usershift table\*/

if pUserID!='' then

call splitter(pUserID,',');

select count(SplitValue) INTO NOOFUSERSHFT from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFUSERSHFT DO

select SplitValue INTO USERSHFT from splitResultsTmpTable where TmpID = i;

INSERT INTO usershift

(

ShiftID,

UserID,

EffectedFrom,

AddedOn

)

VALUES

(

pShiftID,

stringSplit(USERSHFT,'~',1),

stringSplit(USERSHFT,'~',2),

pAddedOn

);

SET i = i + 1;

END WHILE;

end if;

commit;

elseif pTypeFlg= 2 Then

select US.UserID,concat(U.FName,' ',U.LName) UName, U.EmployeeID,US.EffectedFrom,US.AddedOn

from usershift US LEFT JOIN userpersonalinfo U ON US.UserID = U.UserID

where US.ShiftID = pShiftID;

elseif pTypeFlg = 3 Then

if pShiftID != 0 then

select if(count(US.ShiftID)>0,'Y','N') as isShift,concat(U.FName,' ',U.LName) FullName,U.EmployeeID

from usershift US LEFT JOIN userpersonalinfo U ON US.UserID = U.UserID

where US.UserID = pUserID and US.EffectedFrom = pAddedOn and US.ShiftID != pShiftID;

else

select if(count(US.ShiftID)>0,'Y','N') as isShift,concat(U.FName,' ',U.LName) FullName,U.EmployeeID

from usershift US LEFT JOIN userpersonalinfo U ON US.UserID = U.UserID

where US.UserID = pUserID and US.EffectedFrom = pAddedOn;

end if;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_shiftType` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_shiftType`(

In pShiftTypeID int(11),

in pOrgID int(11),

in pShiftTypeTitle varchar(255),

in pDescription text,

in pStartTime time,

in pEndTime time,

in pAddedOn datetime,

in pisActive char(1),

in pTypeFlg int(11)

)

BEGIN

DECLARE LASTSHIFTTYPEID int;

DECLARE EXIT HANDLER FOR 1451

SELECT 'FKC-ERROR'; /\*Foreign Key Constraint Error\*/

if pTypeFlg= 1 Then

INSERT INTO shifttype

(

OrgID,

ShiftTypeTitle,

Description,

StartTime,

EndTime,

AddedOn,

isActive

)

VALUES

(

pOrgID,

pShiftTypeTitle,

pDescription,

pStartTime,

pEndTime,

pAddedOn,

pisActive

);

SET LASTSHIFTTYPEID = LAST\_INSERT\_ID();

select LASTSHIFTTYPEID;

elseif pTypeFlg= 2 Then

UPDATE shifttype ST SET

ST.ShiftTypeTitle = pShiftTypeTitle,

ST.Description = pDescription,

ST.StartTime = pStartTime,

ST.EndTime = pEndTime,

ST.isActive = pisActive

WHERE ST.ShiftTypeID = pShiftTypeID;

elseif pTypeFlg= 3 Then

Select ST.ShiftTypeID,ST.ShiftTypeTitle,ST.Description,

ST.StartTime,ST.EndTime,ST.AddedOn,ST.isActive

from shifttype ST

where ST.ShiftTypeID=pShiftTypeID AND ST.OrgID=pOrgID;

elseif pTypeFlg= 4 Then

Delete from shifttype Where ShiftTypeID= pShiftTypeID;

select 'SUCCESS';

elseif pTypeFlg= 5 Then

Select ST.ShiftTypeID,ST.ShiftTypeTitle,ST.Description,

ST.StartTime,ST.EndTime,ST.AddedOn,ST.isActive

from shifttype ST

where ST.OrgID=pOrgID order by ST.ShiftTypeID desc;

elseif pTypeFlg= 6 Then

Select ST.ShiftTypeID,ST.ShiftTypeTitle,ST.Description,

ST.StartTime,ST.EndTime,ST.AddedOn,ST.isActive

from shifttype ST

where ST.OrgID= pOrgID and

ST.ShiftTypeTitle= pShiftTypeTitle;

elseif pTypeFlg= 7 Then

if pShiftTypeID!=0 then

Select ST.ShiftTypeID,ST.ShiftTypeTitle,ST.Description,

ST.StartTime,ST.EndTime,ST.AddedOn,ST.isActive

from shifttype ST

where ST.OrgID= pOrgID and

ST.ShiftTypeID!= pShiftTypeID and

(

(ST.StartTime >= pStartTime and ST.EndTime <= pEndTime)

OR

(ST.EndTime >= pStartTime and ST.StartTime <= pEndTime)

);

else

Select ST.ShiftTypeID,ST.ShiftTypeTitle,ST.Description,

ST.StartTime,ST.EndTime,ST.AddedOn,ST.isActive

from shifttype ST

where ST.OrgID= pOrgID and

(

(ST.StartTime >= pStartTime and ST.EndTime <= pEndTime)

OR

(ST.EndTime >= pStartTime and ST.StartTime <= pEndTime)

);

end if;

elseif pTypeFlg= 8 Then

UPDATE shifttype ST SET

ST.isActive = pisActive

WHERE ST.ShiftTypeID = pShiftTypeID;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_supervisorapproval` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_supervisorapproval`(

in pSupervisorApprvID int(11),

in pUserID text,

in pSupervisorID int(11),

in pAppliedDate datetime,

in pRequestStatus varchar(10),

in pStatusNote text,

in pStatusUpdateDate datetime,

in pStatusUpdteBy int(11),

in pOrgID int(11),

in pTypeFlg smallint(1)

)

BEGIN

DECLARE SUPERVAPPROVL,isAlreadyRequested,isAlreadyRequestedID int;

DECLARE i,NOOFSPV int;

start transaction;

if pTypeFlg= 1 Then

select count(SAPV.SupervisorApprvID) INTO isAlreadyRequested

from supervisorapproval SAPV where SAPV.UserID = pUserID

AND SAPV.SupervisorID = pSupervisorID AND SAPV.OrgID = pOrgID;

if isAlreadyRequested = 0 then

INSERT INTO supervisorapproval

(

UserID,

SupervisorID,

AppliedDate,

RequestStatus,

StatusNote,

StatusUpdateDate,

StatusUpdteBy,

OrgID

)

VALUES

(

pUserID,

pSupervisorID,

pAppliedDate,

pRequestStatus,

pStatusNote,

pStatusUpdateDate,

pStatusUpdteBy,

pOrgID

);

else

select SAPV.SupervisorApprvID INTO isAlreadyRequestedID

from supervisorapproval SAPV where SAPV.UserID = pUserID

AND SAPV.SupervisorID = pSupervisorID AND SAPV.OrgID = pOrgID;

Update supervisorapproval Set

AppliedDate= pAppliedDate,

RequestStatus= pRequestStatus

Where SupervisorApprvID= isAlreadyRequestedID;

end if;

/\*SET SUPERVAPPROVL = LAST\_INSERT\_ID();\*/

elseif pTypeFlg= 2 Then

call splitter(pUserID,',');

select count(SplitValue) INTO NOOFSPV from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFSPV DO

Update supervisorapproval Set

RequestStatus= pRequestStatus,

StatusNote= pStatusNote,

StatusUpdateDate= pStatusUpdateDate,

StatusUpdteBy= pStatusUpdteBy

Where UserID = (select SplitValue from splitResultsTmpTable where TmpID = i) AND SupervisorID = pSupervisorID AND OrgID = pOrgID;

SET i = i + 1;

END WHILE;

elseif pTypeFlg= 3 Then

select SAPV.\*,

CONCAT(UP.FName,' ',UP.LName) as UserName,UP.EmployeeID as UserEmployeeID,UP.UserPicture

from supervisorapproval SAPV

LEFT JOIN userpersonalinfo UP ON SAPV.UserID = UP.UserID

where SAPV.RequestStatus = 'Applied' and SAPV.SupervisorID = pSupervisorID;

Else

select SAPV.UserID,SAPV.SupervisorID,SAPV.AppliedDate,SAPV.RequestStatus,SAPV.StatusNote,SAPV.StatusUpdateDate,SAPV.StatusUpdteBy,SAPV.OrgID

from supervisorapproval SAPV where SAPV.UserID IN (pUserID) AND SAPV.SupervisorID =pSupervisorID AND SAPV.OrgID = pOrgID;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_Survey` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_Survey`(

In pSurveyID int(11),

IN pOrigID int(11),

in pTitle varchar(200),

in pDescription varchar(200),

in pCreatedOn datetime(3),

in pExpiresOn datetime(3),

IN pUserID VARCHAR(15),

in pPublishDate Datetime,

In pTypeFlg Int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO survey

(

OrgID,

Title,

Description,

CreatedOn,

ExpiresOn,

CreatedBy,

PublishDate

)

VALUES

(

pOrigID ,

pTitle ,

pDescription ,

pCreatedOn ,

pExpiresOn ,

pUserID ,

pPublishDate

);

elseif pTypeFlg= 2 Then

Update survey Set

OrgID= pOrigID,

Title= pTitle,

Description= pDescription,

CreatedOn= pCreatedOn,

ExpiresOn= pExpiresOn,

CreatedBy= pUserID,

PublishDate= pPublishDate

Where SurveyID= pSurveyID

;

Else

Select \* from survey;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_SurveyOption` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_SurveyOption`(

In pSurveyOptionID int(11),

IN pQuestionID int(11),

in pOptionText varchar(50),

in pIsAnswer tinyint(4),

in pAnswerText Char(10),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO surveyoption

(

QuestionID,

OptionText,

IsAnswer,

AnswerText

)

VALUES

(

pQuestionID ,

pOptionText ,

pIsAnswer ,

pAnswerText

);

elseif pTypeFlg= 2 Then

Update surveyoption Set

QuestionID= pQuestionID,

OptionText= pOptionText,

IsAnswer= pIsAnswer,

AnswerText= pAnswerText

Where SurveyOptionID= pSurveyOptionID

;

Else

Select \* from surveyoption;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_surveyquestions` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_surveyquestions`(

In pSurveyQuestionID int(11),

IN pQuestionText varchar(200),

in pQuestionType varchar(200),

in pOptions varchar(2000),

in pSurveyID Int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO surveyquestions

(

QuestionText,

QuestionType,

Options,

SurveyID

)

VALUES

(

pQuestionText ,

pQuestionType ,

pOptions ,

pSurveyID

);

elseif pTypeFlg= 2 Then

Update surveyquestions Set

QuestionText= pQuestionText,

QuestionType= pQuestionType,

Options= pOptions,

SurveyID= pSurveyID

Where SurveyQuestionID= pSurveyQuestionID

;

Else

Select \* from surveyquestions;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_SurveyResponse` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_SurveyResponse`(

In pSurveyResponseID int(11),

IN pSurveyID int(11),

in pQuestionID int(11),

in pResponseText varchar(200),

in pUserID int(11),

in pUserEmail varchar(150),

in pSelectedOptionID int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO surveyresponse

(

SurveyID,

QuestionID,

ResponseText,

UserID,

UserEmail,

SelectedOptionID

)

VALUES

(

pSurveyID ,

pQuestionID ,

pResponseText ,

pUserID ,

pUserEmail ,

pSelectedOptionID

);

elseif pTypeFlg= 2 Then

Update surveyresponse Set

SurveyID= pSurveyID,

QuestionID= pQuestionID,

ResponseText= pResponseText,

UserID= pUserID,

UserEmail= pUserEmail,

SelectedOptionID= pSelectedOptionID

Where SurveyResponseID= pSurveyResponseID;

Else

Select \* from surveyresponse;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_teaminfo` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_teaminfo`(

In pTeamID int(11),

in pTeamName varchar(50),

in pDepartmentID int(11),

in pOrgID int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO teaminfo

(

TeamName,

DepartmentID,

OrgID

)

VALUES

(

pTeamName ,

pDepartmentID ,

pOrgID

);

elseif pTypeFlg= 2 Then

Update teaminfo Set

TeamName= pTeamName,

DepartmentID= pDepartmentID,

OrgID= pOrgID

Where TeamID= pTeamID

;

Else

Select \* from teaminfo;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_teamsupervisor` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_teamsupervisor`(

In pTeamSupervisorID int(11),

in pTeamID int(11),

in pSupervisorID int(11),

in pManagerID int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO teamsupervisor

(

TeamID,

SupervisorID,

ManagerID

)

VALUES

(

pTeamID ,

pSupervisorID ,

pManagerID

);

elseif pTypeFlg= 2 Then

Update teamsupervisor Set

TeamID= pTeamID,

SupervisorID= pSupervisorID,

ManagerID= pManagerID

Where TeamSupervisorID= pTeamSupervisorID

;

Else

Select \* from teamsupervisor;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_token` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_token`(

In pTokenID int(11),

in pUserID int(11),

in pAuthToken varchar(250),

in pIssuedOn datetime(3),

in pExpiredOn datetime(3),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO token

(

UserID,

AuthToken,

IssuedOn,

ExpiredOn

)

VALUES

(

pUserID ,

pAuthToken ,

pIssuedOn ,

pExpiredOn

);

elseif pTypeFlg= 2 Then

Update token Set

UserID= pUserID,

AuthToken= pAuthToken,

IssuedOn= pIssuedOn,

ExpiredOn= pExpiredOn

Where TokenID= pTokenID

;

Else

Select \* from token;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_userpersonalinfo` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_userpersonalinfo`(

in pUserID int(11),

in pOrgID int(11),

in pFName varchar(50),

in pLName varchar(50),

in pAddress varchar(150),

in pSiteLocation varchar(150),

in pEmail varchar(100),

in pWorkPhone varchar(50),

in pCellNo varchar(50),

in pEmployeeID varchar(50),

in pDeptID int(11),

in pAge varchar(5),

in pDateOfJoining varchar(10),

in pOthers longtext,

in pCreatedByUserID int(11),

in pPwd varchar(50),

in pPwdResetRequired tinyint(1),

in pDeleted tinyint(1),

in pDeptName varchar(50),

in pResignationDate date,

in pReleaseDate date,

in pReleaseType char(1),

in pEmploymentType varchar(30),

in pTypeFlg int(11)

)

BEGIN

DECLARE EXIT HANDLER FOR 1451

SELECT 'FKC-ERROR'; /\*Foreign Key Constraint Error\*/

start transaction;

if pTypeFlg= 1 Then

/\*select @vCnt :=count(DeptID) from department where DepartmentName= pDeptName;

if @vCnt=0 then

INSERT INTO department

(

DepartmentName,

OrgID

)

VALUES

(

pDeptName,

pOrgID

);

End if;

Select @vDeptID := DeptID from department where DepartmentName= pDeptName;\*/

INSERT INTO userpersonalinfo

(

OrgID,

FName,

LName,

Address,

SiteLocation,

Email,

WorkPhone,

CellNo,

EmployeeID,

DeptID,

Age,

DateOfJoining,

Others,

CreatedByUserID,

Pwd,

PwdResetRequired,

Deleted,

EmploymentType

)

VALUES

(

pOrgID ,

pFName ,

pLName ,

pAddress ,

pSiteLocation ,

pEmail ,

pWorkPhone ,

pCellNo ,

pEmployeeID ,

NULL ,

pAge ,

if(pDateOfJoining = '', NULL, pDateOfJoining) ,

pOthers ,

pCreatedByUserID,

pPwd ,

1,

pDeleted,

pEmploymentType

);

Select \* from userpersonalinfo where Email = pEmail;

elseif pTypeFlg= 2 Then

Update userpersonalinfo U Set

U.FName= pFName,

U.LName= pLName,

U.Address= pAddress,

U.SiteLocation = pSiteLocation,

U.WorkPhone= pWorkPhone,

U.CellNo= pCellNo,

U.EmployeeID= pEmployeeID,

U.Email= pEmail,

U.DateOfJoining = if(pDateOfJoining = '', NULL, pDateOfJoining),

U.DeptID= if(pDeptID = 0, NULL, pDeptID),

U.ResignationDate = pResignationDate,

U.ReleaseDate = pReleaseDate,

U.ReleaseType = pReleaseType,

U.EmploymentType = pEmploymentType

Where U.UserID= pUserID and U.OrgID = pOrgID;

elseif pTypeFlg= 3 Then

delete from userpersonalinfo where UserID = pUserID and OrgID = pOrgID;

select 'SUCCESS';

/\*De-activate User\*/

elseif pTypeFlg= 4 Then

Update userpersonalinfo U Set

U.Deactivated= 'Y',

U.DeactivatedOn = now()

Where U.UserID= pUserID and U.OrgID = pOrgID;

select 'SUCCESS';

/\*Re-activate User\*/

elseif pTypeFlg= 5 Then

Update userpersonalinfo U Set

U.Deactivated= 'N'

Where U.UserID= pUserID and U.OrgID = pOrgID;

select 'SUCCESS';

/\* check duplicate email\*/

elseif pTypeFlg= 6 Then

SELECT \* FROM userpersonalinfo U

WHERE U.Email= pEmail AND U.UserID != pUserID;

/\* check duplicate EmployeeID\*/

elseif pTypeFlg= 7 Then

SELECT \* FROM userpersonalinfo U

WHERE U.EmployeeID= pEmployeeID AND U.UserID != pUserID and U.OrgID = pOrgID;

Else

Select \* from userpersonalinfo where Email= pEmail and OrgID = pOrgID;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_userphoto` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_userphoto`(

In pUserID int(11),

in pOrgID int(11),

in pPhoto varchar(255)

)

BEGIN

Update userpersonalinfo Set

UserPicture = pPhoto

Where UserID= pUserID and OrgID= pOrgID;

Select UserID,UserPicture from userpersonalinfo Where UserID= pUserID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_userproperty` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_userproperty`(

In pUserPropertyID int(11),

IN pPropertyName varchar(50),

in pPropertyType varchar(50),

in pOrgID int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO userproperty

(

PropertyName,

PropertyType,

OrgID

)

VALUES

(

pPropertyName,

pPropertyType,

pOrgID

);

elseif pTypeFlg= 2 Then

Update userproperty Set

PropertyName= pPropertyName,

PropertyType= pPropertyType,

OrgID= pOrgID

Where UserPropertyID= pUserPropertyID

;

Else

Select \* from userproperty;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_userrole` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_userrole`(

In pRoleID int(11),

IN pRoleName varchar(50),

in pActions varchar(250),

in pOrgID int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO userrole

(

RoleName,

Actions,

OrgID

)

VALUES

(

pRoleName,

pActions,

pOrgID

);

elseif pTypeFlg= 2 Then

Update userrole Set

RoleName= pRoleName,

Actions= pActions,

OrgID= pOrgID

Where RoleID= pRoleID

;

Else

Select \* from userrole;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_usertype` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_usertype`(

In pUserTypeID int(11),

IN pUserEmploymentType varchar(50),

in pOrgID int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO usertype

(

UserEmploymentType,

OrgID

)

VALUES

(

pUserEmploymentType,

pOrgID

);

elseif pTypeFlg= 2 Then

Update usertype Set

UserEmploymentType= pUserEmploymentType,

OrgID= pOrgID

Where UserTypeID= pUserTypeID

;

Else

Select \* from usertype;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_userworkinfo` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_userworkinfo`(

In pUserWorkInfoID int(11),

IN pUserID int(11),

in pRoleD int(11),

in pTeamID int(11),

in pTypeFlg int(11)

)

BEGIN

start transaction;

if pTypeFlg= 1 Then

INSERT INTO userworkinfo

(

UserID,

RoleD,

TeamID

)

VALUES

(

pUserID,

pRoleD,

pTeamID

);

elseif pTypeFlg= 2 Then

Update userworkinfo Set

UserID= pUserID,

RoleD= pRoleD,

TeamID= pTeamID

Where UserWorkInfoID= pUserWorkInfoID

;

Else

Select \* from userworkinfo;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_visitor` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_visitor`(

in pVisitorID int(11),

in pOrgID int(11),

in pFullName varchar(255),

in pCompanyName varchar(255),

in pAddress text,

in pMobileNo varchar(15),

in pEmail varchar(255),

in pPhoto text,

in pSignature text,

in pSignUpDate datetime,

in pTypeFlg smallint(1)

)

BEGIN

DECLARE isExistVisitor,SLTDVISITORID int(5);

if pTypeFlg= 1 Then

select count(V.VisitorID) into isExistVisitor from visitor V

where V.MobileNo = pMobileNo and V.OrgID = pOrgID;

if isExistVisitor = 0 then

start transaction;

INSERT INTO visitor

(

OrgID,

FullName,

CompanyName,

Address,

MobileNo,

Email,

Photo,

Signature,

SignUpDate

)

VALUES

(

pOrgID,

pFullName,

pCompanyName,

pAddress,

pMobileNo,

pEmail,

pPhoto,

pSignature,

pSignUpDate

);

SET SLTDVISITORID = LAST\_INSERT\_ID();

select SLTDVISITORID as VISITORID;

commit;

else

select V.VisitorID as VISITORID from visitor V where V.MobileNo = pMobileNo and V.OrgID = pOrgID;

end if;

elseif pTypeFlg= 2 Then

Update visitor Set

FullName= pFullName,

CompanyName = pCompanyName,

Address = pAddress,

Email = pEmail,

Photo = pPhoto,

Signature = pSignature

Where VisitorID= pVisitorID;

else

Select V.VisitorID,V.FullName,V.CompanyName,V.Address,V.MobileNo,V.Email,V.Photo,V.Signature

from visitor V where V.MobileNo = pMobileNo and V.OrgID = pOrgID;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_visitorAttendance` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_visitorAttendance`(

in pVisitorAttendanceID text,

in pVisitorID int(11),

in pMeetingID int(11),

in pCheckInTime datetime,

in pCheckInLocation varchar(255),

in pCheckOutTime datetime,

in pCheckOutLocation varchar(255),

in pMeetToName varchar(255),

in pMeetToMobileNo varchar(15),

in pMeetToUserID int(11),

in pMeetReason text,

in pLogTime datetime,

in pTypeFlg char(20)

)

BEGIN

DECLARE VISITORATTNID,SELECTEDMEETID,i,NOOFVISITORATTNID int;

DECLARE VISITORATTNSTS varchar(50);

if pTypeFlg= 'CHECKIN' Then

start transaction;

INSERT INTO visitorattendance

(

VisitorID,

MeetingID,

CheckInTime,

CheckInLocation,

MeetToName,

MeetToMobileNo,

MeetToUserID,

MeetReason,

LogTime

)

VALUES

(

pVisitorID,

pMeetingID,

pCheckInTime,

pCheckInLocation,

pMeetToName,

pMeetToMobileNo,

pMeetToUserID,

pMeetReason,

pLogTime

);

SET VISITORATTNID = LAST\_INSERT\_ID();

IF pMeetingID=0 THEN

UPDATE visitorattendance SET MeetingID=VISITORATTNID,VisitorType='P' WHERE VisitorAttendanceID=VISITORATTNID;

ELSE

UPDATE visitorattendance SET MeetingID=pMeetingID,VisitorType='C' WHERE VisitorAttendanceID=VISITORATTNID;

END IF;

select VISITORATTNID;

commit;

elseif pTypeFlg= 'CHECKOUT' Then

/\*Static Card\*/

if pMeetingID >0 then

select VA.VisitorAttendanceID,VA.VisitAttenStatus into VISITORATTNID,VISITORATTNSTS

from visitorattendance VA

left join visitor V ON VA.VisitorID = V.VisitorID

where V.OrgID = pMeetingID and VA.StaticQrCardNo = pVisitorID

order by VA.VisitorAttendanceID desc limit 0,1;

/\*Dynamic Card\*/

else

select VA.VisitorAttendanceID,VA.VisitAttenStatus into VISITORATTNID,VISITORATTNSTS

from visitorattendance VA

left join visitor V ON VA.VisitorID = V.VisitorID

where V.VisitorID = pVisitorID

order by VA.VisitorAttendanceID desc limit 0,1;

end if;

case VISITORATTNSTS

WHEN 'CheckedOut' THEN

Update visitorattendance Set

CheckOutLocation= pCheckOutLocation,

CheckOutTime= pCheckOutTime

Where VisitorAttendanceID= VISITORATTNID;

ELSE

Update visitorattendance Set

VisitAttenStatus= 'CheckedOut-NotVarified',

CheckOutLocation= pCheckOutLocation,

CheckOutTime= pCheckOutTime

Where VisitorAttendanceID= VISITORATTNID;

END CASE;

select V.VisitorID,V.FullName,VA.VisitorAttendanceID,VA.MeetToUserID

from visitorattendance VA LEFT JOIN visitor V ON VA.VisitorID = V.VisitorID

where VisitorAttendanceID= VISITORATTNID;

elseif pTypeFlg= 'CANCEL' Then

Update visitorattendance Set

VisitAttenStatus= 'Cancelled'

Where VisitorAttendanceID= pVisitorAttendanceID;

elseif pTypeFlg= 'APRVCHECKIN' Then

call splitter(pVisitorAttendanceID,',');

select count(SplitValue) INTO NOOFVISITORATTNID from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFVISITORATTNID DO

select MeetingID into SELECTEDMEETID from

visitorattendance where VisitorAttendanceID = (select SplitValue from splitResultsTmpTable where TmpID = i);

Update visitorattendance Set

MeetingStatus= if(pCheckInLocation='Accepted','Accepted','Rejected'),

VisitAttenStatus= if(VisitAttenStatus='CheckedIn',if(pCheckInLocation='Accepted','CheckedIn-Accepted','CheckedIn-Rejected'),VisitAttenStatus),

HostCheckInRqstStsLogTime = pLogTime

Where VisitAttenStatus IN ('CheckedIn','CheckedOut-NotVarified') and VisitorAttendanceID IN

(Select VisitorAttendanceID from (Select VisitorAttendanceID from visitorattendance where MeetingID = SELECTEDMEETID) as temp);

SET i = i + 1;

END WHILE;

elseif pTypeFlg= 'APRVCHECKOUT' Then

select MeetingID into SELECTEDMEETID from

visitorattendance where VisitorAttendanceID= pVisitorAttendanceID;

Update visitorattendance Set

VisitAttenStatus= 'CheckedOut',

HostSignedOffLogTime = pLogTime

Where MeetingStatus='Accepted' and VisitAttenStatus IN('CheckedIn-Accepted','CheckedOut-NotVarified') and VisitorAttendanceID IN

(Select VisitorAttendanceID from (Select VisitorAttendanceID from visitorattendance where MeetingID = SELECTEDMEETID) as temp);

elseif pTypeFlg= 'PNDCHECKOUTLIST' Then

Select VA.MeetingID,VA.VisitorType,V.VisitorID,V.FullName,V.CompanyName,V.Address,V.MobileNo,V.Email,V.Photo,V.Signature,

VA.VisitorAttendanceID,VA.CheckInTime,VA.CheckInLocation,VA.CheckOutTime,VA.CheckOutLocation,VA.MeetToName,VA.MeetToMobileNo,VA.MeetToUserID,

