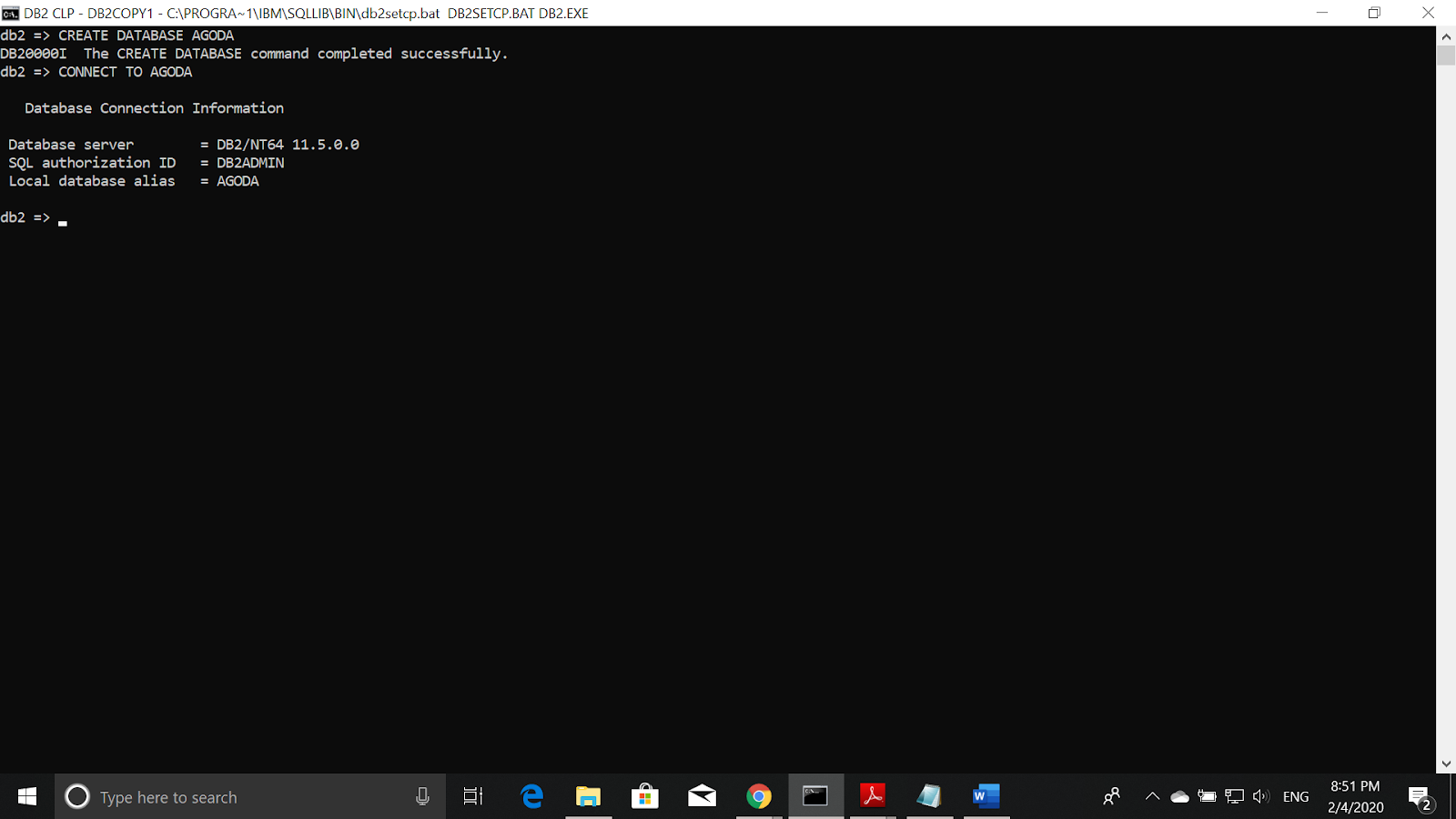
# **DATA DEFINITION COMMAND (DDL)**

## CREATE AND CONNECT TO NEW DATABASE

CREATE DATABASE AGODA

CONNECT TO AGODA



Explanation : The syntaxes above created and connected to the new database.

## CREATE TABLES

CREATE TABLE HOTEL(

HOTEL\_ID VARCHAR(5) PRIMARY KEY NOT NULL,

HOTEL\_NAME VARCHAR(50),

HOTEL\_PHONE VARCHAR(15),

HOTEL\_RATING DECIMAL(3,1),

HOTEL\_BRANCH VARCHAR(20),

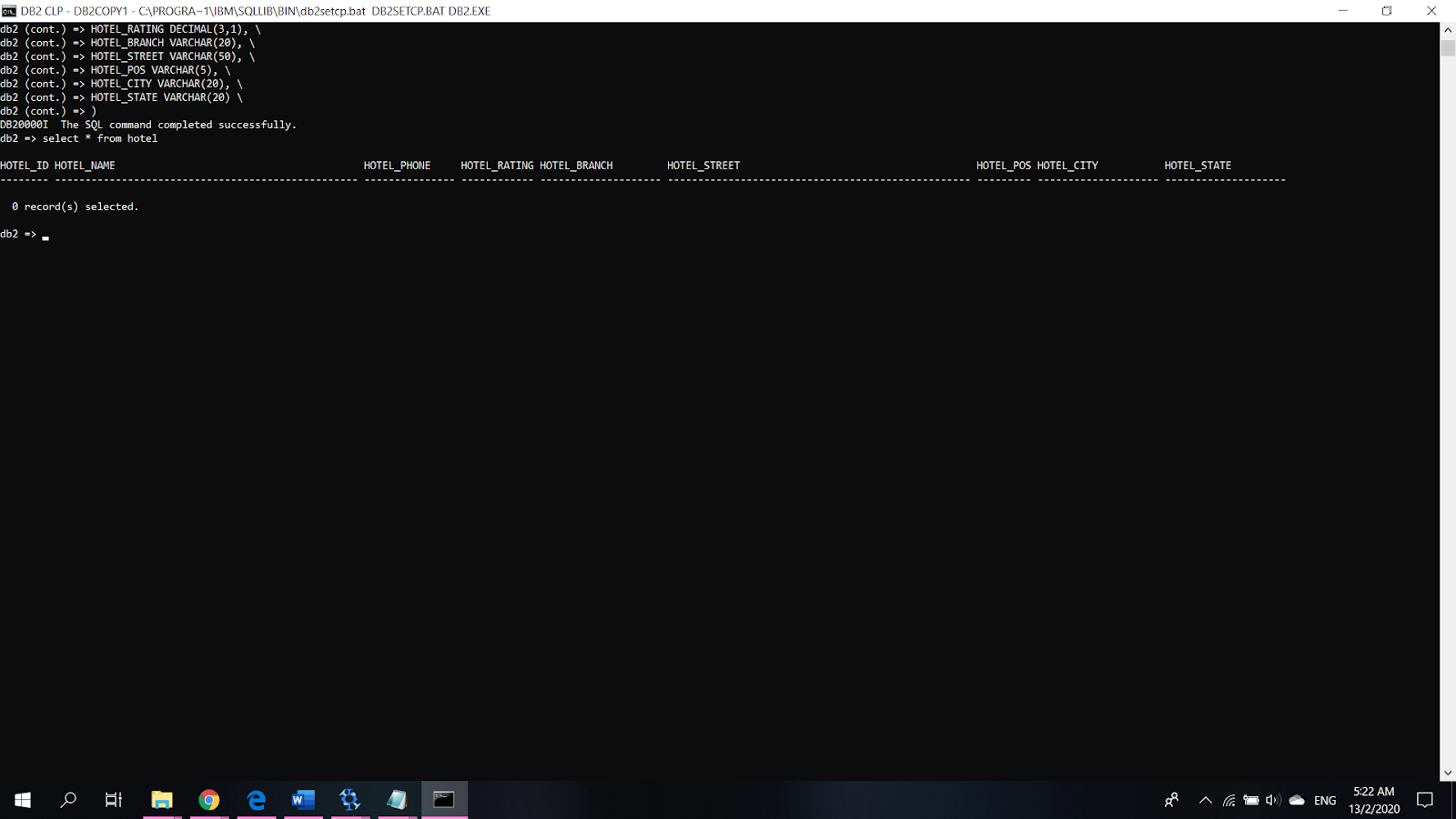
HOTEL\_STREET VARCHAR(50),

HOTEL\_POS VARCHAR(5),

HOTEL\_CITY VARCHAR(20),

HOTEL\_STATE VARCHAR(20)

)



Explanation : The syntaxes above created the HOTEL table.

CREATE TABLE ROOM(

CREATE TABLE ROOM(

ROOM\_ID VARCHAR(5) PRIMARY KEY NOT NULL,

ROOM\_TYPE VARCHAR(50),

ROOM\_STATUS VARCHAR(20) DEFAULT 'Available',

ROOM\_PRICE DECIMAL(10,2),

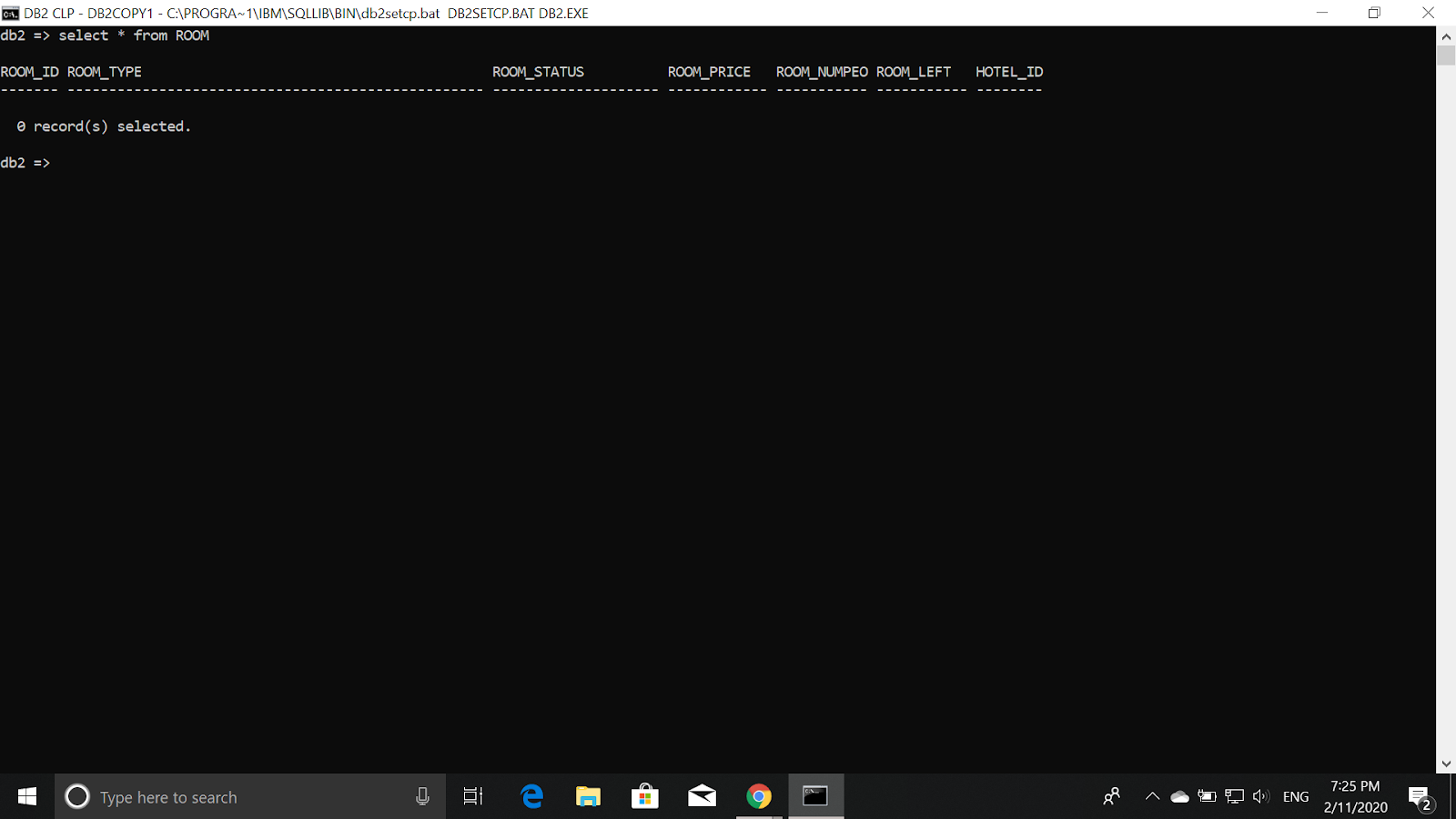
ROOM\_NUMPEO INTEGER,

ROOM\_LEFT INTEGER,

HOTEL\_ID VARCHAR(5),

FOREIGN KEY(HOTEL\_ID) REFERENCES HOTEL

)



Explanation : The syntaxes above created the ROOM table.

CREATE TABLE GUEST(

GUEST\_ID VARCHAR(5) PRIMARY KEY NOT NULL,

GUEST\_NRIC VARCHAR(30) NOT NULL UNIQUE ,

GUEST\_NAME VARCHAR(50) ,

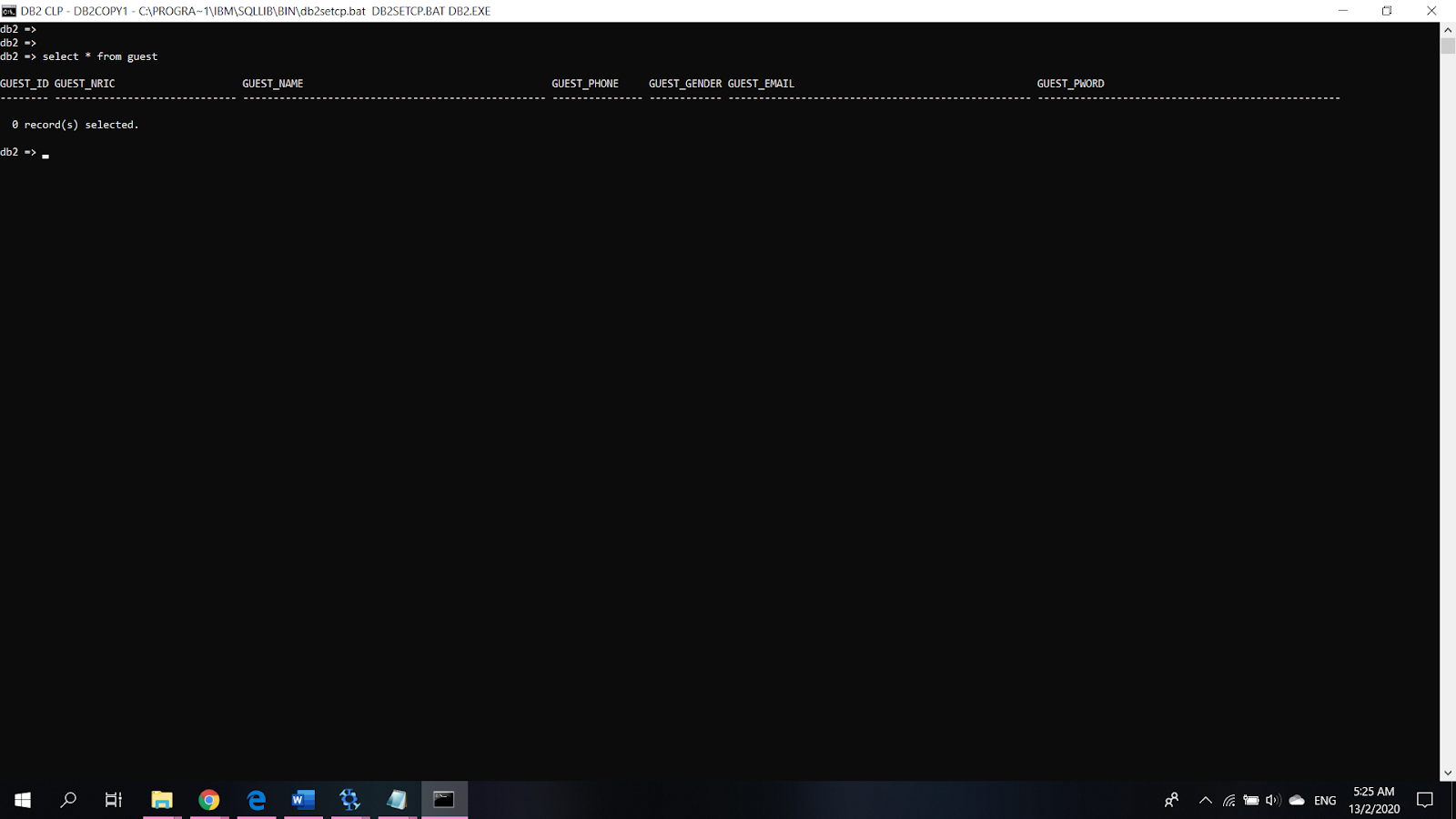
GUEST\_PHONE VARCHAR(15) ,

GUEST\_GENDER CHAR(1) CHECK (GUEST\_GENDER IN ('F','M')),

GUEST\_EMAIL VARCHAR(50) NOT NULL UNIQUE,

GUEST\_PWORD VARCHAR(50)

)



Explanation : The syntaxes above created the GUEST table and GUEST\_GENDER can be either F or M only.

