



STRUCTURAL QUERY LANGUAGE

Oracle Structural Query Language

لغة الاستعلام SQL :

هي عبارة عن مجموعة من الاوامر التي يحتاجها المبرمجين وكذلك المستخدمين للوصول للبيانات الموجودة ضمن قاعدة أوراكل.

تم تطوير هذه اللغة في البداية من قبل شركة IBM وذلك في منتصف السبعينات ثم قامت شركة اوراكل بتطويرها . أوراكل (Oracle) هي عبارة عن نظام لإدارة قواعد البيانات العلائقية، وقد قامت شركة أوراكل بتطويرها في عام 1980، وتُعتبر أوراكل (Oracle) أول قاعدة بيانات مصممة للحوسبة الشبكية (Grid Computing) ، وتساعد أوراكل على توفير أكثر الطرق مرونة وفعالية من حيث التكلفة لإدارة المعلومات والتطبيقات، وأهم ما يميزها أنها تعمل على منصات وأنظمة التشغيل الرئيسية مثل (Windows) ،: (Unix)، لينكس، (macOS)،

تعد أوراكل قاعدة بيانات ارتباطية يتم من خلالها الوصول إلى البيانات بواسطة المستخدم من خلال لغة التطبيق أو الاستعلام التي تسمى SQL .

، وخادم قاعدة البيانات (Database Server) هو المفتاح لحل مشاكل إدارة المعلومات، وبشكل عام يقوم الخادم بإدارة كمية كبيرة من البيانات بشكل موثوق به في بيئة متعددة المستخدمين؛ بحيث يتمكن العديد من المستخدمين من الوصول إلى نفس البيانات في نفس الوقت، ويتم إنجاز كل هذا مع تقديم أداء عالٍ كما ويمنع خادم قاعدة البيانات الوصول غير المصرح به ويوفر حلاً فعالاً في حال حدوث أي خطأ، وتعد الذاكرة والعمليات التي يتم تنفيذها على الملفات هي المكونات الأساسية لقواعد البيانات



يمكن التعامل مع قاعدة البيانات من خلال استخدام SQL حيث يمكن :

- انشاء الجداول
- تعديل البيانات
- حذف الجداول
- اضافة البيانات الى الجدول.
- التعديل على البيانات المدخلة.
- البحث عن البيانات.



SELECT STATEMENT

- الشكل العام لجملّة select كما يلي:

```
SELECT *|{[DISTINCT] column [alias],...}  
FROM table;
```

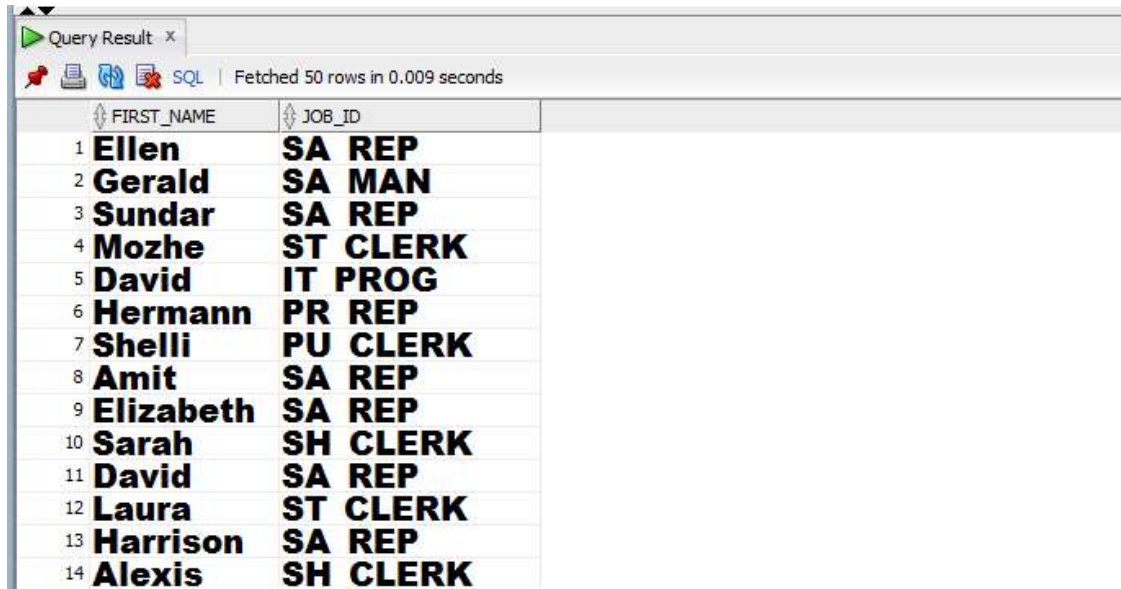
<i>SELECT</i>	Is a list of one or more columns
*	Selects all columns
<i>DISTINCT</i>	Suppresses duplicates
<i>column/expression</i>	Selects the named column or the expression
<i>alias</i>	Gives different headings to the selected columns
<i>FROM table</i>	Specifies the table containing the columns

على سبيل المثال :

```
SELECT department_id, location_id  
FROM departments;
```

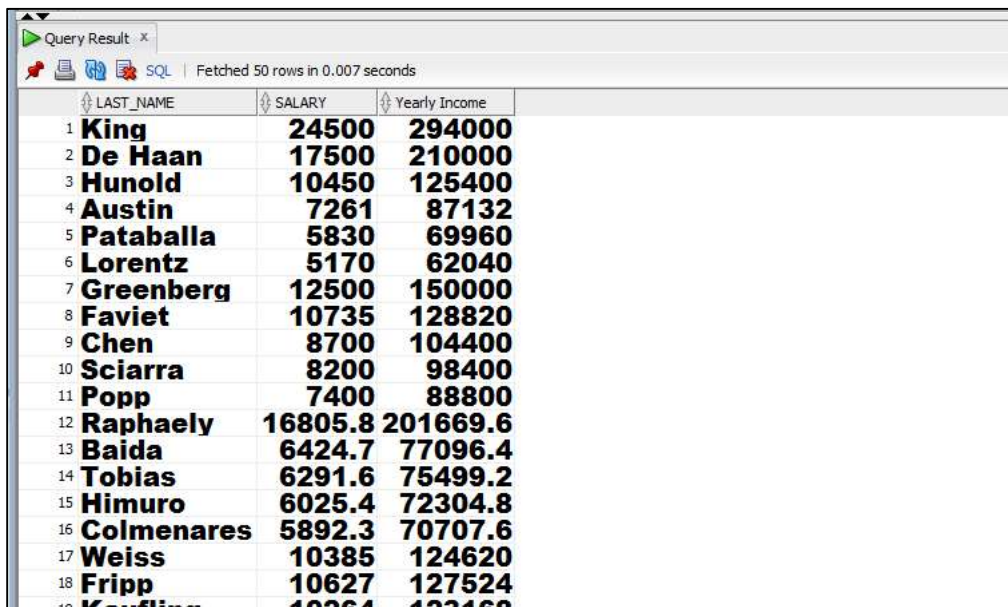
Example :

```
Select FIRST_NAME ,job_id
from employees;
```



	FIRST_NAME	JOB_ID
1	Ellen	SA REP
2	Gerald	SA MAN
3	Sundar	SA REP
4	Mozhe	ST CLERK
5	David	IT PROG
6	Hermann	PR REP
7	Shelli	PU CLERK
8	Amit	SA REP
9	Elizabeth	SA REP
10	Sarah	SH CLERK
11	David	SA REP
12	Laura	ST CLERK
13	Harrison	SA REP
14	Alexis	SH CLERK

```
SELECT last_name, salary,
salary *12 as "Yearly Income"
FROM employees;
```



