

## Database Hotel

/\*Create the hotel database with the defined place\*/

--By Jajang Kusnendar

CREATE DATABASE HOTEL

ON

(NAME = 'Hotel\_dat',  
FILENAME = 'C:\NTC\PROJEK\Q3\Hotel.mdf',  
SIZE = 12,  
MAXSIZE = 50,  
FILEGROWTH = 5)

LOG

ON

(NAME = 'Hotel\_log',  
FILENAME = 'C:\NTC\PROJEK\Q3\Hotel.ldf',  
SIZE = 10,  
MAXSIZE = 50,  
FILEGROWTH = 5)

/\*Create the database with the default place\*/

CREATE DATABASE HOTEL

--The order to use the HOTEL database

USE HOTEL

--To drop the database

DROP DATABASE HOTEL

-----  
-----CREATE TABLE-----  
-----

/\*Create tables for the HOTEL database\*/

--From Entities

--No. 1

--Create table Customer from Entity

CREATE TABLE Customer

(

Customer\_id char(4) NOT NULL  
CONSTRAINT pk\_CusId PRIMARY KEY CLUSTERED,  
Customer\_name varchar(30) NOT NULL,  
Address varchar(25) NOT NULL,  
Gender char(1)  
CONSTRAINT ck\_gender CHECK (Gender='M' or Gender='F'),  
Phone\_no varchar(13),  
Identity\_no char(9) NOT NULL,

)

--to drop the customer tabel

drop table Customer

--The order to return all rows from the Customer Table

SELECT\* FROM Customer

-----  
--No. 2

--Create table Room Category

CREATE TABLE RoomCategory

(

Category\_id char(4) NOT NULL  
CONSTRAINT pk\_CategoryId PRIMARY KEY CLUSTERED,  
Category\_name varchar(10),  
Room\_price money

)

--Untuk menghilangkan table RoomCategory

drop table RoomCategory

--The order to return all rows from RoomCategory Table

SELECT\* FROM RoomCategory

## Database Hotel

```
-----  
--No. 3  
--Create table RoomStatus from Entity  
CREATE TABLE RoomStatus  
(  
    Room_id          char(4) NOT NULL  
                    CONSTRAINT pk_roomid PRIMARY KEY CLUSTERED,  
    Category_id      char(4)  
                    CONSTRAINT fk_CategoryId FOREIGN KEY  
                    REFERENCES RoomCategory(Category_id),  
    Room_status      varchar(4)  
                    CONSTRAINT chk_status  
                    CHECK (Room_status=' ON' or Room_status=' OFF' )  
)
```

```
--Untuk menghapus table RoomStatus  
drop table RoomStatus
```

```
--The order to return all rows from RoomStatus Table  
SELECT* FROM RoomStatus
```

```
-----  
--No. 4  
--Create table Employee  
CREATE TABLE Employee  
(  
    Employee_id      char(4) NOT NULL  
                    CONSTRAINT pk_EmpCode PRIMARY KEY CLUSTERED,  
    Employee_name     varchar(23) NOT NULL,  
    Address           varchar(30),  
    Phone_no          char(13)  
                    CONSTRAINT chk_phone  
                    CHECK (Phone_no LIKE  
' ([0-9][0-9][0-9]) - [0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
    Gender            char(2)  
                    CONSTRAINT ck_gender2  
                    CHECK (Gender=' M' or Gender=' F' ),  
)
```

```
--Untuk menghilangkan Table Employee  
drop table Employee
```

```
--The order to return all rows from Employee table  
SELECT* FROM Employee
```

```
-----  
/*Membuat tabel dari relationship */
```

```
--No. 5  
--Create table Booking  
CREATE TABLE Booking  
(  
    Booking_no        char(4) NOT NULL  
                    CONSTRAINT pk_bookno PRIMARY KEY CLUSTERED,  
    Booking_date       datetime,  
    Booking_name       varchar(23) NOT NULL,  
    CheckIn_date       datetime  
)
```

```
--Untuk menghilangkan Table Booking  
drop table Booking
```

```
--The order to return all rows from Booking table  
SELECT* FROM Booking
```

```
-----  
--No. 6  
--Create table BookingDetail from relationship  
CREATE TABLE BookingDetail
```

