Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

CSU34041

INFORMATION MANAGEMENT-II

PROJECT

TRIPPING – Travelling Made Easy

Submitted By

Harsh Dhingra

19323904

Introduction

I made a database for tours and travel company in India. This company organizes trips to different parts of India. India is the tourist hub of South Asia, and it embraces more than 5 million tourists from other locations of the world. There is a vast distribution of culture from India's northernmost part, i.e., Kashmir, to the southernmost part, i.e., Kanyakumari. Each tradition and culture symbolizes a different flavor of the place.

Need for the project

- Travel scam is on the rise, and many locals tend to cheat foreign nationals.
- To reduce the amount of time spent on planning a trip.

The project aims to provide users with affordable custom-designed travel packages that satisfy all the customer's needs. This will let the customer sit back and enjoy the trip instead of managing hotels, dates, cabs, and tourist attractions.

Project Structure

I decided to represent the database in a RDBMS, where several tables were made. Tables were made for admin, customer, packages, payment, location, hotels, cabs and each table had few attributes. All the attributes and relations can be viewed in the project in the E-R diagrams ahead in the project.

The customer table usually records all basic travel information such as phone number, first name, last name, age, sex, email id, address, no. of people travelling. Here email id is set as primary key.

An admin table is created with two attributes Username and name, where username is set to be a primary key.

A table which would be showing different package that is the package table which have package code, package detail , price and package type.

A location table, a payment table which would have an order id , customer id and amount to be paid.

A hotel table which has attributes such as hotel number, hotel name , available room, booked rooms, price and hotel number is set to be the primary key

A cab table which has attributes cab number, driver name , fare per km, no. of passengers. Cab number is set as primary key.

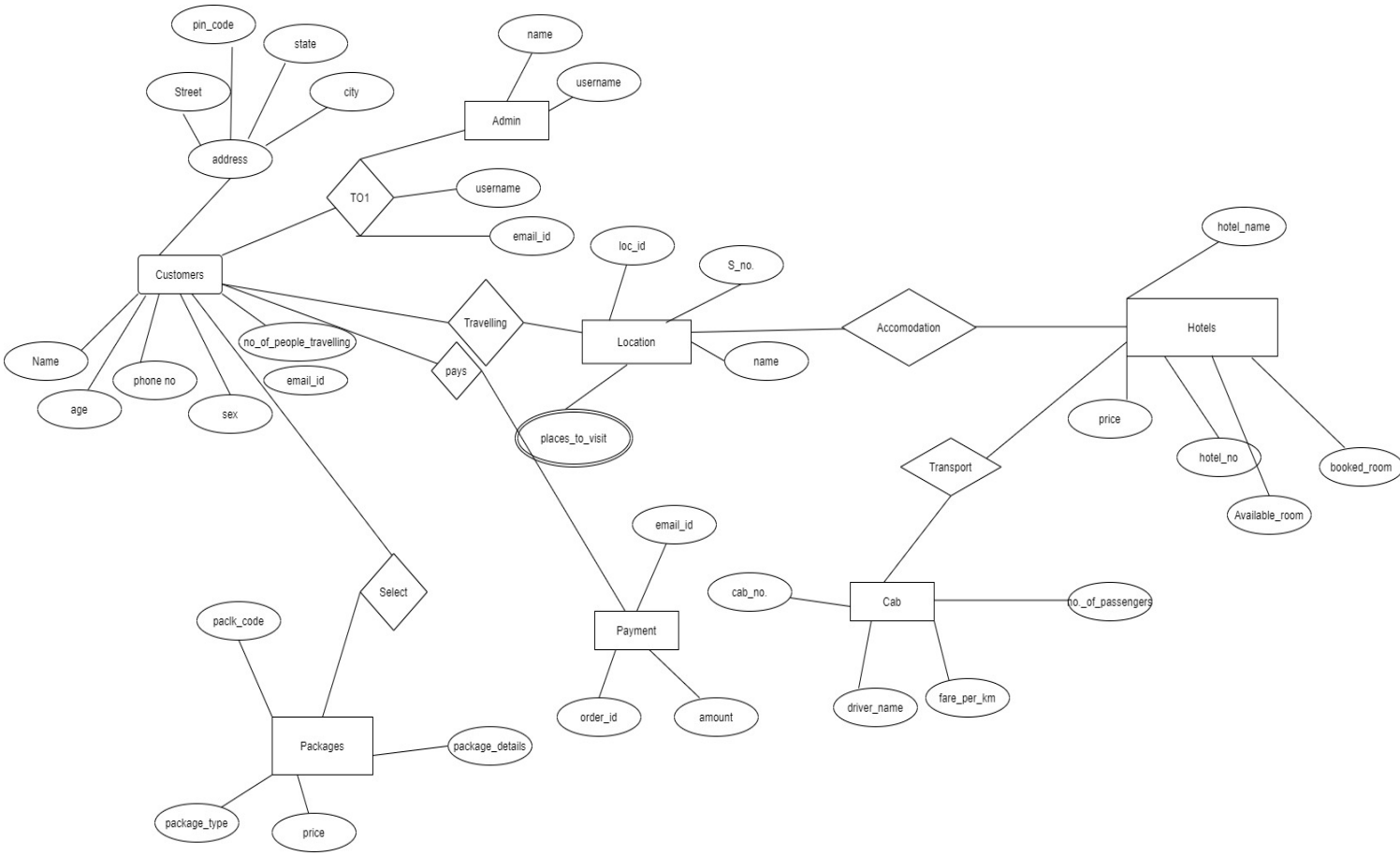
There are various other tables too that would be seen ahead in the project.

