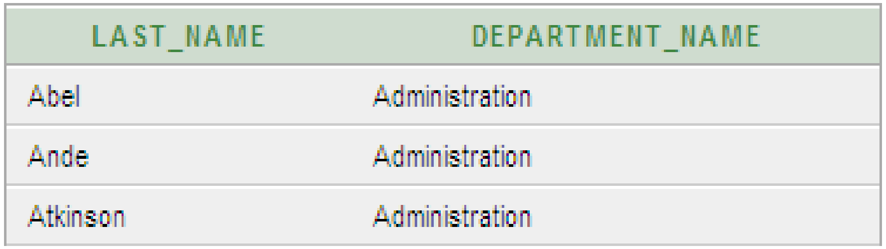
ORACLE®

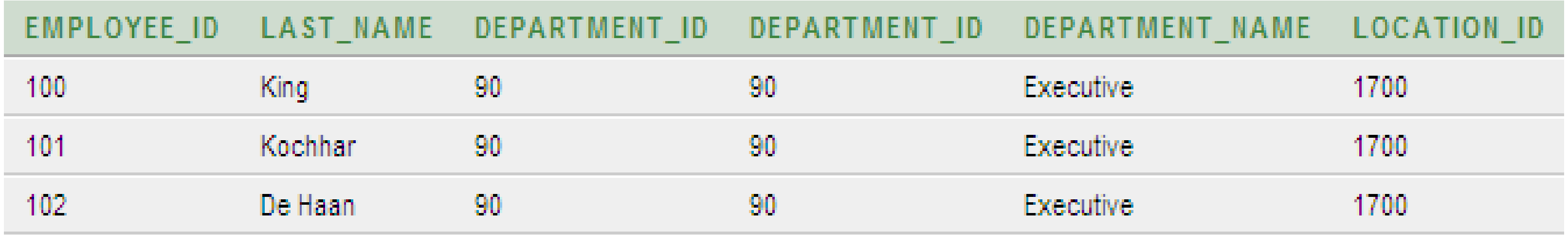
Displaying Data From Tables (Equi Join)

```
SELECT last_name,department_name
FROM employees,departments
```



LAST_NAME	DEPARTMENT_NAME
Abel	Administration
Ande	Administration
Atkinson	Administration

```
SELECT employees.employee_id,employees.last_name,
employees.department_id,departments.department_id,
departments.department_name, departments.location_id
FROM employees,departments
WHERE employees.department_id=departments.department_id
```

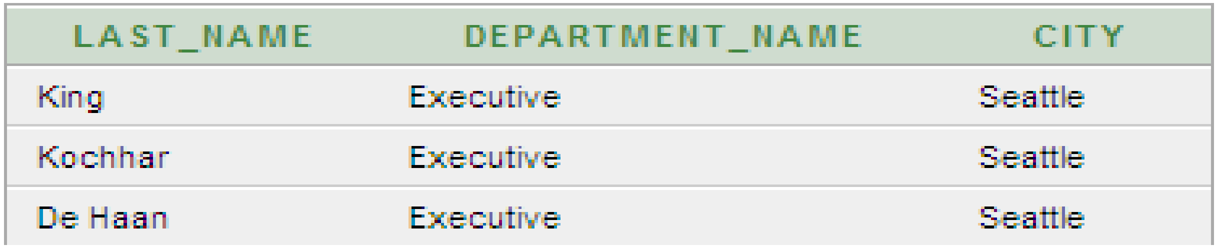


EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
100	King	90	90	Executive	1700
101	Kochhar	90	90	Executive	1700
102	De Haan	90	90	Executive	1700

Equi Join

```
SELECT e.employee_id,e.last_name, e.department_id,d.department_id,
d.department_name, d.location_id
FROM employees e,departments d
WHERE e.department_id=d.department_id

SELECT e.last_name, d.department_name, i.city
FROM employees e,departments d,locations i
WHERE e.department_id=d.department_id AND d.location_id=i.location_id
```

