# ABSTRACT:

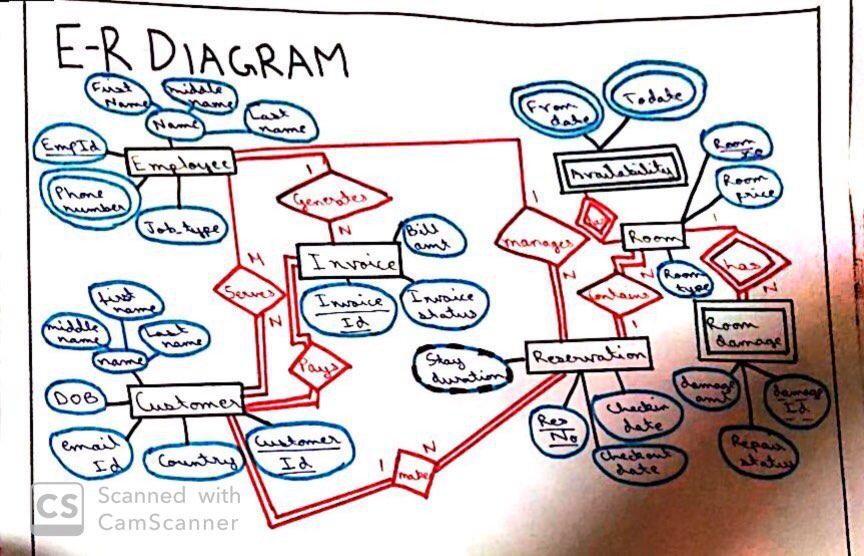
Customer can make reservation for stay in hotel. The reservation data is managed by Employee of the hotel. Reservation generates invoice which is to be paid by customer. Reservation contains room details.

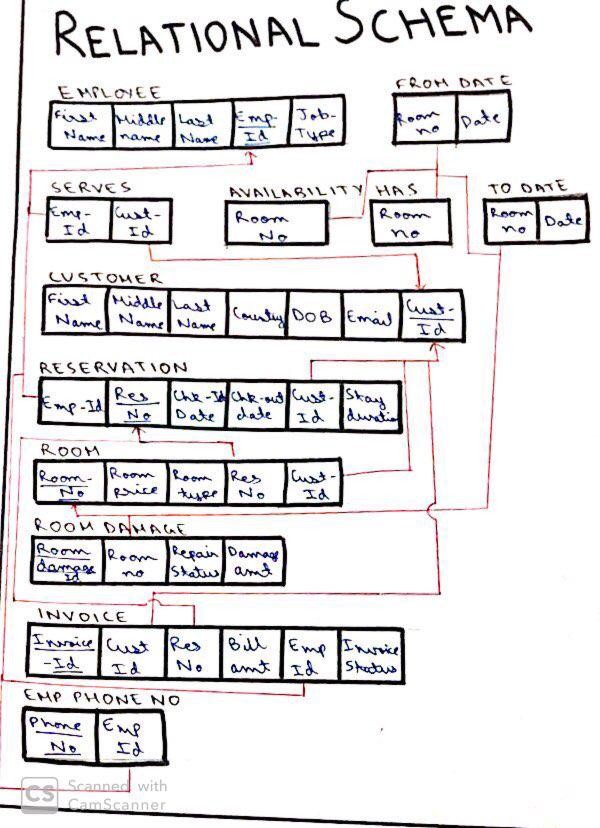
Room details can also be accessed by the customer, if any, room details also contains room damage details.

# PROJECT DESCRIPTION:

We have created a database schema according to following requirements:

1. Hotel stores employee data as their unique emp-id, Name (First, Middle, Last), Phone no. and Job-type.
2. Employee manages the reservation made by the customer, they also serve to various customer requirements.
3. Customer data is also store by the hotel as their unique Customer Id, Complete Name, E-mail, DOB and country.
4. Invoice contains bill amount, invoice status and Invoice id which is generated by the reservation which contains data as Reservation no., chk in date, chk out date and stay duration.
5. Reservation contains Room details as Room no., Room Price and Room type, and further, if any, contains Room Damage Details.
6. Not all employees manage reservations.
7. All reservations are made by customer.
8. A customer may have several room details.
9. Invoice is only generated if customer makes a reservation.
10. If any, Room damage details have data like Room damage id, Repair status and damage amount.





