Vehicle Subscription Service

CSE311 Project

Team Members:

1. Md. Abu Ammar - 1821944642
2. Sadia Afrin Tamanna- 1812030042
3. Tamalika Bakshi- 1812469042

Contents

PROJECT DESCRIPTION:

As Bangladesh is a developing country, most of the people can not afford a personal vehicle. Again, some foreign visitors come to visit our country and sometimes they need vehicles for a certain period.It is also not meaningful to buy a vehicle with a very high percent of tax. Also, some buyers can not change their vehicles in the next time if they need, for not having enough money. Here, we bring the solution through "VEHICLE SUBSCRIPTION" to solve the problem. Here we will provide the desired vehicles of customers for a certain period with money pre-paid system. They will use the vehicles for that certain period as a temporary owner. One day before their period ends, they will get an alert notification through the system where they will get two options (1) extending the subscription period or (2) have to return the vehicle. We will manage this whole function with a database system.

DATABASE PLANNING:

A plan to collect requirements, analysis, design, implementation and testing of a database.

First, a client can choose a vehicle according to his/her desire. Then, they will be offered to choose any available subscription package so that he/she can enjoy the discount if wants. Onwards they will have to pay the subscription fee. Every transection of the subscriber will be recorded. Then, they can pick their vehicles from any of the warehouses. There will be our employees to serve the subscribers and maintain the vehicles. At the end of the subscription period, a subscriber have to return the vehicle or he/she can extend the period.

REQUIREMENTS COLLECTION:

The process of collection and analysing the requirements of the database by understanding the scope and boundaries of the database application and the major user views. User views refers to the perspective of a particular user usage.- Analysis involves finding WHAT and HOW!

In order to make our project we need some some softwares.We used “MySQL workbench” for running the queries and the same software for drawing the ER diagram.

