الفصل السادس

استخدام الدوال المعرفة مسبقا

Built-in Functions

الدوال في **SQL** *عبارة عن برامج صغيرة تمكنك من أداء الكثير من الوظائف التي تحتاجها أثناء تعاملك مع قواعد البيانات ولذلك فالعبارة* **Built-in Functions** *تعني أن هذه الدوال موجودة داخل* **SQL** *وكل ما عليك أنك تستدعيها للتنفيذ .*

بالانتهاء من هذا الفصل ستكتسب المعارف وتتدرب على المهارات التى تجعلك قادرا على:

- *ما هي الدوال* ***Built-In Functions***
- *الدوال الإحصائية*
- *دوال التاريخ والوقت*
- *الدوال الرياضية*
- *دوال تقريب الأرقام*
- *دوال التعامل مع السلاسل الحرفية*
- *دوال متنوعة*

تستخدم معظم لغات البرمجة الدوال الجاهزة أو **Built-in Functions** وهي عبارة عن برامج جاهزة يتم استدعائها لأداء وظيفة معينة ولذلك يتم ترجمتها أحيانا ب"الوظائف الجاهزة" وهي تسهل التعامل مع البيانات وتمكنك من الحصول علي النتائج بسرعة كما في حالة الرغبة في الحصول علي مجموع السجلات أو متوسطاتها أو تحويل الحروف الصغيرة إلي حروف كبيرة ...الخ .

فيما يلي سوف نقسم الدوال ألي مجموعات متشابهه ليسهل عليك دراستها وتذكرها .

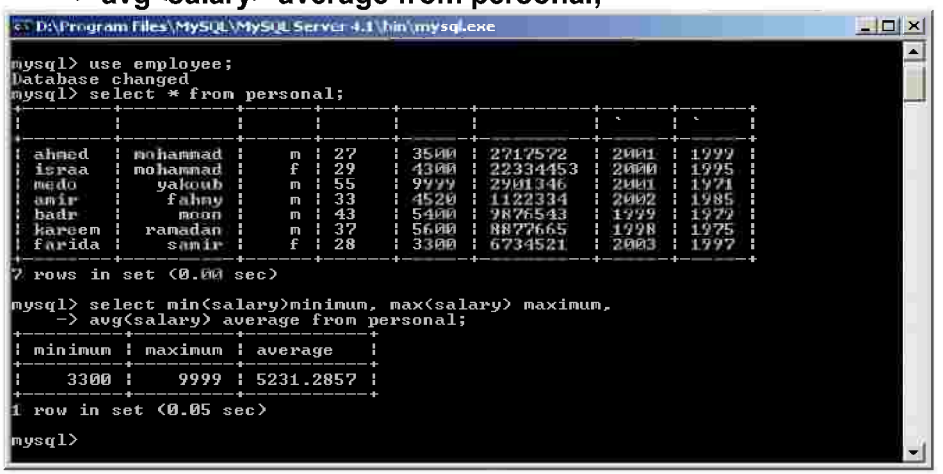
الدوال الإحصائية :

يمكنك استخدام جملة SELECT لتنفيذ بعض العمليات الحسابية علي بيانات الجدول باستخدام بعض الدوال Functions . يوضح الجدول التالي الدوال الإحصائية التي يمكن استخدامها مع جملة SELECT . ووظيفة كل منها :

الدالة	وظيفتها
AVG	تحسب المتوسط الحسابي لمجموع القيم الموجودة في حقل معين . فإذا اشتملت جملة SELECT علي الاختيار WHERE فإنها تحسب المتوسط الحسابي للسجلات التي ينطبق عليها الشرط الموجود في الاختيار WHERE فقط
COUNT	تحسب عدد السجلات في الجدول أو عدد السجلات التي ينطبق عليها الشرط المحدد بالاختيار WHERE
MIN	تحسب القيمة الصغرى لمجموعة قيم موجودة في حقل معين أو تلك التي ينطبق عليها الشرط WHERE
MAX	تحسب القيمة الكبرى لمجموعة قيم موجودة في حقل معين أو تلك التي ينطبق عليها الشرط WHERE
SUM	تحسب مجموع القيم الموجودة في حقل معين أو تلك التي ينطبق عليها الشرط WHERE
STDDEV	تستخدم للحصول علي الانحراف المعياري لبيانات حقل في جميع السجلات أو تلك التي ينطبق عليها الشرط WHERE