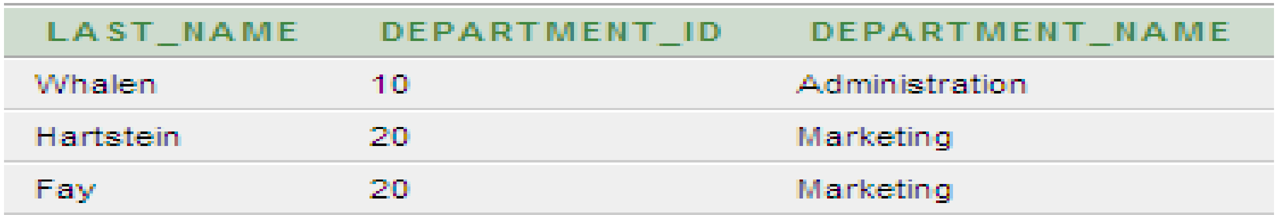


LAST_NAME	DEPARTMENT_NAME	CITY
King	Executive	Seattle
Kochhar	Executive	Seattle
De Haan	Executive	Seattle

Equi-Join: A special case of condition join where the condition c contains only **equalities.**

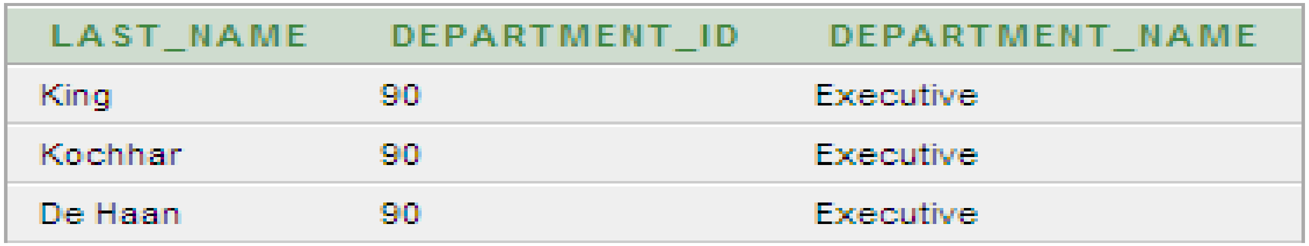
Outer-joins

SELECT e.last_name, e.department_id,d.department_name
FROM employees e,departments d
WHERE e.department_id(+)=d.department_id وەکو right outer join



LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing

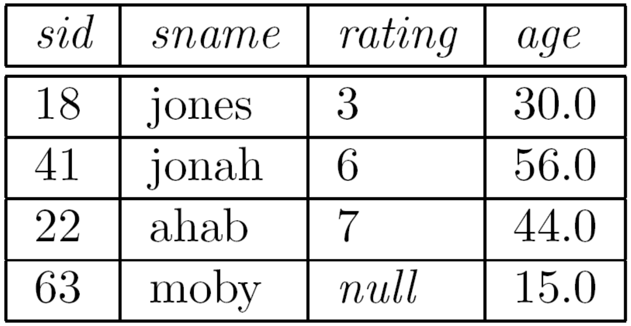
SELECT e.last_name, e.department_id,d.department_name
FROM employees e,departments d
WHERE e.department_id=d.department_id(+) وەکو Left outer join



LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
King	90	Executive
Kochhar	90	Executive
De Haan	90	Executive

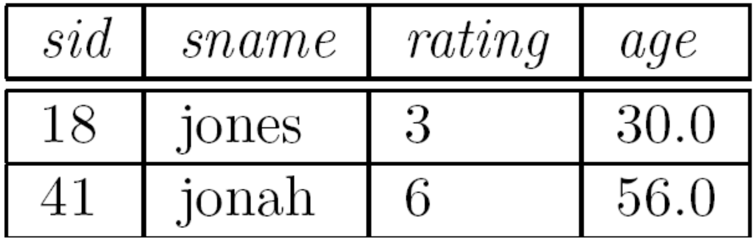
ئەوەی هێمای + لەپێشە مانای دەرەکیە وە ئەوەی تر سەرەکیە

Consider the instance of Sailors shown in Figure 5.1. Let us define instance S1 of Sailors to consist of the first two tuples, instance S2 to be the last two tuples, and S to be the given instance.