CREATE TABLE PROMOTION(

PROMO\_ID VARCHAR(5) PRIMARY KEY NOT NULL,

PROMO\_TYPE VARCHAR(30) ,

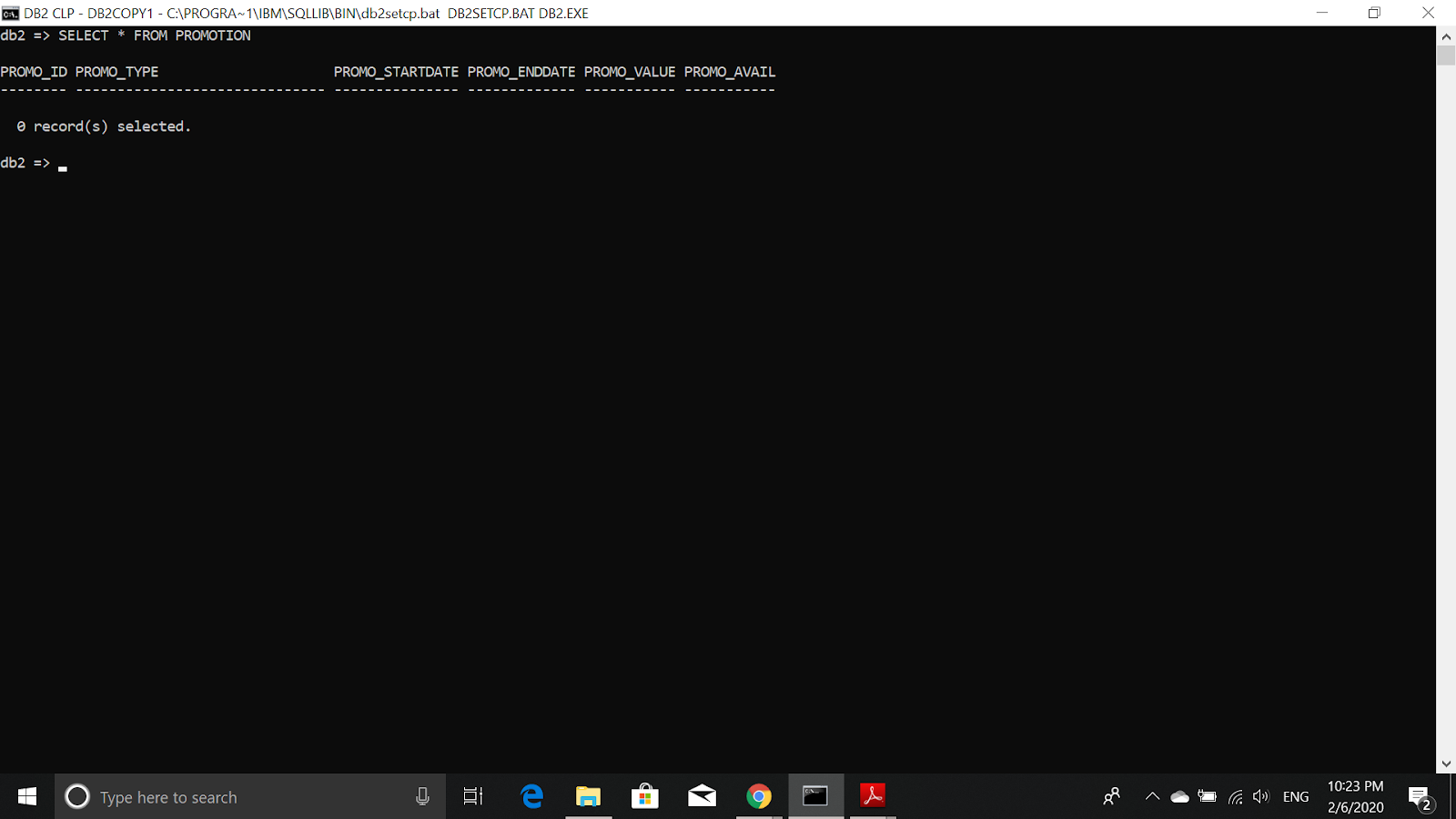
PROMO\_STARTDATE DATE ,

PROMO\_ENDDATE DATE ,

PROMO\_VALUE DECIMAL(3,2) ,

PROMO\_AVAIL INTEGER

)



Explanation : The syntaxes above created the PROMOTION table.

CREATE TABLE BOOKINGSUMMARY(

BKGSUM\_ID INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY(START WITH 100 INCREMENT BY 1),

TOTAL\_PRICE DECIMAL(10,2) ,

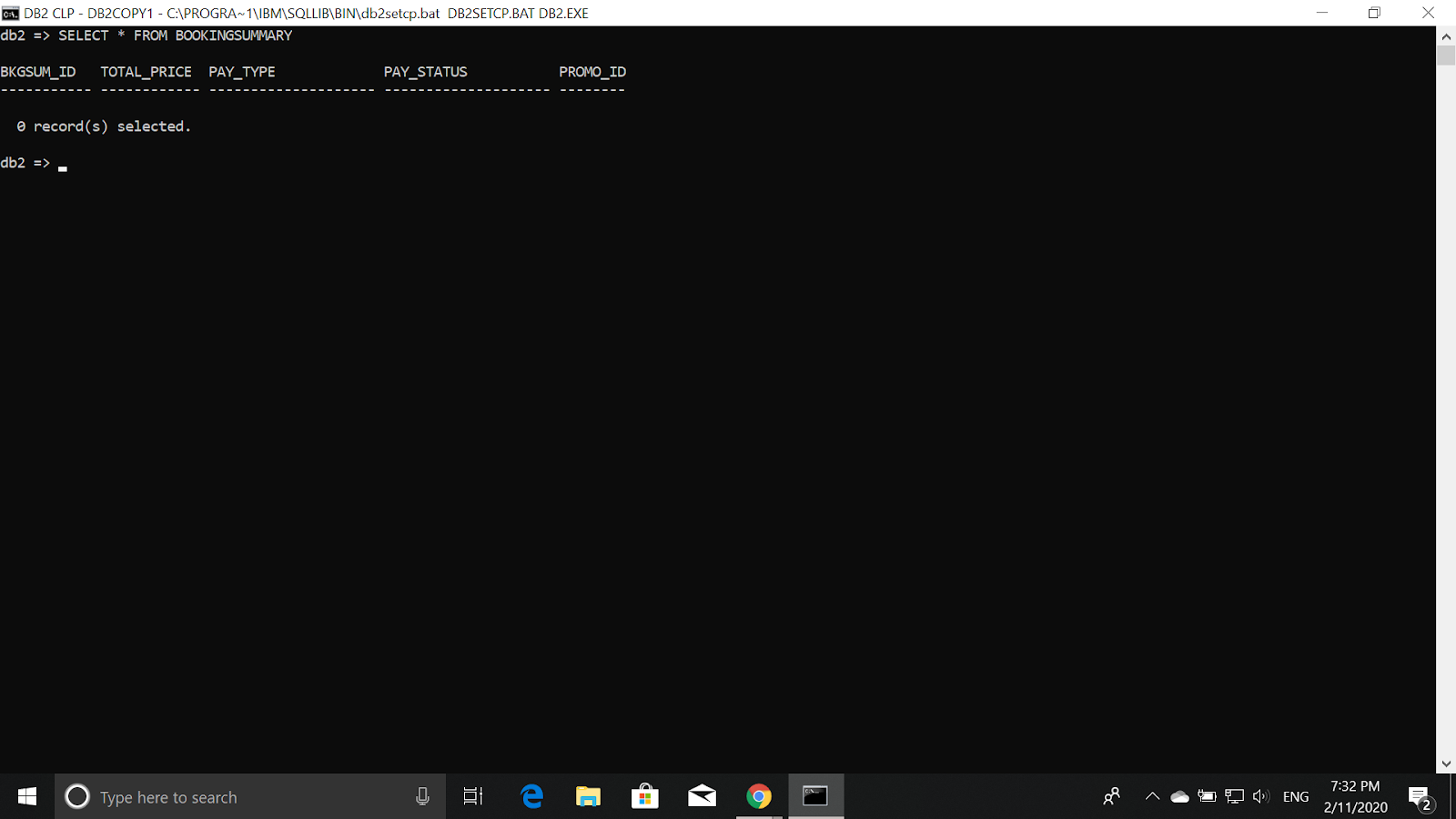
PAY\_TYPE VARCHAR(20) ,

PAY\_STATUS VARCHAR(20) ,

PROMO\_ID VARCHAR(5),

FOREIGN KEY(PROMO\_ID) REFERENCES PROMOTION

)



Explanation : The syntaxes above created the BOOKINGSUMMARY table and the BKSUM\_ID will automatically increment by 1 for every new row inserted to BOOKINGSUMMARY.

CREATE TABLE BOOKINGDETAILS(

BKGDET\_ID VARCHAR(5) PRIMARY KEY NOT NULL,

CHECKIN\_D DATE NOT NULL,

CHECKOUT\_D DATE NOT NULL,

TOTAL\_ROOM\_PRICE DECIMAL(10,2),

GUEST\_ID VARCHAR(5),

ROOM\_ID VARCHAR(5),

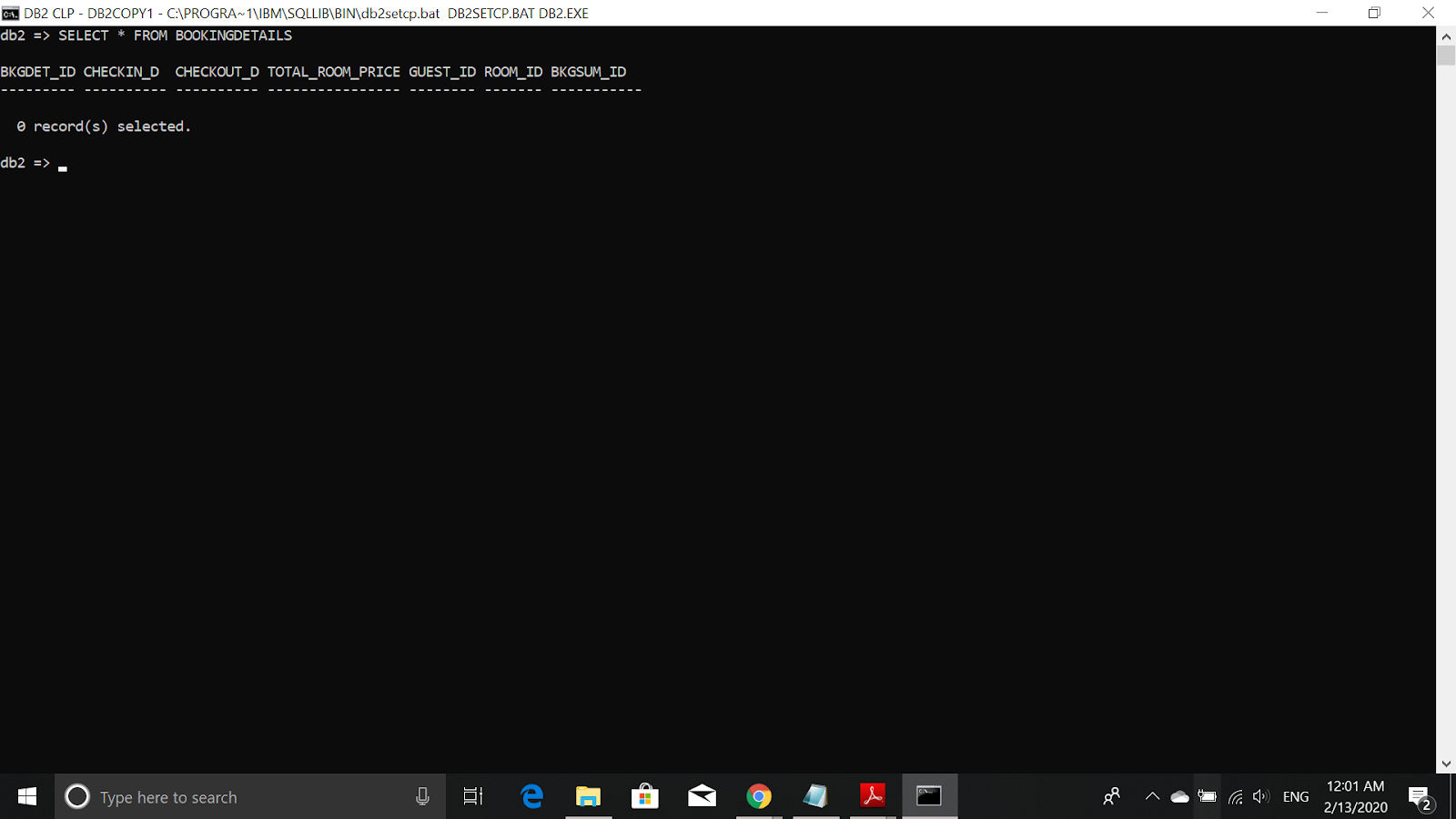
BKGSUM\_ID INTEGER,

FOREIGN KEY(GUEST\_ID) REFERENCES GUEST,

FOREIGN KEY(ROOM\_ID) REFERENCES ROOM,

FOREIGN KEY (BKGSUM\_ID) REFERENCES BOOKINGSUMMARY

)



Explanation : The syntaxes above created the BOOKINGDETAILS table.

# 

# **DATA INSERTION**

**HOTEL**

INSERT INTO HOTEL(HOTEL\_ID, HOTEL\_NAME, HOTEL\_PHONE, HOTEL\_RATING, HOTEL\_STREET, HOTEL\_POS, HOTEL\_CITY, HOTEL\_STATE ) VALUES ('H0001', 'Citadines Dpulze Cyberjaya','60386899999' ,8.5 , 'Cyber 12', '63000', 'Cyberjaya', 'Selangor')

INSERT INTO HOTEL VALUES('H0002', 'Grand Cyberjaya Hotel ','60122365118', 7.7, 'Grand Hotel ', 'Cyber 12', '63000', 'Cyberjaya', 'Selangor')

INSERT INTO HOTEL VALUES('H0003', 'Grand Kapar Hotel', '60333623020', 6, 'Grand Hotel', '30 Jalan Klang Sentral', '42200', 'Klang', 'Selangor')

INSERT INTO HOTEL VALUES('H0004', 'OYO Hotel Shah Alam ','60444555661', 8.2 ,'OYO Hotel ', '19 Jalan Keluli', '40000', 'Klang', 'Selangor')

INSERT INTO HOTEL VALUES('H0005', 'Melia Kuala Lumpur','60327852828', 7.8, 'Melia Hotel', '16 Jalan Imbi', '55100', 'Kuala Lumpur', 'Kuala Lumpur')

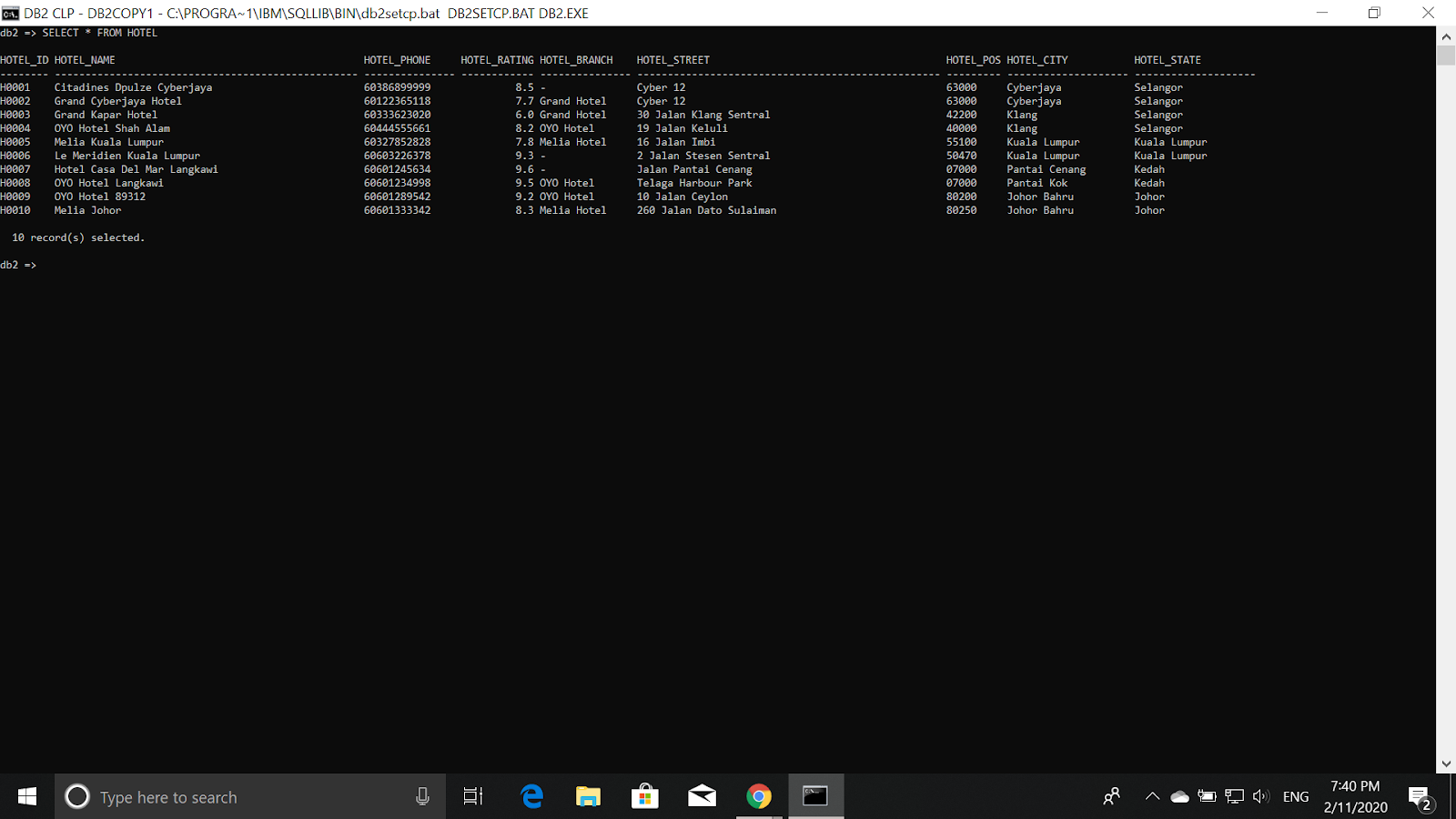
INSERT INTO HOTEL(HOTEL\_ID, HOTEL\_NAME, HOTEL\_PHONE, HOTEL\_RATING, HOTEL\_STREET, HOTEL\_POS, HOTEL\_CITY, HOTEL\_STATE ) VALUES('H0006', 'Le Meridien Kuala Lumpur', '60603226378', 9.3, '2 Jalan Stesen Sentral', '50470', 'Kuala Lumpur', 'Kuala Lumpur')

INSERT INTO HOTEL(HOTEL\_ID, HOTEL\_NAME, HOTEL\_PHONE, HOTEL\_RATING, HOTEL\_STREET, HOTEL\_POS, HOTEL\_CITY, HOTEL\_STATE ) VALUES('H0007', 'Hotel Casa Del Mar Langkawi', '60601245634', 9.6,'Jalan Pantai Cenang', '07000', 'Pantai Cenang', 'Kedah')