### SQL QUERIES:

**Created employee table**

create table Employee ( First\_N varchar2(10), Middle\_N varchar2(10),

Last\_N varchar2(10), Job\_type varchar2(5), Emp\_id number(5)

);

### Inserted values in to employee table

insert into Employee values ( 'Akshat','','Gupta','OFF',1111);

insert into Employee values

( 'Ishaan','','Jain','CLS',1112);

insert into Employee values

( 'Tim','John','Snik','RECP',1113);

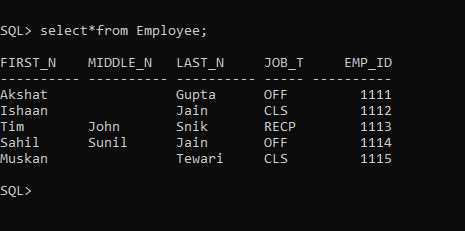
insert into Employee values

( 'Sahil','Sunil','Jain','OFF',1114);

insert into Employee values

( 'Muskan','','Tewari','CLS',1115);

Select \* from Employee;



### Room Table

create table Room ( Room\_No number(3),

Room\_Price number(5), Room\_type varchar2(4), Res\_No number(5), Cust\_id number(5)

);

### Inserted values in to room table

insert into Room values

( 101,2500,'STD',1234,8922);

insert into Room values

( 102,3500,'PREM',2345,8925);

insert into Room values

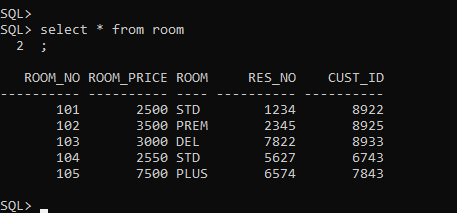
( 103,3000,'DEL',7822,8933);

insert into Room values

( 104,2550,'STD',5627,6743);

insert into Room values ( 105, 7500, 'PLUS',6574,7843);

select \* from Room;



### Creating customer table

create table Customer ( First\_N varchar2(10), Middle\_N varchar2(10),

Last\_N varchar2(10), Email varchar2(20),

Country varchar2(10), DOB date,

Cust\_id number(5)

);

### Inserting data into customer table

insert into Customer values (

'Muskan','','Tewari','muskantewari1@gmail.com','India',date '1996-08-02',8922);

insert into Customer values (

'Piyush','Sunil','Thakur','piyush98@gmail.com','China',date '1992-06-22',8925);

insert into Customer values ( 'Suresh','','Shukla','suresh66@gmail.com','India',date '1989-02-11',

8933);

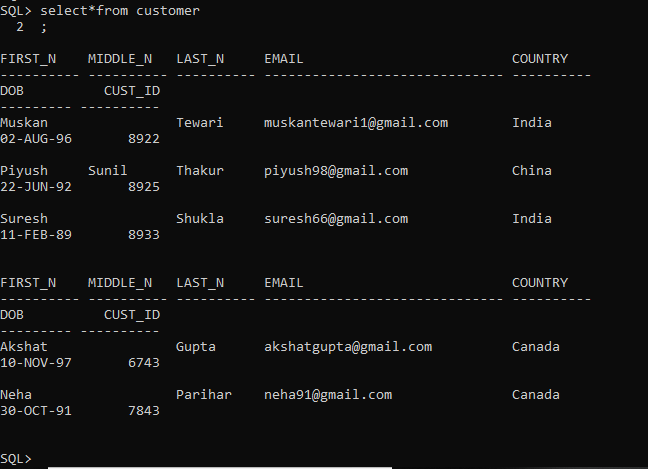
insert into Customer values (

'Akshat','','Gupta','akshatgupta@gmail.com','Canada',date '1997-11-10',6743);

insert into Customer values ( 'Neha','','Parihar','neha91@gmail.com','Canada',date '1991-10-30',

7843);

select \* from Customer;



### Created Room\_damage table

create table Room\_damage

( Room\_damage\_id number(5),

Room\_no number(3),

Repair\_status varchar2(10), Damage\_amt number(5),

constraint room\_dam\_pk primary key (Room\_damage\_id,Room\_no)

);

### Inserting values into room\_damage table :

insert into Room\_damage values ( 6734,102,'PENDING',550);

insert into Room\_damage values ( 7923,103,'DONE',400);

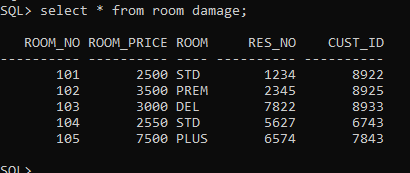
insert into Room\_damage values (

2343,102,'DONE',2000);

insert into Room\_damage values ( 7802,101,'PENDING',700);

insert into Room\_damage values ( 8923,101,'DONE',780);

Select \* from Room\_damage



### Created Serves table

create table Serves ( Emp\_id number(5), Cust\_id number(5), constraint serves\_pk primary key (Emp\_id,Cust\_id)

);

### Insert into Serves table

insert into Serves values ( 1113,8922);

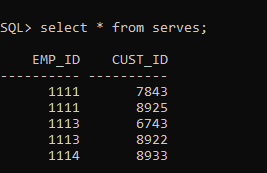
insert into Serves values ( 1111,8925);

insert into Serves values ( 1114,8933);

insert into Serves values ( 1113,6743);

insert into Serves values ( 1111,7843);

select \* from Serves;



### Created invoice table

create table Invoice (

Emp\_id number(5), Inv\_id number(5),

Cust\_id number(5), Res\_No number(5),

Bill\_amt number(5), Inv\_status varchar2(10), constraint invoice\_four\_pk primary key (Emp\_id, Inv\_id, Cust\_id,Res\_no)

