

مفاهيم قواعد البيانات
**Database Concepts and
Design**

المستوى : الرابع

رمز المقرر : ٢٢٣ حسب

المتطلبات السابقة : ١٢١ حسب

طبيعة المقرر : ساعتين نظري + ساعتين عملي

المرجع : أصول نظم قواعد البيانات - الجزء الأول

تأليف : أ.د. رامي المصري / أ.د. شامكانت نافاث

ترجمة د.م. خالد ناصر السيد

استاذة المادة

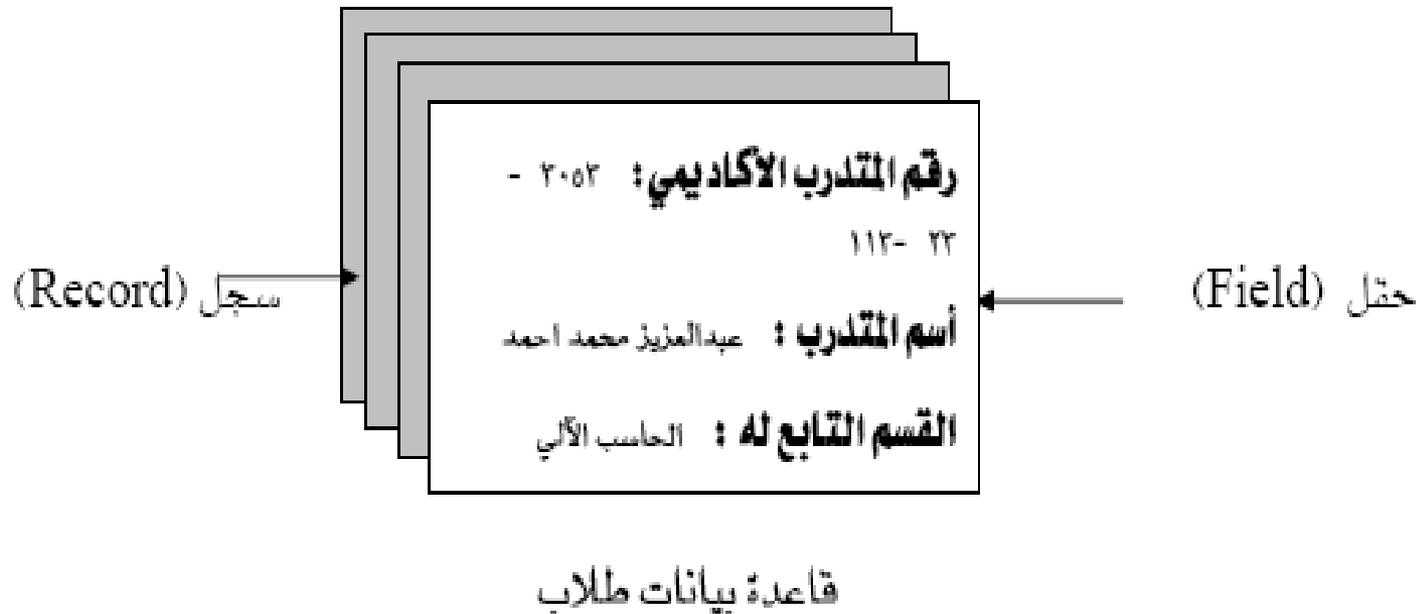
م/ ليندا البديري

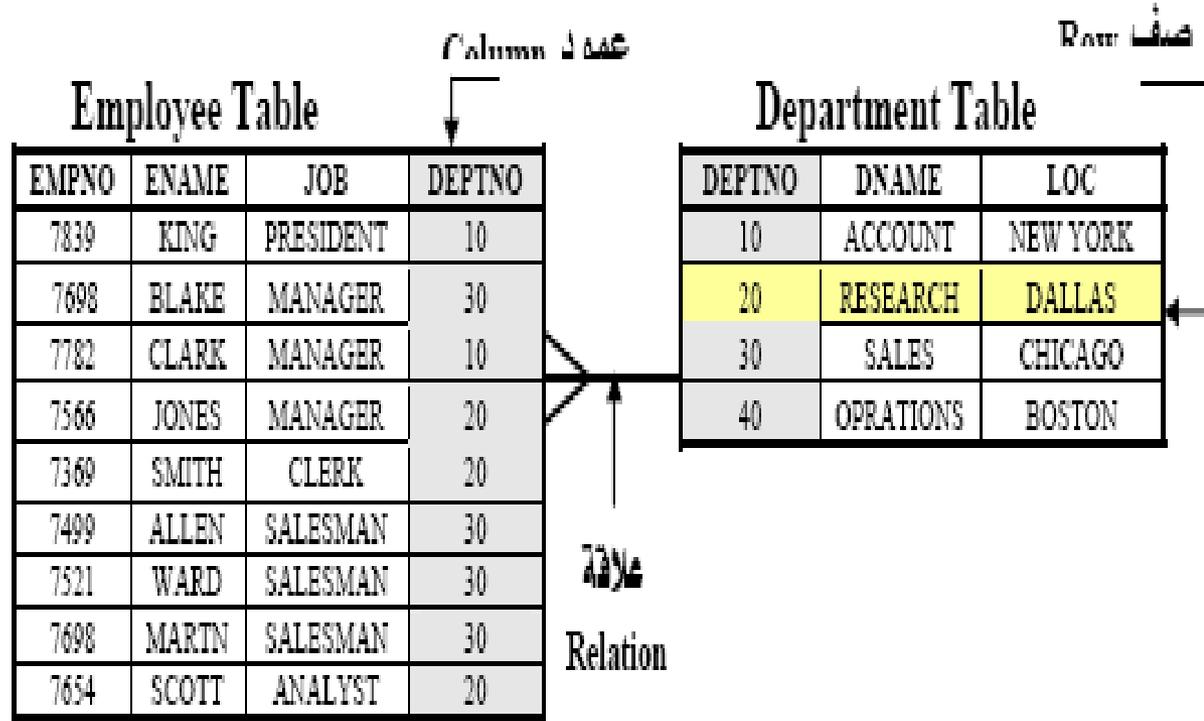
مقدمة قواعد بيانات اوراكل

***INTRODUCTION TO
ORACLE DATABASE***



مفهوم قواعد البيانات (Database Concepts)





الشكل رقم (٣)

قامت شركة أوراكل باعتماد لغة تسمى لغة الاستفسارات SQL (Structured Query Language) للتعامل مع قواعد البيانات العلائقية وهي لغة سهلة تقوم بإنشاء الأشياء (Objects) الخاصة بقاعدة البيانات مثل الجداول والتعامل معها وتقوم بعمل جميع الاستفسارات اللازمة والتي نريد أن نعرفها من قاعدة البيانات ويطلق عليها لغة (SQL) ، كما قامت شركة أوراكل أيضاً بعمل تطبيق أو بيئة تستقبل الأوامر الخاصة بلغة الاستفسارات SQL وهذه البيئة تسمى محرر الـ (SQL*PLUS) ويمكن من خلال هذا المحرر استقبال الأوامر الخاصة بلغة SQL وتنفيذها وتعديل الأخطاء الموجودة في الأمر وجميع العمليات الأخرى ، وسوف نقوم بشرح مفصل عن كل من لغة SQL والمحرر SQL*PLUS فيما يلي :

لغة الاستفسارات Structured Query Language (SQL)

هي لغة تستخدم لإصدار جميع الأوامر التي تتعلق بقاعدة البيانات ، وتنقسم هذه اللغة إلى خمسة

أقسام رئيسة يمكن من خلالها إصدار الأوامر الخاصة بكل قسم ، والجدول التالي يوضح الأقسام

المختلفة من هذه اللغة ووصف الأوامر لكل قسم .

القسم	الأمر	وصف الأمر
Data Retrieval	SELECT	أمر استرجاع البيانات من جدول أو كائن