Database Hotel

```
(
    BookingDetail_id    char(4)
                        CONSTRAINT pk_bookdtl PRIMARY KEY CLUSTERED,
    Booking_no          char(4)
                        CONSTRAINT fk_bookno FOREIGN KEY
                        REFERENCES Booking(Booking_no),
    Category_id         char(4)
                        CONSTRAINT fk_roomid FOREIGN KEY
                        REFERENCES RoomCategory(Category_id),
    Total_room          int
)

```

```
--Untuk menghilangkan Table BookingDetail
drop table BookingDetail
```

```
--The order to return all rows from BookingDetail Table
SELECT* FROM BookingDetail
```

-----  
--No. 7

```
--Create table Payment
CREATE TABLE PaymentMode
```

```
(
    PaymentMode_id    char(6)
                        CONSTRAINT pk_payment PRIMARY KEY CLUSTERED,
    Payment_mode       varchar(10)
)

```

```
--Untuk menghilangkan Table Payment
drop table PaymentMode
```

```
--The order to return all rows from payment table
SELECT* FROM PaymentMode
```

-----  
--No. 8

```
--Create table Transaction
CREATE TABLE Transactions
```

```
(
    Transaction_no     char(4) NOT NULL
                        CONSTRAINT pk_transNo PRIMARY KEY CLUSTERED,
    Employee_id        char(4)
                        CONSTRAINT Fk_EmpCode FOREIGN KEY
                        REFERENCES Employee(Employee_id),
    Transaction_date    datetime,
    Customer_id         char(4)
                        CONSTRAINT fk_CusId FOREIGN KEY
                        REFERENCES Customer(Customer_id),
    Total_cost          money,
    PaymentMode_id     char(6)
                        CONSTRAINT fk_paymentmode FOREIGN KEY
                        REFERENCES PaymentMode(PaymentMode_id)
)

```

```
--Untuk mengilangkan table Transactions
drop table Transactions
```

```
--The order to return all rows from Transactions table
SELECT* FROM Transactions
```

-----  
--No. 1

```
--create function to count total day in TransactionDetail Table
CREATE FUNCTION CountTotalDay (@CheckIn_date datetime, @CheckOut_date datetime)
RETURNS int
AS
BEGIN
    RETURN (datediff(dd, @CheckOut_date, @CheckIn_date))
END
```

## Database Hotel

```
--No. 9
--Create table Transaction detail
CREATE TABLE TransactionDetail
(
    TransactionDetail_id char(5) NOT NULL
    CONSTRAINT pk_TransDtl_Id PRIMARY KEY CLUSTERED,
    Transaction_no char(4)
    CONSTRAINT fk_Transno FOREIGN KEY
    REFERENCES Transactions(Transaction_no),
    CheckIn_date datetime,
    CheckOut_date datetime,
    Room_id char(4)
    CONSTRAINT fk_RoomId2 FOREIGN KEY
    REFERENCES RoomStatus(Room_id),
    Room_price money
    CONSTRAINT Chk_roomPrice
    CHECK (Room_price=1250000 or Room_price=750000 or
Room_price=500000),
    Total_day AS (dbo. CountTotalDay(CheckOut_date, CheckIn_date))
)
```

```
--Untuk menghilangkan table TransactionDetail
drop table TransactionDetail
```

```
--The order to return all rows from TransactionDetail table
SELECT* FROM TransactionDetail
```

```
-----
--No. 10
--Create table Credit Card
CREATE TABLE CreditCard
(
    CreditCard_no char(12)
    CONSTRAINT pk_creditCardno PRIMARY KEY CLUSTERED,
    CardHolders_name varchar(30),
    CardExpiryDate datetime
)
```

```
--Untuk menghilangkan table Credit Card
drop table CreditCard
```

```
--To return all rows from Credit card table
SELECT* FROM CreditCard
```

```
-----
--No. 11
--Create table Payment
CREATE TABLE Payment
(
    Payment_id char(4) NOT NULL
    CONSTRAINT pk_PaymentId PRIMARY KEY CLUSTERED,
    Payment_date datetime,
    Transaction_no char(4)
    CONSTRAINT fk_TransNo2 FOREIGN KEY
    REFERENCES Transactions(Transaction_no),
    PaymentMode_id char(6)
    CONSTRAINT fk_paymentmode1 FOREIGN KEY
    REFERENCES PaymentMode(PaymentMode_id),
    CreditCard_no char(12)
    CONSTRAINT fk_CreditCard FOREIGN KEY
    REFERENCES CreditCard(CreditCard_no)
)
```

```
--To remove Payment table
drop table Payment
```

```
--To return all rows from Payment Table
SELECT* FROM Payment
```

