

الآيه

بسم الله الرحمن الرحيم

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ (1) الْإِنْسَانَ مِنْ

عَاقِقٍ (2) رَبُّكَ الْأَعْلَى (3) عَلَّمَ بِالْقَلَمِ (4)

عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ (5)

وَقَلِّدْ رَبِّ زِدْنِي عِلْمًا

صدق الله العظيم

الإهداء

إلي حكمتي وعلمي
إلي أدبي وحلمي
إلي طريقي المستقيم
إلي ينبوع الصبر والتفائل والامل
إلي كل من في الوجود
بعد الله ورسوله
أمي الغاليه

إلي من كلفه الله بالهيبه والوقار الي من علمني العطاء بدون إنتظار إلي
من احمل أسمه بكل افتخار أرجو من الله أن يمد في عمرك لتري ثمارا
قد حان قطفها ... بعد طول إنتظار ... وستبقي كلماتك نجوم اهتدي بها اليوم
وفي الغد والي الابد
والدي العزيز
إلي سندي وقوتي ... وملاذي بعد الله

إلي من اظهرو لي ما هو أجمل من الحياه
أخوتي
إلي كل من أزرني ويتمني لي التوفيق

شكر و عرفان

أشكر الله العلي القدير الذي انعم علي بنعمت العقل والدين القائل
في محكم التنزيل (وفوق كل ذي علم عليم) سور ه يوسف
صدق الله العظيم .

وقال رسول الله صلى الله عليه وسلم: ((من صنع اليكم
معروفا فكافئوه، فإن لم تجدوا ماتكافؤنهبه فادعوا له حتى تروا
أنكم كافأتموه))... (رواه أبو داود)
وأثنى ثناء حسنا علي

الى

أخواني واخواتي الى من سرنا سوياً و نحن نشق الطريق معاً
نحو النجاح والابداع الى اصدقائي و صديقاتي الى من صاغوا
لنا من علمهم حروفاً ومن فكرهم منارة تنير لنا سيرة العلم و

النجاح

الى

أساتذتنا الكرام.

المستخلص

مع تصاعد استخدام قواعد البيانات في الانظمة، و زيادة كمية البيانات المخزنة فيها نسبة لترابط الانظمة
مع بعضها البعض عن طريق شبكات الحاسوب، و زيادة اعتمادية الانظمة عليها قد يتسبب التأخير في الوصول
الي تلك البيانات، أو فقدان البيانات في خسائر مادية كبيرة؛ تزايدت الحاجة الى برمجيات تساعد على المحافظة
على سلامة تلك البيانات، أو تسهيل، أو تسريع الوصول اليها؛ يهتم هذا البحث بإنشاء اداة تساعد مستخدمي
نظام تخطيط موارد المؤسسة أودو في استنساخ بيانات الانظمة الفرعية من نظام إلى نظام اخر ، آخذاً في
الاعتبار الترابط المنطقي للبيانات مع النظم الفرعية، علماً بأن الادوات الحالية تتعامل مع نظام ادارة قواعد
البيانات مباشرة متجاهلة القيود المفروضة على تلك البيانات بواسطة الانظمة الفرعية، وبعد إنشاء أداة
الاستنساخ تم زيادة إتاحة الانظمة الفرعية و خفض الضغط علي الشبكة نسبة لامكانية الانظمة الفرعية
المواصلة في العمل دون الاتصال بالخادم المركزي .

Abstract

With the increasing use of databases in the systems, and the increase in the amount of data stored in them in relation to the interconnection of systems with each other through computer networks, And the increasing reliability of systems on them , Any delays to access those data or loss of data may cause large financial losses,The need for a software that helps to maintain the integrity of such data, Facilitate or speed up access to it has increased, This research is concerned with the creation of a tool that helps users of the enterprise resource planning system Odoo to replicate the data of subsystems (modules) from one system to another, Taking into account the logical interconnection of data with subsystems, Knowing thatthe current tools deal with the database management system directly, Ignoring the restrictions on those data by subsystems.After the creation of the replication tool, The availability of sub-systems was increased and the pressure on the network was reduced in

relation to the possibility that subsystems can continue to work without connecting to the central server.

جدول المصطلحات

المصطلح	شرح المصطلح	شرح المصطلح
ERP	Enterprise Resource planning	تخطيط مواد المؤسسة
UML	Unified Modeling Language	لغة النمذجة الموحدة
SQL	Structured Query Language	لغة الاستعلامات البنوية
DBMS	Database management system	نظام إدارة قواعد البيانات
ODOORPC	Odoo Remote Procedure Call	نداءات الدوال البعيدة للاودو
SSL	Secure Sockets Layer	طبقة النقل المؤمنة
JSON	JavaScript Object Notation	طريقة لوصف البيانات

طريقة لوصف البيانات	Extensible Markup Language	XML
شبكات النطاق الواسع	Wide Area Network	WAN
قواعد البيانات الموزعة	Distributed Database	DDB
مخطط نمذجة عمليات الاعمال	Business Process Modeling Notation	PBMN

جدول الاشكال

رقم الصفحة	الشكل	الرقم
10	الشكل (3.1) يوضح المنهجية الخاصة بالنظام	1
14	الشكل (3.2) يوضح مخطط عمليات النظام الرئيسية	2
15	الشكل (3.3) يوضح حالات الاستخدام للنظام	3
16	الشكل (3.4) يوضح تسلسل بداية عملية الاستنساخ	4
17	الشكل (3.5) يوضح عملية الاتصال بين خادمين	5
18	الشكل (3.6) يوضح عملية قراءة البيانات من خوادم ال Odoo	6
19	الشكل (3.7) يوضح عملية الكتابة للبيانات	7
20	الشكل (3.8) يوضح الفئات التي بني عليها النظام	8
21	الشكل (4.1) يوضح كيفية إضافة معلومات الخادم	9
22	الشكل (4.2) يوضح قائمة بأسماء الخوادم الموجودة في النظام	10
23	الشكل (4.3) يوضح كيفية إضافة معلومات عملية الاستنساخ	11
24	الشكل (4.4) يوضح كيفية إضافة نظام فرعي لعملية الاستنساخ	12

