**MY SQL COMPLETE QUERIES**

use employees;

***# SQL SELECT STATEMENT#***

select \* from employees;

select \* from employees where first\_name = 'Denis' and gender = 'M';

select \* from employees where first\_name = 'Denis' or gender = 'M';

select \* from employees where first\_name = 'Denis' and ( gender = 'M' or gender = 'M');

select \* from employees where first\_name = 'Denis' or first\_name = 'Mark' or first\_name = 'Nathan';

select \* from employees where first\_name in ('Denis' ,'Mark' ,'Nathan');

select \* from employees where first\_name not in ('Denis' ,'Mark' ,'Nathan');

select \* from employees where first\_name like ( 'Den%');

select \* from employees where first\_name like ( '%en');

select \* from employees where first\_name like ( '%en%');

select \* from employees where first\_name not like ( 'Den%');

select \* from employees where first\_name not like ( '%en');

select \* from employees where first\_name not like ( '%en%');

select \* from employees where hire\_date between '1990-01-01' and '2000-01-01';

select \* from employees where hire\_date not between '1990-01-01' and '2000-01-01';

select \* from employees where first\_name is null;

select \* from employees where first\_name is not null;

select \* from employees where first\_name <> 'mark';

select \* from employees where first\_name != 'mark';

select \* from employees where hire\_date > '2000-01-01';

select distinct \* from employees;

***# SQL AGGREGATE FUNCTION #***

select count(salary) from salaries;

select count(\*) from salaries;

select count(distinct (salary))from salaries;

select sum(salary) from salaries;

select min(salary) from salaries;

select max(salary) from salaries;

select avg(salary) from salaries;

select round( avg(salary),2 )from salaries;

***# SQL GROUP BY , HAVING , ORDER BY #***

select \* from employees order by first\_name ;

select \* from employees order by first\_name desc;

select \* from employees group by first\_name ;

select emp\_no,count(salary) from salaries group by emp\_no;

select First\_name , count(first\_name) from employees group by emp\_no;

select First\_name , count(first\_name) as total from employees group by emp\_no;

select \* from employees having hire\_date > '2000-01-01';

select \* from employees group by first\_name having count(first\_name) > 250 order by first\_name ;

select First\_name , count(first\_name) as total from employees where hire\_date >'1999-01-01'group by first\_name having count(first\_name) < 200 order by first\_name ;

select \* from employees limit 500;

***# SQL JOINS #***

select e.emp\_no,e.first\_name,e.last\_name,e.hire\_date,s.salary from employees e inner join salaries s on e.emp\_no = s.emp\_no order by e.emp\_no;

select e.emp\_no,e.first\_name,e.last\_name,e.hire\_date,s.salary from employees e left join salaries s on e.emp\_no = s.emp\_no order by e.emp\_no;

select e.emp\_no,e.first\_name,e.last\_name,e.hire\_date,s.salary from employees e right join salaries s on e.emp\_no = s.emp\_no order by e.emp\_no;

select e.emp\_no,e.first\_name,e.last\_name,e.hire\_date,s.salary from employees e cross join salaries s on e.emp\_no = s.emp\_no order by e.emp\_no;

select e.emp\_no,e.first\_name,e.last\_name,e.hire\_date,s.salary from employees e  join salaries s on e.emp\_no = s.emp\_no where e.first\_name = 'George';

***# SQL JOINS AGGREGATE FUNCTION #***

select e.emp\_no,e.gender,avg(s.salary) as Avg\_Sal from employees e inner join salaries s on e.emp\_no = s.emp\_no group by gender;

***# JOIN WITH JOIN #***

select e.emp\_no,e.first\_name,e.last\_name,e.hire\_date, t.title,s.salary from employees e  join titles t on e.emp\_no = t.emp\_no join salaries s on t.emp\_no = s.emp\_no;

***# UNION #***

select first\_name from employees union select salary from salaries;

***# SUB QUERY #***

select e.first\_name ,e.last\_name from employees e where e.emp\_no in (select s.salary from salaries s);

select e.first\_name ,e.last\_name from employees e where exists (select s.salary from salaries s);

***# PROCEDURE WITHOUT PARAMETER #***

delimiter $$

create procedure select\_employe()

begin

select \* from employees limit 1000;

end$$

delimiter ;

***to call***

call employees.select\_employe();

call select\_employe();

***# PROCEDURE WITH IN PARAMETER #***

DELIMITER $$

USE employees $$

CREATE PROCEDURE emp\_salary ( IN p\_emp\_no INTEGER )

BEGIN

SELECT e.first\_name , e.last\_name , s.salary , s . from\_date , s.to\_date

FROM  employees e

JOIN

salaries s ON e.emp\_no = s.emp\_no

WHERE e.emp\_no = p\_emp\_no;

END $$

DELIMITER ;

***# PROCEDURE WITH IN AND OUT PARAMETER #***

DELIMITER $$

USE employees $$

CREATE PROCEDURE emp\_avg\_salary ( IN p\_emp\_no INTEGER , out p\_avg\_salary decimal (10,2) )

BEGIN

SELECT avg( s.salary)into p\_avg\_salary

FROM  employees e

JOIN salaries s ON e.emp\_no = s.emp\_no

WHERE e.emp\_no = p\_emp\_no;

END $$

DELIMITER ;

***# FUNCTION #***

DELIMITER $$

CREATE function F\_emp\_avg\_salary ( p\_emp\_no INTEGER ) returns decimal (10,2)

deterministic no sql reads sql data

BEGIN

declare v\_avg\_salary decimal(10,2);