PL/SQL codes for insertion of data into customer table, various stored procedures and triggers on primary key violation and checking phone no.

E-R Diagram

ER Diagram refers to **Entity Relationship Diagram.**

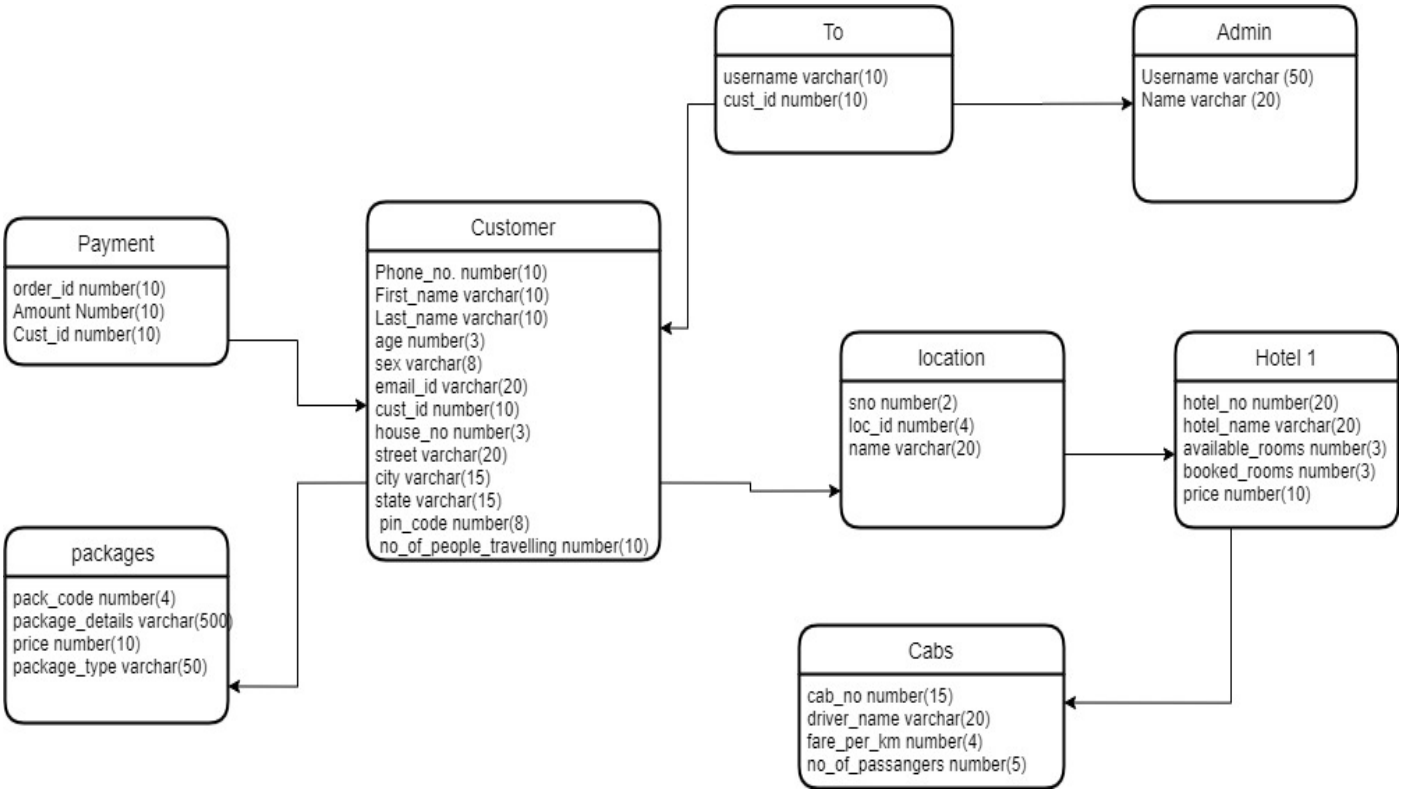
It focuses on the relationship of elements within entities instead of the relationship between entities themselves



Normalized Table

Normalization is a database design technique which organizes tables in a manner that reduces redundancy and dependency of data.

It divides larger tables to smaller tables and links them using relationships.



