مقدمة

السبب الذي جعلني اشرع في كتابة هذا الكتاب ما لاحظته من افتقار الساحة لمثل هذه التطبيقات مما جعلني اعتمد على الله اولا ثم على نفسي وبعض المراجع الاجنبية في تعلم هذا الامر والمضي قدما حتى اصبحت بحمد الله من مبرمجي قواعد بيانات اكسس

عن هذا الكتاب :

هذا الكتاب ليس للمبتدئين ولكن للعاملين على برنامج اكسس والذين سبق لهم التعامل مع الاستعلامات سواء الاستعلامات الهيكلية او استعلامات SQL

اهداء :

الى روح ابي الذي زرع فينا العطاء ولروح الاستاذ محمد انور رحمه الله الذي طالما تعلمت منه اسالكم الدعاء لي بحسن الخاتمة والدعاء لابي وللأستاذ محمد انور بان يسكنه الله جناته

وهذه النسخة هديه خاصة لموقع كتب والذي احمد الله ان هناك حتى الان من لديه حب العطاء بدون مقابل ولطالما تعلمت منه ومن الكتب التي قرأتها فيه وهذا ما دفعني لرد بعض الجميل لهذا الموقع وللقائمين عليه ولطلاب العلم ويسعدني التواصل مع القراء على البريد الاليكتروني

Khaled201167@yahoo.com

او هاتفيا على 01192724811 اتصالات مصر

SQL ميكروسوفت أكسس

تعتبر SQL لغة الاستعلامات هامة جدا لأي مطور قواعد بيانات إذ أنه بواسطتها يمكنه تخصيص استعلام والاستفادة بأهم ميزات تشغيل قواعد البيانات

كيفية عمل استعلام SQL

بعد فتح مشروع جديد في قواعد بيانات أكسس اختر استعلام Query ثم جديد New واختر إنشاء استعلام واختر طريقة عرض التصميم design واختر الغاء الامر cancel ثم من شريط الادوات اختر عرض view ثم اكتب العبارات التي ترغب بها SQL

هناك ثلاث أنواع اساسية من العبارات في SQL العبارتان الاجباري والذي لا يستقيم الاستعلام الا بهما هما SELECT , FROM

والعبارة الاختيارية WHERE

كما ان هناك عبارات اخرى لتقوية الاستعلام سنوردها في حينها

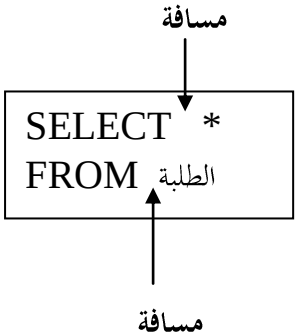
العبارة الأولى SELECT وتعني اختيار الحقول المراد تضمينها في الاستعلام من جدول معين

العبارة الثانية FROM وتعني اسم الجدول المتضمن هذه الحقول

وبدون هاتين العبارتين لا يستقيم الاستعلام

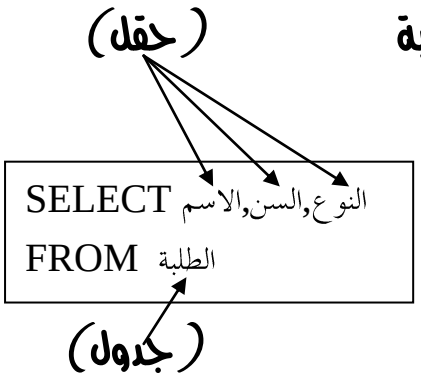
مثال في جدول الطلبة يوجد عدة حقول الاسم – السن – النوع – مادة التخطيط – مادة التصميم – مادة التطبيقات – التقدير العام- تاريخ الميلاد-

أولا لإظهار جميع الحقول في الجدول دون التقيد بشرط نكتب العبارة التالية



حيث أن علامة النجمة [*] تعني جميع حقول الجدول

ثانيا لاختيار حقول معينة من الجدول نكتب العبارة التالية



.

حيث تم اختيار ثلاث حقول فقط من الجدول دون التقيد بأي شرط

مع ملاحظة عدم وجود مسافات بين الحقول وكتابة الكومة[,] بين كل حقل وأخر بعد تحويل لوحة المفاتيح إلى وضع اللغة الانجليزية

ثالثا التقيد بشروط معينة

عندما نريد الاستعلام عن حقل أو مجموعة حقول أو جميع حقول الجدول عندما يكون احد الحقول أو مجموعة حقول تساوي قيمة معينة لا بد من كتابة العبارة الثالثة في لغة الاستعلامات وهي ما قلنا عنها أنها اختيارية

عبارة *WHERE* وتعني متى ينطبق هذا الأمر

في امثال السابق عندما نريد عرض جميع الحقول في الجدول بشرط أن يكون النوع ذكر نكتب العبارة كما يلي

```
SELECT (*)
FROM الطلبة
WHERE النوع= "ذكر"
```

مع ملاحظة انه عند كتابة شرط نصي مثل ذكر لابد من وجود علامة تنصيص [*QUOTATION*] حول الشرط أما إن كان رقم فلا داعي لوجودها

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن=20
```

هذه العبارة تعرض الحقول المختارة من الجدول عندما يكون سن الطالب يساوي *20* أي أنها تعرض اسم وسن و نوع الطلبة الذي يكون عمرهم *20* سنة

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن>10
```

هذه العبارة تعرض الحقول المختارة من الجدول عندما يكون سن الطالب اكبر من 10 أي أنها تعرض اسم وسن و نوع الطلبة الذي يكون عمرهم اكبر من 10 سنوات

استعمال أكثر من شرط في العبارة

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن= 25  AND  النوع="ذكر"
```

مسـافة

هذه العبارة تعرض الحقول المختارة من الجدول عندما يكون سن الطالب 25 سنة ويكونوا من الذكور

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن= 25 or السن=30
```

هذه العبارة تعرض الحقول المختارة من الجدول عندما يكون سن الطالب 25 سنة أو 30

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن= 25 or السن=30  AND  النوع="ذكر"
```

هذه العبارة تعرض الحقول المختارة من الجدول عندما يكون سن الطالب 25 سنة أو 30 سنة

مع ملاحظة أن الطلبة يكونوا من الذكور فمن هم سن 30 سنة دون 25 سنة

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن= 25 AND النوع="ذكر" or السن=30 AND النوع="ذكر"
```

أما هذه العبارة تعرض الاسم-السن-النوع
للطلبة الذكور من السنين 25 و 30 سنة ولاحظ عند استخدام or تفصل
بين الشروط أما and تجمع بينهم

استخدام دالة IN للحصول على أكثر من فئة للبيانات

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن IN (10,15,20)
```

هذه العبارة تعرض بيانات الطلبة والذي يكون عمرهم أما 10 أو 20 أو
30 سنة

كما يمكن استخدامها مع AND هكذا

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن IN (10,15,20) AND [مادة التطبيقات] IN (16,18,20)
```

هذه العبارة تعرض بيانات الطلبة والذي يكون عمرهم أما 10 أو 20 أو
30 سنة

ويكون أيضا مجموعهم في مادة تطبيقات الحاسب أما 16 أو 18 أو 20

ولاحظ القوسين المربعين حول مادة التطبيقات لان الاسم يتكون من مقطعين

استخدام دالة NOT مع IN

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن NOT IN (10,15,20)
```

هذه العبارة تعرض بيانات الطلبة والذي يكون عمرهم أي رقم ما عدا 10 أو 20 أو 30 سنة

استخدام دالة OR مع IN

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن IN (10,15,20) OR [التقدير العام] IN("جيد جدا","امتياز")
```

هذه العبارة تعرض بيانات الطلبة والذي يكون عمرهم أي رقم ما عدا 10 أو 20 أو 30 سنة أو أن يكون تقديره العام جيد جدا أو امتياز بغض النظر عن سنه

استخدام دالة BETWEEN_AND

تستخدم لعرض مجموعة من البيانات بين رقمين معينين في الحقل المستخدم

مثال

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن BETWEEN 20 AND 30
```

أما هذه العبارة تعرض الاسم- السن- النوع للطلبة من 20 حتى 30 سنة

أي عرض أي طالب يكون مثلا 22او27 مسافة

تداخل دالتي BETWEEN_AND مع NOT

يمكننا استخدام الدالتين عند الحاجة إلى عرض مجموعة معينة من البيانات ماعدا بيانات بين رقمين

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE NOT السن BETWEEN 20 AND 30
```

مسافة

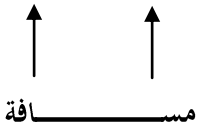
هذه العبارة تقوم بعرض الاسم – السن – النوع لجميع الطلبة فيما عدا المرحلة السنية بين 20 و30 سنة

استخدام الدالة LIKE

وهي دالة نصية تستخدم لعرض احد الحقول النصية بدلالة حرف أو مجموعة حروف في أول أو في أخر الكلمة فمثلا لعرض أسماء جميع الطلبة الذين يبدأ اسمهم بحرف م نكتب الأتي

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE الاسم LIKE "م*"
```

ولعرض الطلبة الذين يبدأ اسمهم بحرف نكتب مح بدل من م وهكذا



ولعرض الطلبة الذين ينتهي اسمهم بحرف [د] نعكس وضع النجمة ويمكن أيضا وضعحرفين أو أكثر

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE الاسم LIKE "*د"
```

هذه العبارة تعرض بيانات الطلبة الذين يحتوي أسمائهم على حرف [د] في أي مكان في الاسم