25	الشكل (4.5) يوضح كيفية جدولة عملية الاستساخ	13
26	الشكل (4.6) يوضح عملية اختيار النظام الفرعي (Modules)	14
27	الشكل (4.7) يوضح عملية إضافة النظام الفرعي (Module)	15
28	الشكل (4.8) يوضح عملية اختيار الجدول (Model)	16
29	الشكل (4.9) يوضح عملية إضافة شرط لجدول (Model) .	17
30	الشكل (4.10) يوضح التفاصيل الخاصة لكل جدول (Model)	18

جدول الجداول

رقم الصفحة	الجدول	الرقم
9	الجدول (2.1) جدول يوضح اوجه الشبه والاختلاف بين النظام المقترح والدراسات السابقة	1
10	الجدول (2.2) جدول يوضح اوجه الشبه والاختلاف بين النظام المقترح والدراسات السابقة	2
34	جدول المراجع	3

الفهرس : -

أ الاية
ب الاهداء
ج الشكر والعرفان
د المستخلص
هـABSTRACT
و جدول المصطلحات
ز جدول الاشكال
ح جدول الجداول
ط الفهرس
 1. الباب الاول : المقدمة
1 1.1 تمهيد
1 1.1.1 قواعد البيانات
1 1.1.2 قواعد البيانات الموزعة
2 1.1.3 مفهوم الاستساخ

2	1.2 مشكلة البحث
3	1.3 اهداف البحث
3	1.4 أهمية البحث
3	1.5 فرضيات البحث
4	1.6 حدود البحث
4	1.7 هيكلية البحث
	2. الباب الثاني : الخلفية النظرية والدراسات السابقة
5	2.1 المقدمة
5	2.2 الخلفية النظرية
6	2.3 الدراسات السابقة
6	2.3.1 بوكاردو (Bucardo)
6	2.3.2 PgCluster
7	2.3.3 Pgpool II
7	2.3.4 Slony
	3. الباب الثالث : منهجية وتحليل النظام
10	3.1 المقدمة
10	3.2 المنهجية
10	3.2.1 مراحل تطوير النظام
14	3.3 تحليل النظام
14	3.3.1 نموذج مخطط عمليات الاعمال (BPMN)
15	3.3.2 مخطط حالات الاستخدام (Use Case Diagram)
16	3.3.3 مخططات التتابع (Sequence Diagram)
20	3.3.4 مخطط الفئات (Class Diagram)
	4. الباب الرابع : اختبار النظام
21	4.1 المقدمة
21	4.2 مهام النظام الرئيسية
21	4.2.1 إدارة معلومات الخادم
23	4.2.2 إدارة عمليات الاستنساخ
26	4.2.3 إدارة معلومات الانظمة الفرعية

31 النتائج
32 التوصيات
33 الخاتمة
34 المراجع

الباب الاول

المقدمة

1.1 تمهيد

هذا الفصل يتناول تعريفاً بسيطاً عن قواعد البيانات وقواعد البيانات الموزعة ومفهوم الاستنساخ لها ، كما يتناول مشكلة البحث والاهداف .

• قواعد البيانات

تُعرف قاعدة البيانات (Database) بأنها آلية منظمة لتخزين وإدارة واسترجاع المعلومات . حيث تمكن المستخدمين من البحث فيها، أو مقارنتها، أو الرجوع إليها، أو تغييرها بسرعة مثالية، ويتم إنشاؤها وصيانتها باستخدام لغات برمجة خاصة بها، ومن أكثر اللغات شيوعاً واستخداماً في قواعد البيانات لغة SQL [1].

وعادة ما تكون البيانات المخزنة في قواعد البيانات كبيرة الحجم ومخزنة في قاعدة بيانات مركزية يتم الوصول اليها عن طريق شبكات الحاسوب ونجد ان قواعد البيانات المركزية تصبح غير عملية في حالة إزديات ضخامة البيانات مما أدى إلي إتجاه مطوري الانظمة الي إستخدام قواعد البيانات الموزعة [1].

• قواعد البيانات الموزعة (DDB)

تُعرف بأنها قواعد بيانات متعددة ، مترابطة مع بعضها البعض ترابطاً منطقياً و موزعة توزيعاً جغرافياً على عدة اجهزة حاسوب وتكون مرتبطة بواسطة الشبكة محلية او شبكة الانترنت .

تساهم قواعد البيانات الموزعة في تحسين أداء قاعدة البيانات ، ويمكن ان تكون في عدة مواقع عمل متناثرة او في نفس الموقع . ويمكن ان يكون التقسيم بالنسبة لقاعدة البيانات الواحدة او لعدة قواعد بيانات في حالة كان التقسيم لقاعدة بيانات واحدة يمكن ان يكون تقسيم افقي (Horizontal Fragmentation) او تقسم رأسي (Vertical Fragmentation) او خليطاً من النوعين Mixed (Hybrid) Fragmentation واذا كانت هذه التقسيمات مخزنة في أكثر من موقع يقال عليها انها مستنسخة (replicated) [1] .

• مفهوم الاستنساخ (Replication)

هو تخزين نسخة من قواعد البيانات في اكثر من موقع ، في الحالات القصوى للاستنساخ يتم نسخ قاعدة البيانات بالكامل لكل موقع ، مما يخلق بيئة استنساخ متكاملة (fully replicated distributed database) مما يزيد الاتاحية بصورة ملحوظة نسبة لامكانية كل موقع العمل باستقلال عن بقية المواقع ، زيادة إلي ذلك يقل استخدام الشبكة [2].

في اغلب الحالات لا يتم نسخ قاعدة البيانات بالكامل نسبة لان ذلك يستهلك كمية كبيرة من موارد التخزين و إنما يتم الإستنساخ الجزئي لها (partial replication) حيث يكون جزء فقط من قواعد البيانات مخزن في اكثر من موقع وقد يكون هذا الجزء منسوخ مرة واحدة او اكثر .

