**SQL> --**

**SQL> -- ---------------------------------------------------------**

**SQL> -- COMP 2714**

**SQL> -- SET 2C**

**SQL> -- LAB01 Exercise / Assignment Asn01**

**SQL> -- Herradura, Lancelei A00960501**

**SQL> -- Kobunnoi, Kunlaya A00959419**

**SQL> -- email: lherradura@my.bcit.ca**

**SQL> -- email: kkobunnoi@my.bcit.ca**

**SQL> -- ---------------------------------------------------------**

**SQL> -- START C:\Users\Kunlaya\workspace\COMP2714\Assignments\Asn1\_HerraduraL\_KobunnoiK.sql**

**SQL> -- ---------------------------------------------------------**

**SQL> --**

**SQL> ALTER SESSION SET NLS\_DATE\_FORMAT='YYYY-MM-DD';**

**Session altered.**

**SQL> SELECT SYSDATE**

**2 FROM DUAL;**

**SYSDATE**

**----------**

**2016-09-21**

**SQL> -- ---------------------------------------------------------**

**SQL> -- Questions: Given the following database tables required**

**SQL> --**

**SQL> -- Hotel (hotelNo, hotelName, city)**

**SQL> -- Room (roomNo, hotelNo, type, price)**

**SQL> -- Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)**

**SQL> -- Guest (guestNo, guestName, guestAddress)**

**SQL> -- --------------------------------------------------------**

**SQL> -- Drop tables first before creating**

**SQL> -- ---------------------------------------------------------**

**SQL> -- Q1) (5 marks) Write the DDL statement to remove all 4 tables,**

**SQL> -- plus the OldBooking archive table.**

**SQL> -- ---------------------------------------------------------**

**SQL> DROP TABLE OldBooking;**

**Table dropped.**

**SQL> DROP TABLE Booking;**

**Table dropped.**

**SQL> DROP TABLE Guest;**

**Table dropped.**

**SQL> DROP TABLE Room;**

**Table dropped.**

**SQL> DROP TABLE Hotel;**

**Table dropped.**

**SQL> -- ---------------------------------------------------------**

**SQL> -- Q2) Create the Hotel (from Q.7.10 - 3 marks) and Room (7 marks)**

**SQL> -- tables with all relevant integrity constraints, including:**

**SQL> --**

**SQL> -- a. type (Room table) must be one of Single, Double, or Family.**

**SQL> -- b. price (Room table) must be between $10.00 and $100.00.**

**SQL> -- c. room number (Room table) must be between 1 and 100.**

**SQL> -- ---------------------------------------------------------**

**SQL> --CREATE Hotel Table**

**SQL> CREATE TABLE Hotel**

**2 (hotelNo CHAR(8) NOT NULL**

**3 ,hotelName VARCHAR2(20) NOT NULL**

**4 ,city VARCHAR2(30) NOT NULL**

**5 ,CONSTRAINT PKHotel PRIMARY KEY (hotelNo)**

**6 )**

**7 ;**

**Table created.**

**SQL> --CREATE Room Table**

**SQL> CREATE TABLE Room**

**2 (roomNo CHAR(8) NOT NULL**

**3 ,hotelNo CHAR(8) NOT NULL**

**4 ,type VARCHAR2(6) NOT NULL**

**5 ,price DECIMAL(5,2) NOT NULL**

**6 ,CONSTRAINT PKRoom PRIMARY KEY (roomNo)**

**7 ,CONSTRAINT FKRoom FOREIGN KEY (hotelNo) REFERENCES Hotel (hotelNo)**

**8 ,CONSTRAINT chkType CHECK(type IN('Single', 'Double', 'Family'))**

**9 ,CONSTRAINT chkPrice CHECK(price BETWEEN 10 AND 100)**

**10 ,CONSTRAINT chkRoomNo CHECK(roomNo BETWEEN 1 AND 100)**

**11 )**

**12 ;**

**Table created.**

**SQL> -- ---------------------------------------------------------**

**SQL> -- Q3) Create the Guest (from Q.7.11 - 3 marks) and Booking**

**SQL> -- (7 marks) tables with all relevant integrity constraints.**

**SQL> -- ---------------------------------------------------------**

**SQL> -- Create Guest Table**

**SQL> CREATE TABLE Guest**

**2 (guestNo CHAR(8) NOT NULL**

**3 ,guestName VARCHAR2(30) NOT NULL**

**4 ,guestAddress VARCHAR2(75) NOT NULL**

**5 ,CONSTRAINT PKGuest PRIMARY KEY (guestNo)**

**6 )**

**7 ;**

**Table created.