## Database Hotel

```
-----  
--No. 12  
--Create table Identity  
CREATE TABLE IdentityCustomer  
(  
    Identity_no      char (9) NOT NULL  
                    CONSTRAINT pk_IdentityNo PRIMARY KEY CLUSTERED,  
    Identity_type    varchar (5)  
)  
  
--To remove Payment table  
drop table IdentityCustomer  
  
--To return all rows from Payment Table  
SELECT* FROM IdentityCustomer
```

### ----- ----- INSERT ----- -----

```
--No. 1  
--Insert values into Customer table  
INSERT INTO Customer  
VALUES ('0001', 'Michael  
Jordan', 'Jl. W. R. Supratman, Jakarta', 'M', '(021)-5841234', '10100489')  
INSERT INTO Customer  
VALUES ('0002', 'Kobe Bryant', 'Jl. Jend. Sudirman,  
Jakarta', 'M', '(021)-5843421', '11100588')  
INSERT INTO Customer  
VALUES ('0003', 'Steve Nash', 'Jl. Rasuna Said,  
Jakarta', 'M', '(021)-5843423', '12100687')  
INSERT INTO Customer  
VALUES ('0004', 'Maria Renata', 'Jl. Gjh Mada,  
Jakarta', 'F', '(021)-5843454', '13100786')  
INSERT INTO Customer  
VALUES ('0005', 'Marchy Namira', 'Jl. P. J. Mangu,  
Jakarta', 'F', '(021)-5845674', '14100885')  
  
SELECT* FROM Customer
```

```
-----  
--No. 2  
--Insert values into RoomCategory  
INSERT INTO RoomCategory  
VALUES ('DL01', 'Deluxe', 1250000)  
INSERT INTO RoomCategory  
VALUES ('ST02', 'Standard', 750000)  
INSERT INTO RoomCategory  
VALUES ('SM03', 'Small', 500000)  
  
SELECT* FROM RoomCategory
```

```
-----  
--No. 3  
--Insert values into RoomStatus table  
INSERT INTO RoomStatus  
VALUES ('R001', 'DL01', 'OFF')  
INSERT INTO RoomStatus  
VALUES ('R002', 'DL01', 'OFF')  
INSERT INTO RoomStatus  
VALUES ('R003', 'DL01', 'ON')  
INSERT INTO RoomStatus  
VALUES ('R004', 'DL01', 'ON')  
INSERT INTO RoomStatus
```

## Database Hotel

```
VALUES ('R005', 'DL01', 'ON')
INSERT INTO RoomStatus
VALUES ('R006', 'DL01', 'ON')
INSERT INTO RoomStatus
VALUES ('R007', 'DL01', 'ON')
INSERT INTO RoomStatus
VALUES ('R008', 'DL01', 'ON')
INSERT INTO RoomStatus
VALUES ('R009', 'DL01', 'ON')
INSERT INTO RoomStatus
VALUES ('R010', 'DL01', 'ON')
INSERT INTO RoomStatus
VALUES ('R011', 'ST02', 'OFF')
INSERT INTO RoomStatus
VALUES ('R012', 'ST02', 'OFF')
INSERT INTO RoomStatus
VALUES ('R013', 'ST02', 'ON')
INSERT INTO RoomStatus
VALUES ('R014', 'ST02', 'ON')
INSERT INTO RoomStatus
VALUES ('R015', 'ST02', 'ON')
INSERT INTO RoomStatus
VALUES ('R016', 'ST02', 'ON')
INSERT INTO RoomStatus
VALUES ('R017', 'ST02', 'ON')
INSERT INTO RoomStatus
VALUES ('R018', 'ST02', 'ON')
INSERT INTO RoomStatus
VALUES ('R019', 'ST02', 'ON')
INSERT INTO RoomStatus
VALUES ('R020', 'ST02', 'ON')
INSERT INTO RoomStatus
VALUES ('R021', 'SM03', 'OFF')
INSERT INTO RoomStatus
VALUES ('R022', 'SM03', 'OFF')
INSERT INTO RoomStatus
VALUES ('R023', 'SM03', 'ON')
INSERT INTO RoomStatus
VALUES ('R024', 'SM03', 'ON')
INSERT INTO RoomStatus
VALUES ('R025', 'SM03', 'ON')
INSERT INTO RoomStatus
VALUES ('R026', 'SM03', 'ON')
INSERT INTO RoomStatus
VALUES ('R027', 'SM03', 'ON')
INSERT INTO RoomStatus
VALUES ('R028', 'SM03', 'ON')
INSERT INTO RoomStatus
VALUES ('R029', 'SM03', 'ON')
INSERT INTO RoomStatus
VALUES ('R030', 'SM03', 'ON')
```

```
SELECT* FROM RoomStatus
```

