view_place : Booking.Free Taxi : SELECT "TID","PlateNum" FROM taxi_booking.taxi WHERE "isFree" = TRUE;

Free Taxi

TAXI

(s_id,t_id)

C

SESSION_TAXI

(s_id,t_id)

(t_id,pn)

S_ID

s_id

Create Booking

s_id

A

s_id

Leave Phone Number

(s_id, v_n)

D

(s_id,v_n)

Finalize Booking

INT

INT

SESSION_PHONE

s_id

s_id

S_ID Generator

input ();
output (s_id);
action
let
val x = getSessionID();
val SOME s_id = Int.fromString x
in
INSERTSESSION(x);
(s_id)
end;

input (t_id);
output ();
action
let
in
RESERVE(t_id)
end;

input ();
output (v_n);
action
let
val v_n = getPhoneNr()
in
(v_n)
end;

input (t_id,v_n,v_a,v_t);
output (b_id,pd_id,ph_id);
action
let
val x1 = getBookingID()
val x2 = getPickupID()
val x3 = getPhoneID()
val SOME b_id = Int.fromString x1
val SOME pd_id = Int.fromString x2
val SOME ph_id = Int.fromString x3
in
INSERTPHONE(x3,v_n);
INSERTPICKUP(x2,v_a,v_t);
ADDBOOKING(x1,t_id,x2,x3);
(b_id,pd_id,ph_id)
end;

(s_id,v_a,v_t)

B

s_id

Leave Pickup Data

(s_id,v_a,v_t)

E

INT

SESSION_PICKUP

(b_id,t_id,pd_id,ph_id)

Bookings

BOOKING

connectDB("Taxi_Connection",9000)

disconnectDB("Taxi_Connection")

input ();
output (v_a,v_t);
action
let
val v_a = getAddress()
val v_t = getTime()
in
(v_a,v_t)
end;

colset UNIT = unit;
colset INT = int;
colset STRING = string;
colset SESSION_TAXI = product INT*INT;
colset TAXI = product INT*STRING;
colset SESSION_PHONE = product INT*STRING;
colset BOOKING = product INT*INT*INT*INT;
colset SESSION_PICKUP = product INT*STRING*STRING;
colset SESSION = INT;

var t_id,s_id, limit: INT;
var pn, v_n, v_a:STRING;