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE الاسم LIKE "*د*"
```

هذه العبارة تعرض بيانات الطلبة الذين يحتوي أسمائهم على حرف [د] في أي مكان في الاسم

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE الاسم NOT LIKE "*د*"
```

هذه العبارة تعرض بيانات الطلبة الذين تبدأ أسمائهم بحرف من [د حتى س]

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE الاسم NOT LIKE "[س-د]*"
```

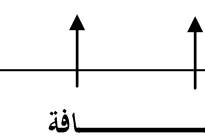
استخدام دالة NOT

وتستخدم في الحقول النصية والرقمية على حد سواء

وذلك لعدم عرض فئة معينة من البيانات فمثلا عند الحاجة إلى عرض جميع تقديرات الطلبة ما عدا أولائك الذين حصلوا على مقبول نكتب الصيغة التالية

وهذه العبارة تعرض جميع بيانات الطلبةماعدا الطلبة الذين حصلوا على تقدير عام = مقبول

```
SELECT (*)
FROM الطلبة
WHERE NOT [التقدير العام] = "مقبول"
```



مع ملاحظة وضع قوسين مربعين عند أي

اسم حقل يتكون من مقطعين مثل التقدير العام

وأيضا وضع علامة تنصيص عندا يكون الحقل نصي وليس رقمي

استخدام دالة *NOT_AND*

وتستخدم في الحقول النصية والرقمية على حد سواء

وذلك لعدم عرض فئات معينة من البيانات اثنين أو أكثر

```
SELECT  (*)
FROM  الطلبة
WHERE  NOT  [التقدير العام] = "مقبول"  AND  NOT  [التقدير العام] = "جيد"
```

مســـــــــــــــــــــــافة

هذه العبارة تعرض جميع بيانات الطلبة ماعدا الطلبة الذين حصلوا على تقدير

عام = مقبول أو جيد

يمكن اختصار العبارة السابقة كما يلي حيث ارمز <> معناه لا تساوي

```
SELECT  (*)
FROM  الطلبة
WHERE   [التقدير العام] <> "مقبول"  AND  [التقدير العام] <> "جيد"
```

استخدام دالة *IS NULL* ودالة *IS NOT NULL* بمفردهما

وتستخدم لعرض البيانات التي تخلو من فئة معينة أو التي يقتصر على فئة معينة مثل

في قاعدة بيانات الطلبة لو انه يوجد بيان برقم تليفون الطالب فهناك من يسجل تليفون وهناك من لا يسجله فلو إننا نريد معرفة الطلبة الذين لديهم تليفون أو العكس كالأتي

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE [رقم التليفون] IS NOT NULL
```

هذه العبارة تعرض اسم – سن – نوع جميع الطلبة الذين يسجلون رقم تليفون في قاعدة البيانات

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE [رقم التليفون] IS NULL
```

هذه العبارة تعرض اسم– سن – نوع جميع الطلبة الذين لم يسجلوا رقم تليفون في قاعدة البيانات

التخصيص في وقت التنفيذ(استعلام المعلمات)

وهي عبارة معناها أن المستخدم بنفسه هو الذي يخصص البيانات وليس المبرمج

بمعنى أن كل ما تقدم يضعه المبرمج في مرحلة الإعداد لقاعدة البيانات ولكن عند التنفيذ لا يمكن للمستخدم تخصيص بيانات أخرى لا يكون المبرمج أعدها له وعمليا فمثلا في المثال السابق [مثال دالة NOT BETWEEN] لو حفظنا الاستعلام كما هو عندما يقوم المستخدم بفتحه في كل مرة يعرض له بيانات الطلبة فيما عدا الطلبة الذين يكون عمرهم بين 20 و30 سنة ولو أراد المستخدم تخصيص احد الأعمار الأخرى فلن يستطيع المستخدم إلا باستخدام التعبيرات المخصصة

ويسمى هذا النوع[استعلام المعلمات)

مثال : باستخدام هذه العبارة تقوم بعرض رسالة على المستخدم وقت التنفيذ أي عند تشغيل الاستعلام وبناء على ما سيكتبه المستخدم سيتم عرضه فلو انه كتب في مربع الرسالة 10 سيتم عرض اسم-سن – نوع الطلبة الذين يكون عمرهم 10 سنوات فقط

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن = [اكتب السن]
```

ملحوظة لا بد من كتابة المربعين حول أي عبارة ولو كانت من مقطع واحد

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن BETWEEN [الحد الأدنى للسن] AND [الحد الأقصى للسن]
```

مسافة

هذه العبارة تعرض على المستخدم رسالتين وتطالبه بإدخالين لتحديد السن يقع بين الإدخال الأول والثاني فلو قام المستخدم بوضع 10 في الإدخال الأول و20 في الإدخال الثاني لعرض الاستعلام البيانات بين هاذين السنين

كما يمكن أيضا إظهار رسالة للمستخدم تطلب منه إدخال الحرف الأول أو الأخير من الاسم المراد عرضه هكذا

يعرض هذا الاستعلام بيانات الاسم – السن – النوع للطلبة الذين يبدأ اسمهم بالحرف الذي سيقوم بإدخاله المستخدم

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن = LIKE [اكتب الحرف الأول] & "*"
```

يعرض هذا الاستعلام بيانات الاسم – السن – النوع للطلبة الذين ينتهي اسمهم بالحرف الذي سيقوم بإدخاله المستخدم

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن = LIKE "*" & [اكتب الحرف الأخير]
```

يعرض هذا الاستعلام بيانات الاسم – السن – النوع للطلبة الذين يحتوي اسمهم على أي حرف يقوم بإدخاله المستخدم

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن = LIKE "*" & [اكتب احد الحروف] & "*"
```

كما يمكن التخصيص هكذا

يعرض هذا الاستعلام بيانات الاسم – السن – النوع للطلبة الذين يزيد عمرهم عن الإدخال الذي يقوم بإدخاله المستخدم

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن > [اكتب السن]
```

استخدام دالة IN مع المعلمات

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن IN( [اكتب السن الأول] , [اكتب السن الثاني], [اكتب السن الثالث])
```

هذه العبارة تطلب من المستخدم عدة إدخالات بعدد الأقواس المربعة الموجودة في قوس IN []

وبناء على هذه المدخلات من المستخدم وقت التنفيذ يتحدد الأعمار التي ستظهر بياناتها

فمثلا لو أن المستخدم قام بكتابة 22 في الإدخال الأول ثم 25 ثم 33 تظهر بيانات الطلبة لهذه الأعمار فقط دون بقية الطلبة

كما يمكن استخدامها مع NOT هكذا

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن NOT IN( [اكتب السن الأول] , [اكتب السن الثاني], [اكتب السن الثالث])
```

هذه العبارة تظهر بيانات جميع الطلبة عدا الأعمار التي سيقوم المستخدم بإدخالها

فمثلا لو أن المستخدم قام بكتابة 22 في الإدخال الأول ثم 25 ثم 33 تظهر بيانات الطلبة لكل الأعمار عدا هذه الأعمار

دالة NOT NULL مع استعلام المعلمات

عندما يقوم المستخدم بإدخال قيم لا تماثل الموجود بحقل البيانات فان الاستعلام لا يرجع أي قيمة

أي في المثال السابق إذا قام المستخدم بكتابة سن الطالب 50 سنة وكان لا يوجد طالب بهذا السن فان الاستعلام لا يعرض أي بيانات أو يعرض بيانات صف واحد خالي من البيانات

ولتفادي هذا الوضع وعرض جميع البيانات نستخدم دالة IS NOT NULL

يعرض هذا الاستعلام بيانات الطلبة الذين يكون عمرهم العمر الذي يقوم المستخدم بوضعه وإذا وضع سن غير موجود بالحقل فان الاستعلام يعرض جميع البيانات

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن = [اكتب السن] OR IS NOT NULL
```

استخدام شرطين مع المعلمات باستخدام AND

هذه العبارة تعرض اسم – سن – نوع جميع الطلبة الذين يكون عمرهم بين الإدخال الأول والثاني

فإذا تم 10 في الإدخال الأول و 20 في الثاني يعرض الاستعلام بيانات الطلبة الذين يزيد عمرهم عن 10 سنوات ولا يزيد عن 20 سنة

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن > [اكتب السن] AND السن < [اكتب السن]
```

استخدام شرطين مع المعلمات باستخدام OR

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن = [اكتب السن] OR السن = [اكتب السن]
```

هذه العبارة تعرض اسم – سن – نوع جميع الطلبة الذين يكون عمرهم يساوي الإدخال الأول أو الثاني

فقط فإذا تم 10 في الإدخال الأول و 20 في الثاني يعرض الاستعلام بيانات الطلبة للأعمار 10 سنوات او20 سنة فقط دون غيرهم

استحداث حقل غير موجود بالجدول المصدر

يعتمد الاستعلام في جميع معاملاته على الحقول المتاحة من الجدول المصدر [مصدر البيانات]

ولكننا في بعض الأحيان يمكننا استحداث حقل غير متضمن في الجدول يسمى حقل محسوب

```
SELECT النوع,السن,الاسم,([مادة التصميم]+[مادة التطبيقات]+[مادة التخطيط]) AS [المجموع الكلي]
FROM الطلبة
```

على سبيل المثال في قاعدة البيانات الطلبة إذا كان لا يوجد حقل التقدير العام يمكننا إنشاءه في الاستعلام كما يلي :