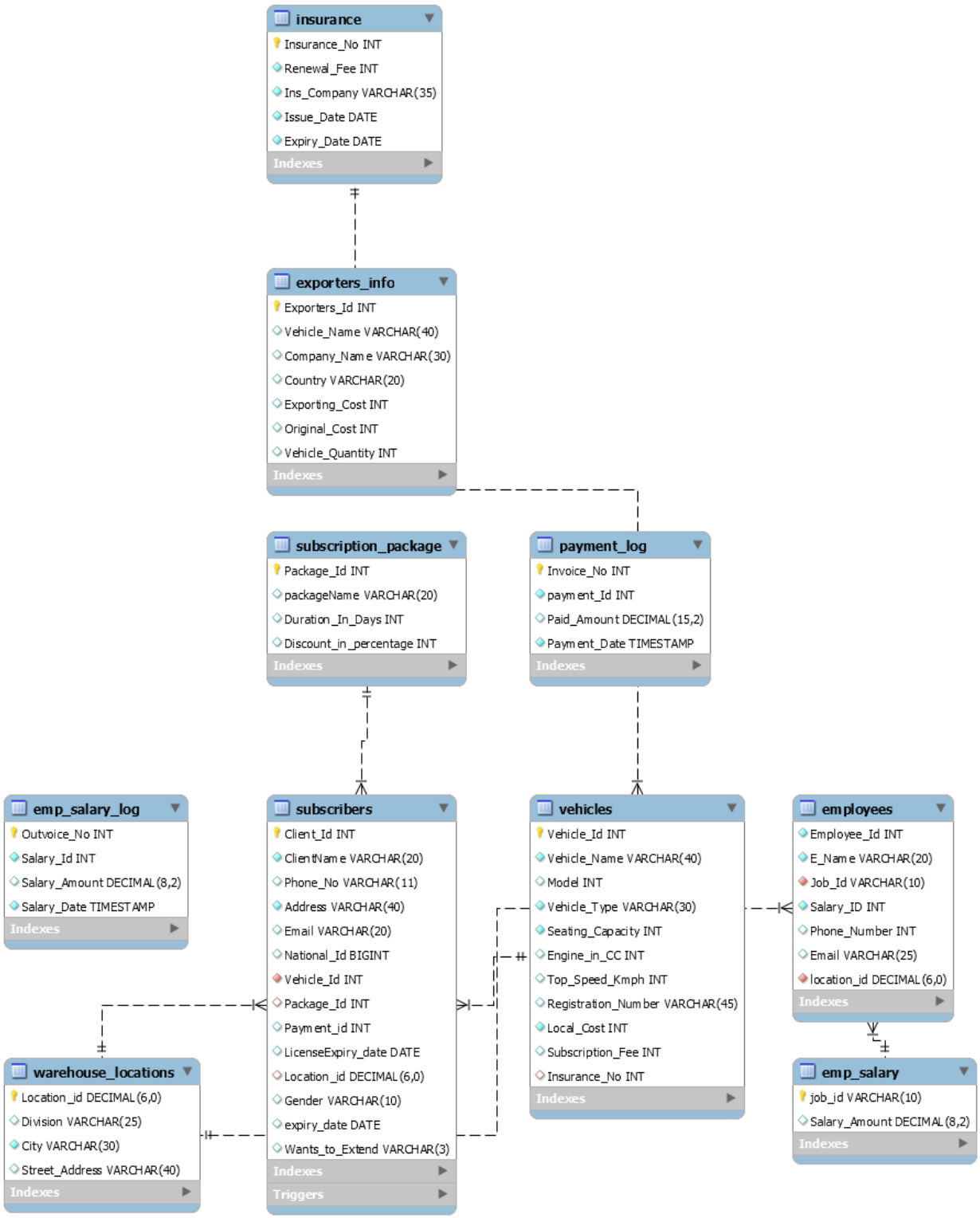
DATABASE DESIGN

The process of creating a design for a database that will support the enterprise operations and objectives.

In our project, we have 10 entities -

They are, (1) Vehicles, (2) Subscribers, (3) Subscription_Package, (4) Warehouse_Locations, (5) Employees, (6) Emp_Salary, (7) Emp_Salary_Log, (8) Exporters_Info, (9) Insurance and (10) Payment_Log.

ER DIAGRAM:


insurance
Insurance_No INT
Renewal_Fee INT
Ins_Company VARCHAR(35)
Issue_Date DATE
Expiry_Date DATE
Indexes
exporters_info
Exporters_Id INT
Vehicle_Name VARCHAR(40)
Company_Name VARCHAR(30)
Country VARCHAR(20)
Exporting_Cost INT
Original_Cost INT
Vehicle_Quantity INT
Indexes
subscription_package
Package_Id INT
packageName VARCHAR(20)
Duration_In_Days INT
Discount_in_percentage INT
Indexes
payment_log
Invoice_No INT
payment_Id INT
Paid_Amount DECIMAL(15,2)
Payment_Date TIMESTAMP
Indexes
emp_salary_log
Outvoice_No INT
Salary_Id INT
Salary_Amount DECIMAL(8,2)
Salary_Date TIMESTAMP
Indexes
subscribers
Client_Id INT
ClientName VARCHAR(20)
Phone_No VARCHAR(11)
Address VARCHAR(40)
Email VARCHAR(20)
National_Id BIGINT
Vehicle_Id INT
Package_Id INT
Payment_id INT
LicenseExpiry_date DATE
Location_id DECIMAL(6,0)
Gender VARCHAR(10)
expiry_date DATE
Wants_to_Extend VARCHAR(3)
Indexes
Triggers
vehicles
Vehicle_Id INT
Vehicle_Name VARCHAR(40)
Model INT
Vehicle_Type VARCHAR(30)
Seating_Capacity INT
Engine_in_CC INT
Top_Speed_Kmph INT
Registration_Number VARCHAR(45)
Local_Cost INT
Subscription_Fee INT
Insurance_No INT
Indexes
employees
Employee_Id INT
E_Name VARCHAR(20)
Job_Id VARCHAR(10)
Salary_ID INT
Phone_Number INT
Email VARCHAR(25)
location_id DECIMAL(6,0)
Indexes
warehouse_locations
Location_id DECIMAL(6,0)
Division VARCHAR(25)
City VARCHAR(30)
Street_Address VARCHAR(40)
Indexes
emp_salary
job_id VARCHAR(10)
Salary_Amount DECIMAL(8,2)
Indexes