1.2 مشكلة البحث

تتمثل مشاكل البحث في الاتي :-

- إعتداد الانظمة الطرفية علي النظام المركزي لقواعد البيانات (Centralized Database) للحصول على المعلومات و البيانات اللازمة و في حالة حدوث عطل في النظام المركزي تتوقف بقية الانظمة الطرفية عن العمل .
 - بطء نقل البيانات التي يتم طلبها من والى النظام المركزي .
 - زيادة الضغط على الشبكة .
 - انخفاض توافر البيانات في النظام.
- هنا نظرًا لأن العديد من المستخدمين يدخلون إلى الخادم نفسه ، فقد يؤدي ذلك إلى إبطاء تنفيذ طلبات البحث.
- الكوارث الطبيعية التي تؤدي الى اتلاف التبيانات.

1.3 اهداف البحث

تتلخص اهداف البحث بالاضافة الى حل المشاكل المذكوره اعلاه كالاتي :-

- تنفيذ مفهوم الاستنساخ (Replication) لقواعد بيانات ال (postgresql) في الانظمة الفرعية .
- تخفيف الضغط على النظام المركزي .
- تقليل كمية استخدام بيانات الشبكة .
- تحسين الاداء .
- تحسين الاتاحية .

1.4 اهمية البحث

تتمثل اهمية البحث في إنشاء أداة تقوم بتطبيق مفهوم الاستنساخ وجزء من مفاهيم قواعد البيانات

الموزعة اخذة في الاعتبار عمل ذلك لبيانات الأنظمة الفرعية (modules) لنظام الاودو. وان الادوات

المستخدمة في استنساخ البيانات تتعامل مع قواعد البيانات مباشرة دون أخذ علاقتها بالانظمة الفرعية.

1.5 فرضيات البحث

- الفرضيه الاولى :
تصميم (Odoo modules)يقوم بعملية الاستنساخ (Replication)
- الفرضيه الثانيه :
وجود نظام اودو مركزي وانظمة طرفيه .

• الفرضية الثالثة :

وجود انظمة اودو طرفيه لاتحتوي على بيانات سابقه و يقوم النظام بإستتساخ البيانات من النظام المركزي الى الانظمه الطرفيه .

• الفرضية الرابعة:

تطابق نسخ (software version) الانظمة الفرعيه مع النظام المركزي على مستوى الاودو وعلى مستوى الانظمه الفرعيه له .

1.6 حدود البحث

تطبيق مفهوم الاستتساخ (Replication) غير المتزامن على نظام ادارة قواعد البيانات (PostgreSQL) الذي يخزن بيانات انظمة فرعية بنظام الاودو 11 .

1.7 هيكلية البحث

يتكون هذا البحث من خمس فصول مرتبة كالآتي :-

. الفصل الأول : المقدمة وتحتوي على مشاكل البحث ،اهداف البحث اهمية البحث ،فرضيات البحث، مايتضمنه البحث، وحدود البحث .

. الفصل الثاني: ويحتوي على نبذة عن الدراسات السابقة و المستخدمة حالياً في عمليات الاستتساخ .

. الفصل الثالث: يحتوي على مرحلة جمع البيانات ، تحليل وتصميم النظام والادوات المستخدمة .

. الفصل الرابع : يحتوي على النتائج وشاشات التنفيذ والتوصيات.

الباب الثاني

الخلفية النظرية و الدراسات السابقة

2.1 المقدمة

يتناول هذا الباب الخلفية النظرية عن النظام المقترح و جزء من الدراسات السابقة التي تم فيها استخدام مفهوم الاستنساخ ووجهه التشابه والاختلاف بين كل الدراسات والنظام المقترح .

2.2 الخلفية النظرية

يهتم هذا البحث بتطبيق مفهوم استنساخ البيانات (Data Replication) مع نظام تخطيط موارد المؤسسة الاودو (Odoo ERP) على وجه الخصوص في الانظمة الفرعية له (modules). ويهدف البحث لمعالجة بعض المشاكل التي يواجهها النظام او التقليل منها بقدر الامكان .

حيث سيتم استنساخ جزء من قواعد البيانات الاودو (Odoo Postgresql) التي يحتاجها كل نظام فرعي (modules) الى مكان عمله وفي نهاية كل فترة عمل تتم مزامنة البيانات مع النظام المركزي مما يقلل الوقت المستهلك في نقل البيانات غير المكتملة بين النظام الرئيسي والانظمة الفرعية كما يساعد في زيادة اتاحية البيانات للانظمة الفرعية دون الرجوع الي النظام الرئيسي ، مما يؤدي الي تقليل الضغط على الشبكة في ذلك الوقت و تقليل الضغط على النظام الرئيسي وبتحسن الاداء في الانظمة الفرعية .

يمكن تقسيم الاستنساخ علي حسب الخادم المشارك في عملية الاستنساخ الي متحكم مرسل او مستقبل للبيانات (Master OR Slave) ، تم علي حسب الزمن الي متزامن او غير متزامن (Synchronous OR Asynchronous) او علي حسب مفاهيم قواعد البيانات الي

:

2.2.1 النسخ المتماثل للمعاملات (Transactional Replication) [2]

وفيه يراقب عامل الاستنساخ (replication agent) للتغييرات في قاعدة البيانات وينقل تلك التغييرات إلى خوادم النسخ الاحتياطي الأخرى. يمكن أن يحدث هذا الإرسال على الفور أو على أساس دوري. يستخدم النسخ المتماثل للمعاملات في حالات الخادم للخادم (server - server scenarios) [2].

2.2.2 النسخ المتماثل للقطات (Snapshot Replication)

يتم اخذ لقطة من البيانات من خادم واحد ونقلها إلى خادم آخر أو قاعدة بيانات أخرى على نفس الخادم بعد المزامنة الأولية، يمكن للنسخ المتماثل ل snapshot تحديث البيانات في الجداول المنشوره بشكل دوري على الرغم من أن النسخ المتماثل للصور هو اسهل شكل من أشكال النسخ المتماثل، إلا أنه يتطلب نسخ جميع عناصر البيانات في كل مرة يتم فيها تحديث جدول [2].

2.2.3 دمج النسخ المتماثلة (Merge Replication)

يسمح للنسخ المتماثلة بالعمل بشكل مستقل ويمكن للكيانين العمل دون اتصال. عندما يكون الاتصال متاحاً يتحقق عامل النسخ المتماثل من التغييرات على كل مجموعات البيانات و تعديل قاعدة البيانات وفقاً لذلك. في حالة حدوث تعارض في المعاملة ، فإنه يستخدم خوارزمية تدقيق تعارض محددة مسبقاً لتحقيق التناسق ، ويتم استخدام دمج النسخ المتماثل في الغالب في البيئات اللاسلكية.