	LAST_NAME	SALARY	Yearly Income
1	King	24500	294000
2	De Haan	17500	210000
3	Hunold	10450	125400
4	Austin	7261	87132
5	Pataballa	5830	69960
6	Lorentz	5170	62040
7	Greenberg	12500	150000
8	Faviet	10735	128820
9	Chen	8700	104400
10	Sciarra	8200	98400
11	Popp	7400	88800
12	Raphaely	16805.8	201669.6
13	Baida	6424.7	77096.4
14	Tobias	6291.6	75499.2
15	Himuro	6025.4	72304.8
16	Colmenares	5892.3	70707.6
17	Weiss	10385	124620
18	Fripp	10627	127524
19	Kaufling	10264	123168

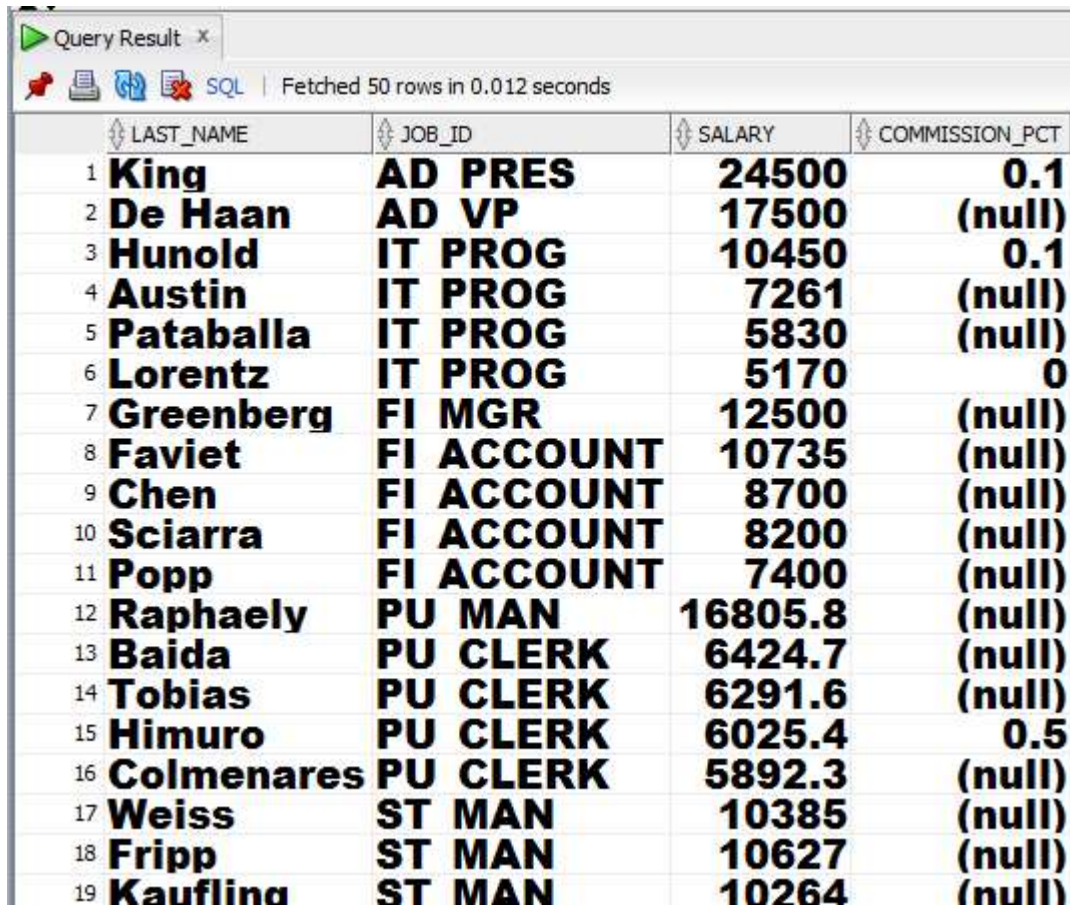
استخدام Null :

- قيمة غير معلومة وغير معرفة.
- لا تساوي "صفر" ولا تعتبر مسافة.

Example :

```

SELECT last_name, job_id, salary, commission_pct
FROM employees;
  
```



Query Result x

SQL | Fetched 50 rows in 0.012 seconds

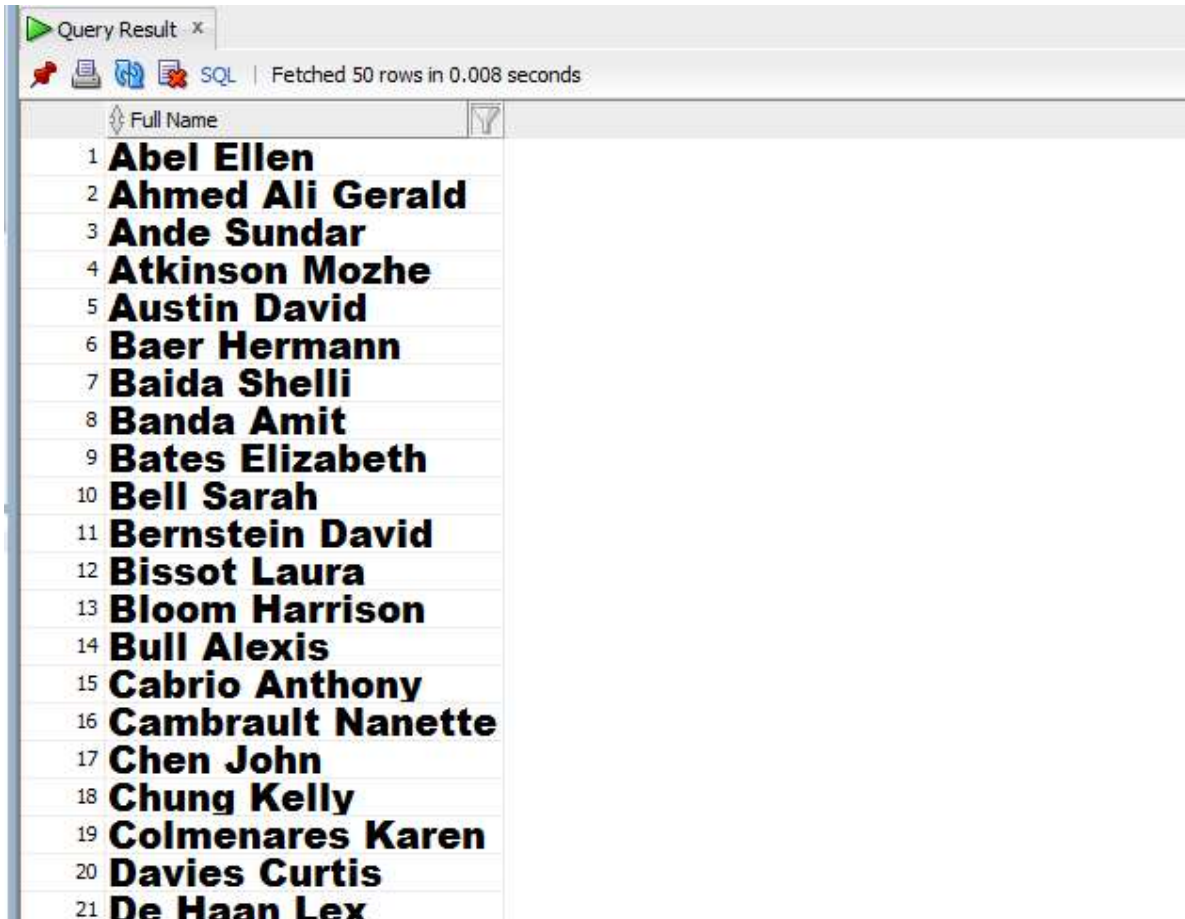
	LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
1	King	AD PRES	24500	0.1
2	De Haan	AD VP	17500	(null)
3	Hunold	IT PROG	10450	0.1
4	Austin	IT PROG	7261	(null)
5	Pataballa	IT PROG	5830	(null)
6	Lorentz	IT PROG	5170	0
7	Greenberg	FI MGR	12500	(null)
8	Faviet	FI ACCOUNT	10735	(null)
9	Chen	FI ACCOUNT	8700	(null)
10	Sciarra	FI ACCOUNT	8200	(null)
11	Popp	FI ACCOUNT	7400	(null)
12	Raphaely	PU MAN	16805.8	(null)
13	Baida	PU CLERK	6424.7	(null)
14	Tobias	PU CLERK	6291.6	(null)
15	Himuro	PU CLERK	6025.4	0.5
16	Colmenares	PU CLERK	5892.3	(null)
17	Weiss	ST MAN	10385	(null)
18	Fripp	ST MAN	10627	(null)
19	Kaufling	ST MAN	10264	(null)

استخدام Concatenation Operator :

- يربط اعمدة او حروف معاً.
- يمثل هكذا ||.
- التعبير الناتج يكون من نوعية Character.