INSERT INTO HOTEL VALUES('H0008', 'OYO Hotel Langkawi', '60601234998', 9.5, 'OYO Hotel ', 'Telaga Harbour Park', '07000', 'Pantai Kok', 'Kedah')

INSERT INTO HOTEL VALUES('H0009', 'OYO Hotel 89312', '60601289542', 9.2, 'OYO Hotel', '10 Jalan Ceylon', '80200', 'Johor Bahru', 'Johor')

INSERT INTO HOTEL VALUES('H0010', 'Melia Johor', '60601333342', 8.3, 'Melia Hotel', '260 Jalan Dato Sulaiman', '80250', 'Johor Bahru', 'Johor')



**ROOM**

INSERT INTO ROOM VALUES('R0001','One-Bedroom Suite', DEFAULT, 299.00, 2 , 20, 'H0001')

INSERT INTO ROOM VALUES('R0002', 'Two-Bedroom Suite', DEFAULT, 399.00, 4, 25, 'H0001')

INSERT INTO ROOM VALUES('R0003', 'Deluxe Double Room', 'Not Available', 120.00, 2, 0, 'H0001')

INSERT INTO ROOM VALUES('R0004', 'Standard Single Room', DEFAULT, 49.00, 1, 14, 'H0002')

INSERT INTO ROOM VALUES('R0005', 'Deluxe Double Room', DEFAULT,69.00, 2, 30, 'H0002')

INSERT INTO ROOM VALUES('R0006', 'Deluxe Single Room', 'Not Available', 69.00, 1, 0, 'H0002')

INSERT INTO ROOM VALUES('R0007', 'Deluxe Double Room', DEFAULT, 69.00, 2, 23, 'H0003')

INSERT INTO ROOM VALUES('R0008', 'Standard Single Room', DEFAULT,50.00, 1, 15, 'H0003')

INSERT INTO ROOM VALUES('R0009 ', 'Deluxe Single Room', DEFAULT, 69.00, 1, 10, 'H0003')

INSERT INTO ROOM VALUES('R0010', 'Standard Family Room', DEFAULT, 250.00, 6 ,23, 'H0004')

INSERT INTO ROOM VALUES('R0011', 'Standard Double Room', DEFAULT, 98.00, 2 ,42, 'H0004')

INSERT INTO ROOM VALUES('R0012' ,'Deluxe Double Room', DEFAULT, 128.00, 2, 37, 'H0004')

INSERT INTO ROOM VALUES('R0013', 'Deluxe Single Room', DEFAULT, 180.00, 1 , 21, 'H0005')

INSERT INTO ROOM VALUES('R0014', 'Deluxe Double Room', DEFAULT, 180.00, 2, 13, 'H0005')

INSERT INTO ROOM VALUES('R0015', 'Standard Double Room', 'Not Available', 140.00, 2 , 0, 'H0005')

INSERT INTO ROOM VALUES('R0016', 'Deluxe Family Room', DEFAULT, 650.00, 6, 19, 'H0006')

INSERT INTO ROOM VALUES('R0017', 'Standard Family Room', 'Not Available', 550.00, 6, 0, 'H0006')

INSERT INTO ROOM VALUES('R0018', 'Deluxe Family Room', DEFAULT, 830.00, 4, 20, 'H0006')

INSERT INTO ROOM VALUES('R0019', 'Standard Family Room', DEFAULT, 450.00, 6, 47, 'H0007')

INSERT INTO ROOM VALUES('R0020', 'Deluxe Family Room', DEFAULT,540.00, 6, 34, 'H0007')

INSERT INTO ROOM VALUES('R0021', 'Two-Bedroom Suite', DEFAULT, 710.00, 4, 7, 'H0007')

INSERT INTO ROOM VALUES('R0022', 'Deluxe Double Room', DEFAULT, 380.00, 2, 18, 'H0008')

INSERT INTO ROOM VALUES('R0023', 'Deluxe Single Room', DEFAULT, 380.00, 1, 16, 'H0008')

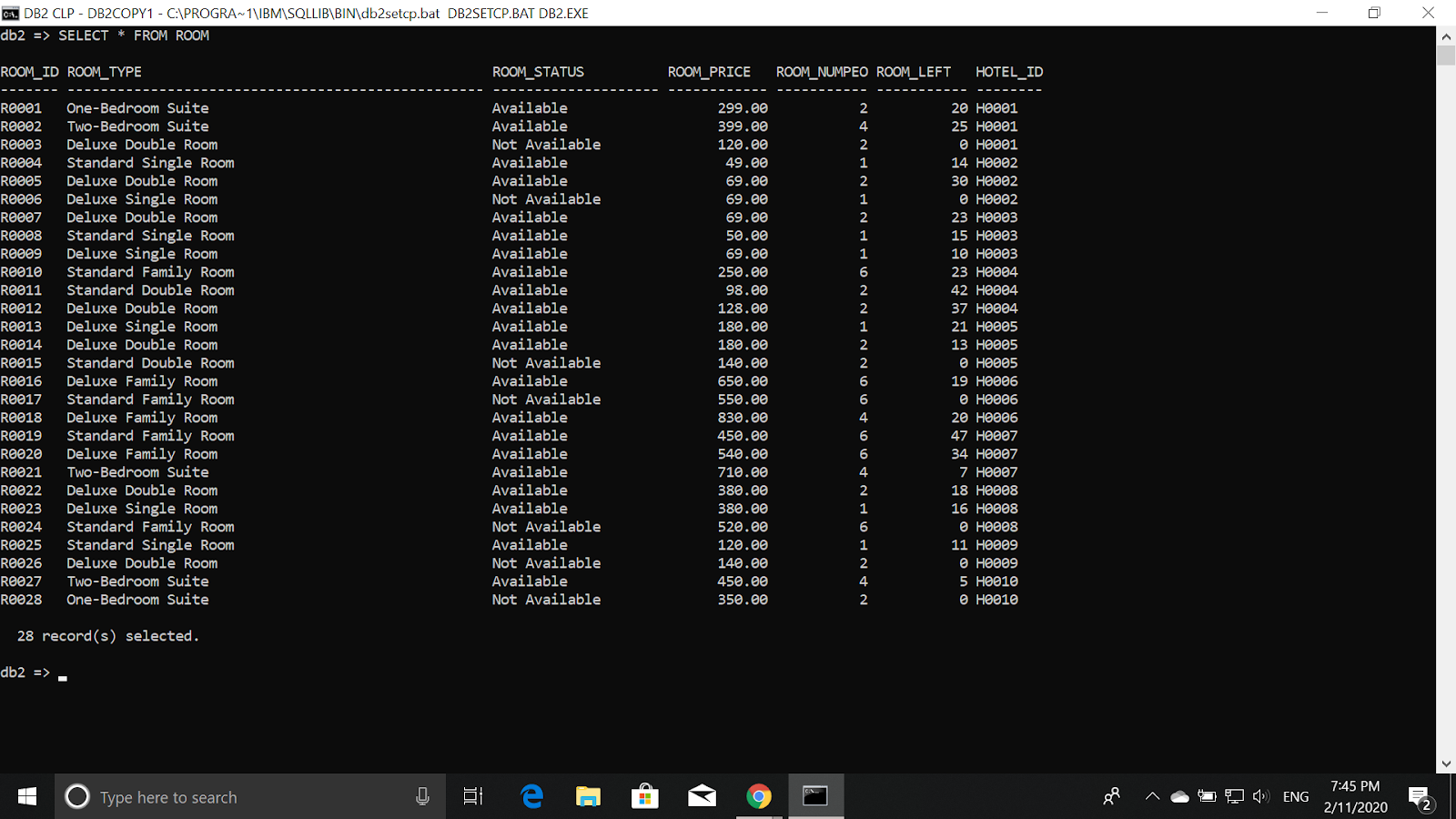
INSERT INTO ROOM VALUES('R0024', 'Standard Family Room', 'Not Available', 520.00, 6, 0, 'H0008')

INSERT INTO ROOM VALUES('R0025', 'Standard Single Room', DEFAULT, 120.00, 1, 11, 'H0009')

INSERT INTO ROOM VALUES('R0026', 'Deluxe Double Room', 'Not Available',140.00, 2, 0, 'H0009')

INSERT INTO ROOM VALUES('R0027', 'Two-Bedroom Suite', DEFAULT, 450.00, 4, 5, 'H0010')

INSERT INTO ROOM VALUES('R0028', 'One-Bedroom Suite', 'Not Available',350.00, 2, 0, 'H0010')



**GUEST**

INSERT INTO GUEST VALUES('G0001', '000311-01-1981', 'SOH XUAN WEI', '01234574636', 'M', 'xuanwei122@gmail.com', 'xuanwei520')

INSERT INTO GUEST VALUES('G0002', '000412-02-1343', 'LIM KAI XUAN', '01345735783', 'M', '[kaixuan8899@gmail.com](mailto:kaixuan8899@gmail.com)','kaikai1122')

INSERT INTO GUEST VALUES('G0003', '000520-07-1499', 'NG JUN WEI', '01758835345', 'M', 'weiweihaha12@gmail.com', 'weiwei0909')

INSERT INTO GUEST VALUES('G0004', '000510-01-1388', 'TEO YEE XUAN', '01223474634', 'F', 'yeexuan0920@gmail.com', 'yeexuan123')

INSERT INTO GUEST VALUES('G0005', '000520-01-1399', 'SOH YU ZHEN', '01234523487', 'M', 'sohsoh1212@gmail.com', 'sohsoh45678')

INSERT INTO GUEST VALUES('G0006', '000430-01-1677', 'SOH JIN ZHE', '01256435332', 'M', 'sohzhezhe@gmail.com', 'zhezhe1677')

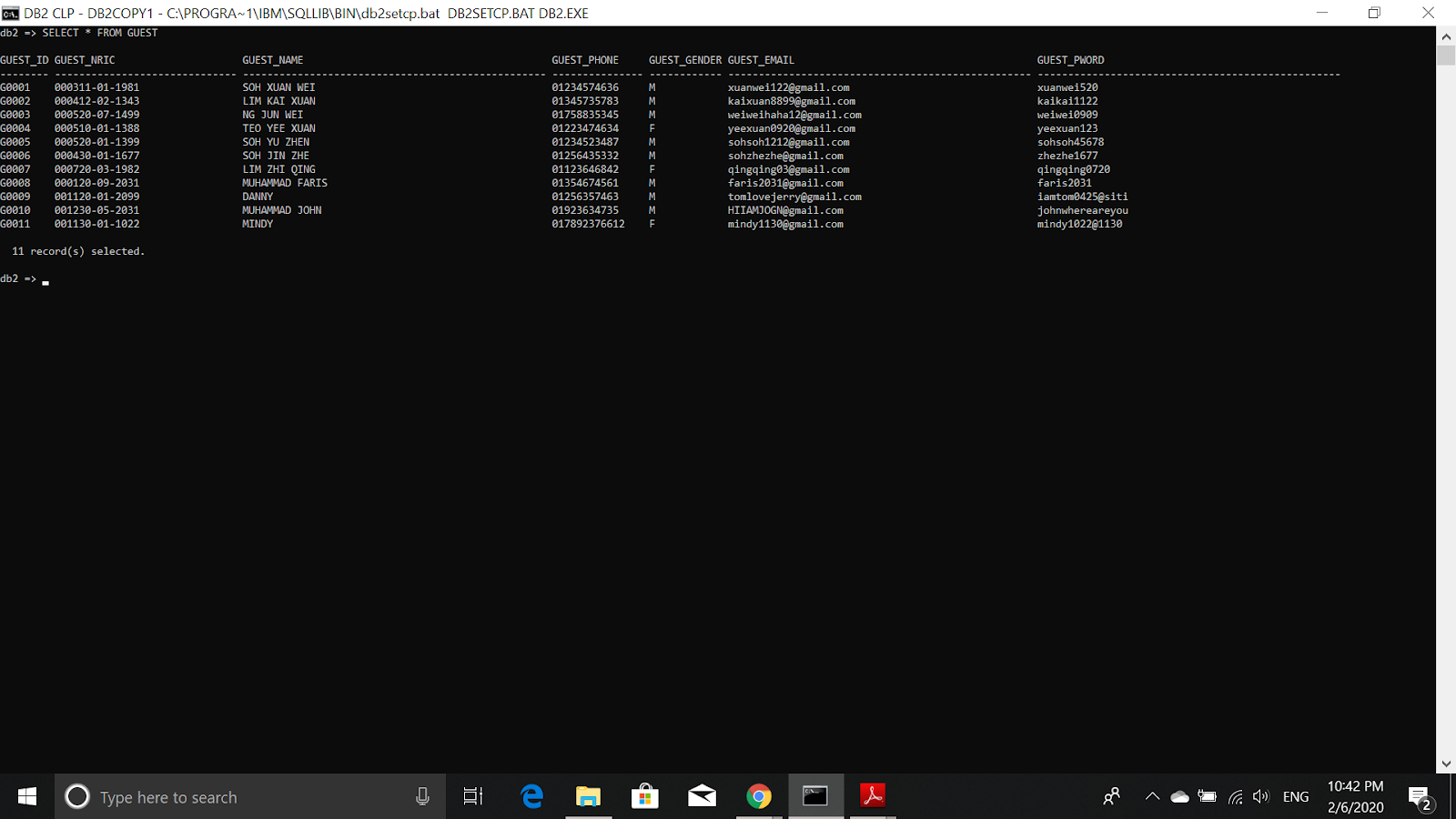
INSERT INTO GUEST VALUES('G0007', '000720-03-1982', 'LIM ZHI QING', '01123646842', 'F', 'qingqing03@gmail.com', 'qingqing0720')

INSERT INTO GUEST VALUES('G0008', '000120-09-2031', 'MUHAMMAD FARIS', '01354674561', 'M', 'faris2031@gmail.com', 'faris2031')

INSERT INTO GUEST VALUES('G0009', '001120-01-2099', 'DANNY' ,'01256357463', 'M', 'tomlovejerry@gmail.com', 'iamtom0425@siti')

INSERT INTO GUEST VALUES('G0010', '001230-05-2031', 'MUHAMMAD JOHN', '01923634735', 'M', 'HIIAMJOGN@gmail.com', 'johnwhereareyou')

INSERT INTO GUEST VALUES('G0011', '001130-01-1022', 'MINDY', '017892376612', 'F', 'mindy1130@gmail.com', 'mindy1022@1130')



**PROMOTION**

INSERT INTO PROMOTION VALUES('D0001', 'HARI MERDEKA DISCOUNT', '2020-08-31','2020-09-01', 0.7, 10)

INSERT INTO PROMOTION VALUES('D0002', 'CHINESE NEW YEAR DISCOUNT', '2020-01-25', '2020-01-26', 0.7,10)

INSERT INTO PROMOTION VALUES('D0003', 'HARI RAYA DISCOUNT', '2020-05-23','2020-05-24', 0.65, 9)

INSERT INTO PROMOTION VALUES('D0004', 'NEW YEAR DISCOUNT', '2019-12-31','2020-01-01', 0.75,10)

INSERT INTO PROMOTION VALUES('D0005', 'VALENTINE DAY DISCOUNT', '2020-02-14', '2020-02-15', 0.75,10)

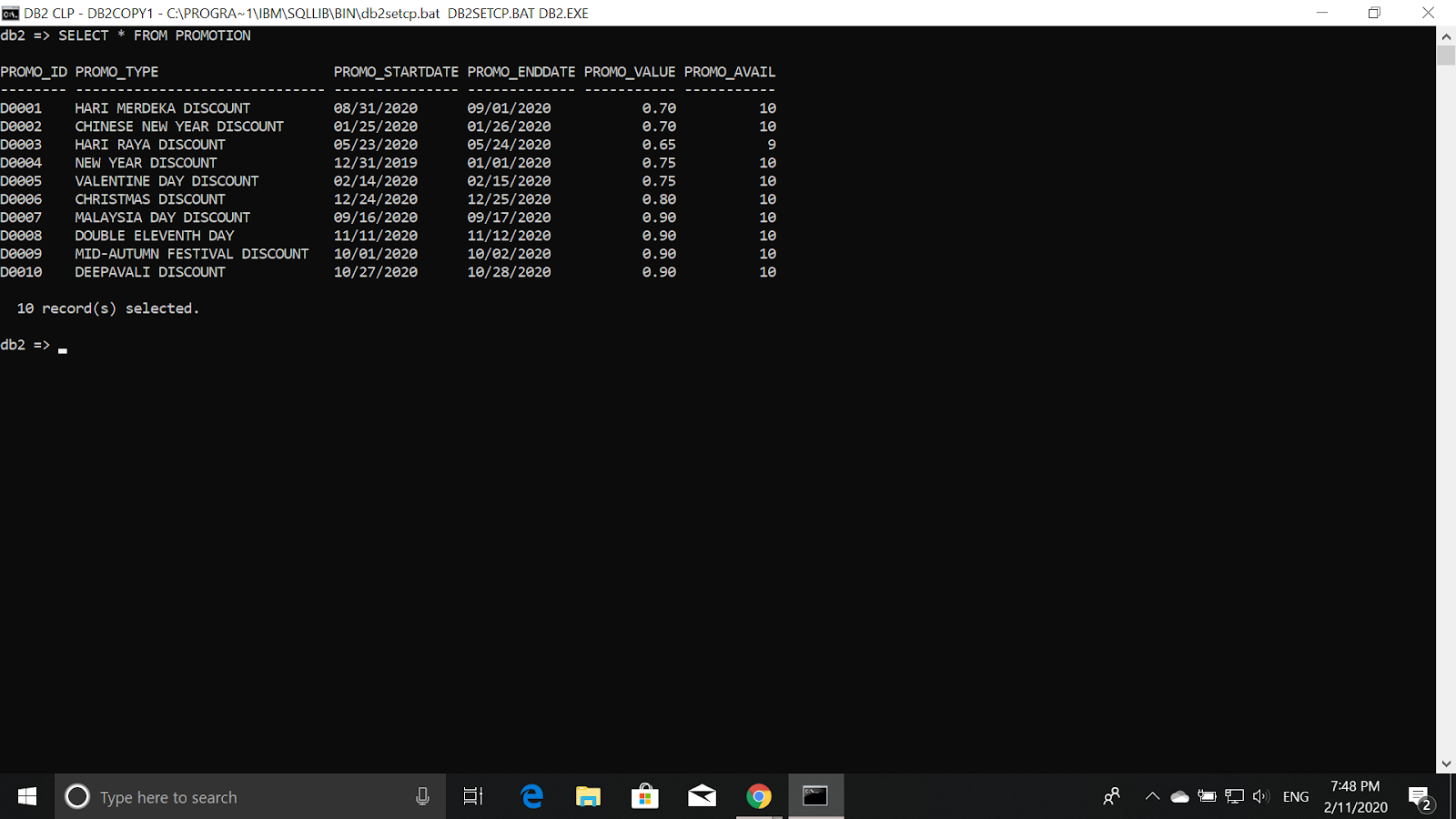
INSERT INTO PROMOTION VALUES('D0006', 'CHRISTMAS DISCOUNT', '2020-12-24', '2020-12-25', 0.8, 10)

