



EMPLOYEE DATABASE MANAGEMENT SYSTEM

FINAL REPORT

Abhilash Kosaraju | DBMS sec-06 | 12/13/2017

|NEUID:001205393|

Problem Statement:

Employee Database is the most widely deployed database seen in every organization. The employee database helps the organization to manage the employees and assists HR with workforce management. The employee database plays a critical role in analyzing the employee data and helps the organization to make critical HR decisions. I wish to design the replica of the employee database which holds employee details and employee activities with the help of various techniques discussed in the classroom. The main objective of this project is to implement an employee information system in an organization to return accurate information about employee activities by executing SQL queries in MySQL database

PURPOSE:

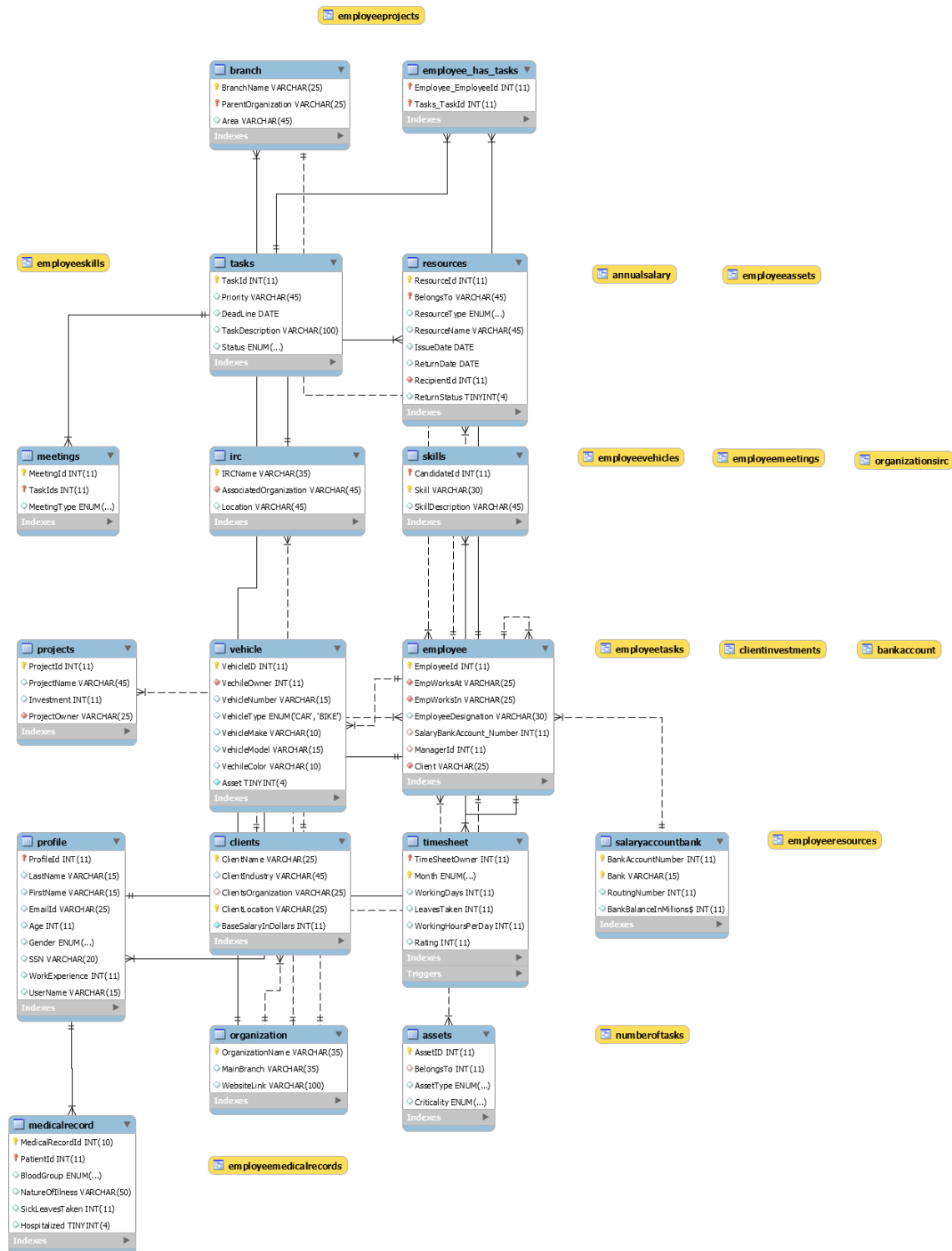
The proposed database design imitates the employee database as in maintaining employee profile, employee vehicle details, employee bank account details, employee medical records, employee skill sets and also manages the resources of organization's Information Resource Center(IRC).

TECHNIQUES:

Stored Procedures are employed to perform data retrieval operations from the database. Views are created to present the related data. Triggers are employed to perform some calculations. Joins were mostly used to concatenate relevant details in subqueries. Functions are used to return aggregate values for analysis.

Project demo:

EER Diagram

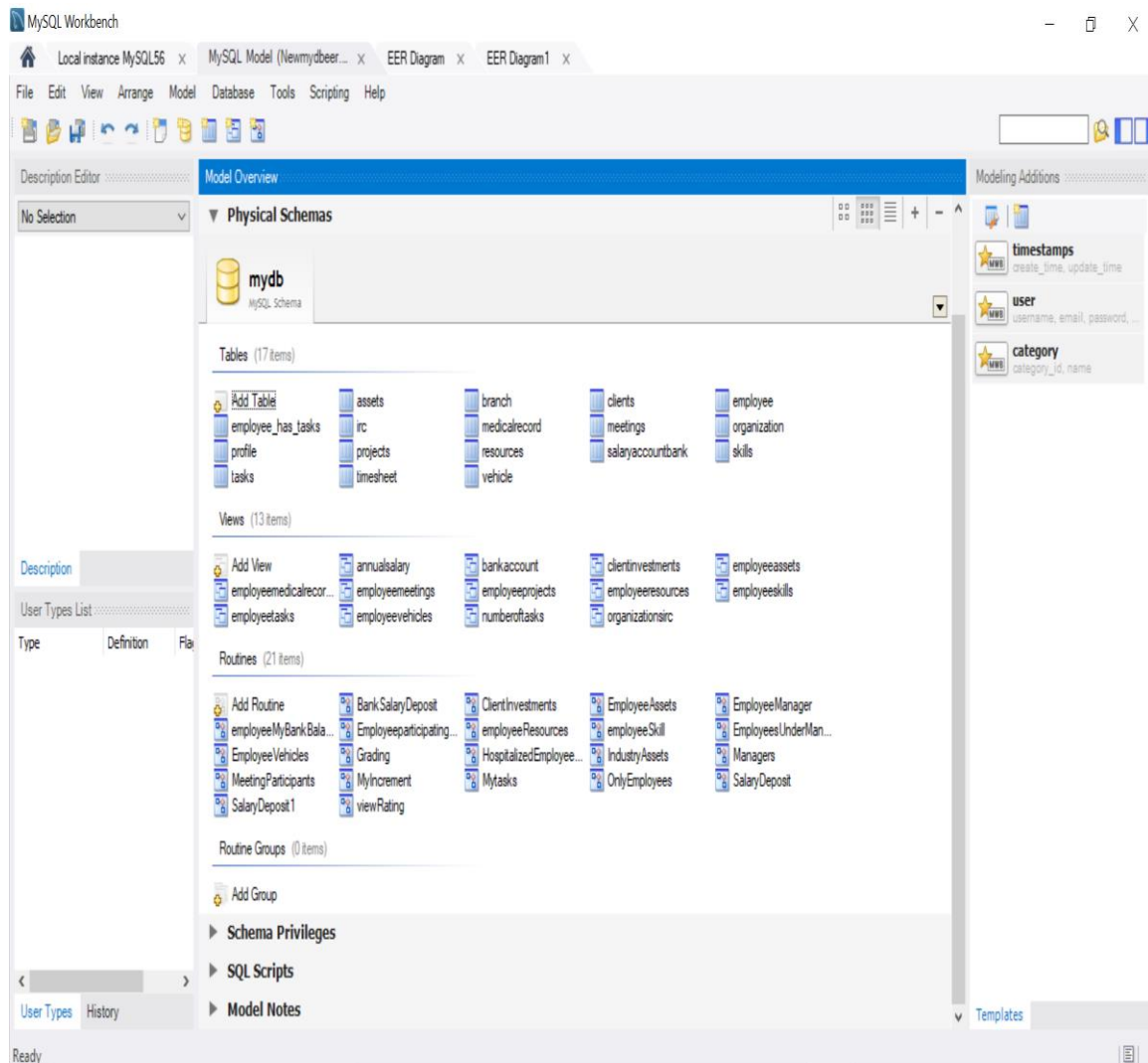


The Physical Schema

A total of 17 tables are defined and utilized in the database design.

13 views have been created to display the relevant data.

And 21 routines have been employed to retrieve the data and also to perform calculations for mathematical presentation of data.



Employee Data

MySQL Workbench interface showing the **annualsalary** table data. The query executed is `SELECT * FROM mydb.annualsalary;`. The result grid displays the following data:

TimeSheetOwner	Rating	ClientName	Organization	BaseSalaryInDollars	AnnualSalary
1401	4.1667	Beatty Group	Yambee	1200	5000.0000
1402	4.0000	Beier-Johns	Yambee	1800	7200.0000
1403	4.0000	Berner and Sons	Feednation	1550	6200.0000
1404	3.0000	Brekke Inc	Browsecat	1200	3600.0000
1405	4.0000	Lemke and Monahan	Feednation	1400	5600.0000
1406	4.0000	Lemke, Sauer and Wunsch	Browsecat	1250	5000.0000
1407	4.0000	Lubowitz-O'Reilly	Yambee	1200	4800.0000
1408	5.0000	Monahan-Glover	Feednation	1700	8500.0000
1409	4.0000	Olson-Nienow	Browsecat	1500	6000.0000
1410	4.0000	Schwalter and Mitchell	Demimbu	1700	6800.0000

AnnualSalary will be calculated from the base salary defined in the client and the rating for the employee calculated in the Timesheet.

MySQL Workbench interface showing the **timesheet** table data. The query executed is `SELECT * FROM mydb.timesheet;`. The result grid displays the following data:

TimeSheetOwner	Month	WorkingDays	LeavesTaken	WorkingHoursPerDay	Rating
1401	JAN	22	0	8	5
1402	JAN	22	1	8	4
1401	FEB	22	1	8	4
1402	FEB	22	2	8	4
1401	MAR	22	4	8	4
1402	MAR	22	3	8	4
1403	MAR	22	1	8	4
1401	APR	22	3	8	4
1402	APR	22	4	8	4
1404	APR	22	4	8	3
1401	MAY	22	4	8	4
1402	MAY	22	4	8	4
1405	MAY	22	4	8	4
1401	JUN	22	4	8	4
1402	JUN	22	4	8	4
1406	JUN	22	4	8	4
1401	JUL	22	4	8	4
1402	JUL	22	6	8	4
1407	JUL	22	7	8	4
1401	AUG	22	3	8	4
1402	AUG	22	3	8	4
1408	AUG	22	0	8	5
1401	SEP	22	1	8	4
1402	SEP	22	4	8	4
1409	SEP	22	5	8	4
1401	OCT	22	1	8	4

