تحسين عملية معالجة البيانات في قواعد البيانات الموزعة باستخدام نظام الوقت الفوري

سعدي حمد ثلج اللهيبي

قسم علوم الحاسوب ، كلية علوم الحاسوب والرياضيات ، جامعة تكريت ، تكريت ، العراق

altage_2004@yahoo.com

(تاريخ الاستلام: 6 / 5 / 2012 ---- **تاريخ القبول:** 19 / 7 / 2012)

الملخص:

في هذا البحث تم أنشاء نظام قاعدة بيانات تقوم على شطر (Dividing) وتقسيم الجدول الواحد الى عدة مستخدمين للاستفادة من المعلومة داخله وتوزيعها واستخدامها من قبل اكثر من مستخدم ، واخذ صورة لها في ملفات ذاكرة مؤقتة الـ (Buffer) وربطها في المصدر الـ(Binding Source) في قواعد البيانات الـ (Database) ، وإعطاء تفويضات محددة لكل مستخدم مما يتيح لمستخدمي النظام ضمان الامنية التامة على بياناتهم في الـ Database وكذلك تم اعتماد نظام المحادثة الداخلية والرسائل (Chatting & Massaging) لمستخدمي النظام ، حيث يمكن من خلالها مخاطبة جميع المستخدمين برسالة واحدة مهما كان عددهم بمجرد تحديد المستخدم المطلوب مخاطبته . وتم أيضا عمل نظام للبحث المتعدد الطلبات حيث تم استخدام 24 حقل للبحث باستخدام الزمن الحقيقي(Real Time) في عملية الاستعلام ويتمكن المستخدم بالبحث عن طريق أي معلومة متوفرة لدية ضمن الحقول المتعددة.

أثبت النظام قدرته على الوصول إلى المعلومة المطلوبة من قبل عدة مستفيدين بسرعة جيدة .

1. المقدمة:

قاعدة البيانات الموزعة تعرّف بأنها تشمل على مجموعة من البيانات المخزونة والمنظمة بطريقة تفي بمتطلبات المستخدمين لكل البيانات ، ويتم تخزين البيانات التي تكون في شكلٍ هيكلي يسهم في تطوير التطبيقات مستقبلاً، وتخزن قاعدة البيانات مؤشرات تعريفية بملفاتها المختلفة وسجلاتها المتنوعة في إطار ملف رئيس(Master File) الذي يستخدم بصفةٍ مشتركةٍ لكل ملفات قواعد البيانات .

أن أهمية نظم قواعد البيانات الموزعة تعمل على تقليل البيانات المكررة مما يعني أن البرامج تحفظ بسهولة أكبر تسهيلات الاستدعاء يمكن تكاملها ومركزيتها بدلاً من برمجتها في كل تطبيق قاعدة البيانات وهي الحل لكثير من مشاكل المعالجة التي تواجهها مراكز معالجة البيانات في المنظمات المختلفة، ونظرياً يمكن النظر إلى قاعدة البيانات على أنها مشروع بيانات مشتركة تترابط فيه البيانات معاً، وهذا يعني أن تنظيم البيانات تتابعياً (Sequential) وأن المعالجة تتم بشكل متسلسل (Serial) ، حيث أن التخزين التتابعي (Sequential Storage) يتم فيه تخزين السجلات المختلفة بملحقٍ معينٍ في مواقع ماديةٍ متجاورة ومتتابعة في وسيلة التخزين المستخدمة ، وبذلك فإن التسلسل المادي للسجلات الخاصة بأي ملفٍ هو التسلسل المنطقي نفسه العائد له [1].

وأن زيادة التطورات التكنولوجية الحديثة المتصلة بالمعالجة المباشرة والنظم المتكاملة (Integrated Systems) والعلاقات المتداخلة بين البيانات أدى إلى ظهور "نظم إدارة قواعد البيانات (DBMS) Database Management System التي تشمل لغة الاستفسار (Query Language) أي التعامل مع البيانات من خلال أسئلة محددة وواضحة توجه للنظام ، وأن تركيز المستفيدين ينصب على نظم المعرفة كأساس لقواعد البيانات وظهور برامج متطورة منها لغة SQL Server ولغة ORACLE ....الخ. والجدير بالذكر أن بنك المعلومات لا ينتج المعلومة، ولكن يؤهلها لتصبح بمتناول المستفيدين من خلال معالجة تساعد في التواجد والبحث عن الوثيقة والمعلومات المطلوبة عند الحاجة [1].

كما قدم (Sanjay) عرض قاعدة بيانات واسعة النطاق لتحسين المرونة باستعمال خوارزميات الانقسام الساكن وتطبيقها على قواعد البيانات الموزعة بأكملها للحصول على الاجابة بالسرعة والدقة العالية [2].

2. المفاهيم الأساسية :

أصبحت نظم قاعدة البيانات جزءاً رئيسياً من علم الحاسوب الحديثة. وقد تم إنتاج عدة برنامج تستخدم فى حفظ البيانات والمعلومات خاصة بكل مجالات العمل المحوسب وتجهيز تقارير على شكل جداول منظمة والاستفسار عن البيانات باستخدام لغة الاستعلام (SQL) .