INSERT INTO PROMOTION VALUES('D0007', 'MALAYSIA DAY DISCOUNT', '2020-09-16', '2020-09-17', 0.9,10)

INSERT INTO PROMOTION VALUES('D0008', 'DOUBLE ELEVENTH DAY', '2020-11-11', '2020-11-12',0.9,10)

INSERT INTO PROMOTION VALUES('D0009', 'MID-AUTUMN FESTIVAL DISCOUNT', '2020-10-01', '2020-10-02', 0.9,10)

INSERT INTO PROMOTION VALUES('D0010', 'DEEPAVALI DISCOUNT', '2020-10-27', '2020-10-28',0.9, 10)



**BOOKINGSUMMARY**

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS)VALUES(default, 828.00, 'CASH', 'PAID')

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 452.00, 'CASH', 'PENDING')

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 2370.00, 'ONLINE TRANSFER', 'PAID')

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 1100.00, 'ONLINE TRANSFER', 'PAID')

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 1660.00, 'CASH', 'PENDING')

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 1620.00, 'ONLINE TRANSFER', 'PAID')

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 240.00, 'CASH', 'PENDING')

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 49.00, 'CREDIT CARD', 'PAID')

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 900.00, 'ONLINE TRANSFER','PAID')

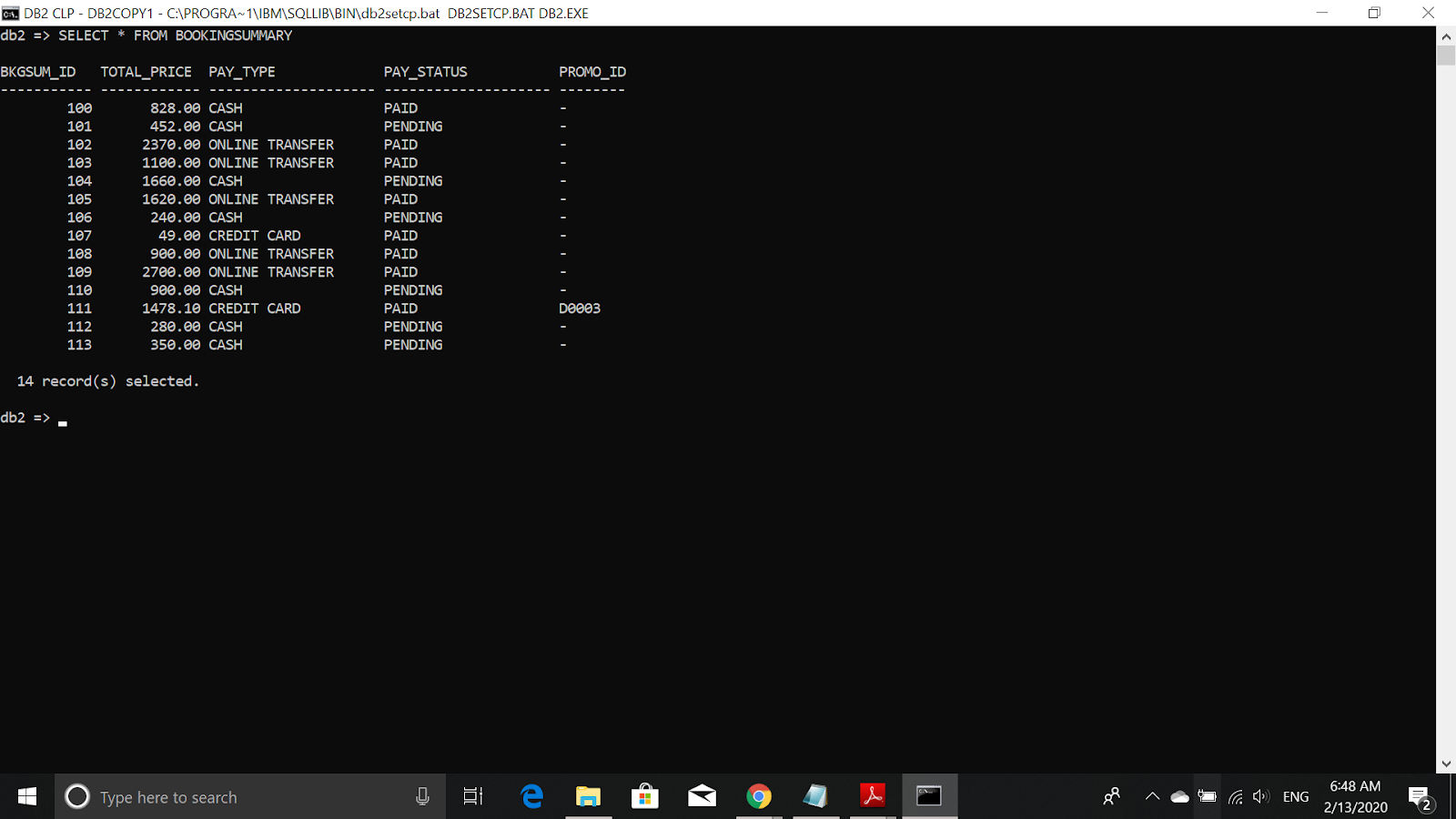
INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 2700.00, 'ONLINE TRANSFER', 'PAID')

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 900.00, 'CASH', 'PENDING')

INSERT INTO BOOKINGSUMMARY VALUES(default, 1478.10, 'CREDIT CARD', 'PAID', 'D0003')

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 280.00, 'CASH', 'PENDING')

INSERT INTO BOOKINGSUMMARY(BKGSUM\_ID, TOTAL\_PRICE, PAY\_TYPE, PAY\_STATUS) VALUES(default, 350.00,'CASH', 'PENDING')



**BOOKINGDETAILS**

INSERT INTO BOOKINGDETAILS VALUES('B0001', '2020-01-27', '2020-02-02', 414.00, 'G0001', 'R0006',100)

INSERT INTO BOOKINGDETAILS VALUES('B0002', '2020-01-27', '2020-02-02', 414.00, 'G0001', 'R0007',100)

INSERT INTO BOOKINGDETAILS VALUES('B0003', '2020-02-27', '2020-02-29', 196.00, 'G0002', 'R0011',101)

INSERT INTO BOOKINGDETAILS VALUES('B0004', '2020-02-27', '2020-02-29', 256.00, 'G0002', 'R0012',101)

INSERT INTO BOOKINGDETAILS VALUES('B0005', '2020-02-28', '2020-03-02', 420.00, 'G0003', 'R0015',102)

INSERT INTO BOOKINGDETAILS VALUES('B0006', '2020-02-28', '2020-03-02', 1950.00, 'G0003', 'R0016',102)

INSERT INTO BOOKINGDETAILS VALUES('B0007', '2020-03-24', '2020-03-26', 1100.00, 'G0004', 'R0017',103)

INSERT INTO BOOKINGDETAILS VALUES('B0008', '2020-03-28', '2020-03-30', 1660.00, 'G0004', 'R0018',104)

INSERT INTO BOOKINGDETAILS VALUES('B0009', '2020-04-05', '2020-04-08', 1620.00, 'G0005', 'R0020',105)

INSERT INTO BOOKINGDETAILS VALUES('B0010', '2020-04-11', '2020-04-13', 240.00, 'G0006', 'R0003',106)

INSERT INTO BOOKINGDETAILS VALUES('B0011', '2020-04-27', '2020-04-28', 49.00, 'G0007', 'R0004',107)

INSERT INTO BOOKINGDETAILS VALUES('B0012', '2020-05-01', '2020-05-03', 900.00, 'G0008', 'R0027',108)

INSERT INTO BOOKINGDETAILS VALUES('B0013', '2020-05-11', '2020-05-14', 1140.00, 'G0009', 'R0023',109)

INSERT INTO BOOKINGDETAILS VALUES('B0014', '2020-05-11', '2020-05-14', 1560.00, 'G0009', 'R0024',109)

INSERT INTO BOOKINGDETAILS VALUES('B0015', '2020-05-20', '2020-05-22', 900.00, 'G0009', 'R0027',110)

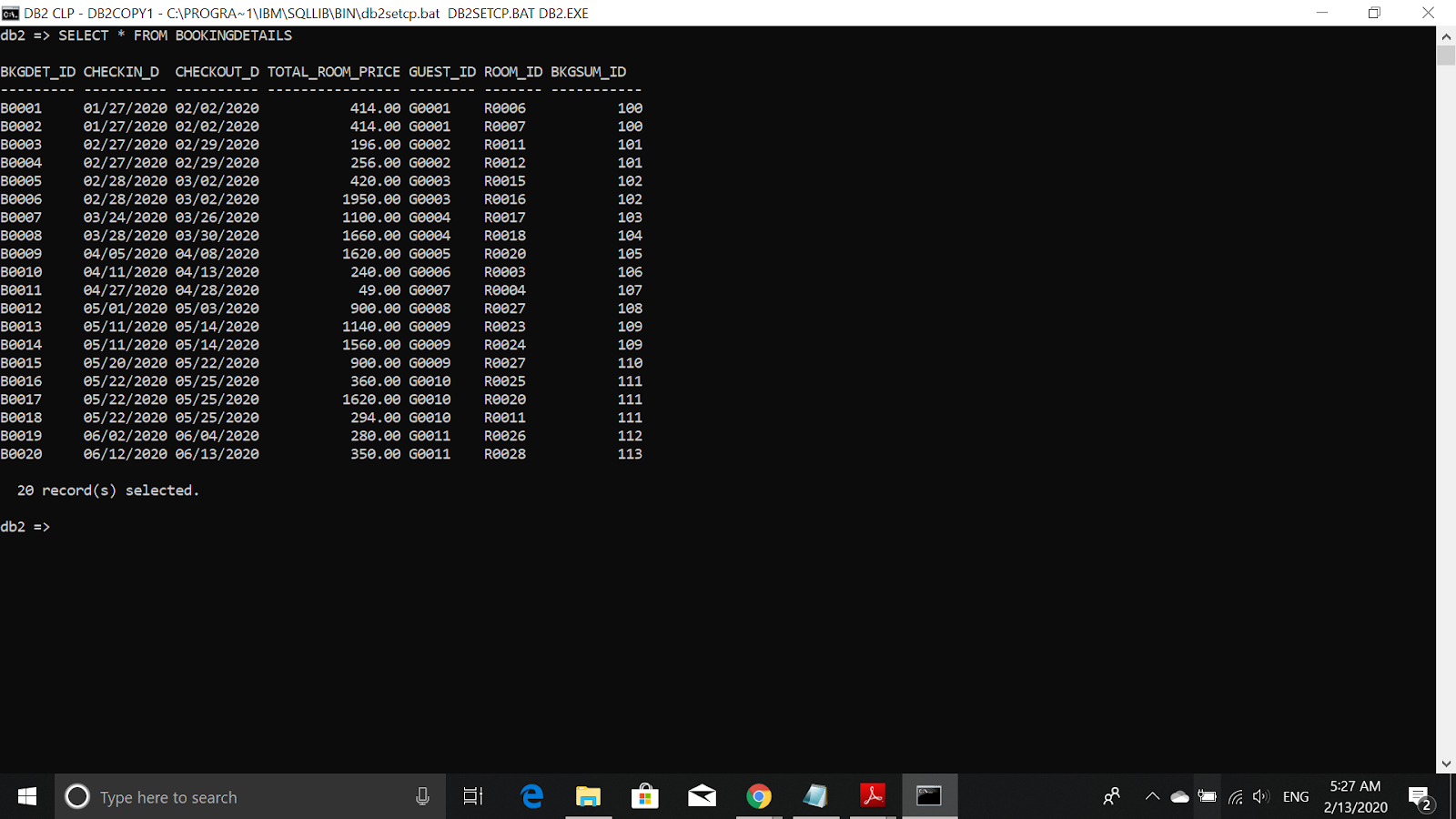
INSERT INTO BOOKINGDETAILS VALUES('B0016', '2020-05-22', '2020-05-25', 360.00, 'G0010', 'R0025',111)

INSERT INTO BOOKINGDETAILS VALUES('B0017', '2020-05-22', '2020-05-25', 1620.00, 'G0010', 'R0020',111)

INSERT INTO BOOKINGDETAILS VALUES('B0018', '2020-05-22', '2020-05-25', 294.00, 'G0010', 'R0011',111)

INSERT INTO BOOKINGDETAILS VALUES('B0019', '2020-06-02', '2020-06-04', 280.00, 'G0011', 'R0026',112)

INSERT INTO BOOKINGDETAILS VALUES('B0020', '2020-06-12', '2020-06-13', 350.00, 'G0011', 'R0028',113)