DATABASE IMPLEMENTATION

Implementation is the process of coding /programming the actual software (here database)

- In the implementation stage, programmers code the database according to the design stage

- Creating Database

- Creating Schemas / Tables

Creating Schemas

```
create schema  Vehicle_Subscription;
```

Creating Tables

```
create table Exporters_Info
(
    Exporters_Id int (5),
    Vehicle_Name varchar (40),
    Company_Name varchar (30),
    Country varchar (20),
    Exporting_Cost int (10),
    Original_Cost int (10),
    Vehicle_Quantity int (5),
    primary key (Exporters_Id)
);
```

```
create table Vehicles
 (
    Vehicle_Id int not null primary key auto_increment,
    Vehicle_Name varchar(40) not null,
    Model int,
    Vehicle_Type varchar(30) not null,
    Seating_Capacity int not null,
    Engine_in_CC int,
    Top_Speed_Kmph int,
    Registration_Number varchar(45),
    Local_Cost int not null,
    Subscription_Fee int,
    Insurance_No int,
    foreign key(Insurance_No) references Insurance(Insurance_No)
);
```

```
create table Insurance
(
    Insurance_No int(20)  not null primary key,
    Renewal_Fee int(6) not null,
    Ins_Company varchar(35) not null,
    Issue_Date date not null,
    Expiry_Date date not null
    );
```

```
create table Subscribers
(
    Client_Id int primary key not null,
    ClientName varchar (20) not null,
    Phone_No varchar (11),
    Address varchar (40) not null,
    Email varchar (20),
    National_Id int8,
    Vehicle_Id int (5) not null,
    Package_Id int (5),
    Payment_id int (5),
    LicenseExpiry_date date,
    Location_id NUMERIC(6),
    Gender varchar (10),
    expiry_date date,
    Wants_to_Extend varchar(3),
    foreign key(Vehicle_Id) references Vehicles(Vehicle_Id),
    foreign key(Package_Id) references Subscription_Package(Package_Id),
    foreign key(Location_Id) references warehouse_locations(Location_Id)
);
```

```
create table Emp_Salary
(
job_id varchar(10) primary key  not null,
Salary_Amount numeric(8,2)
);
```

```
create table Subscription_Package
( Package_Id int,
    packageName varchar (20),
    Duration_In_Days int (10),
    Discount_in_percentage int (5),
    primary key (Package_Id));
```

```
create table Payment_Log
(
Invoice_No int primary key not null auto_increment,
payment_Id int not null,
Paid_Amount decimal(15,2),
Payment_Date timestamp not null default now()
);
```

```
CREATE TABLE Warehouse_Locations
(
Location_id NUMERIC(6) PRIMARY KEY not null,
Division VARCHAR(25),
City VARCHAR(30) NOT NULL,
Street_Address VARCHAR(40));
```

```
CREATE TABLE Employees(
Employee_Id int not null,
E_Name VARCHAR(20) not null,
Job_Id VARCHAR(10) NOT NULL,
Salary_ID int NOT NULL ,
Phone_Number INT(11),
Email VARCHAR(25),
location_id NUMERIC(6)NOT NULL,
Foreign key(job_Id) references emp_salary(job_id),
Foreign key(location_id) references Warehouse_Locations(location_id)
);
```

```
create table Emp_Salary_Log
(
Outvoice_No int primary key not null auto_increment,
Salary_Id int not null,
Salary_Amount decimal(8,2),
Salary_Date timestamp  not null default now()
);
```