(DML) Data Manipulation Language	INSERT	أمر إضافة بيانات إلى جدول أو كائن
	UPDATE	أمر التعديل في بيانات جدول أو كائن
	DELETE	أمر حذف بيانات جدول أو كائن
(DDL) Data Definition Language	CREATE	أمر إنشاء جدول أو كائن
	Alter	أمر التعديل في جدول أو كائن
	DROP	أمر إلغاء جدول أو كائن
	RENAME	أمر تغيير الاسم جدول أو كائن
	TRUNCATE	إلغاء جزء أو بتر جزء من جدول أو كائن
Transaction Control	COMMIT	تثبيت البيانات في الجدول
	ROLLBACK	الرجوع عن تثبيت البيانات
	SAVEPOINT	الرجوع لنقطة معينة
(DCL) Data Control Language	GRANT	إعطاء الصلاحيات للمستخدمين للدخول على البيانات
	REVOKE	سحب الصلاحيات من المستخدمين

الصيغة (الشكل) العامة لجملته SELECT .

SELECT * or Columns [alias]
FROM Table
WHERE condition or conditions
ORDER BY Column or Alias [ASC or DESC] ;

متطلبات وإرشادات كتابة جمل SQL .

يوجد هناك بعض الإرشادات التي يجب مراعاتها عند كتابة جملة SQL لتكون الجملة صحيحة وقابلة للتنفيذ ، وهذه الإرشادات هي :

- ١ - يمكن كتابة مكونات جملة SQL بالأحرف الكبيرة أو الصغيرة فهذا لا يؤثر على سلامة الجملة وذلك لأن جملة SQL غير حساسة للحروف `Not Case Sensitive` .
- ٢ - يفصل بين أسماء الحقول باستخدام الفاصلة (و) .
- ٣ - يمكن كتابة جملة SQL في عدة سطور فهذا لا يؤثر في صحة الجملة .
- ٤ - لا يمكن فصل الكلمات المحجوزة للغة أو اختصارها ، والكلمات المحجوزة تسمى `Keywords` وهي مثل (`SELECT , FROM , WHERE , ORDER BY`) .
- ٥ - يفضل كتابة الجملة على أسطر ليسهل قراءتها وفهمها .
- ٦ - لا بد من الإعلان عن نهاية الجملة بواسطة (;) .
- ٧ - ملحوظة : أوامر محرر `SQL*PLUS` لا يوضع بعدها الفاصلة المنقوطة (;) .