# **DATA MANIPULATION LANGUAGE (DML)**

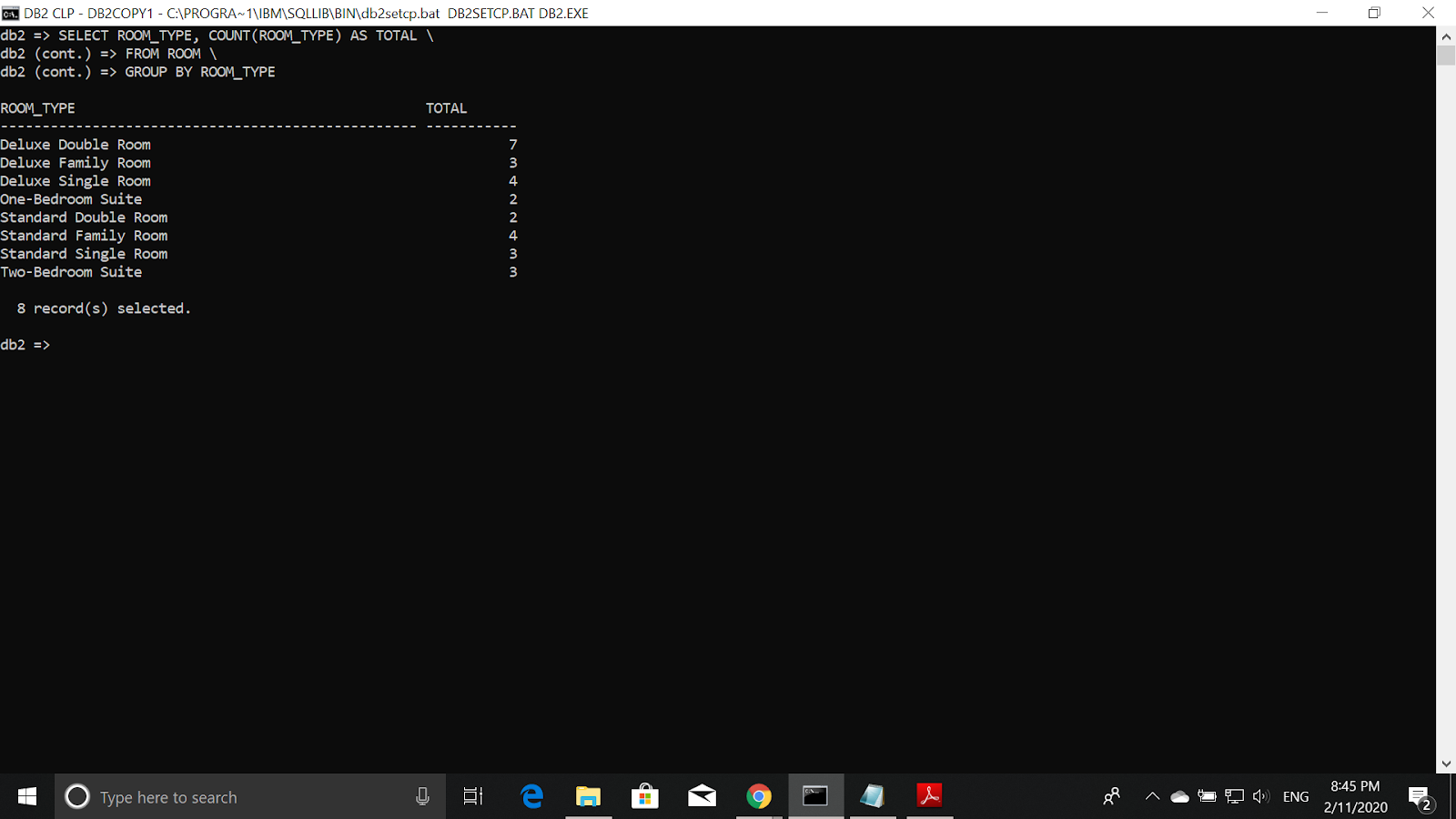
## **AGGREGATE FUNCTION**

The total room type provided in the system

SELECT ROOM\_TYPE, COUNT(ROOM\_TYPE) AS TOTAL

FROM ROOM

GROUP BY ROOM\_TYPE



The total Deluxe Double Room left for the hotels in Selangor

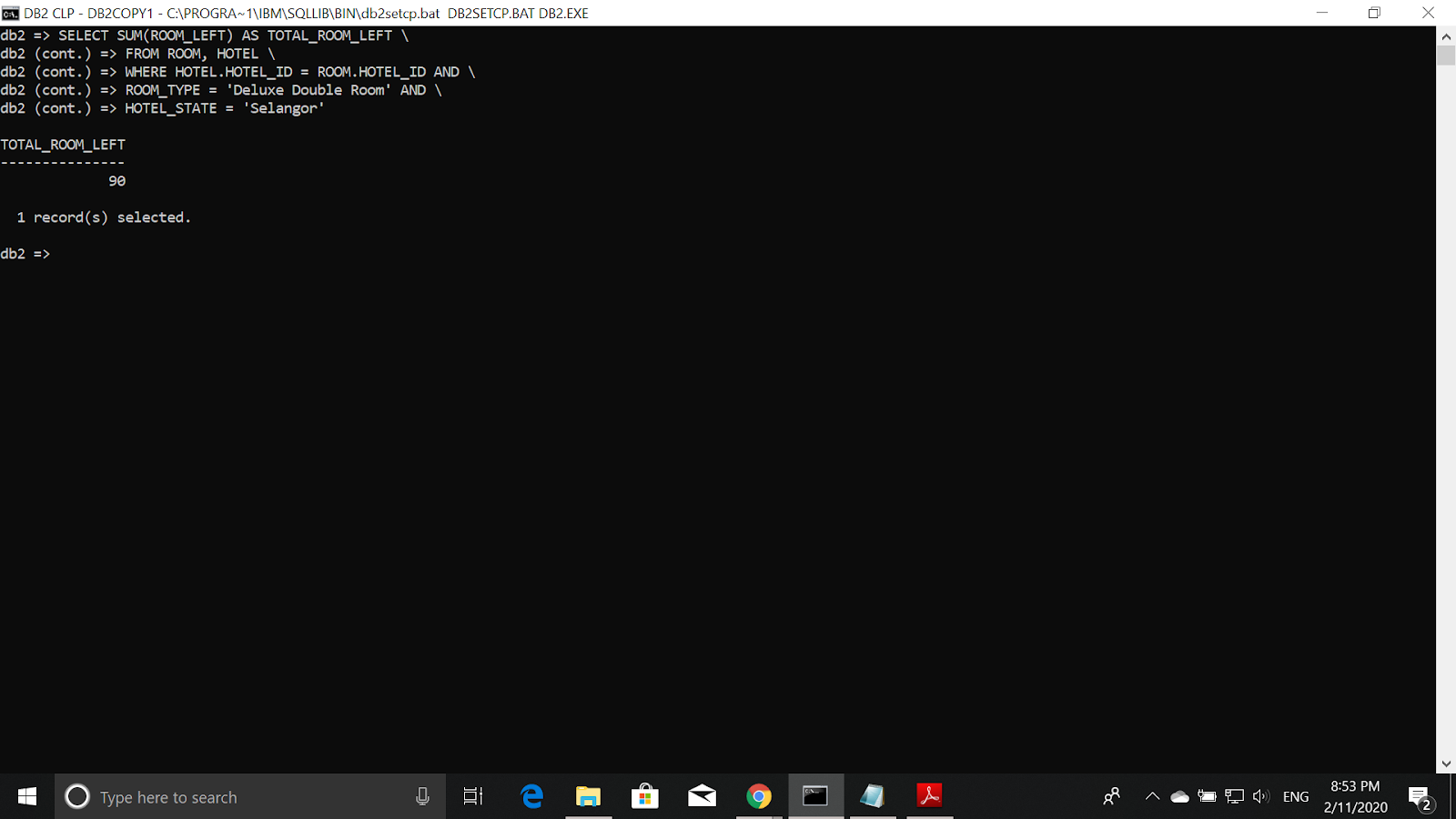
SELECT SUM(ROOM\_LEFT) AS TOTAL\_ROOM\_LEFT

FROM ROOM, HOTEL

WHERE HOTEL.HOTEL\_ID = ROOM.HOTEL\_ID AND

ROOM\_TYPE = 'Deluxe Double Room' AND

HOTEL\_STATE = 'Selangor'



The Average Price of each hotel

SELECT HOTEL\_NAME, DECIMAL(AVG(ROOM\_PRICE),6,2) AS AVERAGE\_PRICE

FROM HOTEL, ROOM

WHERE HOTEL.HOTEL\_ID = ROOM.HOTEL\_ID

GROUP BY HOTEL\_NAME



## **QUERY WITH A GROUP BY AND HAVING CLAUSES**

The guest name with the number of bookings made for more than or equal 2

SELECT GUEST\_NAME, COUNT(BOOKINGDETAILS.GUEST\_ID) AS TOTAL\_BOOKINGS

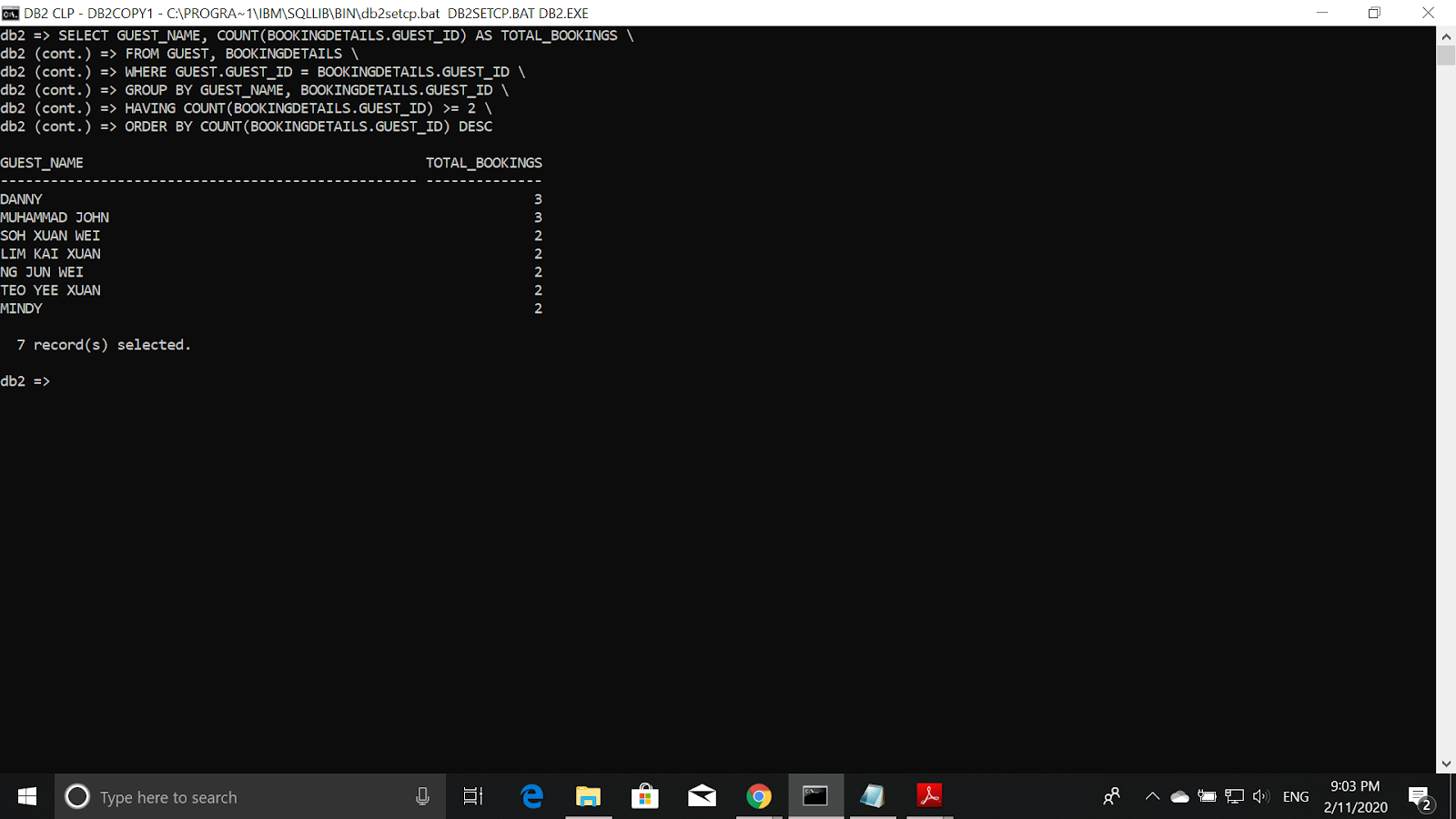
FROM GUEST, BOOKINGDETAILS

WHERE GUEST.GUEST\_ID = BOOKINGDETAILS.GUEST\_ID

GROUP BY GUEST\_NAME, BOOKINGDETAILS.GUEST\_ID

HAVING COUNT(BOOKINGDETAILS.GUEST\_ID) >= 2

ORDER BY COUNT(BOOKINGDETAILS.GUEST\_ID) DESC



The hotel name with more than 2 types of room available

SELECT HOTEL\_NAME, COUNT(ROOM\_STATUS) AS TOTAL\_ROOM\_TYPE\_AVAILABLE

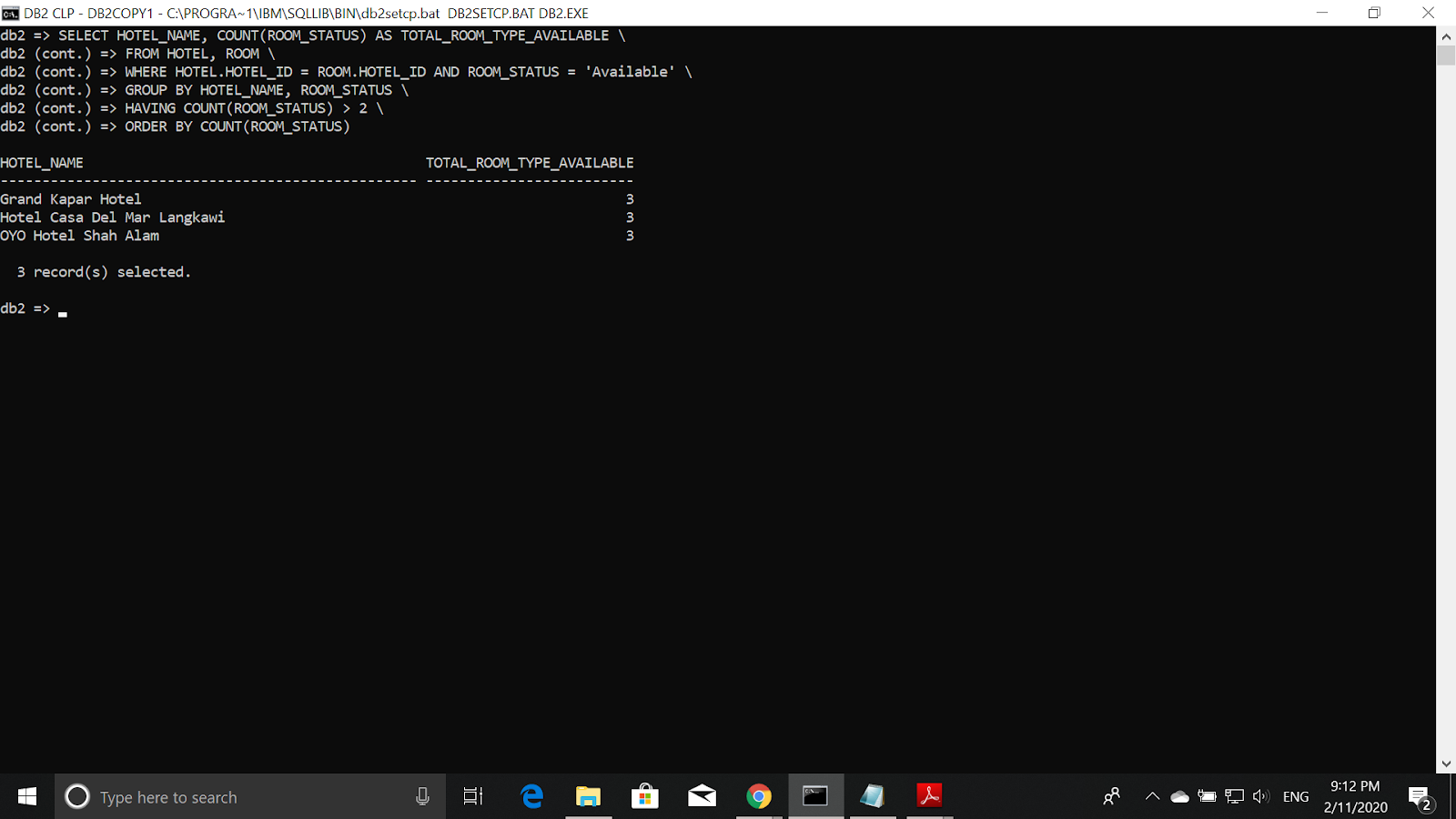
FROM HOTEL, ROOM

WHERE HOTEL.HOTEL\_ID = ROOM.HOTEL\_ID AND ROOM\_STATUS = 'Available'

GROUP BY HOTEL\_NAME, ROOM\_STATUS

HAVING COUNT(ROOM\_STATUS) > 2

ORDER BY COUNT(ROOM\_STATUS)



**Hotel name and the number of guests booked with the hotel rating more than 9**

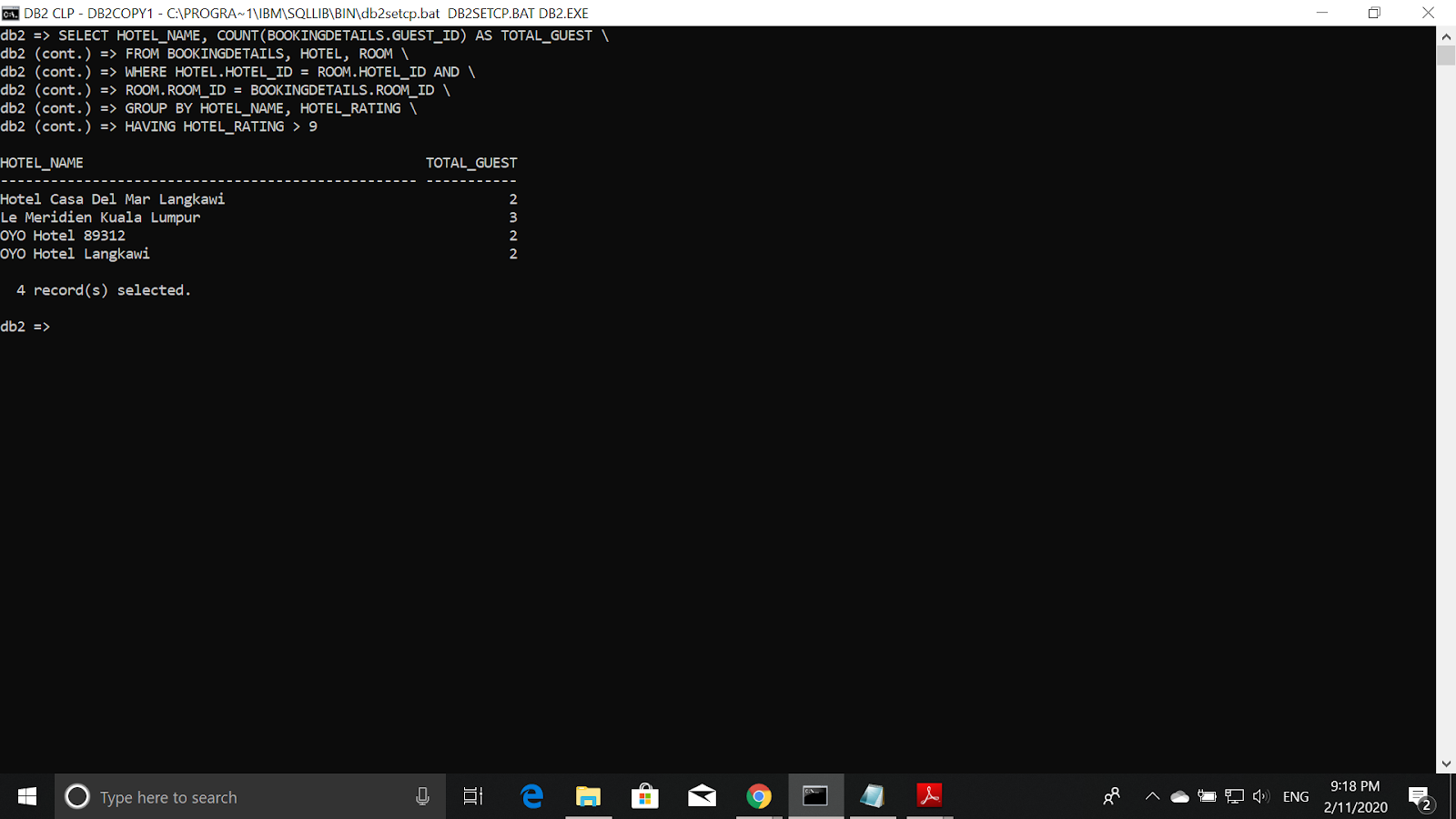
SELECT HOTEL\_NAME, COUNT(BOOKINGDETAILS.GUEST\_ID) AS TOTAL\_GUEST FROM BOOKINGDETAILS, HOTEL, ROOM

WHERE HOTEL.HOTEL\_ID = ROOM.HOTEL\_ID AND

ROOM.ROOM\_ID = BOOKINGDETAILS.ROOM\_ID

GROUP BY HOTEL\_NAME, HOTEL\_RATING

HAVING HOTEL\_RATING > 9



## 

## **TRIGGERS**

**Trigger that will give error message when inserting data to room**

CREATE TRIGGER NEWROOMTRG

NO CASCADE BEFORE INSERT ON ROOM

REFERENCING NEW AS N

FOR EACH ROW MODE DB2SQL

BEGIN

IF INSERTING AND ((N.ROOM\_TYPE = 'Deluxe Single Room' OR N.ROOM\_TYPE = 'Standard Single Room') AND ROOM\_NUMPEO != 1)

THEN

SIGNAL SQLSTATE '75001' SET MESSAGE\_TEXT = 'INVALID MAXIMUM PEOPLE IN THAT ROOM';

ELSEIF INSERTING AND ((N.ROOM\_TYPE = 'Deluxe Double Room' OR N.ROOM\_TYPE = 'Standard Double Room') AND ROOM\_NUMPEO != 2)

THEN

SIGNAL SQLSTATE '75001' SET MESSAGE\_TEXT = 'INVALID MAXIMUM PEOPLE IN THAT ROOM';

ELSEIF INSERTING AND ((N.ROOM\_TYPE = 'Deluxe Family Room' OR N.ROOM\_TYPE = 'Standard Family Room') AND ROOM\_NUMPEO != 6)

THEN

SIGNAL SQLSTATE '75001' SET MESSAGE\_TEXT = 'INVALID MAXIMUM PEOPLE IN THAT ROOM';