حيث ان هذا البحث يستخدم الإستنساخ الجزئي لقواعد البيانات

2.3 لدر اسات السابقة:

2.3.1 بوكاردو (Bucardo)

هو نظام إستتساخ خاص بنظام إدارة قواعد البيانات (PostgreSQL) يتيح نسخ كل او بعض من قواعد البيانات ، و هو نظام غير مترامن ، تم تطويره بواسطة جون جينسينوفريج ساينيو لشركة End Point Corporation، وهو الآن قيد الاستخدام في العديد من المنظمات . و هو برنامج مجاني مفتوح المصدر تم إصداره بموجب ترخيص [3] BSD.

PgCluster 2.3.2

هي أداة إستتساخ متعددة ومتزامنة تعمل على PostgreSQL مبنية على بنية الانفصال التام (Shard Nothing)، تم تطويرها بواسطة Atsushi Mitanni وفريق مطور PGCluster. وتم كتابتها بلغة ال C مرخصة بموجب رخصة BSD [4].

Pgpool II 2.3.3

هي أداة إستتساخ مكتوبة بلغة C ومبنية على (PgCluster) ، تعمل كوسيط بين خوادم ال postgresql وعملاء ال postgresql . لا تتطلب تعديلات postgresql أساسية وتوفر للمستخدم العديد من الميزات منها : [5]

- موازنة أحمال الشبكة (connection Pooling) عن طريق توزيع الطلبات على الخوادم المتوفرة .
- إستتساخ البيانات بين عدة خوادم (postgresql)
- التنفيذ المتوازي للاستعلامات .

Slony 2.3.4

هو نظام إستنساخ يعمل على نسخ متتالي للبيانات علي خادم (**postgresql**) واحد
وعدة عملاء . تمت كتابتها بلغة **C** وهي مصممة للاستخدام في مراكز البيانات وأماكن النسخ
الاحتياطي ، حيث يكون توافر العملاء شئ مسلم به في هذه البيئات . [6]

الجدول (2.1) يوضح اوجه الشبه والاختلاف بين النظام المقترح والدراسات السابقة

اسم الدراسة	قابلية التوسع	قابلية التوسع للقراءة	قابلية التوسع للكتابة	قابلية التنفيذ المنفصل / WAN	الدوال / الاجراءات	لغة التطوير	التنفيذ المتوازي للاستعلامات	التوافر العالي	تطوير ال PostgeSql	التعديل على نواة ال PostgreSql
Bucardo	نعم	نعم	نعم	نعم	نعم	C	نعم	ليسة تلقائية	نعم	لا
Pgpool II	نعم	نعم	لا	لا	نعم	C	نعم	نعم	لا	لا
PgCluster	نعم	بعض	نعم	لا	نعم	C	لا	نعم	لا	نعم
slony	نعم	نعم	لا	نعم	نعم	C	لا	نعم	نعم	لا

وضح اوجه الشبه والاختلاف بين النظام المقترح والدراسات السابقة

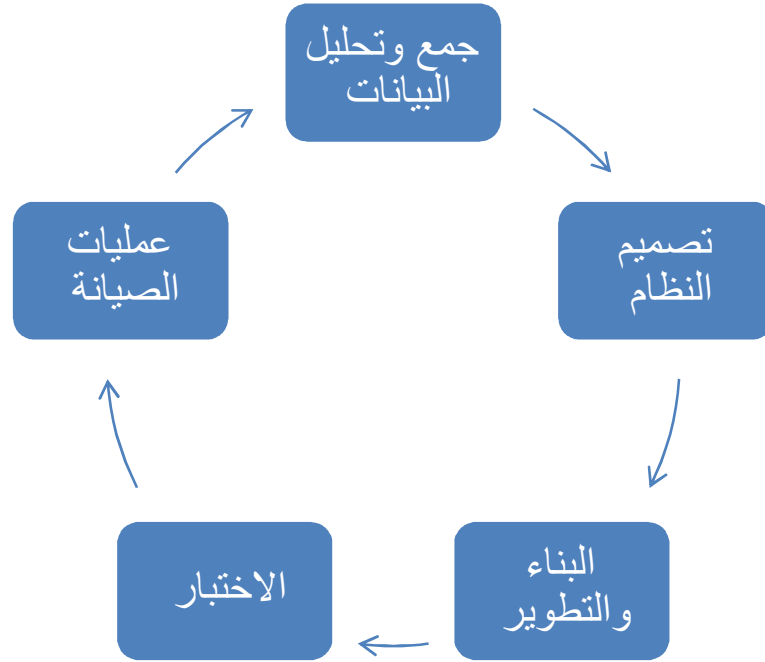
Program	License	Maturity	Replication Method	Sync	Connection Pooling	Load Balancing
PgCluster	BSD	Not production ready	Master-Master	Synchronous	No	Yes
Pgpool-II	BSD	Recent release	Statement-Based Middleware	Synchronous	Yes	Yes
slony	BSD	Stable	Primary-Replica	Asynchronous	No	No
Bucardo	BSD	Stable	Master-Master, Primary-Replica	Asynchronous	No	No

3.1 مقدمة

في هذا الباب سوف نتناول منهجية النظام المقترح والتحليل الخاص به ونبذة مبسطة عن كل أداة مستخدمة في عملية التحليل والتصميم .

3.2 المنهجية

تتلخص منهجية النظام في الشكل (3.1) من جمع البيانات وتحليلها ، تصميم النظام ، بناء النظام وتطويره ، عملية الاختبار و عمليات الصيانة .



الشكل (3.1) يوضح المنهجية الخاصة بالنظام .

3.2.1 مراحل تطوير النظام :

3.2.1.1 مرحلة جمع البيانات

في هذه المرحلة تم تعريف المتطلبات التي يجب أن يستوفيهما النظام وذلك عن طريق إستخدام كل من :

- المقابلات وذلك بمقابلة مجموعة من الاساتذة المختصين في هذا المجال
- الاطلاع على الدراسات السابقة .