Trigger For calculating the Rating of the Employee based on the number of days an employee worked and leaves taken.

```
CREATE DEFINER='root'@'localhost' TRIGGER
`mydb`.`timesheet_BEFORE_UPDATE` BEFORE UPDATE ON `timesheet` FOR EACH
ROW

BEGIN

If ((new.WorkingHoursPerDay - new.LeavesTaken)*(new.workingdays) = 176) then
set new.Rating = 5;

elseif (160< (new.WorkingHoursPerDay- new.LeavesTaken)*(new.workingdays) < 176)
then set new.Rating =4;

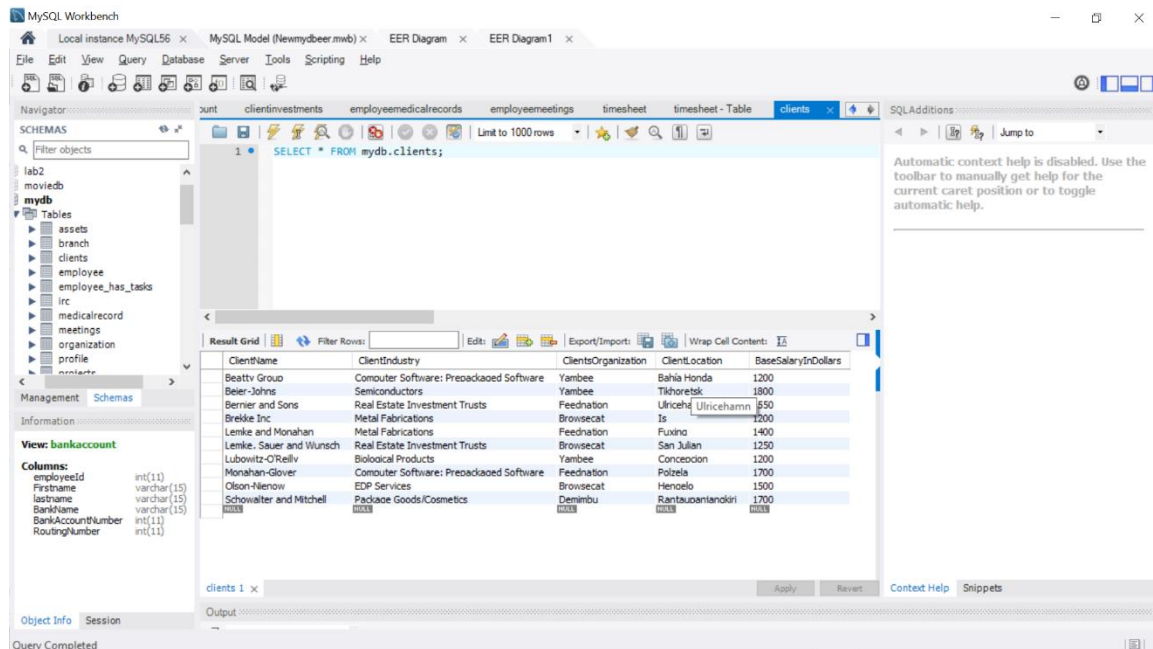
elseif (144 < (new.WorkingHoursPerDay- new.LeavesTaken)*(new.workingdays) < 160)
then set new.Rating=3;

elseif ( 128 <(new.WorkingHoursPerDay- new.LeavesTaken)*(new.workingdays) < 144)
then set new.Rating=2;

else

set new. Rating=1;

end if;
```



Client Investments in the Organization

The screenshot shows the MySQL Workbench interface with the 'clientinvestments' table selected in the Navigator. The SQL editor contains the query: `SELECT * FROM mydb.clientinvestments;`. The Result Grid displays the following data:

Organization	Clientname	Projectid	Project	Industry	Investment
Browsecat	Lenke, Sauer and Wunsch	10	Juliett	Real Estate Investment Trusts	12000
Browsecat	Brekke Inc	8	Delta	Metal Fabrications	40000
Browsecat	Olson-Nienow	11	Kilo	EDP Services	10000
Feednathon	Bernier and Sons	7	Echo	Real Estate Investment Trusts	30000
Feednathon	Monahan-Glover	12	Lima	Computer Software: Prepackaged Software	17000
Feednathon	Lenke and Monahan	9	Hotel	Metal Fabrications	15000
Yambee	Beatty Group	1	Aloha	Computer Software: Prepackaged Software	50000
Yambee	Beier-Johns	5	Whisky	Semiconductors	65000
Yambee	Beatty Group	2	Beta	Computer Software: Prepackaged Software	40000
Yambee	Lubowitz-O'Reilly	6	Foxtrot	Biological Products	20000
Yambee	Beatty Group	3	Gamma	Computer Software: Prepackaged Software	70000
Yambee	Beier-Johns	4	Charlie	Semiconductors	10000

Employee Medical Records

The screenshot shows the MySQL Workbench interface with the 'employeemedicalrecords' table selected in the Navigator. The SQL editor contains the query: `SELECT * FROM mydb.employeemedicalrecords;`. The Result Grid displays the following data:

employeeid	FirstName	LastName	BloodGroup	NatureOfIllness	SickLeavesTaken	Hospitalized
1401	Gaven	Almak	A+	Flu Relief Therapov Day Time	1	1
1401	Gaven	Almak	A+	Fluconazole	2	0
1401	Gaven	Almak	A+	rifamoin	10	1
1402	Cvnde	Radband	B+	Fluconazole	2	0
1402	Cvnde	Radband	B+	Alcohol Prep	6	0
1402	Cvnde	Radband	B+	Alcohol Prep	7	1
1403	Courtnav	Deneve	AB+	FENTANYL TRANSDERMAL SYSTEM	3	0
1403	Courtnav	Deneve	AB+	Ximino	3	1
1404	Viroe	Anberoeer	O+	Ximino	4	1
1404	Viroe	Anberoeer	O+	Buxom Show Some Skin Weichtless	4	1
1405	Michaela	Beraft	A-	Ver	5	0
1406	Kellvann	Stilmann	B-	Alcohol Prep	6	0
1407	Vittorio	Ruoert	AB-	Clean and Clear Essentials Deep Cl...	7	1
1408	Carminie	Rummins	AB+	LBL COULEUR LUXE AMPLIFIER XP	8	1
1409	Cv	Carberrv	O+	rifamoin	9	0
1410	Edee	Burnham	A-	Buxom Show Some Skin Weichtless	10	0