المثال التالي يحسب القيمة الصغرى والكبرى ومتوسط مرتبات الموظفين في جدول Personalانظر شكل ٦ – ١

```
mysql > use employee ;
Database changed
Mysql > select * from personal ;
Mysql> select min <salary>minimum, max<salary> maximum,
    -> avg<salary> average from personal;
```



شكل ٦-١

المثال التالي يمكنك من الحصول على مجموع حقل salary

```
mysql > select sum ( salary ) total from personal ;
```

total
27500

1 row in set (0.10 sec)

دوال التاريخ والوقت :

فيما يلي نشرح الدوال التي تتعامل مع التاريخ والوقت ولا يستغنى أي مبرمج في أي نظام عن الحاجة لمعرفة الوقت و التاريخ الحاليان .أو إضافة أو طرح يوم أو شهر أو مدة زمنية لحقل تاريخي للحصول على نتائج معينة . تابع معنا الشرح التالي :

الدالة Last_Day

هل تعرف كيف تحسب السنة الكبيسة ؟ وهل تعرف عدد الأيام في كل شهر من الأشهر الميلادية ومتى يكون فبراير ٢٨ ومتى يكون ٢٩ يوم ؟ وما هي الشهور ذات الثلاثين يوما والأخرى ذات الواحد وثلاثون يوما ؟ تستطيع باستخدام query في Oracle أن تعرف آخر يوم في الشهر لأي شهر تريد باستخدام الوظيفة Last_day ، أما بيني وبينك فان السنة الكبيسة هي التي تقبل القسمة على ٤ مثل سنة ٢٠٠٤ فهي تقبل القسمة على ٤ دون باقٍ ، هذه السنة يكون عدد أيامها ٣٦٦ يوما لأن شهر فبراير فيها ٢٩ يوما .

```
SQL> SELECT * FROM TASKS;
task          startdate          enddate
delivery      01-APR-2004        28-APR-2004
installaton   29-APR-2004        15-MAY-2004
testing       01-MAY-2004        15-JUN-2004
training      01-JUL-2004        31-JUL-2004
OJT           01-AUG-2004        31-DEC-2004
handover      05-JAN-2005        20-JAN-2005
```

نحاول الحصول الآن على آخر يوم في كل شهر من حقل startdate ولنر النتيجة :

Oracle

```
SQL> SELECT STARTDATE, LAST_DAY(STARTDATE) LAST
FROM TASKS;

startdate          last
01-Feb-2004        29-Feb-2004
29-Apr-2004        30-Apr-2004
01-May-2004        31-May-2004
01-Jul-2004        31-Jul-2004
01-Aug-2004        31-Aug-2004
05-Jan-2005        31-Jan-2005
16-Feb-2005        28-Feb-2005
7 ROWS SELECTED.
```

نلاحظ أن فبراير ٢٠٠٤ عبارة عن ٢٩ يوم بينما فبراير ٢٠٠٥ هو ٢٨ يوم فقط

الدالة Months_Between

تقوم بعملية طرح بين حقلين من حقول التاريخ والنتيجة تكون عدد الأشهر ما بين الحقلين ، مثال :

Oracle

```
SQL> SELECT TASK, STARTDATE,
        >ENDDATE, MONTHS_BETWEEN(ENDDATE, STARTDATE)
        >DURATION FROM TASKS;

TASK        STARTDATE       ENDDATE         DURATION
ONE         15-MAY-2004     12-NOV-2004     6
```

إضافة تاريخ أو شهر للتوقيت ADD_MONTHS/ ADD_DATE

هي في Oracle الوظيفة ADD_Months بينما في mysql تكتب هكذا date_add نفترض أننا نريد اضافة شهرين لتاريخ انهاء المهمة Task Enddate فلنبدأ بكتابة query في نظام قواعد البيانات Oracle هكذا:

Oracle

```
SQL>SELECT TASK, STARTDATE, ENDDATE ORIGINAL,
        2 ADD_MONTHS(ENDDATE,2) NEW,
        3 FROM TASKS,

TASK            STARTDATE       ORIGINAL        NEW
--------------- --------------- --------------- ---------------
DELIVERY        1-APR-2004      28-APR-2004     28-JUN-2004
INSTALLATON     29-APR-2004     15-MAY-2004     15-JUL-2004
TESTING         1-MAY-2004      15-JUN-2004     15-AUG-2004
TRAINING        1-JUL-2004      31-JUL-2004     31-SEP-2004
OJT             1-AUG-2004      31-DEC-2004     28-FEB-2005
HANDOVER        5-JAN-2005      20-JAN-2005     2-MAR-2004
6 rows selected.
```

بينما تكتب هكذا في MYSQL :

```
MySQL>select task, startdate, enddate original,
>date_add(enddate, interval 2 month)
```

>new from tasks;

task	startdate	original	new
delivery	2004-04-01	2004-04-28	2004-06-28
installaton	2004-04-29	2004-05-15	2004-07-15
testing	2004-05-01	2004-06-15	2004-08-15
training	2004-07-01	2004-07-31	9/31/2004
OJT	2004-08-01	2004-12-31	2005-02-28
handover	2005-01-05	2005-01-20	2005-03-20

6 rows in set (0.07 sec)

أحيانا أعمد إلي كتابة جملة **ORACLE SQL** بالحروف الكبيرة وجملة **MySQL** بالحروف الصغيرة للتمييز فقط !

دالة إظهار تاريخ اليوم

تاريخ اليوم أو ما يطلق عليه **Current Date** نحتاجه غالبا ليكون جزء من أي تقرير تتم طباعته من قواعد البيانات ، ويتم تخزين التاريخ عند تثبيت نظام قواعد البيانات لأول مرة ويقوم برنامج التثبيت بأخذ التاريخ من نظام التشغيل الذي يتم تثبيت نظام قواعد البيانات عليه مثل **Windows, Unix, Linux** ولنأخذ أمثلة تاريخ اليوم من ثلاثة من قواعد البيانات :

Sybase
SELECT GETDATE();
النتيجة
2005-02-08 09:10:20.083

MySQL
SELECT current_date;
النتيجة
SELECT current_date (1 Records)

current_date
2005-02-08

generated 2/8/2005 9:14:14 AM by MySQL-Front 1

Oracle
SELECT SYSDATE FROM DUAL

النتيجة

08-FEB-2005

تأمل معي هذا المثال

```
mysql>select now() tarekh,
>month(now()) month, day(now()) today,
>year(now()) year, hour(now()) clock,
>minute(now()) minute;
```

Tarekh	Month	Today	Year	Clock	minute
2004-12-15 19:59:45	12	15	2004	19	59

6 fields, 1 records

في هذا المثال حصلنا على تاريخ اليوم now() وسميناه tarekh ثم حصلنا على الشهر ثم اليوم ثم السنة ثم الساعة ثم الدقيقة كل واحدة في حقل بمفرده ، راجع الجملة السابقة وقارن مع النتيجة التي حصلت عليها !

اتفقت Sybase مع ميكروسوفت على استخدام Microsoft SQL Server في كلا نظامي قواعد البيانات حيث تتبنى Sybase تقنية SQL Server وقد أصدرت Microsoft ما يسمى Transact-SQL وهو عبارة عن Microsoft SQL وأصبح هو المستخدم حاليا من كل من Sybase و Microsoft !!!!

إضافة قيم للتاريخ

MySQL
```
select hire_date,
date_add(hire_date, interval 1 month)
new_date
from hr;
```

hire_date	NEW_date
1999-10-04	1999-11-04
2000-02-12	2000-03-12
2001-02-28	2001-03-28
2000-08-03	2000-09-03
1996-09-01	1996-10-01
1995-11-08	1995-12-08
1998-07-13	1998-08-13
2003-12-22	2004-01-22

في هذا المثال قمنا بإضافة شهر واحد للتاريخ وقمنا بطباعة التاريخ الموجود داخل الجدول والتاريخ بعد التعديل new_date طبعا باستخدام MySQL

سنستخدم المثال السابق مع Microsoft Transact-SQL ونفس المثال يمكن تطبيقه مع Sybase بدون تعديلات :

Microsoft SQL
```
SELECT HIRE_DATE,
DATEADD(MONTH,1, HIRE_DATE) NEWDATE
FROM HR
```

hire_date	new_date
1999-10-04	1999-11-04
2000-02-12	2000-03-12
2001-02-28	2001-03-28
2000-08-03	2000-09-03
1996-09-01	1996-10-01
1995-11-08	1995-12-08

```
1998-07-13      1998-08-13
2003-12-22      2004-01-22
```

