### 

Deliverable 1

Team 2

Wonjin Choi (Member 1)

Jagjivan Singh Deol (Member 4)

Nimesh D Desilva (Member 3)

Youssef Mohamed Elalamy (Member 2)

\*Member 5 has been split up by all four members of the group

### 

**CREATE TABLES**

**Customer table**

Create table customer(

Customer\_ID integer NOT NULL

Customer\_first\_name varchar2(30)

Customer\_last\_name varchar2(30)

Customer\_phone integer

Customer\_address varchar2(50

Customer\_city varchar2(20)

Customer\_state varchar2(2)

Credit\_card varchar2(16)

PRIMARY KEY (Customer\_ID)

);

**Hotel table**

CREATE table hotel(

hotel\_id integer NOT NULL,

hotel\_name varchar2(50),

hotel\_address varchar2(50),

hotel\_address2 varchar2(30),

hotel\_state varchar2(2),

hotel\_zip number,

hotel\_city varchar2(30),

hotel\_phone number,

Hotel\_room\_type varchar2(50),

Hotel\_capacity integer,  
Hotel\_total\_cost integer,

Hotel\_room\_available number,

Customer\_id integer,

Event\_id integer,

Service\_id integer,

Reservation\_id integer,

PRIMARY KEY (hotel\_id),

Foreign key (customer\_id) references customer(customer\_id),

Foreign key(event\_id) references event(event\_id),

Foreign key(service\_id) references service(service\_id)

Foreign key (reservation\_id) references reservation(reservation\_id)

);

**Event table**

Create table Event(

Event\_ID integer NOT NULL,

Event\_name varchar(30),

Event\_type varchar(30),

Room\_type varchar(30),

Room\_id integer,

Num\_of\_rooms integer,

Event\_price integer,

Event\_amt\_days number,

Customer\_ID integer,

Hotel\_id integer,

Service\_id integer,

Total\_capacity integer,

Num\_of\_Customer integer,

start\_date date,

end\_date date,

PRIMARY KEY (Event\_ID),

Foreign key(hotel\_id) references hotel(hotel\_id),

Foreign key(customer\_id) references customer(customer\_id),

Foreign key(service\_id) references service(service\_id)

);

**Reservation table**

Create table Reservation(

Reservation\_ID integer NOT NULL,

Customer\_ID integer,

Room\_type varchar2(30),

Room\_id integer,

Num\_of\_rooms integer,

Service\_include varchar2(30),

Check\_in\_date timestamp,

Check\_out\_date timestamp,

Reservation\_price integer,

Reservation\_cancel varchar2(10),

Date\_of\_reserve date,

Hotel\_id integer,

Service\_id integer,

Event\_id integer,

PRIMARY KEY (Reservation\_ID),

FOREIGN KEY (hotel\_id) references hotel (hotel\_id),

Foreign key(service\_id) references service(service\_id)

Foreign key(event\_id) references event(event\_id)

);

**Service Table**

Create Table Service(

Service\_ID integer NOT NULL,

Service\_Type varchar2(30),

Num\_of\_Services integer,

Service\_price integer,

Customer\_ID integer,

Event\_id integer,

Reservation\_id integer,  
PRIMARY KEY (Service\_ID),

Foreign key (reservation\_id) references reservation(reservation\_id)

Foreign key(event\_id) references event(event\_id)

);

**Income Report**

Create table Income(

Income\_IDinteger

Event\_price integer,

Reservation\_price integer,

Service\_price integer

Total\_price integer

Service\_id integer,

Event\_id integer,

Primary key(Income\_id),

Foreign key(service\_id) references service(service\_id)

Foreign key(event\_id) references service(event\_id)

);

Member 1: **James Choi**

1. Add a new hotel: Create a new hotel with appropriate information about the hotel as input parameters. Minimum parameters are Address, Phone, Event Room Types available etc. Have in mind that you will need more parameters.

**Create or replace procedure New\_hotel(**

**Hotel\_address as varchar2,**

**Hotel\_Phone as number,**

**event \_room\_type as varchar2  
)**

2. Find a hotel: Provide as input the address of the hotel and return its hotel ID

**Create or replace function hotel\_search(**

**Hotel\_Address in varchar2**

**)**

**Return Integer IS**

**Hotel\_id in integer;**

3. Display hotel info: Given a hotel ID, display all information about that hotel

**Create or replace function display\_hotel\_info(**

**Hotel\_id in integer**

**)**

**DBMS\_OUTPUT.PUT\_LINE (‘ Hotel Location: ‘ || hotel\_address varchar || hotel\_address2 varchar || hotel\_state varchar || hotel\_zip number || hotel\_city varchar || hotel\_phone number || hotel\_room\_type varchar|| hotel\_capacity integer);**

4. Add Event Room: Given a hotel ID, add a new room for a specific event to that hotel. The room types are: Small Hall, Medium hall, Large hall.

**Create or replace procedure add\_event\_room (**

**Hotel\_id in integer,**

**)**

**Return Integer is**

**Room\_type in varchar**

5.Report Hotels and Event Rooms In State: Given a state, display event room information of all hotels in that particular state. Include total capacity per event room type per hotel.