مثال (١) : عرض جميع الحقول من جدول الإدارات DEPT .

```
SQL> SELECT *  
2 FROM dept ;
```

DEPTNO -----	DNAME -----	LOC -----
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

مثال (٢) : عرض حقول معينة من جدول الإدارات DEPT .

```
SQL> SELECT deptno , dname  
2 FROM dept ;
```

DEPTNO	DNAME
-----	-----
10	ACCOUNTING
20	RESEARCH
30	SALES
40	OPERATIONS

مثال (٢) : عرض حقول بأسماء مستعارة من جدول الموظفين .

```
SQL> SELECT ename AS name , sal salary , job "employee job"  
2 FROM emp ;
```

NAME	SALARY	employee job
SMITH	800	CLERK
ALLEN	1600	SALESMAN
WARD	1250	SALESMAN
JONES	2975	MANAGER
MARTIN	1250	SALESMAN
BLAKE	2850	MANAGER
CLARK	2450	MANAGER
SCOTT	3000	ANALYST
KING	5000	PRESIDENT
TURNER	1500	SALESMAN
ADAMS	1100	CLERK

استخدام العمليات الحسابية وأولويات تنفيذها مع جملة SELECT :

من الممكن إجراء عمليات حسابية على الحقول العددية للحصول على معلومة معينة فمثلاً إذا أردنا إظهار الموظفين ورواتبهم في سنة فإننا نقوم بضرب راتب كل موظف في العدد ١٢ بشكل التالي ($SAL * 12$) ، وكذلك عند إظهار إجمالي الراتب لكل موظف بعد إضافة ٥٠٠ ريال عليه ($SAL + 500$) . لاحظ أن العمليات الحسابية على الحقول لا تؤثر على البيانات المخزنة داخل الجدول .

المعاملات الحسابية التي تستخدم في العمليات الحسابية Arithmetic Operators

١ - الجمع (+) .

٢ - الطرح (-) .

٣ - الضرب (*) .

٤ - القسمة (/) .

يمكن استخدام المعاملات الحسابية في جميع أجزاء جملة SQL ماعدا الجزء الخاص بـ FROM

مثال (٤) : عرض رواتب الموظفين السنوية من جدول الموظفين .

```
SQL> SELECT ename , sal , sal*12 "annual salary"  
2 FROM emp ;  
3
```

ENAME	SAL	annual salary
SMITH	800	9600
ALLEN	1600	19200
WARD	1250	15000
JONES	2975	35700

أولويات تنفيذ العوامل الحسابية Operator Precedence

عند إجراء عملية حسابية كبيرة على حقل من الحقول لابد أن تعرف كيفية حسابها ولعرفة ذلك

لا بد أن تعرف أولوية تنفيذ العوامل داخل جملة SQL فهي تنفذ بالترتيب التالي :

- ١ - أولوية تنفيذ العمليات الحسابية للضرب والقسمة ثم للجمع والطرح .
- ٢ - العمليات من نفس الأولوية تنفذ من اليسار إلى اليمين .
- ٢ - عند وجود الأقواس في العمليات الحسابية يكون ما بداخلها له الأولوية وينفذ حسب الفقرة رقم (١) .

لاحظ الفرق بين العمليتين التاليتين :

$$1 - 100*(40+10) = 100*50=5000 .$$

$$2 - (100*40)+10 = 4000+10= 4010 .$$

مثال (5) : عرض رواتب الموظفين السنوية من جدول الموظفين .

```
SQL> SELECT ename , sal , 12*sal+100
2 FROM emp;
```

لاحظ أولوية التنفيذ :

العملية رقم (1) تنفذ أولاً ثم العملية رقم (2) .

ENAME	SAL	12*SAL+100
SMITH	800	9700
ALLEN	1600	19300

مثال (6) : عرض رواتب الموظفين السنوية من جدول الموظفين .

```
SQL> SELECT ename , sal , 12*(sal+100)
2 FROM emp;
```

لاحظ أولوية التنفيذ :

العملية رقم (1) تنفذ أولاً لوجود الأقواس ثم

العملية رقم (2) .

ENAME	SAL	12*(SAL+100)
SMITH	800	10800
ALLEN	1600	20400

استخدام أداة الربط بين الحقول (||) Concatenation .

لعمل سلسلة من الحقول نقوم بربط حقلين أو أكثر باستخدام أداة الربط (||) والتي تسمى Concatenation ، ويكون ناتج الربط بين الحقول هو حقل واحد فقط ، ومن الممكن أن نربط مع الحقول نص معين نضعه بين علامتي تنصيص فردية (' ') ، والمثال التالي يوضح ذلك .

```
SQL> SELECT ename, job , ename||job as "employees"  
2 FROM emp ;
```

ENAME	JOB	employees
SMITH	CLERK	SMITHCLERK
ALLEN	SALESMAN	ALLENSALESMAN
WARD	SALESMAN	WARDSALESMAN
JONES	MANAGER	JONESMANAGER
MARTIN	SALESMAN	MARTINSALESMAN

```
SQL> SELECT ename, job , ename||' is a '||job as "employees"  
2 FROM emp ;
```

ENAME	JOB	employees
SMITH	CLERK	SMITH is a CLERK
ALLEN	SALESMAN	ALLEN is a SALESMAN
WARD	SALESMAN	WARD is a SALESMAN
JONES	MANAGER	JONES is a MANAGER
MARTIN	SALESMAN	MARTIN is a SALESMAN

استخدام عبارة DISTINCT لمنع تكرار السجلات :

عند إظهار محتويات الجدول نجد تكرار بعض القيم للحقل الواحد دون فائدة ، فمثلاً لو طلب منك معرفة أرقام الإدارات التي ينتمي إليها الموظفين في جدول الموظفين (EMP) ، فإنك بالطبع سوف تكتب هذا الأمر .

```
SQL> SELECT deptno  
2 FROM emp ;
```

DEPTNO

20
30
30
20
30
30

لاحظ التكرار في أرقام الإدارات



```
SQL> SELECT DISTINCT deptno  
2 FROM emp ;
```

```
DEPTNO
```

```
-----
```

```
10
```

```
20
```

```
30
```