Employee Meetings

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists various databases, with 'employeeassets' selected. The 'Query' pane in the center contains the SQL query: `SELECT * FROM mydb.employeeemedicalrecords;`. The 'Result Grid' pane on the right displays the query results in a table format. The table has columns: employeeId, FirstName, LastName, BloodGroup, NatureOfIllness, SickLeavesTaken, and Hospitalized. The results show 10 rows of data for employees with IDs 1401 through 1410.

employeeId	FirstName	LastName	BloodGroup	NatureOfIllness	SickLeavesTaken	Hospitalized
1401	Gaven	Almak	A+	Flu Relief Theraov Day Time	1	1
1401	Gaven	Almak	A+	Fluconazole	2	0
1401	Gaven	Almak	A+	rifampin	10	1
1402	Cvnde	Radband	B+	Fluconazole	2	0
1402	Cvnde	Radband	B+	Alcohol Preo	6	0
1402	Cvnde	Radband	B+	Alcohol Preo	7	1
1403	Courtnav	Deneve	AB+	FENTANYL TRANSDERMAL SYSTEM	3	0
1403	Courtnav	Deneve	AB+	Ximino	3	1
1404	Virgie	Amberoe	O+	Ximino	4	1
1404	Virgie	Amberoe	O+	Buxom Show Some Skin Weightless	4	1
1405	Michaela	Becraft	A-	Ver	5	0
1406	Kellvann	Stilmann	B-	Alcohol Preo	6	0
1407	Vittorio	Ruoert	AB-	Clean and Clear Essentials Deeo Cl...	7	1
1408	Carminne	Runnins	AB+	LBEL COULEUR LUXE AMPLIFIER XP	8	1
1409	Cv	Carberv	O+	rifampin	9	0
1410	Edee	Burnham	A-	Buxom Show Some Skin Weightless	10	0

Employee Projects

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists various databases, with 'employeeassets' selected. The 'Query' pane in the center contains the SQL query: `SELECT * FROM mydb.employeeemeetings;`. The 'Result Grid' pane on the right displays the query results in a table format. The table has columns: employeeId, FirstName, LastName, Deadline, TaskDescription, Priority, and MeetingId. The results show 10 rows of data for employees with IDs 1401 through 1403.

employeeId	FirstName	LastName	Deadline	TaskDescription	Priority	MeetingId
1401	Gaven	Almak	2017-05-19	Opiate antagonists causing adverse effects in t...	HIGH	31
1401	Gaven	Almak	2017-09-20	Unspecified agent primarily affecting skin.	LOW	32
1401	Gaven	Almak	2017-10-20	Benign neoplasm of tonsil	MEDIUM	33
1401	Gaven	Almak	2017-11-20	Illegally induced abortion, complicated by emboli...	HIGH	34
1401	Gaven	Almak	2017-12-20	"Light-for-dates" with signs of fetal malnutrition...	LOW	35
1401	Gaven	Almak	2018-01-23	Macroductilia of toes	LOW	36
1402	Cvnde	Radband	2017-09-20	Unspecified agent primarily affecting skin.	LOW	32
1402	Cvnde	Radband	2017-12-20	"Light-for-dates" with signs of fetal malnutrition...	LOW	35
1402	Cvnde	Radband	2018-01-23	Macroductilia of toes	LOW	36
1402	Cvnde	Radband	2018-02-21	Vaginal hematoma	MEDIUM	37
1403	Courtnav	Deneve	2018-03-24	Puerperal septic thromboembolitis, unspecified a...	LOW	38
1403	Courtnav	Deneve	2017-10-20	Benign neoplasm of tonsil	MEDIUM	33
1403	Courtnav	Deneve	2017-11-20	Illegally induced abortion, complicated by emboli...	HIGH	34
1403	Courtnav	Deneve	2018-03-24	Puerperal septic thromboembolitis, unspecified a...	LOW	38
1403	Courtnav	Deneve	2018-04-15	Stage I necrotizing enterocolitis in newborn	MEDIUM	39
1403	Courtnav	Deneve	2018-05-28	Special screening for traumatic brain injury	HIGH	40

Employee's Resources from Information Resource Center associated with Organization

The screenshot shows the MySQL Workbench interface with the 'employeeresources' table selected in the 'Schemas' pane. The 'Query' pane contains the SQL statement: `SELECT * FROM mydb.employeeresources;`. The 'Result Grid' displays the following data:

employeeId	Firstname	lastname	ResourceName	ResourceType	Owner	Return
1408	Carmine	Rummins	Escalade, Incorporated	BOOK	O'Kon-Brakus	2017-0
1407	Vittorio	Ruoert	PFSweb, Inc.	RESEARCH MATERIAL	Oudlev Grouo	2017-1
1409	Cv	Carberv	TCG BDC, Inc.	EMATERIAL	Maver: Prohaska and Hamill	2017-0
1401	Gaven	Almak	National Research Corporation	RESEARCH MATERIAL	Swift, Schowalter and Mills	2016-0
1401	Gaven	Almak	BlackRock Income Trust Inc. (The)	JOURNAL	Marks, Connellv and Beronaum	2017-0
1401	Gaven	Almak	Medlev LLC	EMATERIAL	Moen LLC	2017-1
1403	Courtnav	Deneve	VSE Corporation	BOOK	Mitchell-Beer	2017-0
1406	Kellvann	Stillmann	Campoio World Holdings, Inc.	BOOK	Reinoer and Sons	2017-1
1404	Vroie	Amberoeer	Kinastone Companies, Inc	EMATERIAL	Littel, Little and Stamm	2017-1
1410	Edee	Burnham	First Trust Dorsey Wright Focus 5 ETF	JOURNAL	Cronin-Yost	2018-0

Employee Assets:

The screenshot shows the MySQL Workbench interface with the 'employeeassets' table selected in the 'Schemas' pane. The 'Query' pane contains the SQL statement: `SELECT * FROM mydb.employeeassets;`. The 'Result Grid' displays the following data:

employeeId	Firstname	lastname	AssetId	AssetType	Criticality
1401	Gaven	Almak	1	VEHICLE	HIGH
1401	Gaven	Almak	2	LAPTOP	LOW
1401	Gaven	Almak	7	VPN	HIGH
1402	Cvnde	Radband	3	LAPTOP	HIGH
1402	Cvnde	Radband	4	VEHICLE	LOW
1403	Courtnav	Deneve	6	VEHICLE	LOW
1404	Vroie	Amberoeer	8	LAPTOP	MEDIUM
1404	Vroie	Amberoeer	9	VEHICLE	HIGH
1405	Michaela	Becraft	5	VEHICLE	HIGH
1405	Michaela	Becraft	10	LAPTOP	MEDIUM
1407	Vittorio	Ruoert	11	VEHICLE	MEDIUM
1408	Carmine	Rummins	12	VEHICLE	HIGH
1409	Cv	Carberv	13	VEHICLE	MEDIUM
1410	Edee	Burnham	14	VEHICLE	HIGH

Employee Skills and WorkExperience

The screenshot shows the MySQL Workbench interface with the 'employeeskills' table selected in the 'Tasks' tab. The table structure is as follows:

employeeid	Firstname	lastname	skill	SkillDescription	Experience
1401	Gaven	Almak	CSRs	MSDS	1
1401	Gaven	Almak	Cycle Counting	High Sense Of Urgency	1
1401	Gaven	Almak	EBP	WebLogic	1
1401	Gaven	Almak	Health Insurance	Grievances	1
1402	Cynide	Radband	EBP	WebLogic	2
1403	Courtnav	Deneve	Immunoloov	Usability Engineering	3
1404	Vroie	Amberoeer	NHS Commissioning	PeopleSoft	4
1405	Michaela	Betract	CSRs	MSDS	5
1406	Kellvann	Stillnam	Cycle Counting	High Sense Of Urgency	6
1407	Vittorio	Ruoport	Duty Drawback	Real Estate License	7
1408	Carmine	Rummins	LTSP	MCEV	8
1409	Cv	Carbervv	Award Ceremonies	Biblical Studies	9
1410	Edee	Burnham	E-commerce	Air Force	10

Employee's Tasks

The screenshot shows the MySQL Workbench interface with the 'employeetasks' table selected in the 'Tasks' tab. The table structure is as follows:

EmployeeId	EmpWorksAt	EmployeeDesignation	ParentOrganization	TaskDescription	Priority
1401	Stodholm	Data Coordinator	Demibu	Oxiate antaonists causing adverse effects in t...	HIGH
1401	Stodholm	Data Coordinator	Demibu	Unspecified agent primarily affecting skin.	LOW
1401	Stodholm	Data Coordinator	Demibu	Benion neoclasia of tonsil	MEDIUM
1401	Stodholm	Data Coordinator	Demibu	Ileally induced abortion, complicated by emboli...	HIGH
1401	Stodholm	Data Coordinator	Demibu	"Light-for-dates" with signs of fetal malnutrition...	LOW
1402	Albertville	Programmer Analyst III	Gabtvoo	Unspecified agent primarily affecting skin.	LOW
1402	Albertville	Programmer Analyst III	Gabtvoo	"Light-for-dates" with signs of fetal malnutrition...	LOW
1402	Albertville	Programmer Analyst III	Gabtvoo	Macroductilia of toes	LOW
1402	Albertville	Programmer Analyst III	Gabtvoo	Vaginal hematoma	MEDIUM
1402	Albertville	Programmer Analyst III	Gabtvoo	Pueroral septic thrombophlebitis, unspecified a...	LOW
1402	Albertville	Programmer Analyst III	Ouire	Unspecified agent primarily affecting skin.	LOW
1402	Albertville	Programmer Analyst III	Ouire	"Light-for-dates" with signs of fetal malnutrition...	LOW
1402	Albertville	Programmer Analyst III	Ouire	Macroductilia of toes	LOW
1402	Albertville	Programmer Analyst III	Ouire	Vaginal hematoma	MEDIUM
1402	Albertville	Programmer Analyst III	Ouire	Pueroral septic thrombophlebitis, unspecified a...	LOW
1403	Sever do Vo...	Food Chemist	Realcube	Oxiate antaonists causing adverse effects in t...	HIGH
1403	Sever do Vo...	Food Chemist	Realcube	Unspecified agent primarily affecting skin.	LOW
1403	Sever do Vo...	Food Chemist	Realcube	Benion neoclasia of tonsil	MEDIUM
1403	Sever do Vo...	Food Chemist	Realcube	Ileally induced abortion, complicated by emboli...	HIGH
1403	Sever do Vo...	Food Chemist	Realcube	Pueroral septic thrombophlebitis, unspecified a...	LOW
1403	Sever do Vo...	Food Chemist	Realcube	Stage I necrotizing enterocolitis in newborn	MEDIUM
1403	Sever do Vo...	Food Chemist	Realcube	Social screening for traumatic brain injury	HIGH
1404	Lakeland	Geologist I	Feednation	Ileally induced abortion, complicated by emboli...	HIGH

Employee Vehicles

MySQL Workbench interface showing the **employeevehicles** table data. The query executed is `SELECT * FROM mydb.employeevehicles;`. The result grid displays 10 rows of data with columns: **employeeId**, **Firstname**, **lastname**, **VehicleType**, **VehicleNumber**, **VehicleMake**, **VehicleModel**, **VehicleColor**, and **CompanyAsset**.