**Create or replace function report\_rooms**

**( hotel\_state in varchar2)**

**Event\_price in integer,**

**Total\_capacity in integer,**

**DBMS\_OUTPUT.PUT.LINE( ‘ Total capacity: ’|| total\_capacity|| ‘Per hotel’ );**

Member 2: **Youssef Elalamy**

6. Make an event reservation: Input parameters: Hotel ID, guest’s name, start date, end date, event type, date of reservation, number of people attending, etc. Output: event reservation ID (this is called confirmation code in real-life). NOTE: Only one person can make an event reservation. However, the same person can make multiple reservations. Event types: Birthday, Wedding, Conference, Workshop, Hackathon, University Admission, etc. Also make sure that the reserved hall has capacity that can hold the number of people attending. For example, for a conference of 500 people, a customer must reserve 2 medium halls and a large hall for each day of the conference, usually 3 consecutive days.

**CREATE OR REPLACE FUNCTION reserve\_event(**

**hotel\_id IN integer,**

**customer\_first\_name IN varchar,**

**customer\_last\_name IN varchar,**

**start\_date IN date,**

**end\_date IN date,**

**event\_type IN varchar,**

**date\_of\_reserve IN date,**

**num\_of\_customers IN integer**

**)**

**RETURN INTEGER IS**

**Reservation\_ID integer;**

7. Find an event reservation: Input is a person’s name and date, hotel ID. Output is event reservation ID

**CREATE OR REPLACE FUNCTION reserve\_event\_finder(**

**customer\_first\_name IN varchar,**

**customer\_last\_name IN varchar,**

**date\_of\_reserve IN date,**

**hotel\_id IN integer**

)

**RETURN INTEGER IS**

**Reservation\_ID integer;**

8. Cancel an event: Input the event reservationID and mark the reservation as cancelled (do NOT delete it)

**CREATE OR REPLACE FUNCTION event\_cancel(**

**Reservation\_ID IN integer**

**)**

**RETURN INTEGER IS**

**Reserveration\_cancel IN varchar2;**

9. ShowCancelations: Print all canceled events in the event management system. Show event reservation ID, hotel name, location, event type, room type, dates

**CREATE OR REPLACE PROCEDURE cancellist AS**

**CURSOR cursor\_cancel IS**

**…**

**DBMS\_OUTPUT.PUT\_LINE (‘ Hotel Name: ‘ || hotel\_name varchar2);**

**DBMS\_OUTPUT.PUT\_LINE (‘ Hotel Location: ‘ || hotel\_address varchar2 || hotel\_address2 varchar2 || hotel\_state varchar2 || hotel\_zip number || hotel\_city varchar2);**

**DBMS\_OUTPUT.PUT\_LINE (‘ Event Type: ‘ || Event\_type varchar2);**

**DBMS\_OUTPUT.PUT\_LINE (‘ Room Type: ‘ || Room\_type varchar2);**

**DBMS\_OUTPUT.PUT\_LINE (‘ Dates: ‘ || start\_date date || end\_date date);**

Member 3: **Nimesh De Silva**

10. Change an event Date: Input the event reservation ID and change event start and end date, if there is availability in the same or larger room type for the new date interval

**CREATE OR REPLACE FUNCTION change\_event\_date(**

**Reservation\_ID IN integer**

**Start\_Date IN date**

**End\_Date IN date**

**)**

**DBMS\_OUTPUT.PUT\_LINE (‘ Event Name: ‘ || event\_name varchar2);**

**DBMS\_OUTPUT.PUT\_LINE (‘ Event Date: ‘ || start\_date date || ‘-’|| end\_date date);**

11. Change an event RoomType: Input the reservation ID and change reservation room type if there is availability for the new room type during the reservation’s date interval

**CREATE OR REPLACE FUNCTION change\_event\_date(**

**Reservation\_ID IN integer**

**)**

**AS CURSOR ced IS**

**DBMS\_OUTPUT.PUT\_LINE (‘ Room Type: ‘ || room\_type varchar2);**

12. Show specific event: Given an event type (Birthday, Wedding, etc.) display all events of that type in all hotels along with the address of the hotel and the date of the event.

**CREATE OR REPLACE FUNCTION display\_events(**

**Event\_Type in varchar(20)**

**)**

**AS CURSOR de IS**

**DBMS\_OUTPUT.PUT\_LINE (‘ Event Name: ‘ || event\_name varchar2);**

**DBMS\_OUTPUT.PUT\_LINE (‘ Event Date: ‘ || start\_date date || ‘-’|| end\_date date);**

**DBMS\_OUTPUT.PUT\_LINE (‘ Hotel Name: ‘ | hotel\_name varchar2);**

**DBMS\_OUTPUT.PUT\_LINE (‘ Hotel Location: ‘ || hotel\_address varchar2 || hotel\_address2 varchar2 || hotel\_state varchar2 || hotel\_zip number || hotel\_city varchar2);**

13. Show events by person: Given a person’s name, find all events under that name

**CREATE OR REPLACE FUNCTION display\_customer\_events(**

**Customer\_first\_name in varchar2,**

**Customer\_last\_name in varchar2)**

**)**

**AS CURSOR dce IS**

**DBMS\_OUTPUT.PUT\_LINE (‘Event Name: ‘|| event\_name varchar2);**

14. Total Monthly Income Report: Calculate and display total income from all sources of all hotels. Totals must be printed by month, and for each month by event and service type. Include discounts.