إظهار البناء الداخلي للجداول باستخدام الأمر DESCRIBE (DESC)

لإظهار معلومات حول أسماء الحقول وأنواعها الموجودة في جدول معين أي لإظهار البناء الداخلي للجدول نستخدم الأمر (DESCRIBE) والذي يمكن اختصاره إلى الأحرف التالية (DESC) .

```
desc jobs;
```

Object Type TABLE Object JOBS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>JOBS</u>	<u>JOB_ID</u>	Varchar2	10	.	.	1	.	.	Primary key of jobs table.
	<u>JOB_TITLE</u>	Varchar2	35	A not null column that shows job title, e.g. AD_VP, FI_ACCOUNTANT
	<u>MIN_SALARY</u>	Number	.	6	0	.	✓	.	Minimum salary for a job title.
	<u>MAX_SALARY</u>	Number	.	6	0	.	✓	.	Maximum salary for a job title

1-4

التعامل مع القيمة NULL

```
select * from employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	-	103	60
105	David	Austin	DAUSTIN	590.423.4569	25-JUN-97	IT_PROG	4800	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-98	IT_PROG	4800	-	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	4200	-	103	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-94	FI_MGR	12000	-	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	16-AUG-94	FI_ACCOUNT	9000	-	108	100

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

عند إجراء أي عملية حسابية عليها يكون النتائج دائماً (قيمة خالية NULL) ، يمكنك أن

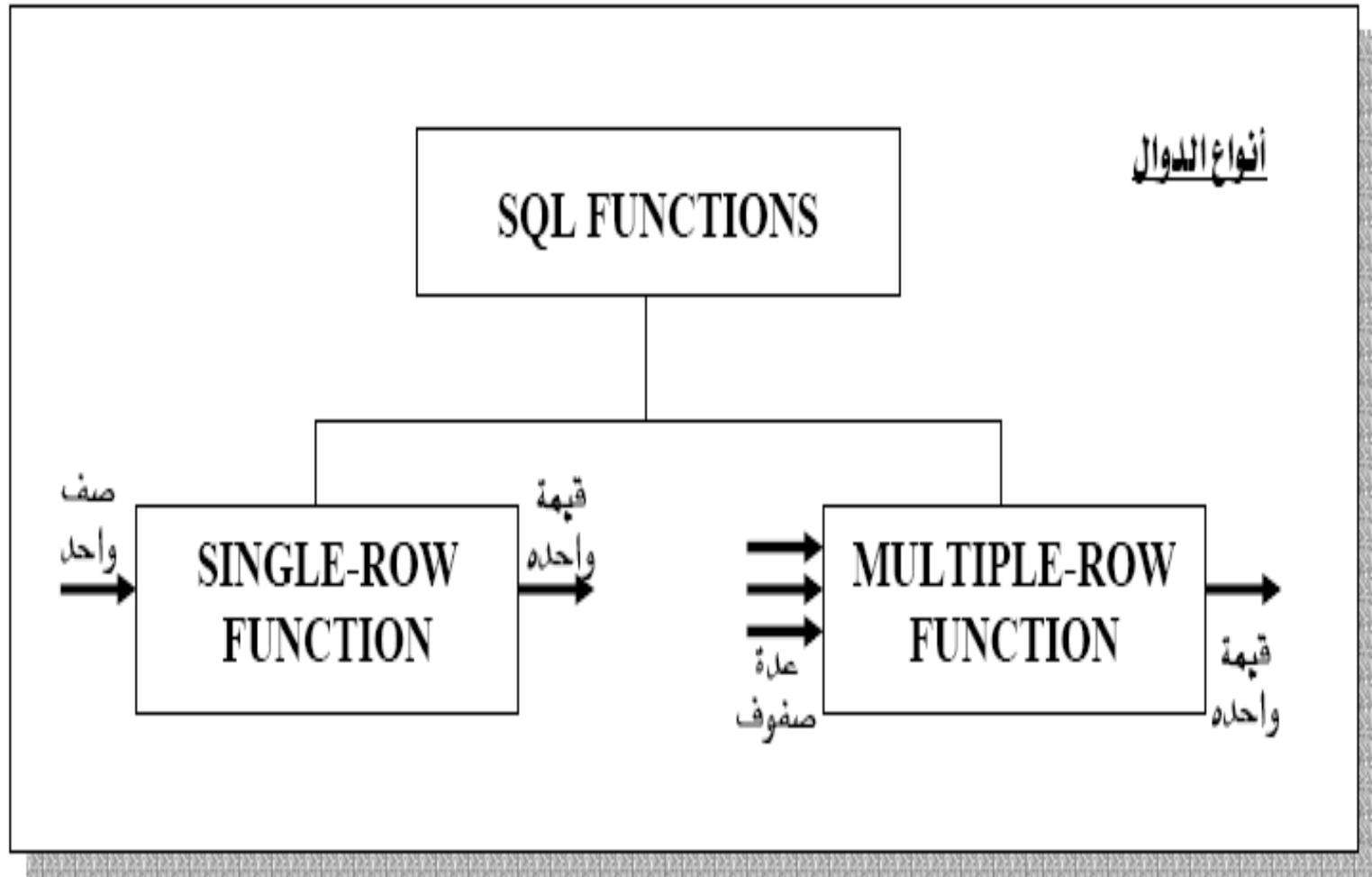
تلاحظ ذلك في المثال التالي : `select first_name , salary , salary + commission_pct`
`from employees;`

FIRST_NAME	SALARY	SALARY+COMMISSION_PCT
Steven	24000	-
Neena	17000	-
Lex	17000	-
Alexander	9000	-
Bruce	6000	-
David	4800	-
Valli	4800	-
Diana	4200	-
Nancy	12000	-
Daniel	9000	-
More than 10 rows available. Increase rows selector to view more rows.		

