UNIVERSITY OF THE PUNJAB
FACULTY OF COMPUTING & INFORMATION TECHNOLOGY
DEPARTMENT OF SOFTWARE ENGINEERING
BSSE Fall 21

Database Systems Project

Project Title

Hotel Management System

Group Members

BSEF21M046 - Muhammad Usman

BSEF21M055 - Muhammad Bilal (Lead Member)

BSEF20A027 - Muhammad Ahmad (Add/Drop)

Instructor

Sir Asif Sohail

Introduction:

In this project we will develop a data base for hotel management system that will manage all the resources efficiently .

It will allocate rooms to customers and also keep track the availability of rooms. System will keep track of all services used by a customer. System can manage the payment system so that user can make payments iteratively. Sytem is also able to make a record of customer history.

Attributes of System:

Relation (BookingNo, Booking_Type, Booking_Date, Checkin_date, Checkout_date, CustomerNo, Customer_Name, Customer_CNIC, Customer_PhoneNo, Customer_Email, Total_Bill, Bill_Status, Remaining_Payment, Payment_ID, Payment_Date, Payment_Amount, Payment_Method, RoomNo, Room_Type, Room_Catagory, Room_Price, Service_ID, Service_Name, Service_Price)

Bottom Up

Normalization:

Now, we try to evaluate each table using normalization:

• FIRST NORMAL FORM(1NF):

For a relation to be in first normal form (1NF), there must not exist any multi-valued attribute within the relation. In the given relation there exist multi value attributes so the relation is not in 1NF. Now we have to remove multi value attributes in another relation.

- Booking (**<u>BookingNo</u>**, Booking_Type, Booking_Date, Checkin_date, Checkout_date, CustomerNo, Customer_Name, Customer_CNIC, Customer_PhoneNo, Customer_Email, Total_Bill, Bill_Status, Remaining_Payment)

- Room (**<u>BookingNo, RoomNo</u>**, Room_Type, Room_Catagory, Room_Price)
  BookingNo is FK that refers to Booking relation

- Service (**<u>BookingNo, Service_ID</u>**, Service_Name, Service_Price)
  BookingNo is FK that refers to Booking relation

- Payment (**<u>BookingNo, Payment_ID</u>**, Payment_Date, Payment_Amount, Payment_Method)
  BookingNo is FK that refers to Booking relation

As in the above tables, we can see that there doesn't exist any multi-valued attribute so we can declare this relation to be in first normal form.

• SECOND NORMAL FORM(2NF):

A table is in 2nd Normal Form (2NF) if it is in 1st Normal Form (1NF) and there must not be any partial dependencies, meaning that for each non-key attribute, its value must be dependent only on the entire candidate key and not on a part of it.

- Booking (**<u>BookingNo</u>**, Booking_Type, Booking_Date, Checkin_date, Checkout_date, CustomerNo, Customer_Name, Customer_CNIC, Customer_PhoneNo, Customer_Email, Total_Bill, Bill_Status, Remaining_Payment)

- Booked_Room (**<u>BookingNo, RoomNo</u>**)

  BookingNo and RoomNo are both FKs that refer to Booking and Room relation respectively

- Room (**<u>RoomNo</u>**, Room_Type, Room_Catagory, Room_Price)

- Booked_Service (**<u>BookingNo, Service-ID</u>**)

  BookingNo and Service_ID are both FKs that refer to Booking and Service relation respectively

- Service (**<u>Service_ID</u>**, Service_Name, Service_Price)

- Payment (**<u>BookingNo, Payment_ID</u>**, Payment_Date, Payment_Amount, Payment_Method)

  BookingNo is FK that refers to Booking relation

• THIRD NORMAL FORM(3NF):

For a table to be in third normal form, it must be in 2nd normal form and there must not exist any transitive dependency in the relation which states that:

Non-prime attributes → non-prime attributes

- Booking (**<u>BookingNo</u>**, Booking_Type, Booking_Date, Checkin_date, Checkout_date, CustomerNo, Total_Bill, Bill_Status, Remaining_Payment)

  CustomerNo is FK that refers to Customer relation

- Customer (**<u>CustomerNo</u>**, Customer_Name, Customer_CNIC, Customer_PhoneNo, Customer_Email)

- Booked_Room (**<u>BookingNo, RoomNo</u>**)

  BookingNo and RoomNo are both FKs that refer to Booking and Room relation respectively

- Room (**<u>RoomNo</u>**, Room_Type, Room_Catagory, Room_Price)

- Booked_Service (**<u>BookingNo, Service-ID</u>**)

  BookingNo and Service_ID are both FKs that refer to Booking and Service relation respectively

- Service (**<u>Service_ID</u>**, Service_Name, Service_Price)

- Payment (**<u>BookingNo, Payment_ID</u>**, Payment_Date, Payment_Amount, Payment_Method)

  BookingNo is FK that refers to Booking relation

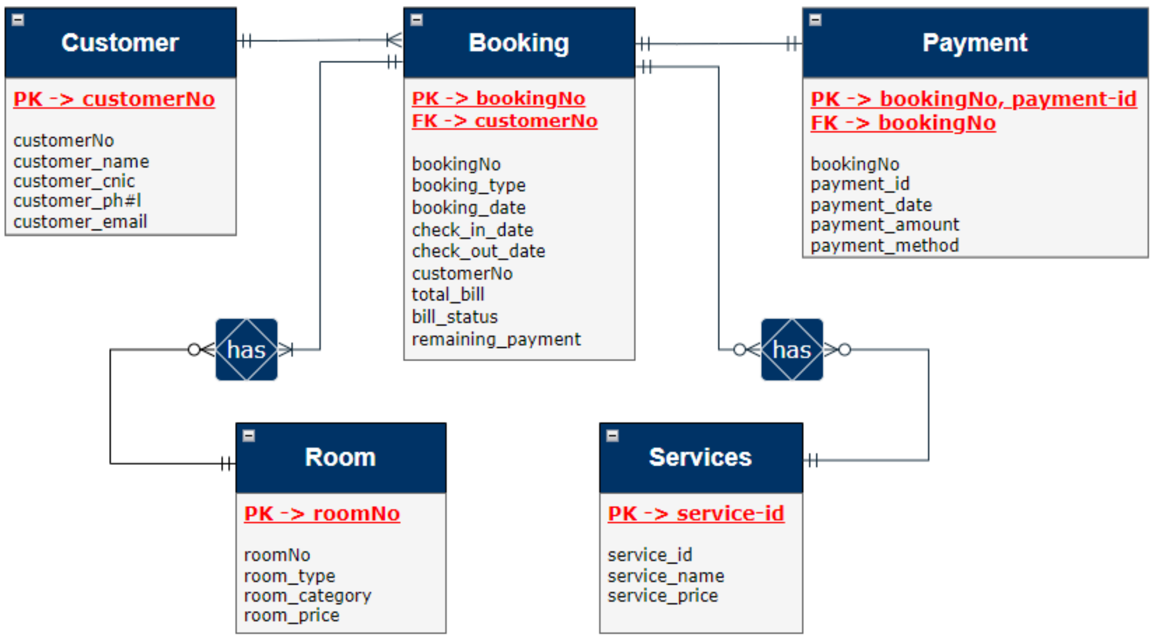
All relational schemas within the system witness that there doesn't exist any such relation within this system, so all the tables are already in 3NF.

Top Down

Entity Relation Diagram of System:

The following Entity-Relationship Diagram (ERD) represents the relationships of different entities in this database system with respect to their attributes and primary as well as foreign keys. It also describes this system's 1:1, 1:M, and M: N relationships.

ERD DIAGRAM



Connectivity Table

Entity	**Relationship**	**Connectivity**	**Entity**
Customer	Has	1:M	Booking
Payment	BelongsTo	M:1	Booking
Room	BelongsTo	M:N	Booking
Services	BelongsTo	M:N	Booking

Relational Schema

1. Booking (**BookingNo**, Booking_Type, Booking_Date, Checkin_date, Checkout_date, **CustomerNo,** Total_Bill, Bill_Status, Remaining_Payment)
   CustomerNo is FK that refers to Customer relation

2. Customer (**CustomerNo,** Customer_Name, Customer_CNIC, Customer_PhoneNo, Customer_Email)

3. Booked_Room (**BookingNo, RoomNo**)
   BookingNo and RoomNo are both FKs that refer to Booking and Room relation respectively

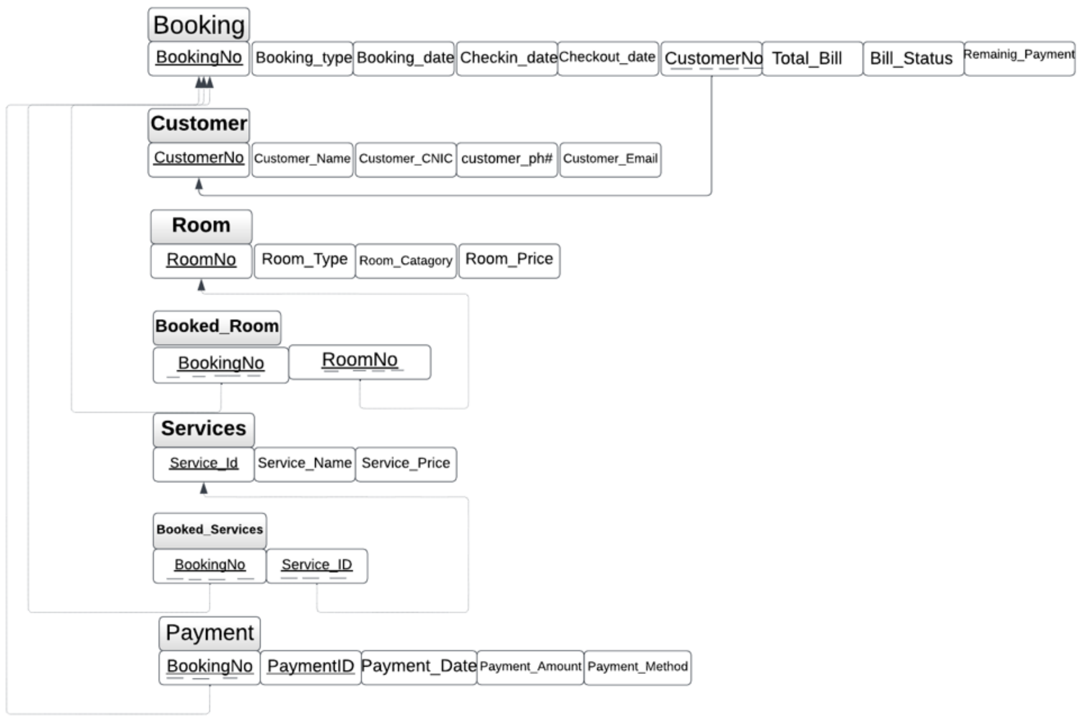
4. Room (**RoomNo**, Room_Type, Room_Catagory, Room_Price)