Example :

```
SELECT last_name || ' ' || first_name as "Full Name"  
FROM employees;
```



Query Result x

SQL | Fetched 50 rows in 0.008 seconds

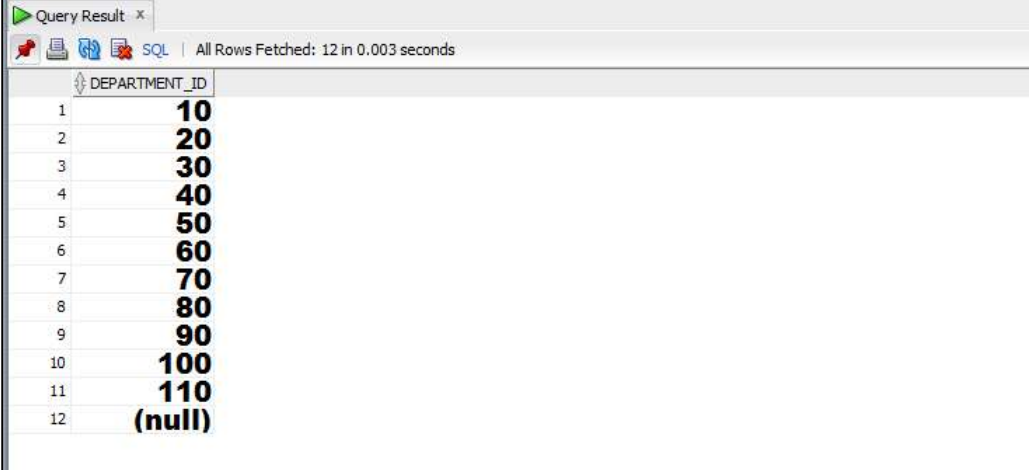
	Full Name
1	Abel Ellen
2	Ahmed Ali Gerald
3	Ande Sundar
4	Atkinson Mozhe
5	Austin David
6	Baer Hermann
7	Baida Shelli
8	Banda Amit
9	Bates Elizabeth
10	Bell Sarah
11	Bernstein David
12	Bissot Laura
13	Bloom Harrison
14	Bull Alexis
15	Cabrio Anthony
16	Cambraut Nanette
17	Chen John
18	Chung Kelly
19	Colmenares Karen
20	Davies Curtis
21	De Haan Lex

استخدام DISTINCT :

- يستخدم لعدم تكرار البيانات في الأعمدة المستخدمة.

Example :

```
SELECT DISTINCT department_id
FROM employees
ORDER By department_id;
```

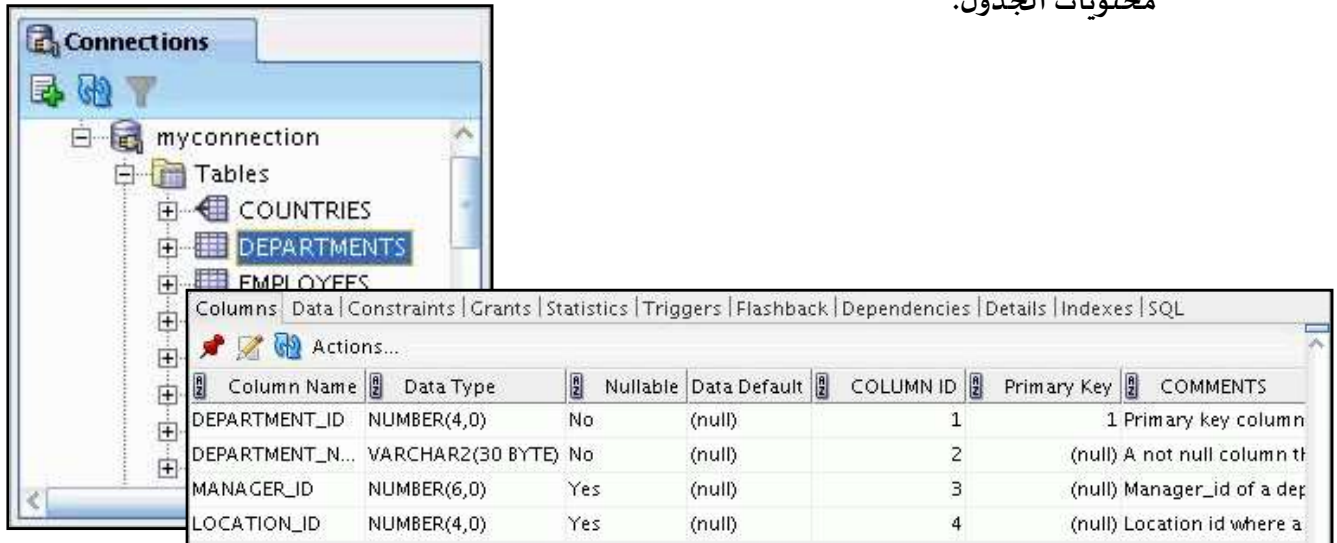


The screenshot shows a SQL query result window titled "Query Result". The window displays the results of a query that selects distinct department IDs from the employees table, ordered by department ID. The results are as follows:

	DEPARTMENT_ID
1	10
2	20
3	30
4	40
5	50
6	60
7	70
8	80
9	90
10	100
11	110
12	(null)

أمر DESCRIBE :

- يستخدم لعرض مكونات الجدول.
- يمكن معرفة مكونات الجدول باختياره من Connections Tree ثم استخدام Columns tab لرؤية محتويات الجدول.



DESCRIBE employees

```

DESCRIBE Employees
Name                Null              Type
-----
EMPLOYEE_ID         NOT NULL         NUMBER(6)
FIRST_NAME
LAST_NAME           NOT NULL         VARCHAR2(25)
EMAIL               NOT NULL         VARCHAR2(25)
PHONE_NUMBER
HIRE_DATE           NOT NULL         DATE
JOB_ID              NOT NULL         VARCHAR2(10)
SALARY
COMMISSION_PCT      NUMBER(2,2)
MANAGER_ID          NUMBER(6)
DEPARTMENT_ID       NUMBER(4)
  
```

استخدام WHERE :

- يمكن تقييد نتائج عبارة select عن طريق إضافة شرط على العبارة كما يلي:

```
Select * |{ [distinct] column ,expression  
                [alias] ,-----}  
From table_name  
[where conditions] ;
```

• حيث يتم إضافة العبارة Where إلى عبارة select بعد from.

• أمثلة:

استعلام يعيد بيانات الموظفين الذين يعملون في القسم رقم 20.

استعلام يعيد راتب الموظف smith.

• صيغ ومعاملات المقارنة في Oracle:

○ يمكن استخدام مجموعة من معاملات المقارنة ضمن عبارة where مثل:

▪ <>,<=>,>=,<>

▪ [not] Between-----and-----.(تاريخ ,أرقام , سلاسل محرفيه).

▪ is null, is not null, in (list), not in (list)

▪ المعاملات المنطقية: or, and, not.