);

### Inserted values in invoice table

insert into Invoice values ( 1111,10223,7843,6574,12550,'PENDING');

insert into Invoice values

( 1111,10312,8925,2345,5600,'PAID');

insert into Invoice values

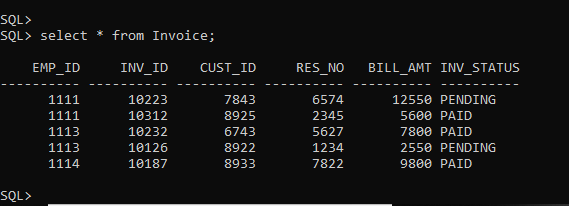
( 1113,10232,6743,5627,7800,'PAID');

insert into Invoice values ( 1113,10126,8922,1234,2550,'PENDING');

insert into Invoice values

( 1114,10187,8933,7822,9800,'PAID');

select \* from Invoice;



### Reservation table created

create table reservation ( Emp\_id number(5), Cust\_id number(5), Res\_no number(5), Chk\_in date,

Chk\_out date,

Stay\_dur number(3) as (Chk\_out - Chk\_in)

);

### Inserting into reservation table

insert into reservation (emp\_id, cust\_id, res\_no, chk\_in, chk\_out) values (1111,7843,6574,date '2018-06-18',date '2018-06-22');

insert into reservation (emp\_id, cust\_id, res\_no, chk\_in, chk\_out) values (1111,8925,2345,date '2018-07-09',date '2018-07-11');

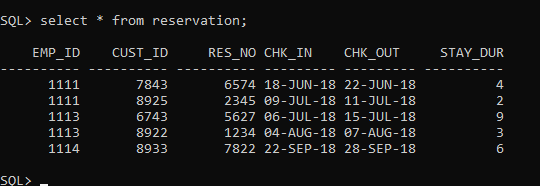
insert into reservation (emp\_id, cust\_id, res\_no, chk\_in, chk\_out) values (1113,6743,5627,date '2018-07-06',date '2018-07-15');

insert into reservation (emp\_id, cust\_id, res\_no, chk\_in, chk\_out) values (1113,8922,1234,date '2018-08-04',date '2018-08-07');

insert into reservation (emp\_id, cust\_id, res\_no, chk\_in, chk\_out) values (1114,8933,7822,date '2018-09-22',date '2018-09-28');

select \* from reservation;

### Reservation table



**All table primary key**

alter table Employee

add constraint Emp\_id\_pk primary key (Emp\_id);

alter table serves

add constraint serves\_pk primary key (Cust\_id, Emp\_id);

alter table customer

add constraint customer\_pk primary key (Cust\_id);

alter table reservation

add constraint reservation\_pk primary key (Res\_no);

alter table room

add constraint room\_pk primary key (Room\_no); alter table room\_damage

add constraint room\_damage\_pk primary key (Room\_damage\_id, Room\_no);

alter table invoice

add constraint invoice\_pk primary key (Cust\_id, Emp\_id, Inv\_id, Res\_no);

alter table emp\_phone\_no

add constraint emp\_phone\_no\_pk primary key (Phone\_no, Emp\_id);

### All foreign key linking

alter table Serves add constraint serves\_fk foreign key (Emp\_id) references Employee(Emp\_id);

alter table Serves

add constraint serves\_fk2 foreign key (Cust\_id) references Customer(Cust\_id);

alter table Reservation

add constraint serves\_fk3 foreign key (Emp\_id) references Employee(Emp\_id);

alter table Reservation

add constraint serves\_fk4 foreign key (Cust\_id) references Customer(Cust\_id);

alter table Room

add constraint serves\_fk5 foreign key (Res\_no) references Reservation(Res\_no);

alter table Room add constraint serves\_fk6 foreign key (Cust\_id) references Customer(Cust\_id);

alter table Room\_damage

add constraint serves\_fk7 foreign key (Room\_no) references Room(Room\_no);

alter table Invoice add constraint serves\_fk8 foreign key (Emp\_id) references Employee(Emp\_id);

### Dealing with multi-valued attributes

create table Emp\_phone\_no

( Emp\_id number(5), Phone\_no number(10),

constraint Emp\_phone\_fk foreign key (Emp\_id) references Employee (Emp\_id)

);

create type phone\_no as varray(5) of number(38); alter table Employee

add emp\_phone Phone\_no;

update Employee set Emp\_phone =

Phone\_no(9415118032,9616103567)

where Emp\_id=1111;

update Employee set Emp\_phone = Phone\_no(9792977807) where Emp\_id=1112;