5. Booked_Service (**BookingNo, Service-ID**)
   BookingNo and Service_ID are both FKs that refer to Booking and Service relation respectively

6. Service (**Service_ID**, Service_Name, Service_Price)

7. Payment (**BookingNo, Payment_ID**, Payment_Date, Payment_Amount, Payment_Method)
   BookingNo is FK that refers to Booking relation

Relational Data Model (Dependency Diagram)



Relations Description

Table Name: Customer

Attribute	Data Type	Size	Constraints
CustomerNo	Number	4	Primary Key
Customer_Name	Varchar2	25	Not Null
Customer_CNIC	Number	13	
Customer_PhoneNo	Number	11	
Customer_Email	Varchar2	50	

Table Name: Booking

Attribute	Data Type	Size	Constraints
BookingNo	Number	4	Primary Key
Booking_Type	Varchar2	10	Can be ‘Individual’ or ‘Group’
Booking_Date	Date		Default Sysdate
Checkin_Date	Date	11	
Checkout_Date	Date	50	
CustomerNo	Number	4	Foreign Key
Total_Bill	Number	6,2	
Bill_Status	Varchar2	7	Can be ‘Paid’ or ‘Pending’
Remaining_Payment	Number	6,2	

Table Name: Room

Attribute	Data Type	Size	Constraints
RoomNo	Number	3	Primary Key
Room_Type	Char	6	Can be in ‘Single’ or ‘Double’
Room_Category	Varchar2	8	Can be in ‘Luxury’ or ‘Standard’

Room_Price	Number	5	

Table Name: Booked_Room

Attribute	Data Type	Size	Constraints
BookingNo	Number	4	Primary Key, Foreign key
RoomNo	Number	3	Primary Key, Foreign key

Table Name: Service

Attribute	Data Type	Size	Constraints
Service_ID	Number	4	Primary Key
Service_Name	Varchar2	50	Can be in ‘Breakfast’, ‘Dinner’, ‘Lunch’, ‘House Keeping’, ‘Snacks’, ‘Laundary’ or ’Wifi’
Service_Price	Number	5	

Table Name: Booked_Service

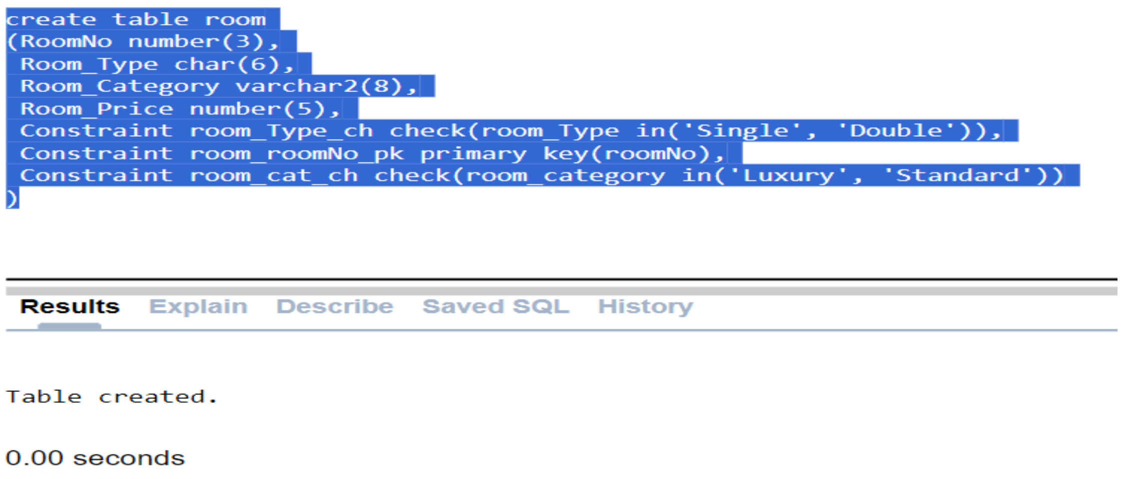
Attribute	Data Type	Size	Constraints

BookingNo	Number	4	Primary Key, Foreign key
Service_ID	Number	3	Primary Key, Foreign key

Table Name: Payment

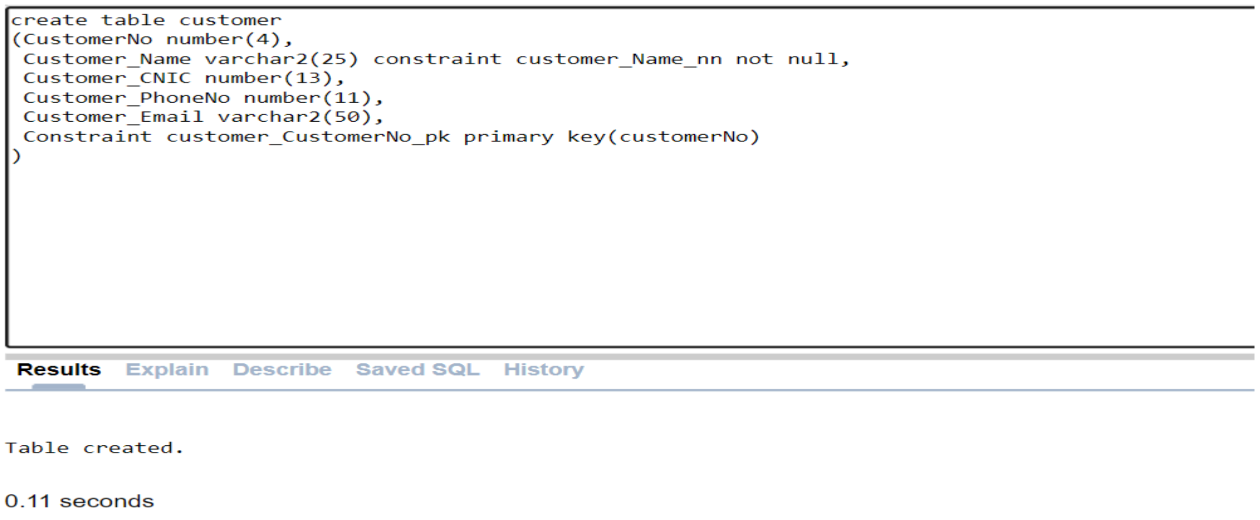
Attribute	Data Type	Size	Constraints
BookingNo	Number	4	Primary Key, Foreign key
Payment_ID	Number	1	Primary Key
Payment_Date	Date		Default Sysdate
Payement_Amount	Number	6	
Payment_Method	Varchar2	10	Can be in ‘Cash’, ‘Online Transaction’ or ‘Debit Card’

SQL Statements for Table Creation



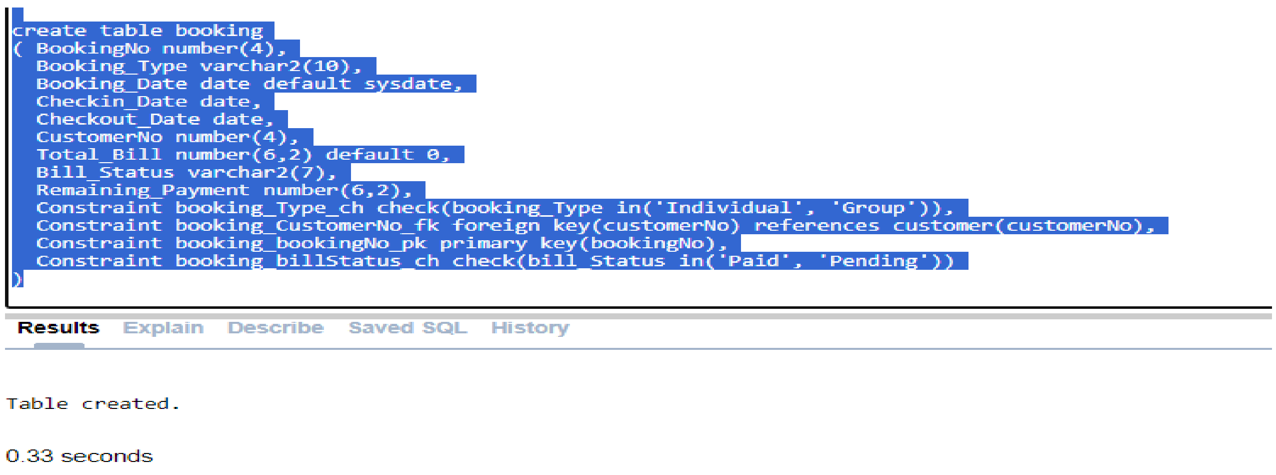
```
create table room
(RoomNo number(3),
 Room_Type char(6),
 Room_Category varchar2(8),
 Room_Price number(5),
 Constraint room_Type_ch check(room_Type in('Single', 'Double')),
 Constraint room_roomNo_pk primary key(roomNo),
 Constraint room_cat_ch check(room_category in('Luxury', 'Standard'))
)
```

```
Table created.

0.00 seconds
```