3.2.1.2 مرحلة تصميم النظام :

في هذه المرحلة تم إعداد تصميم النظام بناءً على البيانات التي تم جمعها ، و تعريف هيكلية النظام و اجزاؤه الأساسية وتم ذلك بإستخدام نموذج مخططات عمليات الاعمال (Business Process Modeling Notation) . و ال (UML) في عملية تحليل وتصميم النظام .

1. نموذج مخططات عمليات الاعمال (PBMN) :-

هي لغة نمذجة غير قياسية تستخدم لاغراض تحليل وتبسيط عمليات الاعمال (Business Process) وتستخدم الرسومات والاسهم لوصف المهام الخاصة بالنظام .

2. مخططات لغة النمذجة الموحدة (UML)

هي لغة نمذجة قياسية موحدة متعددة الأغراض، وهي من أنجح التطبيقات الهندسية في نمذجة النظم الضخمة و البرامج كائنية التوجه (Object Oriented) (وتستخدم لغة النمذجة الموحدة الرسومات لوصف العناصر الأساسية للأنظمة وسير العمليات التي تتم في النظام [13].

3.2.1.3 مرحلة بناء النظام :

تم بناء النظام بناءً علي التقنيات التالية :

1. Python

بايثون هي لغة برمجة مفتوحة المصدر شائعة تستخدم لكل من البرامج القائمة بذاتها والبرمجة في مجموعة واسعة من المجالات .إنه حر ومحمول وسهل الإرهاق ، وهو سهل الاستخدام نسبيا وممتاز بشكل ملحوظ .وقد وجد المبرمجون من كل ركن من أركان صناعة البرمجيات أن تركيز بايثون على إنتاجية المطورين وجودة البرمجيات يعد ميزة استراتيجية في المشاريع الكبيرة والصغيرة على السواء.كنت جديداً في البرمجة أو كنت مطوراً محترفاً ، تم تصميم هذا الكتاب ليوفر لك السرعة في لغة Python بطرق لا يمكن اتباعها في طرق محدودة بعد قراءة هذا الكتاب ، يجب أن تعرف ما يكفي عن Python لتطبيقه في أي مجال من مجالات التطبيقات التي تختار استكشافها [7] .

2. XML

هي لغة تستخدم لوصف البيانات و نقلها عبر الإنترنت، و هي تستخدم الوسوم (Tags) (كما تعتبر اساسا Semantic web - RDF - Resource Description للويب الدلالي او هيكل وصف الموارد [8].Framework

3. Postgresql

هو نظام إدارة قواعد البيانات علائقي DBMS يعتمد التعامل معه على لغة SQL يعمل على منصات متعددة من مثل أنظمة التشغيل المستندة إلى Unix مثل "IBM AIX" و "HP-UX" وغيرها من منصات Linux كذلك يعمل على أنظمة Windows و Mac OS X [9].

4. Odo ERP :

أودو (Odo) والمعروف سابقاً بنظام (Openerp) والذي كان اسمه من قبل (TinyERP) هو أحد الأنظمة التطبيقية لإدارة موارد مخططات المنشآت، وهو نظام مجاني ومفتوح المصدر أيضاً تحت رخصة (AGPL) إلا جزء بسيط مرخص تحت (MPL) وبالتالي فيمكن التعديل في النظام دون أي رسوم إضافية إذا تم نشرها بنفس الرخصة، وقد تم إنشائه من قبل شركة (Openerp S.A[10]).

5. OdoRPC :

هو وحدة بايثون (Python) توفر طريقه سهلة لتجريب خوادم أودو من خلال RPC وسوف يتم التحدث عن مميزاتا بالتفصيل لان معظم العمل مبني عليها [11].

الميزات المدعومة :

- الوصول إلى جميع طرق نماذج البيانات (حتى التصفح) مع واجهة برمجة تطبيقات مشابهة لواجهة برمجية التطبيقات من جانب الخادم.
- استخدام المعلمات المسماة (named parameters) مع اساليب النموذج.
- أرسل المستخدم سياق تقديم الدعم لتدوي.
- تصفح السجلات.
- إدارة قواعد البيانات.
- تنزيل التقارير.
- بروتوكول ال JSON-RPC (يدعم SSL).

ثم تم تصميم نظام فرعي (Odo Module) مبني علي التقنيات السابقة الذكر

3.2.1.4 مرحلة الاختبار :

في هذه المرحلة تم اختبار النظام باستخدام حيث انه تم نقل البيانات من النظام الرئيسي الي النظام الفرعي وفي زمن معين يتم إستنساخ البيانات التي تمت إضافتها في النظام الفرعي مع النظام الرئيسي وللتأكد من ان البيانات التي يتم إستنساخها مع النظام الرئيسي قد وصلت يتم مراقبتها عن طريق مراقب ومحلل الشبكة (Wireshark) في كلا الطرفين .

أو إيرشارك (Wireshark):

هو محلل حزم الشبكة الذي يقوم بمحاولة إتقاط حزم الشبكة ومحاولة عرض بيانات الحزمة بالتفصيل قدر الامكان.

ويمكن القول انه جهاز قياس يستخدم لفحص ما يحدث داخل كبل الشبكة، وهو احد افضل أجهزة تحليل حزم المصدر المفتوح المتاحة [12].