حيث تتسم قواعد البيانات بأنها ضخمة جداً ، إلا أنه ما زال هنالك طريقة لبنائها وذلك ما نحن بصدد القيام به الآن ولكن على مراحل تدريجية والتي هي: (Server & Client, SQL Server, Real Time System, DataBase).

إذ يمكن أن نقسم المهمة المتعلقة ببناء قاعدة البيانات إلى عدة مراحل ثم نقسم هذه المراحل إلى مهام أصغر يمكن إدارتها ومن ثم يتعين علينا أن نتناول كل مرحلة على حدة .

2-1 قواعد البيانات :

قاعدة البيانات (Database) وهي مجموعة جداول مرتبطة مع بعضها بطريقة علائقية(Relational Tables) ويتكون الجدول من سجل (Record) او أكثر من سجل ويتكون السجل من حقل (Field) أو أكثر من حقل ومثال على السجل :السجل الخاص بموظف معين يتكون من عدة حقول مثل رقم الموظف – اسم الموظف – درجة الموظف – تاريخ التعيين – الراتب – والقسم التابع له ...إلخ من بيانات الموظف تخزن في جهاز الحاسوب عَلى نحو منظّم، حيث

يقوم برنامج يسمى مدير قاعدة البيانات (DBMS) بتسهيل التعامل معها والبحث ضمن هذه البيانات، وتمكين المستخدم من الإضافة والحذف والتعديل عليها [3].

2-2 نظام الوقت الحقيقي (Real Time system)

أن نظام الزمن الحقيقي هو احد الانظمة التي تعمل في التوقيت الحرج ، أن في بعض المجالات التي يستخدم فيها النظام يمكن يشكل خطراً على حياة البشر وعلى سبيل المثال أنظمة السيطرة على ملاحة الطائرات في المطارات الدولية ، ويُمْكِنُ أَنْ يُؤَدّي إلى تسبّب ضررِ كبير مثل خسائر ملايين الدولارات .

في نظامِ الوقتِ الحقيقي، يتطلب حاسوب ذو مواصفات عالية ليَعمَلُ بسرعة عالية بما فيه الكفايةُ لمُجَاراة تَفَاعُل المستخدمين مَعه، ليَدْعمُ الحاسوبَ تَفَاعُل الأشخاصِ أَو الشخصِ مَعه ويُزوّدانِ بالسماحيات الخاصة مثل الوصول، الاسترجاع، وعملية الخزن ، وذلك عادة خلال النوعِ نفسهِ مِنْ نظامِ إدارةِ قاعدةِ البيانات، بالإضافة إلى قوّةِ معالجةِ البيانات الحسابيةِ [4].

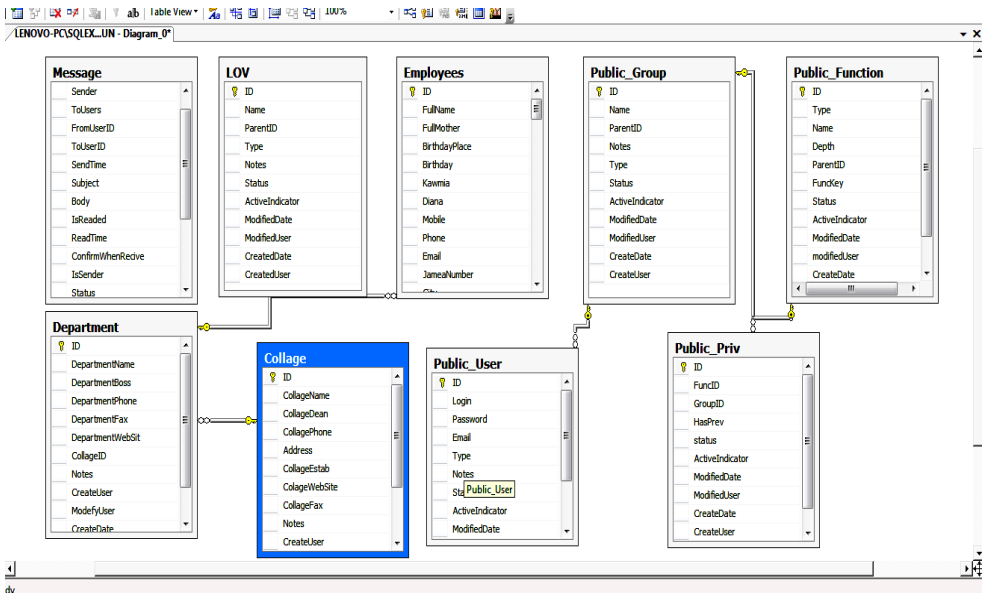
نظام الزمن الحقيقي هو نظام أي (Software-Hardware) أي مجموعة من الاحداث المتفاعلة مع بعضها لتحقيق هدف معين. وفي هذا النظام الحقيقة لا نعتمد فقط على النتائج المنطقية للحسابات وإنما على الزمن الذي يجب أن تنجز النتائج خلاله.

أن مستخدمي قواعد البيانات عادةً ليس لديهم الخبرة بالوقت الحقيقي وعن آلية سير قاعدة البيانات ويعتقدون أن عمل قاعدة البيانات لديهم مقبولة في أدارة أنظمة قواعد البيانات التجارية [4].

