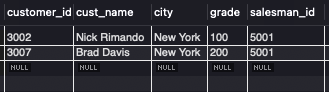
1. customers who are either from newyork or have grade<100

select customer\_id,cust\_name,city,grade,salesman\_id

from customer

where city='new york' or grade<100;



2. customers in new york who gave grade value above 100.

select customer\_id,cust\_name,city,grade,salesman\_id

from customer

where city='new york' and grade>100;

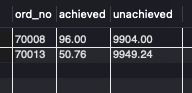


3.orders that exceed 50% of the target value of 6000.

select ord\_no, round((purch\_amt/6000)\*100,2)as achieved,round ((100- purch\_amt/6000)\*100,2) as unachieved

from orders

where purch\_amt > 6000\*0.5;

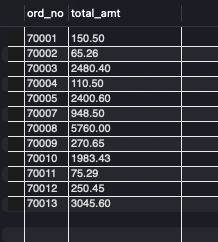


4. calculate the total purchase amount of all orders

select ord\_no,sum(purch\_amt) as 'total\_amt'

from orders

Group by ord\_no;

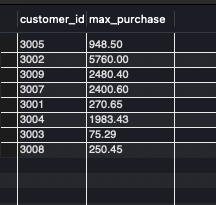


5.the highest purchased amount ordered by each customers

select customer\_id, max(purch\_amt) as max\_purchase

from orders

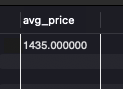
group by customer\_id;



6. calculate the average product price

select avg(pro\_price) as 'avg\_price'

from item\_mast;



7. employees whose department is located at toronto.

select e.first\_name,e.last\_name,e.employee\_id,e.job\_id

from employees e

join departments d

on e.department\_id= d.department\_id

join locations l

on d.location\_id=l.location\_id

where l.city='toronto';



8. employees whose salary is lower than that of employees whose job title is MK\_MAN.excluding mk\_man

select employee\_id,first\_name,last\_name,job\_id

from employees

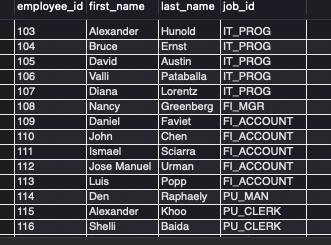
where salary<

(select salary

from employees

where job\_id='MK\_MAN')

and job\_id!= 'MK\_MAN';



9. employees whose department id is 40 or 80.

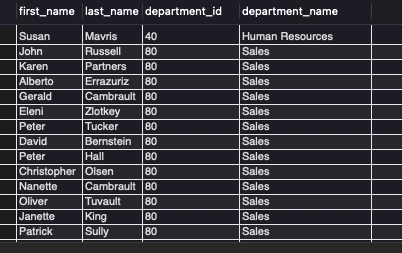
select e.first\_name, e.last\_name, d.department\_id,d.department\_name

from employees e

join departments d

on e.department\_id=d.department\_id

where d.department\_id in (40,80);



10.calculation of avg. salary, number of employees who are getting commission

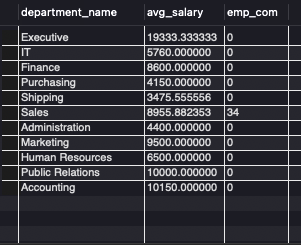
select d.department\_name, avg(e. salary) as avg\_salary,count(commission\_pct) as emp\_com

from employees e

join departments d

on e.department\_id = d.department\_id

group by d.department\_id;



11. employees with the same designation to 169.

select first\_name,last\_name,job\_id,department\_id

from employees

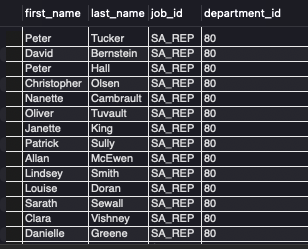
where job\_id=

(select job\_id

from employees

where employee\_id= 169)

and employee\_id!=169;