-----  
-- No. 4

-- Insert values into Employee table

```
INSERT INTO Employee
VALUES ('E001', 'Aston Martin', 'Jl. Perintis, Jakarta', '(021)-5867234', 'M')
INSERT INTO Employee
VALUES ('E002', 'Lindsay Lohan', 'Jl. Bumi, Jakarta', '(021)-5867452', 'F')
INSERT INTO Employee
VALUES ('E003', 'Flying Dutchman', 'Jl. Brawijaya, Jakarta', '(021)-5867234', 'M')
INSERT INTO Employee
VALUES ('E004', 'Britney Spears', 'Jl. Melawai, Jakarta', '(021)-5867234', 'F')
INSERT INTO Employee
VALUES ('E005', 'Hillary Duff', 'Jl. Petukangan, Jakarta', '(021)-5863434', 'F')
```

```
SELECT* FROM Employee
```

## Database Hotel

```
-----  
--No. 5  
--Insert values into Booking table  
INSERT INTO Booking  
VALUES ('B001', '2008/04/14', 'Doni', '2008/04/15')  
INSERT INTO Booking  
VALUES ('B002', '2008/04/14', 'Dini', '2008/04/16')  
INSERT INTO Booking  
VALUES ('B003', '2008/04/14', 'Deni', '2008/04/17')  
INSERT INTO Booking  
VALUES ('B004', '2008/04/14', 'Dani', '2008/04/15')  
INSERT INTO Booking  
VALUES ('B005', '2008/04/14', 'Dono', '2008/04/16')  
  
SELECT* FROM Booking  
-----  
--No. 6  
--Insert values into Booking Detail Table  
INSERT INTO BookingDetail  
VALUES ('D001', 'B001', 'DL01', '2')  
INSERT INTO BookingDetail  
VALUES ('D002', 'B002', 'ST02', '3')  
INSERT INTO BookingDetail  
VALUES ('D003', 'B003', 'DL01', '2')  
INSERT INTO BookingDetail  
VALUES ('D004', 'B004', 'SM03', '2')  
INSERT INTO BookingDetail  
VALUES ('D005', 'B005', 'SM03', '1')  
  
SELECT* FROM BookingDetail  
-----  
--No. 7  
--Insert values into Payment Mode  
INSERT INTO PaymentMode  
VALUES ('PM01', 'Cash')  
INSERT INTO PaymentMode  
VALUES ('PM02', 'CreditCard')  
  
SELECT* FROM PaymentMode  
-----  
--No. 8  
--insert values into Credit Card table  
INSERT INTO CreditCard  
VALUES ('053012300500', 'Michael Jordan', '2010/04/16')  
INSERT INTO CreditCard  
VALUES ('053012300501', 'Kobe Bryant', '2010/02/16')  
INSERT INTO CreditCard  
VALUES ('053012300502', 'Marchy Nami ra', '2010/04/20')  
  
SELECT* FROM CreditCard  
-----  
--No. 9  
--Insert values into Transactions  
INSERT INTO Transactions  
VALUES ('T001', 'E001', '2008/04/14', '0001', 3750000, 'PM02')  
INSERT INTO Transactions  
VALUES ('T002', 'E002', '2008/04/14', '0002', 4750000, 'PM02')  
INSERT INTO Transactions  
VALUES ('T003', 'E001', '2008/04/14', '0003', 2250000, 'PM01')  
INSERT INTO Transactions  
VALUES ('T004', 'E004', '2008/04/14', '0004', 1500000, 'PM01')  
INSERT INTO Transactions  
VALUES ('T005', 'E005', '2008/04/14', '0005', 3000000, 'PM02')  
  
SELECT* FROM Transactions
```