3-2 لغة الاستعلام (SQL Server) STRUCTURED QUERY LANGUAGE

سميت لغة الاستعلام أو لغة البحث وتستخدم غالباً في برامج مدير قواعد البيانات في إنشاء أو تدمير الملفات كما تستخدم في البحث الانتقائي لسجلات دون غيرها بدلاً من استعراض كل الملفات للوصول إلى السجلات المطلوبة، و في الحاسبات الشخصية تستخدم لغات مطورة من لغة البحث في حزم البرامج المعروفة باسم قواعد البيانات مثل DB4,DB3+ وما شابه من حزم برامج.

كما تعتبر لغة SQL Server لغة علائقية تستخدم العلاقات بين الجداول، حيث أ العلاقات هي بدرجة أهمية بالغة ، فهي المسيطر على تكامل قاعدة البيانات ، يمكن القول أن قاعدة بيانات بدون علاقات هي ليست قاعدة بيانات [5]، والشكل رقم (1) يوضح كيفية ربط الجداول مع بعضها البعض باستخدام العلاقات (Relationship) .



الشكل رقم (1) يبين عملية ربط الجداول في لغة SQL Server

ويمكن استخدام لغة الاستعلام بأسلوبين :

- **الاسلوب الأول :**

أسلوب التعامل المباشر مع البيانات (INTERACTIVE) باستخدام التعليمات التالية :

- أنشأ جدول CREATE TABLE
- أنشأ مشهد CREATE VIEW
- أنشأ فهرس CREATE INDEX
- انتقل جدول TABLE DROP
- أظهر DROP VIEW
- فهرس DROP INDEX

الاسلوب الثاني :

كما تستخدم عناصر لغة الاستعلام ضمن إحدى لغات البرمجة الإجرائية: مثل الكوبول، و هذا يوضح لنا أن لغة الاستعلام ليست لغة إجرائية فهي تصف ما نريد (WHAT) وليس كيف ننجز ما نريد (HOW) حيث يتولى مدير قاعدة البيانات الخطوات الإجرائية دون أن تدخل من المستخدم فيما يسمى السياحة الآلية في النظام [6] .

4-2 الخادم والزبون (Server & Client)

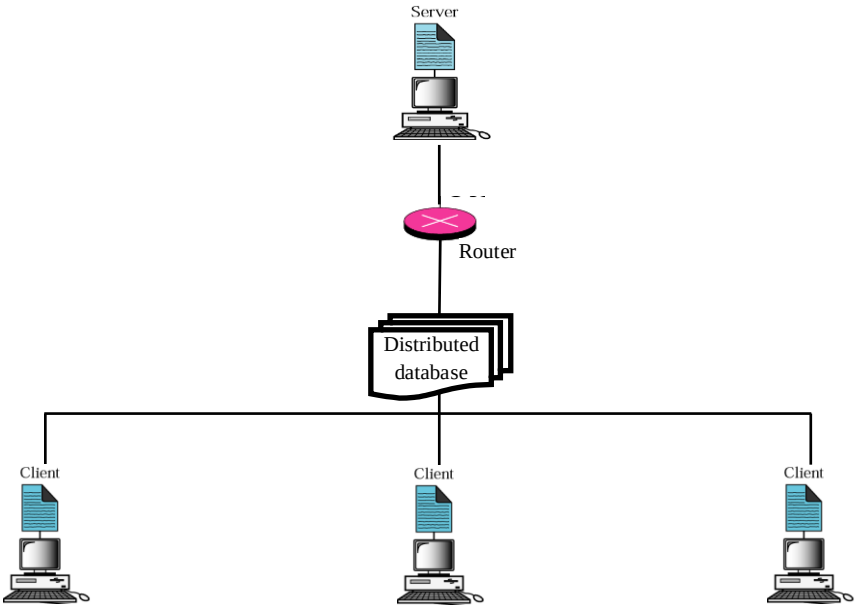
الخـادم Server هـو الخـاص بتشـغيل الشـبكة (Network operating system) ويقدم الخدمات لكل محطات العمل الموجودة على الشبكة ومن الخدمات التي يقدمها [7] :

- تخزين الملفات
- أدارة المستخدمين
- التأمين
- الأوامر الخاصة بالشبكة
- أدارة النظام

الزبون Client

وهـي عبـارة عـن المحطـات الطرفيـة والتـي تكـون مرتبطـة بالخـادم (Server) ، حيث يتم تبادل المعلومات ما بين الخادم والزبون عن طريق (IP, Internet Protocol) المخصص للكل محطة طرفية ، ويعتبـر الزبـون (Client) بمثابـة واجهـة (Interface) للمسـتخدمين [8].

والشكل رقم (2) يوضح عمل الخادم والزبون.