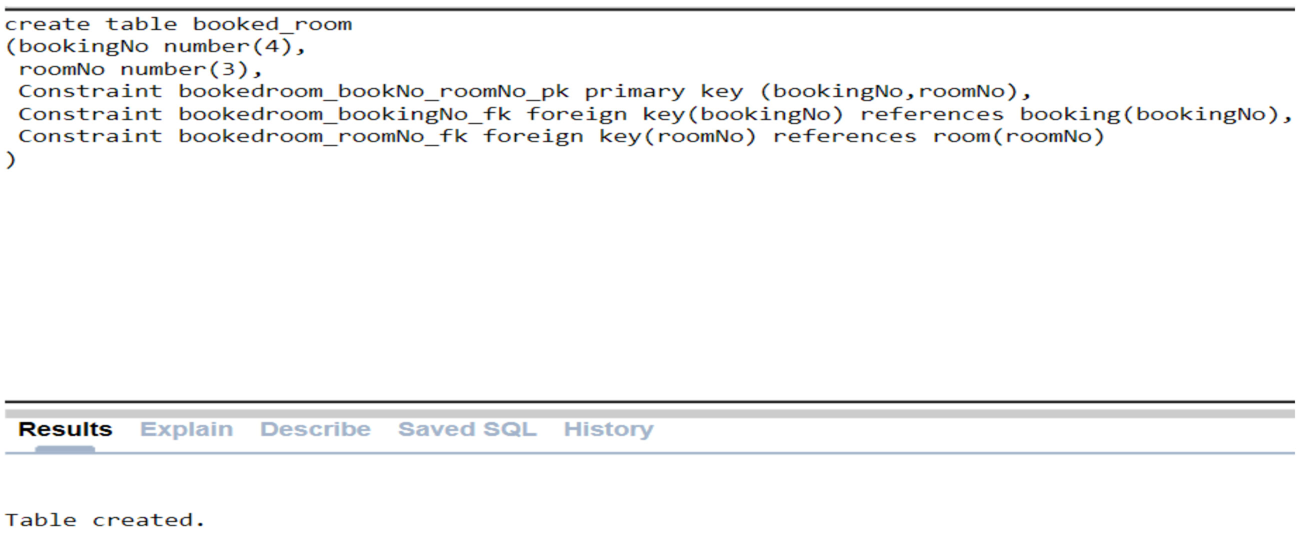
```
create table customer
(CustomerNo number(4),
 Customer_Name varchar2(25) constraint customer_Name_nn not null,
 Customer_CNIC number(13),
 Customer_PhoneNo number(11),
 Customer_Email varchar2(50),
 Constraint customer_CustomerNo_pk primary key(customerNo)
)
```

```
Table created.

0.11 seconds
```



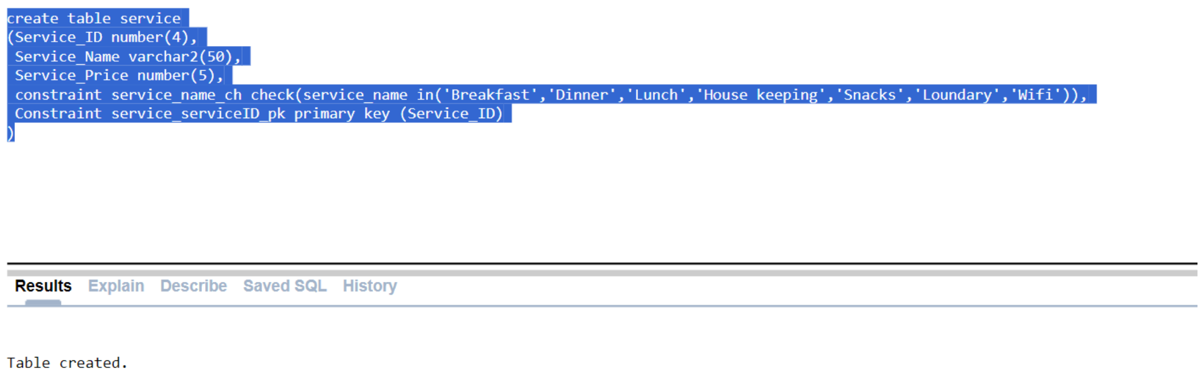
```
create table booking
( BookingNo number(4),
  Booking_Type varchar2(10),
  Booking_Date date default sysdate,
  Checkin_Date date,
  Checkout_Date date,
  CustomerNo number(4),
  Total_Bill number(6,2) default 0,
  Bill_Status varchar2(7),
  Remaining_Payment number(6,2),
  Constraint booking_Type_ch check(booking_Type in('Individual', 'Group')),
  Constraint booking_CustomerNo_fk foreign key(customerNo) references customer(customerNo),
  Constraint booking_bookingNo_pk primary key(bookingNo),
  Constraint booking_billStatus_ch check(bill_Status in('Paid', 'Pending'))
)
```

```
Table created.

0.33 seconds
```



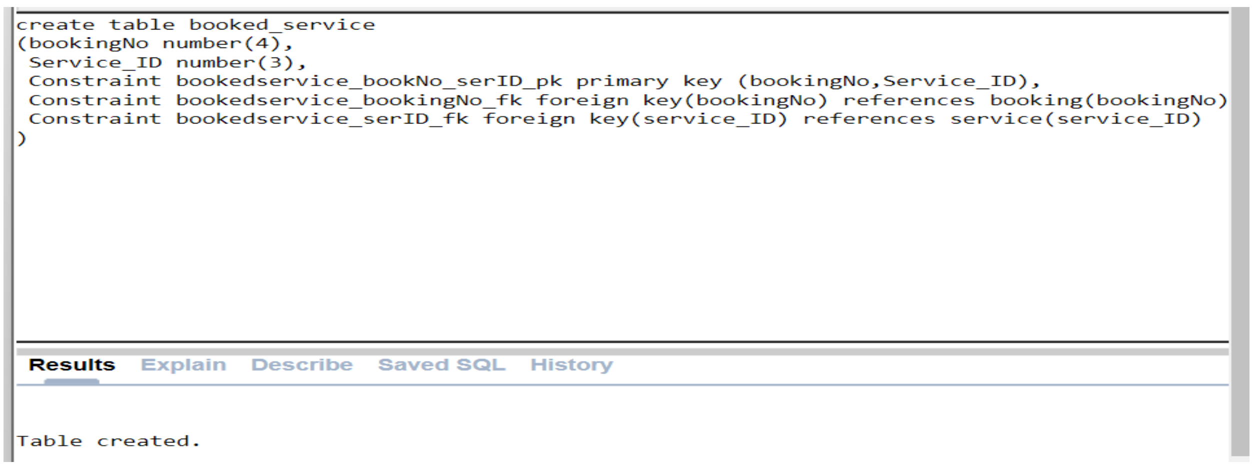
```
create table booked_room
(bookingNo number(4),
 roomNo number(3),
 Constraint bookedroom_bookNo_roomNo_pk primary key (bookingNo,roomNo),
 Constraint bookedroom_bookingNo_fk foreign key(bookingNo) references booking(bookingNo),
 Constraint bookedroom_roomNo_fk foreign key(roomNo) references room(roomNo)
)
```

```
Table created.
```



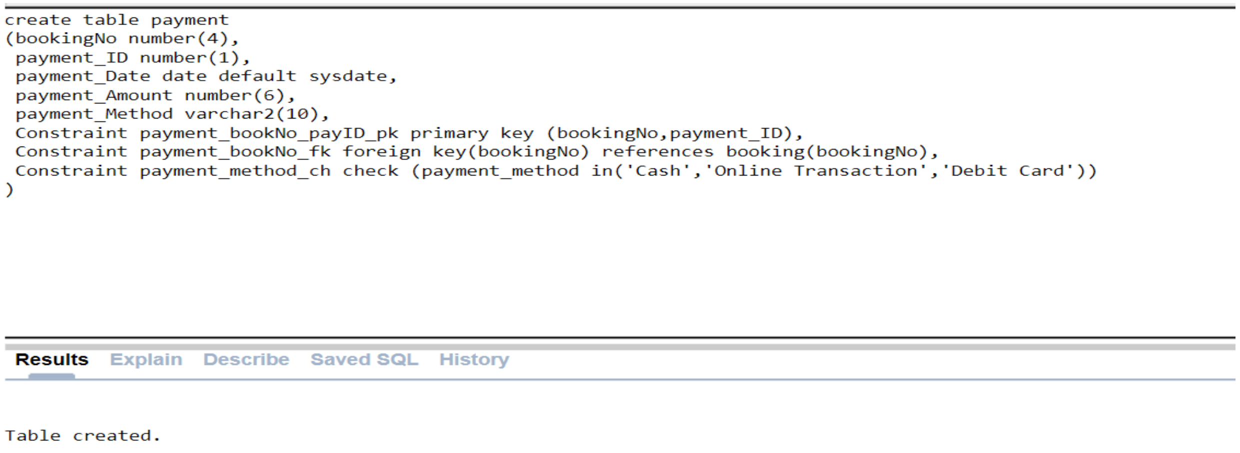
```
create table service
(Service_ID number(4),
 Service_Name varchar2(50),
 Service_Price number(5),
 constraint service_name_ch check(service_name in('Breakfast','Dinner','Lunch','House keeping','Snacks','Loundary','Wifi')),
 Constraint service_serviceID_pk primary key (Service_ID)
)
```

```
Table created.
```



```
create table booked_service
(bookingNo number(4),
 Service_ID number(3),
 Constraint bookedservice_bookNo_serID_pk primary key (bookingNo,Service_ID),
 Constraint bookedservice_bookingNo_fk foreign key(bookingNo) references booking(bookingNo)
 Constraint bookedservice_serID_fk foreign key(service_ID) references service(service_ID)
)
```

```
Table created.
```