VA.MeetReason,VA.VisitAttenStatus,VA.MeetingStatus

from visitorattendance VA

LEFT JOIN visitor V ON VA.VisitorID = V.VisitorID

where VA.MeetToUserID = pMeetToUserID and VisitAttenStatus IN ('CheckedIn','CheckedIn-Accepted','CheckedOut-NotVarified')

order by VA.MeetingID desc,VA.VisitorAttendanceID asc;

elseif pTypeFlg= 'APPRVCHECKOUTLIST' Then

Select VA.MeetingID,VA.VisitorType,V.VisitorID,V.FullName,V.CompanyName,V.Address,V.MobileNo,V.Email,V.Photo,V.Signature,

VA.VisitorAttendanceID,VA.CheckInTime,VA.CheckInLocation,VA.MeetToName,VA.MeetToMobileNo,VA.MeetToUserID,

VA.MeetReason,VA.VisitAttenStatus,VA.MeetingStatus

from visitorattendance VA

LEFT JOIN visitor V ON VA.VisitorID = V.VisitorID

where VA.MeetToUserID = pMeetToUserID and VisitAttenStatus = 'CheckedOut'

order by VA.MeetingID desc,VA.VisitorAttendanceID asc;

elseif pTypeFlg= 'ALLCHECKOUTLIST' Then

Select VA.MeetingID,VA.VisitorType,V.VisitorID,V.FullName,V.CompanyName,V.Address,V.MobileNo,V.Email,V.Photo,V.Signature,

VA.VisitorAttendanceID,VA.CheckInTime,VA.CheckInLocation,VA.CheckOutTime,VA.CheckOutLocation,VA.MeetToName,VA.MeetToMobileNo,VA.MeetToUserID,

VA.MeetReason,VA.VisitAttenStatus,VA.MeetingStatus,VA.HostCheckInRqstStsLogTime,VA.HostSignedOffLogTime,VA.StaticQrCardNo

from visitorattendance VA

LEFT JOIN visitor V ON VA.VisitorID = V.VisitorID

where VA.MeetToUserID = pMeetToUserID

order by VA.MeetingID desc,VA.VisitorAttendanceID asc;

Else

Select VA.MeetingID,VA.VisitorType,V.VisitorID,V.OrgID,V.FullName,V.CompanyName,V.Address,V.MobileNo,V.Email,V.Photo,V.Signature,

VA.VisitorAttendanceID,VA.CheckInTime,VA.CheckInLocation,VA.MeetToName,VA.MeetToMobileNo,VA.MeetToUserID,

VA.MeetReason,VA.VisitAttenStatus,VA.MeetingStatus

from visitorattendance VA

LEFT JOIN visitor V ON VA.VisitorID = V.VisitorID

where VA.VisitorAttendanceID = pVisitorAttendanceID

order by VA.MeetingID desc,VA.VisitorAttendanceID asc;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_weekends` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_weekends`(

In pWeekendsID int(11),

in pDayName varchar(155),

in pOrgID int(11),

in pShowYear varchar(4),

in pTypeFlg int(11)

)

BEGIN

DECLARE LASTWEEKENDSID int;

start transaction;

if pTypeFlg= 1 Then

INSERT INTO weekends

(

DayName,

OrgID

)

VALUES

(

pDayName,

pOrgID

);

SET LASTWEEKENDSID = LAST\_INSERT\_ID();

select LASTWEEKENDSID;

elseif pTypeFlg= 2 Then

Update weekends Set

DayName= pDayName

Where WeekendsID= pWeekendsID

;

elseif pTypeFlg= 3 Then

if pShowYear='' Then

Select W.DayName, W.OrgID

from weekends W

where W.WeekendsID=pWeekendsID AND W.OrgID=pOrgID order by W.WeekendsID desc;

Else

Select W.DayName, W.OrgID

from weekends W

where W.WeekendsID=pWeekendsID AND W.OrgID=pOrgID

order by W.WeekendsID desc;

End If;

elseif pTypeFlg= 4 Then

Delete from departmentweekends

Where WeekendsID= pWeekendsID;

Delete from weekends

Where WeekendsID= pWeekendsID;

elseif pTypeFlg= 5 Then

Select W.DayName, W.OrgID, W.WeekendsID

from weekends W

where W.OrgID=pOrgID

order by W.WeekendsID desc;

elseif pTypeFlg= 6 Then

Select W.WeekendsID

from weekends W

where W.OrgID=pOrgID AND W.DayName=pDayName;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_AddUpdateShow\_weekends2` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_AddUpdateShow\_weekends2`(

In pWeekendsID int(11),

in pDayName varchar(155),

in pOrgID int(11),

in pShowYear varchar(4),

in pTypeFlg int(11)

)

BEGIN

DECLARE LASTWEEKENDSID int;

start transaction;

if pTypeFlg= 1 Then

INSERT INTO weekends

(

DayName,

OrgID

)

VALUES

(

pDayName,

pOrgID

);

SET LASTWEEKENDSID = LAST\_INSERT\_ID();

select LASTWEEKENDSID;

elseif pTypeFlg= 2 Then

Update weekends Set

DayName= pDayName

Where WeekendsID= pWeekendsID

;

elseif pTypeFlg= 3 Then

if pShowYear='' Then

Select W.DayName, W.OrgID

from weekends W

where W.WeekendsID=pWeekendsID AND W.OrgID=pOrgID order by W.WeekendsID desc;

Else

Select W.DayName, W.OrgID

from weekends W

where W.WeekendsID=pWeekendsID AND W.OrgID=pOrgID

order by W.WeekendsID desc;

End If;

elseif pTypeFlg= 4 Then

Delete from weekends

Where WeekendsID= pWeekendsID;

elseif pTypeFlg= 5 Then

Select W.DayName, W.OrgID, W.WeekendsID

from weekends W

where W.OrgID=pOrgID

order by W.WeekendsID desc;

elseif pTypeFlg= 6 Then

Select W.WeekendsID

from weekends W

where W.OrgID=pOrgID AND W.DayName=pDayName;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_addUserPersonalInfo` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_addUserPersonalInfo`(

in `@pFName` varchar(50),

in `@pLName` varchar(50),

in `@pAddress` varchar(150),

in `@pEmail` varchar(50),

in `@pWorkPhone` varchar(20),

in `@pCellNo` varchar(10),

in `@OrgID` int(11),

in `@CreatedByUserID` int(11),

in `@pPassword` varchar(50),

OUT `@pUserID` INT( 11)

)

BEGIN

INSERT INTO userprsonalinfo(FName,LName,Address,Email,WorkPhone,CellNo,CreatedByUserID,OrigId)

VALUES(@pFName,@pLName,@pAddress,@pEmail,@pWorkPhone,@pCellNo,@pCreatedByUserID,@pOrigId);

SET @pUserID = LAST\_INSERT\_ID();

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Add\_leavecredit` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Add\_leavecredit`(

IN pUserID int(11),

IN pLeaveTypeID int(11),

IN pCreditLeave float(5,1),

IN pCreditDate datetime

)

BEGIN

start transaction;

INSERT INTO userleavecredit

(

UserID,

LeaveTypeID,

CreditLeave,

CreditDate

)

VALUES

(

pUserID,

pLeaveTypeID,

pCreditLeave,

pCreditDate

);

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_CreateAdmin` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_CreateAdmin`(

in pOrgID int(11),

in pFName varchar(50),

in pLName varchar(50),

in pAddress varchar(150),

in pEmail varchar(100),

in pWorkPhone varchar(50),

in pCellNo varchar(50),

in pEmployeeID varchar(50),

in pDeptID int(11),

in pAge varchar(5),

in pOthers longtext,

in pCreatedByUserID int(11),

in pPwd varchar(50),

in pPwdResetRequired tinyint(1),

in pDeleted tinyint(1),

in pRole varchar(50)

)

BEGIN

start transaction;

INSERT INTO userpersonalinfo

(

OrgID,

FName,

LName,

Address,

Email,

WorkPhone,

CellNo,

EmployeeID,

DeptID,

Age,

Others,

CreatedByUserID,

Pwd,

PwdResetRequired,

Deleted

)

VALUES

(

pOrgID ,

pFName ,

pLName ,

pAddress ,

pEmail ,

pWorkPhone ,

pCellNo ,

pEmployeeID ,

pDeptID ,

pAge ,

pOthers ,

pCreatedByUserID,

pPwd ,

pPwdResetRequired,

pDeleted

);

Select @vUserId := UserID from userpersonalinfo where Email = pEmail;

/\*Check if the Role Already Exists for the Organization, If Not create it\*/

Select @vRole := Count(RoleName) from userrole where OrgID = pOrgID and UPPER(RoleName) = pRole;

If @vRole < 1 Then

INSERT INTO userrole

(

RoleName,

Actions,

OrgId

)

Values

(

pRole,

'',

pOrgID

);

End If;

/\*Select the RoleID for the Role from the table into a variable to be used\*/

Select @vRoleID := IFNULL(RoleID, 0) from userrole where OrgID = pOrgID and UPPER(RoleName) = pRole;

/\*Check if the Role Already Exists for the Organization, If Not create it\*/

/\*Since the User is newly Created User, Add him to the Role in the Organization\*/

INSERT INTO userworkinfo

(

UserID,

RoleID,

TeamID

)

Values

(

@vUserId,

@vRoleID,

0

);

/\*Return the newly Created User to the Calling Program\*/

commit;

Select \* from userpersonalinfo where Email = pEmail;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_CreateAdminSupervisor` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_CreateAdminSupervisor`(

IN pRoleName varchar(50),

in pOrgID int(11),

in pFName varchar(50),

in pLName varchar(50),

in pEmail varchar(100),

in pCellNo varchar(50),

in pDeptID int(11),

in pDeptName varchar(50),

in pDesignation varchar(100),

in pDeleted tinyint(1),

in pPwd varchar(50)

)

BEGIN

Select @vCtrl := count(RoleID) from userrole where RoleName= pRoleName and OrgID = pOrgID;

If @vCtrl=0 then

INSERT INTO userrole

(

RoleName,

OrgID

)

VALUES

(

pRoleName,

pOrgID

);

End if;

/\*Additional Queries\*/

/\*Select CEO Dept ID\*/

select DeptID INTO @lpDeptID from department where DeptPrtID = 0 and OrgID = pOrgID order by DeptID limit 1;

select @vCnt :=count(DeptID) from department where DepartmentName= pDeptName and OrgID = pOrgID;

if @vCnt=0 then

INSERT INTO department

(

DeptPrtID,

DepartmentName,

OrgID

)

VALUES

(

@lpDeptID,

pDeptName,

pOrgID

);

End if;

Select @vDeptID := DeptID from department where DepartmentName= pDeptName and OrgID = pOrgID;

INSERT INTO userpersonalinfo

(

OrgID,

FName,

LName,

Email,

CellNo,

DeptID,

Pwd,

PwdResetRequired,

Deleted

)

VALUES

(

pOrgID ,

pFName ,

pLName ,

pEmail ,

pCellNo ,

@vDeptID ,

pPwd ,

1,

0

);

Select @vRoleID := RoleID from userrole where RoleName= pRoleName and OrgID = pOrgID;

Select @vUserID := UserID from userpersonalinfo Where Email= pEmail;

INSERT INTO userworkinfo

(

UserID,

RoleID,

Designation

)

VALUES

(

@vUserID,

@vRoleID,

pDesignation

);

Select @vCt := Count(idDeptManager)

from deptmanager Where DeptID= @vDeptID;

if @vCt=0 then

insert into deptmanager

(

DeptID,

DeptManagerID

)

Values

(

@vDeptID,

@vUserID

);

else

Update deptmanager set DeptManagerID= @vUserID Where DeptID= @vDeptID;

End if;

Select \* from userpersonalinfo where UserID = @vUserID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_CurtailedODOfUser` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_CurtailedODOfUser`(

IN pAttnTime datetime,

IN pOrgID int(11),

IN pUserID int(11),

IN pAttnType char(2)

)

BEGIN

DECLARE TARGETODID int(11);

set TARGETODID = 0;

select ifnull(OD.OdID,0) INTO TARGETODID from odapplied OD

where OD.UserID = pUserID

and OD.OdDate = date(pAttnTime)

and (pAttnTime >= OD.OdStartTime and pAttnTime <= OD.OdEndTime)

and OD.RequestStatus IN ('Applied','Approved');

if(TARGETODID > 0) then

if(pAttnType='CI') then

UPDATE odapplied SET OdEndTime = DATE\_SUB(pAttnTime, INTERVAL 1 SECOND) WHERE OdID = (select TARGETODID) and UserID = pUserID;

elseif(pAttnType='CO') then

UPDATE odapplied SET OdStartTime = DATE\_ADD(pAttnTime, INTERVAL 1 SECOND) WHERE OdID = (select TARGETODID) and UserID = pUserID;

end if;

end if;

select TARGETODID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetAllUsersByReleaseDate` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetAllUsersByReleaseDate`(in pOrgID int(11),in pReleaseDate date)

BEGIN

Select UP.UserID,UP.ReleaseDate,UP.OrgID,UP.FName,UP.LName,UP.Email,UP.EmployeeID

from userpersonalinfo UP

where UP.ReleaseDate = pReleaseDate

AND UP.Deactivated='N'

AND UP.OrgID=pOrgID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetApprovedLeaveOfUser` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetApprovedLeaveOfUser`(

IN `pStartDate` date,

IN `pEnddate` date,

IN `pOrgID` INT,

IN `pUserID` INT

)

BEGIN

/\*Get Approved and Applied Leaves of a User\*/

SELECT U.UserID,

LA.LeavAppID,LA.StartDate,LA.EndDate,LA.AppliedDate,LA.PolicyType,

LA.NoOfDays,

LA.IsHalfDay,LA.WhichHalf,

LS.LeaveMasterStatusID,LS.LeaveStatusText,LT.LeaveTypeText

FROM leaveapplied LA

LEFT JOIN leaveappliedstatuslog LLOG ON LLOG.LeavAppID = LA.LeavAppID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

WHERE LA.LeaveStatusID = LLOG.LeaveStatusID

AND U.UserID = pUserID

AND (

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

)

AND LS.LeaveMasterStatusID IN (1,2) ORDER BY LA.StartDate DESC;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetApprovedODOfDept` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetApprovedODOfDept`(

IN `pStartDate` varchar (50),

IN `pEnddate` varchar (50),

IN `pDeptID` INT,

IN `pOrgID` INT,

IN `pUserID` INT

)

BEGIN

Declare UserRole varchar(10);

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pUserID;

/\*Get all hierarchy sub department ids\*/

CALL nex\_GetDeptTree(pDeptID,@AllHrcyDept);

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

Select OD.OdID,OD.OdDate,OD.OdStartTime,OD.OdEndTime,

(TIMESTAMPDIFF(Second,OD.OdStartTime, OD.OdEndTime)) TotalHours,

OdLocationLatLon,OD.OdLocation,OD.RequestStatus,OD.UserID,U.FName,U.LName,U.EmployeeID,U.EmploymentType

from odapplied OD LEFT JOIN userpersonalinfo U ON OD.UserID = U.UserID

where OD.RequestStatus = 'Approved'

and OD.UserID IN

(Select U.UserID from userpersonalinfo U where ((U.DeptID IN (select \* from temp\_AllDept)) or (UserRole = 'ADMIN' AND U.DeptID IS NULL))

and U.OrgID = pOrgID)

and (OD.OdDate >= date(pStartDate) and OD.OdDate <= date(pEnddate));

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetAttendanceCountOfDept` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetAttendanceCountOfDept`(

IN pDeptID text,

IN pDate date,

OUT TotalEmp int(5),

OUT TotalAbsent int(5),

OUT TotalPresent int(5)

)

BEGIN

DECLARE a,p,t INT DEFAULT 0;

DECLARE pDeptIDAll text;

DECLARE DeptMngUserID INT;

call splitter(pDeptID,',');

select group\_concat(SplitValue) INTO pDeptIDAll from splitResultsTmpTable;

/\*Get all hierarchy sub department ids\*/

CALL nex\_GetDeptTree(pDeptIDAll,@AllHrcyDept);

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

SELECT DM.DeptManagerID INTO DeptMngUserID from deptmanager DM

where DM.DeptID IN (pDeptIDAll);

SELECT count(U.UserID) INTO t from userpersonalinfo U

where U.Deactivated = 'N' and U.DeptID IN (select \* from temp\_AllDept) and U.UserID NOT IN (DeptMngUserID);

SELECT count(A.AttendanceID) INTO p from attendance A

LEFT JOIN userpersonalinfo U ON A.UserID = U.UserID

where U.Deactivated = 'N' and U.DeptID IN (select \* from temp\_AllDept) AND DATE(A.LogTime) = pDate;

SET a = t - p;

select t into TotalEmp;

select a into TotalAbsent;

select p into TotalPresent;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetAttendanceForUser` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetAttendanceForUser`(

in pUserEmail varchar (50),

in pStartDate varchar (50),

in pEnddate varchar (50)

)

BEGIN

Select ATT.\* from Attendance ATT inner Join userpersonalinfo UP on ATT.UserID = UP.userID

where UP.Email = pUserEmail and Date(ATT.LogTime) >= Date(pStartDate) and Date(ATT.LogTime) <= Date(pEndDate);

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetAttendanceOfDept` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetAttendanceOfDept`(

IN `pStartDate` varchar (50),

IN `pEnddate` varchar (50),

IN `pDeptID` INT,

IN `pOrgID` INT,

IN `pAttenType` char(2),

IN `pUserID` INT

)

BEGIN

Declare UserRole varchar(10);

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pUserID;

if pAttenType = 'SD' then

Select

UP.UserID,UP.EmployeeID,UP.FName,UP.LName,UP.SiteLocation,

date(ATT.LogTime) as AttnDate,

Time(ATT.CheckInTime) as CheckInTime,

Time(ATT.CheckOutTime) CheckOutTime,

/\*(TIMESTAMPDIFF(Hour, ATT.CheckInTime, ATT.CheckOutTime) + (TIMESTAMPDIFF(Minute, ATT.CheckInTime, ATT.CheckOutTime) - ((TIMESTAMPDIFF(Hour, ATT.CheckInTime, ATT.CheckOutTime)) \* 60)) \*.01) TotalHours,\*/

(TIMESTAMPDIFF(Second, CheckInTime, CheckOutTime)) TotalHours,

ATT.CheckInLocation,ATT.CheckOutLocation,

ATT.AutoCheckOut

from attendance ATT inner Join userpersonalinfo UP on ATT.UserID = UP.UserID

where Date(ATT.LogTime) >= Date(pStartDate) and Date(ATT.LogTime) <= Date(pEndDate)

and ATT.UserID IN

/\*(Select U.UserID from userpersonalinfo U where (U.DeptID = pDeptID or U.DeptID IS NULL) \*/

(Select U.UserID from userpersonalinfo U where ((U.DeptID = pDeptID) or (UserRole = 'ADMIN' AND U.DeptID IS NULL))

and U.OrgID = pOrgID) order by UP.FName,ATT.LogTime asc;

else

/\*Get all hierarchy sub department ids\*/

CALL nex\_GetDeptTree(pDeptID,@AllHrcyDept);

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

Select

UP.UserID,UP.EmployeeID,UP.EmploymentType,UP.FName,UP.LName,UP.SiteLocation,

date(ATT.LogTime) as AttnDate,

Time(ATT.CheckInTime) as CheckInTime,

Time(ATT.CheckOutTime) CheckOutTime,

/\*(TIMESTAMPDIFF(Hour, ATT.CheckInTime, ATT.CheckOutTime) + (TIMESTAMPDIFF(Minute, ATT.CheckInTime, ATT.CheckOutTime) - ((TIMESTAMPDIFF(Hour, ATT.CheckInTime, ATT.CheckOutTime)) \* 60)) \*.01) TotalHours,\*/

(TIMESTAMPDIFF(Second, CheckInTime, CheckOutTime)) TotalHours,

ATT.CheckInLocation,ATT.CheckOutLocation,

ATT.AutoCheckOut

from attendance ATT inner Join userpersonalinfo UP on ATT.UserID = UP.UserID

where Date(ATT.LogTime) >= Date(pStartDate) and Date(ATT.LogTime) <= Date(pEndDate)

and ATT.UserID IN

(Select U.UserID from userpersonalinfo U where ((U.DeptID IN (select \* from temp\_AllDept)) or (UserRole = 'ADMIN' AND U.DeptID IS NULL))

and U.OrgID = pOrgID) order by UP.FName,ATT.LogTime asc;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetAttendanceOfOrg` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetAttendanceOfOrg`(

IN `pStartDate` datetime,

IN `pEndDate` datetime,

IN `pOrgID` INT

)

BEGIN

Select

ATT.AttendanceID,

ATT.UserID,

date(ATT.LogTime) as AttnDate,

ATT.LogTime as AttnDateTime,

Time(ATT.CheckInTime) as CheckInTime,

Time(ATT.CheckOutTime) CheckOutTime,

ATT.CheckInLocation,ATT.CheckOutLocation,

ATT.AutoCheckOut,UP.Fname,UP.Lname,UP.EmployeeID,DPM.DeptManagerID

from attendance ATT

left join userpersonalinfo UP on ATT.UserID = UP.UserID

left join deptmanager DPM on DPM.DeptID = UP.DeptID

where ATT.CheckInTime!='' and ATT.CheckOutTime!=''

and ATT.LogTime >= pStartDate and ATT.LogTime <= pEndDate and ATT.AutoCheckOut!='Y'

and UP.OrgID = pOrgID order by ATT.LogTime asc;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetAttendanceSummaryByType` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetAttendanceSummaryByType`(

IN `pOrgID` INT(11),

IN `pDeptIDOrLoc` text,

IN `pTargetStartDate` date,

IN `pTargetEndDate` date,

IN `pFilterBy` varchar(50),

IN `pGroupBy` varchar(50)

)

BEGIN

Declare NoOfUser int default 0;

Select count(UserID) into NoOfUser from userpersonalinfo where

OrgID=pOrgID AND IF(pGroupBy='dept',DeptID,SiteLocation)=pDeptIDOrLoc;

IF pGroupBy='dept' THEN

CASE pFilterBy

WHEN 'DP' THEN

SELECT date(A.LogTime) as `Date`,((COUNT(AttendanceID)/NoOfUser)\*100) as Percent FROM attendance A

LEFT JOIN userpersonalinfo UPI ON UPI.UserID=A.UserID

WHERE

UPI.DeptID=pDeptIDOrLoc

AND UPI.OrgID=pOrgID

AND date(A.LogTime) >= pTargetStartDate

AND date(A.LogTime) <= pTargetEndDate

GROUP BY DATE(A.LogTime);

WHEN 'OD' THEN

SELECT OD.OdDate as `Date`,((COUNT(OdID)/NoOfUser)\*100) as Percent FROM odapplied OD

LEFT JOIN userpersonalinfo UPI ON UPI.UserID=OD.UserID

WHERE

UPI.DeptID=pDeptIDOrLoc

AND UPI.OrgID=pOrgID

AND OD.OdDate >= pTargetStartDate

AND OD.OdDate <= pTargetEndDate

AND OD.RequestStatus='Approved'

GROUP BY OdDate;

WHEN 'ODP' THEN

SELECT CO.CompOffDate as `Date`,((COUNT(CompOffID)/NoOfUser)\*100) as Percent FROM compoffapplied CO

LEFT JOIN userpersonalinfo UPI ON UPI.UserID=CO.UserID

WHERE

UPI.DeptID=pDeptIDOrLoc

AND UPI.OrgID=pOrgID

AND CO.CompOffDate >= pTargetStartDate

AND CO.CompOffDate <= pTargetEndDate

AND CO.RequestStatus='Approved'

AND CO.RequestType='Payout'

GROUP BY CompOffDate;

END CASE;

ELSEIF pGroupBy='loc' THEN

CASE pFilterBy

WHEN 'DP' THEN

SELECT date(A.LogTime) as `Date`,((COUNT(AttendanceID)/NoOfUser)\*100) as Percent FROM attendance A

LEFT JOIN userpersonalinfo UPI ON UPI.UserID=A.UserID

WHERE

UPI.SiteLocation=pDeptIDOrLoc

AND UPI.OrgID=pOrgID

AND date(A.LogTime) >= pTargetStartDate

AND date(A.LogTime) <= pTargetEndDate

GROUP BY DATE(A.LogTime);

WHEN 'OD' THEN

SELECT OD.OdDate as `Date`,((COUNT(OdID)/NoOfUser)\*100) as Percent FROM odapplied OD

LEFT JOIN userpersonalinfo UPI ON UPI.UserID=OD.UserID

WHERE

UPI.SiteLocation=pDeptIDOrLoc

AND UPI.OrgID=pOrgID

AND OD.OdDate >= pTargetStartDate

AND OD.OdDate <= pTargetEndDate

AND OD.RequestStatus='Approved'

GROUP BY OdDate;

WHEN 'ODP' THEN

SELECT CO.CompOffDate as `Date`,((COUNT(CompOffID)/NoOfUser)\*100) as Percent FROM compoffapplied CO

LEFT JOIN userpersonalinfo UPI ON UPI.UserID=CO.UserID

WHERE

UPI.SiteLocation=pDeptIDOrLoc

AND UPI.OrgID=pOrgID

AND CO.CompOffDate >= pTargetStartDate

AND CO.CompOffDate <= pTargetEndDate

AND CO.RequestStatus='Approved'

AND CO.RequestType='Payout'

GROUP BY CompOffDate;

END CASE;

END IF;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetCalendarWOAndPH` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetCalendarWOAndPH`(

IN pUserID int(11),

IN pStartDate date,

IN pEndDate date,

IN pOrgID int(11)

)

BEGIN

declare i,NoOFTmpRecords,CalEffect INT DEFAULT 5;

/\*declare CalEffectStartDate,CalEffectEndDate DATE;\*/

declare CalEffectStartDate,CalEffectEndDate varchar(10);

declare AllHoldays,AllWeekends TEXT;

CREATE TEMPORARY TABLE UserAllAssignedCalendarsTmpTable (

TmpID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

UserID int(11) NOT NULL,

CalendarID int(11) NOT NULL,

EffectedFrom DATE NOT NULL

);

CREATE TEMPORARY TABLE UserGetAllHolidayWeekendTmpTable (

TmpHWID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

StartDate DATE ,

EndDate DATE ,

AllHolidays TEXT,

AllWeekends TEXT

);

INSERT INTO UserAllAssignedCalendarsTmpTable (UserID,CalendarID,EffectedFrom)

SELECT userid, calendarid, effectedfrom FROM usercalendar

WHERE userid = pUserID

AND effectedfrom <= pStartDate

ORDER BY effectedfrom DESC

LIMIT 1;

INSERT INTO UserAllAssignedCalendarsTmpTable (UserID,CalendarID,EffectedFrom)