الشكل رقم (2): يوضح هيكلية الربط بين الخادم والزبون (Client – Server)

3. الاجراء العملي :

تم جمع معلومات عامة عن كل وحدات وتشكيلات الجامعة ومن خلال ذلك تم تحديد مدخلات النظام ، حيث تم الأخذ بنظر الاعتبار كل تفاصيل ومفردات المعطيات المطلوبة من قبل الوحدات الادارية بالجامعة . ثم تم تحليل تلك المعطيات وعلى ضوء ذلك تم بناء النظام الحـالي الـذي يشـمل مراحـل وخصوصـيات وسـماحيات عديـدة لكـل مستخدم من قبل أي وحدة ادارية داخل الجامعة وذلك وفق شروط امنية وحماية محددة لكل مستخدم .

3-1 طريقة جلب البيانات من قاعدة البيانات الموزعة:

1- يقوم المستخدم الاول بطلب البيانات من ال DATABASE.

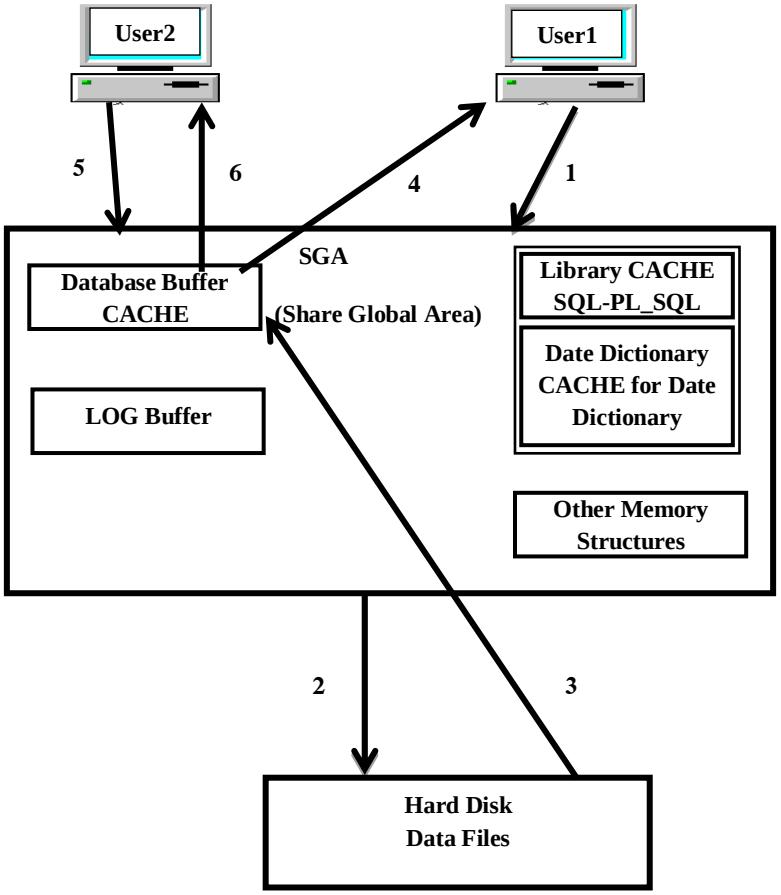
2- يقوم الـ Server Process بدراسة الطلب ويحضر البيانات من الـ Data Files على اعتبار ان البيانات المطلوبة ليس مخزنه في الـ Database Buffer Cache.

3- يتم احضار البيانات من ال Data Files وتخزن في الـ Database Buffer Cache.

4- يتم إظهار البيانات المطلوبة للمستخدم الاول.

5- يقوم مستخدم أخر بعد قليل بطلب ذات البيانات نفسها من الـ Database.

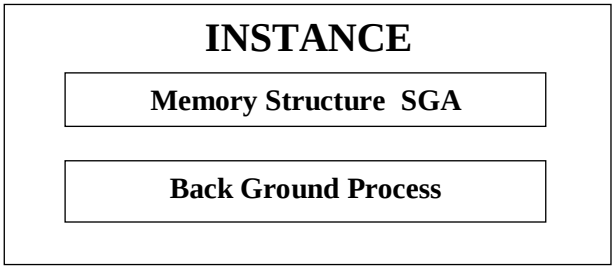
6- يقوم الـ Instance بدراسة الطلب فيجد ان البيانات المطلوبة تم استخدامها قبل قليل من الـ Data Files وما زالت مخزونة في الـ Database Buffer Cache فيستخرج البيانات من الـ Database Buffer Cache دون الحاجة الى استدعاء البيانات من القرص الصلب حيث توجد نسخة منها في ملفات الـ Data Files ومن ثم يتم إظهار البيانات للمستخدم الثاني بسرعة اكبر وجهد أقل وكما مبين بالشكل رقم (3).



الشكل رقم(3) آلية جلب البيانات من قاعدة البيانات

3-1-1 SHARE GLOBAL AREA (SGA) :

وهي هيكلية الذاكرة Memory Structure. الخاصة بال Instance. الموضح بالشكل (4) والتي هي أحد مكونات الـInstance وتحجز المساحة الخاصة لها من الذاكرة عند تشغيل الـ Instance وهي ذاكرة مشتركة بين الاوامر (Processes) التي تأتي للـ Instance مثل الـSQL Query Process وأيضا مشتركة بين مستخدمي قواعد البيانات المختلفين. وتعرف أيضا بـ System Global Area [9].