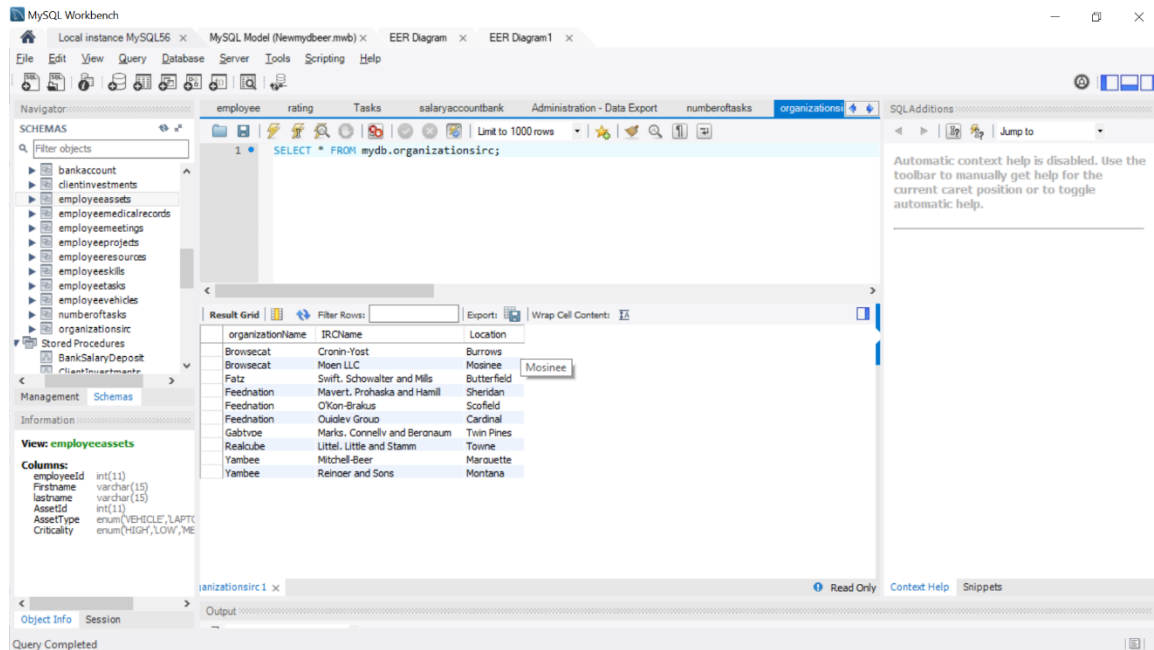
employeeId	Firstname	lastname	VehicleType	VehicleNumber	VehicleMake	VehicleModel	VehicleColor	CompanyAsset
1408	Carline	Runnins	BIKE	79-240-3290	Chrysler	Reatta	Teal	1
1405	Michaela	Bezaft	BIKE	05-656-8046	Tovota	SVX	Puce	1
1407	Vittorio	Ruoert	CAR	28-849-4567	Infiniti	Prizm	Red	1
1409	Cv	Carberrv	CAR	55-286-9252	BMW	R-Class	Red	1
1402	Cvnde	Radband	BIKE	19-974-2600	Cadillac	Ram 1500	Orange	0
1401	Gawen	Almak	CAR	10-857-8826	Ford	Grand Caravan	Green	1
1403	Courthav	Deneve	BIKE	34-256-3982	Mazda	E350	Green	0
1406	Kelviann	Stilmann	CAR	72-238-4202	Acura	Jetta	Aquamarine	0
1404	Virde	Amberoe	CAR	99-052-4524	Bentlev	R8	Pink	1
1410	Edee	Burnham	BIKE	80-560-5026	Buick	Panamera	Pink	0

Number of Employee Tasks grouped by EmployeeId

MySQL Workbench interface showing the **numberoftasks** table data. The query executed is `SELECT * FROM mydb.numberoftasks;`. The result grid displays 10 rows of data with columns: **EmployeeId** and **NumberofTasks**.

EmployeeId	NumberofTasks
1401	6
1402	10
1403	7
1404	2
1405	2
1406	2
1407	2
1408	2
1409	2
1410	2

Information Resource Centers associated with Organizations



STORED PROCEDURES

BankSalaryDeposit

Deposits the incremented salary in the employees SalaryBankAccount

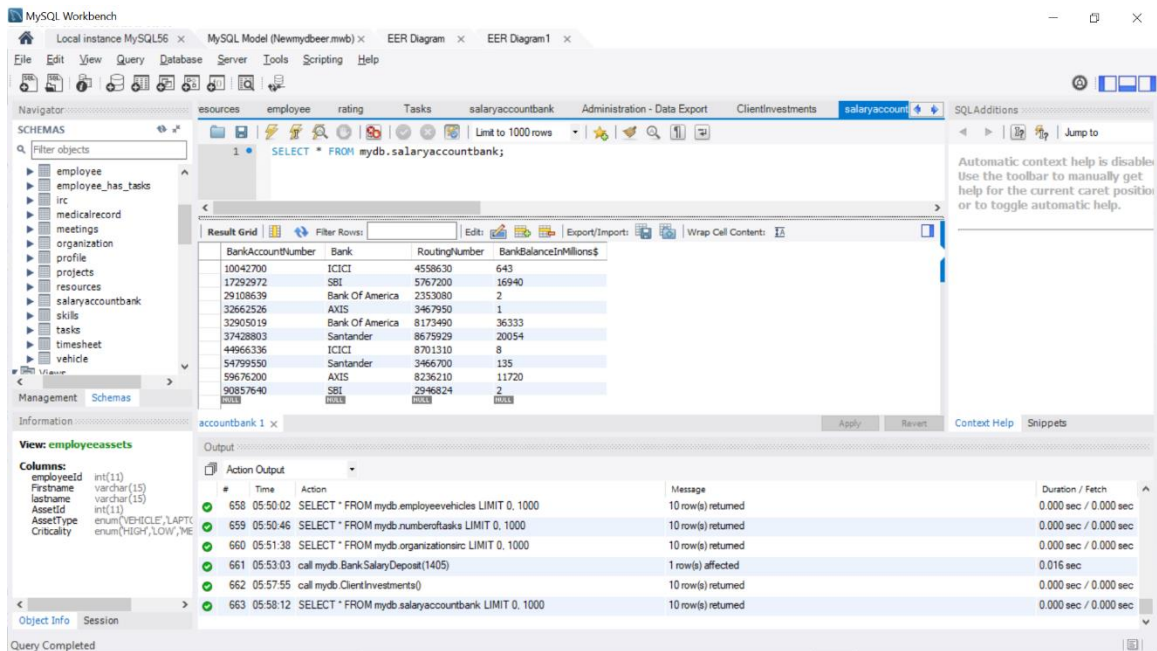
```
CREATE DEFINER='root'@'localhost' PROCEDURE `BankSalaryDeposit`(IN id int)
```

```
BEGIN
```

```
update salaryaccountbank set BankBalanceInMillions$ = BankBalanceInMillions$ +  
SalaryDeposit1(id) where
```

```
BankAccountNumber=(select SalaryBankAccount_Number from employee where  
EmployeeId =id);
```

```
END
```