سنستخدم المثال نفسه مع Oracle

```
Oracle SQL
SELECT HIRE_DATE, ADD_MONTHS(HIRE_DATE, 1)
NEW_DATE FROM HR;
HIRE_DATE    NEW_DATE
-----------  -----------
14-OCT-99    14-NOV-99
12-FEB-00    12-MAR-00
28-FEB-01    28-MAR-01
13-AUG-00    13-SEP-00
11-SEP-96    11-OCT-96
18-NOV-95    18-DEC-95
13-JUL-98    13-AUG-98
22-DEC-03    22-JAN-04
```

استخراج الفرق بين تاريخين :

نحتاج أحيانا إلى معرفة الفرق بين تاريخ اليوم مثلا وتاريخ ميلاد شخص أو تاريخ أخر ، في Microsoft SQL كما في Sybase نستخدم لهذا الغرض الدالة ( )DATEDIFF ولكن النتيجة تخرج إما بعدد الشهور أو الأيام أو السنون كما في الأمثلة التالية:

المثال الأول يعطينا فرق السنوات :

```
SELECT Birth_Date, DATEDIFF(YEAR, Birht_Date, GETDATE())
Years_deferent
FROM HR
 Birth_Date    Years_Deferent
 1970-12-10    35
 1971-01-08    34
 1971-02-18    34
 1971-03-28    34
 1971-04-11    34
 1971-05-22    34
```

1971-06-30	34
1971-12-02	34
1971-07-04	34
1961-09-06	44

وهذا المثال يعطينا فرق الأشهر :

```
SELECT birth_date, DATEDIFF(MONTH, birth_date, GETDATE())
Months_Deferent
FROM HR
```

Birth_Date	Months_Deferent
1970-12-10	410
1971-01-08	409
1971-02-18	408
1971-03-28	407
1971-04-11	406
1971-05-22	405
1971-06-30	404
1971-12-02	398
1971-07-04	403
1961-09-06	521

وهذا المثال يعطينا الفرق بين تاريخ اليوم وتاريخ الميلاد بالأيام :

```
SELECT birth_date, DATEDIFF(DAY, birth_date, GETDATE())
days_Deferent
FROM HR
```

Birth_Date	Days_Deferent
1970-12-10	12480
1971-01-08	12451
1971-02-18	12410
1971-03-28	12372
1971-04-11	12358

1971-05-22	12317
1971-06-30	12278
1971-12-02	12123
1971-07-04	12274
1961-09-06	15862

تحويل التاريخ إلى بيانات حرفية

في هذا المثال سوف نقوم بطباعة اسم الشهر كجزء من تاريخ الميلاد birth_date باستخدام نظام قواعد البيانات SQL :

Microsoft SQL/Sybase

```
SELECT birth_date = datename(month, birth_date)
FROM HR
```

```
birth_date
December
January
February
March
April
May
June
December
July
September
```

تم طباعة اسم الشهر في حقل تاريخ الميلاد

وهذا مثال من نظام قواعد البيانات Oracle وفيه يتم طباعة تاريخ الميلاد كتابة :

Oracle

```
SELECT birth_date, to_char(birth_date, dd, month, yyyy) 'BIRTH
DAY'
FROM HR;
```

birth_date	BIRTH DAY
1977-12-11	11 December 1977
1972-05-12	12 May 1972
1979-10-25	25 October 1979
1970-09-13	13 September 1970
1965-06-15	15 June 1965
1967-06-15	15 June 1967
1970-03-22	22 March 1970
1980-04-01	1 April 1960

دوال التاريخ الأخرى :

نورد يما يلي جدول يبين دوال التاريخ والوقت . استخدم هذا الجدول في ضوء الشرح الذي تقدم للتعامل مع الوقت راجع الشكل العام للتاريخ في كل نظام لتتعرف على كيفية استخدامه

RDBMS	الدالة	استخدامها
SQL	DATEADD	إضافة قيمة للتاريخ
	DATEDIFF	استخراج الفرق بين تاريخين
	DATENAME	طباعة اسم اليوم
	DATEPART	استخراج قيمة رقمية للتاريخ
	DAY	استخراج اسم اليوم
	GETDATE	طباعة التاريخ كما هو في نظام قواعد البيانات
	MONTH	طباعة الشهر
	YEAR	استخراج السنة
MySQL	ADDDATE	إضافة قيمة للتاريخ
	ADDTIME	إضافة إلي الوقت

