**SubQueries**

**What is a subquires?**

If you want to get data from one table based on input value of other tables in this case we using subquires.

Example : If you want ta get data from table employees we need to check the conditions in departments tables .

If you want to fetch data from one table based on another tables data in this situation

* A quires with in other quires

SELECT select\_list FROM table

WHERE exp operator( SELECT select\_list FROM table );

* + - * + Red -> inner quires
        + Black -> outer quires
  + The subquires (inner quires) executes before the main quires(outer quires)
  + The result of sub quires is used by the main quires(outer quires)

**Type of Subquires**

Two type of Subquires

* Single row subquires
* Multiple row subquires

**Single Row Subquires**

Quires that returns only one row from the inner select statement

For Example: Which employees salary grate then Abel?

SELECT last\_name, salary From employees

WHERE salary > (SELECT salary

From employees

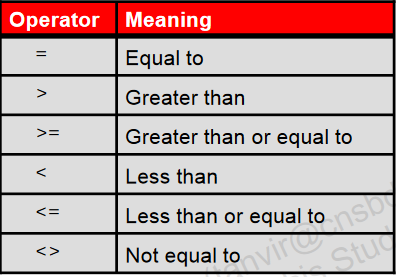
WHERE last\_name = 'Abel');

Single row Subquires:

* Group Function in a Subquires
* Having clause with Subquires

Single row Subquires:

* Return only one row
* Use single row comparison operators



**Greater Than (>)**

SELECT last\_name, job\_id, salary

FROM employees

WHERE salary > (SELECT MIN(salary)

FROM employees);

* **Which employees salary grate than Abel?**

SELECT last\_name, salary

From employees

WHERE salary > (SELECT salary

From employees

WHERE last\_name = 'Abel');

* **Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.**

SELECT last\_name,salary

FROM employees

WHERE salary > (SELECT AVG(salary)

FROM employees)

ORDER BY salary;

**Less Than (<)**

SELECT last\_name, job\_id, salary

FROM employees

WHERE salary < (SELECT MIN(salary)

FROM employees);

* **Which employees salary Less than Abel?**

SELECT last\_name, salary

From employees

WHERE salary < (SELECT salary

From employees

WHERE last\_name = 'Abel');

**Equal to(=)**

SELECT last\_name, job\_id, salary

FROM employees

WHERE salary = (SELECT MIN(salary)

FROM employees);

* **Display the employees whose jobs ID is the same as that of employee 141?**

SELECT last\_name, job\_id, salary

FROM employees

WHERE job\_id = (SELECT job\_id

FROM employee

WHERE employee\_id =141);

* **Display Department Name For Employee number 114?**

SELECT department\_name

From departments

WHERE department\_id =(SELECT department\_id

FROM employees

WHERE employee\_id=&employee\_id);

* **Get Accounting Employees details?**

SELECT \* FROM employees

WHERE department\_id = (SELECT department\_id

FROM DEPARTMENTS

WHERE department\_name ='&department\_name');

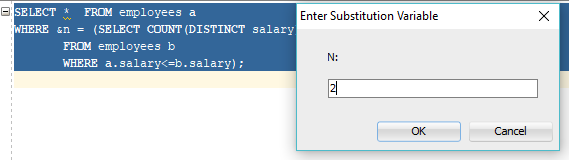
* Display nth highest salary

SELECT \* FROM employees a

WHERE &n = (SELECT COUNT(DISTINCT salary)

FROM employees b

WHERE a.salary<=b.salary);





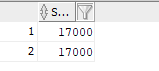
OR

SELECT salary FROM employees a

WHERE &n = (SELECT COUNT(DISTINCT salary)

FROM employees b

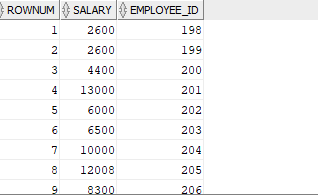
WHERE a.salary<=b.salary);



* **How to find Top Three Salaries from employee table ?**

SELECT rownum,employee\_id

FROM employees;

****

SELECT rownum, salary

FROM ( SELECT salary FROM employees

ORDER BY salary DESC)

WHERE rownum<=3;

****

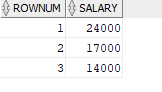
**OR**

SELECT rownum, salary

FROM ( SELECT DISTINCT salary FROM employees

ORDER BY salary DESC )

WHERE rownum<=3;

****

**Greater Than or equal to(>=)**

SELECT last\_name, job\_id, salary

FROM employees

WHERE salary >= (SELECT MIN(salary)

FROM employees);

* **Which employees salary Greater Than or equal to Abel?**

SELECT last\_name, salary

From employees

WHERE salary >= (SELECT salary

From employees

WHERE last\_name = 'Abel');

**Less Than or equal to(<=)**

SELECT last\_name, job\_id, salary

FROM employees

WHERE salary <= (SELECT MIN(salary)

FROM employees);

* **Which employees salary Greater Than or equal to Abel?**

SELECT last\_name, salary From employees

WHERE salary <= (SELECT salary

From employees

WHERE last\_name = 'Abel');

**NOT Equal To(<>)**

SELECT last\_name, job\_id, salary

FROM employees

WHERE salary <> (SELECT MIN(salary)

FROM employees);

* **The HR department needs a query that prompts the user for an employee last name. The query then displays the last name and hire date of any employee in the same department as the employee whose name they supply (excluding that employee). For example, if the user enters Zlotkey, find all employees who work with Zlotkey (excluding Zlotkey)**

UNDEFINE enter\_name

SELECT last\_name, hire\_date

FROM employees

WHERE department\_id = (SELECT department\_id

FROM employees

WHERE last\_name='&enter\_name')

AND

last\_name<>'&enter\_name';

**HAVING Clause with Subqueries**

* The Oracle server executes the subqueries first.
* The Oracle server returns results into the HAVINGclause of the main query.