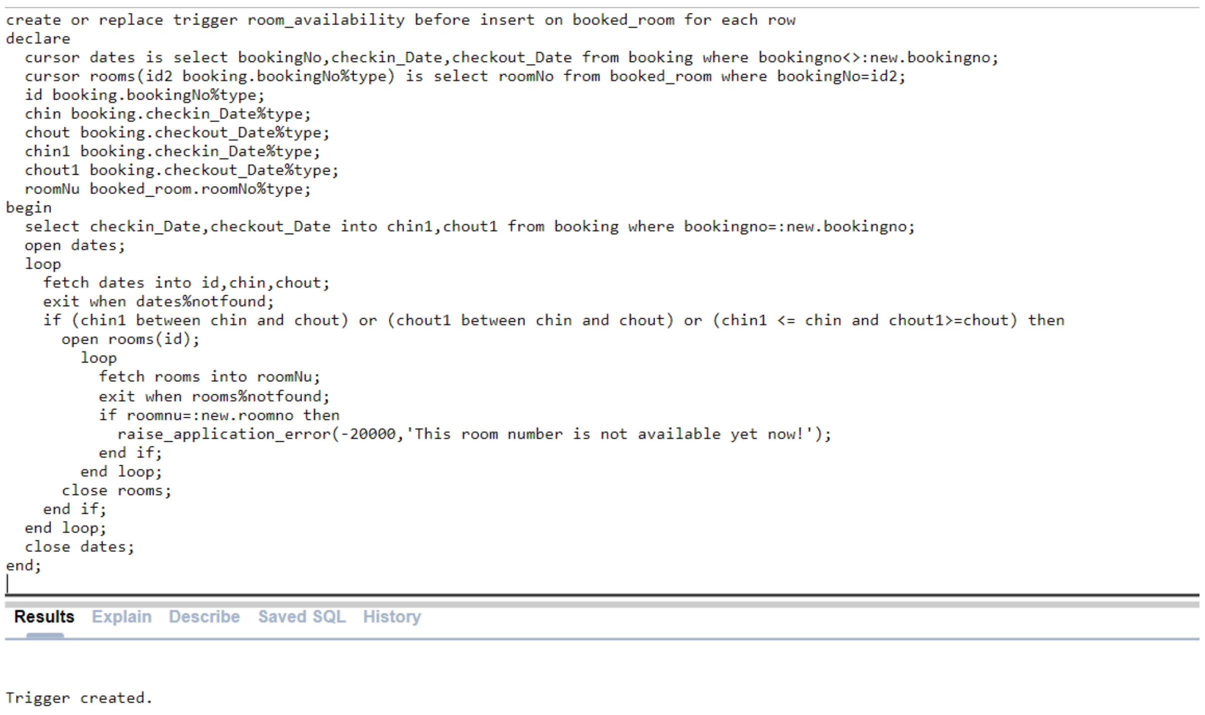


```
create table payment
(bookingNo number(4),
 payment_ID number(1),
 payment_Date date default sysdate,
 payment_Amount number(6),
 payment_Method varchar2(10),
 Constraint payment_bookNo_payID_pk primary key (bookingNo,payment_ID),
 Constraint payment_bookNo_fk foreign key(bookingNo) references booking(bookingNo),
 Constraint payment_method_ch check (payment_method in('Cash','Online Transaction','Debit Card'))
)
```

```
Table created.
```

Triggers

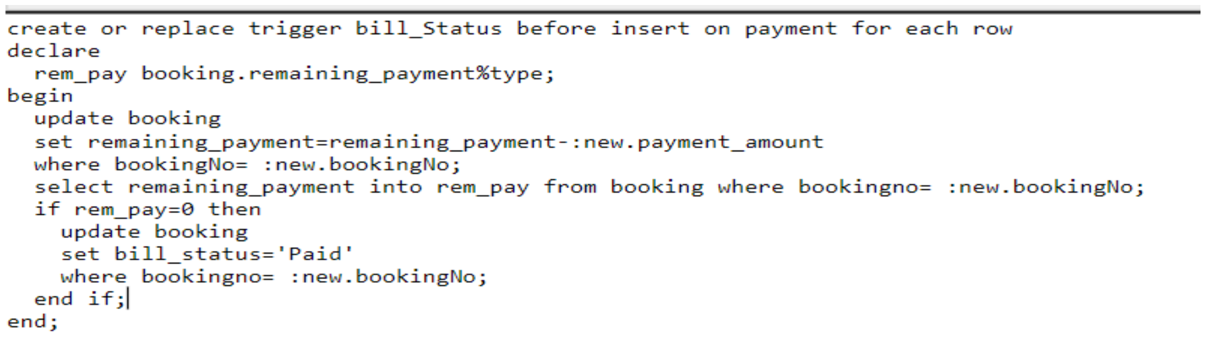
1. Trigger that will restrict to allocate an already booked room



```
create or replace trigger room_availability before insert on booked_room for each row
declare
  cursor dates is select bookingNo,checkin_Date,checkout_Date from booking where bookingno<>:new.bookingno;
  cursor rooms(id2 booking.bookingNo%type) is select roomNo from booked_room where bookingNo=id2;
  id booking.bookingNo%type;
  chin booking.checkin_Date%type;
  chout booking.checkout_Date%type;
  chin1 booking.checkin_Date%type;
  chout1 booking.checkout_Date%type;
  roomNu booked_room.roomNo%type;
begin
  select checkin_Date,checkout_Date into chin1,chout1 from booking where bookingno=:new.bookingno;
  open dates;
  loop
    fetch dates into id,chin,chout;
    exit when dates%notfound;
    if (chin1 between chin and chout) or (chout1 between chin and chout) or (chin1 <= chin and chout1>=chout) then
      open rooms(id);
        loop
          fetch rooms into roomNu;
          exit when rooms%notfound;
          if roomnu=:new.roomno then
            raise_application_error(-20000,'This room number is not available yet now!');
          end if;
        end loop;
      close rooms;
    end if;
  end loop;
  close dates;
end;
```

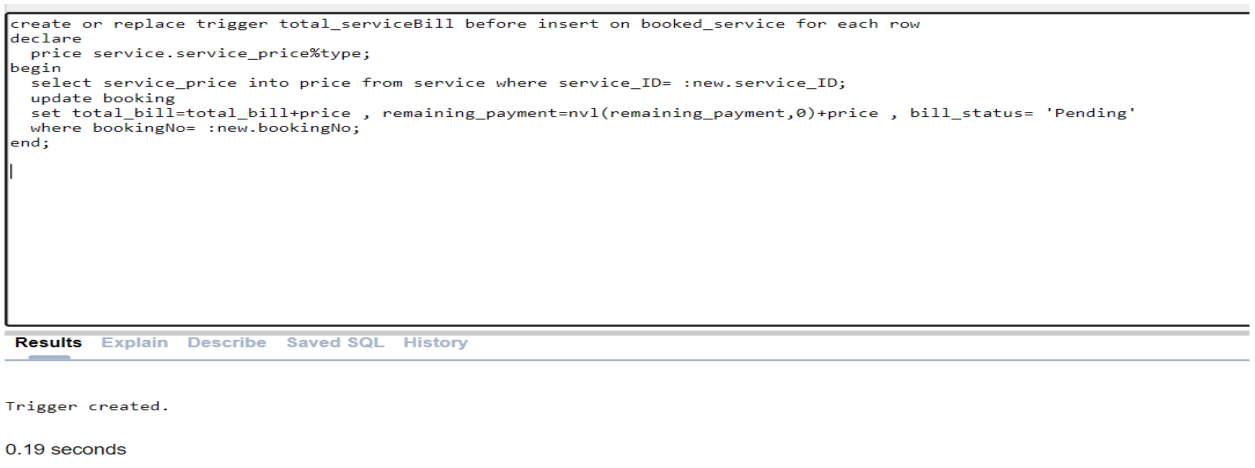
```
Trigger created.
```

2. It will update the status of payment that means payment is pending or paid



```
create or replace trigger bill_Status before insert on payment for each row
declare
  rem_pay booking.remaining_payment%type;
begin
  update booking
  set remaining_payment=remaining_payment-:new.payment_amount
  where bookingNo= :new.bookingNo;
  select remaining_payment into rem_pay from booking where bookingno= :new.bookingNo;
  if rem_pay=0 then
    update booking
    set bill_status='Paid'
    where bookingno= :new.bookingNo;
  end if;
end;
```

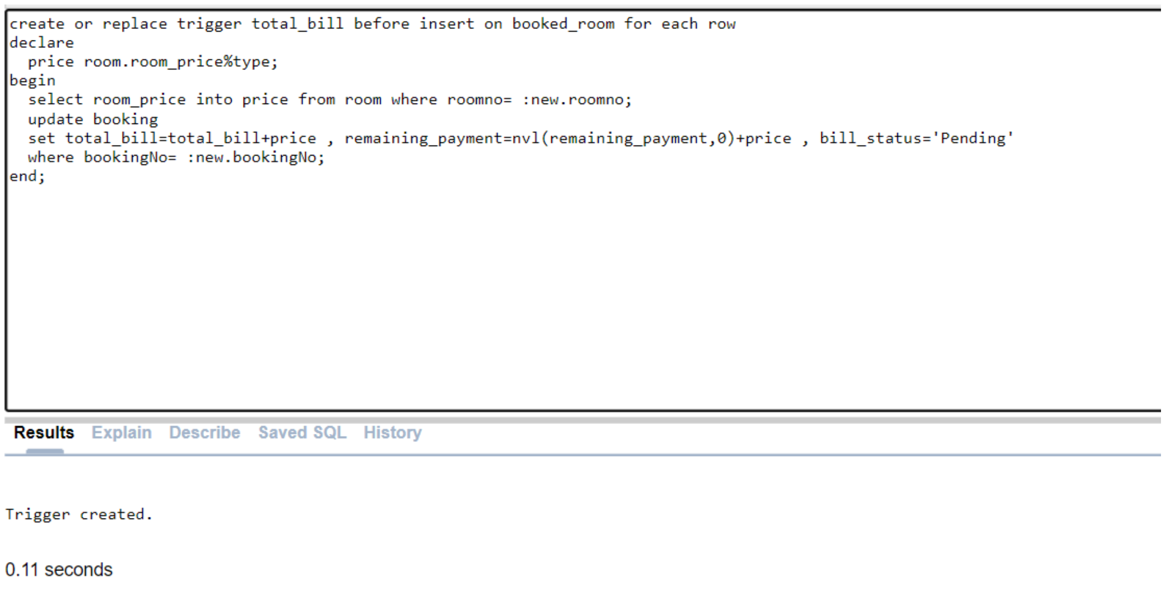
3. It will add total amount of services to total bill used by customer during his/her booking period



```
create or replace trigger total_serviceBill before insert on booked_service for each row
declare
  price service.service_price%type;
begin
  select service_price into price from service where service_ID= :new.service_ID;
  update booking
  set total_bill=total_bill+price , remaining_payment=nvl(remaining_payment,0)+price , bill_status= 'Pending'
  where bookingNo= :new.bookingNo;
end;
```

```
Trigger created.

0.19 seconds
```