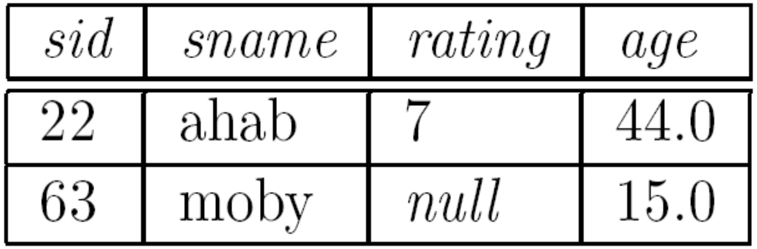
sid	*sname*	*rating*	*age*
18	jones	3	30.0
41	jonah	6	56.0
22	ahab	7	44.0
63	moby	*null*	15.0

Figure 5.1 An Instance of Sailors



sid	*sname*	*rating*	*age*
18	jones	3	30.0
41	jonah	6	56.0

S1



sid	*sname*	*rating*	*age*
22	ahab	7	44.0
63	moby	*null*	15.0

S2

(d) Show the left outer join of S1 with S2, with the join condition being *sid=sid*.

(e) Show the right outer join of S1 with S2, with the join condition being *sid=sid*.

(f) Show the full outer join of S1 with S2, with the join condition being *sid=sid*.

(d)

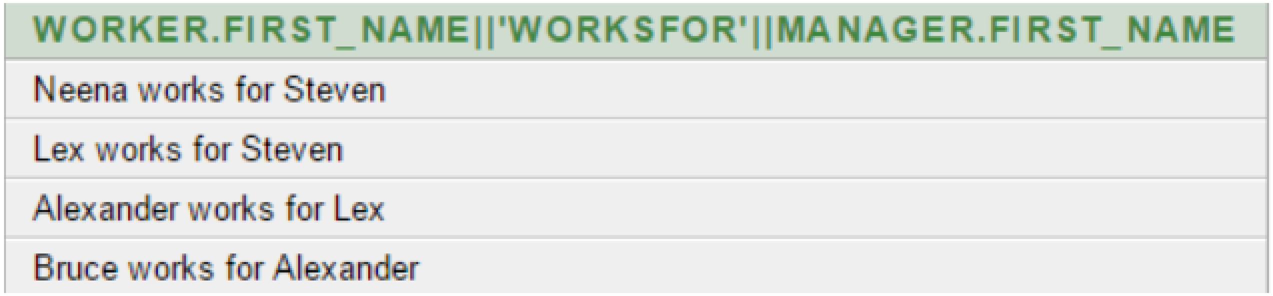
sid	*sname*	*rating*	*age*	*sid*	*sname*	*rating*	*age*
18	jones	3	30.0	*null*	*null*	*null*	*null*
41	jonah	6	56.0	*null*	*null*	*null*	*null*

(f)

sid	*sname*	*rating*	*age*	*sid*	*sname*	*rating*	*age*
18	jones	3	30.0	*null*	*null*	*null*	*null*
41	jonah	6	56.0	*null*	*null*	*null*	*null*
null	*null*	*null*	*null*	22	ahab	7	44.0
null	*null*	*null*	*null*	63	moby	*null*	15.0