تعرض هذه العبارة الاسم – السن – النوع – المجموع الكلي

حيث ان حقل المجموع الكلي هو الحقل الذى استحدثناه وهو غير موجود اصلا في الجدول وهو عبارة عن مجموع الثلاث مواد مجتمعة

مع ملاحظة انه بعد عبارة AS نضع اسم الحقل المراد استحداثه [أي اسم]

ومع ملاحظة ان الاقواس المربعة حول اسم الحقل نتيجة انه من مقطعين ولو كان من مقطع واحد ما احتجنا اليهم

اما الاقواس المربعة حول الحقول المجموعة فهي لازمة حتى ولو مقطع واحد بالاضافة الى القوس العادي

كما انه يمكن استخدام أي عمليات حسابية أخرى مثل

```
SELECT الاسم,السن,النوع,[مادة التخطيط]*100/20) & ("%")  AS [النسبة المئوية]
FROM الطلبة
```

تعرض هذه العبارة الاسم – السن – النوع – النسبة المئوية

حيث ان حقل النسبة المئوية هو الحقل المستحدث وهو يحسب مجموع مادة التخطيط *100 / المجموع الكلي للمادة

ونلاحظ ايضا اننا استخدمنا دالة الجمع النصية [&] لاضافة عبارة اخرى علي المجموع وهي علامة النسبة المئوية [%] ويجب وضعها بين علامتي تنصيص

استحداث حقل غير موجود بالجدول المصدر وكتابة بياناته

في قاعدة بيانات الطلبة لو أننا نريد استحداث حقل لتقديرات مادة التطبيقات على اعتبار إن الحقل المتوفر هو درجة التطبيقات
والحل هو مقارنة الدرجة بمجموعة من الخيارات [الدرجات] وإذا وافقت احد الخيارات اكتب كذا [التقدير] ولعمل هذه العبارة اتبع الآتي

```
SELECT الاسم,السن,النوع, IIF( [مادة التخطيط] >16, "مقبول",IIF( [مادة التخطيط] >10,"جيد",IIF( [مادة التخطيط] >14,"جيد جدا" , IIF( [مادة التخطيط] >18,"امتياز", IIF([مادة التخطيط] <6,"ضعيف", IIF( [مادة التخطيط] >6,"ضعيف جدا"))))) AS [تقدير مادة التخطيط]
FROM الطلبة
```

هذه العبارة تقوم باختبار مجموع مادة التخطيط الهيكلي فإذا كانت اكبر من 18 يكون البيان في الحقل امتياز وهكذا
يمكننا أيضا كتابة بيانات للتقدير العام أن لم يجد ببعض التغيير عن الصيغة السالفة

```
SELECT الاسم,السن,النوع, IIF(([مادة التخطيط]+[مادة التطبيقات]+[مادة التصميم])>54,"امتياز", IIF(([مادة التخطيط]+[مادة التطبيقات]+[مادة التصميم])>48,"جيد جدا", IIF(([مادة التخطيط]+[مادة التطبيقات]+[مادة التصميم])> 42,"جيد", IIF(([مادة التخطيط]+[مادة التطبيقات]+[مادة التصميم])>30,"مقبول",IIF(([مادة التخطيط]+[مادة التطبيقات]+[مادة التصميم])>18,"ضعيف",IIF([مادة التخطيط]+[مادة التطبيقات]+[مادة التصميم])<18,"ضعيف جدا"))))) AS [تقدير مادة التخطيط]
FROM الطلبة
```

الاستعلام بشرط عملية حسابية

إذا فرضنا إننا نريد عرض بيانات الطلبة الذي يزيد مجموعهم في جميع المواد عن 54 أي الحاصلين على تقدير عام امتياز على اعتبار أن الامتياز يبدأ من 90٪ أي 54 درجة

وتصلح أيضا لعرض الطلبة الحاصلين على تقديرات عامة جيد أو جيد جدا في حالة عدم تضمين حقل في الجدول المصدر يحتوي على التقديرات العامة

نقوم بكتابة العبارة التالية

```
SELECT (*)
FROM الطلبة
WHERE ([مادة التخطيط]+[مادة التطبيقات]+[مادة التصميم])>54
```

كما يمكن الاستعلام بأكثر من شرط كالتالي

```
SELECT (*)
FROM الطلبة
WHERE ([مادة التخطيط]+[مادة التطبيقات]+[مادة التصميم])>48 AND ([ مادة التخطيط]+[مادة التطبيقات]+[مادة التصميم])<54
```

وذلك للاستعلام عن جميع بيانات الطلبة الحاصلين على مجموع بين 48و54 درجة أي تقديرهم العام جيد جدا

الفرز في لغة SQL

الفرز هو ترتيب البيانات تصاعديا أو تنازليا وهذه العبارة تصلح لكل العبارات السالفة

هذه العبارة تعرض جميع بيانات الطلبة مرتبة تنازليا حسب سن الطالب

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن > [اكتب السن]
ORDER BY السن DESC
```

هذه العبارة تعرض جميع بيانات الطلبة مرتبة تصاعديا حسب سن الطالب

```
SELECT النوع,السن,الاسم
FROM الطلبة
WHERE السن > [اكتب السن]
ORDER BY السن ASC
```

التخصيص عن طريق نموذج

وهذا الموضوع من الاهمية بمكان اذ ان المبرمجين المتخصصين لا يستخدموا الاستعلامات الا باستخدام هذه الطريقة

وهذه الطريقة تعني انشاء نموذج غير مرتبط به بعض الادوات المحتوية على بيانات للاستعلام بها بدل من كتابة شروط الاستعلام بأنفسنا

مثلا لاحظ هذا المثال لجدول المشتريات *Buying*

اسم الصنف	تاريخ الانتاج	سعر الشراء للوحدة	اسم المورد	رقم التليفون
ماوس usb fox	05-Nov-00	11.5	محمد صبري	010236565
ماوس fox usb	02-Mar-03	10.55	القدس	01156789
ماوس fox usb	28-Dec-00	10.5	القدس	01156789
ماوس fox usb	08-Dec-73	10	محمد صبري	010236565
cpu	06-Apr-05	180.5	جرين	0102687
cpu	03-Apr-05	175	الرواد	0126939877
cpu	06-Apr-02	180	محمد صبري	010236565
cpu 3.6	08-Apr-05	205	القدس	01156789
cpu 3.6	03-Apr-05	195	محمد صبري	010236565
cpu 3.6	05-Apr-05	170	القدس	01156789
cpu 3.6	03-Apr-05	165	جرين	0102687
cpu 3.6	21-Apr-05	201.5	الرواد	0126939877
DVD	06-Apr-05	92.5	جرين	0102687
DVD	03-Apr-05	101.5	الرواد	0126939877
DVD LG	03-Apr-05	100	الرواد	0126939877
DVD R/W LG	06-Apr-05	91.5	جرين	0102687
DVD R/W LG	03-Apr-05	100	الرواد	0126939877

لو اردنا مثلا الاستعلام من هذا الجدول عن بيانات احد الموردين مثلا القدس لكان الاستعلام كما يلي

```
Select *
From buying
Where product = "القدس"
```

وفي كل مرة نريد الاستعلام عن مورد اخر ننشا استعلام اخر ونغير فيه اسم المورد

وبدل من ذلك نقوم بانشاء نموذج جديد نسميه مثلا main_query وننشا فيه اداة مربع تحرير وسرد combobox نسميها مثلا name_product ونقوم بتعبئتها كالاتي

تعيين الخاصية Row source Type نوع مصدر الصف

الى Table/Query أي مصدر القائمة المنسدلة من جدول او استعلام

تعيين الخاصية Row source مصدر الصف الى جملة SQL للإحضار

حقل بيانات من الجدول او الاستعلام المراد احضار بيانات منه بدل من كتابتها بأنفسنا

مع ملاحظة انه لو تم احضار بيانات يحتمل ان تكون فيها بيانات مكررة نكتب الجملة هكذا

Select distinct(الحقل المراد) from الجدول

مثلا

Select distinct(product_name) from buying

فتقوم الاداة بتحميل كل بيانات حقل اسم المورد [جميع اسماء الموردين] الموجود في جدول المبيعات buying

ثم نكتب الشرط في الاستعلام كما يلي

```
Select *
From buying
Where product =
[forms]![main_query]![product_name]
```

فبدل من كتابة اسم المورد نشير الى هذا الاسم في النموذج الذي أنشأناه

ثم ننشأ في النموذج زر نكتب فيه الجملة التالية لفتح الاستعلام من النموذج

Docmd.openquery "buying_query"

وهكذا في كل الاستعلامات نشير الى كل المعايير بدل من كتابتها وهكذا

دالات SQL

اولا دالات التاريخ والوقت

دالة *Date()* تقوم بتضمين تاريخ اليوم بتنسيق النظام الحالي

مثال لتضمين حقل جديد في احد الاستعلامات يتضمن تاريخ اليوم بعد اخر حقل في عبارة select ندرج كومة [,] ثم [تاريخ اليوم] as ()date

دالة *DateAdd* يمكنك استخدامها لإضافة فاصل زمني معيّن أو طرحه من تاريخ ما. على سبيل المثال، يمكنك استخدام الدالة DateAdd لزيادة تاريخ بمقدار 30 يوماً من اليوم او من أي تاريخ اخر أو وقت مدة 45 دقيقة من الآن

مثال:

```
Select date_in,dateadd("m",2,[date_in]) as [تاريخ الشحن]
From buying
```

يقوم هذا الاستعلام بتضمين حقل تاريخ الدخول للسلعة ثم ينشا حقل جديد غير موجود في الجدول المصدر لتاريخ الشحن وهو بعد تاريخ الدخول بشهرين

فلو أن تاريخ الدخول 2010/06/05 يكون ناتج الدالة 2010/08/05

يمكن أيضا الطرح من التاريخ طرح شهرين من تاريخ الدخول

مثال اخر :

```
Select date_in,dateadd("yyyy",-2,[date_in]) as [تاريخ الانتاج]
From buying
```

فلو أن تاريخ الدخول 2010/06/05 يكون ناتج الدالة 2008/06/05

يقوم هذا الاستعلام بتضمين حقل تاريخ الدخول للسلعة ثم ينشا حقل جديد غير موجود في الجدول المصدر لتاريخ الانتاج وهو قبل تاريخ الدخول بسنة كاملة

معاملات الدالة

y	يوم من السنة
yy	سنة بتنسيق رقمين
yyy	يوم من السنة للسنة المذكورة
yyyy	السنة بتنسيق 4 ارقام
q	ربع سنوي
m	الشهر
d	اليوم
w	يوم من الأسبوع
ww	الأسبوع
h	الساعة
n	الدقيقة
s	الثانية

دالة *Datepart*

تقوم بقطع مقطع معين من التاريخ المضمن

مثال datepart("yyyy",date_in) استقطاع السنة من التاريخ المدرج

فلو أن تاريخ date_in = 2010/03/15 تكون نانتج الدالة 2010

وجميع المعاملات السابقة تعمل معها عدا yy، yyy

مثال آخر datepart("ww",date_in) استقطاع رقم الاسبوع من التاريخ المدرج على اعتبار أن السنة 53 اسبوع

فلو أن تاريخ date_in = 2010/11/10 تكون نانتج الدالة 46 أي أن هذا التاريخ يقع في الاسبوع الـ46 من السنة

دالة *datediff*

تقوم بايجاد الفرق بين تاريخين

الصيغة العامة للدالة

DateDiff(interval,date1, date2, firstdayofweek, firstweekofyear)□

حيث interval نوع المعامل المراد معرفة الفارق عن طريقه وهي نفس المعاملات السابقة

date1 التاريخ الأول date2 التاريخ الثاني

firstdayofweek وهو ثابت يحدد اليوم الأول من الأسبوع وهو اختياري اذا لم يذكر يتم حساب الافتراضي في النظام وهو الأحد أول الأسبوع أما إذا أردنا غير ذلك نكتب المعامل هكذا vbSaturday ليبدأ الأسبوع بالسبت

firstweekofyear ثابت يحدد الأسبوع الأول من السنة وهو اختياري اذا لم يذكر يتم حساب الافتراضي في النظام أي الأسبوع الذي ياتي فيه يناير حتى لو يوم واحد في يناير

او vbFirstFourDays البدء بأول أسبوع يتضمن 4 أيام على الأقل في السنة الجديدة او vbFirstFullWeek البدء بأول أسبوع كامل في السنة

مثال : datediff("yyyy",(product_date),date())

هذا المثال يقوم بإيجاد الفرق بين سنة الإنتاج والسنة للتاريخ الحالي فمثلا لو أن تاريخ الإنتاج 2008/11/15 يكون ناتج الدالة 2

مثال آخر: datediff("ww",(product_date),date())

هذا المثال يقوم بإيجاد الفرق بين الأسبوع لتاريخ الإنتاج والأسبوع للتاريخ الحالي فمثلا لو أن تاريخ الإنتاج 2008/04/06 يكون الناتج 136 أسبوع فارق بين أسبوع الإنتاج والأسبوع الحالي

الدالة *Year*

لإيجاد السنة في التاريخ المرفق year(product_date) يقوم باستخراج السنة من تاريخ الإنتاج فمثلا لو أن تاريخ الإنتاج 2008/04/06 يكون الناتج 2008

الدالة *Month*

لإيجاد الشهر في التاريخ المرفق month(product_date) يقوم باستخراج الشهر من تاريخ الإنتاج فمثلا لو أن تاريخ الإنتاج 2008/04/06 يكون الناتج 04

الدالة *Monthname*

لإيجاد اسم الشهر في التاريخ المرفق month(product_date) يقوم باستخراج الشهر من تاريخ الإنتاج فمثلا لو أن تاريخ الإنتاج 2008/04/06 يكون الناتج ابريل

الدالة *Day*

لإيجاد اليوم في التاريخ المرفق Day(product_date) يقوم باستخراج اليوم من تاريخ الإنتاج فمثلا لو أن تاريخ الإنتاج 2008/04/06 يكون الناتج 06

الدالة *weekday*

ارجاع رقم اليوم في الاسبوع

مثال

```
SELECT weekday(date_in)
FROM buying
```

يقوم بارجاع رقم اليوم في التاريخ المذكور

الدالة *weekdayname*

ارجاع اسم اليوم لرقم اليوم في الاسبوع

مثال

```
SELECT WeekdayName(weekday(date_in))
FROM buying
```

يقوم بارجاع اسم اليوم في التاريخ المذكور مثلا 1=السبت ، 2= الاحد

الدالة *Format*

الصيغة العامة للدالة format(expression,"format")

حيث expression تعبير أو حقل يعبر عن تاريخ أو نص أو رقم

"format" التنسيق المطلوب موضوع بين علامتي تنصيص

مثال1 :

```
format(product_date,"dd/mmm/yyyy")
```

في المثال تقوم الدالة بإعادة صياغة تاريخ الإنتاج بصورة مختلفة فمثلا لو أن التاريخ يساوي 2008/04/06 يصبح هكذا 06/ابريل/2008
حيث mmm تعبر عن اسم الشهر ، ddd تعبر عن اسم اليوم

مثال2

لإنشاء حقل لمعرفة اسم يوم الإنتاج

```
Select format(product_date,"ddd") as [يوم الإنتاج]
From buying
```

فلو أن التاريخ = 2008/04/06 يكون الناتج الأحد وهكذا يمكن استخدامها لتقوم بنفس وظيفة الدالات السابقة year – month – day – datepart مع زيادة بعض الإمكانات في المعامل حيث أن الدالات السابقة لا تتعامل مع المعاملات ddd, mmm

مثال3

```
format(NOW(),"HH:MM:SS AMPM") as [وقت الإنتاج]
```

ينتج عن هذا التعبير صياغة الوقت الحالي مثلا هكذا ص 09:15:21
مع ملاحظة أن هذا الوقت لن يتم تحديثه إلا عند إعادة تشغيل الاستعلام مرة أخرى

ملحوظة : يتم استخدام الجدول الاتي لتغير تنسيقات دالة format

y	يوم من السنة
yy	سنة بتنسيق رقمين
yyy	يوم من السنة للسنة المذكورة
yyyy	السنة بتنسيق 4 ارقام
q	ربع سنوي
m	رقم الشهر
mm	رقم الشهر بتنسيق رقمين
mmm	اسم الشهر

d	رقم اليوم
dd	رقم اليوم بتنسيق رقمين
ddd	اسم اليوم
w	يوم من الأسبوع
ww	الأسبوع
h	الساعة
n	الدقيقة
s	الثانية

مثال 4

format(price_in,"00.00") هذا المثال يقوم بإعادة صياغة سعر الشراء بصورة جديدة فمثلا لو أن السعر = 8.25 يصبح 08.25

لو أن السعر = 14.258 يصبح 14.26

ويمكن صياغته بالشكل التالي format(price_in,"##.###") والفارق بينهم أن علامة 0 تحجز مكان للرقم سواء وجد أم لا أما # فهي أن وجد الرقم تقوم بحجز مكان له وإلا فلا ولو أن السعر = 8.2526 يصبح 8.253 بدون أن يضع 0 قبل الرقم 8

الدالة *Dateserial*

الصيغة العامة Dateserial(year-1,month-1,day-1)

ملحوظة : في الدالة Dateadd كنا نقوم بجمع وطرح الايام أو الشهور أو السنين واحدة في كل مرة وكنا نحصل على التاريخ مفرد أي أما سنة أو شهر أو يوم أما هذه الدالة فنقوم بجمع وطرح التواريخ مجمعة يوم وشهر وسنة ونحصل على تاريخ مجمع

مثال

```
Dateserial(year(product_date)-
3,month(product_date)-2,day(product_date)+3)
```

فاذا كان تاريخ الإنتاج 2008/04/06 يكون الناتج 2005/02/09

دالات الاختبار

وهي دالات هامة جدا في اختبار القيم المحتوية عليها الحقول

دالة *Iserror*

تختبر الاخطاء الناتجة عن الدالات الاخرى والتعبيرات

مثال : التعبير التالي يقوم باحضار قيمة من نموذج فرعي الى نموذج اساسي في لاداة غير منضم لاداة Text او Combo

[اسم الحقل]![form].[اسم النموذج الفرعي]

فاذا لم يحتوي السجل على بيانات في هذا الحقل يرجع التعبير Error ولكن يمكن كتابته بهذا الشكل

اسم] ,"",[اسم الحقل]![form].[اسم النموذج الفرعي])Iif(iserror([

[اسم الحقل]![form].[النموذج الفرعي

التجميع

يقصد بالتجميع تجميع السجلات المتشابهة باستخدام احد دالات التجميع

مثال :

```
SELECT Count(class_code) AS [عدد الاصناف]
FROM buying;
```

هذا المثال يقوم بتجميع سجلات كود الصنف باستخدام دالة

التجميع *Count*

عدد الاصناف
108

فيكون الناتج

اما ان اردنا معرفة عدد الاصناف للعملاء المختلفون أي سنقوم باضافة احد الحقول للاستعلام فيجب اخبار البرنامج ان التجميع سيكون عن طريق هذا الحقل فنكتبه هكذا