صيغ المقارنة :

المعامل like ويستخدم كما يلي :

- Like 's%' : سلسلة محرفيه تبدأ بالعبارة او المحرف s
- Like '% s %' : سلسلة محرفيه تحوي في وسطها العبارة s
- Like '% s' : سلسلة محرفيه تنتهي بالعبارة او المحرف s

○ أمثلة :

- استعلام يعيد اسماء الموظفين الذين يقبضون أكثر من 10000.
- استعلام يعيد اسماء الموظفين الذين تاريخ توظيفهم بين عامي 1985-1980.
- استعلام يعيد اسماء وعمل الموظفين الذين يتقاضون عمولة.
- استعلام يعيد بيانات الموظفين الذين لا يعملون ك (clerk, salesman)

Example :

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500 ;
```



	LAST_NAME	SALARY
1	Markle	3367
2	Olson	3246
3	Philtanker	3367

Example :


```
SELECT employee_id, last_name, job_id,
department_id
FROM employees
WHERE department_id = 90 ;
```

EMPLOYEES

	EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
1	100	King	AD_PRES	90
2	101	Kochhar	AD_VP	90
3	102	De Haan	AD_VP	90
4	103	Hunold	IT_PROG	60
5	104	Ernst	IT_PROG	60
6	107	Lorentz	IT_PROG	60

...

“retrieve all
employees in
department 90”



	EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
1	100	King	AD_PRES	90
2	101	Kochhar	AD_VP	90
3	102	De Haan	AD_VP	90

استخدام المعامل IN :

```
SELECT employee_id, last_name, salary, manager_id
FROM employees
WHERE manager_id IN (100, 101, 201) ;
```

	EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
1	101	Kochhar	17000	100
2	102	De Haan	17000	100
3	124	Mourgos	5800	100
4	149	Zlotkey	10500	100
5	201	Hartstein	13000	100
6	200	Whalen	4400	101
7	205	Higgins	12008	101
8	202	Fay	6000	201

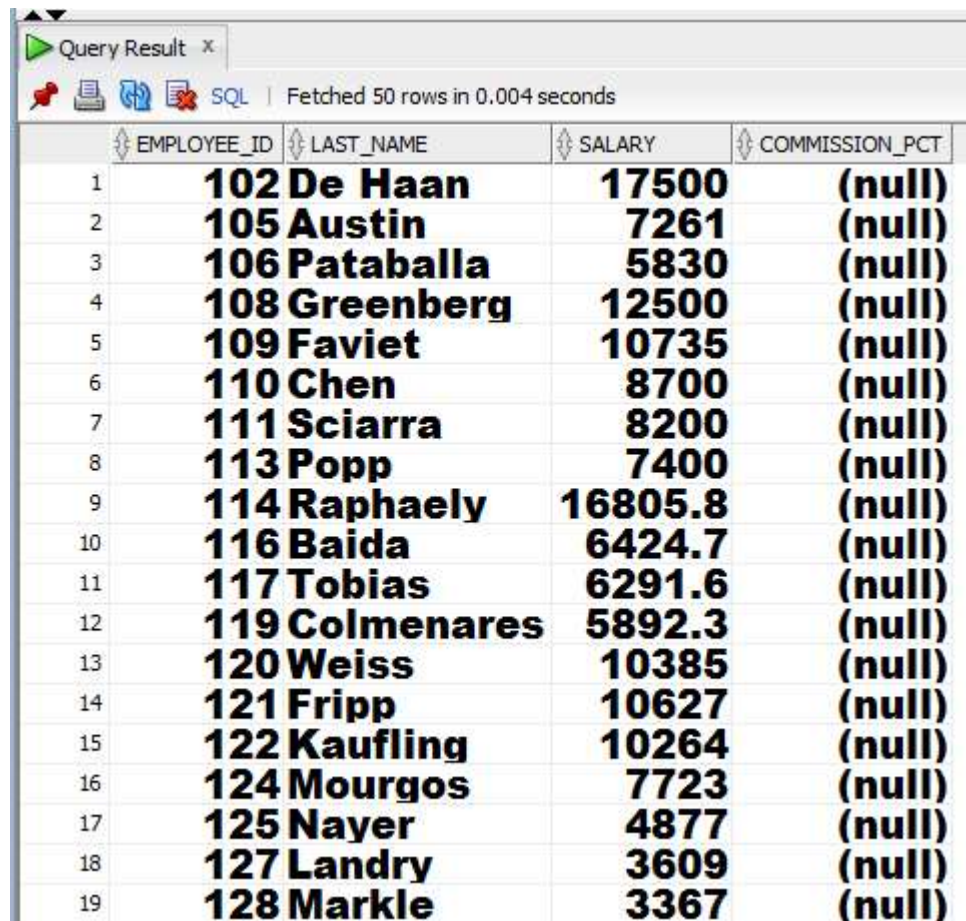
استخدام المعامل LIKE :

```
SELECT last_name
FROM employees
WHERE last_name LIKE '_o%' ;
```

	LAST_NAME
1	Kochhar
2	Lorentz
3	Mourgos

: استخدام المعامل IS NULL

```
SELECT employee_id, last_name,  
       salary, commission_pct  
FROM employees  
WHERE commission_pct IS NULL ;
```



Query Result x
SQL | Fetched 50 rows in 0.004 seconds

	EMPLOYEE_ID	LAST_NAME	SALARY	COMMISSION_PCT
1	102	De Haan	17500	(null)
2	105	Austin	7261	(null)
3	106	Pataballa	5830	(null)
4	108	Greenberg	12500	(null)
5	109	Faviet	10735	(null)
6	110	Chen	8700	(null)
7	111	Sciarra	8200	(null)
8	113	Popp	7400	(null)
9	114	Raphaely	16805.8	(null)
10	116	Baida	6424.7	(null)
11	117	Tobias	6291.6	(null)
12	119	Colmenares	5892.3	(null)
13	120	Weiss	10385	(null)
14	121	Fripp	10627	(null)
15	122	Kaufling	10264	(null)
16	124	Mourgos	7723	(null)
17	125	Nayer	4877	(null)
18	127	Landry	3609	(null)
19	128	Markle	3367	(null)



استخدام المعامل AND :

- يستخدم المعامل AND لاضافة اكثر من شرط لجملة الاستعلام ولا بد ان تتحقق كل الشروط في الصفوف التي ستظهر.

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >= 10000
AND job_id LIKE '%MAN%' ;
```

Query Result .x
SQL | All Rows Fetched: 10 in 0.002 seconds

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
114	Raphaely	PU MAN	16805.8
120	Weiss	ST MAN	10385
121	Fripp	ST MAN	10627
122	Kaufling	ST MAN	10264
145	Russell	SA MAN	14500
146	Partners	SA MAN	14000
148	Ahmed Ali	SA MAN	11500
149	Zlotkey	SA MAN	11000
201	Hartstein	MK MAN	16150
147	Errazuriz	SA MAN	12500



استخدام المعامل OR :

- يستخدم المعامل OR لاضافة اكثر من شرط لجملة الاستعلام ولا بد ان شرط واحد فقط في الصفوف التي ستظهر.