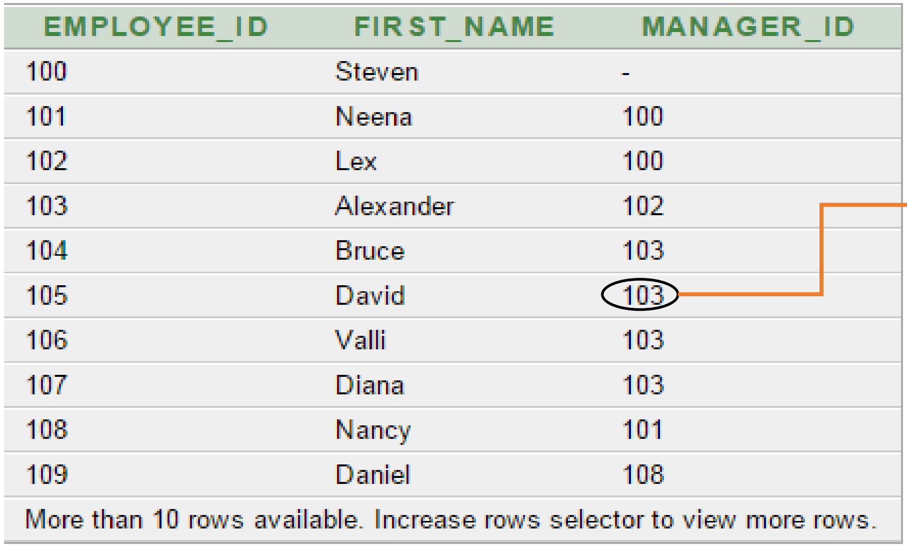
Self Join

```
SELECT worker.First_name || ' works for ' || manager. First_name
FROM employees worker, employees manager
WHERE worker.manager_id= manager.employee_id
```



WORKER.FIRST_NAME||'WORKSFOR'||MANAGER.FIRST_NAME
Neena works for Steven
Lex works for Steven
Alexander works for Lex
Bruce works for Alexander

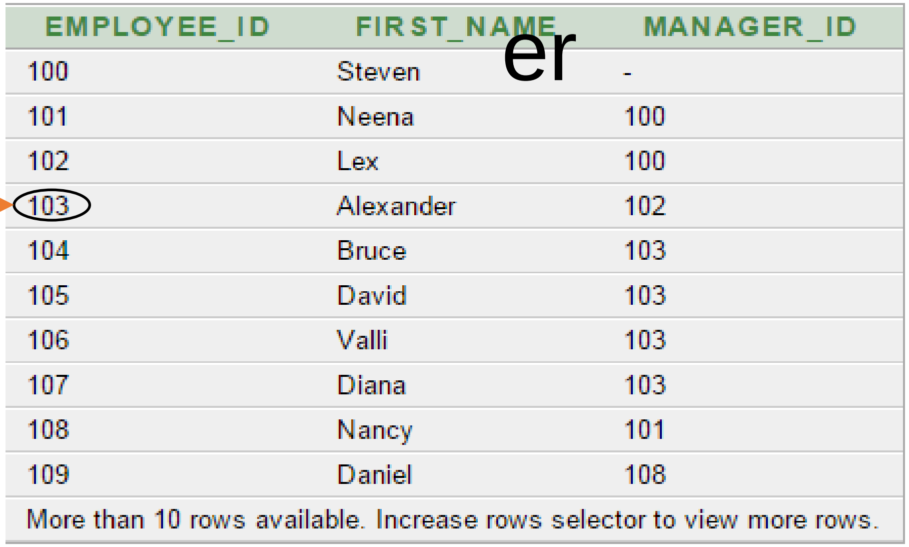
worker



EMPLOYEE_ID	FIRST_NAME	MANAGER_ID
100	Steven	-
101	Neena	100
102	Lex	100
103	Alexander	102
104	Bruce	103
105	David	103
106	Valli	103
107	Diana	103
108	Nancy	101
109	Daniel	108

More than 10 rows available. Increase rows selector to view more rows.

manager

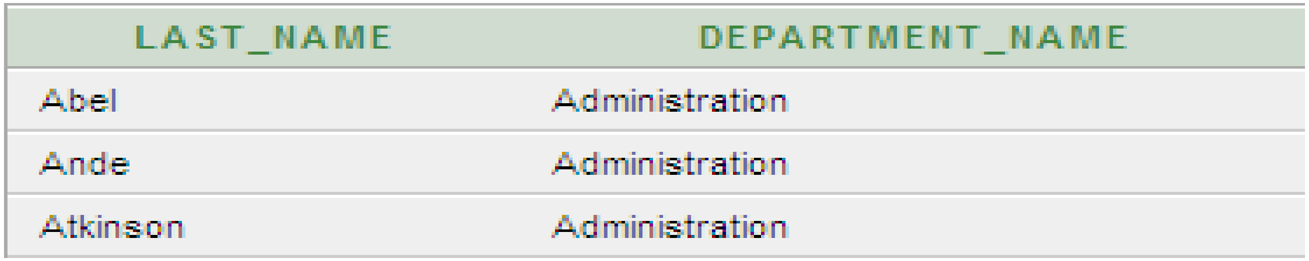


EMPLOYEE_ID	FIRST_NAME	MANAGER_ID
100	Steven	-
101	Neena	100
102	Lex	100
103	Alexander	102
104	Bruce	103
105	David	103
106	Valli	103
107	Diana	103
108	Nancy	101
109	Daniel	108

More than 10 rows available. Increase rows selector to view more rows.