```
SELECT Count(class_code) AS [عدد الاصناف],product_name
FROM buying;
Group by product_name
```

استعلام1	
اسم المورد	عدد الاصناف
الأصدقاء باب اللوق	11
الرواد	29
القدس	21
جرين	11
عائشة	1
على	8
عمر بن الخطاب	15
محمد صبري	12

فيكون الناتج

ولاحظ ان البرنامج قام بتجميع السجلات المتشابهة في حقل الدالة ولكن بتجميع اسم العميل أي يقوم بجمع سجلات كل عميل على حدى ثم تم اضافة اسم العميل لمعرفة كل عدد يخص أي من العملاء

ملحوظة : لو لم يتم اضافة اسم العمسل في جملة Select لن يؤثر في عملية التجميع ولكن لن نعرف أي من الاعداد المذكورة تخص أي من العملاء وهكذا يكون التجميع اما ان اردنا اضافة مستوى اخر للتجميع مثلا لو اردنا

معرفة عدد الاصناف لكل عميل لكل يوم شراء فيتم اضافة تاريخ الشراء للتجميع فتكون هكذا

```
SELECT Count(class_code) AS [عدد الاصناف],product_name,product_date
FROM buying
group by product_name,product_date
```

عدد الاصناف	اسم المورد	تاريخ الانتاج
1	الأصدقاء باب اللوق	ابريل-05-01
2	الأصدقاء باب اللوق	ابريل-05-03
1	الأصدقاء باب اللوق	ابريل-05-06
7	الأصدقاء باب اللوق	ابريل-05-09
8	الرواد	مارس-01-28
3	الرواد	ابريل-01-21
6	الرواد	ابريل-05-03
3	الرواد	ابريل-05-06
4	الرواد	ابريل-05-09
5	الرواد	ابريل-05-21
3	القدس	مارس-01-28
2	القدس	ابريل-01-21

فتكون النتيجة كما ترى وهذه عينة من الجدول الناتج وليس الجدول كلة ونلاحظ انه كلما زادت مستويات التجميع كلما زادت السجلات الناتجة والعكس بالعكس

ثانيا دالات التجميع

دالة *Count* دالة العدد وتستخدم كما سبق لتجميع عدد من السجلات المتشابهة

دالة *Sum* دالة المجموع وتستخدم لجمع عدد من السجلات المتشابهة

مثال :

```
SELECT sum((price_out-price_in)*quantity_in) as [اجمالي مكسب الصنف],class_name
from buying
group by class_name
```

اسم الصنف	اجمالي مكسب الصنف
ماوس usb fox	425.95
cpu	59.5
cpu 3.6	60
DVD	66
DVD LG	65

371.5	DVD R/W LG
154	DVD R/W Sansong
56	flash 16G
302.5	flash 4G
184	flash 8G
688.93	MB G.B G31
240	ram 1G ddr2
90.5	Ram 512 ddr2

تم استخدام دالة Sum لجمع السجلات المتشابهة طبقا لاسم الصنف اي يقوم بجمع كل السجلات طبقا للمعادلة التي تطرح سعر الشراء من سعر البيع ثم يتم ضربه في الكمية لايجاد المكسب ويتم تجميعه بدالة sum بتجميع اسم الصنف فيقوم بحساب المكسب لكل صنف

يمكن استخدام المعايير لايجاد مكسب احد او مجموعة من الاصناف مثلا تابع المثال التالي المراد تجميع مكسب الاصناف التالية dvd , flash 16g , flash 8g

مع العلم ان المكسب هو ناتج طرح سعر الشراء من سعر البيع ثم يتم ضربه في الكمية

الحل :

```
SELECT sum((price_out-price_in)*quantity_in) AS [اجمالي مكسب الصنف]،class_name
FROM buying
WHERE class_name in("dvd","flash 16g","flash 8g")
GROUP BY class_name؛
```

فيكون الناتج

اسم الصنف	اجمالي مكسب الصنف
DVD	66
flash 16G	56
flash 8G	184

المعامل *Having*

في المثال السابق لو اردنا تطبيق احد المعايير على ناتج التجميع لا يمكن استخدام العبارة Where لان where تستخدم قبل التجميع فقط ولا يمكن استخدامها مع ناتج التجميع ولكن تستخدم Having لمعايير التجميع

اسم الصنف	Expr1000
usb fox ماوس	42.55
DVD	92.5
DVD LG	95
DVD R/W LG	458.5
DVD R/W Sansong	194.5
flash 16G	275.5
flash 4G	426
flash 8G	548
Ram 512 ddr2	209.5

مثال : لو اردنا تجميع اسعار الاصناف التي لا تزيد عن 100 جنيه يمكن تطبيق المعيار قبل التجميع باستخدام Where ثم يتم تطبيق الاستعلام هكذا

```
SELECT sum(price_in), class_name
FROM buying
WHERE price_in < 100
GROUP BY class_name;
```

نجد ان البرنامج قام بتجنيب الاصناف التي تزيد عن 100 جنيه قبل عملية التجميع ثم قام بعملية التجميع بعد ذلك

اما اذا اردنا مثلا تجميع الاصناف التي لا يزيد مجموعها عن 400 جنيه فالطبيعي اننا لا يمكن معرفة مجموعها الا بعد تطبيق دالة التجميع وبالتالي لا يمكن استخدام where ولكن نتم كما يلي

اسم الصنف	Expr1000
usb fox ماوس	42.55
DVD	194
DVD LG	195
flash 16G	275.5
Ram 512 ddr2	209.5

```
SELECT sum(price_in), class_name
FROM buying
GROUP BY class_name
having sum(price_in) < 400;
```

الدالة *AVG* دالة المتوسط الحسابي

لايجاد متوسط مكسب الاصناف لكل عميل نقوم بالاتي

اسم المورد	اسم الصنف	متوسط مكسب الصنف
القدس	usb fox ماوس	69.225
محمد صبري	usb fox ماوس	143.75
الرواد	cpu	
جرين	cpu	29.5
محمد صبري	cpu	30
الرواد	cpu 3.6	0
القدس	cpu 3.6	0
جرين	cpu 3.6	45
محمد صبري	cpu 3.6	15
الرواد	DVD	28.5
جرين	DVD	37.5
الرواد	DVD LG	30
عمر بن الخطاب	DVD LG	35

الرواد	DVD R/W LG	30
جرين	DVD R/W LG	40.25
عمر بن الخطاب	DVD R/W LG	37
الرواد	DVD R/W Sansong	14.625
جرين	DVD R/W Sansong	34.5
عمر بن الخطاب	DVD R/W Sansong	30.5
حسن	flash 16G	
هشام	flash 16G	28
الأصدقاء باب اللوق	flash 4G	6.75
القدس	flash 4G	54
على	flash 4G	24.3333333333333
الأصدقاء باب اللوق	flash 8G	28
القدس	flash 8G	33.3333333333333
هشام	flash 8G	28
القدس	MB G.B G31	211.215
جرين	MB G.B G31	133.25
جرين	ram 1G ddr2	60
محمد صبري	ram 1G ddr2	60
الرواد	Ram 512 ddr2	18.1

```
SELECT
avg((price_out-
price_in)*quantity_in)
AS [متوسط مكسب الصنف]،
class_name,product_na
me
FROM buying
GROUP BY
class_name,product_na
me
```

ومن هنا نستخلص انه كلما تم زيادة احد مستويات التجميع مثل اسم العميل على الاستعلام ذادت السجلات الناتجة فقام بتقسيم سجلات الصنف الواحد الى عدة سجلات طبقا للعميل ولكن اذا اجتمع عميل واحد لعدة اصناف يتم تجميعهم وهكذا

دالة *First* السجل الاول

مثال لايجاد اول تاريخ لكل صنف تم التوريد فيه

```
SELECT first(date_in), class_name
FROM buying
GROUP BY class_name
```

اسم الصنف	Expr1000
usb fox ماوس	10/11/2010
cpu	10/11/2010
cpu 3.6	21/04/2010
DVD	06/04/2010
DVD LG	06/04/2010
DVD R/W LG	06/04/2010
DVD R/W Sansong	06/04/2010
flash 16G	28/03/2010
flash 4G	28/03/2010
flash 8G	28/03/2010
MB G.B G31	10/11/2010
ram 1G ddr2	01/04/2010
Ram 512 ddr2	21/04/2010

دالة *Last*

السجل الاخير كما في المثال السابق

دالة *Max*

الاعلى لإيجاد اعلى سجل في مجموعة سجلات

لايجاد اعلى سعر تم التوريد به لاحد الاصناف

```
SELECT max(price_in) as ،[اعلى سعر] class_name
FROM buying
where class_name = "cpu"
GROUP BY class_name
```

اسم الصنف	اعلى سعر
cpu	180.5

دالة *Min*

الاقل لإيجاد اقل سجل في مجموعة سجلات كما سبق

دالة *ABS*

لإيجاد الرقم المطلق بدون اشارة مثال :

```
SELECT abs(price_in) ،class_name
FROM buying
GROUP BY class_name
```

دالة *RIGHT*

اقتطاع جزء من النص من جهة اليمين

مثال