Inserting Data Into DB

```
insert into Exporters_Info (Exporters_Id,Vehicle_Name, Company_Name, Country, Exporting_Cost, Original_Cost, Vehicle_Quantity)
values (701, "BMW X1", "BMW Company","Germany",60000,12000000,3),
       (702,"Audi Q2", "Audi Company","Germany",53000,11250000,3)
```

```
insert into Vehicles( Vehicle_Name, Model, Vehicle_Type, Seating_Capacity, Engine_in_CC, Top_Speed_Kmph, Registration_Number, Local_Cost, Subscription_Fee, Insurance_No)
            values( "BMW X1", 2020, "SportX", 5, 3880, 300, "Dhaka Metro-Mo-0001", 8000000, 60000, 111)
```

```
insert into Insurance( Insurance_No, Renewal_Fee, Ins_Company, Issue_Date, Expiry_Date)
values( 111, 40000, "ABC Insurance Ltd", str_to_date("2021-Jan-01", "%Y-%b-%d"), str_to_date("2022-Jan-01", "%Y-%b-%d"))
```

```
insert into Subscribers (Client_Id,ClientName, Phone_No, Address, Email,
National_Id , Vehicle_Id, Package_Id, Payment_id,
LicenseExpiry_date,Gender,Location_Id,expiry_date, Wants_to_Extend)

values  (201,"Mahbub","11111111111", "Dhamrai", "mahbub@gmail.com",
95047100000,  2, 101, 301, str_to_date('10-Jan-25','%d-%b-%y'), "Male",1006,
str_to_date('13-Mar-21','%d-%b-%y'),"Yes")
```

```
insert into subscription_package(Package_Id, packageName, Duration_In_Days,
Discount_in_percentage)

values  (101, "Championship", 60, 5)
```

```
insert into warehouse_locations (Location_id,Division,City,Street_Address)

values(1001,'DHAKA','GAZIPUR','Bipsot training Area')
```

```
insert into
employees(Employee_Id,E_Name,Job_Id,Salary_ID,Phone_Number,Email,Location_id)

values(551,'KALAM','VEH_ASSO',881,0154298745,'klm@mail.com',1005)
```

```
insert into emp_salary( Job_id, Salary_Amount)

values('VEH_ASSO',27000)
```

DATABASE TESTING

Testing is the process of executing the application programs and database with the intent of finding error.

- Test the database

- Test the application program/s

- Test the connectivity between database and application programs

Testing Functions To Reach Our Goal

We can give the salary to any of our employee by the following function and its also will be automatically recorded in our database(Used Function):

```
CREATE DEFINER=root@localhost FUNCTION Pay_Salary(Salary_Id int,
Salary_Amount decimal(8,2)) RETURNS text CHARSET utf8mb4
    DETERMINISTIC
BEGIN
insert into Emp_Salary_Log(Salary_Id, Salary_Amount)
values (Salary_Id, Salary_Amount);
RETURN "Salary Paid ;)" ;
END
```

At the end of the subscription period, subscribers are geting two options (1) extending the subscription period or (2) have to return the vehicle:

```
CREATE DEFINER=root@localhost FUNCTION
Extend_Subscription_Period_or_return_vehicles(cid int, pid int, WTE
varchar(3)) RETURNS text CHARSET utf8mb4
    DETERMINISTIC
```

```
BEGIN

if  WTE = "Yes" and pid is not null then

update subscribers

 set Package_id=pid, expiry_date = DATE_ADD(CURDATE(), Interval (select
duration_in_days

 from
subscription_package

   where
Package_Id= pid) Day), Wants_to_Extend = wte

 where expiry_date = CURDATE() and client_id = cid;

 RETURN "extended_with_package";


elseif  WTE ="Yes" and pid is null then

update subscribers

 set Package_id= null, expiry_date = DATE_ADD(curdate(), Interval 30 Day),
Wants_to_Extend = "Yes"

 where expiry_date = CURDATE()  and client_id = cid;

 RETURN "extended_without_package";


 elseif wte="No" then

 delete from subscribers

 where client_id = cid;


RETURN "period_ends";

end if;

END
```