ELSEIF INSERTING AND ((N.ROOM\_TYPE = 'One-Bedroom Suite') AND ROOM\_NUMPEO != 2)

THEN

SIGNAL SQLSTATE '75001' SET MESSAGE\_TEXT = 'INVALID MAXIMUM PEOPLE IN THAT ROOM';

ELSEIF INSERTING AND ((N.ROOM\_TYPE = 'Two-Bedroom Suite') AND ROOM\_NUMPEO != 4)

THEN

SIGNAL SQLSTATE '75001' SET MESSAGE\_TEXT = 'INVALID MAXIMUM PEOPLE IN THAT ROOM';

ELSEIF (N.ROOM\_TYPE != 'Deluxe Single Room' AND N.ROOM\_TYPE != 'Standard Single Room' AND

N.ROOM\_TYPE != 'Deluxe Double Room' AND N.ROOM\_TYPE != 'Standard Double Room' AND

N.ROOM\_TYPE != 'Deluxe Family Room' AND N.ROOM\_TYPE != 'Standard Family Room' and

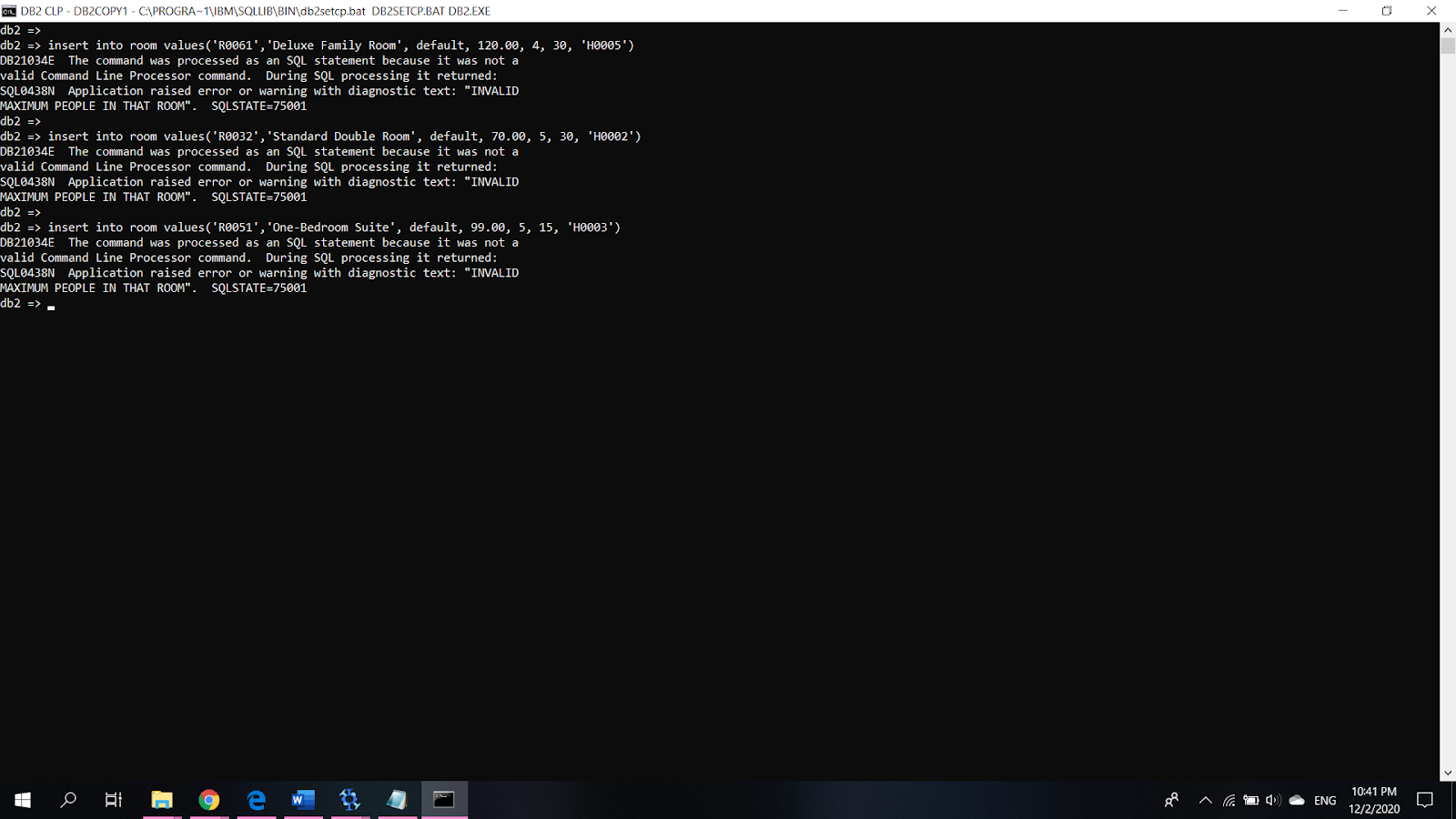
N.ROOM\_TYPE != 'One-Bedroom Suite' AND N.ROOM\_TYPE != 'Two-Bedroom Suite')

THEN

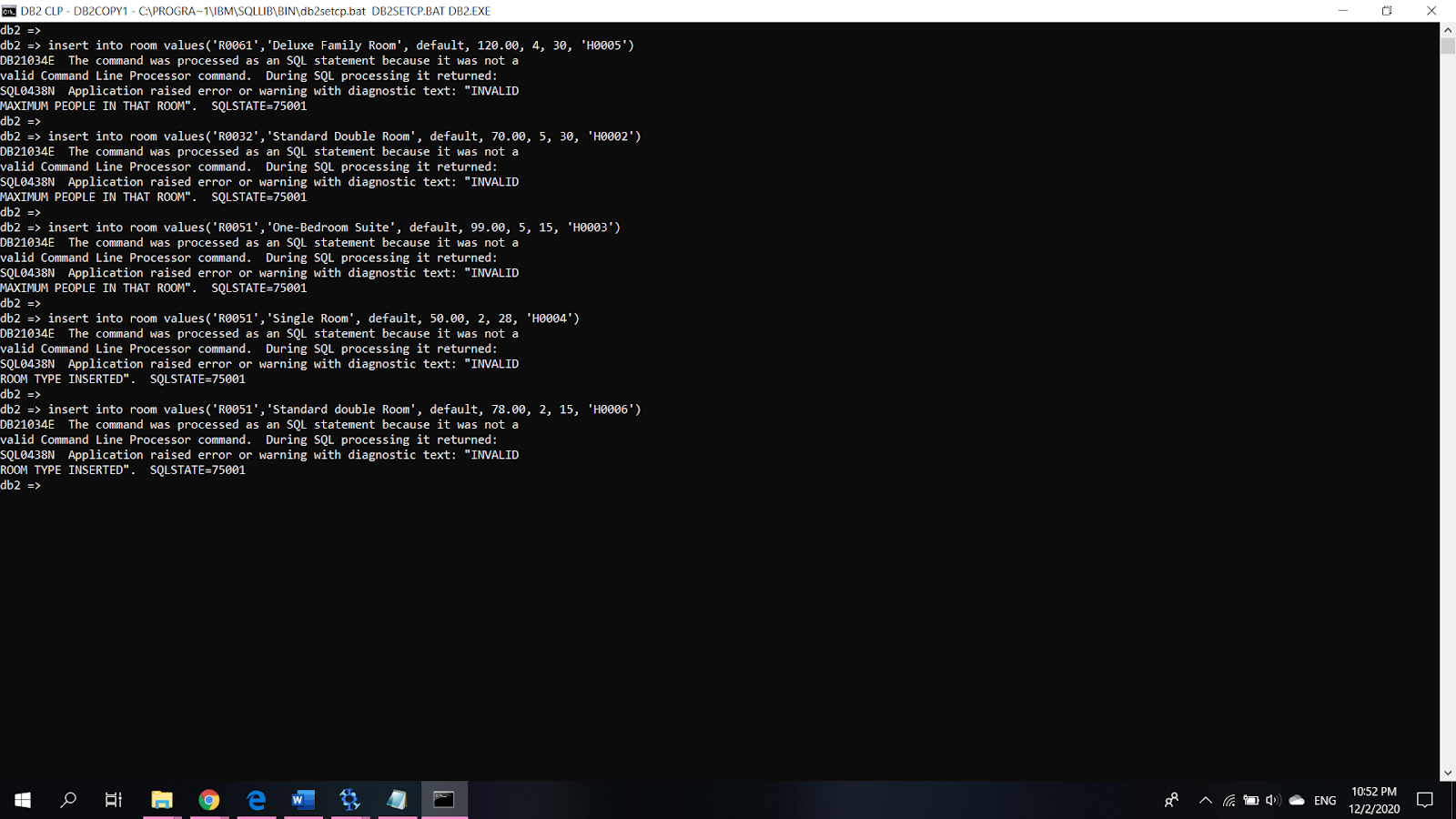
SIGNAL SQLSTATE '75001' SET MESSAGE\_TEXT = 'INVALID ROOM TYPE INSERTED';

END IF;

END



Explanation: This will give an error message when the number of people is invalid for that particular room after inserting data to room



Explanation: This will give an error message when the name of the room type is invalid.

**Trigger to check whether hotel phone is valid**

CREATE TRIGGER CHECK\_HOTEL\_PHO

BEFORE INSERT OR UPDATE OF HOTEL\_PHONE ON HOTEL

REFERENCING NEW AS N

FOR EACH ROW MODE DB2SQL

BEGIN

CASE

WHEN LENGTH(RTRIM(TRANSLATE(HOTEL\_PHONE, '\*', '1234567890'))) = 0

THEN

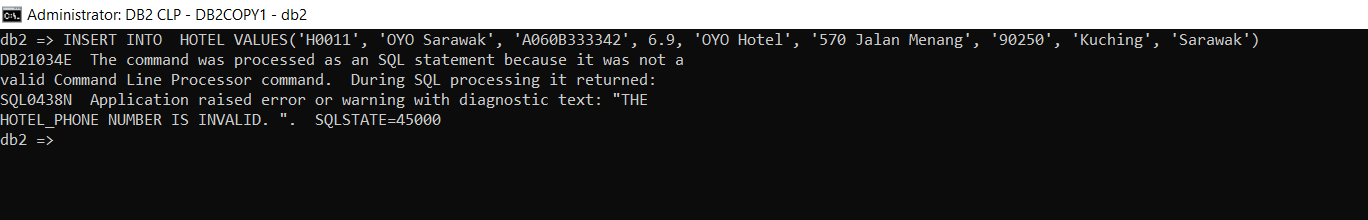
ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'THE HOTEL\_PHONE NUMBER IS INVALID. ' ;

END CASE;

END



Explanation: This trigger will check whether the hotel\_phone contains other than numbers. If hotel\_phone contains other than number, it will show an error message “Hotel\_phone number is valid.”

**Trigger to check email**

CREATE TRIGGER CHECK\_EMAIL

BEFORE INSERT OR UPDATE OF GUEST\_EMAIL ON GUEST

REFERENCING NEW AS N

FOR EACH ROW MODE DB2SQL

BEGIN

IF GUEST\_EMAIL IS NULL

THEN SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'THE GUEST’S EMAIL IS INVALID. ' ;

ELSEIF LOCATE(' ',GUEST\_EMAIL) <> 0 OR LOCATE('/',GUEST\_EMAIL) <> 0 OR LOCATE(' :',GUEST\_EMAIL) <> 0 OR LOCATE(' ;',GUEST\_EMAIL) <> 0

THEN SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = ' THE GUEST’S EMAIL IS INVALID. ' ;

ELSEIF LENGTH(GUEST\_EMAIL) -1 <= LOCATE(' .',GUEST\_EMAIL)

THEN SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'THE GUEST’S EMAIL IS INVALID. ' ;

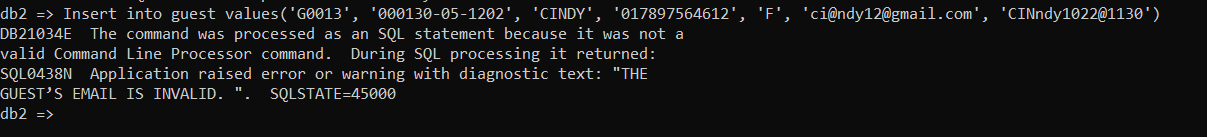
ELSEIF GUEST\_EMAIL LIKE '%@%@%' OR GUEST\_EMAIL NOT LIKE '%@%.%'

THEN SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'THE GUEST’S EMAIL IS INVALID. ' ;

END IF;

END



Explanation: This trigger checks whether the email is valid.

**Trigger to count the total price for each booking details**

CREATE TRIGGER trgTotalRoomPrice

AFTER INSERT ON BOOKINGDETAILS

REFERENCING NEW AS N

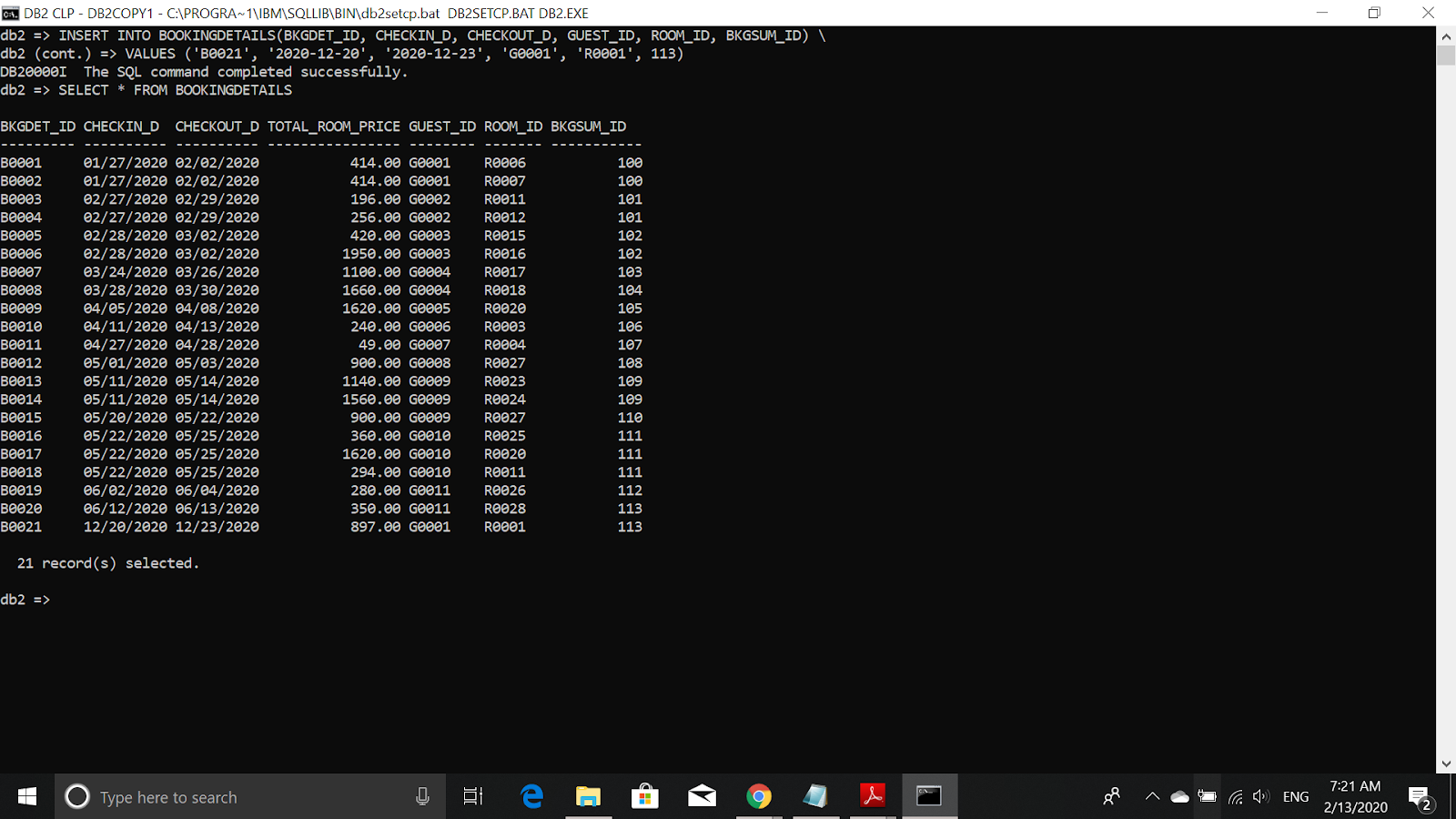
FOR EACH ROW DB2SQL

UPDATE BOOKINGDETAILS

SET TOTAL\_ROOM\_PRICE = (SELECT (DAYS(BOOKINGDETAILS.CHECKOUT\_D) - DAYS(BOOKINGDETAILS.CHECKIN\_D))\*ROOM PRICE

FROM ROOM, BOOKINGDETAILS WHERE ROOM.ROOM\_ID = BOOKINGDETAILS.ROOM\_ID AND BKGDET\_ID = N.BKGDET\_ID)

WHERE BKGDET\_ID = N.BKGDET\_ID



Explanation: The TOTAL\_ROOM\_PRICE will count the total days and the price for the room for one day automatically after inserting the data.