Database Hotel

```
-----  
--No. 10  
--Insert values into TransactionDetail  
INSERT INTO TransactionDetail  
VALUES ('TD001', 'T001', '2008/04/15', '2008/04/18', 'R001', 1250000)  
INSERT INTO TransactionDetail  
VALUES ('TD002', 'T002', '2008/04/16', '2008/04/18', 'R002', 1250000)  
INSERT INTO TransactionDetail  
VALUES ('TD003', 'T002', '2008/04/16', '2008/04/19', 'R011', 750000)  
INSERT INTO TransactionDetail  
VALUES ('TD004', 'T003', '2008/04/17', '2008/04/20', 'R021', 750000)  
INSERT INTO TransactionDetail  
VALUES ('TD005', 'T004', '2008/04/15', '2008/04/18', 'R022', 500000)  
INSERT INTO TransactionDetail  
VALUES ('TD006', 'T005', '2008/04/16', '2008/04/20', 'R012', 750000)  
  
SELECT* FROM TransactionDetail
```

```
-----  
--No. 11  
--Insert values into Payment  
INSERT INTO Payment  
VALUES ('P001', '2008/04/14', 'T001', 'PM02', '053012300500')  
INSERT INTO Payment  
VALUES ('P002', '2008/04/14', 'T002', 'PM02', '053012300501')  
INSERT INTO Payment  
VALUES ('P003', '2008/04/15', 'T003', 'PM01', NULL)  
INSERT INTO Payment  
VALUES ('P004', '2008/04/15', 'T004', 'PM01', NULL)  
INSERT INTO Payment  
VALUES ('P005', '2008/04/15', 'T005', 'PM02', '053012300502')  
  
SELECT* FROM Payment
```

```
-----  
--No. 12  
--Insert values into IdentityCustomer  
INSERT INTO IdentityCustomer  
VALUES ('10100489', 'KTP')  
INSERT INTO IdentityCustomer  
VALUES ('11100588', 'KTP')  
INSERT INTO IdentityCustomer  
VALUES ('12100687', 'KTP')  
INSERT INTO IdentityCustomer  
VALUES ('13100786', 'KTP')  
INSERT INTO IdentityCustomer  
VALUES ('14100885', 'KTP')  
  
SELECT* FROM IdentityCustomer
```

```
-----  
----- VIEW -----  
-----
```

```
-- Create view from multiple table from :  
SELECT* FROM Customer  
SELECT* FROM Transactions  
SELECT* FROM TransactionDetail  
SELECT* FROM Payment
```

```
--><--
```

```
--No. 1  
CREATE VIEW vwCustomerCost  
AS
```



Database Hotel

```
SELECT C.Customer_name, Td.Room_id, Td.Total_day, T.Total_cost
FROM Customer C, TransactionDetail Td, Transactions T
WHERE C.Customer_id = T.Customer_id AND T.Transaction_no =
Td.Transaction_no
```

```
--to return the result
SELECT* FROM vwCustomerCost
```

```
--Create view from multiple table
SELECT * FROM Booking
SELECT * FROM BookingDetail
SELECT * FROM RoomCategory
```

```
--><--
```

```
--No. 2
CREATE VIEW vwBookingDetail
AS
SELECT b.Booking_no, b.Booking_name, b.CheckIn_date, bd.Total_room,
r.Category_id,r.Category_name
FROM Booking b, BookingDetail bd, RoomCategory r
WHERE b.Booking_no = bd.Booking_no AND bd.Category_id = r.Category_id
```

```
--to return the result
SELECT * FROM vwBookingDetail
```

```
--No. 3
CREATE VIEW vwCustomerDetails
AS
SELECT c.Customer_id, c.Customer_name, t.Transaction_no, td.CheckIn_date,
r.Room_id, r.Room_status, t.Total_cost
FROM Customer c, Transactions t, RoomStatus r, TransactionDetail td
WHERE c.Customer_id = t.Customer_id AND t.Transaction_no = td.Transaction_no
AND td.Room_id = r.Room_id
```

```
--to return the result
SELECT * FROM vwCustomerDetails
```

```
-----
----- I N D E X -----
-----
```

```
--Create index from table Customer and Room Status
```

```
CREATE NONCLUSTERED INDEX idxCustomer
ON Customer(Customer_id)

CREATE NONCLUSTERED INDEX idxRoomStatus
ON RoomStatus(Room_id)
```

```
--to check the result
sp_helpindex Customer
sp_helpindex RoomStatus
```

```
-----
----- F U N C T I O N -----
-----
```

```
--No. 1
--Create function to search the Customer Name
CREATE FUNCTION fn_SearchName (@CustomerName varchar(30))
RETURNS TABLE
AS
RETURN (SELECT *
FROM Customer
WHERE Customer_name LIKE '%' + @CustomerName+'%')
```