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >= 10000
OR job_id LIKE '%MAN%' ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD PRES	24500
102	De Haan	AD VP	17500
103	Hunold	IT PROG	10450
108	Greenberg	FI MGR	12500
109	Faviet	FI ACCOUNT	10735
114	Raphaely	PU MAN	16805.8
120	Weiss	ST MAN	10385
121	Fripp	ST MAN	10627
122	Kaufling	ST MAN	10264
124	Mourgos	ST MAN	7723
145	Russell	SA MAN	14500
146	Partners	SA MAN	14000
148	Ahmed Ali	SA MAN	11500
149	Zlotkey	SA MAN	11000
151	Bernstein	SA REP	10000
157	Sully	SA REP	10000
162	Vishney	SA REP	11000
163	Greene	SA REP	10000
168	Ozer	SA REP	12000
169	Bloom	SA REP	10500
174	Abel	SA REP	11500



- يستخدم المعامل NOT لاستبعاد الصفوف التي يتوفر بها الشرط.

```
SELECT last_name, job_id
FROM employees
WHERE job_id
NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP') ;
```

Query Result x
SQL | Fetched 50 rows in 0.007 seconds

	LAST_NAME	JOB_ID
1	Ahmed Ali	SA MAN
2	Baer	PR REP
3	Baida	PU CLERK
4	Bell	SH CLERK
5	Bull	SH CLERK
6	Cabrio	SH CLERK
7	Chen	FI ACCOUNT
8	Chung	SH CLERK
9	Colmenares	PU CLERK
10	De Haan	AD VP
11	Dellinger	SH CLERK
12	Dilly	SH CLERK
13	Errazuriz	SA MAN
14	Everett	SH CLERK
15	Faviet	FI ACCOUNT
16	Fay	MK REP
17	Feeney	SH CLERK
18	Fleaur	SH CLERK
19	Fripp	ST MAN
20	Gates	SH CLERK



أولويات تنفيذ المعاملات:

- يتم تنفيذها المعاملات اذا تواجدت داخل نفس الجملة طبقاً للجدول التالي :

Operator	Meaning
1	Arithmetic operators
2	Concatenation operator
3	Comparison conditions
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Not equal to
7	NOT logical operator
8	AND logical operator
9	OR logical operator

Example :

```
SELECT last_name, department_id, salary
FROM employees
WHERE department_id = 60
OR department_id = 80
AND salary > 10000;
```

```
SELECT last_name, department_id, salary
FROM employees
WHERE (department_id = 60
OR department_id = 80)
AND salary > 10000;
```

ترتيب البيانات :

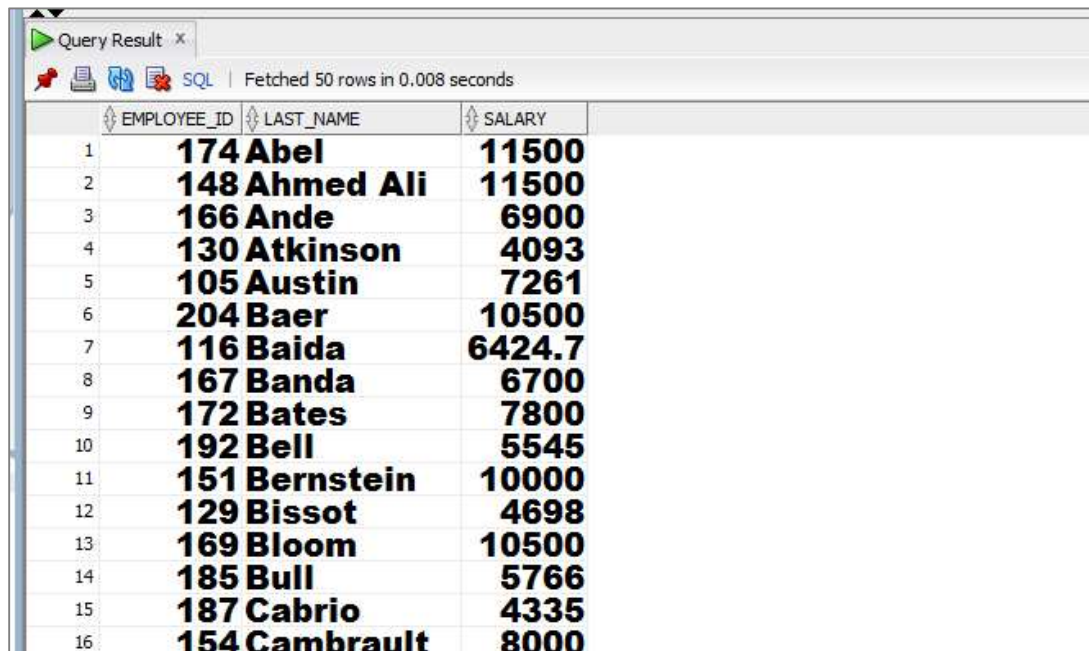
يمكن ترتيب البيانات الناتجة عن عملية select باستخدام العبارة order by والتي تتم إضافتها في نهاية

الاستعلام دوما كما يلي :

```
Select * |{ [distinct] column ,expression [alias]
From table_name
[where conditions]
[order by {column1,column2,...} [ASC|DESC]] ;
```

Example :

```
SELECT  employee_id, last_name, salary
FROM    employee
ORDER BY last_name;
```



	EMPLOYEE_ID	LAST_NAME	SALARY
1	174	Abel	11500
2	148	Ahmed Ali	11500
3	166	Ande	6900
4	130	Atkinson	4093
5	105	Austin	7261
6	204	Baer	10500
7	116	Baida	6424.7
8	167	Banda	6700
9	172	Bates	7800
10	192	Bell	5545
11	151	Bernstein	10000
12	129	Bissot	4698
13	169	Bloom	10500
14	185	Bull	5766
15	187	Cabrio	4335
16	154	Cambrault	8000



ويمكن الترتيب بأكثر من عمود مثال :

```
SELECT last_name, department_id, salary
FROM employees
ORDER BY department_id, salary DESC;
```

	LAST_NAME	DEPARTMENT_ID	SALARY
1	Whalen	10	5100
2	Hartstein	20	16150
3	Fay	20	8450
4	Raphaely	30	16805.8
5	Khoo	30	6690.9
6	Baida	30	6424.7
7	Tobias	30	6291.6
8	Himuro	30	6025.4
9	Colmenares	30	5892.3
10	Mavris	40	7700
11	Fripp	50	10627
12	Weiss	50	10385
13	Kaufling	50	10264
14	Vollman	50	8670
15	Mourgos	50	7723
16	Sarchand	50	5787
17	Bull	50	5766

كذلك يمكن الترتيب باستخدام موقع العمود او الAlias ان وجد مثال :

```
SELECT last_name, department_id, salary
FROM employees
ORDER BY 3, 2 DESC;
```

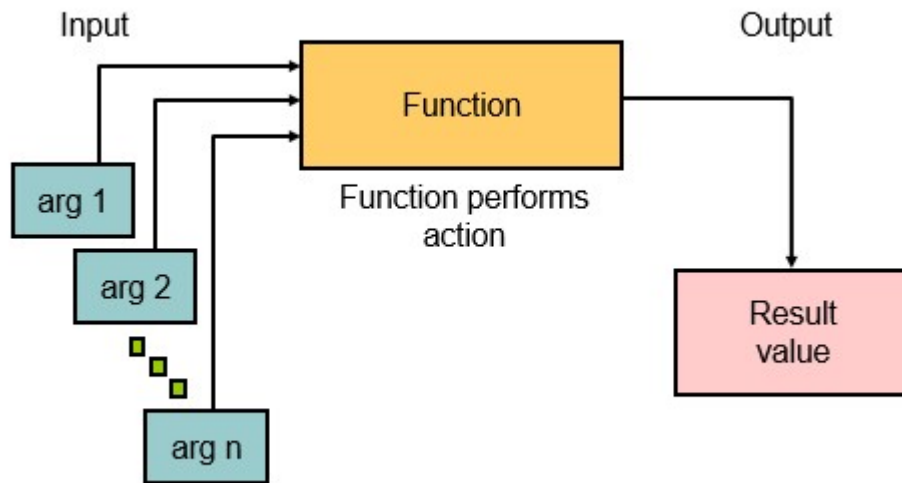
```
SELECT last_name, department_id, salary income
FROM employees
ORDER BY income;
```

SELECT Functions

تعتبر الدوال Functions من اهم الأدوات داخل لغة SQL حيث يمكن باستخدامها :

- اجراءات العمليات الحسابية على البيانات.
- تعديل بيانات الجدول.
- معالجة المخرجات من جمل الاستعلام.
- تحسين مظهر البيانات المعروض.
- تحويل البيانات من نوع لأخر.