Normalization:

The tables produced from ER diagram were found to be normalized as they sucessfully completed all the tests required by the tables to be fulfilled upto 3rd Normal Form

Different tables present in my database and its sturcture:

In my database, I created several tables, the structure for different tables is given below-

Create Table Codes for these tables are present in appendix

Location

Name	Null?	Type
S_no		Number(2)
Location_id	NOT NULL	Number(4)
Name	NOT NULL	Varchar(20)

Packages

Name	Null?	Type
Pack_code		number(4)
Package_ Details	NOT NULL	Varchar(500)
Price	NOT NULL	Number(10)
Package_Type	NOT NULL	Varchar(50)

ADMIN

Name	Null?	Type
Username	NOT NULL	Varchar(50)
Name	NOT NULL	varchar (20)

Customer

Name	Null?	Type
First_name	NOT NULL	varchar2(10)
Last_name	NOT NULL	varchar2(10)
Age	NOT NULL	Number(3)
Sex	NOT NULL	varchar2(8)
Email_id	NOT NULL	varchar2(20)
Phone_no	NOT NULL	Number(10)
Cust_id	NOT NULL	Number(10)
House_no	NOT NULL	Number(3)
Street	NOT NULL	varchar2(20)
City	NOT NULL	Varchar(15)
State	NOT NULL	Varchar(15)
Pin_code	NOT NULL	Number(10)
no_of_people_travelling		Number(10)

Cabs

Name	Null?	Type
Cab_no	NOT NULL	Number(15)
Driver_name	NOT NULL	Varchar(20)
Fare_per_km	NOT NULL	Number(4)
No_of_passengers	NOT NULL	Number(5)

Payment

Name	Null?	Type
Order_id	NOT NULL	number(10)
Amount	NOT NULL	Number(10)
Cust_id	NOT NULL	Number(10)

Hotel1

Name	Null?	Type
Hotel_no	NOT NULL	number(20)
Hotel_name	NOT NULL	varchar2(20)
Available_rooms	NOT NULL	number(3)
Booked_rooms	NOT NUL	number(3)
Price	NOT NULL	number(10)

To

Name	Null?	Type
Username	NOT NULL	varchar2(50)
Cust_id	NOT NULL	number(10)
Gid		number(10)
Payment_id		varchar2(10)

PL/SQL

PLSQL is the extension of SQL, that enables us to combine the power of SQL with procedural statements

PL/SQL CODES FOR INSERTION OF DATA INTO CUSTOMER TABLE

```
create or replace procedure new_user
IS
First_name varchar(10) ;
Last_name varchar(10) ;
age number(3) ;
sex varchar(8) ;
email_id varchar(50) ;
phone_no number(10);
cust_id number(10) ;
house_no number(3) ;
street varchar(20) ;
city varchar(15) ;
state varchar(15) ;
pin_code number(8) ;
no_of_people_travelling number(10) ;
BEGIN
insert into Customer values( '&First_name ', '&Last_name',&age,'&sex', '&email_id', &phone_no ,
&cust_id , &house_no, '&street', '&city', '&state' ,&pin_code, &no_of_people_travelling )
COMMIT:
END;
/
```

Triggers

In SQL, a trigger is an event (SQL queries) - that's evoked(pressed and it executes a set of commands or queries) when a User performs an update or an insert or a delete operation in SQL over some table. SQL triggers provide an alternative way to check the integrity of data.

I have inserted two triggers that would be invoked if

PRIMARY KEY

```
CREATE OR REPLACE TRIGGER PRIMARY_KEY
BEFORE INSERT ON
packages FOR EACH
ROW
DECLARE
E
packages.pack_code%T
YPE; BEGIN
IF (:NEW.pack_code IS NULL) THEN
RAISE_APPLICATION_ERROR(-20099, 'trigger generated error PRIMARY KEY VIOLATION
BECAUSE IT CANNOT BE NULL');
END IF;
```

```
SELECT pack_code INTO E FROM payment WHERE pack_code=:NEW.pack_code;
RAISE_APPLICATION_ERROR(-20003, 'trigger generated error PRIMARY KEY VIOLATION
BECAUSE IT SHOULD BE UNIQUE');
EXCEPTION
WHEN NO_DATA_FOUND
THEN NULL;
END;
```

CHECK PHONE NO.

```
CREATE OR REPLACE TRIGGER PHONE_CHECK
BEFORE INSERT OR UPDATE ON CUSTOMER
FOR EACH ROW
BEGIN
IF LENGTH(:NEW.PHONE_NO) NOT LIKE 10 THEN
RAISE_APPLICATION_ERROR(-20000,'TRIGGER GENERATED ERROR PHONE NO CANNOT
HAVE OTHER THEN 10 DIGITS');
END IF;
END;
/
```

STORED PROCEDURES

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name stored in the database in the compiled form so that several programs can share it.