RDBMS	الدالة	استخدامها
	CONVERT_TZ	تحويل التاريخ والوقت بين بلدان العالم
	CURDATE()	تاريخ اليوم
	CURRENTDATE()	تاريخ اليوم
	CURTIME()	الوقت الآن عبارة عن ثواني ودقائق وساعة
	CURRENTTIME()	الوقت الآن عبارة عن ثواني ودقائق وساعة
	DATE(expr)	استخراج التاريخ
	DATEDIFF()	عدد الأيام بين تاريخين
	DATE_ADD	إضافة قيمة إلي التاريخ
	DATE_SUB	الحصول علي جزء من تاريخ
	DATE_FORMAT	التحكم في شكل التاريخ
	DAY	رقم اليوم في الشهر
	DAYNAME	اسم اليوم في الأسبوع الجمعة مثلا
	DAYOFMONTH	اليوم من الشهر
	DAYOFWEEK	اليوم من الأسبوع
	DAYOFYEAR	اليوم من السنة مثلا يوم ٥٨ هو يوم ٢٧ فبراير
	EXTRACT	التحكم في أجزاء من التاريخ
	FROM_DAYS(N)	أيام من تاريخ معين
	HOUR(time)	عدد الساعات
	LAST_DAY(date)	آخر يوم من الشهر الحالي

RDBMS	الدالة	استخدامها
	NOW()	اليوم مع الوقت
Oracle	NEXT_DAY	اليوم التالي من الأسبوع
	MONTHS_BETWEEN	عدد الشهور بين تاريخين

لمزيد من التفاصيل راجع كتب نظام إدارة قواعد البيانات الذي تستخدمه للتعرف على الشكل العام لكل دالة .

الدوال الرياضية (الحسابية) :

تقوم الدوال الرياضية بتنفيذ بعض العمليات الحسابية علي بيانات الجدول ، وفيما يلي نوضح كيفية التعامل مع أهم هذه الوظائف :

القيمة المطلقة Absolute Values ABS()

تستخدم الدالة (ABS)للحصول على القيمة المطلقة لحقل معين ، مثال :

Oracle
```
Sql> select * from numbers;

x          y
3.1415     4
-45        -787
5          9
-67.667    42
15         55
-7.2       5.3
6 rows selected.
```

نحاول الآن أن نحصل على القيمة الحقيقية (المطلقة) للحقل x

Oracle
```
SQL> SELECT ABS(X) "Absolute_Value"
FROM NUMBERS;
```

```
Asolute_Value
-------------------
3.1415
     45
      5
67.667
     15
    7.2
6 rows selected.

Mysq l> select abs(x) "Asolute Value"
      > from numbers;
```

Asolute_Value
3.1415
45
5
67.667
15
7.2

6 row in set (0.8 sec)

دوال تقريب الأرقام CEIL and Floor

الوظيفة الأولى **CEIL** تقوم بالتقريب الى أقرب رقم صحيح ، بينما الوظيفة الثانية **Floor** تقوم بالتقريب الى أكبر رقم صحيح أي أنها تقريبا تحذف الكسور دون تقريب . ويسأل الرازي عن ذلك ولماذا ؟ لاحظ أنها تكتب في أوراكل **CIEL** بينما تكتب في **mysql** كاملة **CEILING** كما يحلو لبرنامج الاكسل أن يعمل :

والآن لنجرب الوظيفتين **CEIL and FLOOR** ونر النتيجة :

Oracle

```
Select x, CEIL(x) ceiling, FLOOR(x) floor from numbers;
```

x	Ceiling	Floor
3.1415	4	3

-45	1	-45
5	5	5
-67.667	1	-68
15	15	15
-7.2	1	-8

Mysql> select x, ceiling(x) ceiling, floor(x) floor from numbers;

x	Ceiling	Floor
3.1415	4	3
-45	1	-45
5	5	5
-67.667	1	-67
15	15	15
-7.2	1	-8

6 row in set (0.12 sec)