SELECT userid, calendarid, effectedfrom FROM usercalendar

WHERE userid = pUserID

AND effectedfrom >= pStartDate AND effectedfrom <= pEndDate

ORDER BY effectedfrom ASC;

select count(TmpID) INTO NoOFTmpRecords from UserAllAssignedCalendarsTmpTable;

set i = 1;

WHILE i <= NoOFTmpRecords DO

set CalEffectStartDate = '';

set CalEffectEndDate = '';

select CalendarID,EffectedFrom INTO CalEffect,CalEffectStartDate from UserAllAssignedCalendarsTmpTable where TmpID = i order by EffectedFrom ASC;

if(i<NoOFTmpRecords) then

select IFNULL(EffectedFrom,'') INTO CalEffectEndDate from UserAllAssignedCalendarsTmpTable where TmpID = (i+1) order by EffectedFrom ASC;

set CalEffectEndDate = DATE\_SUB(CalEffectEndDate, INTERVAL 1 DAY);

else

set CalEffectEndDate = pEndDate;

end if;

select GROUP\_CONCAT(H.StartDate) INTO AllHoldays

from calendarholiday CH left join holiday H ON CH.HolidayID = H.HolidayID

where H.StartDate >=CalEffectStartDate and H.StartDate <=CalEffectEndDate

and CH.CalendarID = CalEffect order by H.StartDate asc;

select GROUP\_CONCAT(W.DayName) INTO AllWeekends

from calendarweekends CW left join weekends W ON CW.WeekendID = W.WeekendsID

where CW.CalendarID = CalEffect;

INSERT INTO UserGetAllHolidayWeekendTmpTable VALUES(NULL,CalEffectStartDate,CalEffectEndDate,AllHoldays,AllWeekends);

SET i = i + 1;

END WHILE;

select \* from UserGetAllHolidayWeekendTmpTable;

drop TABLE UserAllAssignedCalendarsTmpTable;

drop TABLE UserGetAllHolidayWeekendTmpTable;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetCheckedInStatusForEmp` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetCheckedInStatusForEmp`(

in pUserID int

)

BEGIN

/\*

Select COUNT(CheckInTime) as checkedIN from Attendance where UserID = pUserID

and DATE(checkinTime) = DATE(Now())

and

(

checkOutTime is null

or

CheckOutTime < (Select max(checkinTime) from checkindetailsmore where DATE(checkinTime) = DATE(Now()) and UserID = pUserID and TypeFlg = 'V')

);

\*/

Select COUNT(CheckInTime) as checkedIN from Attendance

where UserID = pUserID and checkOutTime is null

order by CheckInTime desc limit 0,1;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetCheckedOutStatusForEmp` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetCheckedOutStatusForEmp`(

in pUserID int

)

BEGIN

declare IsAttendance int;

select count(AttendanceID) into IsAttendance from attendance A where A.UserID = pUserID;

if IsAttendance > 0 then

/\*

Select COUNT(CheckOutTime) as checkedOut from Attendance where UserID = pUserID

and DATE(CheckOutTime) = DATE(Now());

\*/

Select COUNT(CheckOutTime) as checkedOut from Attendance

where UserID = pUserID and checkOutTime != null

order by CheckInTime desc limit 0,1;

else

select 1 as checkedOut;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetCompOffApplyValidate` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetCompOffApplyValidate`(

in pUserID int(11),

in pCompOffDate date,

in pRequestType varchar(6),

out IsAlrdyApplied char(5),

out IsLeaveAppliedVld char(5),

out IsOffDay char(5),

out OffDayType char(1),

out IsMorThnEightHrs int(11)

)

BEGIN

declare UserDept int;

declare IsMorThnEightHrsATTN,IsMorThnEightHrsOD,IsMorThnEightHrsTOTAL int default 0;

select '' into OffDayType;

select DeptID into UserDept from userpersonalinfo where UserID = pUserID;

if pRequestType = 'Payout' then

select if(count(COMP.CompOffID) > 0,'true','false') into IsAlrdyApplied from compoffapplied COMP

where COMP.UserID = pUserID AND COMP.CompOffDate = pCompOffDate AND (COMP.RequestType='Payout' OR COMP.RequestType='Leave')

AND COMP.RequestStatus IN ('Applied','Approved');

elseif pRequestType = 'Leave' then

select if(count(COMP.CompOffID) > 0,'true','false') into IsAlrdyApplied from compoffapplied COMP

where COMP.UserID = pUserID AND COMP.CompOffDate = pCompOffDate AND COMP.RequestType='Leave'

AND COMP.RequestStatus IN ('Applied','Approved');

end if;

select if(count(COMP.CompOffID) > 0,'true','false') into IsLeaveAppliedVld from compoffapplied COMP

where COMP.UserID = pUserID AND COMP.CompOffDate = pCompOffDate AND COMP.RequestType='Payout'

AND COMP.RequestStatus ='Approved';

if IsLeaveAppliedVld = 'false' then

select if(count(OD.OdID) > 0,'true','false') into IsLeaveAppliedVld from odapplied OD

where OD.UserID = pUserID AND OD.OdDate = pCompOffDate AND OD.RequestStatus = 'Approved';

end if;

select if(count(H.HolidayID) > 0,'true','false') into IsOffDay from holiday H

LEFT JOIN calendarholiday CH ON H.HolidayID = CH.HolidayID

LEFT JOIN usercalendar UC ON UC.CalendarID = CH.CalendarID

where UC.UserID = pUserID AND H.StartDate = pCompOffDate;

if IsOffDay = 'true' then

select 'H' into OffDayType;

elseif IsOffDay = 'false' then

select if(count(W.WeekendsID) > 0,'true','false') into IsOffDay from weekends W

LEFT JOIN calendarweekends CW ON W.WeekendsID = CW.WeekendID

LEFT JOIN usercalendar UC ON UC.CalendarID = CW.CalendarID

where UC.UserID = pUserID AND W.DayName = DAYNAME(pCompOffDate);

if IsOffDay = 'true' then

select 'W' into OffDayType;

end if;

end if;

select IFNULL(TIMESTAMPDIFF(SECOND,A.CheckInTime,A.CheckOutTime),0) into IsMorThnEightHrsATTN

FROM attendance A

WHERE A.UserID = pUserID AND DATE(A.LogTime) = pCompOffDate and A.AutoCheckOut IS NULL limit 0,1;

select COALESCE(SUM(TIMESTAMPDIFF(SECOND,OD.OdStartTime,OD.OdEndTime)),0) into IsMorThnEightHrsOD

FROM odapplied OD

WHERE OD.RequestStatus ='Approved' AND OD.UserID = pUserID AND DATE(OD.OdDate) = pCompOffDate;

Set IsMorThnEightHrsTOTAL = IsMorThnEightHrsATTN + IsMorThnEightHrsOD;

if(IsMorThnEightHrsTOTAL > 0) then

select IsMorThnEightHrsTOTAL into IsMorThnEightHrs;

else

select 0 into IsMorThnEightHrs;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetConsolidatedAttendanceForUser` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetConsolidatedAttendanceForUser`(

in pUserID int,

in pOnDate date,

in pOrgID int

)

BEGIN

select \* from

(

select 'CheckIn' as `Type`,

A.CheckInTime as `Time`,

A.CheckInLocation as `Location`,

'' as `Purpose`,

CIDM.OutDoorAttn as `OutDoorAttn`

from attendance A LEFT JOIN checkindetailsmore CIDM ON (A.UserID = CIDM.UserID and A.CheckInTime = CIDM.CheckIntime)

where A.UserID = pUserID

and date(LogTime) = pOnDate

UNION ALL

select 'CheckIn' as `Type`,

CIDM.CheckIntime as `Time`,

CIDM.Location as `Location`,

'' as `Purpose`,

CIDM.OutDoorAttn as `OutDoorAttn`

from checkindetailsmore CIDM

where CIDM.OrgID = pOrgID and CIDM.UserID = pUserID

and date(CONVERT\_TZ(CIDM.CheckIntime,'+00:00','+05:30')) = pOnDate and CIDM.TypeFlg = 'O'

UNION ALL

select 'CheckOut' as `Type`,

A.CheckOutTime as `Time`,

A.CheckOutLocation as `Location`,

CODM.Purpose as `Purpose`,

CODM.OutDoorAttn as `OutDoorAttn`

from attendance A LEFT JOIN checkoutdetailsmore CODM ON (A.UserID = CODM.UserID and A.CheckOutTime = CODM.CheckOutTime)

where A.UserID = pUserID

and date(LogTime) = pOnDate

and A.AutoCheckOut IS NULL

UNION ALL

select 'CheckOut' as `Type`,

CODM.CheckOuttime as `Time`,

CODM.Location as `Location`,

CODM.Purpose as `Purpose`,

CODM.OutDoorAttn as `OutDoorAttn`

from checkoutdetailsmore CODM

where CODM.OrgID = pOrgID and CODM.UserID = pUserID

and date(CONVERT\_TZ(CODM.CheckOuttime,'+00:00','+05:30')) = pOnDate

UNION ALL

select 'ODStart' as `Type`,

OD.OdStartTime as `Time`,

OD.OdLocation as `Location`,

'' as `Purpose`,

'' as `OutDoorAttn`

from odapplied OD

where OD.UserID = pUserID

and OD.RequestStatus = 'Approved'

and date(OD.OdDate) = pOnDate

UNION ALL

select 'ODEnd' as `Type`,

OD.OdEndTime as `Time`,

OD.OdLocation as `Location`,

OD.OdReason as `Purpose`,

'' as `OutDoorAttn`

from odapplied OD

where OD.UserID = pUserID

and OD.RequestStatus = 'Approved'

and date(OD.OdDate) = pOnDate

) ConsolidatedAttendance order by `Time` asc,`Purpose` asc;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetDashboardData` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetDashboardData`(

IN pDashboardDataID BIGINT(20),

IN pUserID BIGINT(20),

IN pOrgID BIGINT(20),

IN pDeptID BIGINT(20),

IN pStartDate date,

IN pEndDate date,

IN pDashboardData text,

IN pLst30DaysWorkingHrs text,

IN pTypeFlg SMALLINT(2)

)

BEGIN

if pTypeFlg = 1 then

select count(\*) as attandanceOfUser from attendance

where UserID = pUserID AND ( LogTime between date(pStartDate) AND date(pEndDate) );

elseif pTypeFlg = 2 then

select count(\*) as attendanceOfTeam from attendance a

LEFT JOIN userpersonalinfo upi ON upi.UserID = a.UserID

where a.UserID != pUserID

AND ( a.LogTime between date(pStartDate) AND date(pEndDate) )

AND upi.DeptID = pDeptID;

elseif pTypeFlg = 3 then

insert into dashboard\_data ( UserID, OrgID, OnDate, DashboardData, Last30DaysWorkingHours)

values (pUserID, pOrgID, pStartDate, pDashboardData, pLst30DaysWorkingHrs);

elseif pTypeFlg = 4 then

select DashboardDataID,UserID,OrgID,DashboardData,Last30DaysWorkingHours from dashboard\_data where UserID = pUserID AND OnDate = pStartDate;

elseif pTypeFlg = 5 then

Delete from dashboard\_data where DashboardDataID = pDashboardDataID;

elseif pTypeFlg = 6 then

select DashboardDataID,UserID,OrgID from dashboard\_data where UserID = pUserID;

End if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetDeptForOrg` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetDeptForOrg`(

in pOrgID INT

)

BEGIN

SELECT DepartmentName, DeptID from Department where orgID = pOrgID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetDeptTree` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetDeptTree`(

IN pDeptID text,

OUT AllHierarchyDepts text

)

BEGIN

DECLARE i,NOOFDEPT int;

DECLARE USERDEPT text;

CREATE TEMPORARY TABLE DeptTreeTmpTable (

DTTmpID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

DTDept TEXT NOT NULL

);

call splitter(pDeptID,',');

select count(SplitValue) INTO NOOFDEPT from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFDEPT DO

select SplitValue INTO USERDEPT from splitResultsTmpTable where TmpID = i;

INSERT INTO DeptTreeTmpTable (DTDept)

SELECT CONCAT\_WS(',',dept1,dept2,dept3,dept4,dept5,dept6,dept7,dept8,dept9,dept10) from (

(SELECT

GROUP\_CONCAT(DISTINCT d1.DeptID) as dept1,

GROUP\_CONCAT(DISTINCT d2.DeptID) as dept2,

GROUP\_CONCAT(DISTINCT d3.DeptID) as dept3,

GROUP\_CONCAT(DISTINCT d4.DeptID) as dept4,

GROUP\_CONCAT(DISTINCT d5.DeptID) as dept5,

GROUP\_CONCAT(DISTINCT d6.DeptID) as dept6,

GROUP\_CONCAT(DISTINCT d7.DeptID) as dept7,

GROUP\_CONCAT(DISTINCT d8.DeptID) as dept8,

GROUP\_CONCAT(DISTINCT d9.DeptID) as dept9,

GROUP\_CONCAT(DISTINCT d10.DeptID) as dept10

FROM department AS d1

LEFT JOIN department AS d2 ON d1.DeptID = d2.DeptPrtID

LEFT JOIN department AS d3 ON d2.DeptID = d3.DeptPrtID

LEFT JOIN department AS d4 ON d3.DeptID = d4.DeptPrtID

LEFT JOIN department AS d5 ON d4.DeptID = d5.DeptPrtID

LEFT JOIN department AS d6 ON d5.DeptID = d6.DeptPrtID

LEFT JOIN department AS d7 ON d6.DeptID = d7.DeptPrtID

LEFT JOIN department AS d8 ON d7.DeptID = d8.DeptPrtID

LEFT JOIN department AS d9 ON d8.DeptID = d9.DeptPrtID

LEFT JOIN department AS d10 ON d9.DeptID = d10.DeptPrtID

WHERE d1.DeptID = USERDEPT) tmp

);

SET i = i + 1;

END WHILE;

SELECT GROUP\_CONCAT(DISTINCT(DTDept)) into AllHierarchyDepts from DeptTreeTmpTable;

drop TABLE DeptTreeTmpTable;

/\*

SELECT GROUP\_CONCAT(HierarchyDept.DeptID SEPARATOR ',') as AllHierarchyDept INTO AllHierarchyDepts

from

(

select 1 as ID,pDeptID as DeptID

Union

select 1 as ID, DeptID

from (select \* from department

order by DeptPrtID, DeptID) base,

(select @pv := pDeptID) tmp

where find\_in\_set(DeptPrtID, @pv) > 0

and @pv := concat(@pv, ',', DeptID)

order by DeptID

) as HierarchyDept

GROUP BY HierarchyDept.ID;

\*/

/\*GET SELF JOIN DATA\*/

/\*

select pDeptID as DeptID

Union

select DeptID

from (select \* from department

order by DeptPrtID, DeptID) base,

(select @pv := pDeptID) tmp

where find\_in\_set(DeptPrtID, @pv) > 0

and @pv := concat(@pv, ',', DeptID)

order by DeptID;

\*/

/\*

SELECT

d1.DeptID AS dept1,

d2.DeptID as dept2,

d3.DeptID as dept3,

d4.DeptID as dept4,

d5.DeptID as dept5

FROM department AS d1

LEFT JOIN department AS d2 ON d1.DeptID = d2.DeptPrtID

LEFT JOIN department AS d3 ON d2.DeptID = d3.DeptPrtID

LEFT JOIN department AS d4 ON d3.DeptID = d4.DeptPrtID

LEFT JOIN department AS d5 ON d4.DeptID = d5.DeptPrtID

WHERE d1.DeptID = pDeptID;

\*/

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetDurationSummaryByType` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetDurationSummaryByType`(

IN `pOrgID` INT(11),

IN `pLoggedUserID` INT(11) ,

IN `pGroupByFlag` varchar(50)

)

BEGIN

Declare UserRole varchar(20);

Declare UserDept text;

/\* Get all sub depts of department of a manager\*/

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pLoggedUserID;

Select GROUP\_CONCAT(DeptID) into UserDept from deptmanager where DeptManagerID = pLoggedUserID;

CALL nex\_GetDeptTree(UserDept,@AllHrcyDept);

drop table if exists temp\_AllDept;

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

IF pGroupByFlag='employee' THEN

CASE UserRole

WHEN 'SUPERVISOR' THEN

SELECT DISTINCT UPI.UserID,CONCAT(UPI.FName,IF((UPI.LName!='' AND UPI.LName IS NOT NULL),CONCAT(' ',UPI.LName),'') ) AS Name FROM userpersonalinfo UPI WHERE UPI.OrgID=pOrgID

AND UPI.DeptID IN (SELECT \* from temp\_AllDept) AND UPI.UserID!=pLoggedUserID;

WHEN 'ADMIN' THEN

SELECT DISTINCT UPI.UserID,CONCAT(UPI.FName,IF((UPI.LName!='' AND UPI.LName IS NOT NULL),CONCAT(' ',UPI.LName),'') ) AS Name FROM userpersonalinfo UPI WHERE UPI.OrgID=pOrgID

AND UPI.UserID!=pLoggedUserID;

ELSE

SELECT \* FROM userpersonalinfo where 0 ;

END CASE;

ELSEIF pGroupByFlag='supervisor' THEN

CASE UserRole

WHEN 'SUPERVISOR' THEN

SELECT DISTINCT UPI.UserID,CONCAT(UPI.FName,IF((UPI.LName!='' AND UPI.LName IS NOT NULL),CONCAT(' ',UPI.LName),'') ) AS Name

FROM userpersonalinfo UPI LEFT JOIN deptmanager DM ON UPI.UserID=DM.DeptManagerID

LEFT JOIN department D ON D.DeptID=DM.DeptID

WHERE

UPI.OrgID=pOrgID

AND DM.DeptID IN (SELECT \* FROM temp\_AllDept)

AND UPI.UserID!=pLoggedUserID;

WHEN 'ADMIN' THEN

SELECT DISTINCT UPI.UserID,CONCAT(UPI.FName,IF((UPI.LName!='' AND UPI.LName IS NOT NULL),CONCAT(' ',UPI.LName),'') ) AS Name FROM userpersonalinfo UPI

LEFT JOIN deptmanager DM on UPI.UserID=DM.DeptManagerID

LEFT JOIN department D ON D.DeptID=DM.DeptID

WHERE UPI.OrgID=pOrgID AND DM.DeptManagerID IS NOT NULL

AND UPI.UserID!=pLoggedUserID;

ELSE

SELECT \* FROM userpersonalinfo where 0 ;

END CASE;

END IF;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetEarlyCheckOutOfDept` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetEarlyCheckOutOfDept`(

IN `pStartDate` date,

IN `pEnddate` date,

IN `pDeptID` INT

)

BEGIN

Select

ECOA.EarlyCheckoutApprvID,ECOA.UserID,Date(ECOA.AppliedDate) as AppliedDate,ECOA.RequestStatus

from earlycheckoutapproval ECOA left Join userpersonalinfo UP on ECOA.UserID = UP.UserID

where UP.DeptID = pDeptID

and (Date(ECOA.AppliedDate) between Date(pStartDate) and Date(pEndDate));

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetEmpOfDept` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetEmpOfDept`(

in pDeptID text,

in pOrgID int(11),

in pDate date,

in pTypeFlg int(11)

)

BEGIN

DECLARE TotalEmp,TotalPresent INT;

/\*Get all hierarchy sub department ids\*/

CALL nex\_GetDeptTree(pDeptID,@AllHrcyDept);

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

if(pTypeFlg=1) then

SELECT U.UserID,U.OrgID,U.FName,U.LName,U.UserPicture,U.EmploymentType,U.Email,U.SiteLocation,U.Deactivated,U.CellNo,U.EmployeeID,U.DeptID,D.DepartmentName

FROM userpersonalinfo U

LEFT JOIN department D ON U.DeptID = D.DeptID

where U.DeptID IN (select \* from temp\_AllDept)

and U.OrgID = pOrgID and U.Deactivated = 'N';

elseif(pTypeFlg=2) then

SELECT U.UserID,U.OrgID,U.FName,U.LName,U.UserPicture,U.Email,U.SiteLocation,U.Deactivated,U.CellNo,U.EmployeeID,U.DeptID,D.DepartmentName,

if((select AttendanceID from attendance A where A.UserID = U.UserID and date(A.LogTime) = pDate limit 0,1) IS NULL,'ABS','DP') as AttnStatus,

if((select OdID from odapplied OD where OD.UserID = U.UserID and OD.OdDate = pDate and OD.RequestStatus = 'Approved' limit 0,1) IS NULL,'NO','YES') as OnOD

FROM userpersonalinfo U

LEFT JOIN department D ON U.DeptID = D.DeptID

where U.DeptID IN (select \* from temp\_AllDept)

and U.OrgID = pOrgID and U.Deactivated = 'N';

elseif(pTypeFlg=3) then

select count(U.UserID) INTO TotalEmp

FROM userpersonalinfo U

where U.DeptID IN (select \* from temp\_AllDept)

and U.OrgID = pOrgID and U.Deactivated = 'N';

create TEMPORARY TABLE temp\_AllDept2 like temp\_AllDept;

insert temp\_AllDept2 select \* from temp\_AllDept;

select count(PresentUsers) INTO TotalPresent from (

select A.UserID as PresentUsers

from attendance A

LEFT JOIN userpersonalinfo U ON A.UserID = U.UserID

where U.DeptID IN (select \* from temp\_AllDept) and U.Deactivated = 'N' and date(A.LogTime) = pDate

UNION

select OD.UserID as PresentUsers

from odapplied OD

LEFT JOIN userpersonalinfo U ON U.UserID = OD.UserID

where U.DeptID IN (select \* from temp\_AllDept2) and U.Deactivated = 'N' and OD.RequestStatus = 'Approved' and OD.OdDate = date(CONVERT\_TZ(pDate,'+00:00','+05:30'))

) as temp;

DROP TEMPORARY TABLE temp\_AllDept2;

select TotalEmp,TotalPresent;

end if;

DROP TEMPORARY TABLE temp\_AllDept;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetLastCheckOutDetails` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetLastCheckOutDetails`(

in pUserID int,

in pOnDate datetime

)

BEGIN

select CheckOuttime,Location,Purpose from checkoutdetailsmore CODM

where CODM.UserID = pUserID and CODM.CheckOuttime <= pOnDate and CODM.TypeFlg = 'F'

order by CODM.CheckOuttime desc limit 0,1;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetLeaveApplyValidate` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetLeaveApplyValidate`(

IN pUserID int(11),

IN pLeaveTypeID int(11),

IN pStartDate date,

IN pEndDate date,

IN pIsHalfDay varchar(1),

IN pWhichHalf varchar(1),

IN pEffcDate date,

IN pSandwichLeaveFor varchar(20),

OUT HasLeaveBal char(5),

OUT NoOfDays float(3,1),

OUT IsLeaveOverlapped char(5),

OUT IsStartEndDayHld char(5),

OUT IsLeavesOnOD char(5)

)

BEGIN

declare UserDept int;

declare UserCreditLeave,UserDebitLeave,UserBalanceLeave,

UserAppliedLeaveFor,UserAppliedLeaveForHld,UserExactLeaveCount float;

declare i,NoOFTmpRecords INT DEFAULT 5;

declare tmpStartDate,tmpEndDate DATE;

declare UserDebitLEaveHld,TotalUserDebitLEaveHld float;

declare pStartDateIsHld,pEndDateIsHld float;

set UserCreditLeave = 0.0;

set UserDebitLEave = 0.0;

set UserAppliedLeaveForHld = 0.0;

set UserBalanceLeave = 0.0;

set UserExactLeaveCount = 0.0;

/\*Get User Department\*/

select U.DeptId into UserDept from userpersonalinfo U where U.UserID = pUserID;

/\*Get User's Credit Leaves Count\*/

select SUM(ULC.CreditLeave) into UserCreditLeave

from userleavecredit ULC

Where ULC.UserID = pUserID AND date(ULC.CreditDate) <= pEffcDate AND ULC.LeaveTypeID = pLeaveTypeID and ULC.IsCountable='Y' group by ULC.LeaveTypeID;

/\*Get User's Debit Leaves Count\*/

select SUM(LA.NoOfDays) into UserDebitLEave

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LA.LeaveTypeID = pLeaveTypeID AND LS.LeaveMasterStatusID IN (1,2) AND LA.IsCountable='Y'

group by LA.LeaveTypeID;

/\*Get User's Debit Leaves Count which overlaps with Public Holidays and Weekends\*/

/\*CREATE TEMPORARY TABLE UserDebitLeaveHolidayTmpTable (

TmpID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

StartDate date NOT NULL,

EndDate date NOT NULL

);

INSERT INTO UserDebitLeaveHolidayTmpTable (StartDate,EndDate) select LA.StartDate,LA.EndDate

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LA.LeaveTypeID = pLeaveTypeID AND LS.LeaveMasterStatusID IN (1,2) AND LA.IsCountable='Y';

select count(TmpID) INTO NoOFTmpRecords from UserDebitLeaveHolidayTmpTable;

set i = 1;

set TotalUserDebitLEaveHld = 0.0;

WHILE i <= NoOFTmpRecords DO

set UserDebitLEaveHld = 0.0;

select StartDate,EndDate INTO tmpStartDate,tmpEndDate from UserDebitLeaveHolidayTmpTable

where TmpID = i;

call nex\_GetLeaveInHolidayOrWeekend(pUserID,tmpStartDate,tmpEndDate,pSandwichLeaveFor,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into UserDebitLEaveHld;

set TotalUserDebitLEaveHld = TotalUserDebitLEaveHld + UserDebitLEaveHld;

SET i = i + 1;

END WHILE;

drop TABLE UserDebitLeaveHolidayTmpTable;

\*/

/\*Get User's Balance Leaves Count\*/

/\*set UserBalanceLeave = UserCreditLeave - (UserDebitLEave - TotalUserDebitLEaveHld);\*/

set UserBalanceLeave = UserCreditLeave - UserDebitLEave;

/\*Get User's current Applied Leaves Count\*/

select IF(pIsHalfDay = 'Y',0.5,(DATEDIFF(pEndDate,pStartDate)+1)) into UserAppliedLeaveFor;

/\*Get User's current Applied Leaves count which overlaps with Public Holidays and Weekends\*/

call nex\_GetLeaveInHolidayOrWeekend(pUserID,pStartDate,pEndDate,pSandwichLeaveFor,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into UserAppliedLeaveForHld;

/\*extra check written on 29th Jun by Shams\*/

IF(UserAppliedLeaveForHld IS NULL) THEN

SET UserAppliedLeaveForHld = 0;

END IF;

/\*Final count of User's current Applied Leaves\*/

set UserExactLeaveCount = (UserAppliedLeaveFor - UserAppliedLeaveForHld);

SELECT UserExactLeaveCount INTO NoOfDays;

/\*Is there leave balance\*/

select IF(UserBalanceLeave >= UserExactLeaveCount, 'true', 'false') into HasLeaveBal;

/\*Is the applied leave overlapping with another leave\*/

if (pIsHalfDay = 'Y') then

select IF(count(LA.LeavAppID) > 0,'true','false') as LeaveOverLapped into IsLeaveOverlapped from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (1,2)