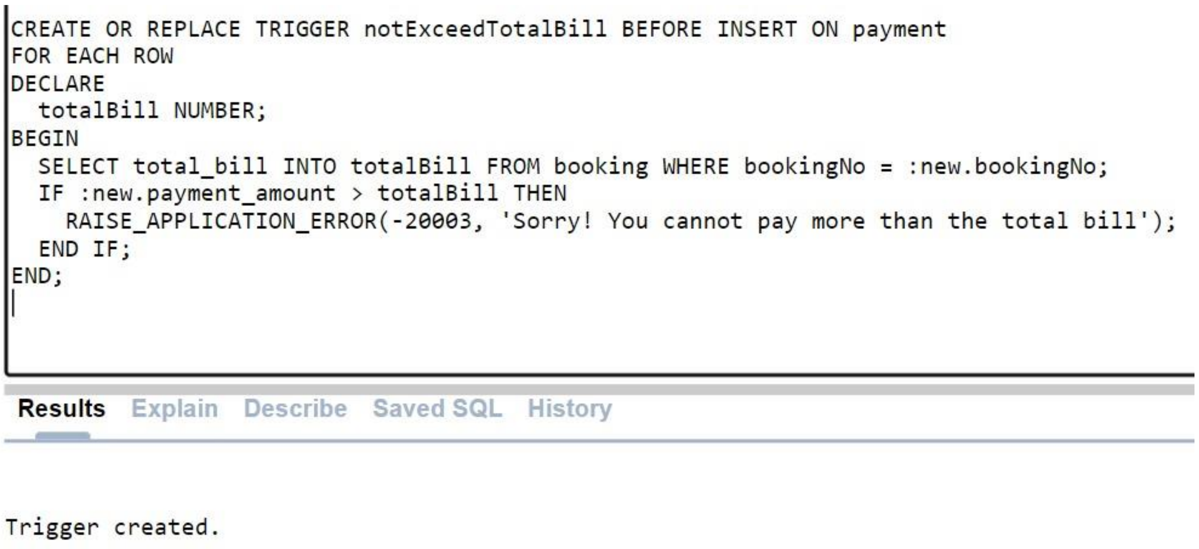
4. It will add prices of all rooms booked by a customer to total bill



```
create or replace trigger total_bill before insert on booked_room for each row
declare
  price room.room_price%type;
begin
  select room_price into price from room where roomno= :new.roomno;
  update booking
  set total_bill=total_bill+price , remaining_payment=nvl(remaining_payment,0)+price , bill_status='Pending'
  where bookingNo= :new.bookingNo;
end;
```

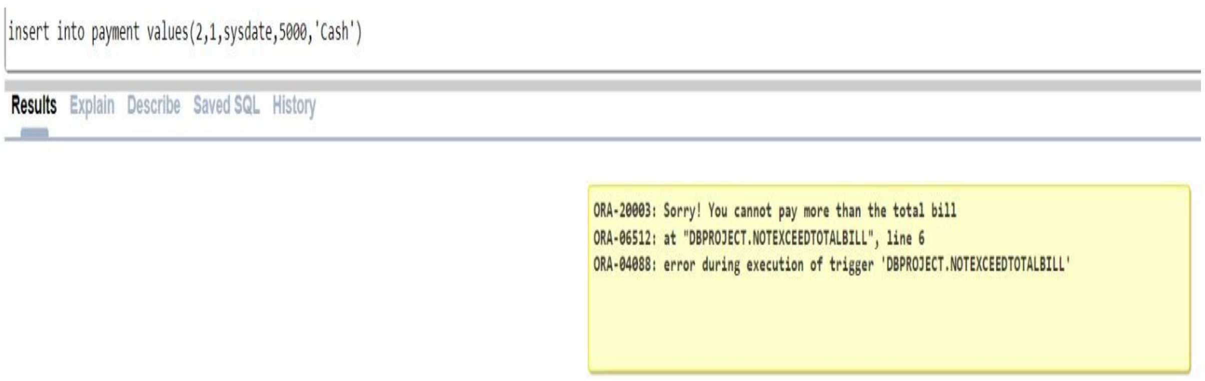
```
Trigger created.

0.11 seconds
```

5. It will restrict user to pay more amount than actual payment



```
CREATE OR REPLACE TRIGGER notExceedTotalBill BEFORE INSERT ON payment
FOR EACH ROW
DECLARE
  totalBill NUMBER;
BEGIN
  SELECT total_bill INTO totalBill FROM booking WHERE bookingNo = :new.bookingNo;
  IF :new.payment_amount > totalBill THEN
    RAISE_APPLICATION_ERROR(-20003, 'Sorry! You cannot pay more than the total bill');
  END IF;
END;
```

```
Trigger created.
```

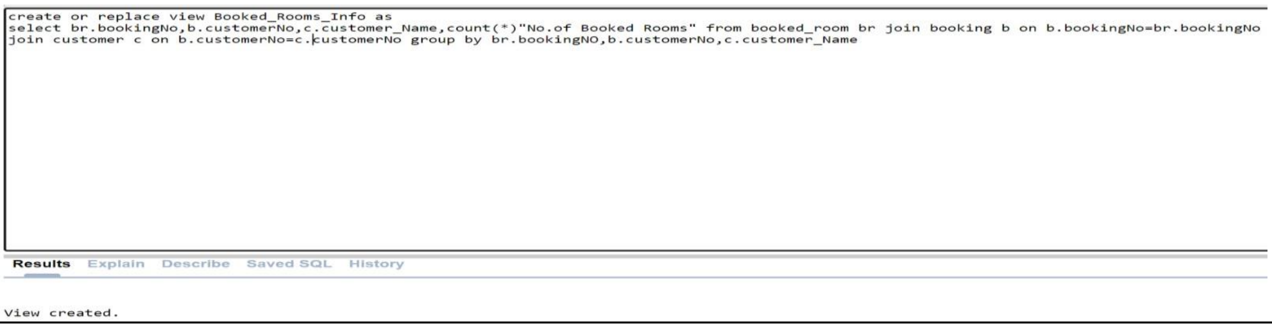


```
insert into payment values(2,1,sysdate,5000,'Cash')
```

```
ORA-20003: Sorry! You cannot pay more than the total bill
ORA-06512: at "DBPROJECT.NOTEXCEEDTOTALBILL", line 6
ORA-04088: error during execution of trigger 'DBPROJECT.NOTEXCEEDTOTALBILL'
```

Views

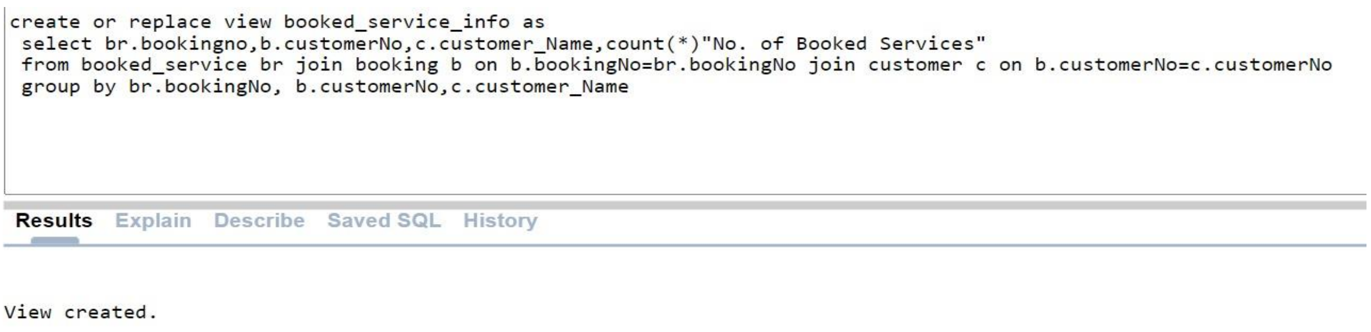
1. It will display the information of total number of booked rooms in a booking



```
create or replace view Booked_Rooms_Info as
select br.bookingNo,b.customerNo,c.customer_Name,count(*)"No.of Booked Rooms" from booked_room br join booking b on b.bookingNo=br.bookingNo
join customer c on b.customerNo=c.customerNo group by br.bookingNO,b.customerNo,c.customer_Name
```

```
View created.
```

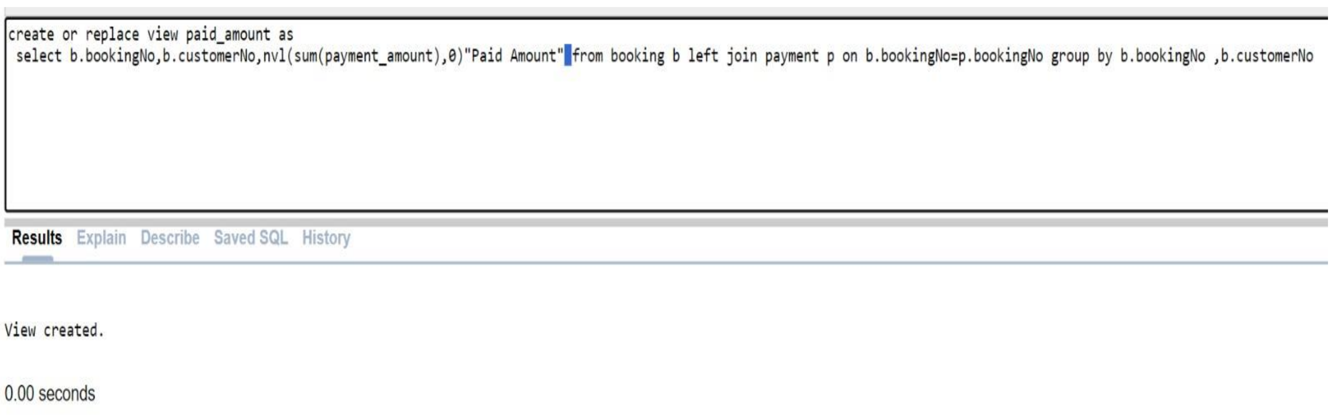
2. It will display the information of total number of booked services in a booking.



```
create or replace view booked_service_info as
 select br.bookingno,b.customerNo,c.customer_Name,count(*)"No. of Booked Services"
 from booked_service br join booking b on b.bookingNo=br.bookingNo join customer c on b.customerNo=c.customerNo
 group by br.bookingNo, b.customerNo,c.customer_Name
```

```
View created.
```