**CREATE OR REPLACE FUNCTION calculate\_total\_income(**

**)**

**AS CURSOR cti IS**

**DBMS\_OUTPUT.PUT\_LINE (‘Month: ‘ || month date);**

**DBMS\_OUTPUT.PUT\_LINE (‘Event Type: ‘ || event\_type varchar2);**

**DBMS\_OUTPUT.PUT\_LINE (‘Service Type: ‘ || service\_type varchar2);**

**DBMS\_OUTPUT.PUT\_LINE (‘Total: ‘ || event\_total number);**

Member 4: **Jagjivan Singh Deol**

15. Add a service to an event: Input: Event reservationID, specific service. Add the service to the event for a particular date. Multiple services are allowed on a reservation for the same date. For meals make sure to multiply the amount by the number of people attending the event.

**CREATE OR REPLACE FUNCTION service\_event(**

**Reservation\_ID IN integer,**

**Service\_Type In Varchar2,**

**RETURN INTEGER IS**

**Service\_Type IN varchar2;**

**)**

16. Reservation Services Report: Input the event reservation ID and display all services on this reservation. Also print the number of attendees of the event. Print “no services for this reservation” if none exists.

**CREATE OR REPLACE FUNCTION reservation\_services\_report(**

**Reservation\_ID IN integer,**

**)**

**DBMN\_OUTPUT.PUT\_LINE (‘Services is as follows: ‘||service\_type);**

**DBMN\_OUTPUT.PUT\_LINE (‘Number of attendees’ || )**

17. Show Specific Service Report: Input the service name, and display information on all reservations that have this service in all hotels

**CREATE OR REPLACE FUNCTION service\_report(**

**Service\_name IN varchar2,**

**RETURN INTEGER IS**

**Reservation\_ID integer**

**)**

18. Services Income Report: Given a hotelID, calculate and display income from all services in all reservations in that hotel.

**CREATE OR REPLACE FUNCTION income\_report(**

**Create table Income(**

**Hotel\_ID**

**RETURN INTEGER IS**

**Income\_ID integer**

**)**

Member 5: **Split up**

19. Show available rooms by type: Given a hotel ID, display the count of all available rooms by room type. **(Youssef Elalamy)**

**CREATE OR REPLACE PROCEDURE room\_type-availability(**

**Hotel\_id IN integer**

**)**

**AS CURSOR rta IS**

**…..**

**DBMS\_OUTPUT.PUT\_LINE ( Count (hotel\_room\_available varchar) );**

20. Event Invoice: Input: Event reservationID Output: **(James Choi)** Name of person that reserved the event  
 Event room number(s), rate per day and possibly multiple rooms (if someone reserved several rooms)   
 Services rendered per date, type, and amount  
 Discounts applied (if any)  
 Total amount to be paid

**CREATE OR REPLACE FUNCTION (**

**Event\_reservation\_id integer**

**)**

**AS Cursor ibsr IS**

**DBMS\_OUTPUT.PUT\_LINE (‘Customer Name: ‘||customer\_first\_name|| ‘ ‘|| customer\_last\_name);**

**DBMS\_OUTPUT.PUT\_LINE (‘Event Room Number: ‘||event\_room\_number number);**

**DBMS\_OUTPUT.PUT\_LINE (‘Rate per day: ‘||rate\_per\_day number);**

**DBMS\_OUTPUT.PUT\_LINE(‘Service-rendered-per-day: ‘||service\_rendered\_per\_day number);**

**DBMS\_OUTPUT.PUT\_LINE (‘Type: ‘|| type varchar2);**

**DBMS\_OUTPUT.PUT\_LINE (‘Amount: ‘|| amount number);**

**DBMS\_OUTPUT.PUT\_LINE (‘Discount: ‘|| discount number);**

**DBMS\_OUTPUT.PUT\_LINE (‘Total Paid: ‘|| paid number);**

21.Income by State Report: Input is a specific state. Print total income from all events as follows: Each output line should contain information of a specific event ID (income from room type, income from services, total income of this event). At the end a grand total of all events income. Include discounts.

(**NIMESH DE SILVA**)

**CREATE OR REPLACE PROCEDURE income\_by\_state\_report(**

**Specific\_state varchar2)**

**AS Cursor ibsr IS**

**DBMS\_OUTPUT.PUT\_LINE (‘Event ID: ‘||event\_id integer);**

**DBMS\_OUTPUT.PUT\_LINE (‘Income from room type: ‘||event\_id number);**

**DBMS\_OUTPUT.PUT\_LINE (‘Income from services: ‘||event\_id number);**

**DBMS\_OUTPUT.PUT\_LINE (‘Total Income: ‘||total\_income number);**

**DBMS\_OUTPUT.PUT\_LINE (‘Grand Total: ‘|| grand\_total number);**