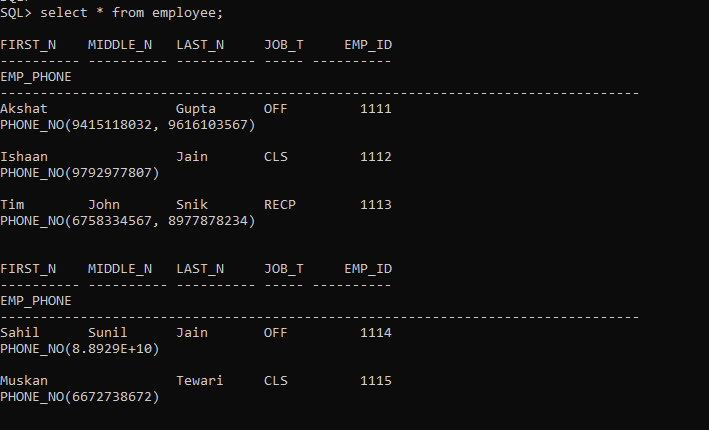
update Employee set Emp\_phone = Phone\_no(6758334567,8977878234)

where Emp\_id=1113;

update Employee set Emp\_phone = Phone\_no(88929392934) where Emp\_id=1114;

update Employee set Emp\_phone = Phone\_no(6672738672) where Emp\_id=1115;

### select \* from employee;



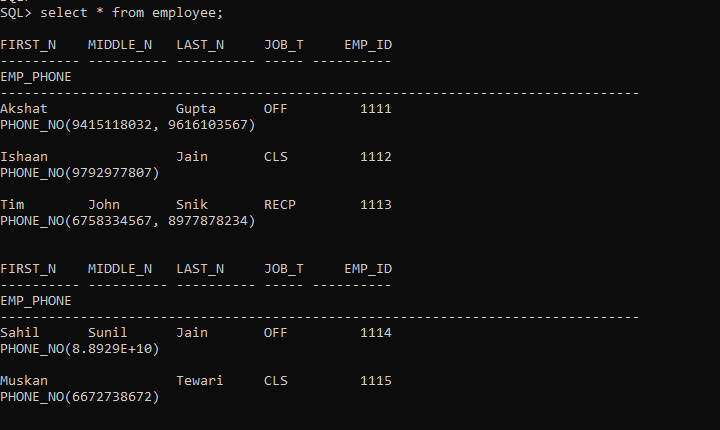
**QUERIES**

**Alter**

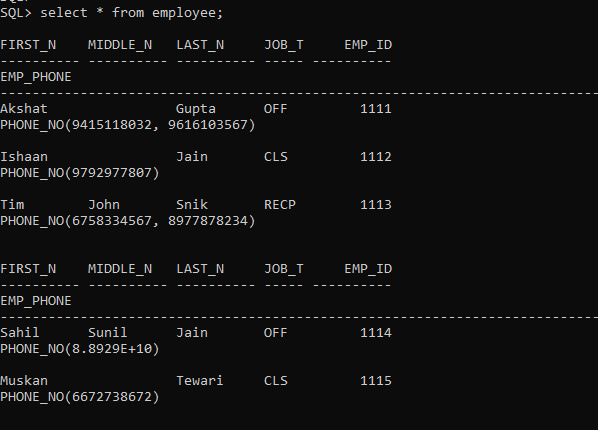
alter table Employee modify job\_type varchar2(5)

constraint emp\_nn not null;

### Delete



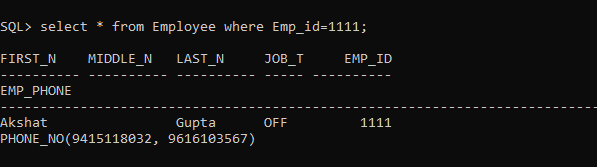
Delete from Employee where emp\_id=1117;



No record found.

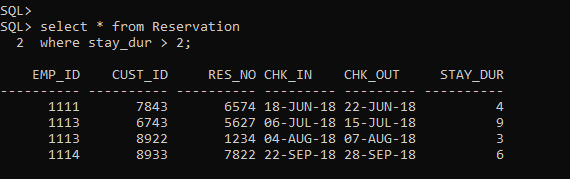
### Select with where clause

select \* from Employee where Emp\_id=1111;

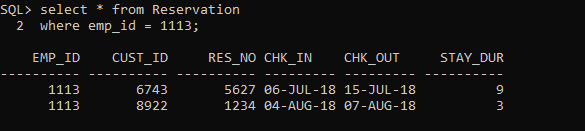


### Comparison Operator

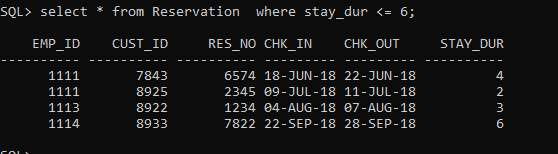
select \* from Reservation where stay\_dur > 2;



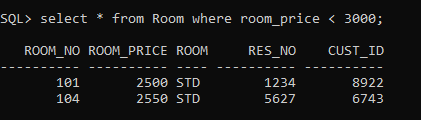
select \* from Reservation where emp\_id = 1113;



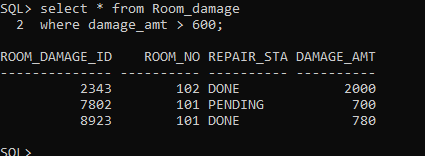
select \* from Reservation where stay\_dur <= 6;



select \* from Room where room\_price < 3000;