## 

## **STORED PROCEDURE**

Increase the number of room after customer had check out

CREATE PROCEDURE CHECK\_ROOM(IN CHECKIN\_D DATE, ROOM\_ID VARCHAR(5))

BEGIN

IF (CHECKIN\_D >(SELECT CHECKOUT\_D FROM BOOKINGDETAILS WHERE ROM\_ID = BOOKINGDETAILS.ROOM\_ID))

THEN UPDATE ROOM SET(ROOM\_LEFT, ROOM\_STATUS) = (ROOM\_LEFT+1 , 'Available') WHERE ROM\_ID = ROOM.ROOM\_ID;

ELSEIF CHECKIN\_D > (SELECT CHECKIN\_D FROM BOOKINGDETAILS WHERE ROM\_ID = BOOKINGDETAILS.ROOM\_ID) AND CHECKIN\_D < (SELECT CHECKOUT\_D FROM BOOKINGDETAILS WHERE ROM\_ID = BOOKINGDETAILS.ROOM\_ID)

THEN SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'THIS ROOM HAS BEEN BOOKED FOR THE DATE PROVIDED. ' ;

ELSEIF (CHECKIN\_D < (SELECT CHECKIN\_D FROM BOOKINGDETAILS WHERE ROM\_ID = BOOKINGDETAILS.ROOM\_ID))

THEN SIGNAL SQLSTATE '45000'

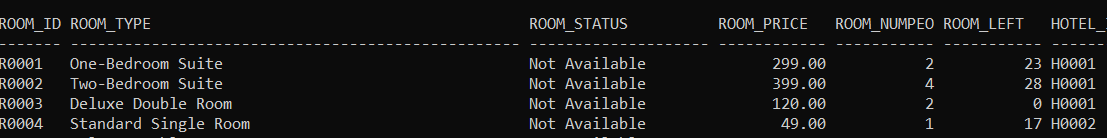
SET MESSAGE\_TEXT = 'CANNOT CHOOSE DATE THAT ALREADY PASS. ' ;

END IF;

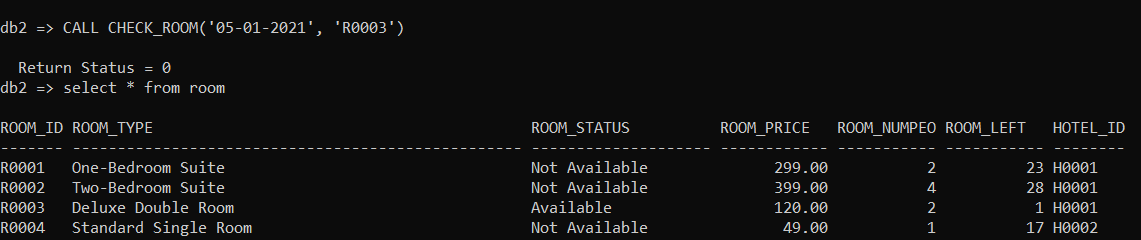
END

To Execute this procedure:

**CALL CHECK\_ROOM(‘05-01-2020’, ‘R0003’)**



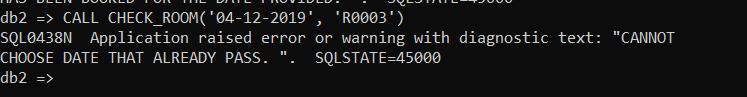
Explanation: Before this procedure is called, the room left of Room\_ID ‘R0003’ is 0 and the room status is Not Available.



Explanation: After this procedure is called, the room left of Room\_ID ‘R003’ become one and the room status changes to available.



Explanation: If a customer enters the date which has been booked by others, an error message will show to customer.



Explanation: If a customer enters a date that already passes, an error message will show to the customer..

**Guest Change Password**

CREATE PROCEDURE ChangePW(IN Email VARCHAR(50), OLD\_P VARCHAR(50), NEW\_P VARCHAR(50))

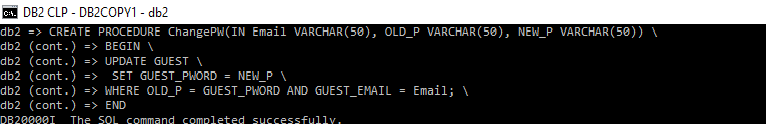
BEGIN

UPDATE GUEST

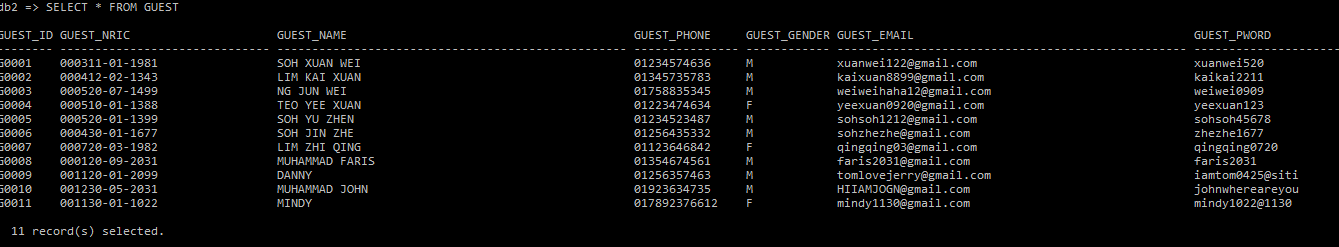
SET GUEST\_PWORD = NEW\_P

WHERE OLD\_P = GUEST\_PWORD AND GUEST\_EMAIL = Email;

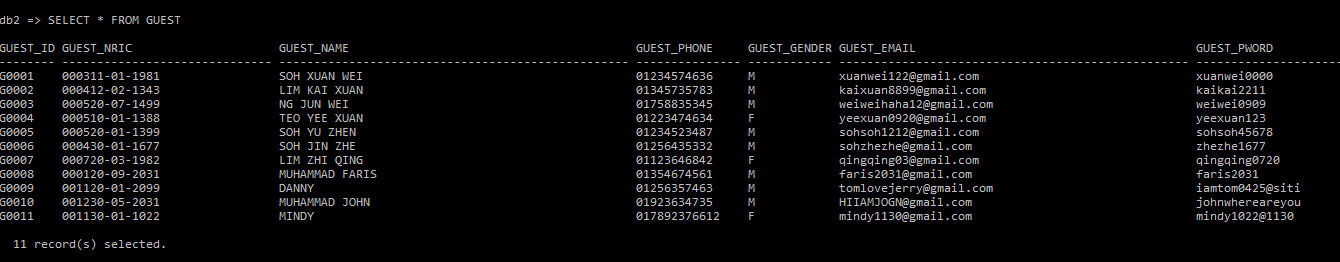
END



Explanation: The code of procedure for change GUEST password



Explanation: Original password result of GUEST



Explanation: The password of SOH XUAN WEI been changed after enter correct email and correct old password.The password change from xuanwei520 to xuanwei0000

To execute this procedure:

**CALL ChangePW('**[**xuanwei122@gmail.com**](mailto:xuanwei122@gmail.com)**', 'xuanwei520', 'xuanwei0000')**

**Procedure to get the promotional price when the date user key in is in promotional period**

CREATE PROCEDURE CHECK\_PROMO (IN CHCK\_IN DATE, CHCK\_OUT DATE)

BEGIN

IF(CHCK\_IN >= '08-31-2020' AND CHCK\_OUT <= '09-01-2020')

THEN UPDATE BOOKINGSUMMARY SET TOTAL\_PRICE = TOTAL\_PRICE\*0.70;

ELSEIF (CHCK\_IN >= '01-25-2020' AND CHCK\_OUT <= '01-26-2020')

THEN UPDATE BOOKINGSUMMARY SET TOTAL\_PRICE = TOTAL\_PRICE\*0.70;

ELSEIF (CHCK\_IN >= '05-23-2020' AND CHCK\_OUT <= '05-24-2020')

THEN UPDATE BOOKINGSUMMARY SET TOTAL\_PRICE = TOTAL\_PRICE\*0.65;

ELSEIF (CHCK\_IN >= '12-31-2019' AND CHCK\_OUT <='01-01-2020')

THEN UPDATE BOOKINGSUMMARY SET TOTAL\_PRICE = TOTAL\_PRICE\*0.75;

ELSEIF (CHCK\_IN >= '02-14-2020' AND CHCK\_OUT <= '02-15-2020')

THEN UPDATE BOOKINGSUMMARY SET TOTAL\_PRICE = TOTAL\_PRICE\*0.75;

ELSEIF (CHCK\_IN >= '12-24-2020' AND CHCK\_OUT <= '12-25-2020')

THEN UPDATE BOOKINGSUMMARY SET TOTAL\_PRICE = TOTAL\_PRICE\*0.80;

ELSEIF (CHCK\_IN >= '09-16-2020' AND CHCK\_OUT <= '09-17-2020')

THEN UPDATE BOOKINGSUMMARY SET TOTAL\_PRICE = TOTAL\_PRICE\*0.90;

ELSEIF (CHCK\_IN >= '11-11-2020' AND CHCK\_OUT <= '11-12-2020')

THEN UPDATE BOOKINGSUMMARY SET TOTAL\_PRICE = TOTAL\_PRICE\*0.90;

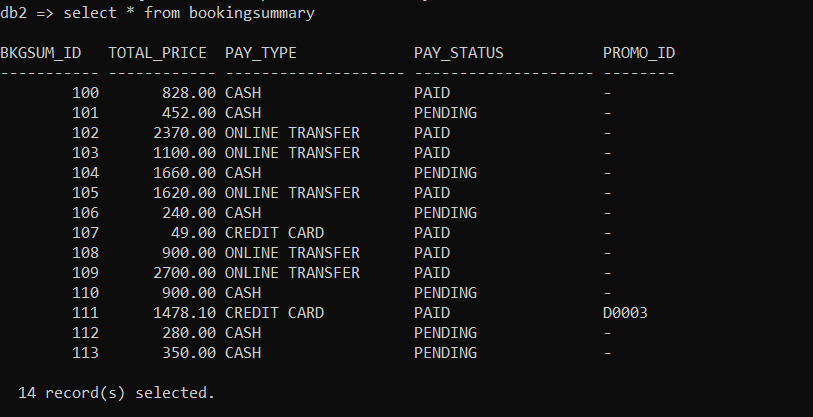
ELSEIF (CHCK\_IN >= '10-01-2020' AND CHCK\_OUT <= '10-02-2020')

THEN UPDATE BOOKINGSUMMARY SET TOTAL\_PRICE = TOTAL\_PRICE\*0.90; ELSEIF (CHCK\_IN >= '10-27-2020' AND CHCK\_OUT <= '10-28-2020') \

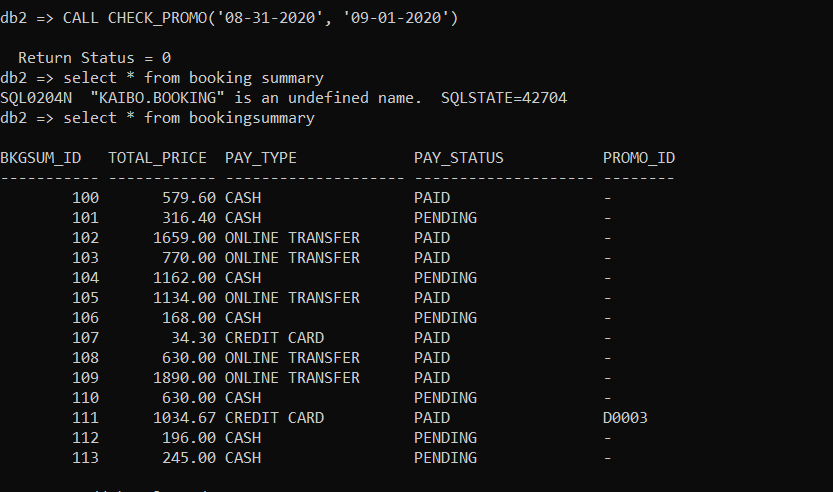
THEN UPDATE BOOKINGSUMMARY SET TOTAL\_PRICE = TOTAL\_PRICE\*0.90;

END IF;

END

****

Explanation: Before the procedure be called

****

Explanation: After the procedure be called, the total price reduce due to promotion

To execute: **CALL CHECK\_PROMO(**'**08-31-2020, 09-01-2020**'**)**

**FUNCTION**

**Room Filter**

CREATE FUNCTION RoomFilter(Budget DECIMAL(6,2), Rating DECIMAL(3,1), NUM\_PEOPLE INTEGER, LOCATION VARCHAR(20))

RETURNS TABLE(HOTEL\_NAME VARCHAR(50), HOTEL\_RATING DECIMAL(3,1), HOTEL\_STATE VARCHAR(20), ROOM\_TYPE VARCHAR(50), ROOM\_NUMPEO INTEGER, ROOM\_PRICE DECIMAL(10,2))

LANGUAGE SQL

READS SQL DATA

NO EXTERNAL ACTION

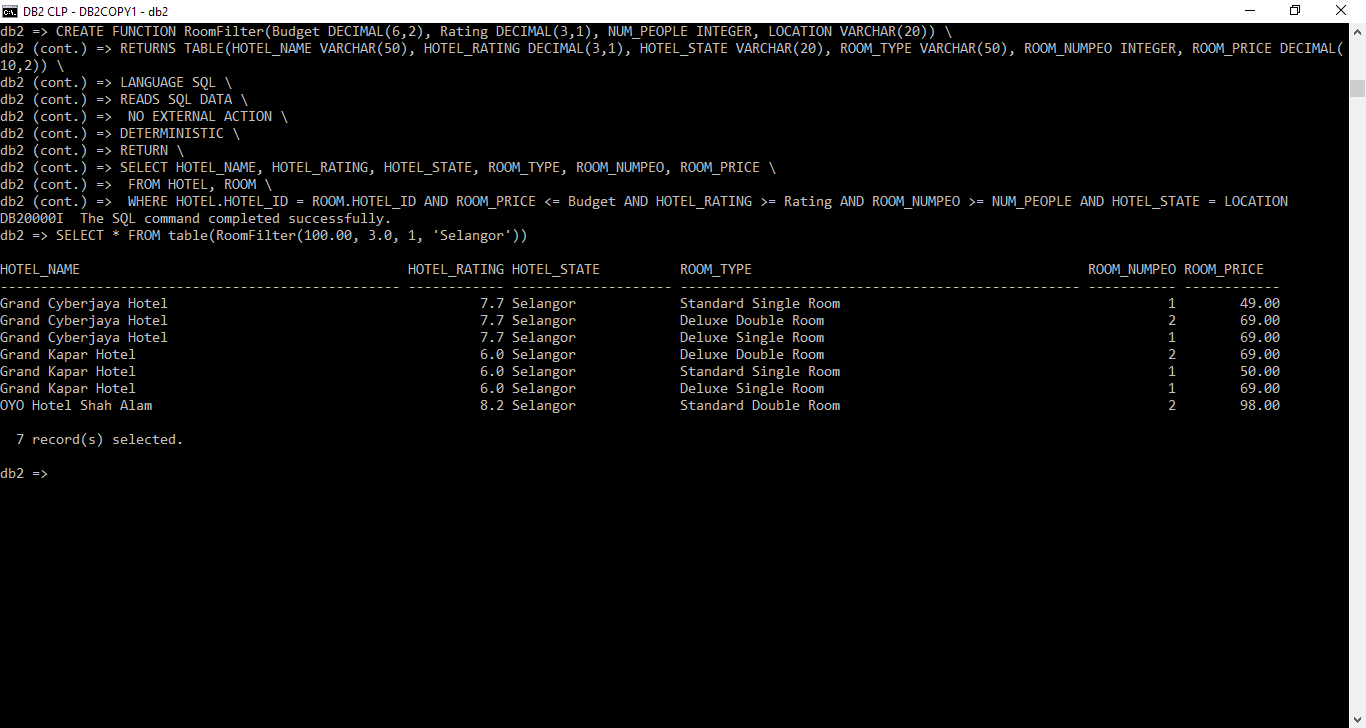
DETERMINISTIC

RETURN

SELECT HOTEL\_NAME, HOTEL\_RATING, HOTEL\_STATE, ROOM\_TYPE, ROOM\_NUMPEO, ROOM\_PRICE

FROM HOTEL, ROOM

WHERE HOTEL.HOTEL\_ID = ROOM.HOTEL\_ID AND ROOM\_PRICE <= Budget AND HOTEL\_RATING >= Rating AND ROOM\_NUMPEO >= NUM\_PEOPLE AND HOTEL\_STATE = LOCATION



Explanation: The function filter the room by the budget, hotel rating, number of people suitable for the room and location of hotel which are inserted by the user.

To execute: SELECT \* FROM table(RoomFilter(100.00, 3.0,1, 'Selangor'))

**Show details of specific type of room for each hotel**

CREATE FUNCTION RoomType(Type VARCHAR(50))

RETURNS TABLE(HOTEL\_NAME VARCHAR(50), ROOM\_TYPE VARCHAR(50), ROOM\_NUMPEO INTEGER, ROOM\_LEFT INTEGER, ROOM\_PRICE DECIMAL(10,2))

LANGUAGE SQL

READS SQL DATA

NO EXTERNAL ACTION

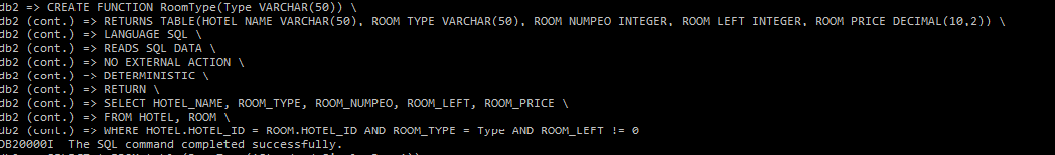
DETERMINISTIC

RETURN

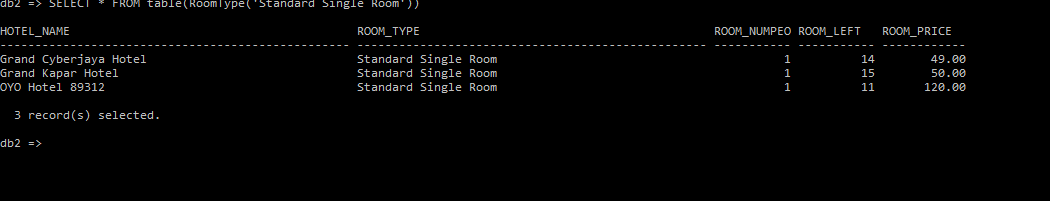
SELECT HOTEL\_NAME, ROOM\_TYPE, ROOM\_NUMPEO, ROOM\_LEFT, ROOM\_PRICE

FROM HOTEL, ROOM

WHERE HOTEL.HOTEL\_ID = ROOM.HOTEL\_ID AND ROOM\_TYPE = Type AND ROOM\_LEFT != 0



Explanation: Code of function that perform details of specific type of room of each hotel



Explanation: The result shows the details of the Standard Single Room of each hotel, if the hotel does not have that kind of room or room is not available then will not show in the result.