دالة باقي القسمة MOD

تعطى هذه الدالة الباقي بعد قسمة الرقم الأول على الرقم الثاني . المثال التالي يظهر بيانات حقلي **X,Y** من جدول **NUMBERS** ثم يظهر باقي قسمة الأول علي الثاني :

SQL> SELECT X, Y, MOD(X, Y) reminder FROM NUMBERS;

x	y	Reminder
03.1415	4.000	3.1415
-45.0000	0.707	0.2480
05.0000	9.000	5.0000
-67.6670	42.000	16.3330
15.0000	55.000	15.0000
-7.2000	5.300	3.4000

6 ROWS SELECTED.

دالة الجذر التربيعي SQRT

نلاحظ أن استخراج الجذر التربيعي يتم على الأرقام الموجبة فقط ولا يتم على القيم السالبة ، لهذا سوف نحاول استخراج الجذر التربيعي لكل من **x** و **y** كالتالي :

SQL> SELECT * , SQRT(ABS(X)) FROM NUMBERS;

هنا استخدمنا ABS(X) لكي نتلافى مشكلة القيم السالبة . والنتيجة هي :

x	SQRT (ABS(X))
3.1415	1.77242771
-45	6.70820393
5	2.23606798
-67.667	8.22599538
15	3.87298335
-7.2	2.68328157

الدالة **EXP**:

ترفع رقم إلي أس :

```
SQL>SELECT X, EXP(X) EXP FROM NUMBERS;
```

x	EXP
3.1415	23.1385
-45	2.9E-20
5	148.413
-67.667	4.1E-30
15	3269017
-7.2	0.00075

دوال التعامل مع السلاسل الحرفية

الدالة **CHR**

تعطي الحرف المقابل لكود ASCII في الجدول الشهير ASCII المستخدم في معظم الحواسب ، لنأخذ مثالا :

```
SQL> SELECT * FROM LETTERS;
```

FIRTSNAME	LASTNAME	LTR	CODE
AHMED	ALI	A	32
ASSAD	HELMY	J	67
RANIA	MOKHTAR	C	65
HAJAR	MOHAMMAD	M	87
HEBA	MOHAMMAD	A	77
ISRAA	MOHAMMAD	G	52

الحقل CODE عبارة عن قيم لحروف ، تعال معي نكتشف هذه الحروف :

```
SQL> SELECT CODE, CHR(CODE) FROM LETTERS;
 CODE    CHR(CODE)
      32
      67  C
      65  A
      87  W
      77  M
      52  4
```

نلاحظ أن القيمة ٣٢ مقابلها شاغر ، في الحقيقة فان قيمة ٣٢ هي مسافة Space أو لنقل Blank .

الوظيفة CHR تكتب هكذا في أوراكل بينما في MYSQL تكتب CHAR كما في EXCEL

دالة ضم الحقول CONCAT

لكي نضم محتويات حقلين ليكونا حقلا واحدا يضم القيمتين بفاصل أو بدون فاصل فاننا نستخدم الدالة CONCAT وتستخدم نفس الاستخدام في Oracle SQL و أيضا Mysql، حاول التدقيق في الصياغة حيث سوف نضم الحقلين ثم سنضع في المرة التالية مسافة بين الحقلين ثم في المرة الأخيرة سنضع فاصلة بين الحقلين .المثال التالي يدمج بيانات حقل FIRSTNAME وحقل LASTNAME ويضع الناتج في عمود باسم CONCATENATED

```
SQL> SELECT firstname, lastname,
> CONCAT(FIRSTNAME, LASTNAME) concatenated
FROM LETTERS;
 FIRTSNAME   LASTNAME    Concatenated
 AHMED       ALI         AHMEDALI
 ASSAD       HELMY       ASSADHELMY
 RANIA       MOKHTAR     RANIAMOKHTAR
 HAJAR       MOHAMMAD    HAJARMOHAMMAD
 HEBA        MOHAMMAD    HEBAMOHAMMAD
 ISRAA       MOHAMMAD    ISRAAMOHAMMAD
```

وفيما يلي ندمج بيانات الحقلين بعد وضع مسافة بينهما :

```
SQL> SELECT firstname, lastname,
>CONCAT(FIRSTNAME," ", LASTNAME) concatenated
>FROM LETTERS;
```