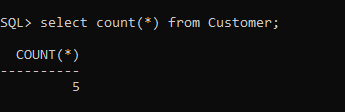


select \* from Room\_damage where damage\_amt > 600;

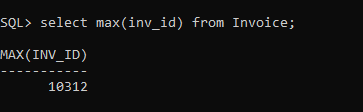


### Aggregate functions

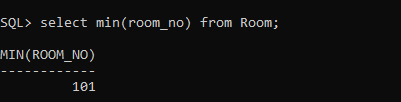
select count(\*) from Customer;



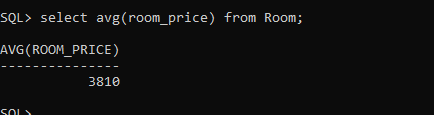
select max(inv\_id) from Invoice;



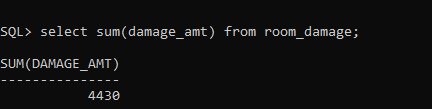
select min(room\_no) from Room;



select avg(room\_price) from Room;

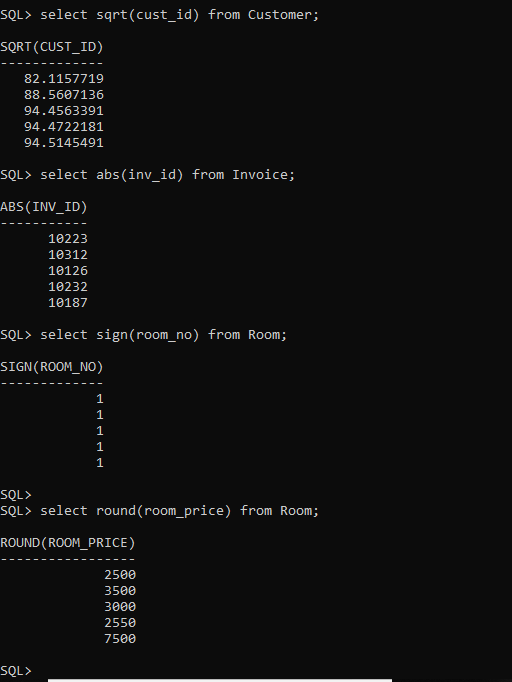


select sum(dama ge\_amt) from room\_damage;

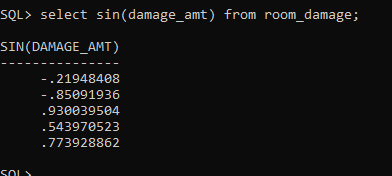


### Numeric functions

select sqrt(cust\_id) from Customer; select abs(inv\_id) from Invoice; select sign(room\_no) from Room; select round(room\_price) from Room;



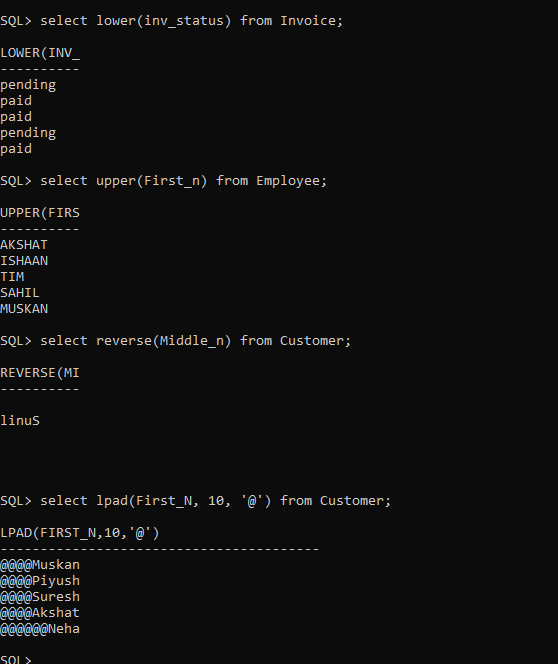
select sin(damage\_amt) from room\_damage;

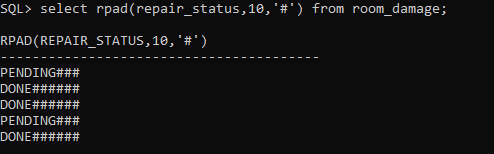


### String function

select lower(inv\_status) from Invoice; select upper(First\_n) from Employee; select reverse(Middle\_n) from Customer;

select lpad(First\_N, 10, '@') from Customer;