3. It will show paid amount of every booking

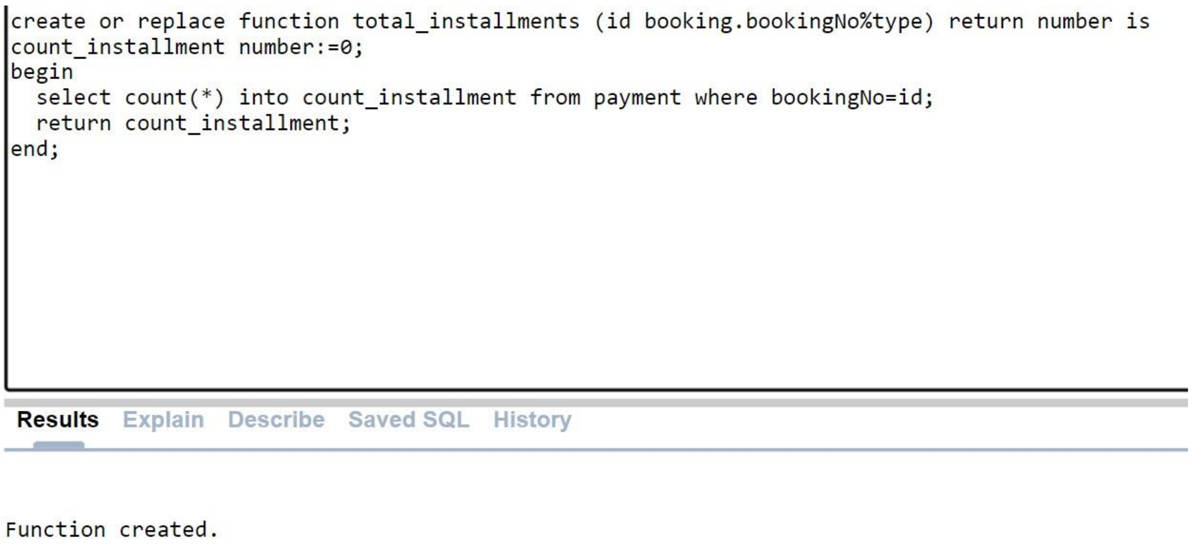


```
create or replace view paid_amount as
 select b.bookingNo,b.customerNo,nvl(sum(payment_amount),0)"Paid Amount"from booking b left join payment p on b.bookingNo=p.bookingNo group by b.bookingNo ,b.customerNo
```

```
View created.

0.00 seconds
```

Functions

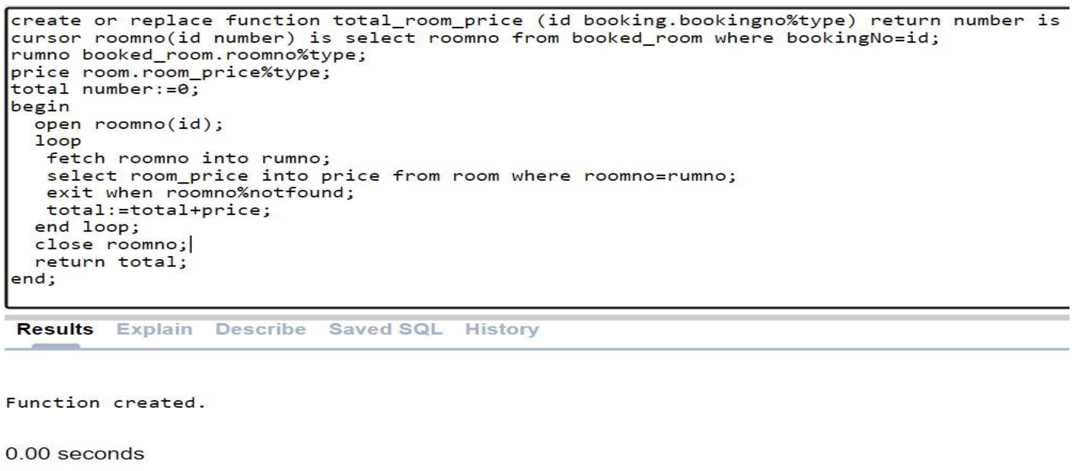
1. It will return total number of instalments made by customer



```
create or replace function total_installments (id booking.bookingNo%type) return number is
count_installment number:=0;
begin
  select count(*) into count_installment from payment where bookingNo=id;
  return count_installment;
end;
```

```
Function created.
```

2. it will display overall bill of all rooms that will be booked in a booking

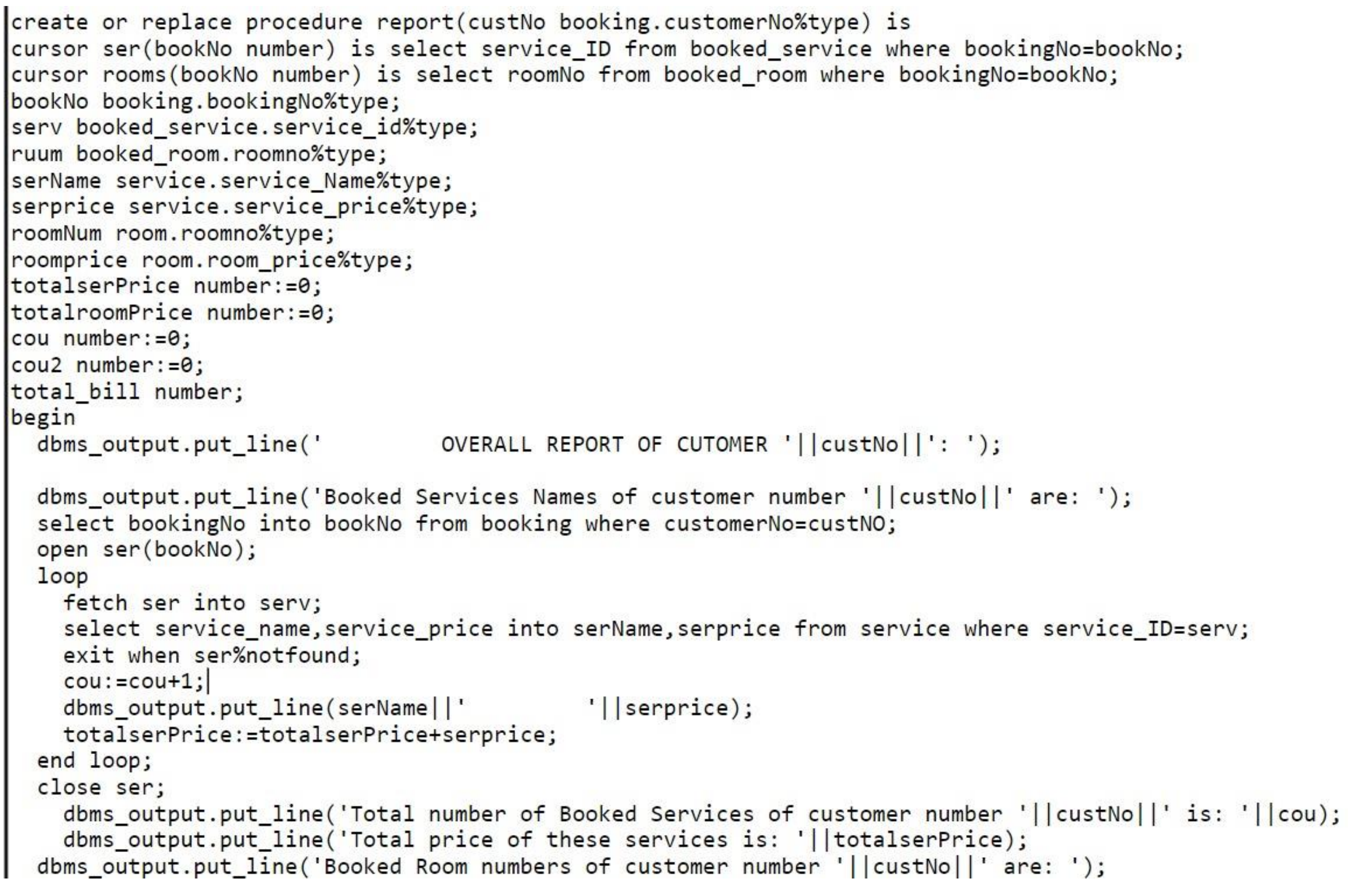


```
create or replace function total_room_price (id booking.bookingno%type) return number is
cursor roomno(id number) is select roomno from booked_room where bookingNo=id;
rumno booked_room.roomno%type;
price room.room_price%type;
total number:=0;
begin
  open roomno(id);
  loop
   fetch roomno into rumno;
   select room_price into price from room where roomno=rumno;
   exit when roomno%notfound;
   total:=total+price;
  end loop;
  close roomno;
  return total;
end;
```