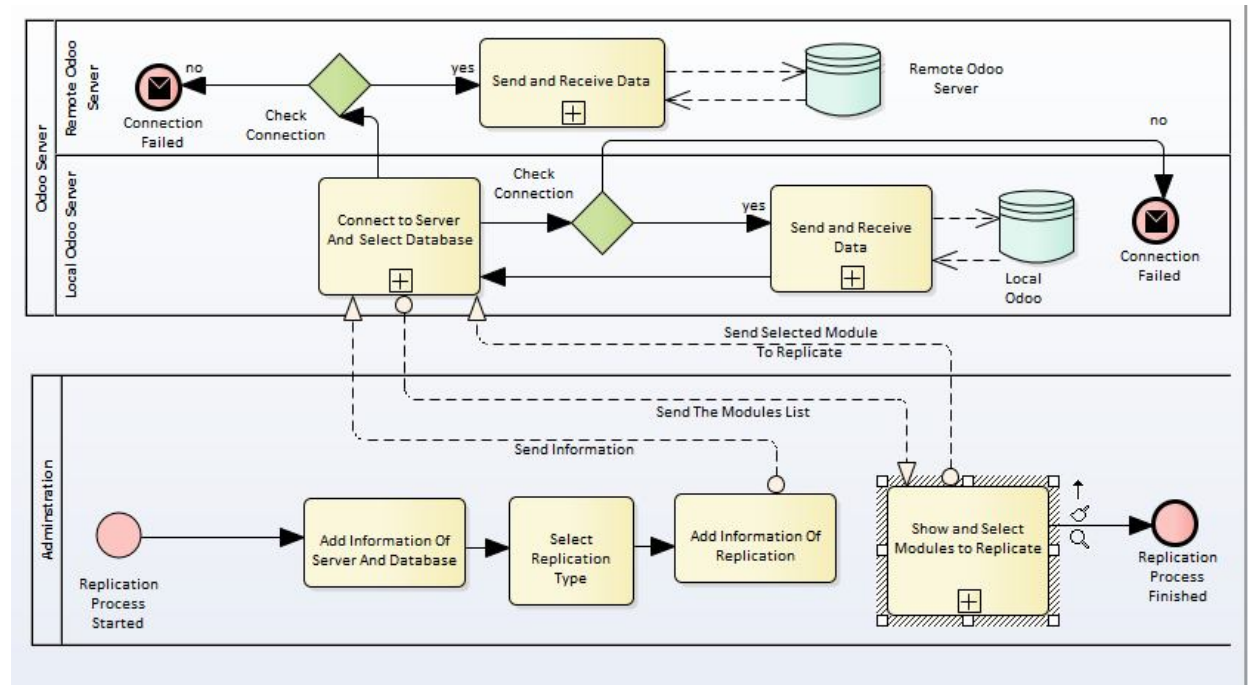
3.2.1.5 مرحلة عمليات الصيانة :

في هذه المرحلة وبعد عملية الاختبار قمنا بمراجعة البيانات التي لم يتم تزامنها ومعرفة اسباب هذا الخطأ ومعالجة هذه الاخطاء

3.3 تحليل النظام

3.3.1 نموذج مخططات عمليات الاعمال (PBMN)

الشكل (3.2) يوضح سير عمل العمليات الرئيسية في النظام من البداية الي النهاية ، حيث تبدأ عملية الاستنساخ (Replication Process Started) عن طريق مدير النظام او عن طريق جدولة المهام بعد ان يقوم مدير النظام بتحديد الزمن الذي تبدأ فيه ، ثم يقوم مدير النظام بإضافة معلومات الاتصال للخادم الذي يتم إستنساخ البيانات معه (Add Information Of Server And Database) ، ثم يقوم بإختيار نوع الاستنساخ والذي يحدد فيه مصدر جلب البيانات واتجاه البيانات (Select Replication Type) ، ثم يقوم بإضافة معلومات عن الاستنساخ حيث يحدد قاعدة البيانات المستهدفة واعتماديات المستخدم (Add Information Of Replication) ومن ثم يقوم البرنامج بالاتصال بالخادم البعيد (Remote Odoo Server) للتأكد من صحة البيانات المدخلة و اعتماديات المستخدم واختيار قاعدة البيانات (Connect to Server And Select Database) ، ومن ثم يقوم المدير بإختيار الانظمة الفرعية (Odoo Modules) المراد إستنساخ البيانات لها (Show and Select Modules to Replicate) ، ومن ثم تبدأ عملية إرسال وإستقبال البيانات (Send and Receive Data) ، وفي حالة حدوث اي خطأ في إدخال البيانات لا يتم الاتصال ويتم الخروج من النظام (Connection Failed) .