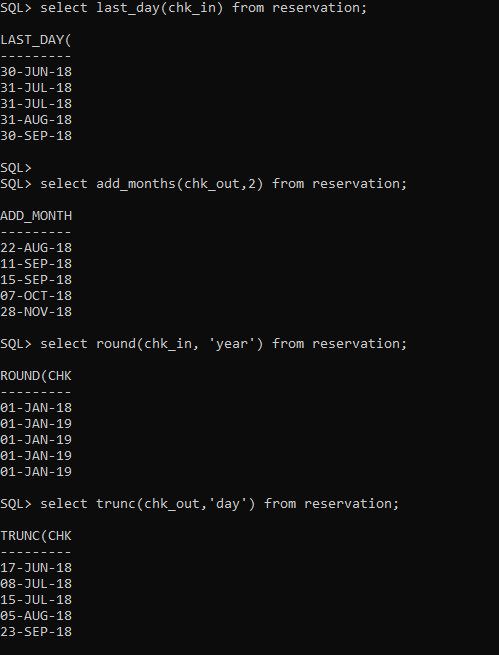


select rpad(repair\_status,10,'#') from room\_damage;

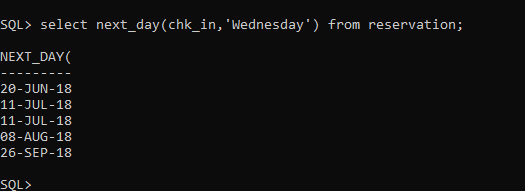
### Date functions

select last\_day(chk\_in) from reservation;

select add\_months(chk\_out,2) from reservation; select round(chk\_in, 'year') from reservation; select trunc(chk\_out,'day') from reservation;



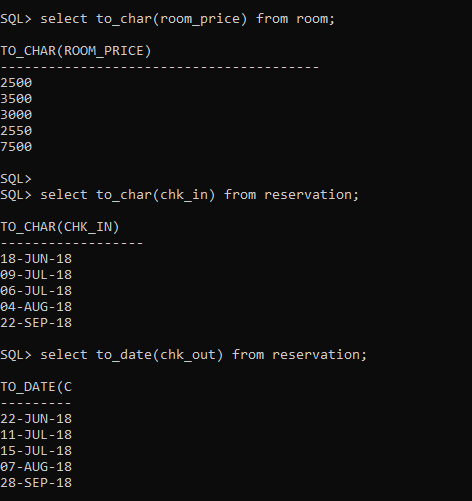
select next\_day(chk\_in,'Wednesday') from reservation;



### Conversion functions

select to\_char(room\_price) from room;

select to\_char(chk\_in) from reservation; select to\_date(chk\_out) from reservation;



### Set operator

select emp\_id, cust\_id from Invoice union all

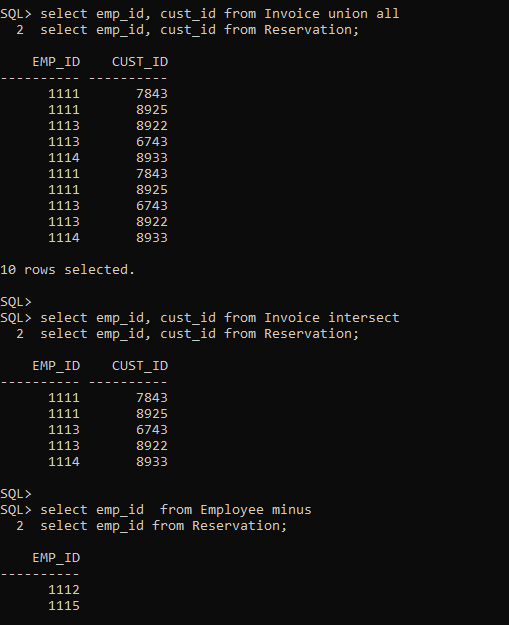
select emp\_id, cust\_id from Reservation;

select emp\_id, cust\_id from Invoice intersect

select emp\_id, cust\_id from Reservation;

select emp\_id from Employee minus

select emp\_id from Reservation;

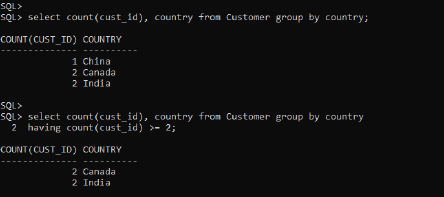


### Group by and having

select count(cust\_id), country from Customer group by country;

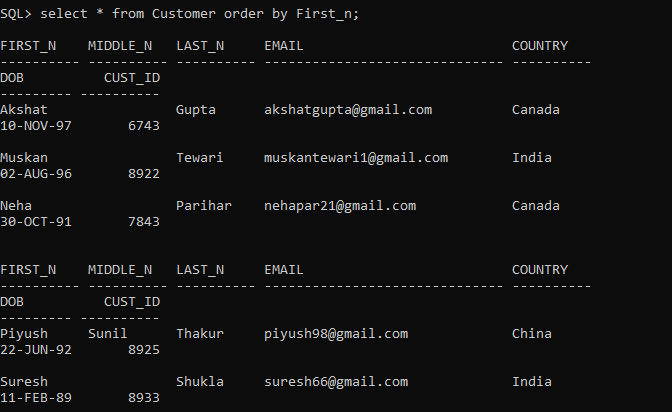
select count(cust\_id), country from Customer group by country

having count(cust\_id) >= 2;