```
SELECT right(class_name,4)
FROM buying
where class_name = "flash 8 GB"
```

يكون الناتج

Expr1000
h 4G
h 4G
h 4G
h 4G
h 4G
h 4G
h 4G
h 4G
h 4G

ولاحظ ان الناتج عبارة عن الاربع حروف الاولى من ناحية اليمين لكل السجلات التي توافق الشرط

دالة LEFT

اقتطاع جزء من النص من جهة اليسار

مثال

```
SELECT left(class_name,4)
FROM buying
where class_name = "flash 8 GB"
```

يكون الناتج

Expr1000
flas
flas
flas
flas
flas
flas
flas
flas
flas

ولاحظ ان الناتج عبارة عن الاربع حروف الاولى من ناحية اليسار لكل السجلات التي توافق الشرط

دالة MID

اقتطاع جزء من النص من جهة منتصف النص

مثال

```
SELECT mid(class_name,7,1)
FROM buying
where class_name = "flash 8 GB"
```

Expr1000
8
8
8
8
8
8
8
8
8

وتعني الدالة اقتصاص حرف واحد من النص المذكور بداية من الحرف رقم 7 مع العلم انه يتم احتساب المسافة على لنها حرف

دالة LEN وهي تقوم بعد عدد الحروف أو الأرقام

مثال

```
SELECT Len(class_name) as [عدد الاحرف]
FROM buying
where class_name = "flash 8 GB"
```

عدد الاحرف
10

دالة UCASE

تحويل الحروف الانجليزية للحروف الكبيرة capital

Expr1000
FLASH 4G
FLASH 4G
FLASH 4G
FLASH 4G
FLASH 4G
FLASH 4G
FLASH 4G
FLASH 4G
FLASH 4G

```
SELECT ucase(class_name[
FROM buying
where class_name="flash 4G"
```

دالة LCASE

تحويل الحروف الانجليزية للحروف الصغيرة small

Expr1000
flash 4g
flash 4g
flash 4g
flash 4g
flash 4g
flash 4g
flash 4g
flash 4g
flash 4g

```
SELECT ucase(class_name[
FROM buying
where class_name="flash 4G"
```

دالة المرجعية الصفرية NZ

تقوم باستبدال الحقول الخالية لحقل صفري

وتستخدم في عمليات جمع وطرح وضرب الحقول حتى تعطي ناتج حيث انه لو تم ادراج حقل يحتوي على قيم خالية في عملية حسابية لا ينتج قيمة فبدل من كتابة اسم الحقل نكتبه بالدالة هكذا

مثال:

```
Select (NZ(price_in)-NZ(price_out)) *NZ(quantity_in))
```

دالة TOP للقيم العليا أو الدنيا

```
SELECT top 3 (class_name)
FROM buying
```

ينتج اول ثلاث قيم في اعلى الجدول

اسم الصنف
MB G.B G31
MB G.B G31
MB G.B G31

يمكن استخدامها في ايجاد اعلى 5 اسعار او اقل 3 اسعار لسلعة معينة ولكن نظر لأنها تقوم بإحضار القيم في اعلى الجدول فقط دون النظر الى قيمتها نقوم

- بترتيب الجدول تصاعديا لإيجاد اقل الاسعار وتنازليا لإيجاد اعلى الاسعار
- مثال المطلوب معرفة اقل 6 اسعار تم الشراء بها

```
SELECT  top 6 (price_in)
FROM buying
      order by price_in
```

فيكون الناتج

سعر الشراء للوحدة
10
10.5
10.55
11.5
32.5
40
40
40

ولكن لاحظ ان عدد السجلات اكثر من 6 وذلك لان التعبير السابق عندما يجد قيم متماثلة لاخر حقل يتم استدعائه يقوم بإحضار

- جميع القيم المكررة

كما يمكن ان تستخدم بالنسبة المؤية بدل من القيم

مثال المطلوب معرفة اعلى 10٪ من اسعار شراء الاصناف المختلفة

```
select top 10 percent price_in
from buying
order by price_in desc
```

فيكون الناتج

سعر الشراء للوحدة
275.5
275
262.5
262.23
205
201.5
195

ولاحظ اننا قمنا بالترتيب تنازليا والترتيب يكون تصاعديا افتراضيا بدون كتابة ESC اما للترتيب التنازلي لا بد من كتابة عبارة DESC عبارة

دالة RND توليد رقم عشوائي

```
Select rnd as [رقم عشوائي]
```

رقم عشوائي
0.533424019813538

يقوم بتوليد رقم عشوائي كسر الواحد الصحيح وفي كل مرة يتم تشغيل الاستعلام يتولد رقم جديد

الدالة INT

للتقريب لاقرب رقم صحيح

بمعنى اخر يقوم بحذف الجزء العشري من الرقم سواء كان رقم او وسيطة حقل

مثال

```
SELECT int((price_out-price_in)*quantity_in) AS [الربح]
FROM buying
```

ملحوظة : في حالة الارقام او الوسائط السالبة يعطي الرقم السالب التالي بعد العلامة

مثال

Int(-99.3) =-100

الدالة *fix*

للتقريب لاقرب رقم صحيح

وهي تقوم بنفس وظيفة الدالة Int في الغاء الجزء العشرب من الرقم ولكن الاختلاف في الارقام السالبة فهي تقرب لاقرب رقم اقل من الرقم المذكور

مثال

fix(-99.3) =-99

الدالة *ROUND*

- تقوم بالتقريب لاقرب رقم عشري مطلوب

مثال :

Round(55.12345,3) = 55.123

مثال :

- SELECT Round((price_out-price_in)*quantity_in,3) AS [الربح]
FROM buying

الربط عن طريق الاستعلامات

ويقصد بالربط هنا إحضار بيانات من عدة جداول او عدة استعلامات وإظهار النانج في استعلام جديد

مثلا بيانات المنتجين من العنوان ورقم الهاتف والهاتف الجوال والرقم الضريبي وغيرة من البيانات التي لا تتكرر توجد في جدول المنتجين

ثم إن بيانات التعاملات اليومية مع هؤلاء المنتجين توجد في جدول آخر [جدول حركة] مثل اسم السلعة والسعر والكمية المشتراه من هذا المنتج

ثم لو أردنا إحضار رقم الهاتف للمنتج من جدول المنتجين (product) في جدول الحركة (buying) اليومية نقوم بربط الجدولين كما يلي

```
SELECT buying.class_name, buying.product_date,
buying.price_in, buying.product_name,
product.product_tel
FROM product LEFT JOIN buying ON
(buying.product_name=product.product_name);
```

وشرح هذا التعبير كما يلي

جملة select

يذكر فيها الحقول المراد تضمينها في الاستعلام مسبوقا باسم الجدول أو الاستعلام المدرج منه الحقل المذكور بينهما نقطة مثلا

[buying.class_name] حيث buying اسم الجدول و class_name اسم الحقل وهكذا في كل الحقول

ثانيا جملة From

يذكر فيه اسم الجدول الأول ثم معامل الربط ثم اسم الجدول الثاني ثم حقول الربط المتساوية أي عندما الحقل المسمى من الجدول المسمى = الحقل المسمى من الجدول المسمى الآخر

أنواع الربط

One to one واحد إلى واحد ومعامل الربط فيه INNER JOIN ويقوم الاستعلام الجديد بإحضار السجلات المتساوية فقط في كلا الجداول أو الاستعلامات

فمثلا في المثال السابق لو استخدمنا معامل الربط المذكور يقوم بإحضار فقط المنجين المذكورين في كلا الجدولين فقط

أي لو ان هناك منتج موجود في جدول المبيعات ليس له بيانات في جدول المنتجين لن يتم ادراج سجلاته من جدول المبيعات

وايضا لو ان هناك منتج موجود في جدول المنتجين ليس له بيانات في جدول المبيعات لن يتم ادراج سجلاته من جدول المنتجين أي يتم ادراج السجلات الموجودة في كلا الجدولين فقط

اسم الصنف	تاريخ الانتاج	سعر الشراء للوحدة	اسم المورد	رقم التليفون
ماوس usb fox	05-Nov-00	11.5	محمد صبري	010236565
ماوس fox usb	02-Mar-03	10.55	القدس	01156789
ماوس fox usb	28-Dec-00	10.5	القدس	01156789
ماوس fox usb	08-Dec-73	10	محمد صبري	010236565
cpu	06-Apr-05	180.5	جرين	0102687
cpu	03-Apr-05	175	الرواد	0126939877
cpu	06-Apr-02	180	محمد صبري	010236565
cpu 3.6	08-Apr-05	205	القدس	01156789
cpu 3.6	03-Apr-05	195	محمد صبري	010236565
cpu 3.6	05-Apr-05	170	القدس	01156789
cpu 3.6	03-Apr-05	165	جرين	0102687
cpu 3.6	21-Apr-05	201.5	الرواد	0126939877
DVD	06-Apr-05	92.5	جرين	0102687
DVD	03-Apr-05	101.5	الرواد	0126939877
DVD LG	03-Apr-05	100	الرواد	0126939877
DVD R/W LG	06-Apr-05	91.5	جرين	0102687
DVD R/W LG	03-Apr-05	100	الرواد	0126939877

جميع المنتجين الذي لهم تعاملات في جدول المبيعات وأيضا لهم أرقام تليفون من جدول المنتجين

One to many واحد إلى كثير ومعامل الربط فيه RIGHT JOIN
ويقوم الاستعلام الجديد بإحضار جميع السجلات من الجدول الثاني وفقط السجلات المرتبطة من الجدول الأول
فمثلا في المثال السابق لو استخدمنا معامل الربط المذكور يقوم بإحضار جميع السجلات من جدول المبيعات وفقط السجلات المرتبطة من جدول المنتجين

أي يقوم بإحضار جميع بيانات جدول المبيعات وفقط تليفون المنتج الذي له تعاملات في جدول المبيعات

اسم الصنف	تاريخ الانتاج	سعر الشراء للوحدة	اسم المورد	رقم التليفون
DVD	03-Apr-05	101.5	الرواد	0126939877
DVD LG	06-Apr-05	95	عمر بن الخطاب	
DVD LG	03-Apr-05	100	الرواد	0126939877
DVD R/W LG	03-Apr-05	100	الرواد	0126939877
DVD R/W LG	03-Apr-05	92	عمر بن الخطاب	
DVD R/W LG	09-Apr-05	95	عمر بن الخطاب	
flash 4G	06-Apr-07	52	الأصدقاء باب اللوق	
flash 4G	06-Apr-07	40	على	0151254
flash 4G	09-Apr-06	52.5	الأصدقاء باب اللوق	
flash 4G	09-Apr-06	41.5	على	0151254
flash 8G	03-Apr-05	92	الأصدقاء باب اللوق	
flash 8G	03-Apr-05	90	القدس	01156789
flash 8G	09-Apr-05	92	هشام	011555555555
flash 8G	01-Apr-08	92	الأصدقاء باب اللوق	
flash 8G	09-Apr-05	90	القدس	01156789
DVD R/W Sansong	09-Apr-05	100	عمر بن الخطاب	

Many to one كثير إلى واحد ومعامل الربط فيه LEFT JOIN

ويقوم الاستعلام الجديد بإحضار جميع السجلات من الجدول الاول وفقط السجلات المرتبطة من الجدول الثاني

فمثلا في المثال السابق لو استخدمنا معامل الربط المذكور يقوم بإحضار جميع السجلات من جدول المنتجين وفقط السجلات المرتبطة من جدول المبيعات

أي يقوم بإحضار جميع بيانات جدول المنتجين وفقط اسماء المنتجين المسجلين في جدول المنتجين

اسم الصنف	تاريخ الانتاج	سعر الشراء للوحدة	اسم المورد	رقم التليفون
				0121234
				012369448
ماوس fox usb	28-Dec-00	10.5	القدس	01156789
ماوس fox usb	05-Nov-00	11.5	محمد صبري	010236565
ماوس fox usb	08-Dec-73	10	محمد صبري	010236565
ماوس fox usb	02-Mar-03	10.55	القدس	01156789
cpu	03-Apr-05	175	الرواد	0126939877
cpu	06-Apr-05	180.5	جرين	0102687
cpu	06-Apr-02	180	محمد صبري	010236565
cpu 3.6	03-Apr-05	165	جرين	0102687
cpu 3.6	08-Apr-05	205	القدس	01156789
cpu 3.6	05-Apr-05	170	القدس	01156789
cpu 3.6	03-Apr-05	195	محمد صبري	010236565
cpu 3.6	21-Apr-05	201.5	الرواد	0126939877

ارقام تليفون منتجين ليس لهم تعاملات في جدول المبيعات وايضا من لهم تعاملات

استخدام ميزات الاستعلامات الجدولية مع لغة SQL

اولا نستخدم الاستعلامات الجدولية احد الصفوف لجعل بياناتها عناوين لاعمدة الاستعلام

فمثلا انظر هذا الاستعلام