```
select first_name , salary , salary + nvl(commission_pct,0)
from employees;
```

FIRST_NAME	SALARY	SALARY+NVL(COMMISSION_PCT,0)
Steven	24000	24000
Neena	17000	17000
Lex	17000	17000
Alexander	9000	9000
Bruce	6000	6000
David	4800	4800
Valli	4800	4800
Diana	4200	4200
Nancy	12000	12000
Daniel	9000	9000
More than 10 rows available. Increase rows selector to view more rows.		

الدوال في لغة SQL



النوع الأول : يسمى دوال الصف الواحد SINGLE_ROW FUNCTIONS وهذا النوع يقوم

بالتعامل مع بيانات صف واحد فقط لإخراج قيمة واحدة وهو الذي سيتم شرحه بالتفصيل في هذا

الفصل .

النوع الثاني : يسمى الدوال التجميعية لأكثر من صف MULTIPLE_ROW FUNCTION

وهذا النوع يقوم بالتعامل مع بيانات أكثر من صف لإخراج قيمة واحدة فقط وسوف يتم شرحه في المحاضرة

القادمة بإذن الله

دوال الصف الواحد : Single-Row Functions

وهي كما ذكرنا عبارة عن دوال تتعامل مع بيانات صف واحد فقط وتكون نتيجتها قيمة واحدة

فقط ، وتستخدم في أي مكان من جملة `SELECT` ما عدا الجزء `FROM` ، وتنقسم إلى عدة أنواع هي :

- دوال حرفية Character Functions .
- دوال رقمية Number Functions .
- دوال تاريخ Date Functions .
- دوال تحويل Conversion Functions .

الدوال الحرفية : Character Functions

وهي دوال تتعامل مع البيانات الحرفية وتكون نتيجتها إما حروفاً أو أرقاماً . والجدول التالي يبين جميع الدوال الحرفية ووظيفة كل منها .

الدالة FUNCTION	وظيفتها
LOWER(column\expression)	دالة تستخدم لتحويل جميع الحروف (عمود أو سلسلة) إلى حروف صغيرة Small
UPPER(column\expression)	دالة تستخدم لتحويل حروف (عمود أو سلسلة) إلى حروف كبيرة Capital
INITCAP(column\expression)	دالة تستخدم لتحويل الحرف الأول فقط من (عمود أو سلسلة) إلى حرف كبير Capital وباقي الحروف تحول إلى حروف صغيرة
CONCAT(column1\expression1, Column2\expression2)	دالة ربط عمودين أو سلسلتين معاً وهي تماماً مثل أداة الربط ()
SUBSTR(column\expression,m,n)	دالة تستخدم لقطع جزء من عمود أو سلسلة بدايةً من الحرف رقم m وعدد الحروف المقطوعة هي n .
LENGTH(column\expression)	دالة تستخدم لإيجاد عدد حروف السلسلة أو العمود (الناتج عدد)
INSTR(column\expression,m)	دالة تستخدم لتحديد مكان حرف معين داخل سلسلة أو عمود (الناتج عدد) والحرف m يعبر عن الحرف المراد تحديد مكانه
LPAD(column\expression,n,'string')	دالة تستخدم لضبط بيانات عمود أو سلسلة ناحية اليمين وذلك بهاء حرف معين من اليسار والحرف n لتحديد الطول بعد الضبط
RPAD(column\expression,n,'string')	دالة تستخدم لضبط بيانات عمود أو سلسلة ناحية اليسار وذلك بهاء حرف معين من اليمين والحرف n لتحديد الطول بعد الضبط
TRIM('character' FROM column\expression)	دالة تستخدم لقطع حرف معين من بداية أو نهاية الكلمة فقط

الجدول DUAL

- هو جدول خاص المالك له هو المستخدم SYSTEM ويمكن لكل المستخدمين الوصول الى هذا الجدول و استخدامه .
- يستخدم هذا الجدول لاكمال الصيغة العامة لجملة select حيث ان select و from متلازمان و في بعض الاحيان تستخدم select للاستعلام عن بيانات غير مخزنة في جداول حقيقية .
- عند الاستعلام عن محتويات هذا الجدول يظهر عمود واحد اسمه Dummy وصف واحد هو X .