تستقبل الدوال مدخل واحد او اكثر وينتج عنها قيمة واحدة في الاغلب.





أمثلة على استخدام الدوال :


```
SELECT last_name, UPPER(last_name), LOWER  
(last_name)  
FROM employees  
WHERE LENGTH (last_name) = 4;
```

	LAST_NAME	UPPER(LAST_NAME)	LOWER(LAST_NAME)
1	Abel	ABEL	abel
2	Ande	ANDE	ande
3	Baer	BAER	baer
4	Bell	BELL	bell
5	Bull	BULL	bull
6	Chen	CHEN	chen
7	Hall	HALL	hall
8	Khoo	KHOO	khoo
9	King	KING	king
10	King	KING	king
11	Ozer	OZER	ozer
12	Popp	POPP	popp
13	Rajs	RAJS	rajs

```
SELECT CONCAT(CONCAT(last_name,  
                    ' - job category is '), job_id)  
FROM employees  
WHERE SUBSTR(job_id, 4) = 'REP';
```

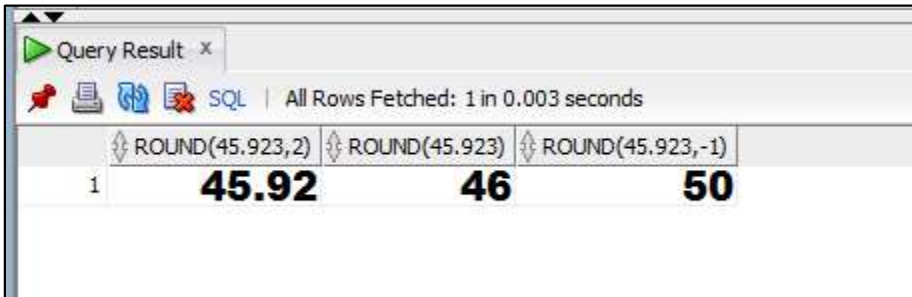
	CONCAT(CONCAT(LAST_NAME, '-JOBCATEGORYIS'),JOB_ID)
1	Abel- job category is SA REP
2	Ande- job category is SA REP
3	Baer- job category is PR REP
4	Banda- job category is SA REP
5	Bates- job category is SA REP
6	Bernstein- job category is SA REP
7	Bloom- job category is SA REP
8	Cambrault- job category is SA REP
9	Doran- job category is SA REP
10	Fay- job category is MK REP
11	Fox- job category is SA REP
12	Grant- job category is SA REP
13	Greene- job category is SA REP

```
SELECT last_name,
       UPPER(CONCAT(SUBSTR (LAST_NAME, 1, 8), '_US'))
FROM   employees
WHERE  department_id = 60;
```



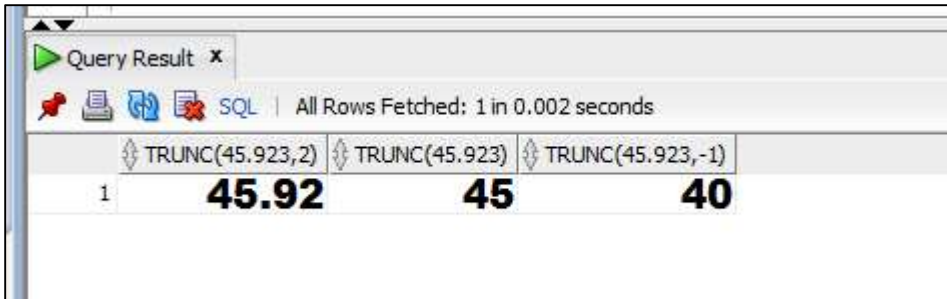
LAST_NAME	UPPER(CONCAT(SUBSTR(LAST_NAME,1,8),'_US'))
Hunold	HUNOLD US
Ernst	ERNST US
Austin	AUSTIN US
Pataballa	PATABALL US
Lorentz	LORENTZ US

```
SELECT ROUND (45.923,2) , ROUND (45.923) ,
       ROUND (45.923,-1)
FROM   DUAL;
```



	ROUND(45.923,2)	ROUND(45.923)	ROUND(45.923,-1)
1	45.92	46	50

```
SELECT TRUNC (45.923,2) , TRUNC (45.923) ,
       TRUNC (45.923,-1)
FROM   DUAL;
```



	TRUNC(45.923,2)	TRUNC(45.923)	TRUNC(45.923,-1)
1	45.92	45	40



أمثلة على استخدام دوال التاريخ :

```
SELECT sysdate FROM dual;
```

```
SELECT SESSIONTIMEZONE, CURRENT_DATE FROM DUAL;
```

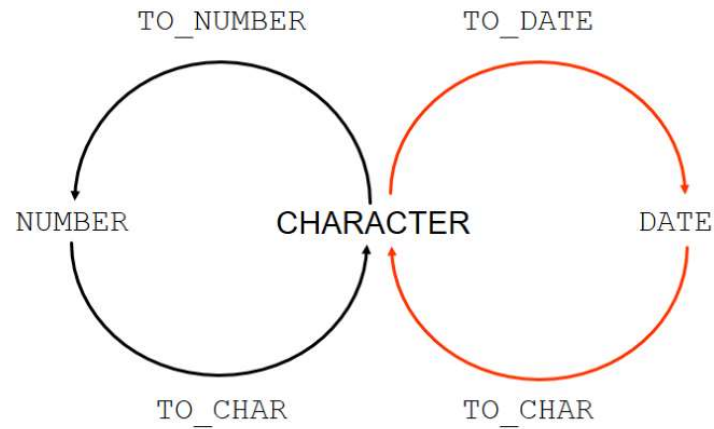
```
SELECT SESSIONTIMEZONE, CURRENT_TIMESTAMP FROM DUAL;
```

```
SELECT last_name, (SYSDATE-hire_date)/7 AS WEEKS  
FROM employees  
WHERE department_id = 90;
```

```
SELECT employee_id,  
       hire_date,  
       MONTHS_BETWEEN (SYSDATE, hire_date),  
       ADD_MONTHS (hire_date, 6) REVIEW,  
       NEXT_DAY (hire_date, 'FRIDAY'),  
       LAST_DAY(hire_date)  
FROM employees  
WHERE MONTHS_BETWEEN (SYSDATE, hire_date) < 150;
```


دوال التحويل :

تستخدم الدوال التالية لتحويل البيانات من نوع الى اخر كما يلي :



أمثلة :

```

SELECT  employee_id, last_name,
        TO_CHAR(hire_date, 'Month/YY') Month_Hired
FROM    employees ;
  
```

EMPLOYEE_ID	LAST_NAME	MONTH_HIRED
1	King	June /87
2	De Haan	January /93
3	Hunold	January /90
4	Austin	June /97
5	Pataballa	February /98
6	Lorentz	February /99
7	Greenberg	August /94
8	Faviet	August /94
9	Chen	September/97
10	Sciarra	September/97
11	Popp	December /99
12	Raphaely	December /94
13	Baida	December /97
14	Tobias	July /97
15	Himuro	November /98
16	Colmenares	August /99



```
SELECT last_name,  
       TO_CHAR(hire_date, 'fmDD Month YYYY')  
       AS HIREDATE  
FROM   employees;
```

	LAST_NAME	HIREDATE
1	King	17 June 1987
2	De Haan	13 January 1993
3	Hunold	3 January 1990
4	Austin	25 June 1997
5	Pataballa	5 February 1998
6	Lorentz	7 February 1999
7	Greenberg	17 August 1994
8	Faviet	16 August 1994
9	Chen	28 September 1997
10	Sciarra	30 September 1997
11	Popp	7 December 1999