Client investments in Organization

CREATE DEFINER='root'@'localhost' PROCEDURE `ClientInvestments`()

BEGIN

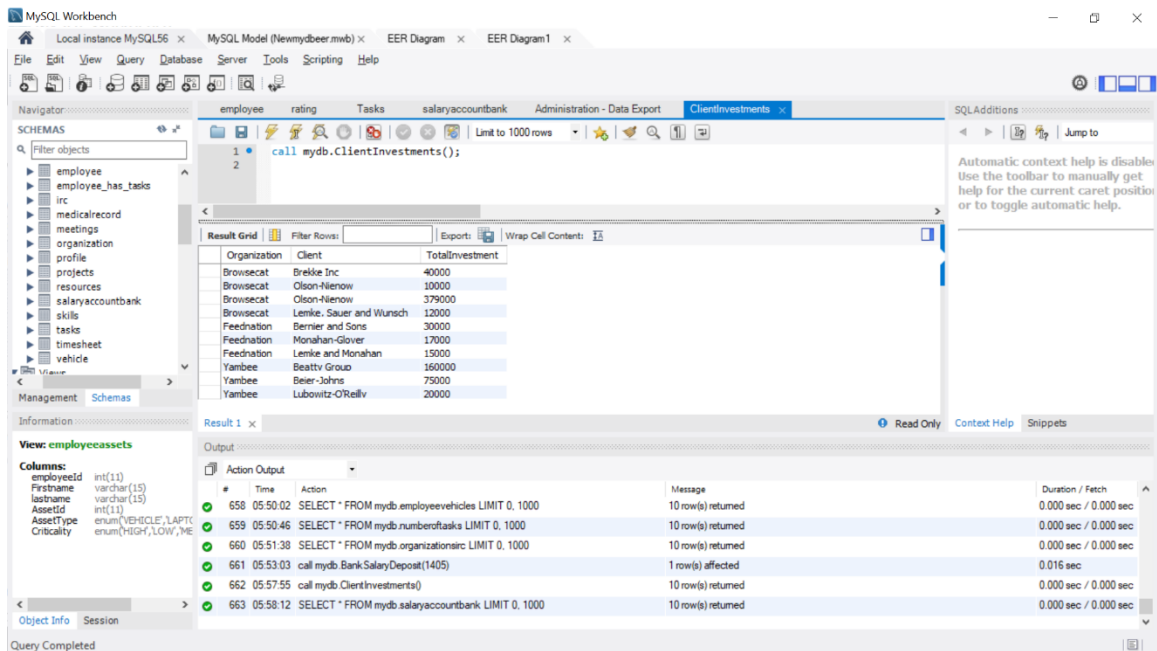
select IFNULL(Organization, 'Org') AS Organization,

IFNULL(Clientname, 'Total') As Client, sum(investment) As TotalInvestment

from clientinvestments

Group by ClientName with rollup;

END



EmployeeManager

Employee's Manager details will be displayed

CREATE DEFINER='root'@'localhost' PROCEDURE `EmployeeManager` (IN id int)

BEGIN

select * from employee

inner join

profile

on employee.EmployeeId = ProfileId

where EmployeeId in (SELECT ManagerId

FROM employee

where EmployeeId= id);

END

EmployeeAssets

Retrieves all the assets of the employee.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `EmployeeAssets` (IN id int)
BEGIN
select employeeId,Firstname,lastname,AssetId,AssetType,Criticality
from employee
inner join profile
on employee.EmployeeId = profile.ProfileId
inner join assets
on employee.employeeId = assets.BelongsTo
where BelongsTo = id
order by EmployeeId;
END
```

EmployeeMyBankBalance

Gives the employee, the total amount in his bank account of the employee

```
CREATE DEFINER='root'@'localhost' PROCEDURE `employeeMyBankBalance` (IN id
int)
BEGIN
select employeeId,FirstName,lastname,b.BankAccountNumber as
AccountNumber,Bank,BankBalanceInMillions$ as mybalance
from bankaccount b
inner join
salaryaccountbank
on b.BankAccountNumber = salaryaccountbank.BankAccountNumber
where employeeId =id;
END
```

EmployeeparticipatingProjects

Presents all the working projects of the employees

```
CREATE DEFINER='root'@'localhost' PROCEDURE  
'EmployeeparticipatingProjects'(IN id int)
```

```
BEGIN
```

```
select * from EmployeeProjects
```

```
where employeeId =1401;
```

```
END
```

employeeResources

Displays all the resources with employee.