```
select * from dual;
```

DUMMY
X

المثال	النتيجة
Select LOWER('GOOD by') from dual ;	good by
Select UPPER('GOOD by') from dual ;	GOOD BY
Select INITCAP('GOOD') from dual ;	Good
Select CONCAT('GOOD' , 'BY') from dual;	GOODBY
Select SUBSTR('GOOD BY',2,3) from dual;	OOD
Select LENGTH('GOOD') from dual;	4
Select INSTR('GOOD','D') from dual;	4
Select LPAD('AHMED',10,'*') from dual ;	*****AHMED
Select RPAD('AHMED',10,'*') from dual ;	AHMED*****
Select TRIM('S' FROM 'SAMI') from dual;	AMI

بعض الأمثلة على الدوال الحرفية

مثال ١ :

```
select employee_id, lower(first_name), upper(first_name), initcap(first_name), concat(first_name,job_id)
from employees;
```

EMPLOYEE_ID	LOWER(FIRST_NAME)	UPPER(FIRST_NAME)	INITCAP(FIRST_NAME)	CONCAT(FIRST_NAME,JOB_ID)
100	steven	STEVEN	Steven	StevenAD_PRES
101	neena	NEENA	Neena	NeenaAD_VP
102	lex	LEX	Lex	LexAD_VP
103	alexander	ALEXANDER	Alexander	AlexanderIT_PROG
104	bruce	BRUCE	Bruce	BruceIT_PROG
105	david	DAVID	David	DavidIT_PROG
106	valli	VALLI	Valli	ValliIT_PROG
107	diana	DIANA	Diana	DianaIT_PROG
108	nancy	NANCY	Nancy	NancyFI_MGR
109	daniel	DANIEL	Daniel	DanielFI_ACCOUNT

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

مثال ٢ :

```
select employee_id,first_name, substr(first_name,2,3), length(first_name), instr(first_name,'n')
from employees;
```

EMPLOYEE_ID	FIRST_NAME	SUBSTR(FIRST_NAME,2,3)	LENGTH(FIRST_NAME)	INSTR(FIRST_NAME,'N')
100	Steven	tev	6	6
101	Neena	een	5	4
102	Lex	ex	3	0
103	Alexander	lex	9	6
104	Bruce	ruc	5	0
105	David	avi	5	0
106	Valli	all	5	0
107	Diana	ian	5	4
108	Nancy	anc	5	3
109	Daniel	ani	6	3

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

مثال ٣ :

```
select employee_id,first_name, trim('D' from first_name), lpad(first_name,10,'*'), rpad(first_name,10,'#')
from employees;
```

EMPLOYEE_ID	FIRST_NAME	TRIM('D'FROMFIRST_NAME)	LPAD(FIRST_NAME,10,'*')	RPAD(FIRST_NAME,10,'#')
100	Steven	Steven	****Steven	Steven####
101	Neena	Neena	*****Neena	Neena#####
102	Lex	Lex	*****Lex	Lex#####
103	Alexander	Alexander	*Alexander	Alexander#
104	Bruce	Bruce	*****Bruce	Bruce#####
105	David	avid	*****David	David#####
106	Valli	Valli	*****Valli	Valli#####
107	Diana	iana	*****Diana	Diana#####
108	Nancy	Nancy	*****Nancy	Nancy#####
109	Daniel	aniel	****Daniel	Daniel####

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

الدوال رقمية : Number Functions

وهي دوال تعمل مع البيانات الرقمية وتكون نتيجتها أرقاماً فقط . والجدول التالي يبين جميع الدوال الرقمية ووظيفة كل منها .

الدالة FUNCTION	وظيفتها
ROUND(column\expression,n)	<p>دالة تستخدم لقص عدد معين من الجزء العشري مع تقريب الأعداد إلى أقرب عدد عشري أو إلى عدد صحيح والحرف n يبين عدد الأرقام بعد العلامة العشرية ، وتوجد حالات للحرف n .</p> <ul style="list-style-type: none">• إذا كان ($n=0$) فإن التقريب يكون إلى أقرب عدد صحيح .• إذا كان ($n>0$) أي عدد موجب فإن التقريب يكون في الجزء بعد العلامة العشرية (الجزء العشري) .• إذا كان ($n<0$) أي عدد سالب فإن التقريب يكون في الجزء قبل العلامة العشرية (الجزء الصحيح) .

FUNCTION الدالة	وظيفتها
TRUNC(column\expression,n)	<p>دالة تستخدم لقص عدد معين من الجزء العشري بدون تقريب ، وأيضاً توجد حالات للحرف n .</p> <ul style="list-style-type: none"> • إذا كان (n=0) فإنه يتم قص الجزء العشري كله ويكون الناتج عدد صحيح . • إذا كان (n>0) أي عدد موجب فإن القص يكون في الجزء بعد العلامة العشرية (الجزء العشري) . • إذا كان (n<0) أي عدد سالب فإن القص يكون في الجزء قبل العلامة العشرية (الجزء الصحيح) .
MOD(m,n)	<p>دالة تستخدم لإيجاد باقي قسمة العدد m على العدد n .</p>

مثال (٤) :

```
SQL>SELECT ROUND(45.923,0),ROUND(45.923,2),ROUND(45.923,-1),ROUND(45.923,-2)
2 FROM dual ;
```

ROUND(45.923,0)	ROUND(45.923,2)	ROUND(45.923,-1)	ROUND(45.923,-2)
.....
46	45.92	50	0

الأعداد

مثال (٥) :

```
SQL> SELECT TRUNC(45.923,0),TRUNC(45.923,2),TRUNC(45.923,-1),TRUNC(45.923,-2)
2 FROM dual ;
```

TRUNC(45.923,0)	TRUNC(45.923,2)	TRUNC(45.923,-1)	TRUNC(45.923,-2)
45	45.92	40	0

مثال ٦ :