CROSS JOIN

```
SELECT last_name,department_name
FROM employees
CROSS JOIN departments
```

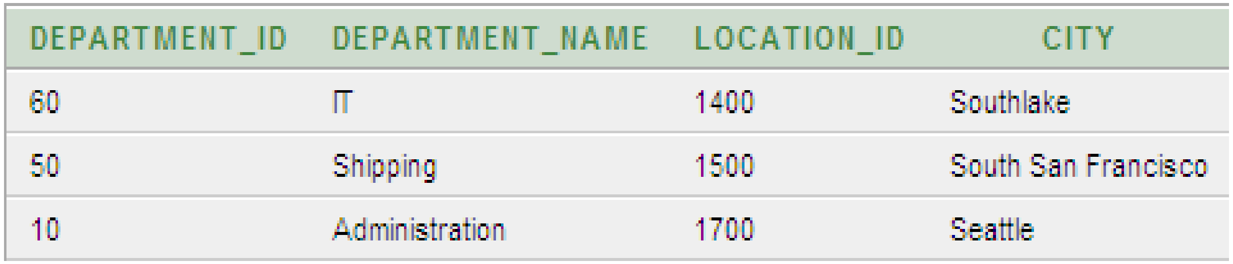


LAST_NAME	DEPARTMENT_NAME
Abel	Administration
Ande	Administration
Atkinson	Administration

Cross join is a Cartesians product

Natural Join

```
SELECT department_id,department_name,location_id,city
FROM departments natural join locations
```

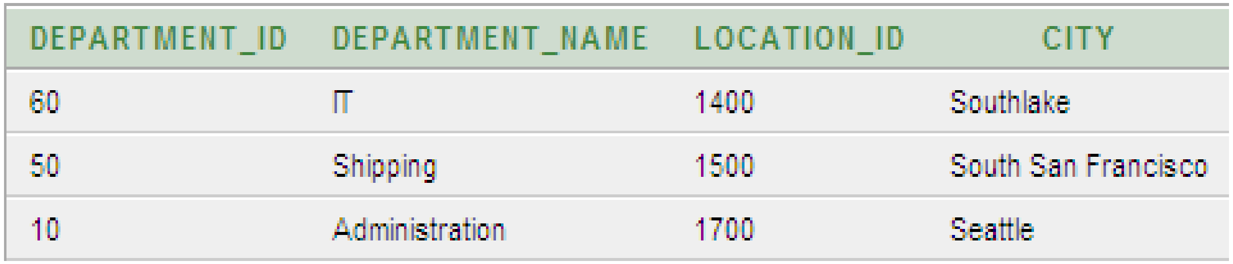


DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
60	IT	1400	Southlake
50	Shipping	1500	South San Francisco
10	Administration	1700	Seattle

Locations is another relation has location_id so natural join will join this attrib. with that of departments

```
SELECT departments.department_id,departments.department_name,locations.location_id,locations.city
FROM departments,locations
WHERE departments.location_id=locations.location_id
```

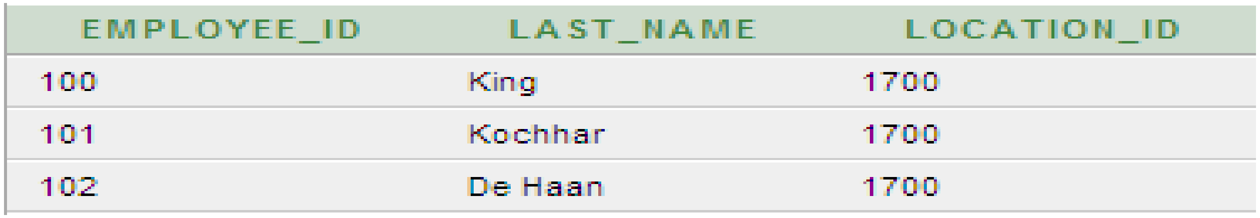
Any how join has to be put after FROM



DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
60	IT	1400	Southlake
50	Shipping	1500	South San Francisco
10	Administration	1700	Seattle