Testing Triggers To Reach Our Goal

When any subscriber subscribed any vehicle his transaction info will be automatically recorded (Used “After Insert” Trigger):

```
CREATE DEFINER=root@localhost TRIGGER subscribers_AFTER_INSERT AFTER INSERT
ON subscribers FOR EACH ROW BEGIN
```

```
if new.package_Id is not null then
    insert into payment_log(payment_id,paid_amount)
    values (new.payment_Id,(select v.Subscription_Fee-
v.Subscription_Fee*(p.Discount_in_percentage/100)
            from subscribers s
            join vehicles v
            on s.vehicle_id=v.vehicle_id
            join subscription_package p
            on s.Package_Id=p.Package_Id
            where s.client_id = new.client_id
            ));
else insert into payment_log(payment_id,paid_amount)
     values (new.payment_Id,(select v.Subscription_Fee
                            from subscribers s
                            join vehicles v
                            on s.vehicle_id=v.vehicle_id
                            where s.client_id = new.client_id)

            );
end if;
END
```

When any customer renew his/her subscription period then their new transaction will also be recorded automatically by “after update” trigger:

```
CREATE DEFINER=root@localhost TRIGGER subscribers_AFTER_UPDATE AFTER UPDATE
ON subscribers FOR EACH ROW BEGIN
if new.package_Id is not null then
    insert into payment_log(payment_id,paid_amount)
```

```
    values (new.payment_Id,(select v.Subscription_Fee-
v.Subscription_Fee*(p.Discount_in_percentage/100)
            from subscribers s
            join vehicles v
            on s.vehicle_id=v.vehicle_id
            join subscription_package p
            on s.Package_Id=p.Package_Id
            where s.client_id = new.client_id
            ));
else insert into payment_log(payment_id,paid_amount)
     values (new.payment_Id,(select v.Subscription_Fee
                            from subscribers s
                            join vehicles v
                            on s.vehicle_id=v.vehicle_id
                            where s.client_id = new.client_id)

            );
end if;
END
```

Testing Queries To Reach Our Goal

Total Unit of Same Vehicles:

```
select count(vehicle_name) Unit, Vehicle_name
from vehicles
group by Vehicle_Name
order by count(vehicle_name) desc;
```

Total Unit of Same Vehicle Type:

```
select count(vehicle_type) "Unit", Vehicle_type

from vehicles

group by Vehicle_type

order by count(vehicle_type) desc, Vehicle_Type;
```

Top Featured Vehicle:

```
select Vehicle_Name, vehicle_type, Seating_Capacity,concat("Engine is ",
Engine_in_CC, " cc and Top Speed is ",Top_Speed_kmph," kmph") "Top Feature",
count(Vehicle_Id) "Unit"

from vehicles

where Top_Speed_Kmph =( select max(Top_Speed_Kmph) from vehicles) and
Engine_in_CC =( select max(Engine_in_CC) from vehicles);
```

Vehicles With Highest Subscription Fees:

```
select distinct vehicle_name, model, Vehicle_Type, Seating_Capacity,
Subscription_Fee

from vehicles

order by Subscription_Fee desc;
```

Most demanding Cars

```
select count(N.vehicle_name) "Total Subscribed Times", N.Vehicle_Name

from (select s.Vehicle_Id , v.vehicle_name

            from subscribers s

            join vehicles v

            on s.Vehicle_Id = v.Vehicle_Id

            order by s.Vehicle_Id) N

        group by N.Vehicle_Name

        order by count(N.vehicle_name) desc;
```