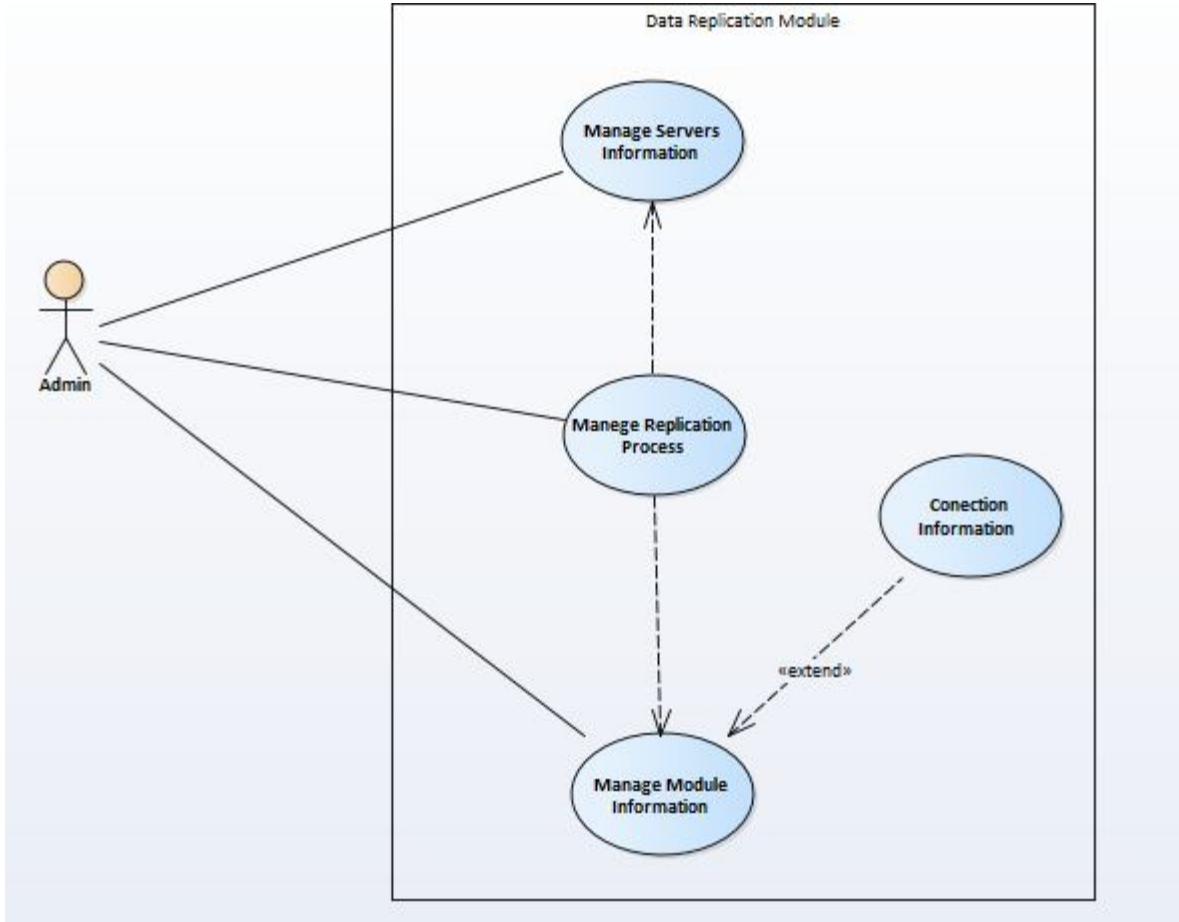


الشكل (3.2) مخطط عمليات النظام الرئيسية

3.3.2 مخطط حالات الاستخدام (Use Case Diagram) :

يقوم النظام بثلاثة وظائف أساسية :-

- إدارة معلومات الخوادم (Manage Servers Information)
- إدارة عمليات الاستنساخ (Manage Replication Process)
- إدارة معلومات الانظمة الفرعية (Manage Module Information)

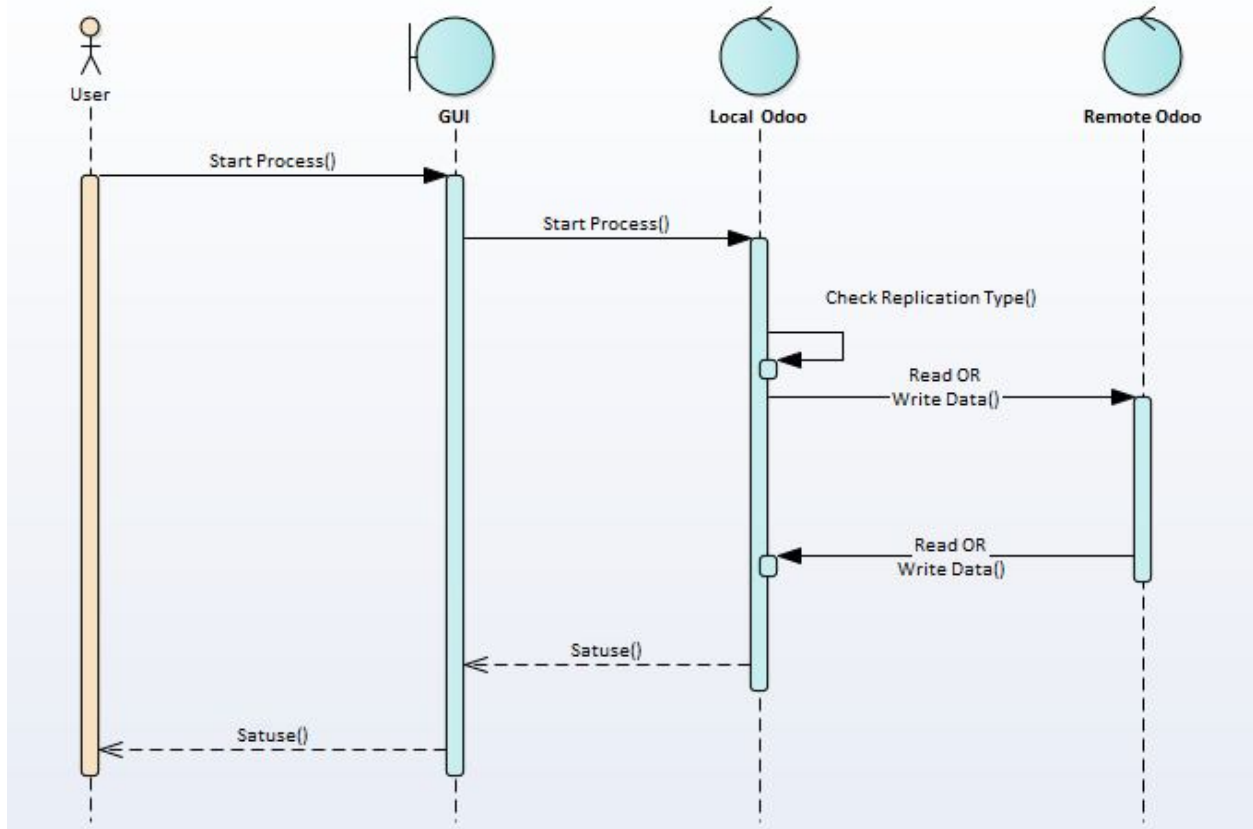


الشكل (3.3) مخطط حالات الاستخدام للنظام

3.3.3 مخططات التتابع (Sequence Diagram) :

3.3.3.1 بداية عملية الاستنساخ :

