**FINAL REPORT**

**Project outline:**

Travel Express is a prototype of an online e-commerce web application for instance home away where in a user can book hotels. In addition to this we will also allow a user to book the mode of transport via flight, road, and train.

Technology used: Java web servlets, JSP, JavaScript, Oracle 11g (Sequence, Procedure)

Design Pattern: Factory design pattern

Architecture: MVC (Model, View, Controller) architecture

Features:

* User can sign up or login.
* If existing user has forgotten his password it can be updated.
* User can select destination and mode of travel Flight, Train, Road.
* User can select hotel.
* User can enter dependents’ details.
* User can apply promo code.
* After making payments, the itinerary of the user will be displayed.

**Description:**

The entities used in the project are:

1. User
2. Region
3. Hotel
4. Itinerary
5. Dependent\_details
6. Customer\_Support
7. Offers
8. Mode\_Of\_Transport
9. Mode\_Air
10. Mode\_Train
11. Mode\_Road
12. User\_Support
13. Technical\_Support

The relationship between these entities can be explained as follows:

* User prepares his itinerary by selecting mode of transport, region and hotel. At one instance, a user can draft multiple itineraries. So, there is a 1: N association between user and itinerary.
* A user can have 0 or many dependents. Also, the dependent entity is a weak entity. Hence, an identifying relationship exists between dependent\_details and user table.
* In case of issues, a user contacts customer support. One user can contact multiple customer support executives. Similarly, one executive can handle many users.
* A user can select many hotels and regions to design his travel plan.

The Entity relationship diagram below explains the relationship in detail:

ER Diagram:

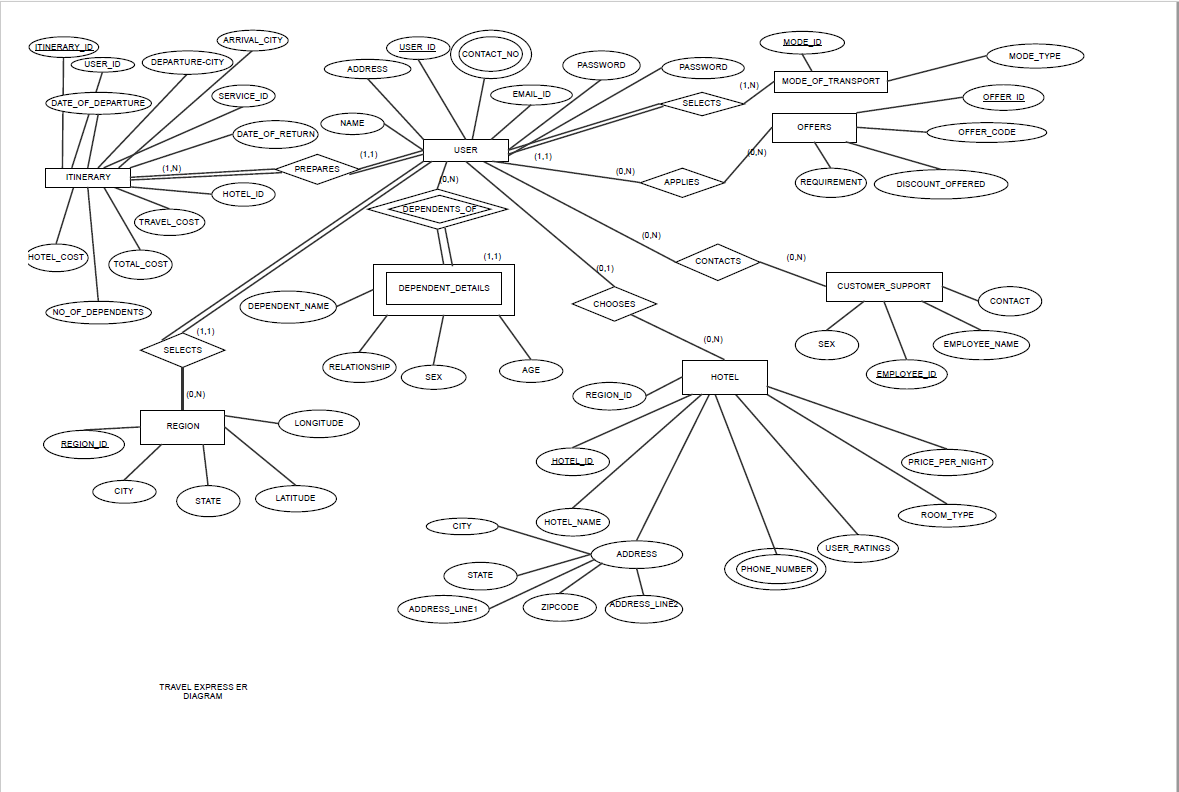


Figure 1

Subclasses and superclass:

The entities mode\_of\_transport and customer support are super class entities in the travel express database.