**

**SQL> -- Create Booking Table**

**SQL> CREATE TABLE Booking**

**2 (hotelNo CHAR(8) NOT NULL**

**3 ,guestNo CHAR(8) NOT NULL**

**4 ,dateFrom DATE NOT NULL**

**5 ,dateTo DATE NOT NULL**

**6 ,roomNo CHAR(8) NOT NULL**

**7 ,CONSTRAINT PKBooking PRIMARY KEY (hotelNo,guestNo,dateFrom)**

**8 ,CONSTRAINT FKBooking FOREIGN KEY (hotelNo) REFERENCES Hotel (hotelNo)**

**9 ,CONSTRAINT FKBooking2 FOREIGN KEY (guestNo) REFERENCES Guest (guestNo)**

**10 ,CONSTRAINT CKRoomNo CHECK (roomNo >= 1 OR roomNo <= 100)**

**11 )**

**12 ;**

**Table created.**

**SQL> -- ---------------------------------------------------------**

**SQL> -- Q4) Insert 3 rows of sample data to each of the 4 tables.**

**SQL> -- ---------------------------------------------------------**

**SQL> -- Insert HOTEL Data**

**SQL> INSERT INTO Hotel**

**2 VALUES('J001','Marriot','Vancouver')**

**3 ;**

**1 row created.**

**SQL> INSERT INTO Hotel**

**2 VALUES('J002','Inn','Chilliwack')**

**3 ;**

**1 row created.**

**SQL> INSERT INTO Hotel**

**2 VALUES('J003','Fraser','Victoria')**

**3 ;**

**1 row created.**

**SQL> --Insert Room Data**

**SQL> INSERT INTO Room**

**2 VALUES ('9','J001','Single',19.00)**

**3 ;**

**1 row created.**

**SQL> INSERT INTO Room**

**2 VALUES ('40','J002','Double',30.00)**

**3 ;**

**1 row created.**

**SQL> INSERT INTO Room**

**2 VALUES ('80','J003','Family',50.00)**

**3 ;**

**1 row created.**

**SQL> --Insert Guest Data**

**SQL> INSERT INTO Guest**

**2 VALUES ('00111100','Dan','402 Fraser st')**

**3 ;**

**1 row created.**

**SQL> INSERT INTO Guest**

**2 VALUES ('00111102','Ben','521 Rupert st')**

**3 ;**

**1 row created.**

**SQL> INSERT INTO Guest**

**2 VALUES ('00111104','Hannah','109 Willindon Ave.')**

**3 ;**

**1 row created.**

**SQL> --Insert Booking Data**

**SQL> INSERT INTO Booking**

**2 VALUES ('J001', '00111100', DATE'2016-04-01', DATE'2016-04-04','9')**

**3 ;**

**1 row created.**

**SQL> INSERT INTO Booking**

**2 VALUES ('J002', '00111102', DATE'2016-05-05', DATE'2016-05-10','80')**

**3 ;**

**1 row created.**

**SQL> INSERT INTO Booking**

**2 VALUES ('J003', '00111104', DATE'2016-10-10', DATE'2016-10-15','40')**

**3 ;**

**1 row created.**

**SQL> COMMIT;**

**Commit complete.**

**SQL> --**

**SQL> -- ---------------------------------------------------------**

**SQL> -- Q5) a. The hotels now have a new room type 'Deluxe'. Modify**

**SQL> -- the check constraint to allow for this.**

**SQL> --**

**SQL> -- b. Need to offer discounts. Add column 'discount' to the**

**SQL> -- Room table, with default value of 0 (percent), max. value of 30.**

**SQL> -- ---------------------------------------------------------**

**SQL> -- Q5a)**

**SQL> ALTER TABLE Room**

**2 DROP CONSTRAINT chkType**

**3 ;**

**Table altered.**

**SQL> ALTER TABLE Room**

**2 ADD CONSTRAINT chkType CHECK(type IN('Single', 'Double','Family', 'Deluxe'))**

**3 ;**

**Table altered.**

**SQL> -- Q5b) Add column discount to table Room**

**SQL> ALTER TABLE Room**

**2 ADD discount DECIMAL(5,2) DEFAULT 0.0 NOT NULL;**

**Table altered.**

**SQL> ;**

**1 ALTER TABLE Room**

**2\* ADD discount DECIMAL(5,2) DEFAULT 0.0 NOT NULL**

**SQL> ALTER TABLE Room**

**2 MODIFY discount DEFAULT 0**

**3 ADD CONSTRAINT chkDiscount CHECK(discount BETWEEN 0 and 30)**

**4 ;**

**Table altered.**

**SQL> -- ---------------------------------------------------------**

**SQL> -- Q6) a. One of the hotels (pick any one) increases the price of**

**SQL> -- its 'Double' room type by 15%. Implement this change.**

**SQL> --**

**SQL> -- b. One of the booking guest decided to arrive on an earlier date**

**SQL> -- and stay till a later dater. Modif the booking accordingly**

**SQL> -- ---------------------------------------------------------**

**SQL> --Q6a) Increase price of room type Double**

**SQL> UPDATE Room**

**2 SET price = price \* 1.15**

**3 WHERE type = 'Double'**

**4 AND hotelNo = 'J002'**

**5 ;**

**1 row updated.