```
SELECT count(class_name) AS [عدد الاصناف],
class_name, product_name
FROM buying
where  product_name in("جرين"،"القدس"،"الرواد")
GROUP BY class_name, product_name
order by product_name
```

اسم المورد	اسم الصنف	عدد الاصناف
الرواد	cpu	1
الرواد	cpu 3.6	1
الرواد	DVD	1
الرواد	DVD LG	1
الرواد	DVD R/W LG	6
الرواد	DVD R/W Sansong	4
الرواد	Ram 512 ddr2	5
القدس	usb fox ماوس	2
القدس	cpu 3.6	2
القدس	flash 4G	4
القدس	flash 8G	3
القدس	MB G.B G31	2
جرين	cpu	1
جرين	cpu 3.6	1
جرين	DVD	1
جرين	DVD R/W LG	2
جرين	DVD R/W Sansong	1
جرين	MB G.B G31	2
جرين	ram 1G ddr2	2

هذا استعلام يقوم بعد عدد الاصناف باستخدام دالة count وتقوم بعرض اسم الصنف واسم المنتج لهذا العدد – من جدول buying – ويتم التجميع عن طريق اسم المنتج واسم الصنف
ايضا باستخدام الشرط where للاصناف المذكورة فقط

ثم انظر الأتي

اسم الصنف	الرواد	القدس	جرين
fox usb ماوس		2	
cpu	1		1
cpu 3.6	1	2	1
DVD	1		1
DVD LG	1		
DVD R/W LG	6		2
DVD R/W Sansong	4		1
flash 4G		4	
flash 8G		3	
MB G.B G31		2	2
ram 1G ddr2			2
Ram 512 ddr2	5		

```
TRANSFORM Count(class_name)
SELECT buying.class_name
FROM buying
WHERE  buying.product_name  In
("جرين"،"القدس"،"الرواد")
GROUP BY class_name
PIVOT product_name;
```

هذا بعد استخدام ميزة الاستعلام الجدولي وجعل بيانات صف اسم المنتج product_name هي عناوين لأعمدة الاستعلام وتحويل البيانات [مجموع الاصناف] افقية بدل من راسية

كما يمكن إضافة حقل المجموع الكلي للأصناف كما يلي

```
TRANSFORM Count(class_name) 
SELECT buying.class_name,
Count(buying.class_name) AS [عدد الاصناف]
FROM buying
WHERE  buying.product_name  In
("جرين"،"القدس"،"الرواد")
GROUP BY class_name
PIVOT product_name;
```

اسم الصنف	عدد الاصناف	الرواد	القدس	جرين
fox usb ماوس	2		2	
cpu	2	1		1
cpu 3.6	4	1	2	1
DVD	2	1		1
DVD LG	1	1		
DVD R/W LG	8	6		2
DVD R/W Sansong	5	4		1
flash 4G	4		4	
flash 8G	3		3	
MB G.B G31	4		2	2
ram 1G ddr2	2			2
Ram 512 ddr2	5	5		

في المثال السابق يقوم الاستعلام بعرض كل المنتجين كعناوين للاعمدة لو اردنا الاقتصار على منتج واحد او اثنين يمكن ذلك عن طريق جملة pivot

```
TRANSFORM Count(class_name)
SELECT class_name, Count(class_name) AS [عدد الاصناف]
FROM buying
WHERE  product_name
GROUP BY class_name
PIVOT product_name in ("جرين"،"القدس"،"الرواد")
```

اسم الصنف	عدد الاصناف	جرين	القدس	الرواد
	0			
usb fox ماوس	4		2	
cpu	3	1		1
cpu 3.6	5	1	2	1
DVD	2	1		1
DVD LG	2			1
DVD R/W LG	11	2		6
DVD R/W Sansong	7	1		4
flash 16G	4			
flash 4G	9		4	
flash 8G	6		3	
MB G.B G31	4	2	2	
ram 1G ddr2	4	2		
Ram 512 ddr2	5			5

ولاحظ عند كتابة الشرط في جملة pivot قام البرنامج بادراج الحقول المختارة فقط ولكن لم يقم بترشيح القيم الكلية [عدد الاصناف] لاننا لم نستخدم جملة where وهذا هو الفرق بين استخدام الشرط في where او استبعاد بعض الحقول في جملة pivot

أما إذا كتبت الاستعلام بهذا الشكل

```
TRANSFORM Count(class_name)
SELECT class_name
FROM buying
WHERE product_name
GROUP BY class_name
PIVOT product_name ="جرين"
```

ينتج هذا الاستعلام

اسم الصنف	-1	0
usb fox ماوس		2
cpu	1	1
cpu 3.6	1	3
DVD	1	1
DVD LG		1
DVD R/W LG	2	6
DVD R/W Sansong	1	4
flash 4G		4
flash 8G		3
MB G.B G31	2	2
ram 1G ddr2	2	
Ram 512 ddr2		5

وهذا يعني الحقل الاول والذي عنوانه [-1] يعني عدد الاصناف لهذا المنتج [جرين] والحقل الثاني والذي عنوانه [0] يعني عدد الاصناف لباقي المنجين غير هذا المنتج

ملحوظة هامة اذا قمت بكتابة احد البيانات المماثلة لأحد المنجين خطأ مثلا حرين بدل جرين يقوم بإدراج حقل خالي من البيانات ولا يدرج الحقل المعني لانه يعتبر الحقل حقل جديد ويبحث له عن بيانات فلا يجد

ملحوظة هامة في حالة انشاء نموذج او تقرير يعتمد على بيانات الاستعلام الجدولي يحدث خطا كبير لانه عندما يتم اضافة منتج جديد في بيانات الجدول او الاستعلام المصدر للاستعلام الجدولي يقوم باضافة حقل في الاستعلام الجدولي لهذا المنتج ولكن لا يقوم باضافته في النموذج او التقرير ويمكن حل هذه المشكلة عن طريق

يمكنك اضافة كافة المنتجين او الموردين لاحد المنتجات في جدول منفصل سواء تم التعامل معه هذه السنة ام لا أي اضافة كافة الموردين المتوقعين للتوريد وليكن جدول product

ثم يمكنك الاستعلام كما يلي