## Database Hotel

```
--To return the result of function
SELECT * FROM fn_SearchName(' Marchy' )

-----
----- RULE -----
-----

-- No. 1
--Create rule for Phone no
CREATE RULE rulPhone
  AS @phone Like '([0-9][0-9][0-9])-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]'

--Rule is added to phone no in customer table
sp_bindrule 'rulPhone', 'Customer.Phone_no'

--for check
--that false
INSERT INTO Customer
  VALUES ('0006', 'Priska', 'Jl. Milan', 'F', '(027)-12834567', '09876543')

--that true
INSERT INTO Customer
  VALUES ('0006', 'Priska', 'Jl. Milan', 'F', '(024)-1234567', '09876543')

-----

--No. 2
--Create rule
CREATE RULE rulTotalCost
  AS @TotalCost>0

--Rule is added to total cost in Transactions table
sp_bindrule 'rulTotalCost', 'Transactions.Total_cost'
sp_help rulTotalCost

--that false
INSERT INTO Transactions
  VALUES ('T006', 'E002', '2008/04/16', '0006', -1500000, 'PM01')

--that true
INSERT INTO Transactions
  VALUES ('T006', 'E002', '2008/04/16', '0006', 1500000, 'PM01')

-----
----- AGGREGATE FUNCTIONS -----
-----

--To count the total record from the customer table (without Distinct)
SELECT 'Total Customer' =COUNT (*) FROM Customer

--To count the available room with the Deluxe type
SELECT 'Available Room-Deluxe' = COUNT(*)
FROM RoomStatus
WHERE Room_status = 'ON' AND Category_id = 'DL01'

--To count the available room with the Standard type
SELECT 'Available Room-Standard' = COUNT(*)
FROM RoomStatus
WHERE Room_status = 'ON' AND Category_id = 'ST02'

--To count the available room with the Small type
SELECT 'Available Room-Small' = COUNT(*)
```

Database Hotel

```
FROM RoomStatus
WHERE Room_status = 'ON' AND Category_id = 'SM03'
```

```
/*The GROUP BY is used summaries the result set as defined in the query
using aggregate functions*/
```

```
--
SELECT rs.Category_id, 'Room Total'=COUNT(Room_status), r.Category_name
FROM RoomStatus rs, RoomCategory r
WHERE rs.Category_id = r.Category_id
GROUP BY r.Category_name, rs.Category_id
```

```
--
SELECT rs.Category_id, 'Room Total'=COUNT(Room_status), r.Category_name
FROM RoomStatus rs, RoomCategory r
WHERE rs.Category_id = r.Category_id AND rs.Room_status = 'ON'
GROUP BY r.Category_name, rs.Category_id
```

```
--
SELECT rs.Category_id, 'Room Total'=COUNT(Room_status), r.Category_name
FROM RoomStatus rs, RoomCategory r
WHERE rs.Category_id = r.Category_id AND rs.Room_status = 'OFF'
GROUP BY r.Category_name, rs.Category_id
```

```
--
SELECT t.Transaction_no AS 'Transaction No', t.Customer_id AS 'Customer Id',
      td.Room_price AS 'Room Price', Total_cost AS 'Total Cost'
FROM Transactions t, TransactionDetail td
ORDER BY t.Transaction_no ASC
COMPUTE SUM(td.Room_price)
```

```
-----
----- PROCEDURE -----
-----
```

```
-- No. 1
-- Store procedure to return details of Customer
CREATE PROCEDURE prcDisplayCustomer @Customer_name varchar (30)
AS
BEGIN
    IF @Customer_name is NULL
    BEGIN
        PRINT 'Usage = Please enter the name'
        RETURN
    END
    PRINT 'List of Customer'
    SELECT DISTINCT 'Customer Id'=c.Customer_id, 'Customer
Name'=c.Customer_name, 'Address'=c.Address, 'Identity Number'=c.Identity_no,
'Identity Type'=i.Identity_type, 'Room Number'=rs.Room_id, 'Date of Check
In'=td.CheckIn_date, 'Date of Check Out'=td.CheckOut_date, 'Total
Cost'=t.Total_cost
FROM Customer c, RoomStatus rs, TransactionDetail td, Transactions t,
IdentityCustomer i
WHERE c.Customer_id=t.Customer_id AND t.Transaction_no=td.Transaction_no
AND td.Room_id=rs.Room_id AND c.Customer_name=@Customer_name
END
```

```
--To remove the procedure
drop procedure prcDisplayCustomer
```

```
--To check the availability from the procedure
sp_helptext prcDisplayCustomer
```

```
--To display the procedure
prcDisplayCustomer 'Kobe Bryant'
```

```
-----
-- No. 2
```