AND LA.UserID = pUserID AND

(

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

)

AND LA.WhichHalf = pWhichHalf;

else

select IF(count(LA.LeavAppID) > 0,'true','false') as LeaveOverLapped into IsLeaveOverlapped from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (1,2)

AND LA.UserID = pUserID AND

(

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

);

end if;

/\*Is StartDate and EndDate is a Holiday or Weekends\*/

set pStartDateIsHld = 0;

set pEndDateIsHld = 0;

call nex\_GetLeaveInHolidayOrWeekend(pUserID,pStartDate,pStartDate,'',@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into pStartDateIsHld;

call nex\_GetLeaveInHolidayOrWeekend(pUserID,pEndDate,pEndDate,'',@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into pEndDateIsHld;

if (pStartDateIsHld > 0 or pEndDateIsHld > 0) then

select 'true' into IsStartEndDayHld;

else

select 'false' into IsStartEndDayHld;

end if;

/\*Is applying leaves in OD\*/

call nex\_GetLeaveInOD(pUserID,pStartDate,pEndDate,@IsLeaveOnOD);

select @IsLeaveOnOD into IsLeavesOnOD;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetLeaveApplyValidateShams` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetLeaveApplyValidateShams`(

IN pUserID int(11),

IN pLeaveTypeID int(11),

IN pStartDate date,

IN pEndDate date,

IN pIsHalfDay varchar(1),

IN pWhichHalf varchar(1),

IN pEffcDate date,

IN pSandwichLeaveFor varchar(20)

)

BEGIN

declare UserCreditLeave,UserBalanceLeave,

UserAppliedLeaveFor,UserAppliedLeaveForHld,UserExactLeaveCount,NoOfDays float(3,1) default 0.0;

declare UserDebitLeave float(3,1) default 0;

set UserCreditLeave = 0.0;

set UserAppliedLeaveForHld = 0.0;

set UserBalanceLeave = 0.0;

set UserExactLeaveCount = 0.0;

/\*Get User's Credit Leaves Count\*/

select SUM(ULC.CreditLeave) into UserCreditLeave

from userleavecredit ULC

Where ULC.UserID = pUserID AND date(ULC.CreditDate) <= pEffcDate AND ULC.LeaveTypeID = pLeaveTypeID and ULC.IsCountable='Y' group by ULC.LeaveTypeID;

/\*Get User's Debit Leaves Count\*/

select SUM(LA.NoOfDays) into UserDebitLeave

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LA.LeaveTypeID = pLeaveTypeID AND LS.LeaveMasterStatusID IN (1,2) AND LA.IsCountable='Y'

group by LA.LeaveTypeID;

set UserBalanceLeave = (UserCreditLeave - UserDebitLeave);

/\*Get User's current Applied Leaves Count\*/

select IF(pIsHalfDay = 'Y',0.5,(DATEDIFF(pEndDate,pStartDate)+1)) into UserAppliedLeaveFor;

/\*Get User's current Applied Leaves count which overlaps with Public Holidays and Weekends\*/

call nex\_GetLeaveInHolidayOrWeekend(pUserID,pStartDate,pEndDate,pSandwichLeaveFor,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into UserAppliedLeaveForHld;

/\*extra check written on 29th Jun by Shams\*/

IF(UserAppliedLeaveForHld IS NULL) THEN

SET UserAppliedLeaveForHld = 0;

END IF;

/\*Final count of User's current Applied Leaves\*/

set UserExactLeaveCount = (UserAppliedLeaveFor - UserAppliedLeaveForHld);

SELECT UserExactLeaveCount INTO NoOfDays;

/\*Is there leave balance\*/

SELECT UserCreditLeave,UserDebitLeave,UserBalanceLeave ;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetLeaveApplyValidate\_BKP` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetLeaveApplyValidate\_BKP`(

IN pUserID int(11),

IN pLeaveTypeID int(11),

IN pStartDate date,

IN pEndDate date,

IN pIsHalfDay varchar(1),

IN pWhichHalf varchar(1),

IN pEffcDate date,

OUT HasLeaveBal char(5),

OUT IsLeaveOverlapped char(5),

OUT IsStartEndDayHld char(5),

OUT IsLeavesOnOD char(5)

)

BEGIN

declare UserDept int;

declare UserCreditLeave,UserDebitLeave,UserBalanceLeave,

UserAppliedLeaveFor,UserAppliedLeaveForHld float;

declare i,NoOFTmpRecords INT DEFAULT 5;

declare tmpStartDate,tmpEndDate DATE;

declare UserDebitLEaveHld,TotalUserDebitLEaveHld float;

declare pStartDateIsHld,pEndDateIsHld float;

set UserCreditLeave = 0.0;

set UserDebitLEave = 0.0;

set UserAppliedLeaveForHld = 0.0;

/\*Get User Department\*/

select U.DeptId into UserDept from userpersonalinfo U where U.UserID = pUserID;

/\*Get User's Credit Leaves Count\*/

select SUM(ULC.CreditLeave) into UserCreditLeave

from userleavecredit ULC

Where ULC.UserID = pUserID AND date(ULC.CreditDate) <= pEffcDate AND ULC.LeaveTypeID = pLeaveTypeID and ULC.IsCountable='Y' group by ULC.LeaveTypeID;

/\*Get User's Debit Leaves Count\*/

select SUM(IF(LA.IsHalfDay = 'Y',0.5,(DATEDIFF(LA.EndDate,LA.StartDate)+1))) into UserDebitLEave

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LA.LeaveTypeID = pLeaveTypeID AND LS.LeaveMasterStatusID IN (1,2) AND LA.IsCountable='Y'

group by LA.LeaveTypeID;

/\*Get User's Debit Leaves Count which overlaps with Public Holidays and Weekends\*/

CREATE TEMPORARY TABLE UserDebitLeaveHolidayTmpTable (

TmpID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

StartDate date NOT NULL,

EndDate date NOT NULL

);

INSERT INTO UserDebitLeaveHolidayTmpTable (StartDate,EndDate) select LA.StartDate,LA.EndDate

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LA.LeaveTypeID = pLeaveTypeID AND LS.LeaveMasterStatusID IN (1,2) AND LA.IsCountable='Y';

select count(TmpID) INTO NoOFTmpRecords from UserDebitLeaveHolidayTmpTable;

set i = 1;

set TotalUserDebitLEaveHld = 0.0;

WHILE i <= NoOFTmpRecords DO

set UserDebitLEaveHld = 0.0;

select StartDate,EndDate INTO tmpStartDate,tmpEndDate from UserDebitLeaveHolidayTmpTable

where TmpID = i;

/\*Deprecated\*/

/\*call nex\_GetLeaveInHolidayOrWeekend(UserDept,tmpStartDate,tmpEndDate,@DaysInHolidayOrWeekend);\*/

/\*New - Using Calendar\*/

call nex\_GetLeaveInHolidayOrWeekend(pUserID,tmpStartDate,tmpEndDate,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into UserDebitLEaveHld;

set TotalUserDebitLEaveHld = TotalUserDebitLEaveHld + UserDebitLEaveHld;

SET i = i + 1;

END WHILE;

drop TABLE UserDebitLeaveHolidayTmpTable;

/\*Get User's Balance Leaves Count\*/

set UserBalanceLeave = UserCreditLeave - (UserDebitLEave - TotalUserDebitLEaveHld);

/\*Get User's current Applied Leaves Count\*/

select IF(pIsHalfDay = 'Y',0.5,(DATEDIFF(pEndDate,pStartDate)+1)) into UserAppliedLeaveFor;

/\*Get User's current Applied Leaves count which overlaps with Public Holidays and Weekends\*/

call nex\_GetLeaveInHolidayOrWeekend(pUserID,pStartDate,pEndDate,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into UserAppliedLeaveForHld;

/\*Final count of User's current Applied Leaves\*/

set UserAppliedLeaveFor = UserAppliedLeaveFor - UserAppliedLeaveForHld;

/\*Is there leave balance\*/

select IF(UserBalanceLeave >= UserAppliedLeaveFor, 'true', 'false') into HasLeaveBal;

/\*Is the applied leave overlapping with another leave\*/

if (pIsHalfDay = 'Y') then

select IF(count(LA.LeavAppID) > 0,'true','false') as LeaveOverLapped into IsLeaveOverlapped from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (1,2)

AND LA.UserID = pUserID AND

(

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

)

AND LA.WhichHalf = pWhichHalf;

else

select IF(count(LA.LeavAppID) > 0,'true','false') as LeaveOverLapped into IsLeaveOverlapped from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (1,2)

AND LA.UserID = pUserID AND

(

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

);

end if;

/\*Is StartDate and EndDate is a Holiday or Weekends\*/

set pStartDateIsHld = 0;

set pEndDateIsHld = 0;

call nex\_GetLeaveInHolidayOrWeekend(pUserID,pStartDate,pStartDate,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into pStartDateIsHld;

call nex\_GetLeaveInHolidayOrWeekend(pUserID,pEndDate,pEndDate,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into pEndDateIsHld;

if (pStartDateIsHld > 0 or pEndDateIsHld > 0) then

select 'true' into IsStartEndDayHld;

else

select 'false' into IsStartEndDayHld;

end if;

/\*Is applying leaves in OD\*/

call nex\_GetLeaveInOD(pUserID,pStartDate,pEndDate,@IsLeaveOnOD);

select @IsLeaveOnOD into IsLeavesOnOD;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetLeaveInHolidayOrWeekend` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetLeaveInHolidayOrWeekend`(

IN pUserID int,

IN pStartDate date,

IN pEndDate date,

IN pSandwichLeaveFor varchar(20),

OUT DaysInHolidayOrWeekend float

)

BEGIN

declare UserAppliedLeaveForHld,UserAppliedLeaveForHldTmp float;

declare tmpStartDate, tmpEndDate date;

declare UserCurrentCalendar int;

SET UserAppliedLeaveForHld = 0.0;

SET tmpStartDate = pStartDate;

SET tmpEndDate = pEndDate;

WHILE tmpStartDate <= tmpEndDate DO

set UserAppliedLeaveForHldTmp = 0.0;

set UserCurrentCalendar = 0;

SELECT calendarid into UserCurrentCalendar FROM usercalendar

WHERE userid = pUserID AND effectedfrom <= tmpStartDate

ORDER BY effectedfrom DESC LIMIT 1;

if UserCurrentCalendar!='' then

select count(H.HolidayID) into UserAppliedLeaveForHldTmp

from calendarholiday CH left join holiday H ON CH.HolidayID = H.HolidayID

where H.StartDate = tmpStartDate and CH.CalendarID = UserCurrentCalendar;

IF UserAppliedLeaveForHldTmp > 0.0 AND pSandwichLeaveFor!='' AND FIND\_IN\_SET("PH", pSandwichLeaveFor) THEN

set UserAppliedLeaveForHldTmp = 0.0;

END IF;

if(UserAppliedLeaveForHldTmp = 0.0) then

select count(W.WeekendsID) into UserAppliedLeaveForHldTmp

from calendarweekends CW left join weekends W ON CW.WeekendID = W.WeekendsID

where CW.CalendarID = UserCurrentCalendar and DAYNAME(tmpStartDate) = W.DayName;

IF UserAppliedLeaveForHldTmp > 0.0 AND pSandwichLeaveFor!='' AND FIND\_IN\_SET("WO", pSandwichLeaveFor) THEN

set UserAppliedLeaveForHldTmp = 0.0;

END IF;

end if;

end if;

if(UserAppliedLeaveForHldTmp > 0.0) then

set UserAppliedLeaveForHld = UserAppliedLeaveForHld + 1;

end if;

SET tmpStartDate = DATE\_ADD(tmpStartDate, INTERVAL 1 DAY);

END WHILE;

select UserAppliedLeaveForHld into DaysInHolidayOrWeekend;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetLeaveInHolidayOrWeekend\_BKP` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetLeaveInHolidayOrWeekend\_BKP`(

IN pUserID int,

IN pStartDate date,

IN pEndDate date,

OUT DaysInHolidayOrWeekend float

)

BEGIN

declare UserAppliedLeaveForHld,UserAppliedLeaveForHldTmp float;

declare tmpStartDate, tmpEndDate date;

declare UserCurrentCalendar int;

SET UserAppliedLeaveForHld = 0.0;

SET tmpStartDate = pStartDate;

SET tmpEndDate = pEndDate;

WHILE tmpStartDate <= tmpEndDate DO

set UserAppliedLeaveForHldTmp = 0.0;

set UserCurrentCalendar = 0;

SELECT calendarid into UserCurrentCalendar FROM usercalendar

WHERE userid = pUserID AND effectedfrom <= tmpStartDate

ORDER BY effectedfrom DESC LIMIT 1;

if UserCurrentCalendar!='' then

select count(H.HolidayID) into UserAppliedLeaveForHldTmp

from calendarholiday CH left join holiday H ON CH.HolidayID = H.HolidayID

where H.StartDate = tmpStartDate and CH.CalendarID = UserCurrentCalendar;

if(UserAppliedLeaveForHldTmp = 0.0) then

select count(W.WeekendsID) into UserAppliedLeaveForHldTmp

from calendarweekends CW left join weekends W ON CW.WeekendID = W.WeekendsID

where CW.CalendarID = UserCurrentCalendar and DAYNAME(tmpStartDate) = W.DayName;

end if;

end if;

if(UserAppliedLeaveForHldTmp > 0.0) then

set UserAppliedLeaveForHld = UserAppliedLeaveForHld + 1;

end if;

SET tmpStartDate = DATE\_ADD(tmpStartDate, INTERVAL 1 DAY);

END WHILE;

select UserAppliedLeaveForHld into DaysInHolidayOrWeekend;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetLeaveInOD` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetLeaveInOD`(

IN pUserID int,

IN pStartDate date,

IN pEndDate date,

OUT IsLeaveOnOD char(5)

)

BEGIN

declare UserAppliedLeaveOnOD int;

declare tmpStartDate, tmpEndDate date;

set UserAppliedLeaveOnOD = 0;

set tmpStartDate = pStartDate;

set tmpEndDate = pEndDate;

whileloop:WHILE tmpStartDate <= tmpEndDate DO

set UserAppliedLeaveOnOD = 0;

select count(OD.OdID) into UserAppliedLeaveOnOD from odapplied OD

where OD.UserID = pUserID and OD.RequestStatus IN ('Applied','Approved') and OD.OdDate = tmpStartDate;

if(UserAppliedLeaveOnOD > 0) then

leave whileloop;

end if;

SET tmpStartDate = DATE\_ADD(tmpStartDate, INTERVAL 1 DAY);

END WHILE;

select IF(UserAppliedLeaveOnOD > 0,'true','false') into IsLeaveOnOD;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetLeaveReport` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`%` PROCEDURE `nex\_GetLeaveReport`(

in pUserID int(11),

in pLeaveTypeID int(11),

in pStartDate date,

in pEndDate date,

in pTypeFlg int(11)

)

BEGIN

DECLARE DEBTILEAVE,NONDEBTILEAVE float;

DECLARE i,NoOFTmpRecords INT DEFAULT 5;

DECLARE tmpStartDate,tmpEndDate DATE;

DECLARE tmpPolicyType,SandwichLeaveFor varchar(20);

DECLARE UserDebitLeaveHld float;

if pTypeFlg= 1 Then

select sum(CreditLeave) as LeaveCredited from userleavecredit

where date(CreditDate) >= pStartDate and date(CreditDate) <= pEndDate

and UserID = pUserID and LeaveTypeID = pLeaveTypeID and IsCountable='Y';

elseif pTypeFlg = 2 Then

/\*Step 1\*/

select

SUM(IF(LA.IsHalfDay = 'Y',0.5,(DATEDIFF(if(LA.EndDate<=pEndDate,LA.EndDate,pEndDate),if(LA.StartDate>=pStartDate,LA.StartDate,pStartDate))+1))) into DEBTILEAVE

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LA.LeaveTypeID = pLeaveTypeID

AND LS.LeaveMasterStatusID IN (2)

AND (

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

) AND IsCountable='Y'

group by LA.LeaveTypeID;

/\*Step 2\*/

CREATE TEMPORARY TABLE DebitLeaveHolidayReportTmpTable (

TmpID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

PolicyType varchar(20) NOT NULL default 'R',

StartDate date NOT NULL,

EndDate date NOT NULL,

LeaveTypeID int NOT NULL,

LeavesInHoliday float NULL

);

INSERT INTO DebitLeaveHolidayReportTmpTable (PolicyType,StartDate,EndDate,LeaveTypeID)

select PolicyType,if(LA.StartDate>=pStartDate,LA.StartDate,pStartDate),if(LA.EndDate<=pEndDate,LA.EndDate,pEndDate),LA.LeaveTypeID

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LA.LeaveTypeID = pLeaveTypeID

AND LS.LeaveMasterStatusID IN (2)

AND (

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

) AND IsCountable='Y';

select count(TmpID) INTO NoOFTmpRecords from DebitLeaveHolidayReportTmpTable;

set i = 1;

WHILE i <= NoOFTmpRecords DO

set UserDebitLeaveHld = 0.0;

select PolicyType,StartDate,EndDate INTO tmpPolicyType,tmpStartDate,tmpEndDate from DebitLeaveHolidayReportTmpTable

where TmpID = i;

CASE tmpPolicyType

WHEN 'S-PH,WO' THEN

SET SandwichLeaveFor = 'PH,WO';

WHEN 'S-WO,PH' THEN

SET SandwichLeaveFor = 'PH,WO';

WHEN 'S-PH' THEN

SET SandwichLeaveFor = 'PH';

WHEN 'S-WO' THEN

SET SandwichLeaveFor = 'WO';

ELSE SET SandwichLeaveFor = '';

END CASE;

call nex\_GetLeaveInHolidayOrWeekend(pUserID,tmpStartDate,tmpEndDate,SandwichLeaveFor,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into UserDebitLeaveHld;

update DebitLeaveHolidayReportTmpTable set LeavesInHoliday = UserDebitLEaveHld where TmpID = i;

SET i = i + 1;

END WHILE;

select sum(LeavesInHoliday) INTO NONDEBTILEAVE

from DebitLeaveHolidayReportTmpTable group by LeaveTypeID order by LeaveTypeID;

drop TABLE DebitLeaveHolidayReportTmpTable;

/\*Step 3\*/

select (DEBTILEAVE - NONDEBTILEAVE) as LeaveDebited;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetLeaveReport\_BKP` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`%` PROCEDURE `nex\_GetLeaveReport\_BKP`(

in pUserID int(11),

in pLeaveTypeID int(11),

in pStartDate date,

in pEndDate date,

in pTypeFlg int(11)

)

BEGIN

DECLARE DEBTILEAVE,NONDEBTILEAVE float;

DECLARE i,NoOFTmpRecords INT DEFAULT 5;

DECLARE tmpStartDate,tmpEndDate DATE;

DECLARE UserDebitLeaveHld float;

if pTypeFlg= 1 Then

select sum(CreditLeave) as LeaveCredited from userleavecredit

where date(CreditDate) >= pStartDate and date(CreditDate) <= pEndDate

and UserID = pUserID and LeaveTypeID = pLeaveTypeID and IsCountable='Y';

elseif pTypeFlg = 2 Then

/\*Step 1\*/

select

SUM(IF(LA.IsHalfDay = 'Y',0.5,(DATEDIFF(if(LA.EndDate<=pEndDate,LA.EndDate,pEndDate),if(LA.StartDate>=pStartDate,LA.StartDate,pStartDate))+1))) into DEBTILEAVE

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LA.LeaveTypeID = pLeaveTypeID

AND LS.LeaveMasterStatusID IN (2)

AND (

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

) AND IsCountable='Y'

group by LA.LeaveTypeID;

/\*Step 2\*/

CREATE TEMPORARY TABLE DebitLeaveHolidayReportTmpTable (

TmpID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

StartDate date NOT NULL,

EndDate date NOT NULL,

LeaveTypeID int NOT NULL,

LeavesInHoliday float NULL

);

INSERT INTO DebitLeaveHolidayReportTmpTable (StartDate,EndDate,LeaveTypeID)

select if(LA.StartDate>=pStartDate,LA.StartDate,pStartDate),if(LA.EndDate<=pEndDate,LA.EndDate,pEndDate),LA.LeaveTypeID

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LA.LeaveTypeID = pLeaveTypeID

AND LS.LeaveMasterStatusID IN (2)

AND (

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

) AND IsCountable='Y';

select count(TmpID) INTO NoOFTmpRecords from DebitLeaveHolidayReportTmpTable;

set i = 1;

WHILE i <= NoOFTmpRecords DO

set UserDebitLeaveHld = 0.0;

select StartDate,EndDate INTO tmpStartDate,tmpEndDate from DebitLeaveHolidayReportTmpTable

where TmpID = i;

call nex\_GetLeaveInHolidayOrWeekend(pUserID,tmpStartDate,tmpEndDate,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into UserDebitLeaveHld;

update DebitLeaveHolidayReportTmpTable set LeavesInHoliday = UserDebitLEaveHld where TmpID = i;

SET i = i + 1;

END WHILE;

select sum(LeavesInHoliday) INTO NONDEBTILEAVE

from DebitLeaveHolidayReportTmpTable group by LeaveTypeID order by LeaveTypeID;

drop TABLE DebitLeaveHolidayReportTmpTable;

/\*Step 3\*/

select (DEBTILEAVE - NONDEBTILEAVE) as LeaveDebited;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetLeaveSummaryByType` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetLeaveSummaryByType`(

IN `pOrgID` INT(11),

IN `pLoggedUserID` INT(11) ,

IN `pYear` INT(5),

IN `pMonth` INT(5),

IN `pFilterBy` varchar(50),

IN `pGroupByFlag` varchar(50)

)

BEGIN

declare startMonthField,endMonthField,payrollStatus varchar(50);

Declare UserRole varchar(20);

Declare UserDept text;

/\* Get all sub depts of department of a manager\*/

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pLoggedUserID;

Select GROUP\_CONCAT(DeptID) into UserDept from deptmanager where DeptManagerID = pLoggedUserID;

CALL nex\_GetDeptTree(UserDept,@AllHrcyDept);

drop table if exists temp\_AllDept;

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

IF pGroupByFlag='dept' THEN

SET max\_sp\_recursion\_depth=2;

call nex\_GetLeaveSummaryByType(pOrgID,pLoggedUserID,pYear,pMonth,'','set\_range');

SET max\_sp\_recursion\_depth=0;

/\*select @startDate,@endDate;\*/

IF UserRole = 'SUPERVISOR' THEN

SELECT D.DeptID,D.DepartmentName ,sum( IF(LA.IsHalfDay = 'Y','0.5',(DATEDIFF(LA.EndDate,LA.StartDate)+1) )) as LeaveTaken

FROM department D LEFT JOIN userpersonalinfo UPI ON (UPI.DeptID=D.DeptID and D.DeptID IN (select \* from temp\_AllDept))

LEFT JOIN leaveapplied LA ON LA.UserID=UPI.UserID

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

GROUP BY D.DeptID ORDER BY LeaveTaken DESC;

ELSEIF UserRole = 'ADMIN' THEN

SELECT D.DeptID,D.DepartmentName ,sum( IF(LA.IsHalfDay = 'Y','0.5',(DATEDIFF(LA.EndDate,LA.StartDate)+1) )) as LeaveTaken

FROM department D LEFT JOIN userpersonalinfo UPI ON UPI.DeptID=D.DeptID

LEFT JOIN leaveapplied LA ON LA.UserID=UPI.UserID

/\*LEFT JOIN deptmanager DM ON DM.DeptID=D.DeptID\*/

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

GROUP BY D.DeptID HAVING LeaveTaken>0 ORDER BY LeaveTaken DESC;

ELSE

SELECT \* FROM userpersonalinfo where 0 ;

END IF;

ELSEIF pGroupByFlag='loc' THEN

SET max\_sp\_recursion\_depth=2;

CALL nex\_GetLeaveSummaryByType(pOrgID,pLoggedUserID,pYear,pMonth,'','set\_range');

SET max\_sp\_recursion\_depth=0;

IF UserRole = 'SUPERVISOR' THEN

SELECT UPI.SiteLocation ,sum( IF(LA.IsHalfDay = 'Y','0.5',(DATEDIFF(LA.EndDate,LA.StartDate)+1) )) as LeaveTaken

FROM userpersonalinfo UPI LEFT JOIN leaveapplied LA ON UPI.UserID=LA.UserID

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

AND UPI.DeptID IN (select \* from temp\_AllDept)

GROUP BY UPI.SiteLocation ORDER BY LeaveTaken DESC;

ELSEIF UserRole = 'ADMIN' THEN

SELECT UPI.SiteLocation ,sum( IF(LA.IsHalfDay = 'Y','0.5',(DATEDIFF(LA.EndDate,LA.StartDate)+1) )) as LeaveTaken

FROM userpersonalinfo UPI LEFT JOIN leaveapplied LA ON UPI.UserID=LA.UserID

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

GROUP BY UPI.SiteLocation ORDER BY LeaveTaken DESC;

ELSE

SELECT \* FROM userpersonalinfo where 0 ;

END IF;

ELSEIF pGroupByFlag='employee' THEN

SET max\_sp\_recursion\_depth=2;

CALL nex\_GetLeaveSummaryByType(pOrgID,pLoggedUserID,pYear,pMonth,'','set\_range');

SET max\_sp\_recursion\_depth=0;

IF UserRole = 'SUPERVISOR' THEN

CASE pFilterBy

WHEN 'freq' THEN

SELECT UPI.UserID,CONCAT(FName,IF((LName!='' AND LName IS NOT NULL),CONCAT(' ',LName),'') ) AS Name, COUNT(LA.LeavAppID) AS Frequency

FROM userpersonalinfo UPI

LEFT JOIN leaveapplied LA ON UPI.UserID=LA.UserID

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

AND UPI.DeptID IN (SELECT \* from temp\_AllDept)

AND UPI.UserID!=pLoggedUserID

GROUP BY UPI.UserID HAVING Frequency>0 ORDER BY Frequency DESC,Name ASC;

WHEN 'noOfDays' THEN

SELECT UPI.UserID,CONCAT(FName,IF((LName!='' AND LName IS NOT NULL),CONCAT(' ',LName),'') ) AS Name,

sum( IF(LA.IsHalfDay = 'Y','0.5',(DATEDIFF(LA.EndDate,LA.StartDate)+1) )) as LeaveTaken

FROM department D LEFT JOIN userpersonalinfo UPI ON (UPI.DeptID=D.DeptID AND D.DeptID IN (SELECT \* FROM temp\_AllDept))

LEFT JOIN leaveapplied LA ON LA.UserID=UPI.UserID

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

AND UPI.UserID!=pLoggedUserID

GROUP BY UPI.UserID HAVING LeaveTaken>0 ORDER BY LeaveTaken DESC,Name ASC;

END CASE;

