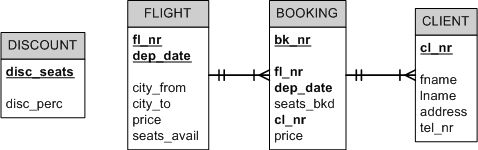
|  |  |  |
| --- | --- | --- |
| **26 MARCH 2012** | **DEVELOPMENT SOFTWARE III** | **2 HOURS** |
| **54 MARKS** | **FULL-TIME** |  |

***ORACLE PL/SQL must be used as the programming language.***

Below is a subset of a database design for an airline company. The database uses four tables to manage the bookings of flights.



|  |  |  |
| --- | --- | --- |
| **FLIGHT** | Contains flight details | |
| fl\_nr | varchar2 (7) | Part of primary key. |
| dep\_date | date | Part of primary key. The departure date for this flight number. |
| city\_from | varchar2(20) | The city the flight departs from. |
| city\_to | varchar2(20) | The arrival city for the flight. |
| price | Number(7,2) | The price per seat |
| seats\_avail | Number(3) | The number of seats available on this flight. When a booking is confirmed, the number of available seats are reduced. |

|  |  |  |
| --- | --- | --- |
| **CLIENT** | Contains client details | |
| cl\_nr | number(5) | Primary key |
| fname | varchar2(20) | First name |
| lname | varchar2(20) | Last name |
| address | varchar2(40) | Address |
| tel\_nr | varchar2(13) | Telephone number |

|  |  |  |
| --- | --- | --- |
| **BOOKING** | Contains details of every booking | |
| bk\_nr | number(4) | Primary key. Use the sequence booking\_nr\_seq. |
| fl\_nr | varchar2(7) | Foreign key, to the flight table. |
| dep\_date | date | Foreign key, to the flight table. |
| seats\_bkd | Number(3) | Number of seats booked for this booking |
| cl\_nr | Number(5) | Foreign key, to the client table. |
| price | Number(9,2) | Price paid by client for all the seats |

|  |  |  |
| --- | --- | --- |
| **DISCOUNT** | Contains details of the discount used when more than 10 seats are booked. | |
| disc\_seats | Number(3) | Primary key |
| disc\_perc | Number(4,2) | Discount percentage |

The discount table currently have only three rows and the data is as follows:

|  |  |
| --- | --- |
| **disc\_seats** | **disc\_perc** |
| 10 | 15 |
| 20 | 20 |
| 30 | 25 |

The data in the discount must be interpreted as follows.

When the number of seats booked are:

* Less than 10, then no discount is given.
* Between 10 and 19 inclusive, then a discount of 15% is given.
* Between 20 and 29 inclusive, then a discount of 20% is given.
* 30 or more, then discount given is 25%.

**QUESTION 1 (9 MARKS)**

Create a procedure, named DISPLAY\_CLIENT, that display the client details on the screen. It receives a client number as an input parameter. Display the output on the screen as follows (the output contains spaces and upper and lower case):

Client number: *cl\_nr*

Client name: *lname fname*

Client address: *address*

Handle only the appropriate exception.

**QUESTION 2 (9 MARKS)**

Create a procedure, named FIND\_DISCOUNT, that receives the number of seats booked as an input parameter, and the returns the discount percentage through an output parameter. Use the DISCOUNT table to retrieve the discount percentage.

**QUESTION 3 (8 MARKS)**

Write a procedure, named CALC\_BOOKING\_PRICE, that calculates the total price for a booking. The procedure receives the number of seats booked and the price per seat as input parameters. The total booking price is returned through an output parameter. Use the previously coded procedure, FIND\_DISCOUNT.

Display the total booking price on the screen, with an appropriate message.

**QUESTION 4 (22 MARKS)**

Write a procedure, named ADD\_BOOKING, that inserts a new row to the booking table. It receives the flight number, departure date, number of seats booked and the client number as input parameters. When a new booking is inserted into the booking table, the number of seats available on that flight must also be reduced.

The following exceptions must be handled by using the RAISE\_APPLICATION\_ERROR procedure:

* If the number of seats received for the specific flight, is more than the number of seats available on that flight, an exception E\_SEATS must be raised.
* For integrity error, -2291, and exception E\_INVALID\_FK, must be handled.
* Also handle any NO\_DATA\_FOUND exceptions.

Give a statement to test this procedure. Use data that will ensure that a row is inserted into the booking table.

**QUESTION 5 (6 MARKS)**

Write a procedure that deletes all flight details from all associated tables in the database. It receives the flight number and departure date as an input parameter. Display the number of rows deleted from every table on the screen.

\*\*\*\*\*\*END OF PAPER\*\*\*\*\*\*\*