Database Hotel

```
--Store procedure to return room status
CREATE PROCEDURE prcTotalRoom @RoomCategory char(4)
AS
BEGIN
    IF @RoomCategory is NULL
    BEGIN
        PRINT 'Usage = Please enter the room category'
        RETURN
    END
    PRINT 'List of available room'
    SELECT 'Room id'=r.Room_id, 'Room Name'=rc.Category_name, 'Room
Status'=r.Room_status
    FROM RoomStatus r, RoomCategory rc
    WHERE r.Category_id=rc.Category_id AND rc.Category_id=@RoomCategory AND
r.Room_status = 'ON'
END

--To remove the procedure
drop procedure prcTotalRoom

--To check the availability from the procedure
sp_help prcTotalRoom

--To display the procedure
prcTotalRoom 'DL01'

prcTotalRoom 'ST02'

prcTotalRoom 'SM03'
```

```
-----
--No. 3
--Store procedure to return all details of customer from the customer table
CREATE PROCEDURE prcCustomer @cusid char(4)
AS
    SELECT * FROM Customer
    WHERE @cusid = Customer_id

--to display the result
EXEC prcCustomer '0001'
```

```
-----
--No. 4
--Store procedure to return all details of transaction from the transaction and
transaction
--detail table
CREATE PROCEDURE prcTransactions @transno char(4)
AS
    SELECT DISTINCT * FROM Transactions t, TransactionDetail td
    WHERE t.Transaction_no = td.Transaction_no AND @transno = t.Transaction_no

DROP PROC prcTransactions

--To display the result
EXEC prcTransactions 'T001'
```

```
-----
----- TRANSACTION -----
-----
--No. 1
--Create transaction for RoomStatus
BEGIN TRANSACTION
INSERT INTO Customer
VALUES ('0007', 'Redha Ramadhian', 'Jl. Turin
Raya', 'M', '(021)-5860403', '19257604')
INSERT INTO Transactions
VALUES ('T007', 'E001', '2008/04/16', '0007', 2500000, 'PM01')
INSERT INTO TransactionDetail
VALUES ('TD007', 'T007', '2008/04/16', '2008/04/18', 'R003', 1250000)
```

Database Hotel

```
UPDATE RoomStatus
SET Room_status = 'OFF'
WHERE Room_id = 'R003'
IF (SELECT Room_status
    FROM RoomStatus
    WHERE Room_id = 'R003') != 'OFF'
    BEGIN
        PRINT 'Transaction Failed : choose another room'
        ROLLBACK TRAN
    END
ELSE
    BEGIN
        PRINT 'Transaction Committed'
        COMMIT TRAN
    END
END
```

```
update RoomStatus
SET Room_status = 'ON'
WHERE Room_id = 'R003'
```

```
select* from RoomStatus
delete from Customer
where Customer_id = '0007'
delete from Transactions
where Transaction_no = 'T007'
delete from TransactionDetail
where TransactionDetail_id = 'TD007'
```

```
-----
--No. 2
--To calculate the total cost by automatically
BEGIN TRANSACTION
declare @totalCost money
SELECT @totalCost = Total_cost FROM Transactions
INSERT INTO Customer
VALUES ('0008', 'Angelina Jolie', 'Jl. Pelangi', 'F', '(021)-5860237', '19983404')
INSERT INTO Transactions
VALUES ('T008', 'E002', '2008/04/16', '0008', @totalCost, 'PM01')
INSERT INTO TransactionDetail
VALUES ('TD008', 'T008', '2008/04/16', '2008/04/18', 'R004', 1250000)
UPDATE Transactions
SET @totalCost = Total_day * 1250000
FROM Transactions t, TransactionDetail td
WHERE t.Transaction_no = td.Transaction_no AND t.Transaction_no = 'T008' AND
Total_cost=@totalCost
UPDATE RoomStatus
SET Room_status = 'OFF'
WHERE Room_id = 'R004'
IF (SELECT Room_status FROM RoomStatus WHERE Room_id = 'R004') != 'OFF'
    BEGIN
        PRINT 'Transaction Failed'
        ROLLBACK TRAN
    END
ELSE
    BEGIN
        PRINT 'Transaction Committed'
        COMMIT TRAN
    END
END
```

```
-----
--
----- TRIGGER
-----
```