**

**SQL> -- Check if price increased**

**SQL> SELECT \***

**2 FROM Room**

**3 WHERE hotelNo = 'J002';**

**ROOMNO HOTELNO TYPE PRICE DISCOUNT**

**-------- -------- ------ ---------- ----------**

**40 J002 Double 34.5 0**

**SQL> -- Q6b) Modify booking for guest who arrived earlier than expected date**

**SQL> -- and left later than expected**

**SQL> UPDATE Booking**

**2 SET dateFrom = DATE'2016-03-29'**

**3 WHERE guestNo = '00111100';**

**1 row updated.**

**SQL> UPDATE Booking**

**2 SET dateTo = DATE'2016-04-05'**

**3 WHERE guestNo = '00111100';**

**1 row updated.**

**SQL> -- Check if date was changed**

**SQL> SELECT \* FROM Booking;**

**HOTELNO GUESTNO DATEFROM DATETO ROOMNO**

**-------- -------- ---------- ---------- --------**

**J001 00111100 2016-03-29 2016-04-05 9**

**J002 00111102 2016-05-05 2016-05-10 80**

**J003 00111104 2016-10-10 2016-10-15 40**

**SQL> --**

**SQL> -- ---------------------------------------------------------**

**SQL> -- Q7) a. Create an OldBooking table to hold archive data rows from**

**SQL> -- the Booking table (from Q.7.12 - 2 marks).**

**SQL> --**

**SQL> -- b. Using the INSERT statement, copy the rows with dateTo before**

**SQL> -- 2016-09-01 from the Booking table to the OldBooking archive**

**SQL> -- table (from Q.7.12 - 2 marks).**

**SQL> --**

**SQL> -- c. Then delete from Booking all booking rows with dateTo**

**SQL> -- before 2016-09-01 (2 marks).**

**SQL> -- --------------------------------------------------------**

**SQL> -- Q7a) Create OldBooking Archive Table**

**SQL> CREATE TABLE OldBooking**

**2 (hotelNo CHAR(8) NOT NULL**

**3 ,guestNo CHAR(8) NOT NULL**

**4 ,dateFrom DATE NOT NULL**

**5 ,dateTo DATE NULL**

**6 ,roomNo CHAR(8) NOT NULL**

**7 )**

**8 ;**

**Table created.**

**SQL> -- Q7b)**

**SQL> INSERT INTO OldBooking**

**2 (SELECT \***

**3 FROM Booking**

**4 WHERE dateTo < DATE'2016-09-01')**

**5 ;**

**2 rows created.**

**SQL> -- Q7c)**

**SQL> DELETE FROM Booking**

**2 WHERE dateTo < DATE'2016-09-01'**

**3 ;**

**2 rows deleted.**

**SQL> -- CHECK**

**SQL> SELECT \* FROM Hotel;**

**HOTELNO HOTELNAME CITY**

**-------- -------------------- ------------------------------**

**J001 Marriot Vancouver**

**J002 Inn Chilliwack**

**J003 Fraser Victoria**

**SQL> SELECT \* FROM Room;**

**ROOMNO HOTELNO TYPE PRICE DISCOUNT**

**-------- -------- ------ ---------- ----------**

**9 J001 Single 19 0**

**40 J002 Double 34.5 0**

**80 J003 Family 50 0**

**SQL> SELECT \* FROM Guest;**

**GUESTNO GUESTNAME**

**-------- ------------------------------**

**GUESTADDRESS**

**---------------------------------------------------------------------------**

**00111100 Dan**

**402 Fraser st**

**00111102 Ben**

**521 Rupert st**

**00111104 Hannah**

**109 Willindon Ave.**

**SQL> SELECT \* FROM Booking;**

**HOTELNO GUESTNO DATEFROM DATETO ROOMNO**

**-------- -------- ---------- ---------- --------**

**J003 00111104 2016-10-10 2016-10-15 40**

**SQL> SELECT \* FROM OldBooking;**

**HOTELNO GUESTNO DATEFROM DATETO ROOMNO**

**-------- -------- ---------- ---------- --------**

**J001 00111100 2016-03-29 2016-04-05 9**

**J002 00111102 2016-05-05 2016-05-10 80**

**SQL>**

**SQL> -- \*\* This last SQL\*Plus SPOOL command line is MOST IMPORTANT !! \*\***

**SQL> -- It is usually the last line of the script file, and its purpose**

**SQL> -- is to close off the current spool output text file, so that all**

**SQL> -- output from this script will be flushed to disk and saved to the**

**SQL> -- SPOOL text file.**

**SQL> -- Otherwise, you may end up missing output content.**

**SQL> -- END**

**SQL> SPOOL OFF**