الشكل رقم(4) يوضح Memory Structure for Instance

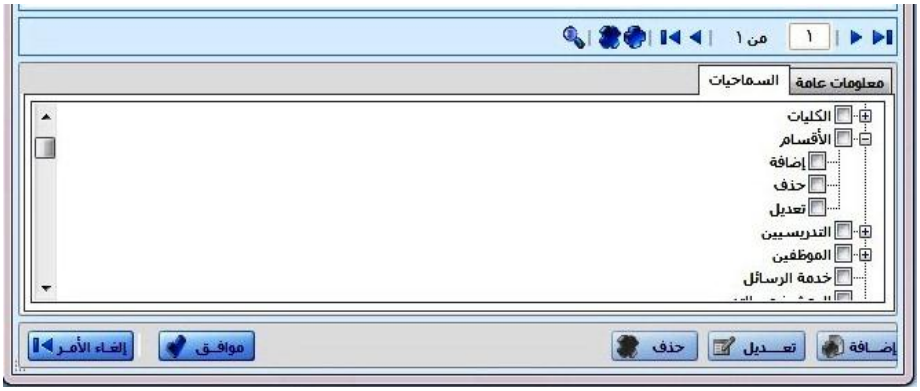
3-2 منح سماحيات الدخول للمستخدمين :

يتم اعطاء سماحية للدخول الى النظام وكل مستخدم له أسم خاص بالنظام (User Name – UN) وكلمة مرور (Password) كما موضح بالشكل(5) وهذا يعطي أكثر خصوصيه وأمنية للبيانات الخاصة بكل مستخدم .

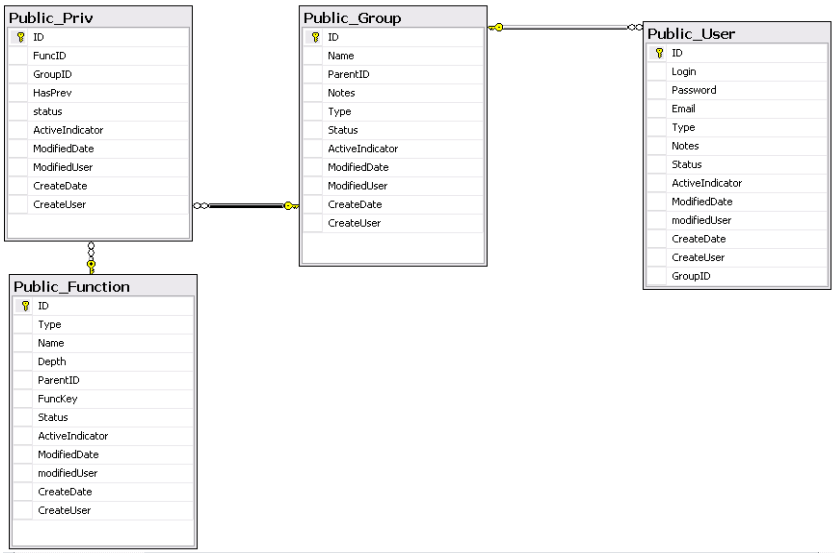
الشكل رقم (5): واجهة الدخول الى النظام الموزع

1-2-3 : منح سماحيات الاستخدام للمستخدمين :

في هذه الفقرة يتم إعطاء السماحيات لكل مستخدم حيث أنه توجد (58) نوع من السماحيات لاستخدام النظام ويتم تحديد طلب السماحيات من قبل مدير النظام(Server) وكما في الشكل (6) ، حيث أن هذه السماحيات تحتوي معلومات خاصة بالمستخدمين الذين يستخدمون النظام وهذه المعلومات تكون مخزونه بجداول معزولة عن الجداول الرئيسية كما موضح بالشكل رقم (7) وعلى سبيل المثال بيانات الكادر التدريسي او الوظيفي، بالإضافة للصلاحيات الموجودة في النظام وحسب الحاجة والمجموعة التي ينتمي إليها مدخل البيانات، على سبيل المثال اذا كان المستخدم من كلية العلوم وطلب من هذا المستخدم اضافة البيانات الجديدة فقط فبهذا سوف يعطى له سماحية الاضافة فقط اما الحذف والتعديل والبحث والسماحيات الأخرى لا يسمح له بها.



الشكل رقم (6) واجهة تحديد السماحيات للمستخدمين



الشكل رقم (7) جداول السماحيات المحددة المستخدمين

3-3 البحث في النظام: تم استخدام 24 حقل للبحث المتعدد المعلومة بالسرعة القصوى ، الاغراض كما مبين بالشكل (8) حيث يتاح للمستخدم الوصول الى

الشكل (8) : واجهة البحث المتعدد الاغراض

وان البيانات تكون مخزونه بنفس الـ Source Binding حيث ان المستعلم لا يرجع الى ال Data Set في كل مرة يتم إحضار العناوين جميعها وطريقة البحث بالدقة المتناهية حيث بمجرد كتابة الحرف الاول ستظهر كل البيانات التي تحتوي على ذلك الحرف باستخدام صيغة (%) في لغة C# وكما مبين في الاجراء البرمجي الاتي:

الكود البرمجي بلغة C# :

```
int TYPE;
public SearchForm(int type)
{
InitializeComponent();
takreidataset = Program.TakreitUnDataSet;
this.TYPE = type;

EmployeesTableAdapter.FillByType(takreidataset.E
mployees,this.TYPE);
}
```

```
private void SearchForm_Load(object sender,
EventArgs e)
{
EmployeesBindingSource.DataSource =
takreidataset;
}

private void
NameSearchTextBox_TextChanged(object sender,
EventArgs e)
{
EmployeesBindingSource.Filter = "FullName
Like \'" + NameSearchTextBox.Text + "%\'";

takreidataset.Employees.DefaultView.RowFilter =
"FullName Like \'" + NameSearchTextBox.Text +
"%\'";
}
```

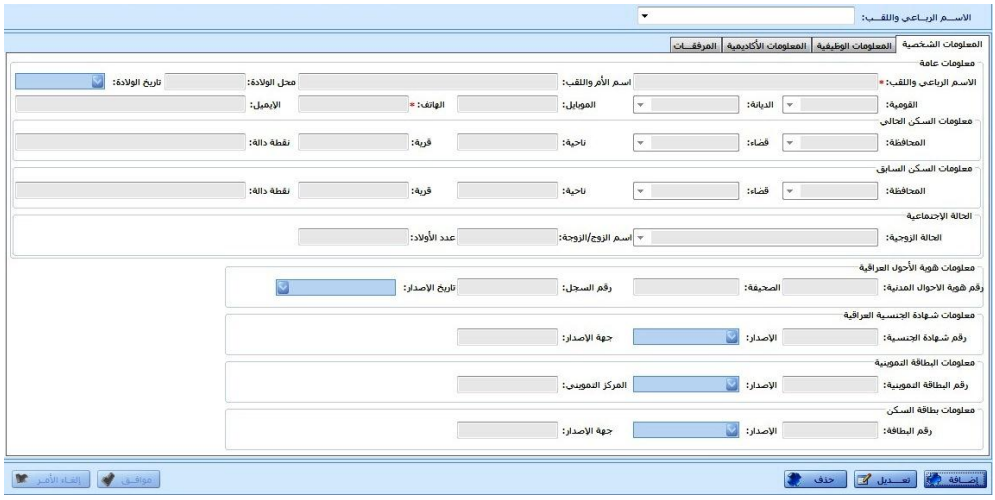
آلية استدعاء البيانات من الجداول التي يتم عرضها على المستخدم.

```
private void NameSearchTextBox_TextChanged(object sender, EventArgs e)
        {
          EmployeesBindingSource.Filter = "FullName Like \'" +
NameSearchTextBox.Text + "%\'";
            takreidataset.Employees.DefaultView.RowFilter = "FullName
Like \'" + NameSearchTextBox.Text + "%\'";
        }
```

هذا الاجراء يعني أي حرف مشابه للحرف المكتوب سيظهر في البحث مثلا عند كتابة الحرف م في مربع البحث سيظهر أي أسم يحتوي الحرف م

4 النتائج:

تم تطبيق النظام باستخدام لغة C# .NET ولغة قواعد البيانات SQL SERVER على حاسبة لاب توب ذو مواصفات (CORE i5 INTEL,H.D 350 GB,RAM 6GB,O.S Windows 7) وتم بناء النظام بواجهة واحده فقط مصممه بلغة C# كما موضح في الشكل (9) لإدخال البيانات الى الجداول المصممة بلغة الـ SQL SERVER وتم تنصيب هذه الواجهة على حاسبة Server وإعطائه (192.168.0.20) Virtual IP ،وربط (TP-LINK) ROUTER على حاسبة الـServer وكذلك تكون هذه الواجهة نفسها موزعة على عدد غير محدود من المستخدمين Clients ، حيث أن واجهة المستخدم تمكنه من الإدخال فقط كون البيانات ترسل الى الـ Server لكي تخزن في قاعدة البيانات الموزعة الرئيسية ، وهي تعتبر مستودعاً للبيانات .



الشكل رقم (9) يبين شكل واجهة أدخال البيانات وتحتوي على 4 صفحات متداخلة (Tabs)

وكما مبين في المقطع البرمجي الآتي:

```
{
InitializeComponent();
this.EmpType = type;
random = new Random();
takreitdataset = Program.TakreitUnDataSet;
if (takreitdataset.Employees.Rows.Count == 0)

EmployeesTableAdapter.FillBycollageid(takreitdataset.Employees, Program.GollageID,this.EmpType);
if (takreitdataset.Department.Rows.Count == 0)

DepartmentTableAdapter.FillBycollage(takreitdataset.Department,Program.GollageID);
LovTableAdapter.Fill(takreitdataset.LOV);
if (this.EmpType == (int)Program.EmployeesType.Tadresei)
{
FunctionKey = "007";
CourcegroupBox.Visible = false;
this.Text = " بيانات التدريسيين;"
}
else
{
```

4-1 التطبيق:

تم تطبيق هذا النظام في جامعة تكريت كقاعدة بيانات موزعة على (30) وحدة ادارية مختلفة من كليات واقسام الجامعة الادارية ، ومن خلال ذلك توصلنا الى نتائج جيدة من حيث الدقة Accuracy في عمليات الادخال مثل (الاضافة ، الحذف والتعديل) وكذلك استرجاع المعلومات .