The following stored procedures were added to my database

Note : These codes are added in Appendix

HOTEL BOOKING

```
set serveroutput on;
CREATE OR REPLACE Procedure hotel_booking(hid in number , npeople in number)
IS
nrooms number;
arooms number;
cost number(8);
crooms number;
brooms number;
drooms number;
price number;
begin
nrooms := npeople/3;
SELECT available_rooms,price,booked_rooms INTO arooms,price,brooms FROM hotel1 WHERE
hotel_no = hid;
if (nrooms <= arooms)
THEN
crooms := arooms-nrooms;
SELECT price,booked_rooms INTO price,brooms FROM hotel1 WHERE
hotel_no = hid;
drooms := brooms + nrooms;
dbms_output.put_line( nrooms || ' rooms have been allotted to you' );
```

```
UPDATE hotel1 SET available_rooms = crooms,booked_rooms = drooms WHERE hotel_no = hid;
cost := nrooms * price ;
dbms_output.put_line('total expense of the hotel is: ' || cost);
else
dbms_output.put_line('rooms have not been allotted');
END IF;
End;
/
declare
npeople number;
hid number;
Begin
hid := &enter_hotel_no;
npeople := &no_of_people_accompanying;
hotel_booking(hid,npeople);
END;
/
```

BOOKING PACKAGES

```
set serveroutput on;
create or replace procedure package_booking(pcode in number)
IS
p number;
cursor c2 is select pack_code,package_type,package_details from packages where pack_code=pcode;
cost number(10);
begin
select price into p from packages where pack_code=pcode;
dbms_output.put_line('you have selected: ');
for rec in c2 loop
dbms_output.put_line(rec.pack_code||' '||rec.package_type||' '||rec.package_details);
end loop;
cost:= p+0.18*p;
dbms_output.put_line('total cost of the package includding GST is :' || cost);
end;
/
DECLARE
cursor c1 is select pack_code,package_type,package_details from packages;
BEGIN
dbms_output.put_line('different packages available are:');
for rec in c1 loop
dbms_output.put_line(rec.pack_code||' '||rec.package_type||' '||rec.package_details);
end loop;
```

```
end;
/
declare
pcode number;
begin
pcode:=&enter_the_pack_code;
PACKAGE_BOOKING(PCODE) ;
END;
/
```

PLACES TO VISIT

```
set serveroutput on;
create or replace procedure tourist_attraction(lid in number)
IS
cursor c2 is select places_to_visit from location_places_to_visit where loc_id=lid;
begin
dbms_output.put_line('famous tourist attractions are');
for rec in c2 loop
dbms_output.put_line(rec.places_to_visit);
end loop;
end;
/
declare
cursor c1 is select distinct loc_id,name from location;
begin
dbms_output.put_line('enter the location id where you want to see the tourist attraction spots');
for rec in c1 loop
dbms_output.put_line(rec.loc_id||' '||rec.name);
end loop;
end;
/
declare
lid location.loc_id%type;
begin
lid:= &enter_loc_id;
tourist_attraction(lid);
end;
/
```

SECURITY

Security is an integral part of every database, and its been given very high importance.

Database security is concerned with deliberate corruption. Access control helps to prevent unauthorized persons from accessing the system. In this database of mine the DBMS will restrict access to the database for users.

A DBA is employed in the tourism company, that would be creating admin accounts. An admin would be granted rights to insert, update details.

VIEWS

A view is a database object that is created using a Select Query with complex logic, so views are said to be a logical representation of the physical data, i.e Views behave like a physical table and users can use them as database objects in any part of SQL queries**.**

```
CREATE VIEW CustBooking AS

SELECT Customer.cust_id, Customer.First_name,
Customer.Last_name,location.name,Hotel1.Hotel_name, Payment.order_id

FROM Customer, location, Hotel1, Payment;
```

APPENDIX

CREATING TABLES

```
CREATE TABLE cabs

(

cab_no number(15) NOT NULL,

driver_name varchar(20) NOT NULL,

fare_per_km number(4) NOT NULL,

no_of_passangers number(5) NOT NULL,

PRIMARY KEY (cab_no)

);
CREATE TABLE Hotel1

(

hotel_no number(20) NOT NULL,

hotel_name varchar(20) NOT NULL,

available_rooms number(3) NOT NULL,

booked_rooms number(3) NOT NULL,

price number(10) NOT NULL,

PRIMARY KEY (hotel_no),

);
```

```
create table Payment(
order_id number(10) NOT NULL,
Amount Number(10) NOT NULL,
Cust_id number(10) NOT NULL,
PRIMARY KEY (order_id, cust_id),
FOREIGN KEY (cust_id) REFERENCES Customer(cust_id)
);

create table To(
username varchar(10) NOT NULL,
cust_id number(10) NOT NULL,
PRIMARY KEY (username, cust_id),
FOREIGN KEY (username) REFERENCES Admin(username),
FOREIGN KEY (cust_id) REFERENCES Customer(cust_id)
);
CREATE TABLE location
(
sno number(2);
loc_id number(4) NOT NULL,
name varchar(20) NOT NULL,
PRIMARY KEY (sno),
);
CREATE TABLE packages
(
pack_code number(4) ,
package_details varchar(500) NOT NULL,
price number(10) NOT NULL,
package_type varchar(50) NOT NULL
);
CREATE TABLE customer
(
First_name varchar(10) NOT NULL,
Last_name varchar(10) NOT NULL,
age number(3) NOT NULL,
sex varchar(8) NOT NULL,
email_id varchar(20) NOT NULL,
```

```
phone_no number(10) NOT NULL,
cust_id number(10) NOT NULL,
house_no number(3) NOT NULL,
street varchar(20) NOT NULL,
city varchar(15) NOT NULL,
state varchar(15) NOT NULL,
pin_code number(8) NOT NULL,
no_of_people_travelling number(10) ,
PRIMARY KEY (email_id));
create table Admin (
Username varchar (50),
Name varchar (20),
Primary key (Username));
```