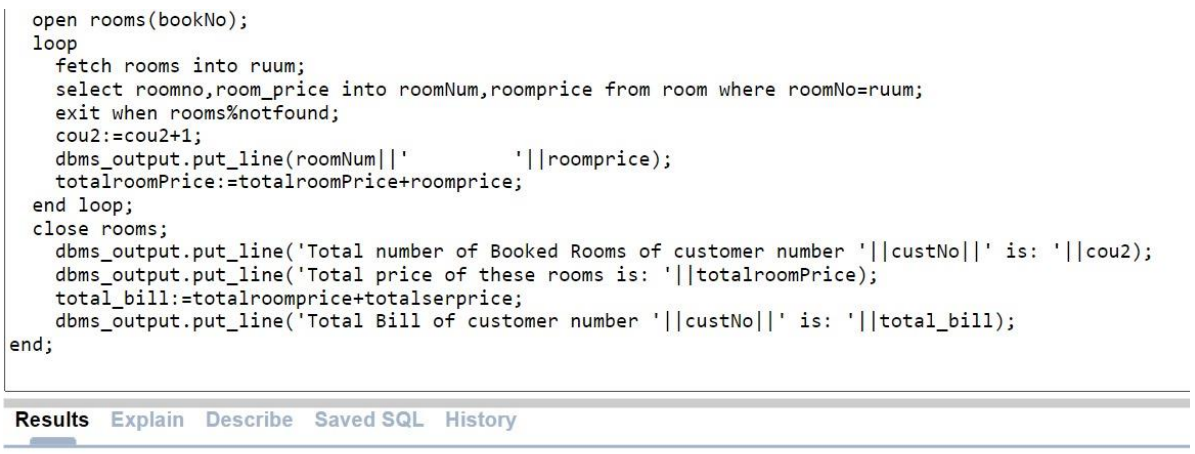
```
Function created.

0.00 seconds
```

Procedure

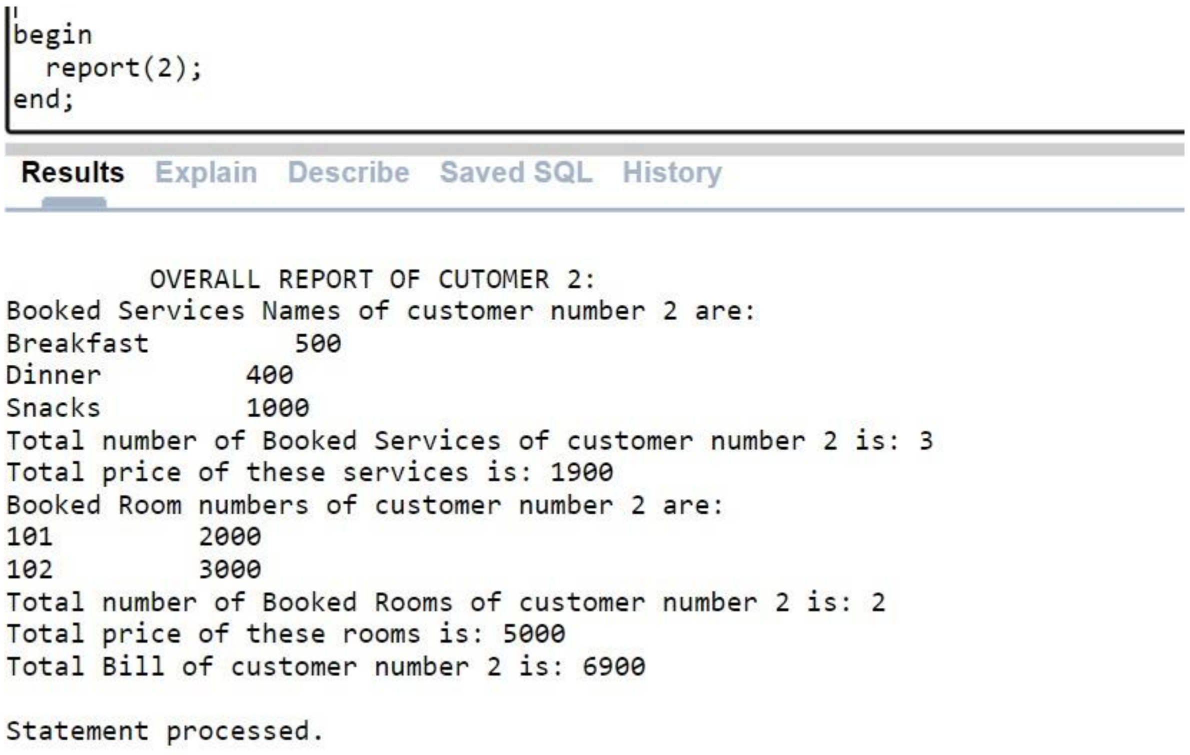
1. It will display overall report of a customer



```
create or replace procedure report(custNo booking.customerNo%type) is
cursor ser(bookNo number) is select service_ID from booked_service where bookingNo=bookNo;
cursor rooms(bookNo number) is select roomNo from booked_room where bookingNo=bookNo;
bookNo booking.bookingNo%type;
serv booked_service.service_id%type;
ruum booked_room.roomno%type;
serName service.service_Name%type;
serprice service.service_price%type;
roomNum room.roomno%type;
roomprice room.room_price%type;
totalserPrice number:=0;
totalroomPrice number:=0;
cou number:=0;
cou2 number:=0;
total_bill number;
begin
  dbms_output.put_line('         OVERALL REPORT OF CUTOMER '||custNo||': ');

  dbms_output.put_line('Booked Services Names of customer number '||custNo||' are: ');
  select bookingNo into bookNo from booking where customerNo=custNO;
  open ser(bookNo);
  loop
    fetch ser into serv;
    select service_name,service_price into serName,serprice from service where service_ID=serv;
    exit when ser%notfound;
    cou:=cou+1;
    dbms_output.put_line(serName||'         '||serprice);
    totalserPrice:=totalserPrice+serprice;
  end loop;
  close ser;
    dbms_output.put_line('Total number of Booked Services of customer number '||custNo||' is: '||cou);
    dbms_output.put_line('Total price of these services is: '||totalserPrice);
  dbms_output.put_line('Booked Room numbers of customer number '||custNo||' are: ');
```



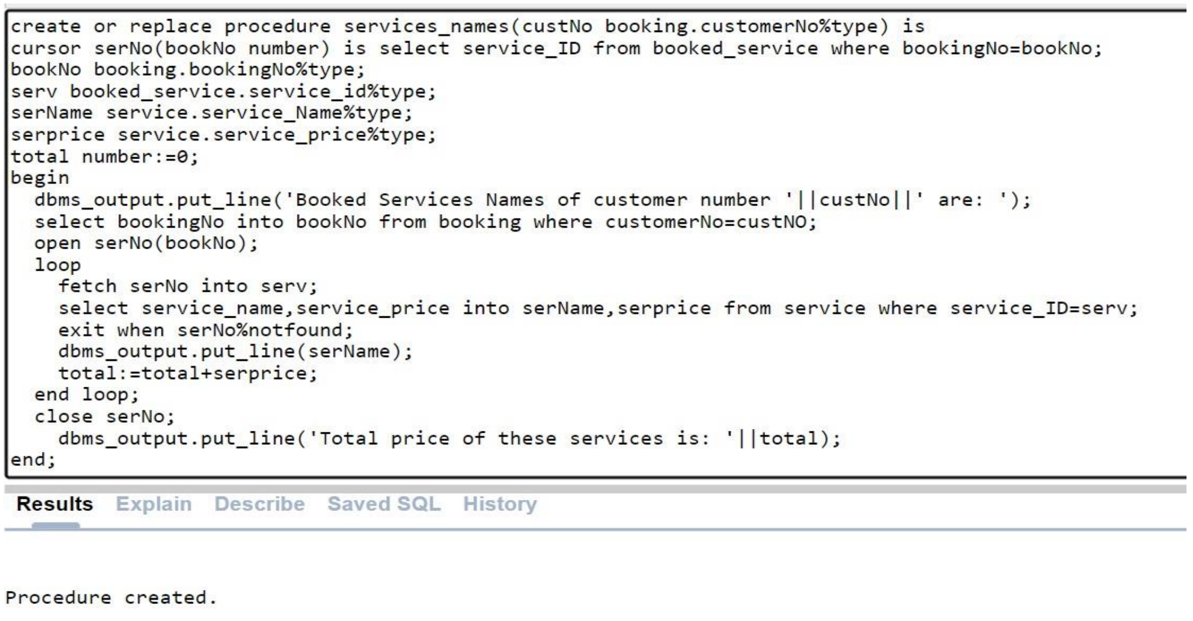
```
  open rooms(bookNo);
  loop
    fetch rooms into ruum;
    select roomno,room_price into roomNum,roomprice from room where roomNo=ruum;
    exit when rooms%notfound;
    cou2:=cou2+1;
    dbms_output.put_line(roomNum||'          '||roomprice);
    totalroomPrice:=totalroomPrice+roomprice;
  end loop;
  close rooms;
    dbms_output.put_line('Total number of Booked Rooms of customer number '||custNo||' is: '||cou2);
    dbms_output.put_line('Total price of these rooms is: '||totalroomPrice);
    total_bill:=totalroomprice+totalserprice;
    dbms_output.put_line('Total Bill of customer number '||custNo||' is: '||total_bill);
end;
```



```
begin
  report(2);
end;
```

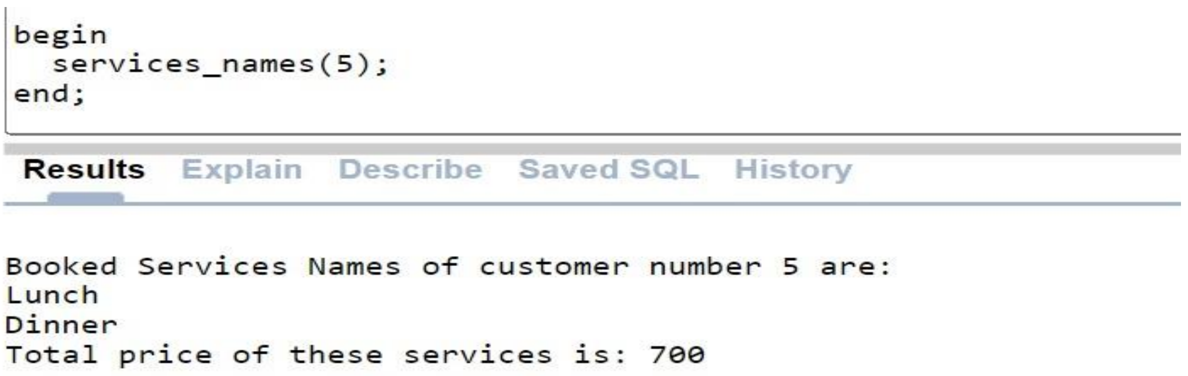
```
         OVERALL REPORT OF CUTOMER 2:
Booked Services Names of customer number 2 are:
Breakfast         500
Dinner         400
Snacks         1000
Total number of Booked Services of customer number 2 is: 3
Total price of these services is: 1900
Booked Room numbers of customer number 2 are:
101          2000
102          3000
Total number of Booked Rooms of customer number 2 is: 2
Total price of these rooms is: 5000
Total Bill of customer number 2 is: 6900

Statement processed.
```