12. employees who earns more than avg. salary

select employee\_id,first\_name,last\_name

from employees

where salary >

(select avg(salary)

from employees);



13. employees who work in finance department

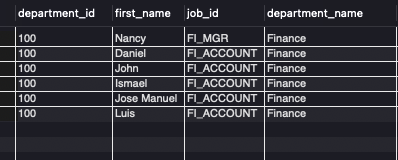
select d.department\_id, e.first\_name,e.job\_id,d.department\_name

from employees e

join departments d

on e.department\_id= d.department\_id

where d.department\_name= 'finance';



14. employees who earn less than id 182.

select employee\_id, first\_name,last\_name,salary

from employees

where salary<

(select salary

from employees

where employee\_id=182)

and employee\_id!=182;



15. procedure for counting employees in each department.

delimiter //

create procedure

countemployeesbydept()

begin

select department\_id,

count(\*) as emp\_count

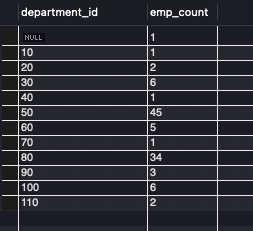
from employees

group by department\_id;

end //

delimiter ;

Call countemployeesbydept;



16. procedure for adding new employees

delimiter //

create procedure

addnewemployee (

in p\_employee\_id int unsigned,

in p\_first\_name varchar(20),

in p\_last\_name varchar(25),

in p\_email varchar(25),

in p\_phone\_number varchar(20),

in p\_hire\_date date,

in p\_job\_id varchar(10),

in p\_salary decimal (8,2),

in p\_commission\_pct decimal(2,2),

in p\_manager\_id int unsigned,

in p\_department\_id int unsigned)

begin

insert into employees

(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,commission\_pct,manager\_id,department\_id)

values(p\_employee\_id, p\_first\_name,p\_last\_name,p\_email,p\_phone\_number,p\_hire\_date,p\_job\_id,p\_salary,p\_commission\_pct,p\_manager\_id,p\_department\_id);

end //

delimiter ;

17. deleting employees from specific department

delimiter //

create procedure

deleteemployeesbydept(

in p\_department\_id int)

begin

delete from employees

where department\_id=p\_department\_id;

end //

delimiter ;

18.procedure for top paid employees in each department

delimiter //

create procedure

gettoppaidemployee ()

begin

select salary, employee\_id,department\_id, max(salary) as max\_sal

from employees

group by department\_id;

end //

delimiter ;

call gettoppaidemployee() ;

19. changing employee salary and job role

delimiter //

create procedure

promoteEmployee (

in p\_new\_salary decimal(8,2),

in p\_employee\_id int,

in p\_new\_job\_id varchar(20)

)

begin

update employees

set salary=p\_new\_salary,

job\_id=p\_new\_job\_id

where employee\_id=p\_employee\_id ;

end //

delimiter ;

20. new manager to all employees in a specific department

delimiter //

create procedure

assignmanagertodepartment (

in p\_department\_id int,

in p\_manager\_id int)

begin

update employees

set manager\_id=p\_manager\_id

where department\_id=p\_department\_id ;

end //

delimiter ;

theorycal assignment

-- 1. compare sql and nosql databases

1. full form

sql. : structured query language

nosql : not only sql

2. data type

sql. : it uses relational databases.

nosql : it uses distributed databases.

3. data form

sql. : data used for sql is mostly in the form of table, rows and columns.

nosql : data used for nosql is in the form of graphs, documents etc.

4. examples

sql : mysql, postgresql,oracle etc.

nosql : mongoDB, couchDB etc.

– 2. normlization vs denormalization

1.NORMALIZATION

i) what is normlization?

Normalization is the process for organizing data to eliminate redundancy.

ii)objectives:

Remove the duplicate data

Eliminate redundancy

iii)normal forms:

1NF

2NF

3NF

BCNF

2.DENORMALIZATION

i) what is denormalization?

Denormalization is the process of combining normalized data.

ii)objectives:

Optimize read operation

Improve query performance