To execute: SELECT \* FROM table(RoomType('Standard Single Room'))

## **VIEW**

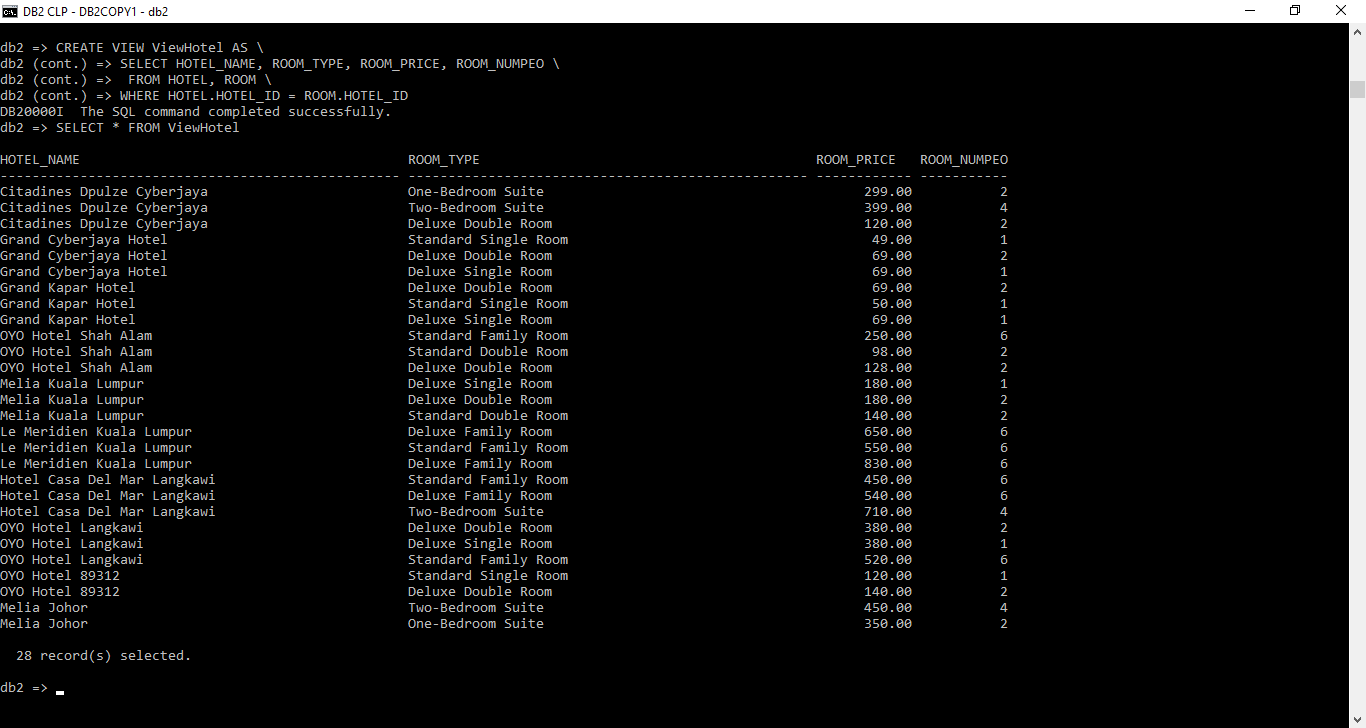
**Quick check for Each Hotel’s Room details**

CREATE VIEW ViewHotel AS

SELECT HOTEL\_NAME, ROOM\_TYPE, ROOM\_PRICE, ROOM\_LEFT

FROM HOTEL, ROOM

WHERE HOTEL.HOTEL\_ID = ROOM.HOTEL\_ID



Explanation : Check the room type, price and room left of each hotel by using view.

## 

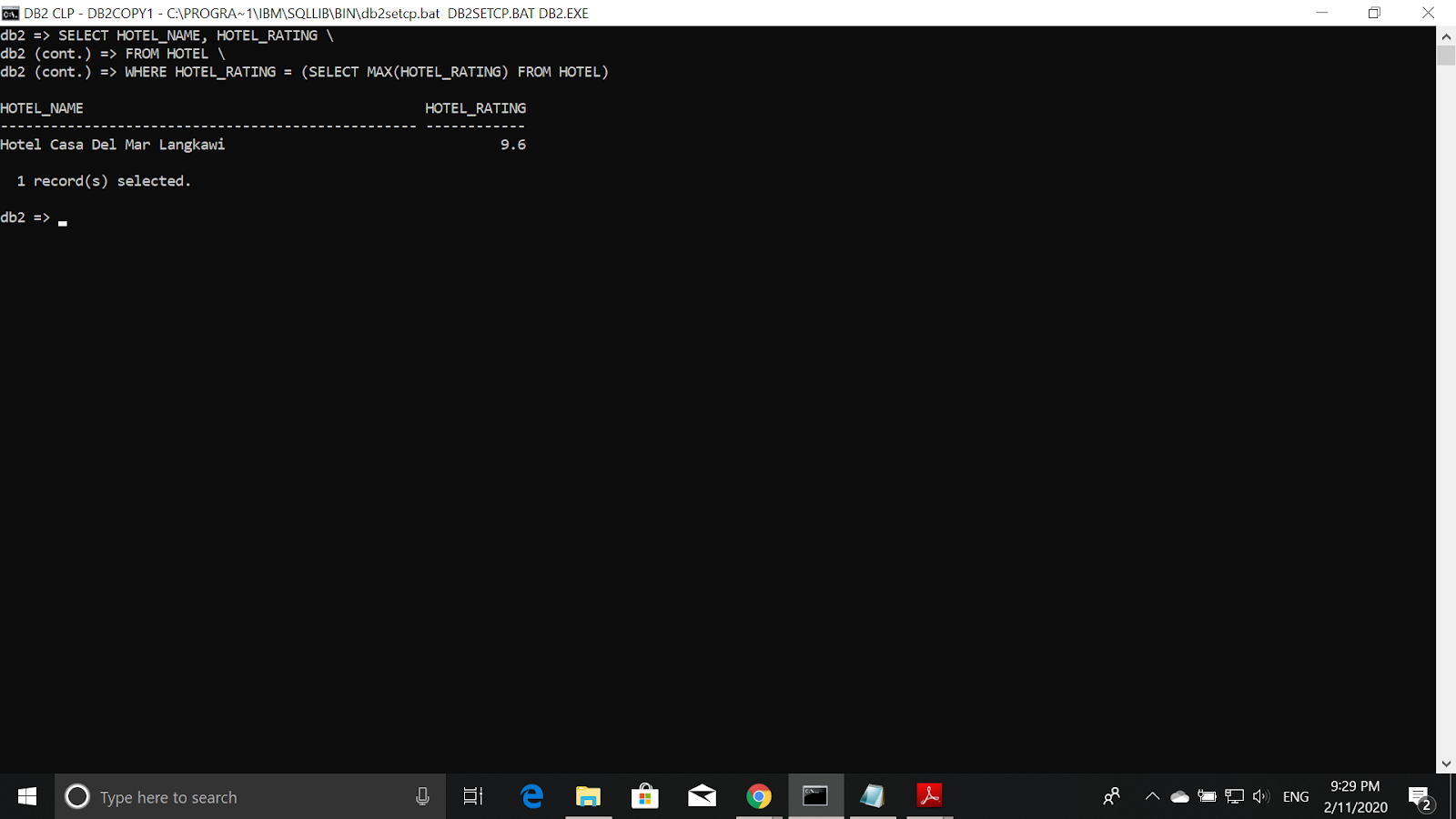
## **SUBQUERIES / NESTED QUERIES**

**The hotel with highest rating**

SELECT HOTEL\_NAME, HOTEL\_RATING

FROM HOTEL

WHERE HOTEL\_RATING = (SELECT MAX(HOTEL\_RATING) FROM HOTEL)



**The guest who made the most payment in the system**

SELECT DISTINCT GUEST\_NAME, TOTAL\_PRICE

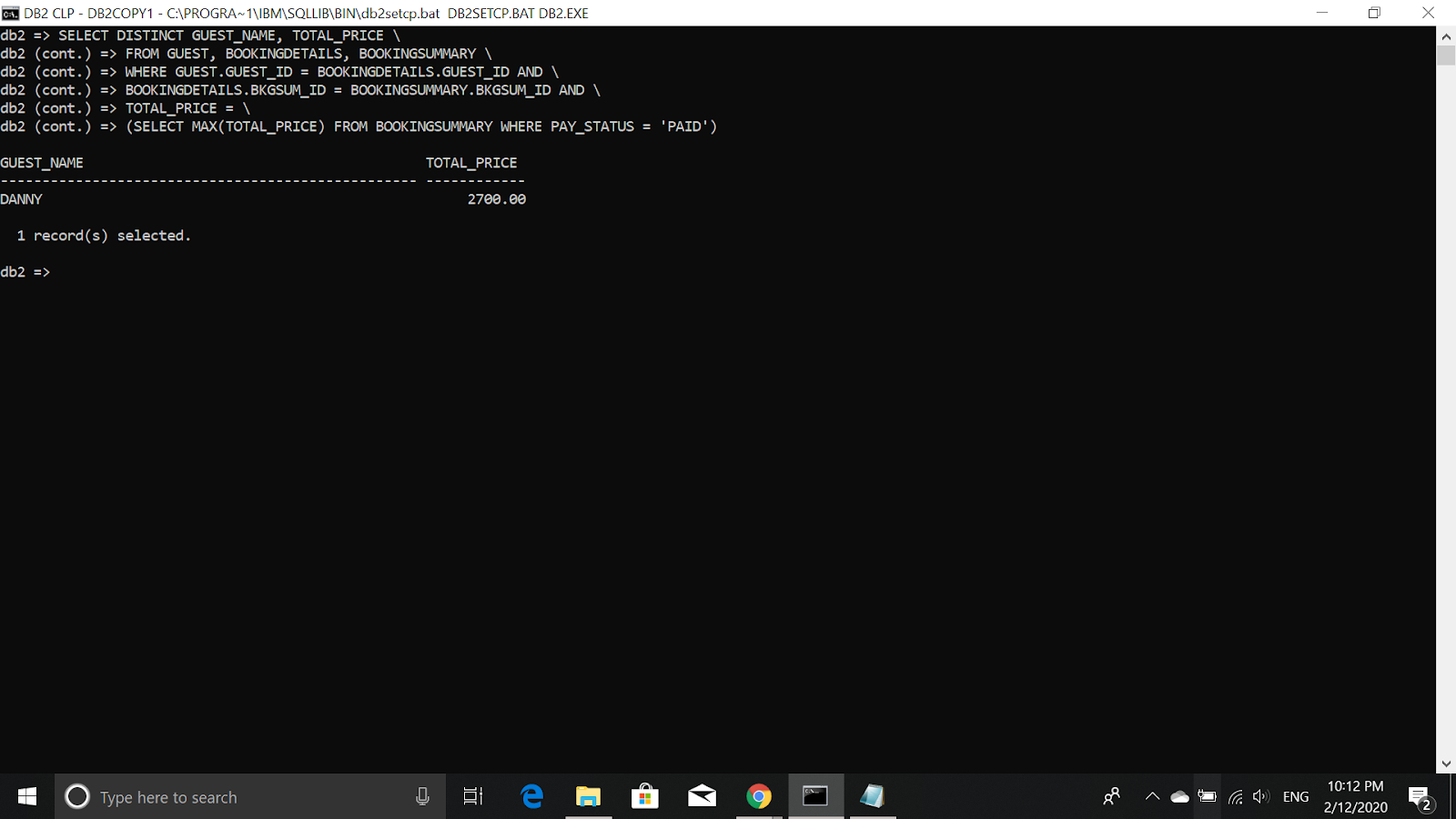
FROM GUEST, BOOKINGDETAILS, BOOKINGSUMMARY

WHERE GUEST.GUEST\_ID = BOOKINGDETAILS.GUEST\_ID AND

BOOKINGDETAILS.BKGSUM\_ID = BOOKINGSUMMARY.BKGSUM\_ID AND

TOTAL\_PRICE = (SELECT MAX(TOTAL\_PRICE) FROM BOOKINGSUMMARY

WHERE PAY\_STATUS = 'PAID')

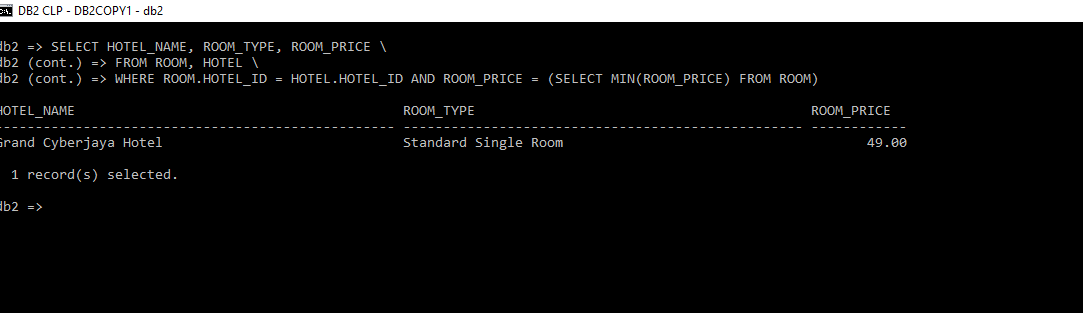


**The Lowest price of hotel’s room**

SELECT HOTEL\_NAME, ROOM\_TYPE, ROOM\_PRICE

FROM ROOM, HOTEL

WHERE ROOM.HOTEL\_ID = HOTEL.HOTEL\_ID AND ROOM\_PRICE = (SELECT MIN(ROOM\_PRICE) FROM ROOM)



## 

## **OTHERS**

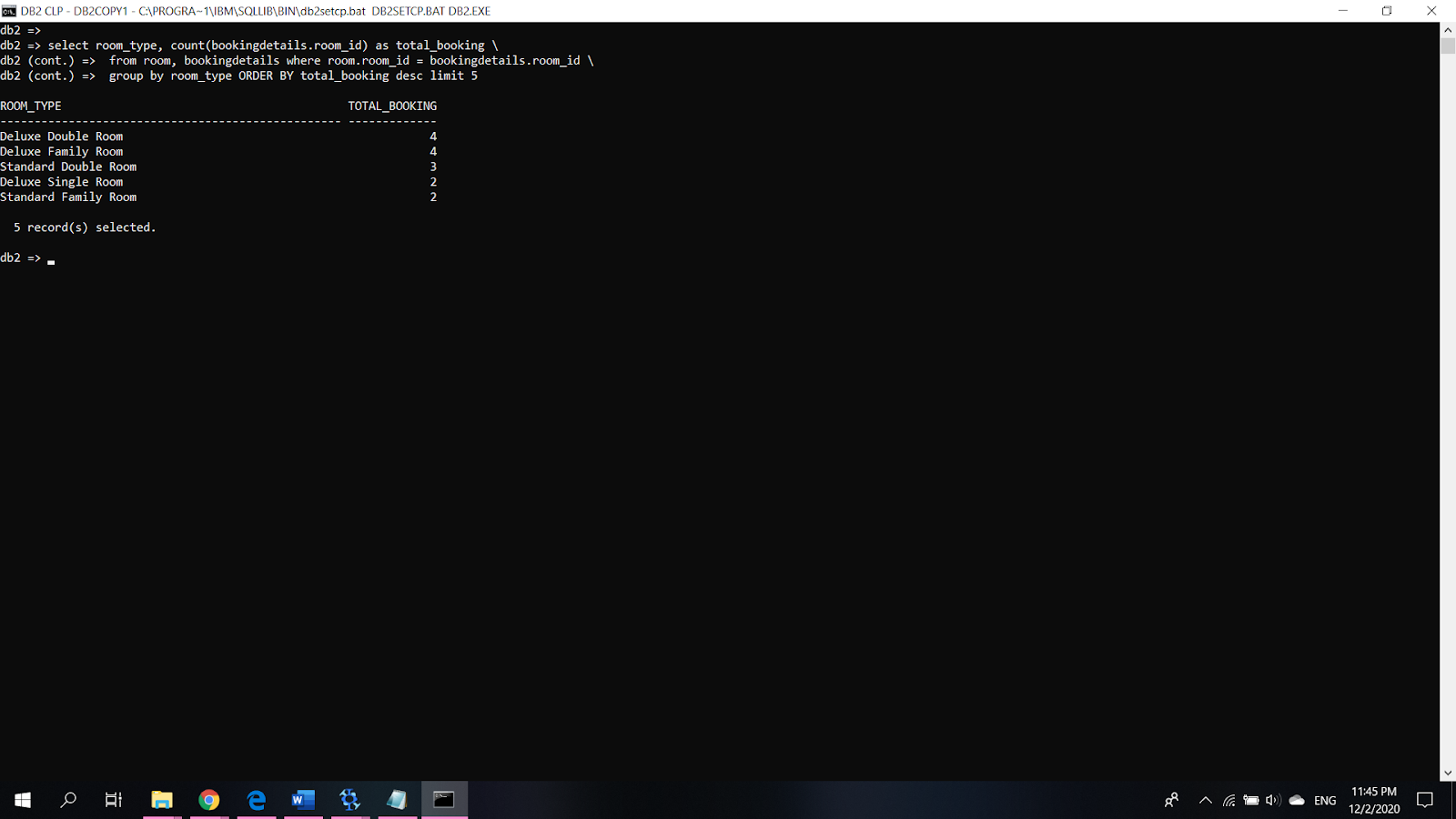
**LIMIT**

**The Top Five Room Types Most Booked by the Guest**

SELECT ROOM\_TYPE, COUNT(BOOKINGDETAILS.ROOM\_ID) AS TOTAL\_BOOKING

FROM ROOM, BOOKINGDETAILS WHERE ROOM.ROOM\_ID = BOOKINGDETAILS.ROOM\_ID

GROUP BY ROOM\_TYPE ORDER BY TOTAL\_BOOKING DESC LIMIT 5



**FETCH**

Display the top 3 highest total price that the guests still hasn’t paid

SELECT DISTINCT BOOKINGSUMMARY.BKGSUM\_ID, GUEST\_NAME,PAY\_TYPE, TOTAL\_PRICE

FROM GUEST, BOOKINGSUMMARY, BOOKINGDETAILS WHERE

BOOKINGSUMMARY.BKGSUM\_ID = BOOKINGDETAILS.BKGSUM\_ID AND GUEST.GUEST\_ID = BOOKINGDETAILS.GUEST\_ID

AND PAY\_STATUS = 'PENDING' ORDER

BY TOTAL\_PRICE DESC FETCH FIRST 3 ROWS ONLY