* Mode\_air, Mode\_Road and Mode\_Train are the subclasses of superclass entity mode\_of\_Transport. A disjoint exists between these superclass-subclass combination where disjoint is formed on the basis of Mode\_Type parameter.
* User\_Support and Technical\_Support are subclasses of superclass Customer\_Support. An overlap exists between these subclasses.

EER Diagram:

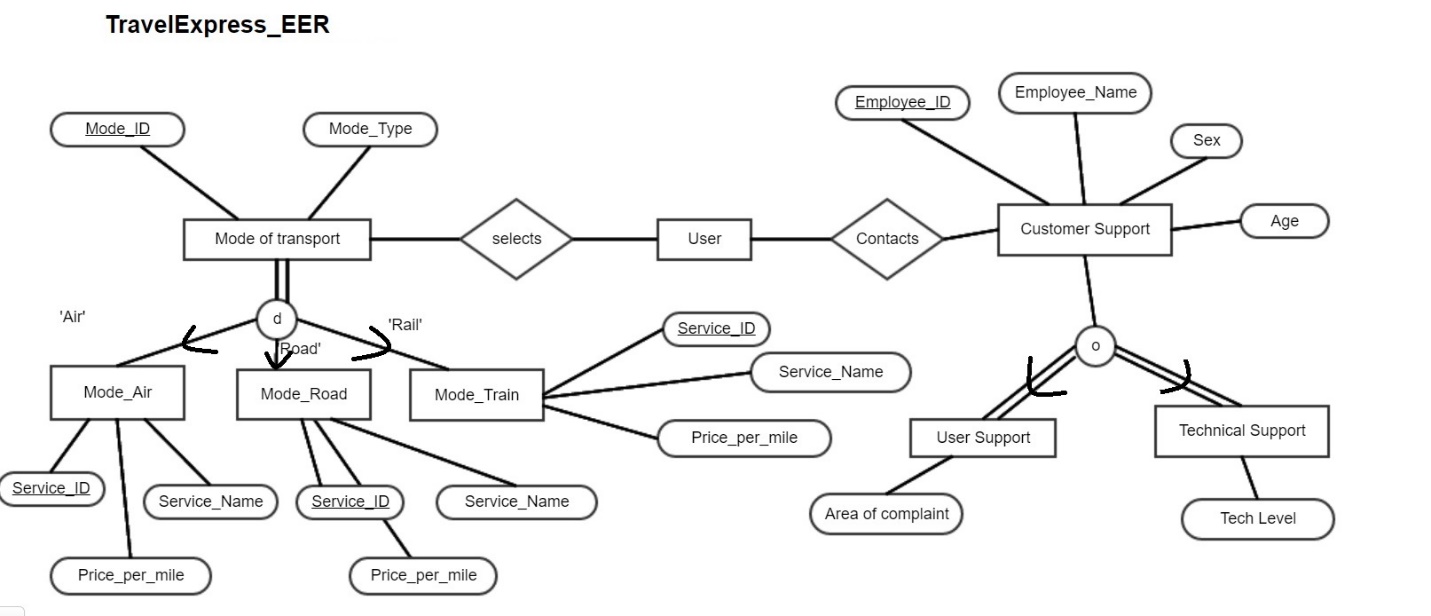


Figure 2

**EXPLANATION OF DATA MODEL TO SQL TABLES:**

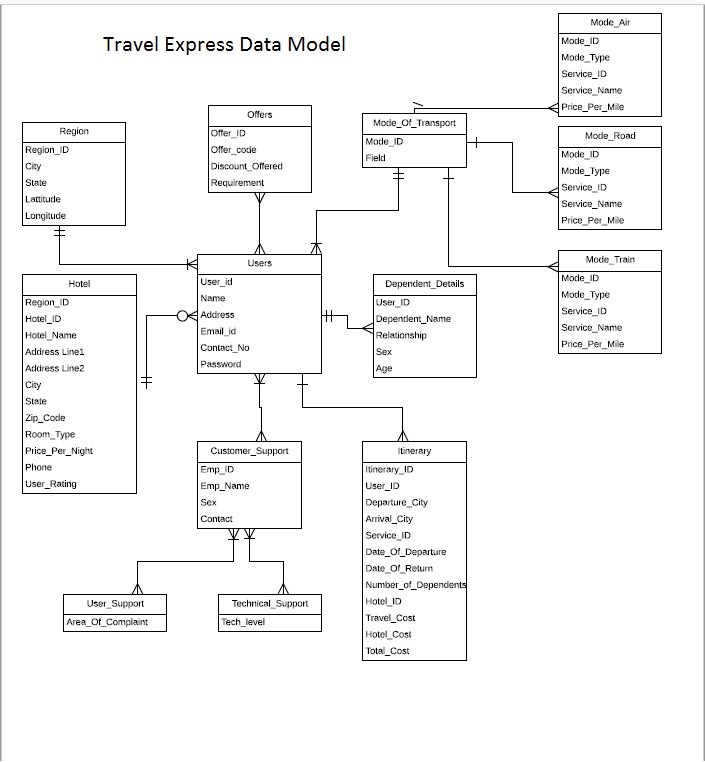


Figure 3

* **Mapping of strong entities:** The strong entities User, Region, Hotels, Customer\_Support, Mode\_Transport and offers will form separate relations i.e. tables in the database.
* **Mapping of 1: N relationships: --**User and itinerary entities are in 1: N relationship. Hence, the primary key of user User\_ID will form the foreign key of itinerary table.