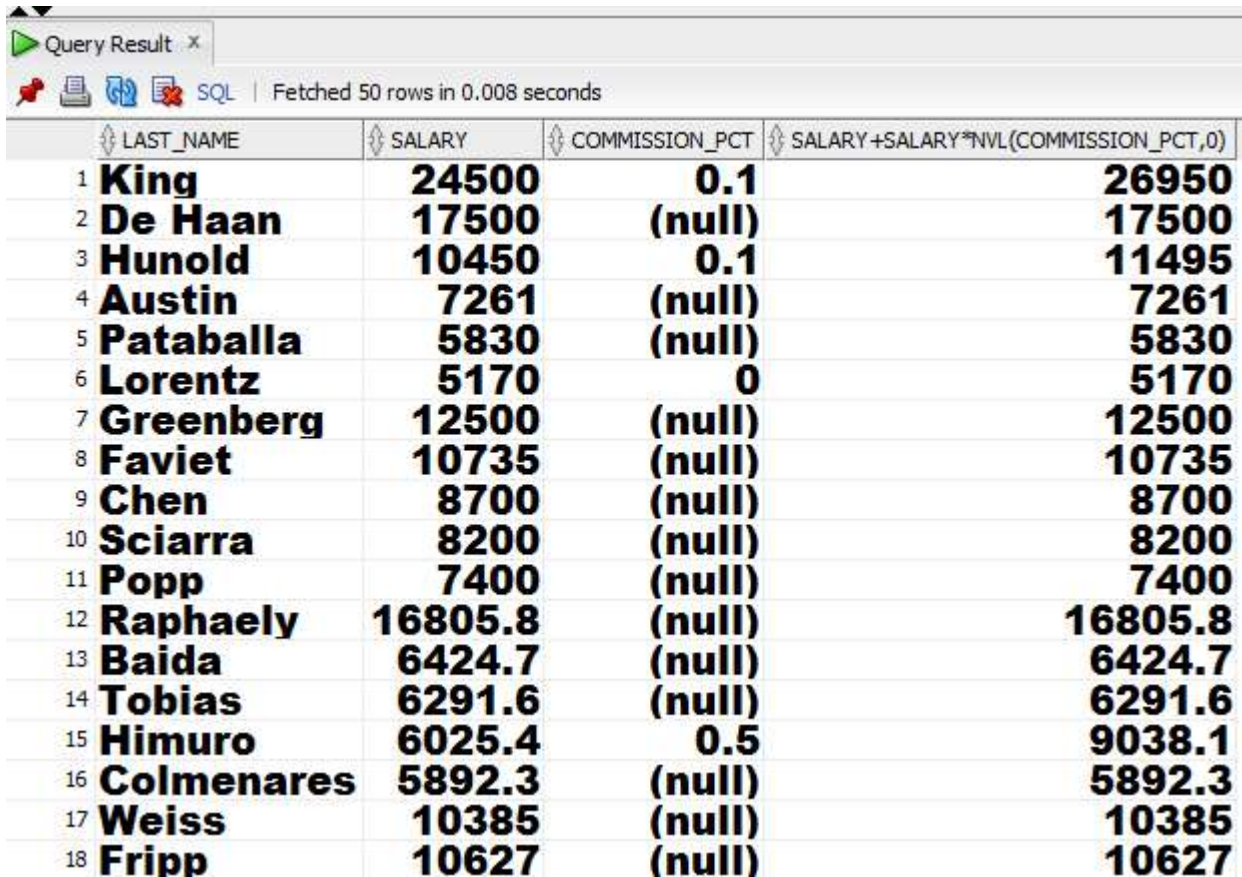
```
SELECT employee_id,  
       last_name,  
       TO_CHAR(salary, '$99,999.00') SALARY  
FROM   employees;
```

	EMPLOYEE_ID	LAST_NAME	SALARY
1	100	King	\$24,500.00
2	102	De Haan	\$17,500.00
3	103	Hunold	\$10,450.00
4	105	Austin	\$7,261.00
5	106	Pataballa	\$5,830.00
6	107	Lorentz	\$5,170.00
7	108	Greenberg	\$12,500.00
8	109	Faviet	\$10,735.00
9	110	Chen	\$8,700.00
10	111	Sciarra	\$8,200.00

معالجة قيم الNull:

يمكن استخدام الدالة NVL لاستبدال قيم الNull بقيم اخرى تستخدم في نتائج الاستعلام.

```
SELECT last_name,
       salary,
       commission_pct,
       salary + salary * NVL(commission_pct, 0)
FROM employees;
```



Query Result x

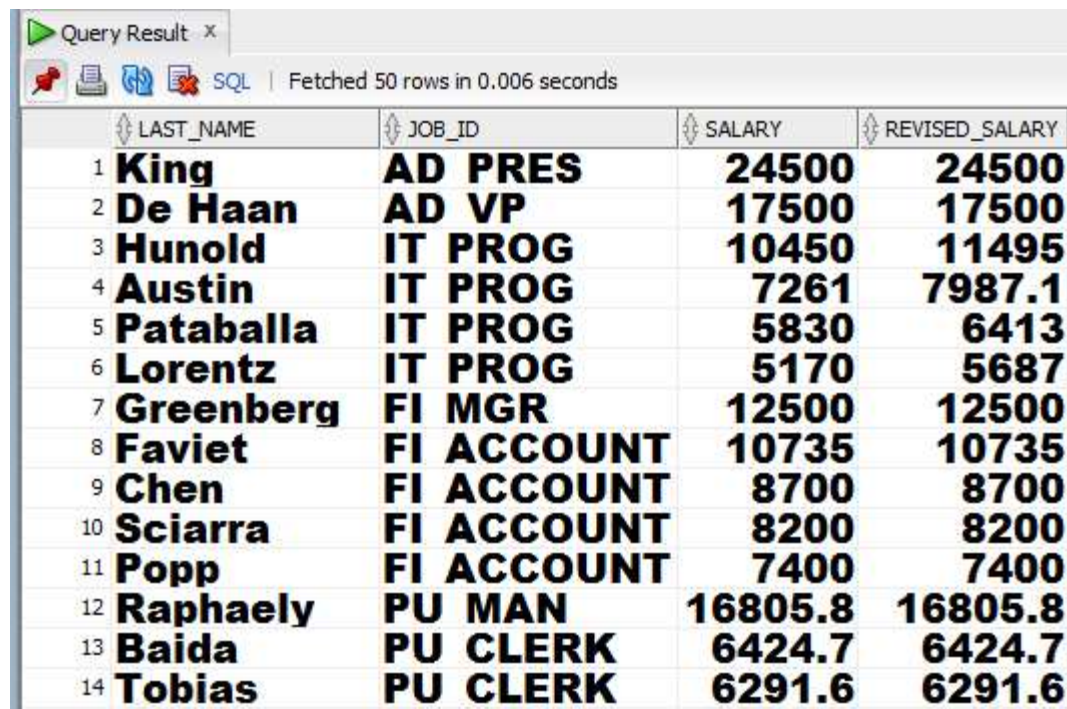
SQL | Fetched 50 rows in 0.008 seconds

	LAST_NAME	SALARY	COMMISSION_PCT	SALARY+SALARY*NVL(COMMISSION_PCT,0)
1	King	24500	0.1	26950
2	De Haan	17500	(null)	17500
3	Hunold	10450	0.1	11495
4	Austin	7261	(null)	7261
5	Pataballa	5830	(null)	5830
6	Lorentz	5170	0	5170
7	Greenberg	12500	(null)	12500
8	Faviet	10735	(null)	10735
9	Chen	8700	(null)	8700
10	Sciarra	8200	(null)	8200
11	Popp	7400	(null)	7400
12	Raphaely	16805.8	(null)	16805.8
13	Baida	6424.7	(null)	6424.7
14	Tobias	6291.6	(null)	6291.6
15	Himuro	6025.4	0.5	9038.1
16	Colmenares	5892.3	(null)	5892.3
17	Weiss	10385	(null)	10385
18	Fripp	10627	(null)	10627