USING

```
SELECT e.employee_id,e.last_name,d.location_id
FROM employees e join departments d using(department_id)
```



EMPLOYEE_ID	LAST_NAME	LOCATION_ID
100	King	1700
101	Kochhar	1700
102	De Haan	1700

ON

```
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)
```

```
SELECT e.last_name emp,m.last_name mgr
FROM employees e join employees m ON (e.manager_id=m.employee_id)
```

```
SELECT employee_id,department_name,city
FROM employees e
JOIN departments d ON (e.department_id=d.department_id)
JOIN locations l ON (d.location_id=l.location_id)
```

LEFT, RIGHT,FULL 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)
```

```
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)
```

```
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)
```

Excercise

```
SELECT e.last_name,e.department_id,d.department_name
FROM employees e, departments d
WHERE e.department_id=d.department_id
```

Left join=full join=this one

```
SELECT  DISTINCT job_id, location_id
FROM employees e, departments d
WHERE e.department_id=d.department_id AND d.department_id =80
```

```
SELECT e.last_name,e.job_id,e.department_id,d.department_name
FROM employees e JOIN  departments d ON (
e.department_id=d.department_id)
JOIN locations l ON (d.location_id=l.location_id)
WHERE LOWER(l.city)='toronto'
```

Excercise

```
SELECT w.last_name "Employees", w.employee_id "Emp#",m.last_name
"Manager", m.employee_id "Mgr#"
FROM employees w JOIN employees m ON (w.manager_id=m.employee_id)
```

```
SELECT w.last_name "Employees", w.employee_id "Emp#",m.last_name
"Manager", m.employee_id "Mgr#"
FROM employees w LEFT OUTER JOIN employees m ON
(w.manager_id=m.employee_id)
```

```
SELECT e.department_id department,e.last_name employee,c.last_name
colleague
FROM employees e JOIN  employees c ON (
e.department_id=c.department_id)
WHERE e.employee_id<>c.employee_id
ORDER BY e.department_id,e.last_name,c.last_name
```

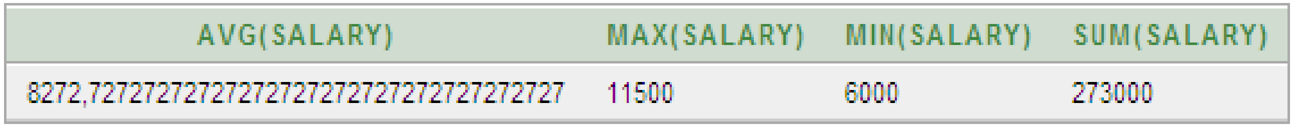
Excercise

```
SELECT e.last_name employee,e.hire_date
FROM employees e , employees davies
WHERE davies.hire_date<e.hire_date
AND davies.last_name='Davies'

 SELECT w.last_name,w.hire_date,m.last_name,m.hire_date
FROM employees w , employees m
WHERE w.manager_id=m.employee_id
AND w.hire_date<m.hire_date
```

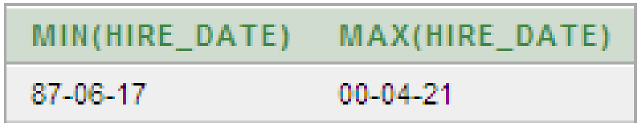
Aggregation

```
SELECT AVG(salary),MAX(salary),MIN(salary),SUM(salary)
FROM employees
WHERE job_id LIKE '%REP%'
```



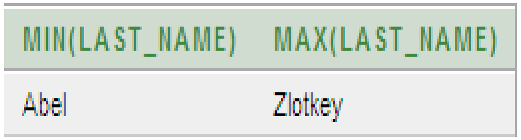
AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
8272,72727272727272727272727272727272727	11500	6000	273000