Entities mode\_air, mode\_road and mode\_train are in 1: N relationship with the table Mode\_transport. Hence, Mode\_ID will be the foreign key in the tables mode\_air, mode\_road and mode\_train.

Entities User\_Support and Technical\_Support are in 1: N relationship with Customer\_Support. The primary key of Customer\_Support- Emp\_ID will form the foreign key in User\_Support and Technical\_Support.

* **Mapping of Weak entity:** Dependent\_detail is the weak entity in the model with user as the owner entity. A table dependent\_detail is created with attributes from dependent\_detail and primary key of Owner entity i.e User as the foreign key. The primary key will be the combination of dependent\_name and user\_id.

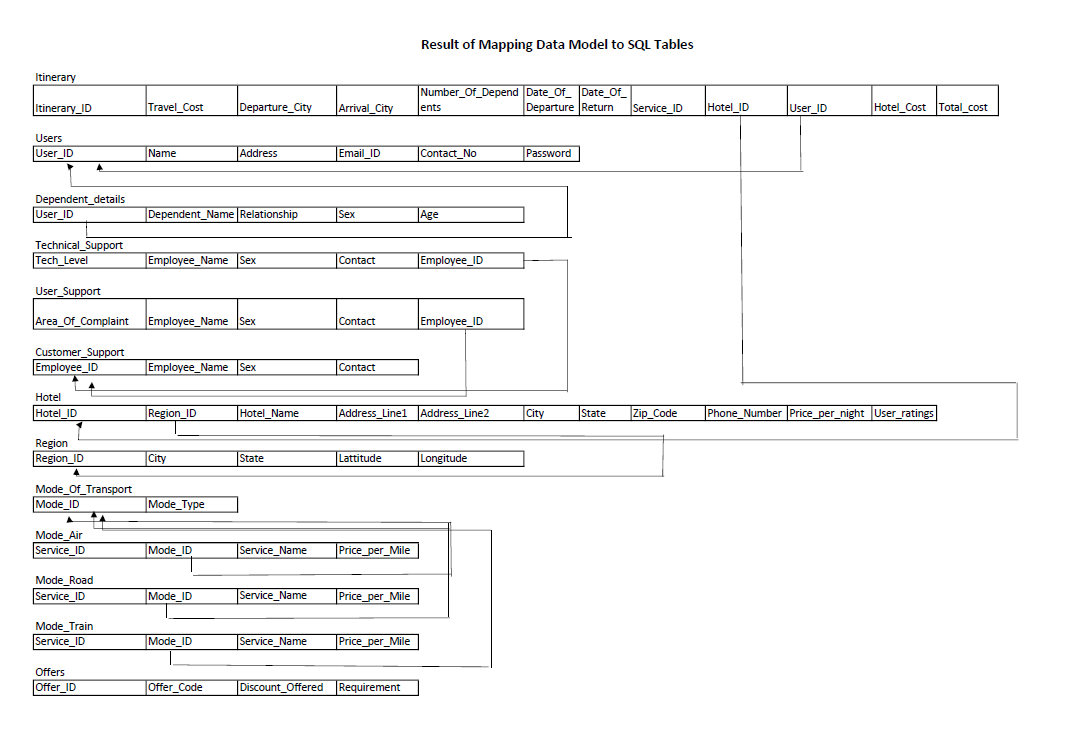
The Mapping can be demonstrated by the following figure:

Figure 4

**SQL IMPLEMENTATION:**

The following tables were created for implementing the SQL Database.

1) CREATE TABLE "HR"."DEPENDENT\_DETAILS"

(

"USER\_ID" NUMBER,

"DEPENDENT\_NAME" VARCHAR2(100 BYTE),

"RELATIONSHIP" VARCHAR2(50 BYTE),

"SEX" VARCHAR2(2 BYTE),

"AGE" NUMBER,

FOREIGN KEY ("USER\_ID")

REFERENCES "HR"."USERS" ("USER\_ID") ENABLE

)

2) CREATE TABLE "HR"."ITINERARY"

(

"ITINERARY\_ID" NUMBER,

"USER\_ID" NUMBER,

"DEPARTURE\_CITY" VARCHAR2(100 BYTE),

"ARRIVAL\_CITY" VARCHAR2(100 BYTE),

"SERVICE\_ID" VARCHAR2(240 BYTE),

"DATE\_OF\_DEPARTURE" DATE,

"DATE\_OF\_RETURN" DATE,

"NUMBER\_OF\_DEPENDENTS" NUMBER,

"HOTEL\_ID" VARCHAR2(240 BYTE),

"TRAVEL\_COST" NUMBER,

"HOTEL\_COST" NUMBER,

"TOTAL\_COST" NUMBER,

PRIMARY KEY (“ITINERARY\_ID"),

FOREIGN KEY("USER\_ID") REFERENCES “HR”.”USERS”( "USER\_ID"),

FOREIGN KEY("HOTEL\_ID") REFERENCES “HR”.” HOTEL"(” HOTEL\_ID")

)

3) CREATE TABLE "HR"."HOTEL"

(

"REGION\_ID" NUMBER,

"HOTEL\_ID" VARCHAR2(240 BYTE),

"HOTEL\_NAME" VARCHAR2(240 BYTE) NOT NULL ENABLE,

"ADDRESS\_LINE1" VARCHAR2(240 BYTE),

"ADDRESS\_LINE2" VARCHAR2(240 BYTE),

"CITY" VARCHAR2(240 BYTE),

"STATE" VARCHAR2(2 BYTE),

"ZIP\_CODE" NUMBER,

"PHONE\_NUMBER" VARCHAR2(50 BYTE),

"PRICE\_PER\_NIGHT" VARCHAR2(100 BYTE),

"USER\_RATINGS" NUMBER,

PRIMARY KEY ("HOTEL\_ID"),

FOREIGN KEY ("REGION\_ID") REFERENCES "HR"."REGION" ("REGION\_ID") ENABLE

)

4) CREATE TABLE "HR"."REGION"

( "REGION\_ID" NUMBER,

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

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

"LATTITUDE" NUMBER(10,7),

"LONGITUDE" NUMBER(10,7),

PRIMARY KEY ("REGION\_ID")

)

5) CREATE TABLE "HR"."OFFERS"

( "OFFER\_ID" NUMBER,

"OFFER\_CODE" VARCHAR2(100 BYTE),

"DISCOUNT\_OFFERED" VARCHAR2(100 BYTE),

"REQUIREMENT" VARCHAR2(100 BYTE),

PRIMARY KEY ("OFFER\_ID")

)

6) CREATE TABLE "HR"."TECHNICAL\_SUPPORT"