تم احتساب زمن الوصول المستغرق لعدد من المستخدمين في كل تجربة وكذلك احتساب عدد البيانات التي فقدت اثناء عملية الادخال ، ومن خلال استخدام المعادلة التالية تم الحصول على معدل الفقدان وكما مبين في الجدول رقم (1) :

$$S = \frac{L}{N} \quad \ldots\ldots\ldots\ldots(1)$$

حيث أن :

S = معدل فقدان البيانات في كل تجربة .

L = عدد البيانات المفقودة في كل تجربة .

N = عدد المستخدمين في كل تجربة .

والجدول رقم (1) يبن النتائج التي تم الحصول عليها من المعادلة اعلاه.

الجدول رقم (1) يبن نتائج معدل الفقدان

التجربة	عدد المستخدمين (N)	عدد البيانات المفقودة (L)	معدل الفقدان (S)
1	1	0	0
2	5	0	0
3	10	0	0
4	15	2	0.13
5	20	5	0.25
6	25	6	0.24
7	30	6	0.20

كذلك تم الحصول على معدل الخطأ لكل تجربة من خلال استخدام معدل الفقدان التي تم الحصول عليها من المعادلة (1) ، وذلك بتنفيذ المعادلة (2) ادناه :

$$Er. = \frac{S}{T} \quad \text{.....(2)}$$

حيث أن :

S = معدل فقدان البيانات في كل تجربة .

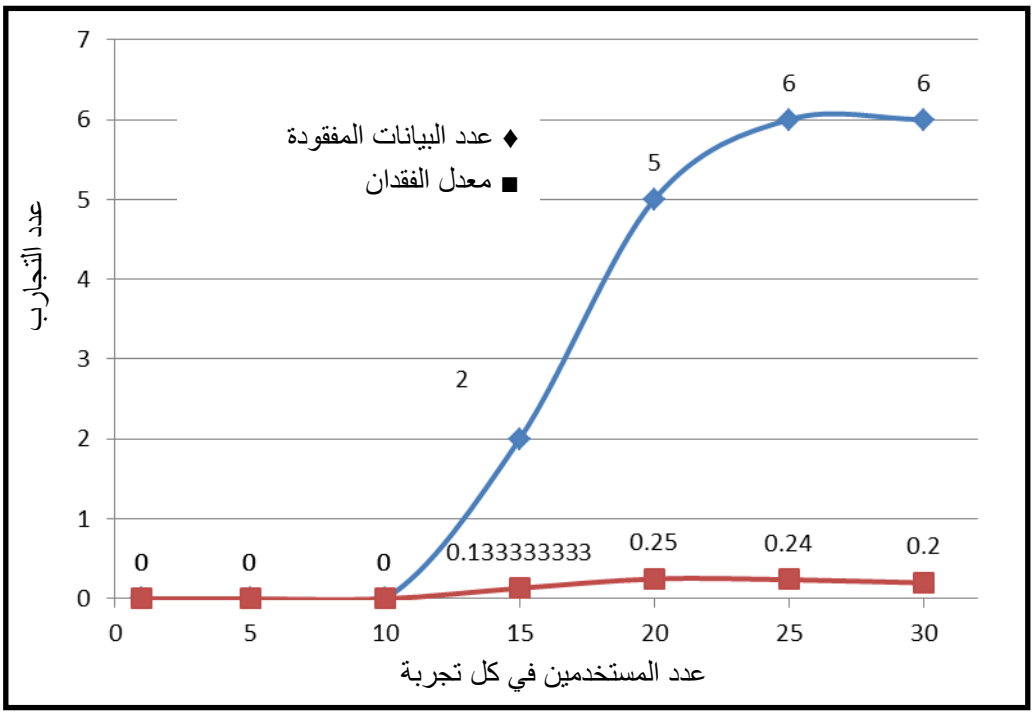
Er. = معدل الخطأ لكل تجربة .