```
SELECT MIN(hire_date),MAX(hire_date)
FROM employees
```



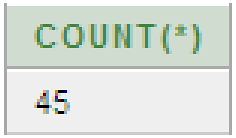
MIN(HIRE_DATE)	MAX(HIRE_DATE)
87-06-17	00-04-21

```
SELECT MIN(last_name),MAX(last_name)
FROM employees
```



MIN(LAST_NAME)	MAX(LAST_NAME)
Abel	Zlotkey

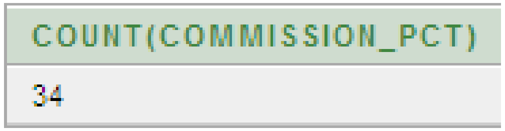
```
SELECT COUNT(*)
FROM employees
WHERE department_id=50
```



COUNT(*)
45

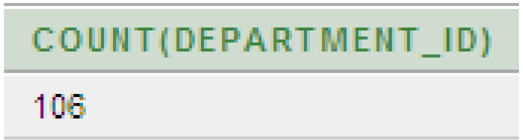
Aggregation

```
SELECT COUNT(commission_pct)
FROM employees
WHERE department_id=80
```



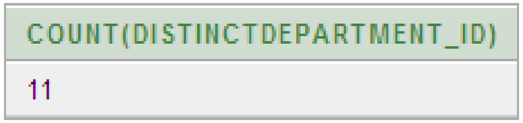
COUNT(COMMISSION_PCT)
34

```
SELECT COUNT(department_id)
FROM employees
```



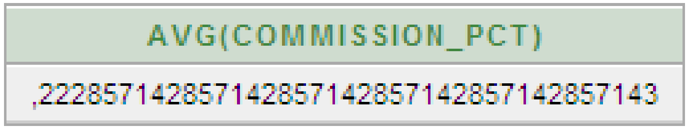
COUNT(DEPARTMENT_ID)
106

```
SELECT COUNT(DISTINCT department_id)
FROM employees
```



COUNT(DISTINCTDEPARTMENT_ID)
11

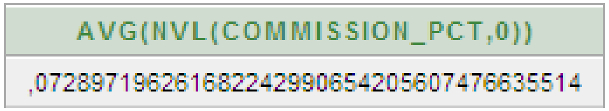
```
SELECT AVG(commission_pct)
FROM employees
```



AVG(COMMISSION_PCT)
,222857142857142857142857142857143

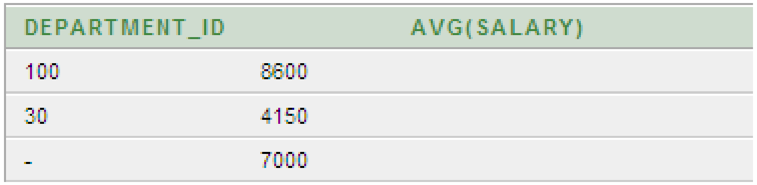
Aggregation

```
SELECT AVG(NVL(commission_pct,0))
FROM employees
```



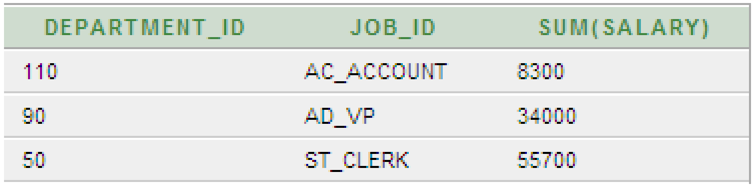
AVG(NVL(COMMISSION_PCT,0))
,072897196261682242990654205607476635514