ELSEIF UserRole = 'ADMIN' THEN

CASE pFilterBy

WHEN 'freq' THEN

SELECT UPI.UserID,CONCAT(FName,IF((LName!='' AND LName IS NOT NULL),CONCAT(' ',LName),'') ) AS Name,

COUNT(LA.LeavAppID) AS Frequency

FROM userpersonalinfo UPI

LEFT JOIN leaveapplied LA ON UPI.UserID=LA.UserID

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

AND UPI.UserID!=pLoggedUserID

GROUP BY UPI.UserID HAVING Frequency>0 ORDER BY Frequency DESC,Name ASC;

WHEN 'noOfDays' THEN

SELECT UPI.UserID,CONCAT(FName,IF((LName!='' AND LName IS NOT NULL),CONCAT(' ',LName),'') ) AS Name,

sum( IF(LA.IsHalfDay = 'Y','0.5',(DATEDIFF(LA.EndDate,LA.StartDate)+1) )) AS LeaveTaken

FROM department D LEFT JOIN userpersonalinfo UPI ON UPI.DeptID=D.DeptID

LEFT JOIN leaveapplied LA ON LA.UserID=UPI.UserID

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

AND UPI.UserID!=pLoggedUserID

GROUP BY UPI.UserID HAVING LeaveTaken>0 ORDER BY LeaveTaken DESC,Name ASC;

END CASE;

ELSE

SELECT \* FROM userpersonalinfo where 0 ;

END IF;

ELSEIF pGroupByFlag='supervisor' THEN

SET max\_sp\_recursion\_depth=2;

CALL nex\_GetLeaveSummaryByType(pOrgID,pLoggedUserID,pYear,pMonth,'','set\_range');

SET max\_sp\_recursion\_depth=0;

IF UserRole = 'SUPERVISOR' THEN

CASE pFilterBy

WHEN 'freq' THEN

SELECT UPI.UserID,CONCAT(FName,IF((LName!='' AND LName IS NOT NULL),CONCAT(' ',LName),'') ) AS Name, COUNT(LA.LeavAppID) AS Frequency

FROM userpersonalinfo UPI

LEFT JOIN leaveapplied LA ON UPI.UserID=LA.UserID

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

LEFT JOIN deptmanager DM on UPI.UserID=DM.DeptManagerID

LEFT JOIN department D ON D.DeptID=DM.DeptID

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

AND DM.DeptID IN (SELECT \* from temp\_AllDept)

AND UPI.UserID!=pLoggedUserID

GROUP BY UPI.UserID HAVING Frequency>0 ORDER BY Frequency DESC,Name ASC;

WHEN 'noOfDays' THEN

SELECT UPI.UserID,CONCAT(FName,IF((LName!='' AND LName IS NOT NULL),CONCAT(' ',LName),'') ) AS Name,

sum( IF(LA.IsHalfDay = 'Y','0.5',(DATEDIFF(LA.EndDate,LA.StartDate)+1) )) as LeaveTaken

FROM department D LEFT JOIN userpersonalinfo UPI ON (UPI.DeptID=D.DeptID AND D.DeptID IN (SELECT \* FROM temp\_AllDept))

LEFT JOIN leaveapplied LA ON LA.UserID=UPI.UserID

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

LEFT JOIN deptmanager DM on UPI.UserID=DM.DeptManagerID

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

AND DM.DeptManagerID IS NOT NULL

AND UPI.UserID!=pLoggedUserID

GROUP BY UPI.UserID HAVING LeaveTaken>0 ORDER BY LeaveTaken DESC,Name ASC;

END CASE;

ELSEIF UserRole = 'ADMIN' THEN

CASE pFilterBy

WHEN 'freq' THEN

SELECT UPI.OrgID,UPI.UserID,CONCAT(FName,IF((LName!='' AND LName IS NOT NULL),CONCAT(' ',LName),'') ) AS Name,

COUNT(LA.LeavAppID) AS Frequency

FROM userpersonalinfo UPI

LEFT JOIN leaveapplied LA ON UPI.UserID=LA.UserID

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

LEFT JOIN deptmanager DM on UPI.UserID=DM.DeptManagerID

LEFT JOIN department D ON D.DeptID=DM.DeptID

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

AND DM.DeptManagerID IS NOT NULL

AND UPI.UserID!=pLoggedUserID

GROUP BY UPI.UserID HAVING Frequency>0 ORDER BY Frequency DESC,Name ASC;

WHEN 'noOfDays' THEN

SELECT UPI.UserID,CONCAT(FName,IF((LName!='' AND LName IS NOT NULL),CONCAT(' ',LName),'') ) AS Name,

sum( IF(LA.IsHalfDay = 'Y','0.5',(DATEDIFF(LA.EndDate,LA.StartDate)+1) )) as LeaveTaken

FROM department D LEFT JOIN userpersonalinfo UPI ON UPI.DeptID=D.DeptID

LEFT JOIN leaveapplied LA ON LA.UserID=UPI.UserID

LEFT JOIN leavestatus LS ON LS.LeaveStatusID=LA.LeaveStatusID AND LS.LeaveMasterStatusID=2

LEFT JOIN deptmanager DM on UPI.UserID=DM.DeptManagerID

WHERE

UPI.OrgID=pOrgID

AND LA.StartDate>=@startDate AND LA.EndDate<=@endDate

AND DM.DeptManagerID IS NOT NULL

AND UPI.UserID!=pLoggedUserID

GROUP BY UPI.UserID HAVING LeaveTaken>0 ORDER BY LeaveTaken DESC,Name ASC;

END CASE;

ELSE

SELECT \* FROM userpersonalinfo where 0 ;

END IF;

ELSEIF pGroupByFlag='set\_range' THEN

SELECT IsActive into payrollStatus from policy\_payroll\_duration where OrgID=pOrgID AND ForYear = pYear;

CASE payrollStatus

WHEN 'Y' THEN

SET startMonthField= (concat("Month",pMonth,"Start"));

SET endMonthField = (concat("Month",pMonth,"End"));

SET @stmtStart = CONCAT('SET @startDate=','(SELECT ',startMonthField,' FROM policy\_payroll\_duration WHERE

OrgID=',pOrgID,' AND ForYear=',pYear,')');

PREPARE startDateStmt FROM @stmtStart;

EXECUTE startDateStmt;

SET @stmtEnd = CONCAT('SET @endDate=','(SELECT ',endMonthField,' FROM policy\_payroll\_duration WHERE

OrgID=',pOrgID,' AND ForYear=',pYear,')');

PREPARE endDateStmt FROM @stmtEnd;

EXECUTE endDateStmt;

IF(@endDate IS NULL || @endDate='') THEN

SET @startDate = CONCAT\_WS('-',pYear,IF(pMonth<10,CONCAT('0',pMonth),pMonth),'01');

SET @endDate = LAST\_DAY(@startDate);

END IF;

WHEN 'N' THEN

SET @startDate = CONCAT\_WS('-',pYear,IF(pMonth<10,CONCAT('0',pMonth),pMonth),'01');

SET @endDate = LAST\_DAY(@startDate);

ELSE

SET @startDate = CONCAT\_WS('-',pYear,IF(pMonth<10,CONCAT('0',pMonth),pMonth),'01');

SET @endDate = LAST\_DAY(@startDate);

END CASE;

ELSEIF pGroupByFlag='get\_range' THEN

SET max\_sp\_recursion\_depth=2;

CALL nex\_GetLeaveSummaryByType(pOrgID,pLoggedUserID,pYear,pMonth,'','set\_range');

SET max\_sp\_recursion\_depth=0;

select @startDate as 'Start',@endDate as 'End';

END IF;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetLisencedModulesOfOrg` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetLisencedModulesOfOrg`(

in pOrgID INT

)

BEGIN

SELECT NL.ModuleID,NM.ModuleName,NL.AddedOn,NL.ModuleOrder,NL.IsActive

from nexlicense NL LEFT JOIN nexmodule NM ON NL.ModuleID = NM.ModuleID

where NL.OrgID = pOrgID and NL.IsActive='Y'

order by NL.ModuleOrder asc;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_getnexorganization` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_getnexorganization`()

BEGIN

select \* from nexorganization;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetOdApplyValidate` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetOdApplyValidate`(

in pUserID int(11),

in pOdStartTime datetime,

in pOdEndTime datetime,

out IsAlrdyApplied char(5),

out IscurrentlyAtOffice char(5),

out IscurrentlyAtOffice2 char(5),

out IsAttendance char(5),

out IsOnLeave char(5)

)

BEGIN

select if(count(OD.OdID) > 0,'true','false') into IsAlrdyApplied from odapplied OD

where OD.UserID = pUserID and

(

(OD.OdStartTime >= pOdStartTime and OD.OdEndTime <= pOdEndTime)

OR

(OD.OdEndTime >= pOdStartTime and OD.OdStartTime <= pOdEndTime)

) and OD.RequestStatus IN ('Applied','Approved');

select if(count(A.AttendanceID) > 0,'true','false') into IscurrentlyAtOffice from attendance A

where date(A.LogTime) = date(pOdStartTime)

and A.UserID = pUserID

and A.CheckInTime IS NOT NULL and A.CheckOutTime IS NULL and A.AutoCheckOut IS NULL

and pOdStartTime > CONVERT\_TZ(A.CheckInTime,'+00:00','+05:30');

select if(count(A.AttendanceID) > 0,'true','false') into IscurrentlyAtOffice2 from attendance A

where date(A.LogTime) = date(pOdStartTime)

and A.UserID = pUserID

and A.CheckInTime IS NOT NULL and A.CheckOutTime IS NULL and A.AutoCheckOut IS NULL

and pOdStartTime <= CONVERT\_TZ(A.CheckInTime,'+00:00','+05:30')

and pOdEndTime >= CONVERT\_TZ(A.CheckInTime,'+00:00','+05:30');

select if(count(A.AttendanceID) > 0,'true','false') into IsAttendance from attendance A

where date(A.LogTime) = date(pOdStartTime)

and A.UserID = pUserID

and A.CheckInTime IS NOT NULL and A.CheckOutTime IS NOT NULL

and A.AutoCheckOut IS NULL

and

(

(CONVERT\_TZ(A.CheckInTime,'+00:00','+05:30') >= pOdStartTime and CONVERT\_TZ(A.CheckOutTime,'+00:00','+05:30') <= pOdEndTime)

OR

(CONVERT\_TZ(A.CheckOutTime,'+00:00','+05:30') >= pOdStartTime and CONVERT\_TZ(A.CheckInTime,'+00:00','+05:30') <= pOdEndTime)

)

;

select if(count(LA.LeavAppID) > 0,'true','false') into IsOnLeave from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID and LS.LeaveMasterStatusID IN (1,2) and

date(pOdStartTime) BETWEEN LA.StartDate AND LA.EndDate;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetOTPCodeForEmp` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetOTPCodeForEmp`(

in pUserID int

)

BEGIN

Select OTPCode from checkinotpdetails where UserID = pUserID and

(OTPTime > DATE\_SUB(Now(), INTERVAL 30 MINUTE)) and OTPverified = 'N'

ORDER BY OTPTime DESC

LIMIT 1;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetQRCheckOutCodeForEmp` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetQRCheckOutCodeForEmp`(

in pUserID int

)

BEGIN

Select QRCode from checkoutdetailsmore where UserID = pUserID and

(Checkouttime > DATE\_SUB(Now(), INTERVAL 30 MINUTE)) ;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetQRCodeForEmp` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetQRCodeForEmp`(

in pUserID int

)

BEGIN

Select QRCode from checkindetailsmore where UserID = pUserID and

(CheckInTime > DATE\_SUB(Now(), INTERVAL 30 MINUTE)) ;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetSupervisorNotifications` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetSupervisorNotifications`(

in pUserID int

)

BEGIN

Select \* from userpersonalinfo UP

inner join department D on Up.DeptID = D.DeptID

inner join deptmanager DM on D.DeptID = DM.DeptID

Where DM.DeptManagerID = 50 and UP.UserID <> pUserID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetSuperVisorsForOrg` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetSuperVisorsForOrg`(

in pOrgID int

)

BEGIN

Select DISTINCT UP.\*, DP.DepartmentName as DeptName,

(Select CONCAT(Fname, ' ', LName, ' (', DP.DepartmentName, ')') from userpersonalinfo where DeptID = UP.DeptID and UserID = DPM.DeptManagerID) as Supervisor,

IFNULL((Select IFNULL(UR.rolename, 'EMPLOYEE') from userrole UR inner join userworkinfo UWI on UR.roleID = UWI.roleID where UWI.UserID = UP.UserID), 'EMPLOYEE') as Role

from department DP inner join deptmanager DPM on DP.DeptID = DPM.DeptID

inner join userpersonalinfo UP on UP.UserID = DPM.DeptManagerID

inner join userrole UR on UR.OrgID = UP.OrgID

inner join userworkinfo UWI on UWI.UserID = UP.UserID

where DP.OrgID = pOrgID and UP.Deactivated = 'N' and UR.roleName in ('ADMIN', 'SUPERVISOR') group by UP.UserID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetUserHoliday` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetUserHoliday`(

IN pUserID int(11),

IN pTodayDate date

)

BEGIN

DECLARE USERCALENDAR int;

select UC.CalendarID INTO USERCALENDAR from usercalendar UC where

UC.EffectedFrom <= pTodayDate and UC.UserID = pUserID order by UC.EffectedFrom desc limit 1;

select H.HolidayID,H.HolidayName,H.HolidayDescription,H.StartDate,H.EndDate,H.OrgID

from calendarholiday CH

LEFT JOIN holiday H ON CH.HolidayID = H.HolidayID

where CH.CalendarID = USERCALENDAR

order by H.StartDate desc;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_GetVisitorOfOrg` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_GetVisitorOfOrg`(

in pOrgID int(10),

in pStartDate date,

in pEndDate date

)

BEGIN

SELECT VA.MeetingID,VA.VisitorType,VA.VisitorID,V.FullName,V.CompanyName,V.MobileNo,V.Email,V.Photo,V.SignUpDate,

VA.CheckInTime,VA.CheckInLocation,VA.CheckOutTime,VA.CheckOutLocation,

VA.MeetToName,VA.MeetToMobileNo,VA.MeetToUserID,VA.MeetReason,date(VA.LogTime) as LogTime,VA.VisitAttenStatus,VA.MeetingStatus,VA.StaticQrCardNo,

concat(U.FName,' ',U.LName) MeetToFullName,U.EmployeeID MeetToEmployeeID

from visitorattendance VA left join visitor V on VA.VisitorID = V.VisitorID

left join userpersonalinfo U on VA.MeetToUserID = U.UserID

where V.OrgID = pOrgID and (date(VA.LogTime) >= date(pStartDate) AND date(VA.LogTime) <= date(pEndDate))

order by VA.MeetingID desc,VA.VisitorAttendanceID asc;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Get\_MngrDept` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Get\_MngrDept`(

In pUserID int(11)

)

BEGIN

Select DeptID from deptmanager where DeptManagerID = pUserID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_KioskCreate` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_KioskCreate`(

in pKioskID int(11),

in pOrgID int(11),

in pUserID int(11),

in pEmail varchar (50),

in pKioskName varchar (50),

in pUniqueDeviceID varchar (50),

in pDeviceType varchar (50),

in pOTP varchar(10),

in pRegisteredOn datetime,

in pTypeFlg int(11)

)

BEGIN

if pTypeFlg= 1 Then

INSERT INTO kiosk

(

OrgID,

UserID,

Email,

KioskName,

UniqueDeviceID,

OTP,

DeviceType

)

VALUES

(

pOrgID,

pUserID,

pEmail,

pKioskName,

pUniqueDeviceID,

pOTP,

pDeviceType

);

elseif pTypeFlg= 2 Then

Update kiosk k Set

k.OTP= pOTP

Where k.KioskID= pKioskID;

elseif pTypeFlg= 3 Then

Select KioskID from kiosk where Email= pEmail and isActive = 'Y';

elseif pTypeFlg= 4 Then

select KioskID,Email from kiosk where Email=pEmail and OTP=pOTP and isActive = 'N';

elseif pTypeFlg= 5 Then

Update kiosk k Set

k.OTPSuccess= 'Y',

k.IsActive='Y' ,

k.RegisteredOn=pRegisteredOn

Where k.KioskID= pKioskID;

delete from kiosk where Email = pEmail and OTPSuccess='N' and IsActive='N';

elseif pTypeFlg= 6 Then

select \* from kiosk where OrgID=pOrgID and OTPSuccess='Y' order by KioskID desc;

elseif pTypeFlg= 7 Then

Update kiosk k Set

k.IsActive='N'

Where k.KioskID= pKioskID;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_KioskLogin` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_KioskLogin`(

in pUserEmail varchar (50),

in pPassword varchar (50),

in pUniqueDeviceID varchar (255)

)

BEGIN

Select UP.UserID,UP.OrgID,UP.FName,UP.LName,UP.Email,UP.UserPicture,UP.WorkPhone,UP.CellNo,UP.EmployeeID,UP.DeptID,

UP.PwdResetrequired,UP.Deactivated,

DP.DepartmentName as DeptName,

(Select GROUP\_CONCAT(DeptID) from deptmanager where DeptManagerID = UP.UserID) as MngrOfDept,

(Select CONCAT(Fname, ' ', LName) from userpersonalinfo where UserID = DPM.DeptManagerID) as Supervisor,

IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID where uwi.UserID = UP.UserID), 'EMPLOYEE') as Role,

IFNULL(UWI.Designation, '') as Designation

from userpersonalinfo UP

LEFT JOIN kiosk KOSK on UP.UserID = KOSK.UserID

LEFT JOIN department DP on UP.DeptID = DP.DeptID

LEFT JOIN deptmanager DPM on DP.DeptID = DPM.DeptID

LEFT JOIN userworkinfo UWI on UP.UserID = UWI.UserID

where UP.Email=pUserEmail and UP.Pwd like binary pPassword and UP.deleted=0 and KOSK.UniqueDeviceID = pUniqueDeviceID and KOSK.IsActive = 'Y'

limit 1;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_OrgByEmail` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_OrgByEmail`(

in pUserEmail varchar (50)

)

BEGIN

Select \* from userpersonalinfo where Email=pUserEmail;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_OrgDetailsByID` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_OrgDetailsByID`(

in pOrgID int (11)

)

BEGIN

Select OrgName,ContactEmail,CreatedDate,IsActive from nexorganization where OrgID=pOrgID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_ResetPassword` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_ResetPassword`(

in pUserEmail varchar (50),

in pPassword varchar (255),

in pCurrentPassword varchar (50),

in pPwdResetRequired varchar (1)

)

BEGIN

UPDATE userpersonalinfo SET Pwd = pPassword, PwdResetRequired = pPwdResetRequired WHERE Email = pUserEmail AND Pwd = pCurrentPassword ;

Select \* from userpersonalinfo where Email = pUserEmail;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_SaveSupervisiorForEmp` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_SaveSupervisiorForEmp`(

in pSupervisorID int,

in pUserID text,

in pOrgID int,

in pDeptID int

)

BEGIN

DECLARE i,NOOFU int;

if pDeptID = 0 then

Select @vDeptID := D.DeptID from department D inner join deptmanager DM on D.DeptID = DM.DeptID

Where DM.DeptManagerID = pSupervisorID and D.OrgID = pOrgID limit 0,1;

else

Select @vDeptID := pDeptID;

end if;

call splitter(pUserID,',');

select count(SplitValue) INTO NOOFU from splitResultsTmpTable;

set i = 1;

WHILE i <= NOOFU DO

Update userpersonalinfo set DeptID = @vDeptID where UserID = (select SplitValue from splitResultsTmpTable where TmpID = i);

SET i = i + 1;

END WHILE;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_ShowEarlyCheckoutOld` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_ShowEarlyCheckoutOld`(

in pSupUserID int(11),

in pOrgID int(11),

in pStartDate date,

in pEndDate date,

IN pFlag char(4)

)

BEGIN

if pFlag= 'ARR' Then

select ECOA.\*,CONCAT(UP.FName,' ',UP.LName) as UserName,UP.EmployeeID as UserEmployeeID,UP.UserPicture,

Time(A.CheckInTime) as CheckInTime,

Time(A.CheckOutTime) CheckOutTime,

(TIMESTAMPDIFF(Second, A.CheckInTime, A.CheckOutTime)) TotalWorkedHours

from earlycheckoutapproval ECOA

LEFT JOIN attendance A ON ECOA.AttendanceID = A.AttendanceID

LEFT JOIN userpersonalinfo UP ON ECOA.UserID = UP.UserID

where ECOA.SupervisorUserID = pSupUserID

AND ECOA.OrgID = pOrgID

AND (DATE\_FORMAT(ECOA.AppliedDate, "%Y-%m-%d") >= pStartDate AND DATE\_FORMAT(ECOA.AppliedDate, "%Y-%m-%d") <= pEndDate)

AND ECOA.RequestStatus IN('Approved','Rejected')

ORDER BY ECOA.AppliedDate DESC;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_Attendance` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_Attendance`(

in pUserID int(11),

in pType Char(1),

in pStartDate date,

in pEndDate date

)

BEGIN

if isNull(pStartDate) Then

Set pStartDate = Date\_Add(Now(), Interval -6 day);

End if;

if isNull(pEndDate) Then

Set pEndDate = date(Now());

End if;

if pType='S' Then

Select A.UserId as UserID,

/\*date(CheckInTime) as AttnDate,\*/

date(LogTime) as AttnDate,

Time(CheckInTime) as CheckInTime,

Time(CheckOutTime) CheckOutTime,

(TIMESTAMPDIFF(Second, CheckInTime, CheckOutTime)) TotalHours,

CheckInLocation, CheckOutLocation,AutoCheckOut

from Attendance ATT inner join userpersonalinfo UP on ATT.UserID = UP.UserID

inner join department D on UP.orgID = D.OrgID

Inner join deptmanager DM on D.DeptID = DM.DeptID

Where DM.DeptManagerID = pUserID

and ATT.UserId<> pUserID

/\*and CheckInTime>= pStartDate and CheckInTime< Date\_Add(pEndDate, interval 1 day)\*/

and date(A.LogTime) >= pStartDate and date(A.LogTime) <= pEndDate

Order by date(CheckInTime);

End If;

if pType='U' Then

Select A.UserId as UserID,

/\*date(CheckInTime) as AttnDate,\*/

date(LogTime) as AttnDate,

Time(CheckInTime) as CheckInTime,

Time(CheckOutTime) CheckOutTime,

(TIMESTAMPDIFF(Second, CheckInTime, CheckOutTime)) TotalHours,

CheckInLocation, CheckOutLocation,AutoCheckOut,

ECOA.WorkDuration ECWorkDuration,ECOA.LessDuration ECLessDuration,ECOA.RequestStatus ECRequestStatus

from attendance A LEFT JOIN earlycheckoutapproval ECOA ON A.AttendanceID=ECOA.AttendanceID

Where A.UserID = pUserID

/\*and CheckInTime>= pStartDate and CheckInTime< Date\_Add(pEndDate, interval 1 day)\*/

and date(A.LogTime) >= pStartDate and date(A.LogTime) <= pEndDate

Order by date(CheckInTime);

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_CompoffListofdept` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_CompoffListofdept`(

IN pUserID int,

in pDeptID text,

in pStartDate date,

in pEndDate date,

in pFlag char(4)

)

BEGIN

DECLARE DeptMngUserID INT;

DECLARE UserRole varchar(10);

DECLARE SelDeptID text;

SELECT group\_concat(DM.DeptManagerID) INTO DeptMngUserID from deptmanager DM where DM.DeptID IN (pDeptID);

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pUserID;

/\*Get all hierarchy sub department ids\*/

if UserRole = 'ADMIN' then

CALL nex\_GetDeptTree(pDeptID,@AllHrcyDept);

SELECT @AllHrcyDept INTO SelDeptID;

else

SELECT pDeptID INTO SelDeptID;

end if;

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select pDeptID as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

if pFlag = 'A' then

SELECT U.UserID,U.FName,U.LName,U.EmployeeID,U.UserPicture,D.DepartmentName,

CMOF.CompOffID,CMOF.UserID,CMOF.CompOffDate,CMOF.RequestType as OrigRequestType,if(CMOF.RequestType='Leave','CO Request','Off Day Present') as RequestType,CMOF.AppliedDate,CMOF.RequestStatus,

CMOF.StatusUpdateDate,CMOF.StatusUpdteBy

FROM compoffapplied CMOF

LEFT JOIN userpersonalinfo U ON CMOF.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

WHERE CMOF.RequestStatus != 'Applied'

AND CMOF.CompOffDate BETWEEN pStartDate AND pEndDate

AND U.DeptID IN (select \* from temp\_AllDept)

AND CMOF.UserID NOT IN (DeptMngUserID)

ORDER BY CMOF.CompOffDate DESC;

elseif pFlag = 'APPV' then

/\*Thuis is using for Report\*/

SELECT U.UserID,U.FName,U.LName,U.EmployeeID,U.UserPicture,D.DepartmentName,

CMOF.CompOffID,CMOF.UserID,CMOF.CompOffDate,CMOF.RequestType as OrigRequestType,if(CMOF.RequestType='Leave','CO Request','Off Day Present') as RequestType,CMOF.AppliedDate,CMOF.RequestStatus,

CMOF.StatusUpdateDate,CMOF.StatusUpdteBy

FROM compoffapplied CMOF

LEFT JOIN userpersonalinfo U ON CMOF.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

WHERE CMOF.RequestStatus = 'Approved'

AND (

CMOF.CompOffDate BETWEEN pStartDate AND pEndDate

OR

date(CMOF.StatusUpdateDate) BETWEEN pStartDate AND pEndDate

)

AND U.DeptID IN (select \* from temp\_AllDept)

AND CMOF.UserID NOT IN (DeptMngUserID)

ORDER BY CMOF.CompOffDate DESC;

else

SELECT U.UserID,U.FName,U.LName,U.EmployeeID,U.UserPicture,D.DepartmentName,

CMOF.CompOffID,CMOF.UserID,CMOF.CompOffDate,CMOF.RequestType as OrigRequestType,if(CMOF.RequestType='Leave','CO Request','Off-day Working') as RequestType,CMOF.AppliedDate,CMOF.RequestStatus,

CMOF.StatusUpdateDate,CMOF.StatusUpdteBy

FROM compoffapplied CMOF

LEFT JOIN userpersonalinfo U ON CMOF.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

WHERE CMOF.RequestStatus = 'Applied'

AND U.DeptID IN (select \* from temp\_AllDept)

AND CMOF.UserID NOT IN (DeptMngUserID)

ORDER BY CMOF.CompOffDate DESC;

end if;

DROP TEMPORARY TABLE temp\_AllDept;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_DeptAttendanceHrs` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_DeptAttendanceHrs`(

IN pDeptID text,

IN pStartDate date,

IN pEndDate date

)