### Order by clause

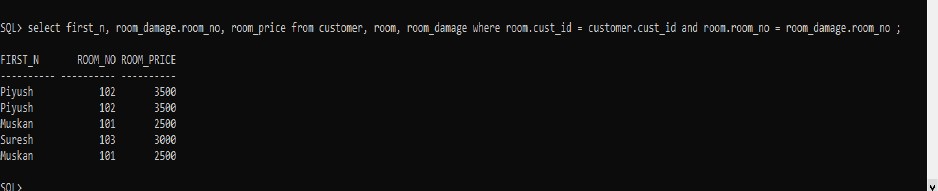
select \* from Customer order by First\_n;



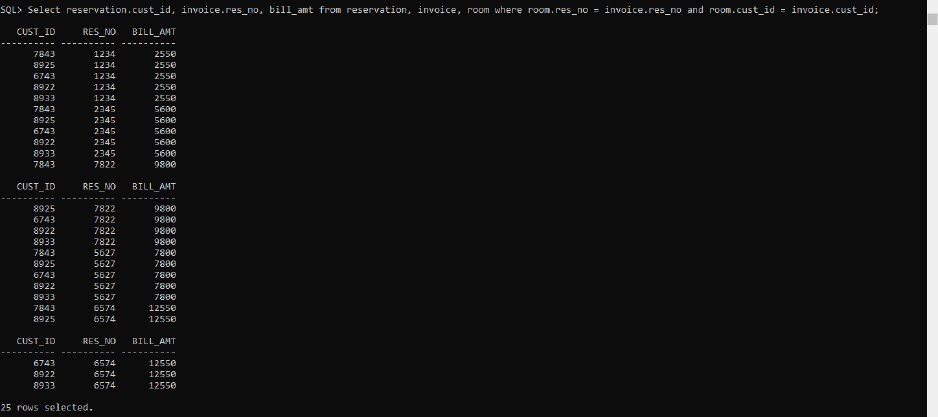
### Join queries - more than two tables (three)

**Join**

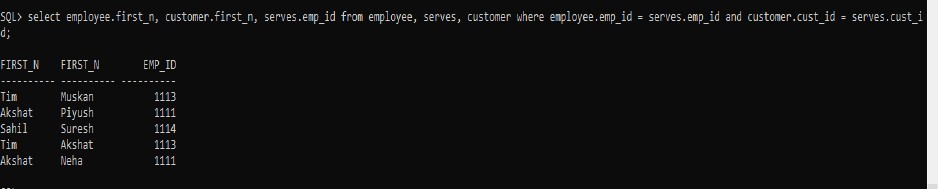
select first\_n, room\_damage.room\_no, room\_price from customer, room, room\_damage where room.cust\_id = customer.cust\_id and room.room\_no = room\_damage.room\_no ;



Select reservation.cust\_id, invoice.res\_no, bill\_amt from reservation, invoice, room where room.res\_no = invoice.res\_no and room.cust\_id = invoice.cust\_id;



select employee.first\_n, customer.first\_n, serves.emp\_id from employee, serves, customer where employee.emp\_id = serves.emp\_id and customer.cust\_id = serves.cust\_id;

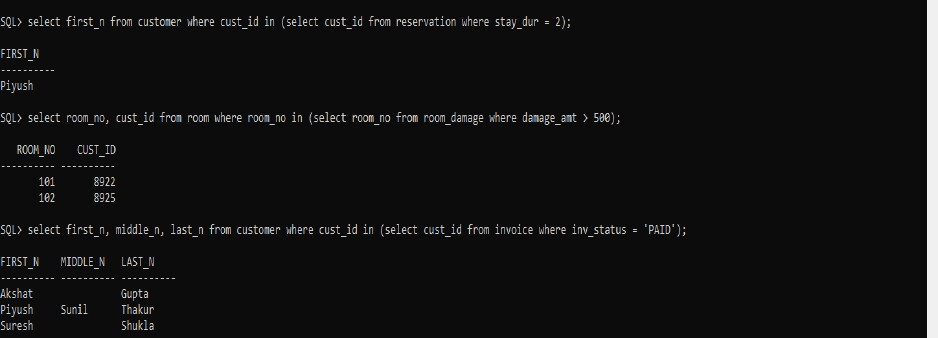


### Subquery

select first\_n from customer where cust\_id in (select cust\_id from reservation where stay\_dur = 2);

select room\_no, cust\_id from room where room\_no in (select room\_no from room\_damage where damage\_amt > 500);

