## Selecting All Columns:

select \*

from departments;

## Selecting Specific Columns:

select department\_id, location\_id

from departments;

select location\_id, department\_id

from departments;

## Column Heading Defaults:

select last\_name, hire\_date, salary

from employees;

## User Arithmetic Operators:

select last\_name, salary, salary + 300

from employees;

## Operators Precedence:

select last\_name, salary, 12 \* salary + 100

from employees;

select last\_name, salary, 12 \* (salary + 100)

from employees;

## Defining a Null Value:

select last\_name, job\_id, salary, commission\_pct

from employees;

## Null Values in Arithmetic Expressions:

select last\_name, 12 \* salary \* commission\_pct

from employees;

## Using Column Aliases:

select last\_name AS name, commission\_pct comm

from employees;

select last\_name AS "Name", salary \* 12 "Annual Salary"

from employees;

## Concatenation Operators:

select last\_name||job\_id AS "Employees"

from employees;

## Using Literal Character Strings:

select last\_name ||' is a '||job\_id AS "Employees Details"

from employees;

select last\_name ||': 1 Month salary = '||salary Monthly

from employees;

## Alternative Quote (q) Operator:

select department\_name ||q' [, it's assigned Manager Id: ] '|| manager\_id AS "Department and Manager"

from departments;

## Duplicate Rows:

select department\_id

from employees;

select DISTINCT department\_id

from employees;

select DISTINCT department\_id, job\_id

from employees;

## Using the Where Clause

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

from employees

where department\_id = 90;

## Character Strings and Dates

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

from employees

where last\_name = 'Whalen';

## Using Comparison Conditions

select last\_name, salary

from employees

where salary <= 3000;

## Using the Between Condition

select last\_name, salary

from employees

where salary between 2500 and 3500;

select last\_name, salary

from employees

where salary not between 2500 and 3500;

select last\_name

from employees

where last\_name between 'King' and 'Smith';

## Using the In Condition

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

from employees

where manager\_id in(100,101,201);

select employee\_id, manager\_id,department\_id

from employees

where last\_name in ('Hartstein', 'Vargas');

select employee\_id, manager\_id,department\_id

from employees

where last\_name not in ('Hartstein', 'Vargas');

## Using the Like Condition

select first\_name

from employees

where first\_name like 'S%';

select last\_name,hire\_date

from employees where hire\_date like '%95';

select last\_name from employees

where last\_name like '\_o%';

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

from employees

where job\_id like '%SA\\_%' escape'\';

select last\_name

from employees

where last\_name not like '%een%';

## Using the null Conditions

select last\_name,manager\_id

from employees

where manager\_id is null;

select last\_name,job\_id,commission\_pct

from employees

where commission\_pct is null;

## Using the And Operator

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

from employees

where salary >= 10000 and job\_id like '%MAN%';

## Using the Or Operator

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

from employees

where salary >= 10000 or job\_id like '%MAN%';

## Using the Not Operator

select last\_name, job\_id

from employees

where job\_id not in('IT\_PROG', 'ST\_CLERK', 'SA\_REF');

select last\_name, job\_id

from employees

where job\_id not in('AC\_ACCOUNT', 'AD\_VP');

select last\_name, job\_id

from employees

where salary not between 10000 and 15000;

## Rules of Precedence

select last\_name, job\_id, salary

from employees

where job\_id = 'SA\_REP' or job\_id = 'AD\_PRES' and salary > 15000;

select last\_name, job\_id, salary

from employees

where (job\_id = 'SA\_REP' or job\_id = 'AD\_PRES') and salary > 15000;

## Using the Order By Clause

select last\_name, job\_id, department\_id, hire\_date

from employees

order by hire\_date;

## Sorting

select last\_name, job\_id, department\_id, hire\_date

from employees

order by hire\_date desc;

select employee\_id, last\_name, salary \* 12 annsal

from employees

order by annsal;

select last\_name, department\_id, salary

from employees

order by department\_id, salary desc;

## Using the & Substitution Variable

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

from employees

where employee\_id = &employee\_num;

## Character and Data Values with Substitution Variables

select last\_name, department\_id, salary\*12

from employees

where job\_id = '&job\_title';

## Specifying Column Names, Expressions, and Text

select employee\_id, last\_name, job\_id, &column\_name

from employees where &condition

order by &order\_column;

## Using the && Substitution Variable

select employee\_id, last\_name, job\_id, &&column\_name

from employees

order by &column\_name;

## Case-Manipulation Functions

select 'The job id for '||upper(last\_name)||' is '||lower(job\_id) as "EMPLOYEE DETAILS"

from employees;

## Using Case-Manipulation Functions

select employee\_id, last\_name, department\_id

from employees

where last\_name = 'higgins';

select employee\_id, last\_name, department\_id

from employees

where lower(last\_name) = 'higgins';

select employee\_id, last\_name, department\_id

from employees

where last\_name = 'Higgins';

## Character-Manipulation Functions

select employee\_id, salary, lpad(salary, 8, 'TK.')