BEGIN

/\*Get all hierarchy sub department ids\*/

CALL nex\_GetDeptTree(pDeptID,@AllHrcyDept);

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

SELECT A.UserID,U.FName,U.LName,U.EmployeeID,U.UserPicture,D.DepartmentName,

SUM(TIMESTAMPDIFF(MINUTE,A.CheckInTime,A.CheckOutTime)) WorkedTime,

count(A.AttendanceID) NoOfAttn

FROM attendance A

LEFT JOIN userpersonalinfo U ON A.UserID = U.UserID

LEFT JOIN department D ON D.DeptID = U.DeptID

WHERE U.DeptID IN (select \* from temp\_AllDept) AND DATE(A.LogTime) BETWEEN pStartDate AND pEndDate

GROUP BY A.UserID

ORDER BY U.FName;

DROP TEMPORARY TABLE temp\_AllDept;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_LeaveCountCalender` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_LeaveCountCalender`(

in pUserID int,

in pDeptID text,

in pStartDate date,

in pEndDate date

)

BEGIN

DECLARE DeptMngUserID,TeamSize INT;

DECLARE UserRole varchar(10);

DECLARE SelDeptID text;

SELECT group\_concat(DM.DeptManagerID) INTO DeptMngUserID from deptmanager DM where DM.DeptID IN (pDeptID);

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pUserID;

/\*Get all hierarchy sub department ids\*/

if UserRole = 'ADMIN' then

CALL nex\_GetDeptTree(pDeptID,@AllHrcyDept);

SELECT @AllHrcyDept INTO SelDeptID;

else

SELECT pDeptID INTO SelDeptID;

end if;

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select SelDeptID as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

CREATE TEMPORARY TABLE temp\_AllDept2

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select SelDeptID as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

CREATE TEMPORARY TABLE temp\_AllDept3

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select SelDeptID as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

SELECT count(U.UserID) INTO TeamSize from userpersonalinfo U where U.DeptID IN (select \* from temp\_AllDept);

select a.Date,

(

select count(LA.UserID) from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

where LS.LeaveMasterStatusID = 2 AND a.Date BETWEEN StartDate AND EndDate

AND U.DeptID IN (select \* from temp\_AllDept) AND LA.UserID NOT IN (DeptMngUserID)

) as NoOfEmpOnLev

, IF(

(

select count(LA.UserID) from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

where LS.LeaveMasterStatusID = 2 AND a.Date BETWEEN StartDate AND EndDate

AND U.DeptID IN (select \* from temp\_AllDept2) AND LA.UserID NOT IN (DeptMngUserID)

)>= (TeamSize\*0.10),'#fb0013',

IF(

(

select count(LA.UserID) from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

where LS.LeaveMasterStatusID = 2 AND a.Date BETWEEN StartDate AND EndDate

AND U.DeptID IN (select \* from temp\_AllDept3) AND LA.UserID NOT IN (DeptMngUserID)

)= ((TeamSize\*0.10)-1),'#f89e00',''

)

)

as ThresholdFlag,TeamSize

from

(select pEndDate - INTERVAL (a.a + (10 \* b.a) + (100 \* c.a)) DAY as Date from

(select 0 as a union all select 1 union all select 2 union all select 3 union all

select 4 union all select 5 union all select 6 union all select 7 union all

select 8 union all select 9) as a cross join (select 0 as a union all select 1 union all

select 2 union all select 3 union all select 4 union all select 5 union all select 6 union all

select 7 union all select 8 union all select 9) as b cross join

(select 0 as a union all select 1 union all select 2 union all

select 3 union all select 4 union all select 5 union all select 6 union all select 7

union all select 8 union all select 9) as c

) a

where a.Date between pStartDate and pEndDate;

DROP TEMPORARY TABLE temp\_AllDept;

DROP TEMPORARY TABLE temp\_AllDept2;

DROP TEMPORARY TABLE temp\_AllDept3;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_LeaveListofdept` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_LeaveListofdept`(

IN pUserID int,

IN pDeptID text,

IN pStartDate date,

IN pEndDate date,

IN pFlag char(4)

)

BEGIN

DECLARE DeptMngUserID INT;

DECLARE UserRole varchar(10);

DECLARE SelDeptID text;

SELECT group\_concat(DM.DeptManagerID) INTO DeptMngUserID from deptmanager DM where DM.DeptID IN (pDeptID);

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pUserID;

/\*Get all hierarchy sub department ids\*/

if UserRole = 'ADMIN' then

CALL nex\_GetDeptTree(pDeptID,@AllHrcyDept);

SELECT @AllHrcyDept INTO SelDeptID;

else

SELECT pDeptID INTO SelDeptID;

end if;

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select SelDeptID as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

If pFlag = 'DLL' Then

SELECT U.UserID,U.EmployeeID,U.DeptID,U.FName,U.LName,U.UserPicture,D.DepartmentName,

LA.LeavAppID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StatusUpdteBy,LA.StartDate,LA.EndDate,LA.AppliedDate,

LA.NoOfDays,LA.PolicyType,

LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,LS.LeaveStatusText,LT.LeaveTypeText

FROM leaveapplied LA

LEFT JOIN leaveappliedstatuslog LLOG ON LLOG.LeavAppID = LA.LeavAppID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

WHERE LA.LeaveStatusID = LLOG.LeaveStatusID

AND U.DeptID IN (select \* from temp\_AllDept)

AND (

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

)

AND LA.UserID NOT IN (DeptMngUserID)

AND LS.LeaveMasterStatusID != 1 ORDER BY LA.StartDate DESC;

elseif pFlag = 'DALL' Then

SELECT U.UserID,U.EmployeeID,U.DeptID,U.FName,U.LName,U.UserPicture,D.DepartmentName,

LA.LeavAppID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StatusUpdteBy,LA.StartDate,LA.EndDate,LA.AppliedDate,

LA.NoOfDays,LA.PolicyType,

LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,LS.LeaveStatusText,LT.LeaveTypeText,LMT.LeaveMasterType,LLOG.StatusNote

FROM leaveapplied LA

LEFT JOIN leaveappliedstatuslog LLOG ON LLOG.LeavAppID = LA.LeavAppID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavemastertype LMT ON LMT.LeaveMasterTypeID = LT.LeaveMasterTypeID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

WHERE LA.LeaveStatusID = LLOG.LeaveStatusID

AND U.DeptID IN (select \* from temp\_AllDept)

AND (

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

)

AND LA.UserID NOT IN (DeptMngUserID)

AND LS.LeaveMasterStatusID = 2 ORDER BY LA.StartDate DESC;

else

SELECT U.UserID,U.EmployeeID,U.DeptID,U.FName,U.LName,U.UserPicture,D.DepartmentName,

LA.LeavAppID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StatusUpdteBy,LA.StartDate,LA.EndDate,LA.AppliedDate,

LA.NoOfDays,LA.PolicyType,

LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,LS.LeaveStatusText,LT.LeaveTypeText,LLOG.StatusNote

FROM leaveapplied LA

LEFT JOIN leaveappliedstatuslog LLOG ON LLOG.LeavAppID = LA.LeavAppID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

WHERE LA.LeaveStatusID = LLOG.LeaveStatusID

AND U.DeptID IN (select \* from temp\_AllDept)

AND LA.UserID NOT IN (DeptMngUserID)

AND LS.LeaveMasterStatusID = 1 ORDER BY LA.StartDate DESC;

END if;

DROP TEMPORARY TABLE temp\_AllDept;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_LeaveListofdept\_BKP` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_LeaveListofdept\_BKP`(

IN pDeptID text,

IN pStartDate date,

IN pEndDate date,

IN pFlag char(4)

)

BEGIN

DECLARE DeptMngUserID INT;

SELECT group\_concat(DM.DeptManagerID) INTO DeptMngUserID from deptmanager DM where DM.DeptID IN (pDeptID);

/\*Get all hierarchy sub department ids\*/

CALL nex\_GetDeptTree(pDeptID,@AllHrcyDept);

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

If pFlag = 'DLL' Then

SELECT U.UserID,U.EmployeeID,U.DeptID,U.FName,U.LName,U.UserPicture,D.DepartmentName,

LA.LeavAppID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StatusUpdteBy,LA.StartDate,LA.EndDate,LA.AppliedDate,

IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,

LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,LS.LeaveStatusText,LT.LeaveTypeText

FROM leaveapplied LA

LEFT JOIN leaveappliedstatuslog LLOG ON LLOG.LeavAppID = LA.LeavAppID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

WHERE LA.LeaveStatusID = LLOG.LeaveStatusID

AND U.DeptID IN (select \* from temp\_AllDept)

AND (

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

)

AND LA.UserID NOT IN (DeptMngUserID)

AND LS.LeaveMasterStatusID != 1 ORDER BY LA.StartDate DESC;

elseif pFlag = 'DALL' Then

SELECT U.UserID,U.EmployeeID,U.DeptID,U.FName,U.LName,U.UserPicture,D.DepartmentName,

LA.LeavAppID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StatusUpdteBy,LA.StartDate,LA.EndDate,LA.AppliedDate,

IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,

LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,LS.LeaveStatusText,LT.LeaveTypeText,LMT.LeaveMasterType,LLOG.StatusNote

FROM leaveapplied LA

LEFT JOIN leaveappliedstatuslog LLOG ON LLOG.LeavAppID = LA.LeavAppID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavemastertype LMT ON LMT.LeaveMasterTypeID = LT.LeaveMasterTypeID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

WHERE LA.LeaveStatusID = LLOG.LeaveStatusID

AND U.DeptID IN (select \* from temp\_AllDept)

AND (

(LA.StartDate >= pStartDate and LA.EndDate <= pEndDate)

OR

(LA.EndDate >= pStartDate and LA.StartDate <= pEndDate)

)

AND LA.UserID NOT IN (DeptMngUserID)

AND LS.LeaveMasterStatusID = 2 ORDER BY LA.StartDate DESC;

else

SELECT U.UserID,U.EmployeeID,U.DeptID,U.FName,U.LName,U.UserPicture,D.DepartmentName,

LA.LeavAppID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StatusUpdteBy,LA.StartDate,LA.EndDate,LA.AppliedDate,

IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,

LA.IsHalfDay,LA.WhichHalf,LA.LeaveReason,LS.LeaveStatusText,LT.LeaveTypeText,LLOG.StatusNote

FROM leaveapplied LA

LEFT JOIN leaveappliedstatuslog LLOG ON LLOG.LeavAppID = LA.LeavAppID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

WHERE LA.LeaveStatusID = LLOG.LeaveStatusID

AND U.DeptID IN (select \* from temp\_AllDept)

AND LA.UserID NOT IN (DeptMngUserID)

AND LS.LeaveMasterStatusID = 1 ORDER BY LA.StartDate DESC;

END if;

DROP TEMPORARY TABLE temp\_AllDept;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_Leavemaster` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_Leavemaster`(

in pTypeFlg int(11)

)

BEGIN

if pTypeFlg= 1 Then

Select \* from leavemasterstatus order by leaveMasterStatus;

elseif pTypeFlg= 2 Then

Select \* from leavemastertype order by leaveMasterType;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_Leaveofdept` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_Leaveofdept`(

IN pUserID int,

IN pDeptID text,

IN pDate datetime,

OUT OnLeave smallint(5),

OUT TotalPresent smallint(5),

OUT OnOD smallint(5),

OUT LevPndRqst smallint(5),

OUT ODPndRqst smallint(5),

OUT CompOffPndRqst smallint(5)

)

BEGIN

DECLARE DeptMngUserID INT;

DECLARE pDeptIDAll text;

DECLARE UserRole varchar(10);

DECLARE SelDeptID text;

call splitter(pDeptID,',');

select group\_concat(SplitValue) INTO pDeptIDAll from splitResultsTmpTable;

SELECT group\_concat(DM.DeptManagerID) INTO DeptMngUserID from deptmanager DM where DM.DeptID IN (pDeptIDAll);

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pUserID;

/\*Get all hierarchy sub department ids\*/

if UserRole = 'ADMIN' then

CALL nex\_GetDeptTree(pDeptID,@AllHrcyDept);

SELECT @AllHrcyDept INTO SelDeptID;

else

SELECT pDeptID INTO SelDeptID;

end if;

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select pDeptID as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

/\*In Office Count\*/

SELECT count(DISTINCT A.AttendanceID) INTO TotalPresent from attendance A

LEFT JOIN userpersonalinfo U ON A.UserID = U.UserID

LEFT JOIN checkindetailsmore CDM ON A.UserID = CDM.UserID and CDM.TypeFlg = 'O' and DATE(CONVERT\_TZ(CDM.CheckIntime,'+00:00','+05:30')) = date(pDate)

where U.DeptID IN (select \* from temp\_AllDept) and U.Deactivated = 'N'

AND DATE(CONVERT\_TZ(A.LogTime,'+00:00','+05:30')) = date(pDate)

AND

(

pDate >= CONVERT\_TZ(CDM.CheckIntime,'+00:00','+05:30')

AND

((A.CheckOutTime IS NULL) OR (A.CheckOutTime < CDM.CheckIntime))

);

/\*In Leave Count\*/

create TEMPORARY TABLE temp\_AllDept2 like temp\_AllDept;

insert temp\_AllDept2 select \* from temp\_AllDept;

SELECT count(LA.LeavAppID) INTO OnLeave

FROM leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

WHERE U.DeptID IN (select \* from temp\_AllDept) and U.Deactivated = 'N' AND (date(pDate) >= LA.StartDate AND date(pDate) <= LA.EndDate)

AND LA.UserID NOT IN (DeptMngUserID)

AND LS.LeaveMasterStatusID = 2

AND LA.UserID NOT IN (

SELECT A.UserID from attendance A

LEFT JOIN userpersonalinfo U ON A.UserID = U.UserID

LEFT JOIN checkindetailsmore CDM ON A.UserID = CDM.UserID and CDM.TypeFlg = 'O' and DATE(CONVERT\_TZ(CDM.CheckIntime,'+00:00','+05:30')) = date(pDate)

where U.DeptID IN (select \* from temp\_AllDept2) and U.Deactivated = 'N'

AND DATE(CONVERT\_TZ(A.LogTime,'+00:00','+05:30')) = date(pDate)

AND

(

pDate >= CONVERT\_TZ(CDM.CheckIntime,'+00:00','+05:30')

AND

((A.CheckOutTime IS NULL) OR (A.CheckOutTime < CDM.CheckIntime))

)

);

DROP TEMPORARY TABLE temp\_AllDept2;

/\*In OD Count\*/

create TEMPORARY TABLE temp\_AllDept3 like temp\_AllDept;

insert temp\_AllDept3 select \* from temp\_AllDept;

SELECT count(OD.OdID) INTO OnOD from odapplied OD

LEFT JOIN userpersonalinfo U ON OD.UserID = U.UserID

WHERE OD.OdDate = date(pDate)

AND OD.RequestStatus = 'Approved'

AND

(

pDate >= OD.OdStartTime AND pDate <= OD.OdEndTime

)

AND U.DeptID IN (select \* from temp\_AllDept) and U.Deactivated = 'N'

AND OD.UserID NOT IN (DeptMngUserID)

AND OD.UserID NOT IN (

SELECT A.UserID from attendance A

LEFT JOIN userpersonalinfo U ON A.UserID = U.UserID

LEFT JOIN checkindetailsmore CDM ON A.UserID = CDM.UserID and CDM.TypeFlg = 'O' and DATE(CONVERT\_TZ(CDM.CheckIntime,'+00:00','+05:30')) = date(pDate)

where U.DeptID IN (select \* from temp\_AllDept3) and U.Deactivated = 'N'

AND DATE(CONVERT\_TZ(A.LogTime,'+00:00','+05:30')) = date(pDate)

AND

(

pDate >= CONVERT\_TZ(CDM.CheckIntime,'+00:00','+05:30')

AND

((A.CheckOutTime IS NULL) OR (A.CheckOutTime < CDM.CheckIntime))

)

);

DROP TEMPORARY TABLE temp\_AllDept3;

SELECT count(LA.LeavAppID) INTO LevPndRqst

FROM leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

WHERE U.DeptID IN (select \* from temp\_AllDept) and U.Deactivated = 'N'

AND LA.UserID NOT IN (DeptMngUserID)

AND LS.LeaveMasterStatusID = 1;

SELECT count(OD.OdID) INTO ODPndRqst from odapplied OD

LEFT JOIN userpersonalinfo U ON OD.UserID = U.UserID

WHERE OD.RequestStatus = 'Applied'

AND U.DeptID IN (select \* from temp\_AllDept) and U.Deactivated = 'N'

AND OD.UserID NOT IN (DeptMngUserID);

SELECT count(COMP.CompOffID) INTO CompOffPndRqst

FROM compoffapplied COMP

LEFT JOIN userpersonalinfo U ON COMP.UserID = U.UserID

WHERE COMP.RequestStatus = 'Applied'

AND U.DeptID IN (select \* from temp\_AllDept) and U.Deactivated = 'N'

AND COMP.UserID NOT IN (DeptMngUserID);

DROP TEMPORARY TABLE temp\_AllDept;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_MyLeaves` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_MyLeaves`(

IN pUserID int(11),

IN pDeptID int(11),

IN pEffcDate date,

IN pFlag char(10)

)

BEGIN

DECLARE i,NoOFTmpRecords INT DEFAULT 5;

DECLARE tmpStartDate,tmpEndDate DATE;

DECLARE UserDebitLeaveHld float;

DECLARE CheckExists int;

SET CheckExists = 0;

If pFlag = 'C' Then

select ULC.LeaveTypeID, LT.LeaveTypeText, SUM(ULC.CreditLeave) CreditLeave

from userleavecredit ULC

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

Where ULC.UserID = pUserID and date(CreditDate) <= pEffcDate AND ULC.IsCountable='Y'

group by ULC.LeaveTypeID

order by ULC.LeaveTypeID;

elseif pFlag = 'D' Then

select LA.LeaveTypeID, LT.LeaveTypeText,

SUM(LA.NoOfDays) DebitLeave

from leaveapplied LA

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (1,2) AND LA.IsCountable='Y'

group by LA.LeaveTypeID order by LA.LeaveTypeID;

elseif pFlag = 'D2' Then

select LA.LeaveTypeID, LT.LeaveTypeText,

SUM(LA.NoOfDays) DebitLeave

from leaveapplied LA

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (2) AND LA.IsCountable='Y'

group by LA.LeaveTypeID order by LA.LeaveTypeID;

elseif pFlag = 'D2-RPT' Then

select LA.LeaveTypeID, LT.LeaveTypeText,

SUM(LA.NoOfDays) DebitLeave

from leaveapplied LA

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (2)

AND (LA.EndDate <= pEffcDate OR LA.StartDate <= pEffcDate) AND LA.IsCountable='Y'

group by LA.LeaveTypeID order by LA.LeaveTypeID;

/\*

elseif pFlag = 'DOH' Then

CREATE TEMPORARY TABLE DebitLeaveHolidayTmpTable (

TmpID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

StartDate date NOT NULL,

EndDate date NOT NULL,

LeaveTypeID int NOT NULL,

LeavesInHoliday float NULL

);

INSERT INTO DebitLeaveHolidayTmpTable (StartDate,EndDate,LeaveTypeID)

select LA.StartDate,LA.EndDate,LA.LeaveTypeID

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (1,2) AND LA.IsCountable='Y' order by LA.LeaveTypeID;

select count(TmpID) INTO NoOFTmpRecords from DebitLeaveHolidayTmpTable;

set i = 1;

WHILE i <= NoOFTmpRecords DO

set UserDebitLeaveHld = 0.0;

select StartDate,EndDate INTO tmpStartDate,tmpEndDate from DebitLeaveHolidayTmpTable

where TmpID = i;

call nex\_GetLeaveInHolidayOrWeekend(pUserID,tmpStartDate,tmpEndDate,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into UserDebitLeaveHld;

update DebitLeaveHolidayTmpTable set LeavesInHoliday = UserDebitLEaveHld where TmpID = i;

SET i = i + 1;

END WHILE;

select LeaveTypeID,sum(LeavesInHoliday) DebitHolidayLeave

from DebitLeaveHolidayTmpTable group by LeaveTypeID order by LeaveTypeID;

drop TABLE DebitLeaveHolidayTmpTable;

\*/

/\*

elseif pFlag = 'DOH-RPT' Then

CREATE TEMPORARY TABLE DebitLeaveHolidayTmpTable (

TmpID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

StartDate date NOT NULL,

EndDate date NOT NULL,

LeaveTypeID int NOT NULL,

LeavesInHoliday float NULL

);

INSERT INTO DebitLeaveHolidayTmpTable (StartDate,EndDate,LeaveTypeID)

select LA.StartDate,LA.EndDate,LA.LeaveTypeID

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (1,2)

AND (LA.EndDate <= pEffcDate OR LA.StartDate <= pEffcDate)

AND LA.IsCountable='Y'

order by LA.LeaveTypeID;

select count(TmpID) INTO NoOFTmpRecords from DebitLeaveHolidayTmpTable;

set i = 1;

WHILE i <= NoOFTmpRecords DO

set UserDebitLeaveHld = 0.0;

select StartDate,EndDate INTO tmpStartDate,tmpEndDate from DebitLeaveHolidayTmpTable

where TmpID = i;

call nex\_GetLeaveInHolidayOrWeekend(pUserID,tmpStartDate,tmpEndDate,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into UserDebitLeaveHld;

update DebitLeaveHolidayTmpTable set LeavesInHoliday = UserDebitLEaveHld where TmpID = i;

SET i = i + 1;

END WHILE;

select LeaveTypeID,sum(LeavesInHoliday) DebitHolidayLeave

from DebitLeaveHolidayTmpTable group by LeaveTypeID order by LeaveTypeID;

drop TABLE DebitLeaveHolidayTmpTable;

\*/

elseif pFlag = 'ISL' Then

select LA.LeavAppID,LA.StartDate,LA.EndDate,LA.IsHalfDay from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LS.LeaveMasterStatusID = 2 and LA.UserID = pUserID

and pEffcDate between LA.StartDate and LA.EndDate

AND LA.IsCountable='Y';

else

/\*

Select count(LA.LeavAppID) INTO CheckExists

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LS.LeaveMasterStatusID = 2 AND LA.StartDate >= pEffcDate AND LA.IsCountable='Y';

If (CheckExists > 0) THEN

Select LA.LeavAppID,LA.UserID,U.DeptID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StartDate,LA.EndDate,

IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,

LT.LeaveTypeText,LS.LeaveStatusText

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LS.LeaveMasterStatusID = 2 AND LA.StartDate >= pEffcDate

AND LA.IsCountable='Y'

order by LA.StartDate asc LIMIT 0,1;

Else

Select LA.LeavAppID,LA.UserID,U.DeptID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StartDate,LA.EndDate,

IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,

LT.LeaveTypeText,LS.LeaveStatusText

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LS.LeaveMasterStatusID = 2 AND LA.StartDate <= pEffcDate

AND LA.IsCountable='Y'

order by LA.StartDate desc LIMIT 0,1;

End if;

\*/

/\*

Select 'Leave' as `Type`,LA.StartDate,LA.EndDate,IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,LT.LeaveTypeText

from leaveapplied LA

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LS.LeaveMasterStatusID = 2 AND LA.StartDate >= pEffcDate

AND LA.IsCountable='Y'

order by LA.StartDate asc LIMIT 0,1;

\*/

(Select 'Leave' as `Type`,LA.LeavAppID,LA.UserID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StartDate,LA.EndDate,LA.NoOfDays,LT.LeaveTypeText,LS.LeaveStatusText

from leaveapplied LA

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LS.LeaveMasterStatusID = 2 AND LA.StartDate >= pEffcDate

AND LA.IsCountable='Y'

order by LA.StartDate asc LIMIT 0,1)

UNION

(

select 'Holiday' as `Type`,H.HolidayID as LeavAppID,0 as UserID,0 as LeaveTypeID,0 as LeaveStatusID,

H.StartDate,H.EndDate,(DATEDIFF(H.EndDate,H.StartDate)+1) as NoOfDays,H.HolidayName,'' as LeaveStatusText

from holiday H

LEFT JOIN calendarholiday CH ON CH.HolidayID = H.HolidayID and H.StartDate >= pEffcDate

where CH.CalendarID = (select UC.CalendarID from usercalendar UC where UC.EffectedFrom <= pEffcDate and UC.UserID = pUserID order by UC.EffectedFrom desc limit 1)

order by H.StartDate asc LIMIT 0,1

)

order by StartDate asc limit 0,1;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_MyLeaves\_BKP` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_MyLeaves\_BKP`(

IN pUserID int(11),

IN pDeptID int(11),

IN pEffcDate date,

IN pFlag char(10)

)

BEGIN

DECLARE i,NoOFTmpRecords INT DEFAULT 5;

DECLARE tmpStartDate,tmpEndDate DATE;

DECLARE UserDebitLeaveHld float;

DECLARE CheckExists int;

SET CheckExists = 0;

If pFlag = 'C' Then

select ULC.LeaveTypeID, LT.LeaveTypeText, SUM(ULC.CreditLeave) CreditLeave

from userleavecredit ULC

LEFT JOIN leavetype LT ON ULC.LeaveTypeID = LT.LeaveTypeID

Where ULC.UserID = pUserID and date(CreditDate) <= pEffcDate AND ULC.IsCountable='Y'

group by ULC.LeaveTypeID

order by ULC.LeaveTypeID;

elseif pFlag = 'D' Then

select LA.LeaveTypeID, LT.LeaveTypeText,

SUM(IF(LA.IsHalfDay = 'Y',0.5,(DATEDIFF(LA.EndDate,LA.StartDate)+1))) DebitLeave

from leaveapplied LA

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (1,2) AND LA.IsCountable='Y'

group by LA.LeaveTypeID order by LA.LeaveTypeID;

elseif pFlag = 'D2' Then

select LA.LeaveTypeID, LT.LeaveTypeText,

SUM(IF(LA.IsHalfDay = 'Y',0.5,(DATEDIFF(LA.EndDate,LA.StartDate)+1))) DebitLeave

from leaveapplied LA

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (2) AND LA.IsCountable='Y'

group by LA.LeaveTypeID order by LA.LeaveTypeID;

elseif pFlag = 'D2-RPT' Then

select LA.LeaveTypeID, LT.LeaveTypeText,

SUM(IF(LA.IsHalfDay = 'Y',0.5,(DATEDIFF(LA.EndDate,LA.StartDate)+1))) DebitLeave

from leaveapplied LA

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (2)

AND (LA.EndDate <= pEffcDate OR LA.StartDate <= pEffcDate) AND LA.IsCountable='Y'

group by LA.LeaveTypeID order by LA.LeaveTypeID;

elseif pFlag = 'DOH' Then

CREATE TEMPORARY TABLE DebitLeaveHolidayTmpTable (

TmpID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

StartDate date NOT NULL,

EndDate date NOT NULL,

LeaveTypeID int NOT NULL,

LeavesInHoliday float NULL

);