SELECT department\_id, MIN(salary)

FROM employees

GROUP BY department\_id

HAVING MIN(salary) > (SELECT MIN(salary)

FROM employees

WHERE department\_id=50);

* **Find the job with the lowest average salary.**

**SELECT job\_id, AVG(salary)**

**FROM employees**

**GROUP BY job\_id**

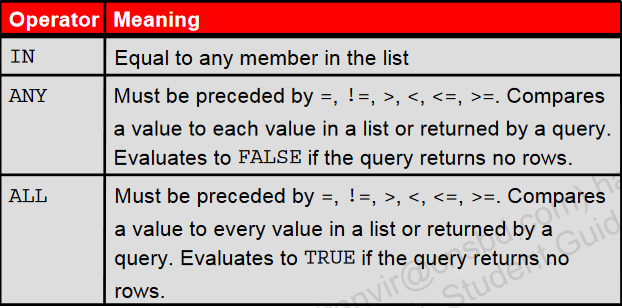
**HAVING AVG(salary) =** (SELECT MIN(AVG(salary))

FROM employees

GROUP BY job\_id);

**Multiple-Row Subqueries**

* Return more than one row
* Use multiple-row comparison operators



**IN**

SELECT last\_name, salary, department\_id

FROM employees

WHERE salary IN (SELECT MIN(salary)

FROM employees

GROUP BY department\_id);

* **Find the employees who earn the same salary as the minimum salary for each department.**

SELECT last\_name, salary, department\_id

FROM employees

WHERE salary IN (2500, 4200, 4400, 6000, 7000, 8300, 8600, 17000);

* **Create a report for HR that displays the last name and salary of every employee who reports to King.**

SELECT last\_name , salary

FROM employees

WHERE manager\_id IN (SELECT employee\_id

FROM employees

WHERE last\_name='King');

* **display the employee number, last name, and salary of all employees who earn more than the average salary and who work in a department with any employee whose last name contains a “u.”?**

SELECT employee\_id, last\_name, salary

FROM employees

WHERE department\_id IN (SELECT department\_id

FROM employees

WHERE last\_name like '%u%')

AND salary > (SELECT AVG(salary)

FROM employees);

* **Create a report for HR that displays the department number, last name, and job ID for every employee in the Executive department?**

SELECT department\_id, last\_name, job\_id

FROM employees

WHERE department\_id IN (SELECT department\_id

FROM departments

WHERE department\_name = 'Executive');

* **The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700.**

SELECT last\_name, department\_id, job\_id

FROM employees

WHERE department\_id IN (SELECT department\_id

FROM departments

WHERE location\_id = 1700);

* **Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains a “u.”**

SELECT employee\_id, last\_name

FROM employees

WHERE department\_id IN (SELECT department\_id

FROM employees

WHERE last\_name like '%u%')

**ANY**

* <ANY means less than the maximum.
* >ANY means more than the minimum.
* =ANY is equivalent to IN

SELECT employee\_id, last\_name, job\_id, salary

FROM employees

WHERE salary < ANY(SELECT salary

FROM employees

WHERE job\_id = 'IT\_PROG')

AND job\_id <> 'IT\_PROG';

SELECT employee\_id, last\_name, job\_id, salary

FROM employees

WHERE salary > ANY(SELECT salary

FROM employees

WHERE job\_id = 'IT\_PROG')

AND job\_id <> 'IT\_PROG';

SELECT employee\_id, last\_name, job\_id, salary

FROM employees

WHERE salary = ANY(SELECT salary

FROM employees

WHERE job\_id = 'IT\_PROG')

AND job\_id <> 'IT\_PROG';

* **Create a report that displays a list of all employees whose salary is more than the salary of any employee from department 60.**

SELECT last\_name FROM employees

WHERE salary > ANY (SELECT salary

FROM employees

WHERE department\_id=60);

**ALL**

* >ALL means more than the maximum
* <ALL means less than the minimum

SELECT employee\_id, last\_name, job\_id, salary

FROM employees

WHERE salary < ALL (SELECT salary

FROM employees

WHERE job\_id = 'IT\_PROG')

AND job\_id <> 'IT\_PROG';

**Using the EXISTSO perator**

The EXISTS operator is used in queries where the query result depends on whether or not certain rows exist in a table. It evaluates to TRUE if the subquery returns at least one row.

* Displays departments that have no employees.

SELECT \* FROM departments

WHERE NOT EXISTS (SELECT \* FROM employees

WHERE employees.department\_id=

departments.department\_id);

**Null Values in a Subquery**

SELECT emp.last\_name

FROM employees emp

WHERE emp.employee\_id NOT IN (SELECT mgr.manager\_id

FROM employees mgr);

* Display all employees who do not have any subordinates

SELECT last\_name

FROM employees

WHERE employee\_id NOT IN (SELECT manager\_id

FROM employees

WHERE manager\_id IS NOT NULL);