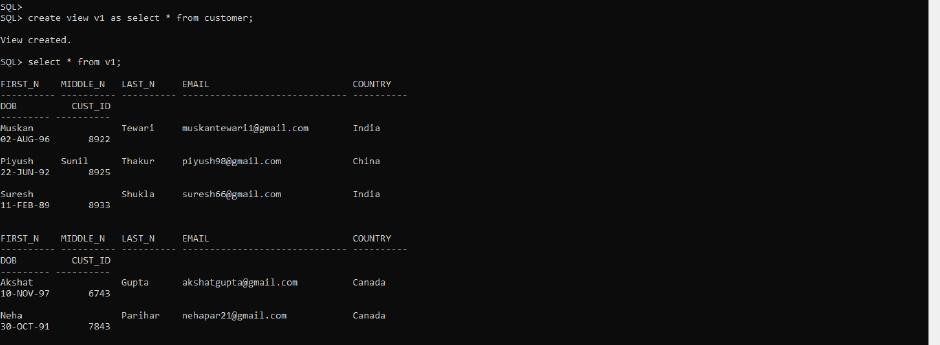
select first\_n, middle\_n, last\_n from customer where cust\_id in (select cust\_id from invoice where inv\_status = 'PAID');



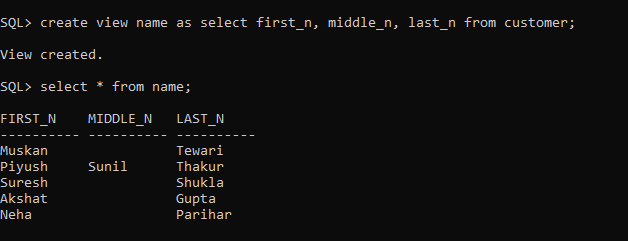
### Views

create view v1 as select \* from customer;

select \* from v1;



create view name as select first\_n, middle\_n, last\_n from customer; select \* from name;



# PROCEDURAL QUERIES

### Cursor

Set serveroutput on; declare country varchar2(5); cursor c2 is select \* from customer where country='India';

rows c2%rowtype;

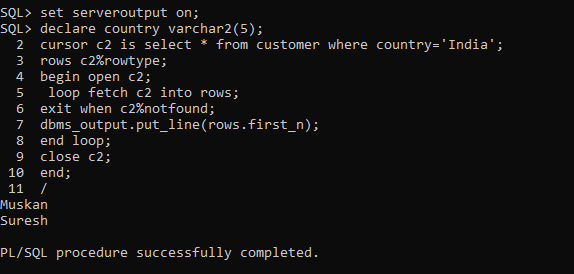
begin open c2; loop fetch c2 into rows; exit

when c2%notfound; dbms\_output.put\_line(ro ws.first\_n); end loop; close c2;

end;

/

Statement processed.



### Procedure

create or replace procedure isPrem(price in room.room\_no%type, guess out varchar) is cost number; begin select room\_price into cost from room where room\_price=price; if cost >= 2900 then

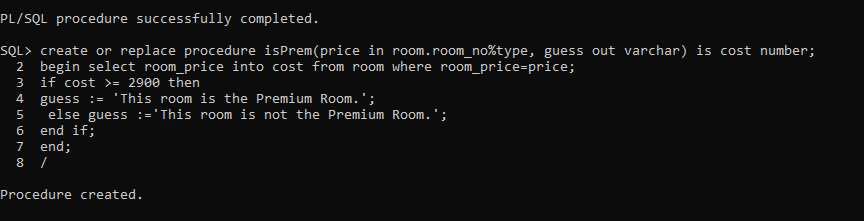
guess := 'This room is the Premium Room.'; else

guess :='This room is not the Premium Room.'; end if;

end;

/

Procedure created.



variable v varchar2(10); exec isPrem(101,:v); print v;

Statement processed.

This room is not the Premium Room.

### Function

create or replace function concession(no number) return number is

cost number; begin select room\_price into cost from room where room\_no=no;

cost:= cost/15; return cost; end

concessio n;

/

Function created.

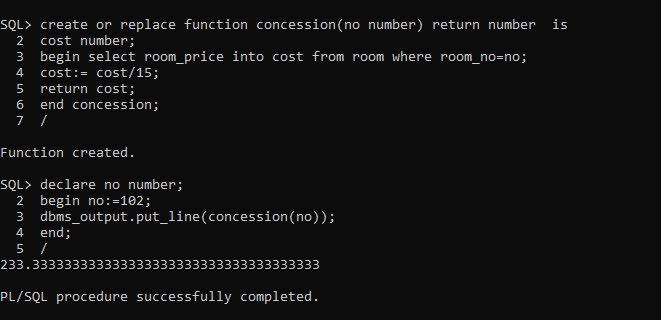
declare no number; begin no:=102;

dbms\_output.put\_line(conc ession(no));

end;

/

Statement processed. 233.333333333333333333333333333333333333



### Trigger

create or replace trigger t2 before update on room

for each row declare diﬀ number; begin

diﬀ := :NEW.room\_price - :OLD.room\_price;

dbms\_output.put\_line('Price before updation: '||:OLD.room\_price); dbms\_output.put\_line('Price after updation: '||: NEW.room\_price); dbms\_output.put\_line('Diﬀerence between the prices: '|| diﬀ); end;

/

Trigger created.

update room set room\_price = 2800 where room\_no = 104; 1 row(s) updated.

Price before updation: 2550 Price after updation: 2800

Diﬀerence between the prices: 250