```
select employee_id, first_name, salary, mod (salary,300)
from employees;
```

EMPLOYEE_ID	FIRST_NAME	SALARY	MOD(SALARY,300)
100	Steven	24000	0
101	Neena	17000	200
102	Lex	17000	200
103	Alexander	9000	0
104	Bruce	6000	0
105	David	4800	0
106	Valli	4800	0
107	Diana	4200	0
108	Nancy	12000	0
109	Daniel	9000	0
More than 10 rows available. Increase rows selector to view more rows.			

دوال التاريخ (Date function)

الدالة FUNCTION	وظيفتها
SYSDATE	دالة تستخدم لعرض تاريخ النظام الموجود بجهاز الحاسب الآلي (تاريخ اليوم الحالي)
MONTHS_BETWEEN(date1,date2)	دالة لإيجاد عدد الأشهر بين تاريخين
ADD_MONTHS(date,n)	دالة لإضافة عدد معين من الأشهر على تاريخ معطى
NEXT_DAY(date,'day')	دالة لإيجاد تاريخ يوم معين بعد تاريخ معطى
LAST_DAY(date)	دالة لإيجاد آخر يوم في الشهر لتاريخ معطى
ROUND(date)	دالة تستخدم لتقريب التاريخ لأقرب شهر أو سنة
TRUNC(date)	دالة تستخدم لقص التاريخ لأقرب شهر أو سنة

المثال	النتيجة	المعنى
MONTHS_BETWEEN('01-SEP-95' , '11-JAN-94')	19.6774194	تم إيجاد عدد الأشهر بين التاريخين المعطيين
ADD_MONTHS('11-JAN-94' , 6)	'11-JUL-94'	تم إضافة ٦ أشهر على التاريخ المعطى
NEXT_DAY('01-SEP-95' , 'FRIDAY')	'08-SEP-95'	إيجاد تاريخ يوم الجمعة بعد التاريخ '01-SEP-95'
LAST_DAY('01-SEP-95')	'30-SEP-95'	إيجاد آخر يوم في شهر سبتمبر September
ROUND('25-JUL-95' , 'MONTH')	01-AUG-95	تم تقريب التاريخ المعطى إلى أقرب شهر وبما أن اليوم هو ٢٥ أي أكبر من ١٥ فتم التقريب إلى أول الشهر الذي يليه

المثال	النتيجة	المعنى
ROUND('25-JUL-95' , 'YEAR')	01-JAN-96	تم تقريب التاريخ المعطى إلى أقرب سنة وبما أن شهر (JULY) هو شهر (٧) فإنه تم التقريب إلى أول السنة التالية
TRUNC('25-JUL-95' , 'MONTH')	'01-JUL-95'	تم قص التاريخ المعطى بدون تقريب إلى أول يوم في الشهر
TRUNC('25-JUL-95' , 'YEAR')	'01-JAN-95'	تم قص التاريخ المعطى بدون تقريب إلى أول السنة الحالية

بعض الأمثلة على دوال التاريخ

مثال ٧ :

```
select sysdate from dual;
```

SYSDATE

08-FEB-13

select employee_id, hire_date, months_between(sysdate, hire_date) : مثال ٨
 from employees;

EMPLOYEE_ID	HIRE_DATE	MONTHS_BETWEEN(SYSDATE, HIRE_DATE)
100	17-JUN-87	307.735514486260454002389486260454002389
101	21-SEP-89	280.606482228195937873357228195937873357
102	13-JAN-93	240.864546744324970131421744324970131422
103	03-JAN-90	277.187127389486260454002389486260454002
104	21-MAY-91	260.606482228195937873357228195937873357
105	25-JUN-97	187.477449970131421744324970131421744325
106	05-FEB-98	180.122611260454002389486260454002389486
107	07-FEB-99	168.05809513142174432497013142174432497
108	17-AUG-94	221.735514486260454002389486260454002389
109	16-AUG-94	221.767772550776583034647550776583034648
More than 10 rows available. Increase rows selector to view more rows.		

select employee_id, hire_date, round(months_between(sysdate, hire_date))
 from employees;

EMPLOYEE_ID	HIRE_DATE	ROUND(MONTHS_BETWEEN(SYSDATE, HIRE_DATE))
100	17-JUN-87	308
101	21-SEP-89	281
102	13-JAN-93	241
103	03-JAN-90	277
104	21-MAY-91	261
105	25-JUN-97	187
106	05-FEB-98	180
107	07-FEB-99	168
108	17-AUG-94	222
109	16-AUG-94	222
More than 10 rows available. Increase rows selector to view more rows.		

مثال ٩ :
`select employee_id, hire_date, add_months(hire_date,3), last_day(hire_date)`
`from employees;`

EMPLOYEE_ID	HIRE_DATE	ADD_MONTHS(HIRE_DATE,3)	LAST_DAY(HIRE_DATE)
100	17-JUN-87	17-SEP-87	30-JUN-87
101	21-SEP-89	21-DEC-89	30-SEP-89
102	13-JAN-93	13-APR-93	31-JAN-93
103	03-JAN-90	03-APR-90	31-JAN-90
104	21-MAY-91	21-AUG-91	31-MAY-91
105	25-JUN-97	25-SEP-97	30-JUN-97
106	05-FEB-98	05-MAY-98	28-FEB-98
107	07-FEB-99	07-MAY-99	28-FEB-99
108	17-AUG-94	17-NOV-94	31-AUG-94
109	16-AUG-94	16-NOV-94	31-AUG-94

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

مثال ١٠ :