2_It will display all services booked by a customer



```
create or replace procedure services_names(custNo booking.customerNo%type) is
cursor serNo(bookNo number) is select service_ID from booked_service where bookingNo=bookNo;
bookNo booking.bookingNo%type;
serv booked_service.service_id%type;
serName service.service_Name%type;
serprice service.service_price%type;
total number:=0;
begin
  dbms_output.put_line('Booked Services Names of customer number '||custNo||' are: ');
  select bookingNo into bookNo from booking where customerNo=custNO;
  open serNo(bookNo);
  loop
    fetch serNo into serv;
    select service_name,service_price into serName,serprice from service where service_ID=serv;
    exit when serNo%notfound;
    dbms_output.put_line(serName);
    total:=total+serprice;
  end loop;
  close serNo;
    dbms_output.put_line('Total price of these services is: '||total);
end;
```

```
Procedure created.
```

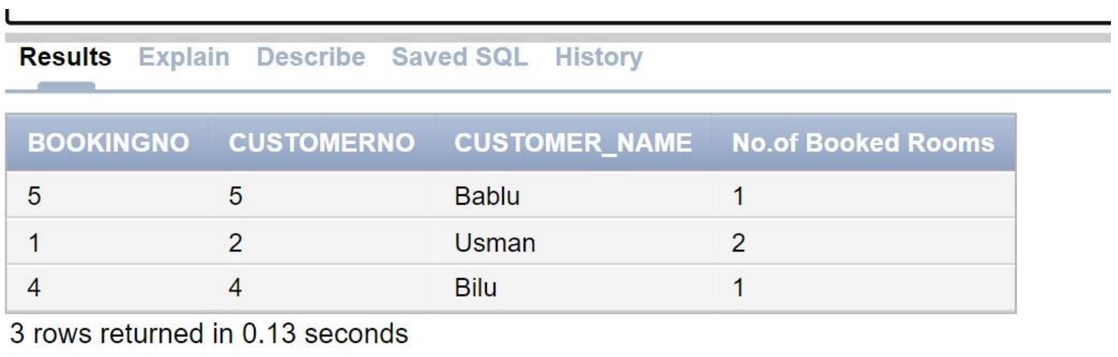


```
begin
  services_names(5);
end;
```

```
Booked Services Names of customer number 5 are:
Lunch
Dinner
Total price of these services is: 700
```

Important Select Statements

1) **Select * from booked_room_info**
   it will display the all rooms booked by a customer

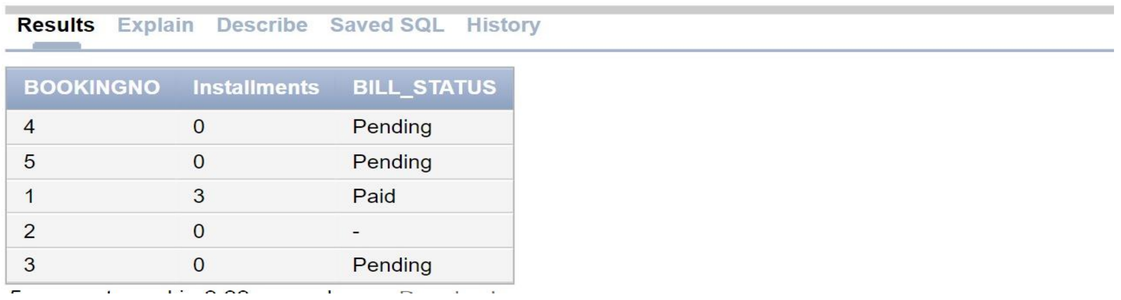


BOOKINGNO	CUSTOMERNO	CUSTOMER_NAME	No.of Booked Rooms
5	5	Bablu	1
1	2	Usman	2
4	4	Bilu	1

3 rows returned in 0.13 seconds

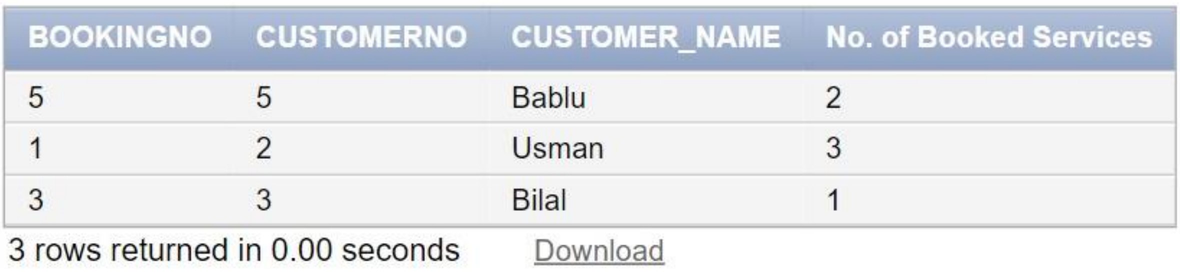
2) **Select bookingNo ,total_installments (bookingNo) "Installments" , bill_status frombooking**

   It will display total instalment made by customer and payment status



BOOKINGNO	Installments	BILL_STATUS
4	0	Pending
5	0	Pending
1	3	Paid
2	0	-
3	0	Pending

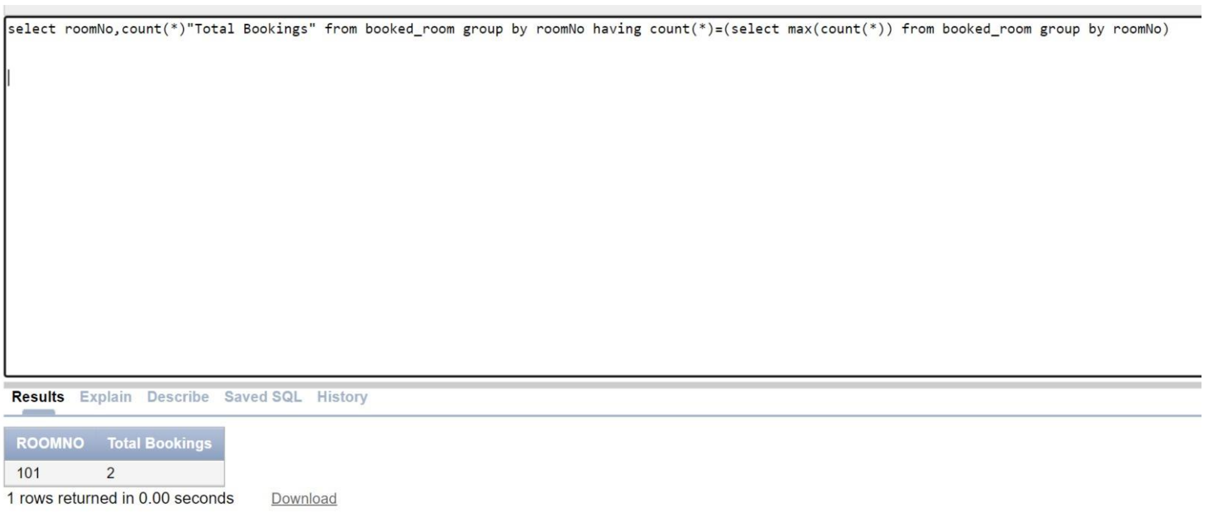
3) **Select * from booked_service_info**
   It will display total services booked by a customer



BOOKINGNO	CUSTOMERNO	CUSTOMER_NAME	No. of Booked Services
5	5	Bablu	2
1	2	Usman	3
3	3	Bilal	1

3 rows returned in 0.00 seconds

4)It will display paid amount against a booking

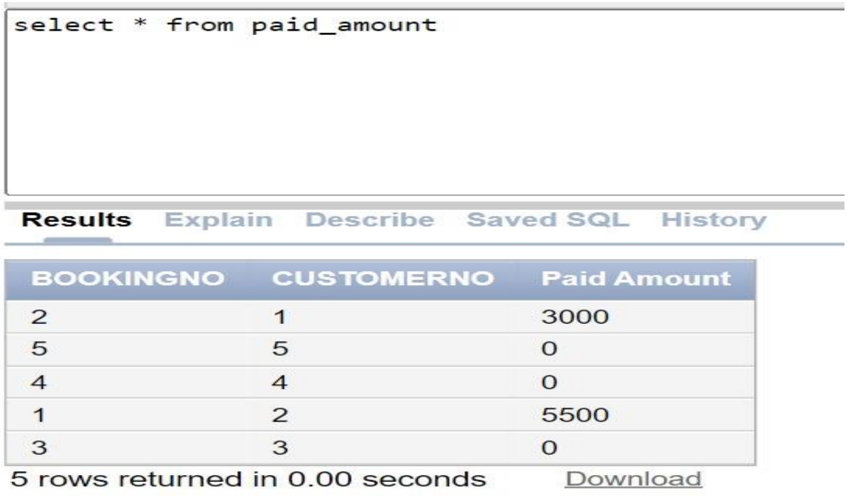


```
select roomNo,count(*)"Total Bookings" from booked_room group by roomNo having count(*)=(select max(count(*)) from booked_room group by roomNo)
```

ROOMNO	Total Bookings
101	2

1 rows returned in 0.00 seconds

5)It will display paid amount against a booking



```
select * from paid_amount
```

BOOKINGNO	CUSTOMERNO	Paid Amount
2	1	3000
5	5	0
4	4	0
1	2	5500
3	3	0

5 rows returned in 0.00 seconds