INSERT INTO DebitLeaveHolidayTmpTable (StartDate,EndDate,LeaveTypeID)

select LA.StartDate,LA.EndDate,LA.LeaveTypeID

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (1,2) AND LA.IsCountable='Y' order by LA.LeaveTypeID;

select count(TmpID) INTO NoOFTmpRecords from DebitLeaveHolidayTmpTable;

set i = 1;

WHILE i <= NoOFTmpRecords DO

set UserDebitLeaveHld = 0.0;

select StartDate,EndDate INTO tmpStartDate,tmpEndDate from DebitLeaveHolidayTmpTable

where TmpID = i;

call nex\_GetLeaveInHolidayOrWeekend(pUserID,tmpStartDate,tmpEndDate,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into UserDebitLeaveHld;

update DebitLeaveHolidayTmpTable set LeavesInHoliday = UserDebitLEaveHld where TmpID = i;

SET i = i + 1;

END WHILE;

select LeaveTypeID,sum(LeavesInHoliday) DebitHolidayLeave

from DebitLeaveHolidayTmpTable group by LeaveTypeID order by LeaveTypeID;

drop TABLE DebitLeaveHolidayTmpTable;

elseif pFlag = 'DOH-RPT' Then

CREATE TEMPORARY TABLE DebitLeaveHolidayTmpTable (

TmpID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

StartDate date NOT NULL,

EndDate date NOT NULL,

LeaveTypeID int NOT NULL,

LeavesInHoliday float NULL

);

INSERT INTO DebitLeaveHolidayTmpTable (StartDate,EndDate,LeaveTypeID)

select LA.StartDate,LA.EndDate,LA.LeaveTypeID

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

Where LA.UserID = pUserID AND LS.LeaveMasterStatusID IN (1,2)

AND (LA.EndDate <= pEffcDate OR LA.StartDate <= pEffcDate)

AND LA.IsCountable='Y'

order by LA.LeaveTypeID;

select count(TmpID) INTO NoOFTmpRecords from DebitLeaveHolidayTmpTable;

set i = 1;

WHILE i <= NoOFTmpRecords DO

set UserDebitLeaveHld = 0.0;

select StartDate,EndDate INTO tmpStartDate,tmpEndDate from DebitLeaveHolidayTmpTable

where TmpID = i;

call nex\_GetLeaveInHolidayOrWeekend(pUserID,tmpStartDate,tmpEndDate,@DaysInHolidayOrWeekend);

select @DaysInHolidayOrWeekend into UserDebitLeaveHld;

update DebitLeaveHolidayTmpTable set LeavesInHoliday = UserDebitLEaveHld where TmpID = i;

SET i = i + 1;

END WHILE;

select LeaveTypeID,sum(LeavesInHoliday) DebitHolidayLeave

from DebitLeaveHolidayTmpTable group by LeaveTypeID order by LeaveTypeID;

drop TABLE DebitLeaveHolidayTmpTable;

elseif pFlag = 'ISL' Then

select LA.LeavAppID,LA.StartDate,LA.EndDate,LA.IsHalfDay from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LS.LeaveMasterStatusID = 2 and LA.UserID = pUserID

and pEffcDate between LA.StartDate and LA.EndDate

AND LA.IsCountable='Y';

else

/\*

Select count(LA.LeavAppID) INTO CheckExists

from leaveapplied LA

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LS.LeaveMasterStatusID = 2 AND LA.StartDate >= pEffcDate AND LA.IsCountable='Y';

If (CheckExists > 0) THEN

Select LA.LeavAppID,LA.UserID,U.DeptID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StartDate,LA.EndDate,

IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,

LT.LeaveTypeText,LS.LeaveStatusText

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LS.LeaveMasterStatusID = 2 AND LA.StartDate >= pEffcDate

AND LA.IsCountable='Y'

order by LA.StartDate asc LIMIT 0,1;

Else

Select LA.LeavAppID,LA.UserID,U.DeptID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StartDate,LA.EndDate,

IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,

LT.LeaveTypeText,LS.LeaveStatusText

from leaveapplied LA

LEFT JOIN userpersonalinfo U ON LA.UserID = U.UserID

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LS.LeaveMasterStatusID = 2 AND LA.StartDate <= pEffcDate

AND LA.IsCountable='Y'

order by LA.StartDate desc LIMIT 0,1;

End if;

\*/

/\*

Select 'Leave' as `Type`,LA.StartDate,LA.EndDate,IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,LT.LeaveTypeText

from leaveapplied LA

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LS.LeaveMasterStatusID = 2 AND LA.StartDate >= pEffcDate

AND LA.IsCountable='Y'

order by LA.StartDate asc LIMIT 0,1;

\*/

(Select 'Leave' as `Type`,LA.LeavAppID,LA.UserID,LA.LeaveTypeID,LA.LeaveStatusID,LA.StartDate,LA.EndDate,IF(LA.IsHalfDay = 'Y','0.5', (DATEDIFF(LA.EndDate,LA.StartDate)+1)) as NoOfDays,LT.LeaveTypeText,LS.LeaveStatusText

from leaveapplied LA

LEFT JOIN leavetype LT ON LA.LeaveTypeID = LT.LeaveTypeID

LEFT JOIN leavestatus LS ON LA.LeaveStatusID = LS.LeaveStatusID

where LA.UserID = pUserID AND LS.LeaveMasterStatusID = 2 AND LA.StartDate >= pEffcDate

AND LA.IsCountable='Y'

order by LA.StartDate asc LIMIT 0,1)

UNION

(

select 'Holiday' as `Type`,H.HolidayID as LeavAppID,0 as UserID,0 as LeaveTypeID,0 as LeaveStatusID,

H.StartDate,H.EndDate,(DATEDIFF(H.EndDate,H.StartDate)+1) as NoOfDays,H.HolidayName,'' as LeaveStatusText

from holiday H

LEFT JOIN calendarholiday CH ON CH.HolidayID = H.HolidayID and H.StartDate >= pEffcDate

where CH.CalendarID = (select UC.CalendarID from usercalendar UC where UC.EffectedFrom <= pEffcDate and UC.UserID = pUserID order by UC.EffectedFrom desc limit 1)

order by H.StartDate asc LIMIT 0,1

)

order by StartDate asc limit 0,1;

End If;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_OdListofdept` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_OdListofdept`(

in pUserID int,

in pDeptID text,

in pStartDate date,

in pEndDate date,

in pFlag char(3)

)

BEGIN

DECLARE DeptMngUserID INT;

DECLARE UserRole varchar(10);

DECLARE SelDeptID text;

SELECT group\_concat(DM.DeptManagerID) INTO DeptMngUserID from deptmanager DM where DM.DeptID IN (pDeptID);

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pUserID;

/\*Get all hierarchy sub department ids\*/

if UserRole = 'ADMIN' then

CALL nex\_GetDeptTree(pDeptID,@AllHrcyDept);

SELECT @AllHrcyDept INTO SelDeptID;

else

SELECT pDeptID INTO SelDeptID;

end if;

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select SelDeptID as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

if pFlag = 'ALL' then

SELECT U.UserID,U.FName,U.LName,U.EmployeeID,U.UserPicture,D.DepartmentName,

OD.OdID,OD.UserID,OD.OdDate,OdStartTime,OdEndTime,

(TIMESTAMPDIFF(Second,OD.OdStartTime, OD.OdEndTime)) TotalHours,

OD.OdReason,OD.OdLocation,OD.AppliedDate,OD.RequestStatus

FROM odapplied OD

LEFT JOIN userpersonalinfo U ON OD.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

where OD.ODDate BETWEEN pStartDate AND pEndDate

AND U.DeptID IN (select \* from temp\_AllDept)

AND OD.UserID NOT IN (DeptMngUserID)

AND OD.RequestStatus!='Applied'

ORDER BY OD.OdDate DESC;

elseif pFlag = 'PND' then

SELECT U.UserID,U.FName,U.LName,U.EmployeeID,U.UserPicture,D.DepartmentName,

OD.OdID,OD.UserID,OD.OdDate,OdStartTime,OdEndTime,

(TIMESTAMPDIFF(Second,OD.OdStartTime, OD.OdEndTime)) TotalHours,

OD.OdReason,OD.OdLocation,

OD.AppliedDate,OD.RequestStatus

FROM odapplied OD

LEFT JOIN userpersonalinfo U ON OD.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

where U.DeptID IN (select \* from temp\_AllDept)

AND OD.UserID NOT IN (DeptMngUserID)

AND OD.RequestStatus='Applied'

ORDER BY OD.OdDate DESC;

else

SELECT U.UserID,U.FName,U.LName,U.EmployeeID,U.UserPicture,D.DepartmentName,

OD.OdID,OD.UserID,OD.OdDate,OdStartTime,OdEndTime,

(TIMESTAMPDIFF(Second,OD.OdStartTime, OD.OdEndTime)) TotalHours,

OD.OdReason,OD.OdLocation,OD.AppliedDate

FROM odapplied OD

LEFT JOIN userpersonalinfo U ON OD.UserID = U.UserID

LEFT JOIN department D ON U.DeptID = D.DeptID

where OD.OdDate > pStartDate

AND U.DeptID IN (select \* from temp\_AllDept)

AND OD.UserID NOT IN (DeptMngUserID)

ORDER BY OD.OdDate DESC;

end if;

DROP TEMPORARY TABLE temp\_AllDept;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_Show\_OrgDetails` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_Show\_OrgDetails`(

In pOrgID int(11),

In pUserID int(11),

in pTypeFlg varchar(255)

)

BEGIN

Declare UserRole varchar(10);

Declare UserDept text;

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pUserID;

CASE pTypeFlg

WHEN 'U' THEN

if UserRole = 'SUPERVISOR' then

/\*Get all hierarchy sub department ids\*/

Select GROUP\_CONCAT(DeptID) into UserDept from deptmanager where DeptManagerID = pUserID;

CALL nex\_GetDeptTree(UserDept,@AllHrcyDept);

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

select UP.\*,D.DepartmentName,IFNULL(UR.RoleName,'EMPLOYEE') Role

from userpersonalinfo UP

LEFT JOIN department D ON D.DeptID = UP.DeptID

LEFT JOIN userworkinfo UWI ON UP.UserID = UWI.USERID

LEFT JOIN userrole UR ON UR.RoleID = UWI.RoleID

where UP.OrgID = pOrgID and UP.DeptID IN (select \* from temp\_AllDept) and UP.UserID != pUserID

ORDER BY concat(UP.FName,' ',LName);

elseif UserRole = 'ADMIN' then

select UP.\*,D.DepartmentName,IFNULL(UR.RoleName,'EMPLOYEE') Role

from userpersonalinfo UP

LEFT JOIN department D ON D.DeptID = UP.DeptID

LEFT JOIN userworkinfo UWI ON UP.UserID = UWI.USERID

LEFT JOIN userrole UR ON UR.RoleID = UWI.RoleID

where UP.OrgID = pOrgID ORDER BY concat(UP.FName,' ',LName);

else

select UP.\*,D.DepartmentName,IFNULL(UR.RoleName,'EMPLOYEE') Role

from userpersonalinfo UP

LEFT JOIN department D ON D.DeptID = UP.DeptID

LEFT JOIN userworkinfo UWI ON UP.UserID = UWI.USERID

LEFT JOIN userrole UR ON UR.RoleID = UWI.RoleID

where UP.OrgID = pOrgID and UP.UserID = pUserID ORDER BY concat(UP.FName,' ',LName);

end if;

/\*select UP.\* ,

IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE')

from userrole ur inner join userworkinfo uwi on ur.roleID = uwi.roleID where uwi.UserID = UP.UserID), 'EMPLOYEE') as Role,

D.DepartmentName

from userpersonalinfo UP LEFT JOIN department D ON D.DeptID = UP.DeptID

where UP.OrgID = pOrgID ORDER BY UP.UserID DESC;\*/

WHEN 'UC' THEN

select count(\*) as NoOfUsers from userpersonalinfo where OrgID = pOrgID;

WHEN 'D' THEN

if UserRole = 'SUPERVISOR' then

/\*Get all hierarchy sub department ids\*/

Select GROUP\_CONCAT(DeptID) into UserDept from deptmanager where DeptManagerID = pUserID;

CALL nex\_GetDeptTree(UserDept,@AllHrcyDept);

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

select D.DeptID,D.DeptPrtID,D.DepartmentName,D.DepartmentDesc,D.OrgID,

DM.DeptManagerID,U.Email,concat(U.FName,' ',U.LName) as UName from department D

LEFT JOIN deptmanager DM ON D.DeptID = DM.DeptID

LEFT JOIN userpersonalinfo U ON U.UserID = DM.DeptManagerID

where D.OrgID = pOrgID and D.DeptID IN (select \* from temp\_AllDept) order by D.DepartmentName;

elseif UserRole = 'ADMIN' then

select D.DeptID,D.DeptPrtID,D.DepartmentName,D.DepartmentDesc,D.OrgID,

DM.DeptManagerID,U.Email,concat(U.FName,' ',U.LName) as UName from department D

LEFT JOIN deptmanager DM ON D.DeptID = DM.DeptID

LEFT JOIN userpersonalinfo U ON U.UserID = DM.DeptManagerID

where D.OrgID = pOrgID order by D.DepartmentName;

else

select D.DeptID,D.DeptPrtID,D.DepartmentName,D.DepartmentDesc,D.OrgID,

DM.DeptManagerID,U.Email,concat(U.FName,' ',U.LName) as UName from department D

LEFT JOIN deptmanager DM ON D.DeptID = DM.DeptID

LEFT JOIN userpersonalinfo U ON U.UserID = DM.DeptManagerID

where D.OrgID = pOrgID and U.UserID = pUserID order by D.DepartmentName;

end if;

WHEN 'H' THEN

select \* from holiday where OrgID = pOrgID;

WHEN 'L' THEN

if UserRole = 'SUPERVISOR' then

/\*Get all hierarchy sub department ids\*/

Select GROUP\_CONCAT(DeptID) into UserDept from deptmanager where DeptManagerID = pUserID;

CALL nex\_GetDeptTree(UserDept,@AllHrcyDept);

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

select U.SiteLocation from userpersonalinfo U

where U.DeptID IN (select \* from temp\_AllDept) group by U.SiteLocation order by U.SiteLocation;

elseif UserRole = 'ADMIN' then

select U.SiteLocation from userpersonalinfo U

where U.OrgID = pOrgID group by U.SiteLocation order by U.SiteLocation;

else

select U.SiteLocation from userpersonalinfo U

where U.OrgID = pOrgID and U.UserID = pUserID group by U.SiteLocation order by U.SiteLocation;

end if;

WHEN 'W' THEN

select \* from weekends where OrgID = pOrgID;

ELSE

select \* from userpersonalinfo where OrgID = pOrgID;

END CASE;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UpdateAttendance` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UpdateAttendance`(

IN pUserID VARCHAR(15)

)

BEGIN

SELECT @vCheckIntime := min(CIN.CheckIntime)

FROM nexat.checkindetails CIN

Where CIN.UserID=pUserID

and Date(CIN.CheckIntime)= Date(Now());

SELECT @vCheckInLocation := CIN.Location

FROM nexat.checkindetails CIN

Where CIN.UserID=pUserID

and CIN.CheckIntime= @vCheckIntime;

SELECT @vCheckOuttime := max(COUT.CheckOuttime)

FROM nexat.checkoutdetails COUT

Where COUT.UserID=pUserID

and Date(COUT.CheckOuttime)= Date(Now());

SELECT @vCheckOutLocation := COUT.Location

FROM nexat.checkoutdetails COUT

Where COUT.UserID=pUserID

and COUT.CheckOuttime= @vCheckOuttime;

Select @vLogInTime := min(LIN.LogInTime) LIN

from logindetails LIN

Where LIN.UserID=pUserID

and Date(LIN.LogInTime)= Date(Now());

SELECT @vLogInLocation := LIN.Location

from logindetails LIN

Where LIN.UserID=pUserID

and LIN.LogInTime= @vLogInTime;

Select @vLogOuttime := max(LOUT.LogOuttime)

from logoutdetails LOUT

Where LOUT.UserID=pUserID

and Date(LOUT.LogOuttime)= Date(Now())

group by LOUT.UserID;

SELECT @vLogOutLocation= LOUT.Location

from logoutdetails LOUT

Where LOUT.UserID=pUserID

and LOUT.LogOuttime= @vLogOuttime;

Select @vCtr := ISNULL(NULLIF(Count(AttendanceID),0)) from nexat.attendance

Where UserID= pUserID and Date(LogTime)=Date(Now());

If @vCtr = 1 Then

Select @vMaxID := ISNULL(NULLIF(Max(AttendanceID),0)) + 1 from nexat.attendance;

INSERT INTO attendance

(

AttendanceID ,

UserID ,

LogInTime ,

CheckInTime ,

CheckOutTime ,

LogOutTime ,

LogInLocation ,

CheckInLocation ,

CheckOutLocation,

LogOutLocation ,

LogTime

)

VALUES

(

@vMaxID ,

pUserID ,

@vLogInTime ,

@vCheckIntime ,

@vCheckOuttime ,

@vLogOuttime ,

@vLogInLocation ,

@vCheckInLocation,

@vCheckOutLocation,

@vLogOutLocation ,

Now()

);

Else

Update attendance Set

CheckInTime= @vCheckIntime,

CheckOutTime= @vCheckOuttime,

LogInTime= @vLogInTime,

LogOutTime= @vLogOuttime,

LogInLocation= @vLogInLocation,

LogOutLocation= @vLogOutLocation,

CheckInLocation= @vCheckInLocation,

CheckOutLocation= @vCheckOutLocation

Where UserID= pUserID

and Date(LogTime)=Date(Now());

End IF;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UpdateAttendanceAndCheckIn` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UpdateAttendanceAndCheckIn`(

IN pUserID VARCHAR(15),

IN pOrigID int(11),

in pLocation varchar(145),

in pUserHost varchar(155),

in pSessionAuthID varchar(250)

)

BEGIN

start transaction;

INSERT INTO checkindetails

(

UserID,

OrgID,

Location,

CheckIntime,

UserHost,

SessionAuthID

)

VALUES

(

pUserID ,

pOrigID ,

pLocation ,

Now() ,

pUserHost ,

pSessionAuthID

);

Select @vCtr := ISNULL(NULLIF(Count(AttendanceID),0)) from nexat.attendance

Where UserID= pUserID and Date(LogTime)=Date(Now());

If @vCtr = 1 Then

INSERT INTO attendance

(

UserID ,

CheckInTime ,

CheckInLocation ,

LogTime

)

VALUES

(

pUserID ,

Now() ,

pLocation ,

Now()

);

Else

SELECT @vCheckIntime := min(CIN.CheckIntime)

FROM nexat.checkindetails CIN

Where CIN.UserID=pUserID

and Date(CIN.CheckIntime)= Date(Now());

SELECT @vCheckInLocation := CIN.Location

FROM nexat.checkindetails CIN

Where CIN.UserID=pUserID

and CIN.CheckIntime= @vCheckIntime;

Update attendance Set

CheckInTime= @vCheckIntime,

CheckInLocation= @vCheckInLocation

Where UserID= pUserID

and Date(LogTime)=Date(Now());

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UpdateAttendanceAndCheckInDetailsMore` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UpdateAttendanceAndCheckInDetailsMore`(

IN pUserID VARCHAR(15),

IN pOrigID int(11),

in pLocation varchar(145),

in pUserHost varchar(155),

in pSessionAuthID varchar(250),

in pTypeFlg Char(1),

in pQRCode varchar(1000),

in pOTPCode varchar(20),

in pCheckINTime varchar(20),

in pLocalCheckINTime varchar(20),

in pLocationLatLong varchar(255),

in pUserImage text,

in pOutdoorAttn varchar(1)

)

BEGIN

DECLARE vCtr1, vMoreCheckInID,vCtr2,vOTPID,vCtr int;

DECLARE vCheckIntime datetime;

DECLARE vCheckInLocation,attendanceLogTime,attendanceLogTimeType varchar(255);

DECLARE FirstCheckIn char(1);

/\*SKIP the Cannot add or update a child row: a foreign key constraint fails ERROR\*/

DECLARE CONTINUE HANDLER FOR 1452

SELECT 'Cannot add foreign key constraint';

SET FirstCheckIn = '';

start transaction;

INSERT INTO checkindetailsmore

(

UserID,

OrgID,

UserImage,

Location,

LocationLatLong,

CheckIntime,

UserHost,

SessionAuthID,

TypeFlg,

QRCode,

OutDoorAttn,

IsActive

)

VALUES

(

pUserID ,

pOrigID ,

pUserImage ,

pLocation ,

pLocationLatLong,

if(pCheckINTime='',Now(),CAST(pCheckINTime as DATETIME)),

pUserHost ,

pSessionAuthID ,

pTypeFlg ,

pQRCode ,

pOutdoorAttn ,

'Y'

);

Select @vCtr1 := Count(MoreCheckInID),@vMoreCheckInID := Max(MoreCheckInID)

INTO vCtr1,vMoreCheckInID

from checkindetailsmore

Where UserID= pUserID and Date(CheckIntime)=Date(CAST(pCheckINTime as DATETIME)) and TypeFlg='O';

if (pTypeFlg = 'O') then

Insert into checkinotpdetails

(

OTPCode,

UserID,

OTPTime,

MoreCheckInID,

OTPverified

)

Values

(

pOTPCode,

pUserID,

CAST(pCheckINTime as DATETIME),

vMoreCheckInID,

'N'

);

ElseIf (pTypeFlg = 'V') Then

/\*

Select @vCtr2 := ISNULL(NULLIF(Count(OTPID),0)),@vOTPID := OTPID

INTO vCtr2,vOTPID

from checkinotpdetails O

Inner Join checkindetailsmore M on O.MoreCheckInID= M.MoreCheckInID

Where M.UserID= pUserID and Date(O.OTPTime)=Date(CAST(pCheckINTime as DATETIME))

and O.OTPCode=pOTPCode and O.OTPverified='N';

\*/

INSERT INTO checkindetails

(

UserID,

OrgID,

Location,

CheckIntime,

UserHost,

SessionAuthID

)

VALUES

(

pUserID ,

pOrigID ,

pLocation ,

CAST(pCheckINTime as DATETIME) ,

pUserHost ,

pSessionAuthID

);

/\*Decision taking - check attendance Log time with GMT time or Local time\*/

If(Date(CAST(pCheckINTime as DATETIME)) = Date(CAST(pLocalCheckINTime as DATETIME))) then

Select @vCtr := ISNULL(NULLIF(Count(AttendanceID),0))

INTO vCtr

from attendance

Where UserID= pUserID and Date(LogTime)=Date(CAST(pCheckINTime as DATETIME));

set attendanceLogTime = pCheckINTime;

set attendanceLogTimeType = 'GMT';

Else

Select @vCtr := ISNULL(NULLIF(Count(AttendanceID),0))

INTO vCtr

from attendance

Where UserID= pUserID and Date(LogTime)=Date(CAST(pLocalCheckINTime as DATETIME));

set attendanceLogTime = pLocalCheckINTime;

set attendanceLogTimeType = 'LOCAL';

End If;

/\*

SELECT @vCheckIntime := min(CIN.CheckIntime)

INTO vCheckIntime

FROM checkindetailsmore CIN

Where CIN.UserID=pUserID and TypeFlg = 'O'

and Date(CIN.CheckIntime)= Date(CAST(pCheckINTime as DATETIME));

SELECT @vCheckInLocation := CIN.Location

INTO vCheckInLocation

FROM checkindetailsmore CIN

Where CIN.UserID=pUserID and TypeFlg = 'O'

and DATe(CIN.CheckIntime)= Date(CAST(pCheckINTime as DATETIME))

Order by CIN.CheckIntime

LIMIT 1;

\*/

set vCheckIntime = CAST(pCheckINTime as DATETIME);

set vCheckInLocation = pLocation;

If vCtr = 1 Then

INSERT INTO attendance

(

UserID ,

CheckInTime ,

CheckInLocation ,

LogTime,

LogTimeType

)

VALUES

(

pUserID ,

vCheckIntime ,

vCheckInLocation ,

CAST(attendanceLogTime as DATETIME),

attendanceLogTimeType

);

SET FirstCheckIn = 'Y';

/\*else

Update attendance Set

CheckInTime= vCheckIntime,

CheckInLocation= vCheckInLocation

Where UserID= pUserID

and Date(LogTime)=Date(CAST(attendanceLogTime as DATETIME));

\*/

End If;

/\*Update checkinotpdetails set OTPverified='Y' Where OTPID= vOTPID;\*/

Else

SELECT @vCheckIntime := min(CIN.CheckIntime)

INTO vCheckIntime

FROM checkindetailsmore CIN

Where CIN.UserID=pUserID

and Date(CIN.CheckIntime)= Date(CAST(pCheckINTime as DATETIME));

SELECT @vCheckInLocation := CIN.Location

INTO vCheckInLocation

FROM checkindetailsmore CIN

Where CIN.UserID=pUserID

and CIN.CheckIntime= @vCheckIntime;

update checkindetails set

Location= vCheckInLocation,

CheckIntime= vCheckIntime

Where UserID= pUserID

and Date(CheckIntime)=Date(CAST(pCheckINTime as DATETIME));

End if;

commit;

Select A.AttendanceID,pTypeFlg,FirstCheckIn from attendance A where A.UserID = pUserID order by A.AttendanceID desc limit 0,1;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UpdateAttendanceAndCheckOut` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UpdateAttendanceAndCheckOut`(

IN pUserID VARCHAR(15),

IN pOrigID int(11),

in pLocation varchar(145),

in pUserHost varchar(155),

in pSessionAuthID varchar(250)

)

BEGIN

start transaction;

INSERT INTO checkoutdetails

(

UserID,

OrgID,

Location,

CheckOuttime,

UserHost,

SessionAuthID

)

VALUES

(

pUserID ,

pOrigID ,

pLocation ,

Now() ,

pUserHost ,

pSessionAuthID

);

Select @vCtr := ISNULL(NULLIF(Count(AttendanceID),0)) from nexat.attendance

Where UserID= pUserID and Date(LogTime)=Date(Now());

If @vCtr = 1 Then

INSERT INTO attendance

(

UserID ,

CheckOutTime ,

CheckOutLocation ,

LogTime

)

VALUES

(

pUserID ,

Now() ,

pLocation ,

Now()

);

Else

SELECT @vCheckOuttime := max(COUT.CheckOuttime)

FROM nexat.checkoutdetails COUT

Where COUT.UserID=pUserID

and Date(COUT.CheckOuttime)= Date(Now());

SELECT @vCheckOutLocation := COUT.Location

FROM nexat.checkoutdetails COUT

Where COUT.UserID=pUserID

and COUT.CheckOuttime= @vCheckOuttime;

Update attendance Set

CheckOutTime= @vCheckOuttime,

CheckOutLocation= @vCheckOutLocation

Where UserID= pUserID

and Date(LogTime)=Date(Now());