from employees;

select employee\_id, salary, lpad(salary, 8, '\*')

from employees;

select employee\_id, salary, rpad(salary, 8, '\*')

from employees;

select employee\_id, salary, rpad(salary, 8, 'TK.')

from employees;

select employee\_id, first\_name, replace(first\_name, 'a', 'b')

from employees;

select employee\_id, first\_name, trim('a' from first\_name)

from employees;

select employee\_id, first\_name, trim('A' from first\_name)

from employees;

## Using the Character-Manipulation Functions

select employee\_id, concat(first\_name, last\_name) Name, job\_id, length (last\_name), instr(last\_name, 'a') "Contains 'a'?"

from employees

where substr(job\_id, 4) = 'REP';

select employee\_id, concat(first\_name, last\_name) Name, job\_id, length (last\_name), instr(last\_name, 'a') "Contains 'a'?"

from employees

where substr(last\_name, -1, 1) = 'n';

## Using the Round Function

select round(45.923, 2), round(45.923, 0), round(45.923, -1)

from dual;

## Using the Trunc Function

select trunc(45.923, 2), trunc(45.923, 0), trunc(45.923, -1)

from dual;

## Using the Mod Function

select last\_name, salary, mod(salary, 5000)

from employees

where job\_id = 'SA\_REP';

## Working with Dates

select last\_name, hire\_date

from employees

where hire\_date < '01-FEB-88';

## Using Arithmetic Operators with Dates

select last\_name, (sysdate-hire\_date)/7 as weeks

from employees

where department\_id = 90;

## Working with Dates

select last\_name, hire\_date

from employees;

select last\_name, hire\_date

from employees

where hire\_date > '02/07/2007';

select last\_name, hire\_date

from employees

where hire\_date <= '02/07/2007';

select last\_name, hire\_date

from employees

where hire\_date = '02/07/2007';

select sysdate

from dual;

select sysdate + 6

from dual;

select sysdate + 30

from dual;

select sysdate - 30

from dual;

select last\_name, (sysdate - hire\_date) / 7 as months, round((sysdate - hire\_date) / 7) as months from employees;

select last\_name, (sysdate - hire\_date) / 30 as days, round((sysdate - hire\_date) / 30) as days from employees;

## Using Date Functions

select last\_name, hire\_date, months\_between('06/17/2003', '08/16/2002')

from employees;

select last\_name, hire\_date, add\_months('06/17/2003', 9)

from employees;

select last\_name, hire\_date, next\_day('06/17/2003', 'friday')

from employees;

select last\_name, hire\_date, last\_day('06/17/2003')

from employees;

## Using the To\_Char Function with Dates

select last\_name, to\_char(hire\_date, 'MM/YY') MONTHS\_HIRED

from employees;

select last\_name, to\_char(hire\_date, 'dd " of " month') MONTHS\_HIRED

from employees;

select last\_name, to\_char(hire\_date, 'hh24:mi:ss am') MONTHS\_HIRED

from employees;

## Using the To\_Char Function with Dates

select last\_name, to\_char(hire\_date, 'fmdd month yyyy') as HIREDATE

from employees;

## Using To\_Char Function with Numbers

select last\_name, to\_char(salary \* 100, '$99,99,999.00') SALARY

from employees;

## Using the To\_Number and To\_Date Functions

select last\_name, hire\_date

from employees

where hire\_date = to\_date('May 24, 2007', 'fmMonth dd, yyyy');

## RR Date Formate: Example

select last\_name, to\_char(hire\_date, 'dd-Mon-yyyy')

from employees

where hire\_date < to\_date('01-Jan-2008', 'dd-Mon-rr');

select last\_name, to\_char(hire\_date, 'day/mon/yyyy')

from employees

where hire\_date = to\_date('Jun 17, 2003', 'fmMonth dd, yyyy');

## Nesting Functions

select last\_name, upper(concat(substr(last\_name, 1, 3), '\_US'))

from employees

where department\_id = 60;

select last\_name, substr(last\_name, 1, 3), concat(substr(last\_name, 1, 3), '\_US'), upper(concat(substr(last\_name, 1, 3), '\_US'))

from employees

where department\_id = 60;

select to\_char(next\_day(add\_months(hire\_date, 6), 'Friday'), 'fmDay, Month DDth, YYYY') "Next 6 Month Review"

from employees

order by hire\_date;

## Using the NVL Function

select last\_name, salary, nvl(commission\_pct, 0), (salary \* 12) + (salary \* 12 \* nvl(commission\_pct, 0)) AN\_SAL

from employees;

select last\_name, salary, commission\_pct, (salary \* 12) + (salary \* 12 \* commission\_pct) AN\_SAL from employees;

## Using the NVL2 Function

select last\_name, salary, commission\_pct, nvl2(commission\_pct, 'SAL + COMM', 'SAL') income

from employees where department\_id in (50, 80);

select last\_name, salary, commission\_pct, nvl2(commission\_pct, salary + commission\_pct, salary) as TOTAL

from employees where department\_id in (50, 80);

## Using the NullIf Function

select first\_name, length(first\_name) "expr1", last\_name, length(last\_name) "expr2", nullif(length(first\_name), length(last\_name)) result

from employees;