```
SELECT department_id,AVG(salary)
FROM employees
GROUP BY department_id
```



DEPARTMENT_ID	AVG(SALARY)
100	8600
30	4150
-	7000

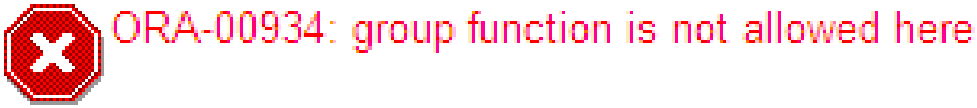
```
SELECT department_id,job_id,SUM(salary)
FROM employees
GROUP BY department_id,job_id
```



DEPARTMENT_ID	JOB_ID	SUM(SALARY)
110	AC_ACCOUNT	8300
90	AD_VP	34000
50	ST_CLERK	55700

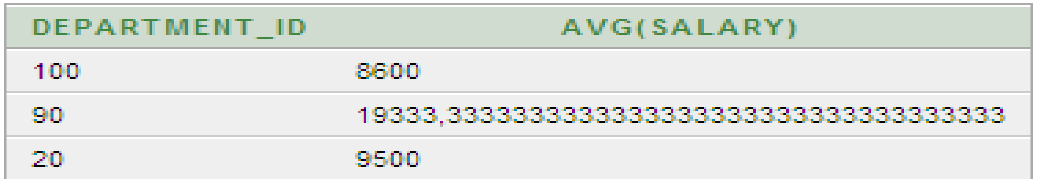
Aggregation

```
SELECT department_id,AVG(salary)
FROM employees
WHERE AVG(salary)>8000
GROUP BY department_id
```



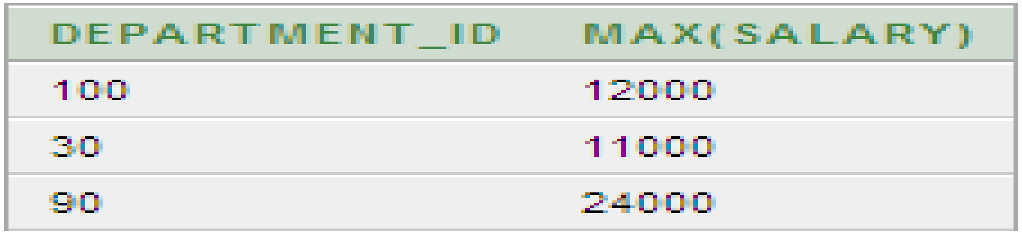
ORA-00934: group function is not allowed here

```
SELECT department_id,AVG(salary)
FROM employees
HAVING AVG(salary)>8000
GROUP BY department_id
```



DEPARTMENT_ID	AVG(SALARY)
100	8600
90	19333,33333333333333333333333333333333333333
20	9500

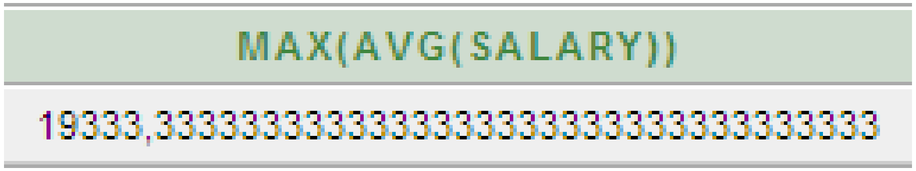
```
SELECT department_id,MAX(salary)
FROM employees
GROUP BY department_id
HAVING MAX(salary)>10000
```



DEPARTMENT_ID	MAX(SALARY)
100	12000
30	11000
90	24000

Aggregation

```
SELECT MAX(sum(salary))
FROM employees
GROUP BY department_id
```



MAX(AVG(SALARY))
19333,33333333333333333333333333333333333

Excercise

```
SELECT job_id,COUNT(*)
FROM employees
GROUP BY job_id

SELECT COUNT(DISTINCT manager_id) "Number of Managers"
FROM employees

SELECT MAX(salary)-MIN(salary) DIFFERENCE
FROM employees

SELECT manager_id,MIN(salary)
FROM employees
WHERE manager_id IS NOT NULL
GROUP BY manager_id
HAVING MIN(salary)>6000
ORDER BY MIN(salary) DESC
```

Excercise

```
SELECT d.department_name "Name",d.location_id "Location",COUNT(*) "Number of people", ROUND(AVG(e.salary),2) "Salary"
FROM employees e, departments d
WHERE e.department_id=d.department_id
GROUP BY d.department_name,d.location_id
```