FIRTSNAME	LASTNAME	Concatenated
AHMED	ALI	AHMED ALI
ASSAD	HELMY	ASSAD HELMY
RANIA	MOKHTAR	RANIA MOKHTAR
HAJAR	MOHAMMAD	HAJAR MOHAMMAD
HEBA	MOHAMMAD	HEBA MOHAMMAD
ISRAA	MOHAMMAD	ISRAA MOHAMMAD

أما المثال التالي فيستخدم فاصلة بين الحقلين بعد دمجهما :

```
SQL> SELECT firstname, lastname,
>CONCAT(FIRSTNAME,",", LASTNAME) concatenated
>FROM LETTERS;
```

FIRTSNAME	LASTNAME	Concatenated
AHMED	ALI	AHMED,ALI
ASSAD	HELMY	ASSAD,HELMY
RANIA	MOKHTAR	RANIA,MOKHTAR
HAJAR	MOHAMMAD	HAJAR,MOHAMMAD
HEBA	MOHAMMAD	HEBA,MOHAMMAD
ISRAA	MOHAMMAD	ISRAA,MOHAMMAD

دالة تحويل الحروف كلها الى حروف كبيرة **UPPER**

```
SQL> SELECT FIRSTNAME, UPPER(FIRSTNAME) FROM LETTERS;
```

Firtsname	UPPER (FIRTSNAME)
Ahmed	AHMED
Assad	ASSAD
Rania	RANIA
Hajar	HAJAR
Heba	HEBA
Israa	ISRAA

دالة تحويل الحروف كلها الى حروف صغيرة **LOWER**

```
SQL> SELECT FIRSTNAME, LOWER(FIRSTNAME) FROM LETTERS;
 FIRTSNAME   LOWER(FIRSTNAME)
 AHMED       ahmed
 ASSAD       assad
 RANIA       rania
 HAJAR       hajar
 HEBA        heba
 ISRAA       israa
6 ROWS SELECTED.
```

الدالة **INSTR**

عندما نستخدم هذه الدالة مع **QUERY** فان النتيجة تكون رقم يشير الى ترتيب الحرف داخل حقل . تستخدم الدالة أربعة معاملات الأول الحقل الذي سنبحث فيه والثاني الحرف الذي سنبحث عنه والثالث ترتيب الحرف الذي سيبدأ عنده البحث والرابع عدد الحروف التي سيكتبها ، فمثلا نريد أن نعرف ترتيب حرف **a** داخل حقل **firstname** على أن يبدأ الاختبار من الحرف الثاني تأمل معي محتويات الحقل :

```
Firtsname
Fahmy
Mustafa
Husny
Fuad
```

نكتب الدالة هكذا :

```
SQL > SELECT FIRSTNAME, INSTR(FIRSTNAME, 'a', 2, 1)
      >from letters;
```

في هذا المثال نبحث عن ترتيب حرف **a** داخل حقل **FIRSTNAME** علي أن يبدأ البحث ابتداء من الحرف الثاني في الكلمة .

```
firstname          instr(firstname, 'a', 2, 1)
 Fahmy              2
 Mustafa            5
 Husny              2
 Fuad               3
```

الدالة **Length**

تعطينا طول الحقل ، مثلا

```
mysql> select firstname, length(firstname) from letters;
```

Firstname	Length(firstname)
Fahmy	5
Mustafa	7
Husny	5
Fuad	4

الدالة **SUBSTR**

تستخدم لاستخراج جزء من عبارة معينة . ولابد أن تحدد لها ثلاثة معاملات الأول العبارة النصية التي ستستخرج جزء منها ، والثاني الحرف الذي ستبدأ من عنده ، والثالث عدد الأحرف التي ستستخرجها .

المثال التالي يستخرج ثلاثة حروف فقط من حقل الاسم الأول ابتداء من الحرف الثاني في الحقل

```
SQL > SELECT FIRSTNAME , SUBSTR ( FIRSTNAME,2,3 )
      > FROM LETTERS
```

FIRSTNAME	SUB
Fahmy	ahm
Mustafa	ust
Husny	usn
Fuad	uad

وفي MYSQL تستخدم MID بدلا من SUBSTR هكذا :

```
MYSQL > Select Firstname , mid ( Firstname,2,3 )
```

Firstname	mid (Firstname,2,3)
Fahmy	ahm
Mustafa	ust
Husny	usn
Fuad	uad