```
select employee_id, hire_date, next_day(hire_date, 'Friday')  
from employees;
```

EMPLOYEE_ID	HIRE_DATE	NEXT_DAY(HIRE_DATE, 'FRIDAY')
100	17-JUN-87	19-JUN-87
101	21-SEP-89	22-SEP-89
102	13-JAN-93	15-JAN-93
103	03-JAN-90	05-JAN-90
104	21-MAY-91	24-MAY-91
105	25-JUN-97	27-JUN-97
106	05-FEB-98	06-FEB-98
107	07-FEB-99	12-FEB-99
108	17-AUG-94	19-AUG-94
109	16-AUG-94	19-AUG-94

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

دوال التحويل : Conversion Functions

يوجد أنواع كثيرة من البيانات (DATATYPE) يمكن تخزينها داخل الجداول ومن أهمها بيانات رقمية (NUMBER) وبيانات حرفية (CHARACTER) وبيانات التاريخ (DATE). ودوال التحويل تقوم بتحويل البيانات من نوع إلى آخر ، وهي عبارة عن ثلاثة أنواع والجدول التالي يبين هذه الأنواع ووظيفة كل منها :

الدالة FUNCTION	وظيفتها
TO_CHAR (DATE/NUMBER , 'fmt')	تستخدم هذه الدالة لتحويل البيانات الرقمية أو بيانات التاريخ إلى بيانات حرفية بشكل معين (FORMAT) حسب الطلب fmt .
TO_DATE (CHAR , 'fmt')	تستخدم لتحويل البيانات الحرفية إلى بيانات من نوع التاريخ بشكل معين (FORMAT) حسب الطلب fmt .
TO_NUMBER (CHAR , 'fmt')	تستخدم لتحويل البيانات الحرفية إلى بيانات رقمية بشكل معين (FORMAT) حسب الطلب fmt .

دالة التحويل TO_CHAR .

أولاً : استخدام هذه الدالة لتحويل بيانات التاريخ DATE إلى بيانات حرفية .

الشكل العام :

```
TO_CHAR(DATE, 'fmt')
```

مثال ١١ :

```
select sysdate ,to_char(sysdate,'dd/mm/yyyy')  
from dual;
```

SYSDATE	TO_CHAR(SYSDATE,'DD/MM/YYYY')
08-FEB-13	08/02/2013

الجدول التالي يوضح بعض الأشكال التي يمكن استخدامها عند تحويل التاريخ إلى بيانات حرفية بشكل معين .

YYYY	إظهار السنة كاملة بالأرقام (القرن + السنة) مثل 2004 .
YY	إظهار رقمين فقط من السنة 04 .
YEAR	إظهار السنة كاملة كتابة (TWO THOUSAND FOUR) وحالة أحرف الكتابة تتوقف على حالة أحرف كلمة YEAR .
MM	إظهار الشهر في شكل رقمين 01 .
MONTH	إظهار الشهر كتابة (JANURY) .
MON	إظهار الثلاث حروف الأولى من اليوم (JAN) .
DAY	إظهار اليوم كاملاً كتابةً (FRIDAY) .
HH12:MI:SS AM	إظهار الوقت بنظام ١٢ ساعة وهل هو صباحاً أم مساءً (04:30:50 PM)

مثال ١٢ :

```
select employee_id,to_char(hire_date,'day "OF" month yyyy hh12:mi:ss am ')  
from employees;
```

EMPLOYEE_ID	TO_CHAR(HIRE_DATE,'DAY"OF"MONTHYYYYHH12:MI:SSAM')
100	wednesday OF june 1987 12:00:00 am
101	thursday OF september 1989 12:00:00 am
102	wednesday OF january 1993 12:00:00 am
103	wednesday OF january 1990 12:00:00 am
104	tuesday OF may 1991 12:00:00 am
105	wednesday OF june 1997 12:00:00 am
106	thursday OF february 1998 12:00:00 am
107	sunday OF february 1999 12:00:00 am
108	wednesday OF august 1994 12:00:00 am
109	tuesday OF august 1994 12:00:00 am
More than 10 rows available. Increase rows selector to view more rows.	

دالة التحويل TO_CHAR .

ثانياً : استخدم هذه الدالة لتحويل البيانات الرقمية NUMBER إلى بيانات حرفية .

الشكل العام :

```
TO_CHAR(NUMBER, 'fmt')
```

مثال ١٣ :

```
select employee_id,to_char(salary,'$99,999') salary  
from employees;
```

EMPLOYEE_ID	SALARY
100	\$24,000
101	\$17,000
102	\$17,000
103	\$9,000
104	\$6,000
105	\$4,800
106	\$4,800
107	\$4,200
108	\$12,000
109	\$9,000
More than 10 rows available. Increase rows selector to view more rows.	

الجدول التالي يوضح بعض الرموز التي يمكن استخدامها عند تحويل البيانات الرقمية إلى بيانات حرفية لتظهر بشكل معين .

9	عدد تكرار هذا الرقم يمثل عدد الخانات التي تظهر ، مثلاً عندما نكتب (99) معناها ظهور رقمين وهكذا .
099	يعني ظهور الرقم وقبله صفر .
990	يعني ظهور صفراً إذا كانت القيمة معدومة
\$99	إظهار علامة \$ قبل الرقم .
.	إظهار العلامة العشرية .
,	إظهار فواصل بين كل ثلاثة أرقام (فاصلة الألوف) .
MI	إظهار علامة السالب (-) يمين الرقم إذا كان سالباً .

دالة التحويل TO_DATE .

الشكل العام :

```
TO_DATE(CHAR , 'fmt')
```

مثال ١٤ :

```
select to_date('may 22,1988','month dd,yyyy')  
from dual;
```

```
TO_DATE('MAY22,1988','MONTHDD,YYYY')
```

```
22-MAY-88
```

THE END