SELECT avg( s.salary)

into v\_avg\_salary

FROM  employees e

JOIN

salaries s ON e.emp\_no = s.emp\_no

WHERE e.emp\_no = p\_emp\_no;

return v\_avg\_salary;

END $$

DELIMITER ;

***tocall***

select F\_emp\_avg\_salary(11300);

***# WINDOWS FUNCTION #***

select emp\_no, salary, row\_number () over ( partition by emp\_no) as row\_num from salaries;

select emp\_no, salary, row\_number () over ( order by emp\_no) as row\_num from salaries;

select emp\_no, salary, row\_number () over ( order by emp\_no desc) as row\_num from salaries;

***# SEVERAL WINDOWS FUNCTION #***

select emp\_no, salary,

row\_number () over () as row\_num1 ,

row\_number () over ( partition by emp\_no) as row\_num2 ,

row\_number () over ( partition by emp\_no order by salary desc) as row\_num3 ,

row\_number () over ( order by salary desc) as row\_num

from salaries

order by emp\_no;

***# WINDOW W IN FUNCTION 1#***

select emp\_no, salary,

row\_number () over w as row\_num from salaries

window w as ( partition by emp\_no) ;

***# WINDOW W IN FUNCTION 2#***

select distinct emp\_no, salary,

row\_number () over w as row\_num from salaries

window w as ( partition by emp\_no) ;

***# RANK METHOD #***

select emp\_no, salary,

rank () over w as rank\_num from salaries

where emp\_no = 11839

window w as ( partition by emp\_no order by salary desc) ;

***# DENSE RANK METHOD #***

select emp\_no, salary,

dense\_rank () over w as rank\_num from salaries

where emp\_no = 11839

window w as ( partition by emp\_no order by salary desc) ;

***# RANK METHOD WITH JOIN #***

SELECT e.emp\_no,

RANK() OVER w as employee\_salary\_ranking, s.salary

FROM employees e

JOIN salaries s ON s.emp\_no = e.emp\_no

WHERE e.emp\_no BETWEEN 10500 AND 10600

WINDOW w as (PARTITION BY e.emp\_no ORDER BY s.salary DESC);

***# DENSE RANK METHOD WITH JOIN #***

SELECT e.emp\_no,

DENSE\_RANK() OVER w as employee\_salary\_ranking,s.salary,e.hire\_date,s.from\_date,

(YEAR(s.from\_date) - YEAR(e.hire\_date)) AS years\_from\_start

FROM employees e

JOIN salaries s ON s.emp\_no = e.emp\_no

AND YEAR(s.from\_date) - YEAR(e.hire\_date) >= 5

WHERE e.emp\_no BETWEEN 10500 AND 10600

WINDOW w as (PARTITION BY e.emp\_no ORDER BY s.salary DESC);

***# LAG AND LEAD VALVE WINDOW FUNCTION - MODEL 1 #***

SELECT emp\_no,salary,

LAG(salary) OVER w AS previous\_salary,

LEAD(salary) OVER w AS next\_salary,

salary - LAG(salary) OVER w AS diff\_salary\_current\_previous,

LEAD(salary) OVER w - salary AS diff\_salary\_next\_current

FROM salaries

WHERE salary > 80000 AND emp\_no BETWEEN 10500 AND 10600

WINDOW w AS (PARTITION BY emp\_no ORDER BY salary);

***# LAG AND LEAD VALVE WINDOW FUNCTION - MODEL 2 #***

SELECT emp\_no,salary,

LAG(salary) OVER w AS previous\_salary,

LAG(salary, 2) OVER w AS 1\_before\_previous\_salary,

LEAD(salary) OVER w AS next\_salary,

LEAD(salary, 2) OVER w AS 1\_after\_next\_salary

FROM salaries

WINDOW w AS (PARTITION BY emp\_no ORDER BY salary)

LIMIT 1000;

***# AGGREGATE FUNCTIONS IN THE CONTEXT OF WINDOW FUNCTIONS -1  #***

SELECT s1.emp\_no, s.salary, s.from\_date, s.to\_date

FROM salaries s

JOIN

(SELECT emp\_no, MIN(from\_date) AS from\_date

FROM salaries

GROUP BY emp\_no) s1 ON s.emp\_no = s1.emp\_no

WHERE

s.from\_date = s1.from\_date;

***# AGGREGATE FUNCTIONS IN THE CONTEXT OF WINDOW FUNCTIONS - 2 #***

SELECT

de2.emp\_no, d.dept\_name, s2.salary, AVG(s2.salary) OVER w AS average\_salary\_per\_department

FROM

(SELECT de.emp\_no, de.dept\_no, de.from\_date, de.to\_date

FROM dept\_emp de

JOIN

(SELECT emp\_no, MAX(from\_date) AS from\_date

FROM dept\_emp

GROUP BY emp\_no) de1 ON de1.emp\_no = de.emp\_no

WHERE de.to\_date < '2002-01-01'

AND de.from\_date > '2000-01-01'

AND de.from\_date = de1.from\_date) de2

JOIN

(SELECT s1.emp\_no, s.salary, s.from\_date, s.to\_date

FROM salaries s

JOIN

(SELECT emp\_no, MAX(from\_date) AS from\_date

FROM salaries

GROUP BY emp\_no) s1 ON s.emp\_no = s1.emp\_no

WHERE

s.to\_date < '2002-01-01'

AND s.from\_date > '2000-01-01'

AND s.from\_date = s1.from\_date) s2 ON s2.emp\_no = de2.emp\_no

JOIN

departments d ON d.dept\_no = de2.dept\_no

GROUP BY de2.emp\_no, d.dept\_name

WINDOW w AS (PARTITION BY de2.dept\_no)

ORDER BY de2.emp\_no, salary;