Insertion of values into different tables

packages

```
Insert into packages values(1,'Shimla,3N-4D,Hotel-ParkVilla,Food,Cabs',33000,'GOLD');
Insert into packages values(2,'Shimla,2N-3D,Hotel-SA_Residency,Food,Cabs',17000,'SILVER');
Insert into packages values(3,'Manali,3N-4D,Hotel-ParkView,Food,Cabs',33000,'GOLD');
Insert into packages values(4,'Manali,2N-3D,Hotel-ParkAvenue,Food,Cabs',18000,'SILVER');
Insert into packages values(5,'Mussorie,3N-4D,Hotel-The_LOTUS,Food,Cabs',28000,'GOLD');
Insert into packages values(6,'Mussories,2N-3D,Hotel-ParkVilla,Food,Cabs',18000,'SILVER');
Insert into packages values(7,'Ladakh,3N-4D,Hotel-MoonView,Food,Cabs',38000,'GOLD');
Insert into packages values(8,'Ladakh,2N-3D,Hotel-Paradise,Food,Cabs',30000,'SILVER');
Insert into packages values(9,'Goa,3N-4D,Hotel-The_Lalit,Food,Cabs',38000,'GOLD');
Insert into packages values(10,'Goa,2N-3D,Hotel-The_Taj,Food,Cabs',33000,'GOLD');
```

hotel info

```
Insert into Hotel1 values(1,'A1',10,0,3000);
Insert into Hotel1 values(2,'A2',19,0,3500);
Insert into Hotel1 values(3,'A3',12,0,2000);
Insert into Hotel1 values(4,'A4',8,0,3800);
```

```
Insert into Hotel1 values(5,'B1',10,0,3000);
Insert into Hotel1 values(6,'B2',10,0,2000);
Insert into Hotel1 values(7,'B3',11,0,3000);
Insert into Hotel1 values(8,'B4',19,3,3000);
Insert into Hotel1 values(9,'C1',10,4,3000);
Insert into Hotel1 values(10,'C2',13,10,1800);
Insert into Hotel1 values(11,'C3',10,9,3000);
Insert into Hotel1 values(12,'C4',10,0,5000);
Insert into Hotel1 values(13,'D1',18,0,4000);
Insert into Hotel1 values(14,'D2',10,0,3000);
Insert into Hotel1 values(15,'D3',17,0,2000);
Insert into Hotel1 values(16,'D4',10,0,6000);
Insert into Hotel1 values(17,'E1',19,0,6000);
Insert into Hotel1 values(18,'E2',18,4,2000);
Insert into Hotel1 values(19,'E3',16,5,4000);
Insert into Hotel1 values(20,'E4',10,6,3000);
```

locations

```
Insert into location values(1,1,'Shimla');
Insert into location values(2,1,'Shimla');
Insert into location values(3,1,'Shimla');
Insert into location values(4,1,'Shimla');
Insert into location values(5,2,'Manali');
Insert into location values(6,2,'Manali');
Insert into location values(7,2,'Manali');
Insert into location values(8,2,'Manali');
Insert into location values(9,3,'Mussorie');
Insert into location values(10,3,'Mussorie');
Insert into location values(11,3,'Mussorie');
Insert into location values(12,3,'Mussorie');
Insert into location values(13,4,'Ladakh');
Insert into location values(14,4,'Ladakh');
Insert into location values(15,4,'Ladakh');
Insert into location values(16,4,'Ladakh');
```

```
Insert into location values(17,5,'Goa');
Insert into location values(18,5,'Goa');
Insert into location values(19,5,'Goa');
Insert into location values(20,5,'Goa');
```

places to visit

```
Insert into location_places_to_visit values(1,'jakhoo',1);
Insert into location_places_to_visit values(2,'The Ridge',1);
Insert into location_places_to_visit values(3,'Mall Road',1);
Insert into location_places_to_visit values(4,'kufri',1);
Insert into location_places_to_visit values(5,'Hidimba mandir',2);
Insert into location_places_to_visit values(6,'manu mandir',2);
Insert into location_places_to_visit values(7,'jogni falls',2);
Insert into location_places_to_visit values(8,'bhrigu lake',2);
Insert into location_places_to_visit values(9,'mussorie lake',3);
Insert into location_places_to_visit values(10,'camels back road',3);
Insert into location_places_to_visit values(11,'lal tibba',3);
Insert into location_places_to_visit values(12,'gunhill road',3);
Insert into location_places_to_visit values(13,'pangong tso',4);
Insert into location_places_to_visit values(14,'magnetic hill',4);
Insert into location_places_to_visit values(15,'leh palace',4);
Insert into location_places_to_visit values(16,'chadar trek',4);
Insert into location_places_to_visit values(17,'palolem beach',5);
Insert into location_places_to_visit values(18,'panaji',5);
Insert into location_places_to_visit values(19,'calangute',5);
Insert into location_places_to_visit values(20,'anjuna beach',5);
```