( "EMPLOYEE\_ID" NUMBER,

"EMPLOYEE\_NAME" VARCHAR2(240 BYTE),

"SEX" VARCHAR2(1 BYTE),

"CONTACT" NUMBER,

"TECH\_LEVEL" VARCHAR2(240 BYTE),

PRIMARY KEY ("EMPLOYEE\_ID")

)

7) CREATE TABLE "HR"."USER\_SUPPORT"

( "EMPLOYEE\_ID" NUMBER,

"EMPLOYEE\_NAME" VARCHAR2(240 BYTE),

"SEX" VARCHAR2(1 BYTE),

"CONTACT" NUMBER,

"AREA\_OF\_COMPLAINT" VARCHAR2(240 BYTE),

PRIMARY KEY ("EMPLOYEE\_ID")

)

8) CREATE TABLE "HR"."USERS"

( "USER\_ID" NUMBER,

"NAME" VARCHAR2(100 BYTE),

"ADDRESS" VARCHAR2(100 BYTE),

"EMAIL\_ID" VARCHAR2(50 BYTE) NOT NULL ENABLE,

"CONTACT\_NO" VARCHAR2(10 BYTE) NOT NULL ENABLE,

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

PRIMARY KEY ("USER\_ID")

)

9) CREATE TABLE "HR"."MODE\_OF\_TRANSPORT"

( "MODE\_ID" NUMBER,

"MODE\_TYPE" VARCHAR2(100 BYTE),

PRIMARY KEY ("MODE\_ID")

)

10) CREATE TABLE "HR"."MODE\_AIR"

( "MODE\_ID" NUMBER,

"MODE\_TYPE" VARCHAR2(100 BYTE),

"SERVICE\_ID" VARCHAR2(240 BYTE),

"SERVICE\_NAME" VARCHAR2(240 BYTE),

"PRICE\_PER\_MILE" VARCHAR2(100 BYTE),

PRIMARY KEY ("SERVICE\_ID"),

FOREIGN KEY (“MODE\_ID”) REFRENCES “HR”.”MODE\_OF\_TRANSPORT”(“MODE\_ID”)

)

11) CREATE TABLE "HR"."MODE\_ROAD"

( "MODE\_ID" NUMBER,

"MODE\_TYPE" VARCHAR2(100 BYTE),

"SERVICE\_ID" VARCHAR2(240 BYTE),

"SERVICE\_NAME" VARCHAR2(240 BYTE),

"PRICE\_PER\_MILE" VARCHAR2(100 BYTE),

PRIMARY KEY ("SERVICE\_ID"),

FOREIGN KEY (“MODE\_ID”) REFRENCES “HR”.”MODE\_OF\_TRANSPORT”(“MODE\_ID”)

)

12) CREATE TABLE "HR"."MODE\_TRAIN"

( "MODE\_ID" NUMBER,

"MODE\_TYPE" VARCHAR2(100 BYTE),

"SERVICE\_ID" VARCHAR2(240 BYTE),

"SERVICE\_NAME" VARCHAR2(240 BYTE),

"PRICE\_PER\_MILE" VARCHAR2(100 BYTE),

PRIMARY KEY ("SERVICE\_ID")

FOREIGN KEY (“MODE\_ID”) REFRENCES “HR”.”MODE\_OF\_TRANSPORT”(“MODE\_ID”)

)

13) CREATE TABLE "HR"."CUSTOMER\_SUPPORT"

( "EMPLOYEE\_ID" NUMBER,

"EMPLOYEE\_NAME" VARCHAR2(240 BYTE),

"SEX" VARCHAR2(1 BYTE),

"CONTACT" NUMBER,

PRIMARY KEY ("EMPLOYEE\_ID")

)

**CONCLUSION**

Thus, a database is implemented to support a web application. Having a web based front end removes the requirement of users having to understand and use a database directly, and allows users to connect from anywhere with an internet connection and a basic web browser. The database provides an efficient mechanism for data storage and to execute complex functions. It can also be used for generating reports at a later stage.

During our database management course, we have learned about the basics of database design and its implementation. This project gave us an opportunity to put our new skills into practice. While doing this project, we gained deeper understanding on database design and how it can provide a simple solution to an industrial problem so as to drive more conversions from the users.

**REFERENCES**

* Fundamentals\_of\_Database\_Systems\_6th\_Edition by Elmasri, Navathe
* <http://www.tutorialspoint.com/plsql/>
* https://www.tutorialspoint.com/java/