## Using the Case Expression

select last\_name, job\_id, salary,

case job\_id when 'IT\_PROG' then 1.10 \* salary

when 'IT\_PROG' then 1.15 \* salary

when 'IT\_PROG' then 1.20 \* salary

else salary end "revised\_salary"

from employees;

select last\_name, salary,

(case when salary < 5000 then 'LOW'

when salary < 10000 then 'MEDIUM'

when salary < 20000 then 'GOOD'

else 'EXCELLENT'

end)qualified\_salary

from employees;

## Using the Decode Function

select last\_name, job\_id, salary,

decode (job\_id, 'IT\_PROG', 1.10 \* salary,

'ST\_CLERK', 1.15 \* salary,

'SA\_REP', 1.20 \* salary,

salary)

"revised\_salary"

from employees;

select last\_name, salary,

decode(trunc(salary / 2000, 0),

0, 0.00,

1, 0.09,

2, 0.20,

3, 0.30,

4, 0.40,

5, 0.42,

6, 0.44,

0.45) tax\_rate

from employees

where department\_id = 80;

## Using the Avg and Sum Functions

select round(avg(salary)), max(salary),

min(salary), sum(salary)

from employees

where job\_id like '%REP%';

## Using the Min and Max Functions

select min(hire\_date), max(hire\_date)

from employees;

select min(last\_name), max(last\_name)

from employees;

## Using the Count function

select count(\*)

from employees

where department\_id = 50;

select count(commission\_pct)

from employees

where department\_id = 80;

## Group Functions and Null Values

select avg(commission\_pct)

from employees;

select avg(nvl(commission\_pct, 0))

from employees;

## Using the Group by Clause

select department\_id, avg(salary)

from employees

group by department\_id;

select avg(salary)

from employees

group by department\_id;

select department\_id, avg(salary)

from employees

group by department\_id

order by avg(salary);

## Using the Group by Clause on Multiple Columns

select department\_id dept\_id, job\_id, sum(salary)

from employees

group by department\_id, job\_id;

## Illigal Quaries Using Group Functions

select department\_id, count(last\_name)

from employees;

select department\_id, count(last\_name)

from employees

group by department\_id;

select department\_id, avg(salary)

from employees

where avg(salary) > 8000

group by department\_id;

select department\_id, avg(salary)

from employees

having avg(salary) > 8000

group by department\_id;

## Restricting Group Results with the Having Clause

select department\_id, max(salary)

from employees

group by department\_id

having max(salary) > 10000;

select department\_id, avg(salary)

from employees

group by department\_id

having max(salary) > 10000;

## Using Having Clause

select job\_id, sum(salary) payroll

from employees

where job\_id not like '%REP%'

group by job\_id

having sum(salary) > 13000

order by sum(salary);

## Nesting Group Functions

select max(avg(salary))

from employees

group by department\_id;

## Retrieving Records with Natural Joins

select department\_id, department\_name, location\_id, city

from departments

natural join locations;

select department\_id, department\_name, location\_id, city

from departments

natural join locations

where department\_id in (20, 50);

## Creating Joins with the Using Clause

select l.city, d.department\_name

from locations l join departments d using (location\_id)

where location\_id = 1400;

select l.city, d.department\_name

from locations l join departments d using (location\_id)

where d.location\_id = 1400;

ORA-25154: column part of USING clause cannot have qualifier

## Retrieving Records with the Using Clause

select employees.employee\_id, employees.last\_name, departments.location\_id, department\_id

from employees join departments

using (department\_id);

## Qualifying Ambiguous Column Names

select employees.employee\_id, employees.last\_name, departments.department\_id, departments.location\_id

from employees join departments

on employees.department\_id = departments.department\_id;

## Using Table Aliases

select e.employee\_id, e.last\_name, d.location\_id, department\_id

from employees e join departments d

using (department\_id);

## Retrieving Records with the On clause

select e.employee\_id, e.last\_name, e.department\_id, d.department\_id, d.location\_id

from employees e join departments d

on (e.department\_id = d.department\_id);

## Self-joins Using the On Clause

select e.last\_name emp, m.last\_name mgr

from employees e join employees m

on (e.manager\_id = m.employee\_id);

## Applying Additional Additional Conditions to a Join

select e.employee\_id, e.last\_name, e.department\_id, d.department\_id, d.location\_id

from employees e join departments d

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

and e.manager\_id = 149;

select e.employee\_id, e.last\_name, e.department\_id, d.department\_id, d.location\_id

from employees e join departments d

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

where e.manager\_id = 149;

## Creating Three-Way Joins with the On Clause

select employee\_id, city, department\_name

from employees e

join departments d

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

join locations l

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

## Retriving Records with Nonequijoins

select e.last\_name, e.salary, j.job\_title

from employees e join jobs j

on e.salary between j.min\_salary and j.max\_salary;

## Left Outer Join

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

from employees e left outer join departments d

on (e.department\_id = d.department\_id);

## Right Outer Join

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

from employees e right outer join departments d

on (e.department\_id = d.department\_id);

## Full Outer Join

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

from employees e full outer join departments d

on (e.department\_id = d.department\_id);