customer

```
insert into customer values
('Abhishek','sharma',19,'male','dbmsproject',8054358745,21,'sdhshadh','adjbja','asjbdjs',1401,8);
insert into customer values
'Abhi','shek',15,'female','jhsajjsjghsa',7894558745,1,'sdhshadh','adjbja','asjbdjs',15001,5);
insert into customer values
('aditi','sgdhas',20,'male','asmdbnas',536458745,31,'sdhshadh','adjbja','asjbdjs',1001,12);
insert into customer values
('aman','rao',21,'male','abnjzssc',8452158745,22,'sdhshadh','adjbja','asjbdjs',140,8);
```

```
insert into customer values
('shikher','tomar',24,'female','asdfg',8054358745,41,'sdhshadh','adjbja','asjbdjs',15001,5);
insert into customer values
('abh','qwerty',32,'male','snbcs',8054386542,23,'sdhshadh','adjbja','asjbdjs',14501,4);
insert into customer values
('Ab','sdfgh',25,'female','wassde',8054358654,22,'sdhshadh','adjbja','asjbdjs',101,8);
insert into customer values
('Abhis','qwef',29,'male','wwdwdwd',8054358845,27,'sdhshadh','adjbja','asjbdjs',1450,8);
insert into customer values
('Abhis','sjhdjh',49,'male','dwdawnda',8054359745,51,'sdhshadh','adjbja','asjbdjs',1451,3);
insert into customer values
('hek','ankdn',52,'female','dawdnwa',8054358745,21,'sdhshadh','adjbja','asjbdjs',01,3);
insert into customer values
('shek','shaaadnd',41,'male','dwadjawb',8054658745,61,'sdhshadh','adjbja','asjbdjs',1001,2);
insert into customer values
('hishek','abns',26,'female','ddnjawd',8054358245,71,'sdhshadh','adjbja','asjbdjs',1450,8);
insert into customer values
('Abhiswe','sadjabs',35,'male','adnnwad',8054158745,81,'sdhshadh','adjbja','asjbdjs',145,2);
```

ADMIN

```
insert into admin values('ADI@123','JERRY');
insert into admin values('ABHI','TERRY');
insert into admin values('ANJAM1','ANJANA');
insert into admin values('ARA0','AMAN');
insert into admin values('SANJ2','SANJAM');
```

CABS

```
INSERT INTO CABS VALUES(12345,'MANJOT',20,3);
INSERT INTO CABS VALUES(56789,'AMRIT',12,4);
INSERT INTO CABS VALUES(98978,'MANDEEP',35,2);
INSERT INTO CABS VALUES(13457,'JASMEET',40,1);
INSERT INTO CABS VALUES(18890,'PURAN',25,3);
```

PAYMENT

```
INSERT INTO PAYMENT VALUES(234,700,'dbmsproject');
INSERT INTO PAYMENT VALUES(314,290,'jhsajjsjghsa');
INSERT INTO PAYMENT VALUES(198,400,'wwdwdwd');
```

```
INSERT INTO PAYMENT VALUES(6789,900,'dwadjawb');
INSERT INTO PAYMENT VALUES(890,500,'asdfg');
```

TO1

```
insert into To1 values('ADI@123','asdfg');
insert into To1 values('ABHI','wwdwdwd');
insert into To1 values('ANJAM1','jhsajjsjghsa');
insert into To1 values('ARA0','dbmsproject');
insert into To1 values('SANJ2','dwadjawb');
```

PL/SQL CODES

PL/SQL CODES FOR INSERTION OF DATA INTO CUSTOMER TABLE

```
create or replace procedure new_user
IS
First_name varchar(10) ;
Last_name varchar(10) ;
age number(3) ;
sex varchar(8) ;
email_id varchar(50) ;
phone_no number(10);
cust_id number(10) ;
house_no number(3) ;
street varchar(20) ;
city varchar(15) ;
state varchar(15) ;
pin_code number(8) ;
no_of_people_travelling number(10) ;
BEGIN
insert into Customer values( '&First_name ', '&Last_name',&age,'&sex', '&email_id', &phone_no ,
&cust_id , &house_no, '&street', '&city', '&state' ,&pin_code, &no_of_people_travelling )
COMMIT:
END;
/
```