```
CREATE DEFINER='root'@'localhost' PROCEDURE 'employeeResources'(IN id int)
```

```
BEGIN
```

```
select employeeId,Firstname,lastname,ResourceName,ResourceType,BelongsTo As  
Owner,ReturnDate,ReturnStatus
```

```
from employee
```

```
inner join profile
```

```
on employee.EmployeeId = profile.ProfileId
```

```
inner join resources
```

```
on employee.employeeId = resources.recipientId
```

```
where RecipientId = id;
```

```
END
```

Employee Skills

Skills of the concerned Employee

```
CREATE DEFINER='root'@'localhost' PROCEDURE `employeeSkill` (IN id int)
BEGIN
select *
from employeeskills
where
employeeId=id;
END
```

EmployeesUnderManager

Displays all the employees under the employee(Manager)

```
CREATE DEFINER='root'@'localhost' PROCEDURE `EmployeesUnderManager` (IN id
int)
BEGIN
SELECT *
FROM employee inner join
profile
on employee.EmployeeId = profile.ProfileId
where ManagerId = 1401;
END
```

EmployeeVehicles

All the vehicles of the employee.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `EmployeeVehicles` (IN id int)
BEGIN
select
employeeId,Firstname,lastname,VehicleType,VehicleNumber,VehicleMake,VehicleModel,VechileColor, Asset as CompanyAsset
from employee
inner join profile
on employee.EmployeeId = profile.ProfileId
inner join vehicle
on employee.employeeId = vehicle.VechileOwner
where VechileOwner = id;
END
```

Grading

Assigns the grades to the employee based on the number of task the employee is working with.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `Grading` (IN id int, OUT Rating
varchar(5))
BEGIN
case
when (select numberoftasks.numberofTasks from numberoftasks where
employeeId=id ) =10
then set rating ='A';
when (select numberoftasks.numberofTasks from numberoftasks where
employeeId=id ) >=5 AND (select numberoftasks.numberofTasks from numberoftasks
where employeeId=id) <10
then set rating ='B';
```



```
else set rating ='C';  
end case;  
END
```

HospitalizedEmployeeMedicalRecords

Gives the hospitalized medical records of the employee

```
CREATE DEFINER=`root` @`localhost` PROCEDURE  
`HospitalizedEmployeeMedicalRecords`(IN id int)  
BEGIN  
select *  
from EmployeeMedicalRecords  
where Hospitalized = 1  
AND employeeId=id  
order by EmployeeId;  
  
END
```

IndustryAssets

Displays the stakes of Organization in different industries.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `IndustryAssets`()
```

```
BEGIN
```

```
select IFNULL(Organization, 'Org') AS Organization,
```

```
IFNULL(Industry,'Total') As ClientIndustry,
```

```
sum(investment) As TotalInvestment
```

```
from clientinvestments
```

```
Group by Industry,Organization with rollup;
```

```
END
```

MeetingParticipants

Displays the participants of the meeting when a meeting input is given

```
CREATE DEFINER='root'@'localhost' PROCEDURE `MeetingParticipants`(IN id int)
```

```
BEGIN
```

```
select employeeId,FirstName,Lastname,Priority as
```

```
TaskPriority,MeetingType,DeadLine as TaskDeadline
```

```
from employeemeetings
```

```
where meetingId =id;
```

```
END
```

MyIncrement

Gives the amount of increment the employee has got, based on the grade he has earned.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `MyIncrement` (IN id int, OUT  
SalaryWithIncrement int)  
  
BEGIN  
  
case  
  
when (select numberoftasks from numberoftasks where employeeId=id ) =10  
  
then set SalaryWithIncrement = ((select annualsalary.AnnualSalary from annualsalary  
where annualsalary.TimeSheetOwner=id)*5);  
  
  
when (select numberoftasks.numberoftasks from numberoftasks where  
employeeId=id ) >=5 AND (select numberoftasks.numberoftasks from  
numberoftasks where employeeId=id) <10  
  
then set SalaryWithIncrement =((select annualsalary.AnnualSalary from annualsalary  
where annualsalary.TimeSheetOwner=id)*4);  
  
  
else set SalaryWithIncrement =((select annualsalary.AnnualSalary from annualsalary  
where annualsalary.TimeSheetOwner=id)*3);  
  
  
end case;  
  
END
```

Mytasks

Displays his tasks to the employee

```
CREATE DEFINER='root'@'localhost' PROCEDURE `Mytasks` (IN id int)
BEGIN
select * from employeeTasks
where
employeeid =id;
END
```

OnlyEmployees

Displays only the employees in the organization and not the managers

```
CREATE DEFINER='root'@'localhost' PROCEDURE `OnlyEmployees` ()
BEGIN
SELECT *
FROM employee inner join
profile
on employee.EmployeeId = profile.ProfileId
where ManagerId IS NOT NULL;
END
```

OnlyManagers

Displays only the managers

```
CREATE DEFINER='root'@'localhost' PROCEDURE `Managers`()  
  
BEGIN  
  
SELECT *  
  
FROM employee inner join  
profile  
on employee.EmployeeId = profile.ProfileId  
where ManagerId IS NULL;  
  
END
```

ViewRating

Average ratings for the employees would be shown in the descending order.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `viewRating`()  
  
BEGIN  
  
    SELECT TimesheetOwner as EmployeeId,rating,clientname,Organization FROM  
annualsalary  
  
    group by organization,clientname  
  
    order by rating desc;  
  
END
```


FUNCTION

SalaryDeposit

Returns the amount to deposited in the employee bank account. Used in the BankSalaryDeposit stored procedure.

```
CREATE DEFINER='root'@'localhost' FUNCTION `SalaryDeposit` (id int) RETURNS  
int(11)
```

```
BEGIN
```

```
DECLARE deposit int;
```

```
case
```

```
when (select numberoftasks from numberoftasks where employeeId=id ) =10
```

```
then set deposit = ((select annualsalary.AnnualSalary from annualsalary where  
annualsalary.TimeSheetOwner=id)*5);
```

```
when (select numberoftasks.numberoftasks from numberoftasks where  
employeeId=id ) >=5 AND (select numberoftasks.numberoftasks from  
numberoftasks where employeeId=id) <10
```

```
then set deposit =((select annualsalary.AnnualSalary from annualsalary where  
annualsalary.TimeSheetOwner=id)*4);
```

```
else set deposit =((select annualsalary.AnnualSalary from annualsalary where  
annualsalary.TimeSheetOwner=id)*3);
```

```
end case;
```

```
return deposit;
```

```
end
```

TO CONCLUDE:

In the above database design, I have tried to address all the significant functionalities of employee database to present the completed system.