## Database Hotel

```
--create trigger to insert date
CREATE TRIGGER trgInsertDate
ON Booking
FOR INSERT
AS
DECLARE @Booking_date datetime
DECLARE @CheckIn_date datetime
SELECT @Booking_date = Booking_date, @CheckIn_date = CheckIn_date
FROM Booking
IF (@Booking_date > @CheckIn_date)
BEGIN
    PRINT 'The Booking date should be less or equal than Check in date'
    ROLLBACK TRANSACTION
END
RETURN

SELECT * FROM Booking

drop trigger trgInsertDate

--that false
INSERT INTO Booking
VALUES ('B006', getdate(), 'Dina', '2008-04-16')

--that true
INSERT INTO Booking
VALUES ('B006', getdate(), 'Dina', '2008-04-10')

-----
----
--No. 2
--Create trigger prevent delete room
CREATE TRIGGER trgRoomStatus
ON RoomStatus
FOR DELETE
AS
PRINT 'The Data is valid and deletion is not allowed'
ROLLBACK TRANSACTION
RETURN

SELECT * FROM RoomStatus

--check
DELETE FROM RoomStatus
WHERE Room_id = 'R005'

-----
----
--No. 3
--Create trigger to use the room
CREATE TRIGGER trgRoom
ON TransactionDetail
FOR INSERT
AS
DECLARE @Room_status varchar(4)
DECLARE @roomid char(4)
SELECT @Room_status = r.Room_status, @roomid = td.Room_id
FROM RoomStatus r, TransactionDetail td
WHERE r.Room_id = td.Room_id
IF (@Room_status = 'OFF')
BEGIN
    PRINT 'The Room is already in used, please, choose another room '
    ROLLBACK TRANSACTION
END

--to drop trigger
drop trigger trgRoom
```

## Database Hotel

```
-- check with insert
INSERT INTO Customer
VALUES ('0009', 'Angelina Jelvy', 'Jl.
Pelangi', 'F', '(021)-5860237', '19983404', 'KTP')

INSERT INTO Transactions
VALUES ('T009', 'E002', getdate(), '0009', 1250000, 'PM01')

INSERT INTO TransactionDetail
VALUES ('TD009', 'T009', getdate(), '2008/04/18', 'R004', 1250000)

-- =====
==--
----- CURSOR
-----
-- cursor to display the customer who get the souvenir
declare curSouvenir cursor for
select Customer_id, Total_day
from Transactions T, TransactionDetail TD
where T.Transaction_no = TD.Transaction_no
open curSouvenir
declare @Customer_id char(4)
declare @TotalDay int
declare @Description varchar (50)
fetch curSouvenir into @Customer_id, @TotalDay
print'-----'
PRINT 'Customer Id | Total Day | Description'
print'-----'
while (@@fetch_status =0)
begin
    if (@TotalDay >= 3 )
        select @Description = 'Congratulation, you get the Souvenir'
    else
        select @Description = 'Thanks for arrived'
        print @Customer_id + ' ' + convert
        (char(11), @TotalDay) + ' ' + @Description
        fetch curSouvenir into @Customer_id, @TotalDay
end

close curSouvenir

-- to delete
deallocate curSouvenir

select * from customer
select * from Transactions
select * from TransactionDetail
```