Unsubscribed Or Remaining Vehicles

```
select Vehicle_Name, vehicle_Type, count(Vehicle_Name) "Unit"
```

```
from vehicles v
left join subscribers s
on v.Vehicle_Id = s.Vehicle_Id
where s.Vehicle_Id is null
group by Vehicle_Name
order by count(Vehicle_Name);
```

Subscribed Vehicle:

```
select distinct v.Vehicle_Id, v.Vehicle_Name
from vehicles v
join subscribers s
on v.Vehicle_Id = s.Vehicle_Id;
```

Subscribers Who Enroll Package:

```
select s.Client_Id ,s.ClientName, v.Subscription_Fee-
v.Subscription_Fee*(p.Discount_in_percentage/100) "Subscriber's payment with
Package"
from subscribers s
join vehicles v
on s.vehicle_id=v.vehicle_id
join subscription_package p
on s.Package_Id=p.Package_Id;
```

Subscribers Who Did Not Enroll Packages:

```
select s.Client_Id ,s.ClientName, v.Subscription_Fee "Subscriber's payment
without Package"
from subscribers s
join vehicles v
on s.vehicle_id=v.vehicle_id
where s.Package_Id is null;
```

Subscription per division:

```
select count(WL.Division) as "Total Subscription", WL.Division
from warehouse_locations WL
join subscribers VS
on WL.location_id = VS.location_id
group by WL.Division;
```

Total Invesment On Vehicle:

```
select sum(e.Exporting_Cost) + sum(v.Local_Cost) as "Total Invesment"
from exporters_info e
join vehicles v
on e.Vehicle_Name = v.Vehicle_Name;
```

Total Expense:

```
select sum(e.Exporting_Cost) + sum(v.Local_Cost) as "Total Expense"
from exporters_info e
join vehicles v
on e.Vehicle_Name = v.Vehicle_Name

union

select sum(salary_amount) from emp_salary_log;
```

Income From Subscription:

```
select sum(Subscription_Fee) as 'INCOME FROM SUBCRITION VCH'from vehicles;
```

Discont In Total:

```
select  SUM(Discount_in_percentage)/100 ' total given DISCOUNT(%) ' from
subscription_package;
```

Subscriber wants to renew or not (Query)

```
select Extend_Subscription_Period_or_return_vehicles(101,null,'Yes');
```

Testing To Update Data:

Adding Discount In New Customer:

```
update subscribers
 set Package_id=(select Discount_in_percentage
 from subscription_package where Package_Id=101 );
```

Increasing Subscription Fees:

```
update vehicles
set Subscription_Fee=60000
where vehicle_id=3;
```

Updating Client's Discount:

```
update Subscribers
set package_id = null
where Client_Id= 203;
```

Updating Subscription Fees:

```
update vehicles
set Subscription_Fee=60000
where Vehicle_id = 1;
```

Testing To Delete Data:

If a vehicle is no longer available for our campany:

```
delete from vehicles
where Vehicle_Id = 20;
```

If Subscription Period Ends:

```
delete from subscribers
where expiry_date = str_to_date('13-Feb-21','%d-%b-%y') and Wants_to_Extend
="No";
```

DISCUSSION

In this project, our goal was to implement such a service by DBMS that we can provide the desired vehicles of customers for a certain period so that they can easily afford a desired vehicle and enjoy their rides. So, we created this “vehicle subscription service” by DBMS in MySQL. In doing this project, we faced several dificulties like maintaining the relation between tables, maintaning data insertion, implementation of queries to perfectly fullfill our desired goal. Again, sometimes we could not get our outputs properly, then we had to sort out the problems in code to get the desired output. Finally, we were able to overcome the dificulties and produced our desired goal. We also considered all the anomalies like insertion, deletion, modification anomalies by normalizing our

relational database with the normalization technique which normaly rely on functional dependency.