T = زمن الوصول بين الـ Client & Server

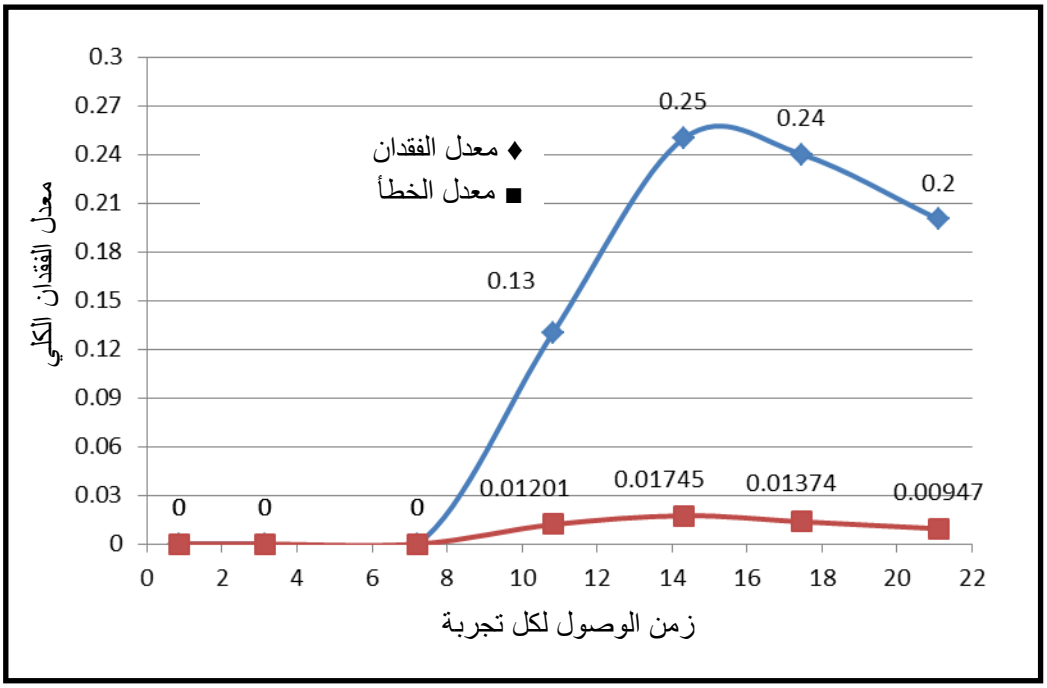
الجدول رقم (2) يوضح نتائج معدل الخطأ لكل تجربة .

التجربة	عدد المستخدمين (N)	زمن الوصول (T)	معدل الفقدان (S)	معدل الخطأ (Er.)
1	1	0.86 ثانية	0	0
2	5	3.16 ثانية	0	0
3	10	7.22 ثانية	0	0
4	15	10.82 ثانية	0.13	0.012
5	20	14.32 ثانية	0.25	0.017
6	25	17.46 ثانية	0.24	0.013
7	30	21.11 ثانية	0.20	0.009

2-4 التحليل : تم تحليل النتائج المستنتجة من المعادلات والمبينة **3-4** بالجداول (1 و 2) أعلاه وكما مبين في الأشكال (12، 13) .



الشكل رقم (10) يوضح نسبة الفقدان للبيانات



الشكل رقم (11) يوضح نسبة الخطأ لكل تجربة

5- التوصيات :

1- استخدام الأنظمة الموزعة في كل المجالات التطبيقية.

2- استخدام تقنية الوقت الحقيقي في بناء الانظمة التطبيقية .

3- العمل على تطوير النظام وجعله يعمل على ادخال البيانات عن طريق اجهزة الهاتف الجوال.

4- استخدام (Real-IP) وتصميم واجهة بلغة(ASP.Net) يمكن استغلال شبكة الانترنيت في عملية التحديث (Update) على قاعدة البيانات (Database) .

المصادر

1- د. حسانة محي الدين ، " قواعد البيانات على الانترنت والإفادة منها "، نادي الإحياء العربي ، مجلة العربية 3000 ، العدد (45) ، 2001.

2- Sanjay Kumar Tiwari, A .K. Sharma, Vishnu Swaroop, "Distributed Real Time Replicated Database Concept And Design" / International Journal Of Engineering Science And Technology (Ijest), Vol. 3 No. 6 June (2011).

3- Ling Liu and Tamer M. Özsu (Eds.) (2009). "Encyclopedia of Database Systems, 4100 p. 60 illus. ISBN 978-0-387-49616-0.

4- John A. Stankovic, Sang Hyuk Son, and Jorgen Hansson "Misconceptions About Real-Time Databases ", IEEE, 0018-9162, June (1999).

5- خضر يوسف ترزي ، " بناء تطبيقات قواعد البيانات العملاقة باستخدام تكنولوجيا مايكروسوفت " ، www.ArabTeam2000.com ، كتاب ، (2001).

6- "Structured Query Language (SQL)". International Business Machines. October 27, 2006. Retrieved 2007-06-10.

7- **عبدالقادر فادي** ، " احتراف برمجة الشبكات والبروتوكولات بواسطة لغة السي شارب" ، عمان–الاردن، (2006).

8- **نهاد محمود** ، مذكرة الشبكات ، www.h4palestine.com .

9- http://www.cb4a.com.

Improving Data Processing Operation in Distributed Databases Using Real Time System

Saadi Hamad Thalej Alluhaibi
Dept of Computer Science. College of Computer & Mathematical Science, Tikrit University, Tikrit, Iraq.
altage_2004@yahoo.com

(**Received:** 6 / 5 / 2012 ---- **Accepted:** 19 / 7 / 2012)

Abstract:
This research deals with distributing data into one schedule then into other ones manipulating the same interface within the same database. In the process, an image is taken from the buffer ,from the binding source in the database by means of giving limited authorizations for each one user to ensure privacy for their own data in the database .Also, this allows chatting and messaging for all the users of this system no matter how many they were.
Multiple-request research system is enabled with a twenty four fields interface benefitting of the Real Time in the process of searching. The user can benefit of any information available for him within the Multiple-request fields.
This system is corresponding software which allows us access to the desired information and for many interlinked users in the same time and the real time.
The system has demonstrated its ability to access to the information required by several users at good speed.