STORED PROCEDURES

HOTEL BOOKING

```
set serveroutput on;
CREATE OR REPLACE Procedure hotel_booking(hid in number , npeople in number)
```

```
IS
nrooms number;
arooms number;
cost number(8);
crooms number;
brooms number;
drooms number;
price number;
begin
nrooms := npeople/3;
SELECT available_rooms,price,booked_rooms INTO arooms,price,brooms FROM hotel1 WHERE
hotel_no = hid;
if (nrooms <= arooms)
THEN
crooms := arooms-nrooms;
SELECT price,booked_rooms INTO price,brooms FROM hotel1 WHERE
hotel_no = hid;
drooms := brooms + nrooms;
dbms_output.put_line( nrooms || ' rooms have been allotted to you' );
UPDATE hotel1 SET available_rooms = crooms,booked_rooms = drooms WHERE hotel_no = hid;
cost := nrooms * price ;
dbms_output.put_line('total expense of the hotel is: ' || cost);
else
dbms_output.put_line('rooms have not been allotted');
END IF;
End;
/
declare
npeople number;
hid number;
Begin
hid := &enter_hotel_no;
npeople := &no_of_people_accompanying;
hotel_booking(hid,npeople);
END;
```

```
/
```

BOOKING PACKAGES

```
set serveroutput on;
create or replace procedure package_booking(pcode in number)
IS
p number;
cursor c2 is select pack_code,package_type,package_details from packages where pack_code=pcode;
cost number(10);
begin
select price into p from packages where pack_code=pcode;
dbms_output.put_line('you have selected: ');
for rec in c2 loop
dbms_output.put_line(rec.pack_code||' '||rec.package_type||' '||rec.package_details);
end loop;
cost:= p+0.18*p;
dbms_output.put_line('total cost of the package includding GST is :' || cost);
end;
/
DECLARE
cursor c1 is select pack_code,package_type,package_details from packages;
BEGIN
dbms_output.put_line('different packages available are:');
for rec in c1 loop
dbms_output.put_line(rec.pack_code||' '||rec.package_type||' '||rec.package_details);
end loop;
end;
/
declare
pcode number;
begin
pcode:=&enter_the_pack_code;
PACKAGE_BOOKING(PCODE) ;
END;
/
```

PLACES TO VISIT

```
set serveroutput on;
create or replace procedure tourist_attraction(lid in number)
IS
cursor c2 is select places_to_visit from location_places_to_visit where loc_id=lid;
begin
dbms_output.put_line('famous tourist attractions are');
for rec in c2 loop
dbms_output.put_line(rec.places_to_visit);
end loop;
end;
/
declare
cursor c1 is select distinct loc_id,name from location;
begin
dbms_output.put_line('enter the location id where you want to see the tourist attraction spots');
for rec in c1 loop
dbms_output.put_line(rec.loc_id||' '||rec.name);
end loop;
end;
/
declare
lid location.loc_id%type;
begin
lid:= &enter_loc_id;
tourist_attraction(lid);
end;
/
```

TRIGGERS

PRIMARY KEY

```
CREATE OR REPLACE TRIGGER PRIMARY_KEY
BEFORE INSERT ON packages
FOR EACH ROW
DECLARE
E packages.pack_code%TYPE;
```

```
BEGIN
IF (:NEW.pack_code IS NULL) THEN
RAISE_APPLICATION_ERROR(-20099, 'trigger generated error PRIMARY KEY VIOLATION
BECAUSE IT CANNOT BE NULL');
END IF;
SELECT pack_code INTO E FROM payment WHERE pack_code=:NEW.pack_code;
RAISE_APPLICATION_ERROR(-20003, 'trigger generated error PRIMARY KEY VIOLATION
BECAUSE IT SHOULD BE UNIQUE');
EXCEPTION
WHEN NO_DATA_FOUND THEN
NULL;
END;
/
```

CHECK PHONE NUMBER

```
CREATE OR REPLACE TRIGGER PHONE_CHECK
BEFORE INSERT OR UPDATE ON CUSTOMER
FOR EACH ROW
BEGIN
IF LENGTH(:NEW.PHONE_NO) NOT LIKE 10 THEN
RAISE_APPLICATION_ERROR(-20000,'TRIGGER GENERATED ERROR PHONE NO CANNOT
HAVE OTHER THEN 10 DIGITS');
END IF;
END;
/
```