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UpdateAttendanceAndCheckOutMore` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UpdateAttendanceAndCheckOutMore`(

IN pUserID VARCHAR(15),

IN pOrigID int(11),

in pLocation varchar(145),

in pUserHost varchar(155),

in pSessionAuthID varchar(250),

in pTypeFlg Char(1),

in pQRCode varchar(500),

in pPurpose varchar(50),

in pCheckOutTime varchar(20),

in pLocationLatLong varchar(255),

in pUserImage text,

in pOutdoorAttn varchar(1)

)

BEGIN

declare LastRecentCheckIn INT;

DECLARE CONTINUE HANDLER FOR 1452

SELECT 'Cannot add foreign key constraint';

start transaction;

INSERT INTO checkoutdetailsmore

(

UserID,

OrgID,

UserImage,

Location,

LocationLatLong,

Checkouttime,

UserHost,

SessionAuthID,

TypeFlg,

QRCode,

Purpose,

OutDoorAttn,

IsActive

)

VALUES

(

pUserID ,

pOrigID ,

pUserImage ,

pLocation ,

pLocationLatLong,

if(pCheckOutTime='',Now(),CAST(pCheckOutTime as DATETIME)),

pUserHost ,

pSessionAuthID ,

pTypeFlg ,

pQRCode ,

pPurpose ,

pOutdoorAttn ,

'Y'

);

/\*Select @vCtr1 := Count(MoreCheckoutID),

@vCheckouttime := Max(Checkouttime) from checkoutdetailsmore

Where UserID= pUserID and Date(Checkouttime)=Date(if(pCheckOutTime='',Now(),CAST(pCheckOutTime as DATETIME)));\*/

if (pTypeFlg = 'F') then

select IFNULL(A.AttendanceID,0) into LastRecentCheckIn from attendance A

where A.UserID= pUserID

and A.CheckInTime >= DATE\_SUB(CAST(pCheckOutTime as DATETIME), INTERVAL 1 DAY) and A.CheckInTime <= CAST(pCheckOutTime as DATETIME)

order by A.CheckInTime desc limit 0,1;

if LastRecentCheckIn > 0 then

Update attendance Set

CheckOutTime= pCheckOutTime,

CheckOutLocation= pLocation,

AutoCheckOut= NULL

Where UserID= pUserID

and AttendanceID = LastRecentCheckIn;

end if;

End if;

select 1;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UpdateAttendanceAndLogkIn` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UpdateAttendanceAndLogkIn`(

IN pUserID VARCHAR(15),

IN pOrigID int(11),

in pLocation varchar(145),

in pUserHost varchar(155)

)

BEGIN

start transaction;

INSERT INTO logindetails

(

UserID,

OrgID,

Location,

LogInTime,

UserHost

)

VALUES

(

pUserID ,

pOrigID ,

pLocation ,

Now() ,

pUserHost

);

Select @vCtr := ISNULL(NULLIF(Count(AttendanceID),0)) from nexat.attendance

Where UserID= pUserID and Date(LogTime)=Date(Now());

If @vCtr = 1 Then

INSERT INTO attendance

(

UserID ,

LogInTime ,

LogInLocation ,

LogTime

)

VALUES

(

pUserID ,

Now() ,

pLocation ,

Now()

);

Else

Select @vLogInTime := min(LIN.LogInTime) LIN

from logindetails LIN

Where LIN.UserID=pUserID

and Date(LIN.LogInTime)= Date(Now());

SELECT @vLogInLocation := LIN.Location

from logindetails LIN

Where LIN.UserID=pUserID

and LIN.LogInTime= @vLogInTime;

Update attendance Set

LogInTime= @vLogInTime,

LogInLocation= @vLogInLocation

Where UserID= pUserID

and Date(LogTime)=Date(Now());

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UpdateAttendanceAndLogkOut` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UpdateAttendanceAndLogkOut`(

IN pUserID VARCHAR(15),

IN pOrigID int(11),

in pLocation varchar(145),

in pUserHost varchar(155)

)

BEGIN

start transaction;

INSERT INTO logoutdetails

(

UserID,

OrgID,

Location,

LogOuttime,

UserHost

)

VALUES

(

pUserID ,

pOrigID ,

pLocation ,

Now() ,

pUserHost

);

Select @vCtr := ISNULL(NULLIF(Count(AttendanceID),0)) from nexat.attendance

Where UserID= pUserID and Date(LogTime)=Date(Now());

If @vCtr = 1 Then

INSERT INTO attendance

(

UserID ,

LogOutTime ,

LogOutLocation ,

LogTime

)

VALUES

(

pUserID ,

Now() ,

pLocation ,

Now()

);

Else

Select @vLogOuttime := max(LOUT.LogOuttime)

from logoutdetails LOUT

Where LOUT.UserID=pUserID

and Date(LOUT.LogOuttime)= Date(Now())

group by LOUT.UserID;

SELECT @vLogOutLocation= LOUT.Location

from logoutdetails LOUT

Where LOUT.UserID=pUserID

and LOUT.LogOuttime= @vLogOuttime;

Update attendance Set

LogOutTime= @vLogOuttime,

LogOutLocation= @vLogOutLocation

Where UserID= pUserID

and Date(LogTime)=Date(Now());

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserAllOfOrg` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserAllOfOrg`(

in pOrgID int,

in pDeptID int

)

BEGIN

if pDeptID = 0 then

Select UP.UserID,UP.EmployeeID,UP.EmploymentType,CONCAT(UP.FName,' ',UP.LName)as FullName,UP.SiteLocation,UP.Deactivated,UP.DeactivatedOn,

IFNULL(UWI.Designation, '') as Designation

from userpersonalinfo UP LEFT OUTER join userworkinfo UWI on UP.UserID = UWI.UserID

where UP.DeptID IS NULL and UP.OrgID = pOrgID

order by UP.EmployeeID,FullName;

else

Select UP.UserID,UP.EmployeeID,UP.EmploymentType,CONCAT(UP.FName,' ',UP.LName)as FullName,UP.SiteLocation,UP.Deactivated,UP.DeactivatedOn,

IFNULL(UWI.Designation, '') as Designation,D.DepartmentName,

(Select CONCAT(Fname, ' ', LName) from userpersonalinfo where UserID = DPM.DeptManagerID) as Supervisor

from userpersonalinfo UP LEFT OUTER join userworkinfo UWI on UP.UserID = UWI.UserID

LEFT JOIN department D ON D.DeptID = UP.DeptID

LEFT JOIN deptmanager DPM on D.DeptID = DPM.DeptID

where UP.DeptID = pDeptID and UP.OrgID = pOrgID

order by UP.EmployeeID,FullName;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserAssignAdmin` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserAssignAdmin`(

in pOrgID int(11),

in pUserID int(11),

in pAssignType char(1)

)

BEGIN

DECLARE AdminRoleID,SupervisorRoleID,IsAlreadyAdmin,IsAlreadySupervisor int;

select RoleID into AdminRoleID from userrole where OrgID = pOrgID and RoleName = 'ADMIN';

select RoleID into SupervisorRoleID from userrole where OrgID = pOrgID and RoleName = 'SUPERVISOR';

select count(UserWorkInfoID) into IsAlreadyAdmin from userworkinfo

where UserID = pUserID and RoleId = AdminRoleID;

Select count(DeptID) into IsAlreadySupervisor from deptmanager

where DeptManagerID = pUserID limit 1;

start transaction;

if pAssignType= 'Y' Then

if IsAlreadyAdmin = 0 then

if IsAlreadySupervisor >0 then

UPDATE userworkinfo SET RoleID = AdminRoleID WHERE UserID = pUserID;

else

INSERT INTO userworkinfo

(

UserID,

RoleID,

TeamID

)

VALUES

(

pUserID,

AdminRoleID,

0

);

end if;

end if;

else

if IsAlreadyAdmin > 0 then

if IsAlreadySupervisor >0 then

UPDATE userworkinfo SET RoleID = SupervisorRoleID WHERE UserID = pUserID;

else

DELETE from userworkinfo where UserID = pUserID and RoleID = AdminRoleID;

end if;

end if;

End If;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserAssignRole` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserAssignRole`(

in pOrgID int(11),

in pUserID int(11),

in pAssignType char(1)

)

BEGIN

DECLARE AdminRoleID,SupervisorRoleID,SecurityRoleID,IsAlreadyAdmin,IsAlreadySupervisor,IsAlreadySecurity int;

select RoleID into AdminRoleID from userrole where OrgID = pOrgID and RoleName = 'ADMIN';

select RoleID into SupervisorRoleID from userrole where OrgID = pOrgID and RoleName = 'SUPERVISOR';

select count(UserWorkInfoID) into IsAlreadyAdmin from userworkinfo

where UserID = pUserID and RoleID = AdminRoleID;

Select count(DeptID) into IsAlreadySupervisor from deptmanager

where DeptManagerID = pUserID limit 1;

SELECT RoleID into SecurityRoleID from userrole where OrgID = pOrgID and RoleName = 'SECURITY';

IF SecurityRoleID IS NULL THEN

INSERT INTO userrole(RoleName,OrgID) VALUES ('SECURITY', pOrgID);

SET SecurityRoleID = LAST\_INSERT\_ID();

END IF;

SELECT count(UserWorkInfoID) into IsAlreadySecurity FROM userworkinfo

WHERE UserID = pUserID and RoleID = SecurityRoleID;

start transaction;

CASE pAssignType

WHEN 'A' THEN

IF IsAlreadyAdmin > 0 OR IsAlreadySupervisor > 0 OR IsAlreadySecurity>0 THEN

UPDATE userworkinfo SET RoleID = AdminRoleID,Designation='Admin' WHERE UserID = pUserID;

ELSE

INSERT INTO userworkinfo

(

UserID,

RoleID,

TeamID,

Designation

)

VALUES

(

pUserID,

AdminRoleID,

0,

'Admin'

);

END IF;

WHEN 'E' THEN

DELETE from userworkinfo where UserID = pUserID;

ELSE

IF IsAlreadyAdmin > 0 Or IsAlreadySupervisor > 0 OR IsAlreadySecurity >0 THEN

UPDATE userworkinfo SET RoleID = SecurityRoleID,Designation='Security' WHERE UserID = pUserID;

ELSE

INSERT INTO userworkinfo

(

UserID,

RoleID,

TeamID,

Designation

)

VALUES

(

pUserID,

SecurityRoleID,

0,

'Security'

);

END IF;

END CASE;

commit;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserByDepartment` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserByDepartment`(

in pDepartmentName varchar (50),

in pOrgID int

)

BEGIN

Select U.UserID,U.FName,U.LName,U.EmployeeID,U.Email,U.UserPicture,U.WorkPhone,U.CellNo,D.DepartmentName

from userpersonalinfo U left join department D ON U.DeptId = D.DeptID

where D.DepartmentName like CONCAT('%',pDepartmentName,'%') and U.OrgID = pOrgID and U.Deactivated='N';

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserByDeptLocation` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserByDeptLocation`(

in pDeptID text,

in pLocationName text,

in pOrgID int,

in pUserID int,

in pTypeFlg int(1)

)

BEGIN

Declare UserRole varchar(10);

Declare UserDept int;

Select IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID

where uwi.UserID = U.UserID), 'EMPLOYEE') into UserRole from userpersonalinfo U where U.UserID = pUserID;

if pTypeFlg = 1 then

select DISTINCT U.SiteLocation

from userpersonalinfo U where U.OrgID = pOrgID and U.SiteLocation IS NOT NULL;

elseif pTypeFlg= 2 then

if pDeptID!='' and pLocationName!='' then

call splitter(pDeptID,',');

CREATE TEMPORARY TABLE IF NOT EXISTS splitResultsTmpTable2 AS (SELECT \* FROM splitResultsTmpTable);

call splitter(pLocationName,',');

select U.UserID,U.FName,U.LName,concat(U.FName,' ',U.LName) as UserName,U.Email,U.EmployeeID,U.UserPicture,U.Deactivated,

U.DeptID,U.SiteLocation,DP.DepartmentName,

IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur inner join userworkinfo uwi on ur.roleID = uwi.roleID where uwi.UserID = U.UserID), 'EMPLOYEE') as Role

from userpersonalinfo U LEFT JOIN department DP on U.DeptID = DP.DeptID

where U.OrgID = pOrgID and U.Deactivated='N'

and U.DeptID IN (select SplitValue from splitResultsTmpTable2) and U.SiteLocation IN (select SplitValue from splitResultsTmpTable);

DROP TABLE splitResultsTmpTable2;

elseif pDeptID!='' and pLocationName='' then

call splitter(pDeptID,',');

select U.UserID,U.FName,U.LName,concat(U.FName,' ',U.LName) as UserName,U.Email,U.EmployeeID,U.UserPicture,U.Deactivated,

U.DeptID,U.SiteLocation,DP.DepartmentName,

IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur inner join userworkinfo uwi on ur.roleID = uwi.roleID where uwi.UserID = U.UserID), 'EMPLOYEE') as Role

from userpersonalinfo U LEFT JOIN department DP on U.DeptID = DP.DeptID

where U.OrgID = pOrgID and U.Deactivated='N'

and U.DeptID IN (select SplitValue from splitResultsTmpTable);

elseif pDeptID='' and pLocationName!='' then

/\*Get all hierarchy sub department ids\*/

Select DeptID into UserDept from deptmanager where DeptManagerID = pUserID;

CALL nex\_GetDeptTree(UserDept,@AllHrcyDept);

CREATE TEMPORARY TABLE temp\_AllDept

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(t.Dept\_Id, ',', n.n), ',', -1) value

FROM (select @AllHrcyDept as Dept\_Id) t CROSS JOIN

(

SELECT a.N + b.N \* 10 + 1 n

FROM

(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) a

,(SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9) b

ORDER BY n

) n

WHERE n.n <= 1 + (LENGTH(t.Dept\_Id) - LENGTH(REPLACE(t.Dept\_Id, ',', '')));

call splitter(pLocationName,',');

select U.UserID,U.FName,U.LName,concat(U.FName,' ',U.LName) as UserName,U.Email,U.EmployeeID,U.UserPicture,U.Deactivated,

U.DeptID,U.SiteLocation,DP.DepartmentName,

IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur inner join userworkinfo uwi on ur.roleID = uwi.roleID where uwi.UserID = U.UserID), 'EMPLOYEE') as Role

from userpersonalinfo U LEFT JOIN department DP on U.DeptID = DP.DeptID

where U.OrgID = pOrgID and U.Deactivated='N' and U.DeptID IN (select \* from temp\_AllDept)

and U.SiteLocation IN (select SplitValue from splitResultsTmpTable);

end if;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserByEmail` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserByEmail`(

in pUserEmail varchar (50),

in pOrgID int

)

BEGIN

if pOrgID = 0 then

Select OrgID,UserID,EmployeeID,FName,LName,Email,Pwd,Deactivated from userpersonalinfo where Email=pUserEmail;

else

Select \* from userpersonalinfo where Email=pUserEmail and OrgID = pOrgID;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserByEmpID` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserByEmpID`(

in pEmpID varchar(50),

in pOrgID int

)

BEGIN

Select UserID,FName,LName,Email,DeptID,EmployeeID from userpersonalinfo U

where U.EmployeeID= pEmpID and U.OrgID = pOrgID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserByMobile` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserByMobile`(

in pMobileNo varchar (50),

in pName varchar (50),

in pOrgID int

)

BEGIN

if pName!='' and pMobileNo!='' then

Select U.UserID,U.FName,U.LName,U.EmployeeID,U.Email,U.UserPicture,U.WorkPhone,U.CellNo,D.DepartmentName

from userpersonalinfo U

left join department D ON U.DeptId = D.DeptID

where

((U.CellNo = pMobileNo or U.WorkPhone = pMobileNo)

or (CONCAT(U.FName,' ',U.LName) like CONCAT('%',pName,'%')))

and U.OrgID = pOrgID and U.Deactivated='N';

elseif pName!='' and pMobileNo='' then

Select U.UserID,U.FName,U.LName,U.EmployeeID,U.Email,U.UserPicture,U.WorkPhone,U.CellNo,D.DepartmentName

from userpersonalinfo U

left join department D ON U.DeptId = D.DeptID

where

(CONCAT(U.FName,' ',U.LName) like CONCAT('%',pName,'%'))

and U.OrgID = pOrgID and U.Deactivated='N';

elseif pName='' and pMobileNo!='' then

Select U.UserID,U.FName,U.LName,U.EmployeeID,U.Email,U.UserPicture,U.WorkPhone,U.CellNo,D.DepartmentName

from userpersonalinfo U

left join department D ON U.DeptId = D.DeptID

where

(U.CellNo = pMobileNo or U.WorkPhone = pMobileNo)

and U.OrgID = pOrgID and U.Deactivated='N';

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserByTenure` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserByTenure`(

in pOrgID int(11),

in pEffectiveDate date,

in pMinYear varchar(5),

in pMaxYear varchar(5)

)

BEGIN

SELECT U.UserID,U.FName,U.LName,concat(U.FName,' ',U.LName) as UserName,

U.Email,U.EmployeeID,U.DateOfJoining,

DATEDIFF(pEffectiveDate,DateOfJoining) as TenureDays,

FLOOR(DATEDIFF(pEffectiveDate,DateOfJoining)/365) as TenureYears

FROM userpersonalinfo U where OrgID = pOrgID

and (FLOOR(DATEDIFF(pEffectiveDate,DateOfJoining)/365) >= pMinYear and FLOOR(DATEDIFF(pEffectiveDate,DateOfJoining)/365) < pMaxYear);

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserCalendars` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserCalendars`(

in pUserID int,

in porgID int

)

BEGIN

select C.CalendarName,C.CalendarDesc,UC.EffectedFrom

from usercalendar UC

LEFT JOIN calendar C ON UC.CalendarID = C.CalendarID

where UC.UserID = pUserID and C.OrgID = porgID

order by UC.EffectedFrom desc;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserDetails` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserDetails`(

in pUserEmail VARCHAR(100),

in porgID int

)

BEGIN

Select UP.\*, DP.DepartmentName as DeptName,

(Select DeptID from deptmanager where DeptManagerID = UP.UserID limit 1) as MngrOfDept,

(Select CONCAT(Fname, ' ', LName) from userpersonalinfo where UserID = DPM.DeptManagerID) as Supervisor,

IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur inner join userworkinfo uwi on ur.roleID = uwi.roleID where uwi.UserID = UP.UserID), 'EMPLOYEE') as Role,

IFNULL(UWI.Designation, '') as Designation

from userpersonalinfo UP LEFT OUTER Join department DP on UP.DeptID = DP.DeptID

LEFT OUTER join deptmanager DPM on DP.DeptID = DPM.DeptID

LEFT OUTER join userworkinfo UWI on UP.UserID = UWI.UserID

where Email = pUserEmail and UP.OrgID = porgID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserDetailsByID` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`111.93.226.178` PROCEDURE `nex\_UserDetailsByID`(

in pUserID VARCHAR(100),

in porgID int

)

BEGIN

if porgID > 0 then

Select UP.UserID,UP.FName,UP.LName,UP.Email,UP.EmployeeID,UP.UserPicture,UP.DeptID,UP.DateOfJoining,UP.ResignationDate,UP.ReleaseDate,UP.ReleaseType,UP.EmploymentType,UP.Address,UP.SiteLocation,UP.CellNo,UP.WorkPhone,

DP.DepartmentName as DeptName,

(Select GROUP\_CONCAT(DeptID) from deptmanager where DeptManagerID = UP.UserID) as MngrOfDept,

(Select CONCAT(Fname, ' ', LName) from userpersonalinfo where userID = DPM.DeptManagerID) as Supervisor,

IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur inner join userworkinfo uwi on ur.roleID = uwi.roleID where uwi.userID = UP.UserID), 'EMPLOYEE') as Role,

IFNULL(UWI.Designation, '') as Designation,

DPM.DeptManagerID

from userpersonalinfo UP LEFT OUTER Join department DP on UP.DeptID = DP.DeptID

LEFT OUTER join deptmanager DPM on DP.DeptID = DPM.DeptID

LEFT OUTER join userworkinfo UWI on UP.UserID = UWI.UserID

where UP.UserID = pUserID and UP.OrgID = porgID;

else

Select UP.UserID,UP.FName,UP.LName,UP.Email,UP.EmployeeID,UP.UserPicture,UP.DeptID,UP.DateOfJoining,UP.ResignationDate,UP.ReleaseDate,UP.ReleaseType,UP.EmploymentType,UP.Address,UP.SiteLocation,UP.CellNo,UP.WorkPhone,

DP.DepartmentName as DeptName,

(Select GROUP\_CONCAT(DeptID) from deptmanager where DeptManagerID = UP.UserID) as MngrOfDept,

(Select CONCAT(Fname, ' ', LName) from userpersonalinfo where UserID = DPM.DeptManagerID) as Supervisor,

IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur inner join userworkinfo uwi on ur.roleID = uwi.roleID where uwi.UserID = UP.UserID), 'EMPLOYEE') as Role,

IFNULL(UWI.Designation, '') as Designation,

DPM.DeptManagerID

from userpersonalinfo UP LEFT OUTER Join department DP on UP.DeptID = DP.DeptID

LEFT OUTER join deptmanager DPM on DP.DeptID = DPM.DeptID

LEFT OUTER join userworkinfo UWI on UP.UserID = UWI.UserID

where UP.UserID = pUserID;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserLogin` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserLogin`(

in pUserEmail varchar (50),

in pPassword varchar (50)

)

BEGIN

/\*

Select UP.\*, DP.DepartmentName as DeptName,

(Select CONCAT(Fname, ' ', LName) from userpersonalInfo where DeptID = UP.DeptID and UserID = DPM.DeptManagerID) as Supervisor,

IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur inner join userworkinfo uwi on ur.roleID = uwi.roleID where uwi.UserID = UP.UserID), 'EMPLOYEE') as Role,

IFNULL(UWI.Designation, '') as Designation

from userpersonalinfo UP LEFT OUTER Join department DP on UP.DeptID = DP.DeptID

LEFT OUTER join deptmanager DPM on DP.DeptID = DPM.DeptID

LEFT OUTER join UserworkInfo UWI on UP.UserID = UWI.UserID

where Email=pUserEmail and Pwd=pPassword and deleted=0;

\*/

Select UP.UserID,UP.OrgID,UP.FName,UP.LName,UP.Email,UP.UserPicture,UP.WorkPhone,UP.CellNo,UP.EmployeeID,UP.DeptID,

UP.PwdResetrequired,UP.Deactivated,

DP.DepartmentName as DeptName,

(Select GROUP\_CONCAT(DeptID) from deptmanager where DeptManagerID = UP.UserID) as MngrOfDept,

(Select CONCAT(Fname, ' ', LName) from userpersonalinfo where UserID = DPM.DeptManagerID) as Supervisor,

IFNULL((Select IFNULL(ur.rolename, 'EMPLOYEE') from userrole ur INNER JOIN userworkinfo uwi on ur.roleID = uwi.roleID where uwi.UserID = UP.UserID), 'EMPLOYEE') as Role,

IFNULL(UWI.Designation, '') as Designation,

concat(SAPUP.FName,' ',SAPUP.LName) as RequestToSupervisor, SAP.RequestStatus as RequestToSupervisorStatus

from userpersonalinfo UP LEFT JOIN department DP on UP.DeptID = DP.DeptID

LEFT JOIN deptmanager DPM on DP.DeptID = DPM.DeptID

LEFT JOIN userworkinfo UWI on UP.UserID = UWI.UserID

LEFT JOIN supervisorapproval SAP on UP.UserID = SAP.UserID

LEFT JOIN userpersonalinfo SAPUP on SAP.SupervisorID = SAPUP.UserID

where UP.Email=pUserEmail and UP.Pwd like binary pPassword and UP.deleted=0

order by SAP.AppliedDate desc

limit 1;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserLogout` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserLogout`(

in pUserID int(11)

)

BEGIN

UPDATE userdevices SET IsLoggedIn='N' WHERE UserID = pUserID;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserRecentAttendance` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserRecentAttendance`(

in pUserID int(11),

in currentDate date,

in pFlag varchar(20)

)

BEGIN

if pFlag = 'AFTER-CHECKOUT' then

Select UP.UserID,UP.FName,UP.LName,UP.Email,UP.EmployeeID,UP.OrgID,DPM.DeptManagerID,

A.AttendanceID,

date(A.CheckInTime) as AttnDate,

Time(A.CheckInTime) as CheckInTime,

Time(A.CheckOutTime) CheckOutTime,

/\*(TIMESTAMPDIFF(Hour, A.CheckInTime, A.CheckOutTime) + (TIMESTAMPDIFF(Minute, A.CheckInTime, A.CheckOutTime) - ((TIMESTAMPDIFF(Hour, A.CheckInTime, A.CheckOutTime)) \* 60)) \*.01) TotalHours\*/

(TIMESTAMPDIFF(Second, CheckInTime, CheckOutTime)) TotalHours

from userpersonalinfo UP

LEFT OUTER Join department DP on UP.DeptID = DP.DeptID

LEFT OUTER join deptmanager DPM on DP.DeptID = DPM.DeptID

LEFT OUTER join attendance A on UP.UserID = A.UserID

where UP.UserID = pUserID and date(A.LogTime) = currentDate;

elseif pFlag = 'BEFORE-CHECKOUT' then

Select

A.AttendanceID,A.UserID,

date(A.CheckInTime) as AttnDate,

Time(A.CheckInTime) as CheckInTime,

Time(A.CheckOutTime) CheckOutTime,

(TIMESTAMPDIFF(Second, CheckInTime, NOW())) TotalHours

from attendance A

where A.UserID = pUserID

and A.CheckInTime >= DATE\_SUB(CAST(NOW() as DATETIME), INTERVAL 1 DAY) and A.CheckInTime <= CAST(NOW() as DATETIME)

order by A.CheckInTime desc limit 0,1;

end if;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `nex\_UserShifts` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `nex\_UserShifts`(

in pUserID int,

in porgID int

)

BEGIN

select S.ShiftTitle,S.Description,S.StartTime,S.EndTime,S.isActive,US.EffectedFrom

from usershift US

LEFT JOIN shift S ON US.ShiftID = S.ShiftID

where US.UserID = pUserID and S.OrgID = porgID

order by US.EffectedFrom desc;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `splitter` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `splitter`(x TEXT, delim varchar(12))

BEGIN

SET @Valcount = substrCount(x,delim)+1;

SET @v1=0;

drop table if exists splitResultsTmpTable;

create temporary

table splitResultsTmpTable (TmpID INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,SplitValue varchar(255));

WHILE (@v1 < @Valcount) DO

set @val = stringSplit(x,delim,@v1+1);

INSERT INTO splitResultsTmpTable (SplitValue) VALUES (@val);

SET @v1 = @v1 + 1;

END WHILE;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2018-08-03 11:06:23