```
TRANSFORM Count(buying.class_name)
SELECT buying.class_name
FROM buying RIGHT JOIN product ON
(buying.product_name=product.product_name)
GROUP BY buying.class_name
PIVOT product.product_name;
```

حيث يتم ربط جدول المنجين بجدول المبيعات ويتم احضار اسماء الموردين من جدول المنتجين product وفي نتيجة الاستعلام سيتم ادراج كل المنتجين سواء من لهم تعاملات ام لا والمنتج الذي ليس له تعاملات سيتم ادراجة بصفر هكذا

اسم الصنف	الرواد	القدس	جرين	حاتم ابراهيم	خالد سعودى	على	محمد صبري	محمد عبد الوهاب	هشام
usb fox ماوس		2					2		
cpu	1		1				1		
cpu 3.6	1	2	1				1		
DVD	1		1						
DVD LG	1								
DVD R/W LG	6		2						
DVD R/W Sansong	4		1						
flash 16G									2
flash 4G		4				3			
flash 8G		3							1
MB G.B G31		2	2						
ram 1G ddr2			2				2		
Ram 512 ddr2	5								

ثم إنشاء النموذج أو الاستعلام بعد ذلك وعند التعامل مع منتج من هؤلاء المدرجين بعد ذلك يتم إحضار بياناته

لغة SQL والإجراءات

وتعني الاجراءات التي يمكن تفعيلها على الجداول مثل انشاء جدول وحذفه والتعديل فيه ونحن هنا لسنا بصدد الانشاء والتعديل حيث يمكننا ذلك بسهولة عن طريق الاكسس ولكن التعديل يكون من الاهمية بمكان في لغة الاستعلامات SQL ولاحظ هذه الامثلة

لو تم خطا تسجيل احد بيانات الصنف مثل سعر الشراء او تاريخ الصلاحية واردنا تعديل هذه البيانات فلا بد من المرور على كل السجلات التي حدث فيها الخطأ ثم تعديل البيانات وما يترتب عليه من مشقة واحتمالية الخطا او نسيان بعض السجلات ولكن مع الامر تحديث Update يكون الامر بمنتهى السهولة

Update employee ← (جدول)

Set price_in = 20 (حقل)

Where class_code = 101 (حقل)

وهذا الاستعلام يقوم بتحديث بيانات سعر الصنف الذي كوده 101 الى السعر الجديد 20 بدل الموجود وذلك لكل السجلات

ويمكن ايضا زيادة السعر القديم بنسبة معين

فمثلا في المثال السابق لو اردنا زيادة سعر هذا الصنف بنسبة 5٪ نكتب التالي

```
Update employee
Set price_in = price_in*1.05
Where class_code = 101
```

ويمكن ايضا انشاء حقل سعر البيع للمنتج وتركه خاليا ثم يتم ادخال سعر البيع عن طريق هذا الاجراء بزيادة نسبة ربح مثلا 20٪ على سعر الشراء هكذا

```
Update employee
Set price_out= price_in*1.2
Where class_code = 101
```

في بعض الاحيان يتم تسجيل بيانات الاصناف بما يسمى اذون الاضافة وفي هذا النوع من التسجيلات تكون الاصناف متشابهة ولكن يمكن اعطاء اسعار مختلفة للعملاء وفي كل اذن كثير من الاصناف المكررة والعملاء ولتسجيل البيانات بطريقة سريعة وصحيحة نقوم بتسجيل البيانات بدون اسعار ثم نقوم بتخصيص استعلام تحديث كما سبق بالاسعار طبقا للشروط

مثال : لتسجيل صنف MB G41 للمورد القدس بسعر 300 جنيه نقوم بالاتي

```
Update employee
Set price_out= 300
Where suplyer_name = "القدس" and class_name = " MB G41"
```

موسوعة العبارات المحجوزة للغة SQL

ABSOLUTE	CLOSE	DAYOFMONTH	FORMAT
ADD	CLUSTERED	DAYOFWEEK	FORMATC
ADMINDB	COALESCE	DAYOFYEAR	URRENCY
ALL	COLLATE	DBPASSWORD	FORMATD
ALPHANUMERIC	COLLATION	DCOUNT	ATETIME
ALTER	COLUMN	DDB	FORMATN
AND	COMMIT	DEC	UMBER
ANY	COMP	DECIMAL	FORMATP
ARE	COMPRESSI	DECLARE	ERCENT
ARRAY	ON	DELETE	FROM
AS	CONCAT	DESC	FROM
ASC	CONNECT	DFIRST	CLAUSE
ASCB	CONNECTIO	DISALLOW	FV
ASCII	N	DISCONNECT	GENERAL
ASCW	CONSTRAIN	DISTINCT	GRANT
ASSERTION	T	DISTINCTROW	GROUP
ATAN	CONSTRAIN	DLAST	GUID
ATN	TS	DLOOKUP	GUIDFRO
AUTHORIZATION	CONTAINER	DMAX	MSTRING
AUTOINCREMEN	CONTAINS	DMIN	HAVING
T	CONVERT	DOMAIN	HEX
AVG	COS	DOUBLE	HOUR
BEGIN	COUNT	DROP	HYPERLIN
BETWEEN	COUNTER	DSTDEV	KPART
BINARY	CREATE	DSTDEVP	IDENTITY
BIT	CSNG	DSUM	IEEEDOUB
BOOLEAN	CSTR	DVAR	LE
BOTH	CURDATE	DVARP	IEEESINGL
BUILDCRITERIA	CURRENCY	ENVIRON	E
BY	CURRENTUS	EQV	IGNORE
BYTE	ER	ERROR	IIF
CASCADE	CURSOR	EUROCONVERT	IMAGE
CATALOG	CURTIME	EVAL	IMESTATU
CBOOL	CVAR	EXCLUSIVECO	S
CBYTE	CVDATE	NNECT	IMP
CCUR	CVERR	EXEC	IN
CDATE	DATABASE	EXECUTE	INDEX
CDBL	DATE	EXISTS	INDEXCRE
CDEC	DATEADD	EXP	ATEDB
CEILING	DATEDIFF	EXTRACT	INNER
CHAR	DATEPART	FALSE	INPUT
CHARACTER	DATESERIA	FETCH	INPUTBOX
CHECK	L	FILTER	INSENSITI
CHOOSE	DATETIME	FIRST	VE
CHR	DATEVALU	FIX	INSERT
CHRB	E	FLOAT	INSTR
CHRW	DAVG	FLOOR	INSTRB
CINT	DAY	FN	INSTRREV
CLNG	DAYNAME	FOREIGN	INT

INTEGER
INTERVAL
INTO
IPMT
IRR
IS
ISARRAY
ISDATE
ISEMPTY
ISERROR
ISMISSING
ISNULL
ISNUMERIC
ISOBJECT
ISOLATION
JOIN
KEY
LANGUAGE
LAST
LBOUND
LCASE
LEFT
LEFTB
LEN
LENB
LENGTH
LEVEL
LIKE
LOCATE
LOG
LOGICAL
LONG
LONGBINARY
LONGCHAR
LONGTEXT
LOWER
LTRIM
MATCH
MAX
MEMO
MID
MIDB
MIN
MINUTE
MIRR
MOD
MONEY
MONTH
MONTHNAME
MSGBOX
NATIONAL
NCHAR
NONCLUSTE
RED
NOT
NOW
NPER
NPV
NTEXT
NULL
NUMBER
NUMERIC
NVARCHAR
NZ
OCT
OLEOBJECT
ON
OPEN
OPTION
OR
ORDER
OUTER
OUTPUT
OWNERACC
ESS
PAD
PARAMETE
RS
PARTIAL
PARTITION
PASSWORD
PERCENT
PIVOT
PMT
POSITION
POWER
PPMT
PRECISION
PREPARE
PRIMARY
PRIVILEGES
PROC
PROCEDURE
PUBLIC
PV
QBCOLOR
QUARTER
RAND
RATE
REAL
REFERENCE
S
REPLACE
RESTRICT
REVOKE
RGB
RIGHT
RIGHTB
RND
ROLLBACK
ROUND
RTRIM
SCHEMA
SECOND
SELECT
SELECTSCHEM
A
SELECTSECURI
TY
SET
SGN
SHORT
SIGN
SIN
SINGLE
SIZE
SLN
SMALLDATETI
ME
SMALLINT
SMALLMONEY
SOME
SPACE
SPLIT
SQL
SQLCODE
SQLERROR
SQLSTATE
SQR
SQRT
STDEV
STDEVP
STR
STRCOMP
STRCONV
STRING
STRINGFROMG
UID
STRREVERSE
SUBSTRING
SUM
SWITCH
SYD
SYSNAME
TABLE
TABLEID
TAN
TEMPORA
RY
TEXT
TIME
TIMER
TIMESERI
AL
TIMESTAM
P
TIMEVALU
E
TINYINT
TO
TOP
TRAILING
TRANSAC
TION
TRANSFOR
M
TRANSLAT
E
TRANSLAT
ION
TRIM
TRUE
TYPENAM
E
UBOUND
UCASE
UNION
UNIQUE
UNIQUEID
ENTIFIER
UNKNOW
N
UPDATE
UPDATEID
ENTITY
UPDATEO
WNER
UPDATESE
CURITY
UPPER
USAGE
USER
USING
VAL

VALUE	VARTYPE	WEEKDAYNAME	WORK
VALUES	VARYING	WHEN	XOR
VAR	VIEW	WHENEVER	YEAR
VARBINARY	WEEK	WHERE	YESNO
VARCHAR	WEEKDAY	WITH	ZONE
VARP			