SCREENSHOTS

```
1 row created.

SQL> INSERT INTO PAYMENT VALUES(234,700,'dbmsproject');

1 row created.

SQL> INSERT INTO PAYMENT VALUES(314,290,'jhsajjsjghsa');

1 row created.

SQL> INSERT INTO PAYMENT VALUES(198,400,'wwdwdwd');

1 row created.

SQL> INSERT INTO PAYMENT VALUES(6789,900,'dwadjawb');

1 row created.

SQL> INSERT INTO PAYMENT VALUES(890,500,'asdfg');

1 row created.

SQL> insert into To1 values('ADI@123','asdfg');

1 row created.

SQL> insert into To1 values('ABHI','wwdwdwd');

1 row created.

SQL> insert into To1 values('ANJAM1','jhsajjsjghsa');

1 row created.

SQL> insert into To1 values('ARA0','dbmsproject');

1 row created.

SQL> insert into To1 values('SANJ2','dwadjawb');

1 row created.

SQL>
```

```
 16  );

Table created.

SQL> CREATE TABLE payment
  2  (
  3    order_id number(10) NOT NULL,
  4    amount number(10)  NOT NULL,
  5    email_id varchar(20) NOT NULL,
  6    PRIMARY KEY (order_id, cust_id),
  7    FOREIGN KEY (email_id) REFERENCES Customer(email_id)
  8  );
  PRIMARY KEY (order_id, cust_id),
                         *
ERROR at line 6:
ORA-00904: "CUST_ID": invalid identifier


SQL> CREATE TABLE payment
  2  (
  3    order_id number(10) NOT NULL,
  4    amount number(10)  NOT NULL,
  5    email_id varchar(20) NOT NULL,
  6    PRIMARY KEY (order_id, email_id),
  7    FOREIGN KEY (email_id) REFERENCES Customer(email_id)
  8  );

Table created.

SQL> CREATE TABLE To
  2  (
  3    username varchar(10) NOT NULL,
  4    cust_id number(10) NOT NULL,
  5    PRIMARY KEY (username, cust_id),
  6    FOREIGN KEY (username) REFERENCES Admin(username),
  7    FOREIGN KEY (cust_id) REFERENCES Customer(cust_id)
  8  );
CREATE TABLE To
             *
ERROR at line 1:
ORA-00903: invalid table name
```

```
Table created.

SQL> CREATE TABLE Customer
  2  (
  3    First_name varchar(10) NOT NULL,
  4    Last_name varchar(10) NOT NULL,
  5    age number(3) NOT NULL,
  6    sex varchar(8) NOT NULL,
  7    email_id varchar(20) NOT NULL,
  8    phone_no number(10) NOT NULL,
  9    cust_id number(10) NOT NULL,
 10    house_no number(3) NOT NULL,
 11    street varchar(20) NOT NULL,
 12    city varchar(15) NOT NULL,
 13    state varchar(15)  NOT NULL,
 14    pin_code number(8) NOT NULL,
 15    no_of_people_travelling number(10) ,
 16    PRIMARY KEY (email_id)
 17  );

Table created.

SQL> CREATE TABLE payment
  2  (
  3    order_id number(10) NOT NULL,
  4    amount number(10)  NOT NULL,
  5    cust_id number(10) NOT NULL,
  6    PRIMARY KEY (order_id, cust_id),
  7    FOREIGN KEY (cust_id) REFERENCES Customer(cust_id)
  8  );
CREATE TABLE payment
             *
ERROR at line 1:
ORA-00955: name is already used by an existing object


SQL> drop table payment;

Table dropped.

SQL> CREATE TABLE payment
  2  (
  3    order_id number(10) NOT NULL,
  4    amount number(10)  NOT NULL,
```

```
SQL> CREATE OR REPLACE TRIGGER PHONE_CHECK
  2  BEFORE INSERT OR UPDATE ON CUSTOMER
  3  FOR EACH ROW
  4  BEGIN
  5  IF LENGTH(:NEW.PHONE_NO) NOT LIKE 10 THEN
  6  RAISE_APPLICATION_ERROR(-20000,'TRIGGER GENERATED ERROR PHONE NO CANNOT HAVE O
THER THEN 10 DIGITS');
  7  END IF;
  8  END;
  9  /

Trigger created.

SQL>
```

```
ATTRIBUTE MESSAGE_NUMBER
--------- --------------
PRIMARY_KEY                    TRIGGER              1          7         44
PL/SQL: ORA-00904: "PACK_CODE": invalid identifier
ERROR                  0

NEW_USER                       PROCEDURE            1         20          1
PLS-00428: an INTO clause is expected in this SELECT statement
ERROR                  428

NAME                           TYPE           SEQUENCE       LINE   POSITION
------------------------------ ------------ ---------- ---------- ----------
TEXT
--------------------------------------------------------------------------------
ATTRIBUTE MESSAGE_NUMBER
--------- --------------

PRIMARY_KEY                    TRIGGER              2          7          1
PL/SQL: SQL Statement ignored
ERROR                  0


SQL> CREATE OR REPLACE TRIGGER PRIMARY_KEY
  2  BEFORE INSERT ON packages
  3  FOR EACH ROW
  4  DECLARE
  5  E packages.pack_code%TYPE;
  6  BEGIN
  7  IF (:NEW.pack_code IS NULL) THEN
  8  RAISE_APPLICATION_ERROR(-20099, 'trigger generated error PRIMARY KEY VIOLATION BECAUSE IT CANNOT BE NULL');
  9  END IF;
 10  SELECT pack_code INTO E FROM payment WHERE pack_code=:NEW.pack_code;
 11  RAISE_APPLICATION_ERROR(-20003, 'trigger generated error PRIMARY KEY VIOLATION
 12  BECAUSE IT SHOULD BE UNIQUE');
 13  EXCEPTION
 14  WHEN NO_DATA_FOUND THEN
 15  NULL;
 16  END;
 17  /
```