استخدام ال Case Expression:

تسمح Case بعرض قيم حسب شروط محددة بالاستعلام على سبيل المثال :

```
SELECT last_name, job_id, salary,
       CASE job_id WHEN 'IT_PROG' THEN 1.10*salary
                   WHEN 'ST_CLERK' THEN 1.15*salary
                   WHEN 'SA_REP' THEN 1.20*salary
                   ELSE salary END
       "REVISED_SALARY"
FROM employees;
```



	LAST_NAME	JOB_ID	SALARY	REVISED_SALARY
1	King	AD PRES	24500	24500
2	De Haan	AD VP	17500	17500
3	Hunold	IT PROG	10450	11495
4	Austin	IT PROG	7261	7987.1
5	Pataballa	IT PROG	5830	6413
6	Lorentz	IT PROG	5170	5687
7	Greenberg	FI MGR	12500	12500
8	Faviet	FI ACCOUNT	10735	10735
9	Chen	FI ACCOUNT	8700	8700
10	Sciarra	FI ACCOUNT	8200	8200
11	Popp	FI ACCOUNT	7400	7400
12	Raphaely	PU MAN	16805.8	16805.8
13	Baida	PU CLERK	6424.7	6424.7
14	Tobias	PU CLERK	6291.6	6291.6

Group Functions

يمكن استخدام الدوال التجميعية Group Function لحساب القيم الاحصائية للبيانات داخل الجدول وعرض هذه الاحصائيات مقسمة على مجموعات كما يلي :

Function	Description
AVG ([DISTINCT ALL] n)	Average value of <i>n</i> , ignoring null values
COUNT	Number of rows, where <i>expr</i> evaluates to something other than null (count all selected rows using *, including duplicates and rows with nulls)
MAX ([DISTINCT ALL] expr)	Maximum value of <i>expr</i> , ignoring null values
MIN ([DISTINCT ALL] expr)	Minimum value of <i>expr</i> , ignoring null values
STDDEV ([DISTINCT ALL] n)	Standard deviation of <i>n</i> , ignoring null values
SUM ([DISTINCT ALL] n)	Sum values of <i>n</i> , ignoring null values
LISTAGG	Orders data within each group specified in the ORDER BY clause and then concatenates the values of the measure column
VARIANCE ([DISTINCT ALL] n)	Variance of <i>n</i> , ignoring null values

أمثلة :

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

	AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
1	8150	11000	6000	32600

يمكن استخدام Group By لتقسيم البيانات الى مجموعات و اظهار البيانات التجميعية لكل مجموعة.

EMPLOYEES

DEPARTMENT_ID	SALARY
1	4400
2	13000
3	6000
4	2500
5	2600
6	3100
7	3500
8	5800
9	9000
10	6000
11	4200
12	11000
13	8600
...	
18	8300
19	12000
20	7000

Average salary in the EMPLOYEES table for each department

DEPARTMENT_ID	AVG(SALARY)
1	7000
2	9500
3	19333.333333333333...
4	10150
5	3500
6	10033.333333333333...
7	4400
8	6400

مثال :

```
SELECT department_id, COUNT(employee_id)
FROM employees
GROUP BY department_id
ORDER BY department_id;
```

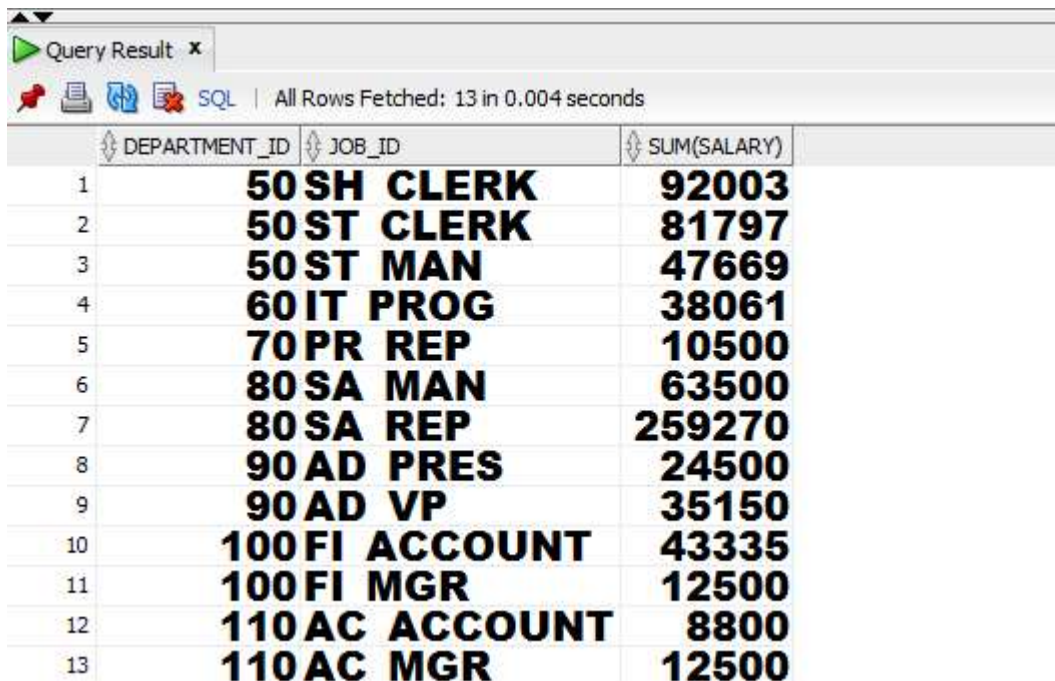
Query Result x

SQL | All Rows Fetched: 12 in 0.003 seconds

DEPARTMENT_ID	COUNT(EMPLOYEE_ID)
1	10
2	20
3	30
4	40
5	50
6	60
7	70
8	80
9	90
10	100
11	110
12	(null)

يمكن تقسيم البيانات طبقاً لأكثر من عمود مثال :

```
SELECT department_id, job_id, SUM(salary)
FROM employees
WHERE department_id > 40
GROUP BY department_id, job_id
ORDER BY department_id;
```



Query Result x

SQL | All Rows Fetched: 13 in 0.004 seconds

	DEPARTMENT_ID	JOB_ID	SUM(SALARY)
1	50	SH CLERK	92003
2	50	ST CLERK	81797
3	50	ST MAN	47669
4	60	IT PROG	38061
5	70	PR REP	10500
6	80	SA MAN	63500
7	80	SA REP	259270
8	90	AD PRES	24500
9	90	AD VP	35150
10	100	FI ACCOUNT	43335
11	100	FI MGR	12500
12	110	AC ACCOUNT	8800
13	110	AC MGR	12500

استخدام HAVING داخل الاستعلام :

يتم استخدام HAVING لتحديد المجموعات التي ستظهر في نتيجة للاستعلام.

EMPLOYEES

	DEPARTMENT_ID	SALARY
1	10	4400
2	20	13000
3	20	6000
4	50	2500
5	50	2600
6	50	3100
7	50	3500
8	50	5800
9	60	9000
10	60	6000
11	60	4200
12	80	11000
13	80	8600
...		
18	110	8300
19	110	12000
20	(null)	7000

The maximum salary per department when it is greater than \$10,000

	DEPARTMENT_ID	MAX(SALARY)
1	20	13000
2	90	24000
3	110	12000
4	80	11000

مثال :

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

	DEPARTMENT_ID	MAX(SALARY)
1	100	12500
2	30	16805.8
3	90	24500
4	20	16150
5	70	10500
6	110	12500
7	50	10627
8	80	14500
9	60	10450