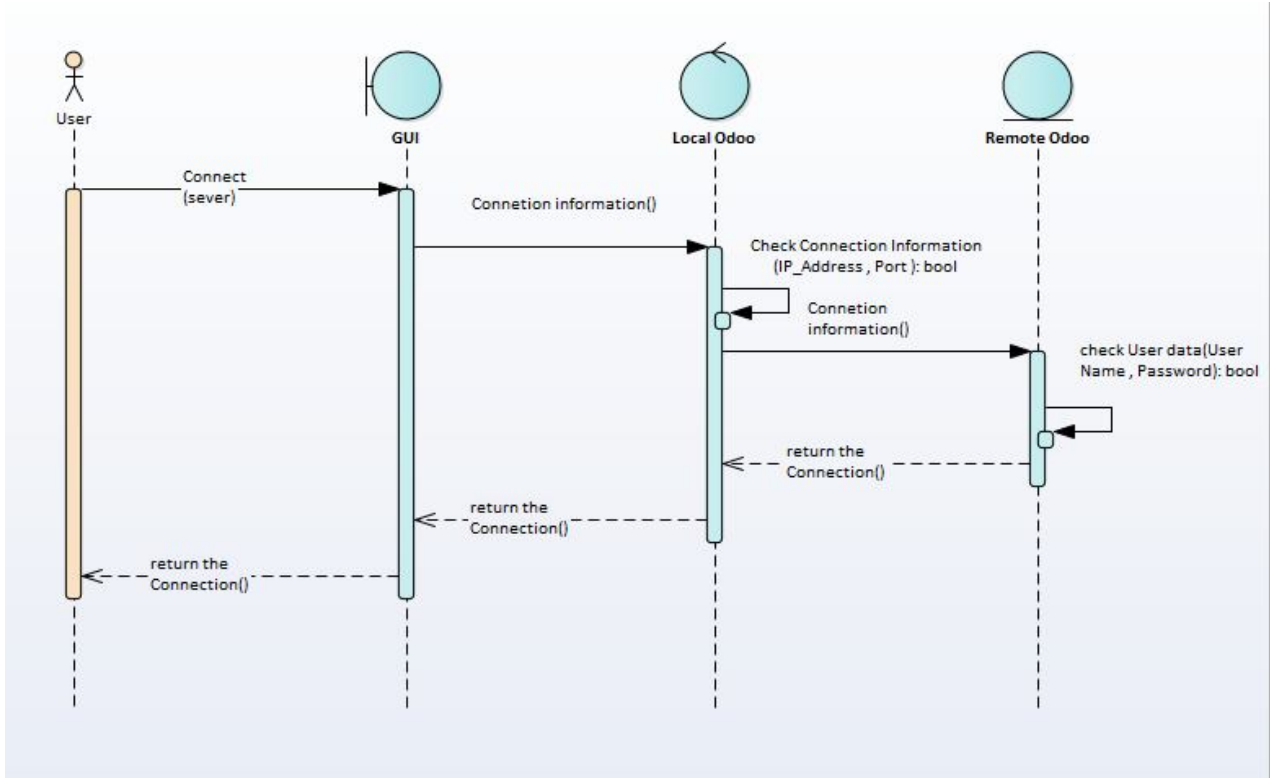
تبدأ عملية الاستنساخ عن طريق مدير النظام او عن طريق جدولة المهام بعد ان يقوم مدير النظام بتحديد الزمن الذي تبدأ فيه ، ثم يقوم بإختيار نوع الاستنساخ (Select Replication Type) ، ومن ثم يقوم البرنامج بالاتصال بالخادم البعيد ، ومن ثم تبدأ عملية قراءة وكتابة البيانات اعتماداً على نوع الاستنساخ (Read and Write Data)



الشكل (3.4) : يوضح تسلسل بداية عملية الاستنساخ

3.3.3.2 عملية الاتصال (Connection) :

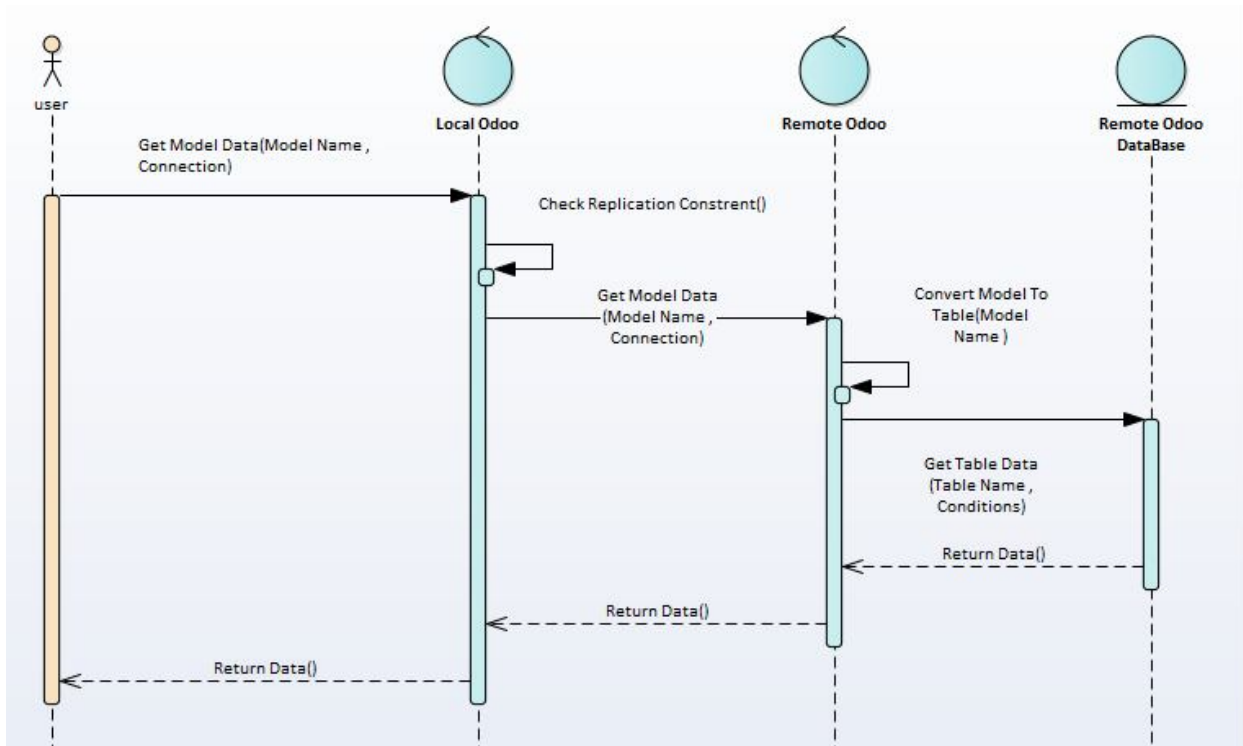
يتم الاتصال على مرحلتين أولاً الاتصال بخادم الودو (Odoo Server) والتأكد من صحة البيانات الخاصة بالاتصال (Check Connection Information) وبعد عملية الاتصال يتم تحديد قاعدة البيانات المستهدفة ثانياً و عن طريق خادم الودو يتم الاتصال بقاعدة البيانات والتحقق من إتماديات المستخدم (Check User Name And Password)، وفي حالة حدوث اي خطأ لا يتم الاتصال ويتم الخروج من النظام .



الشكل (3.5) : يوضح عملية الاتصال بين خادمين

3.3.3.3 عملية القراءة :

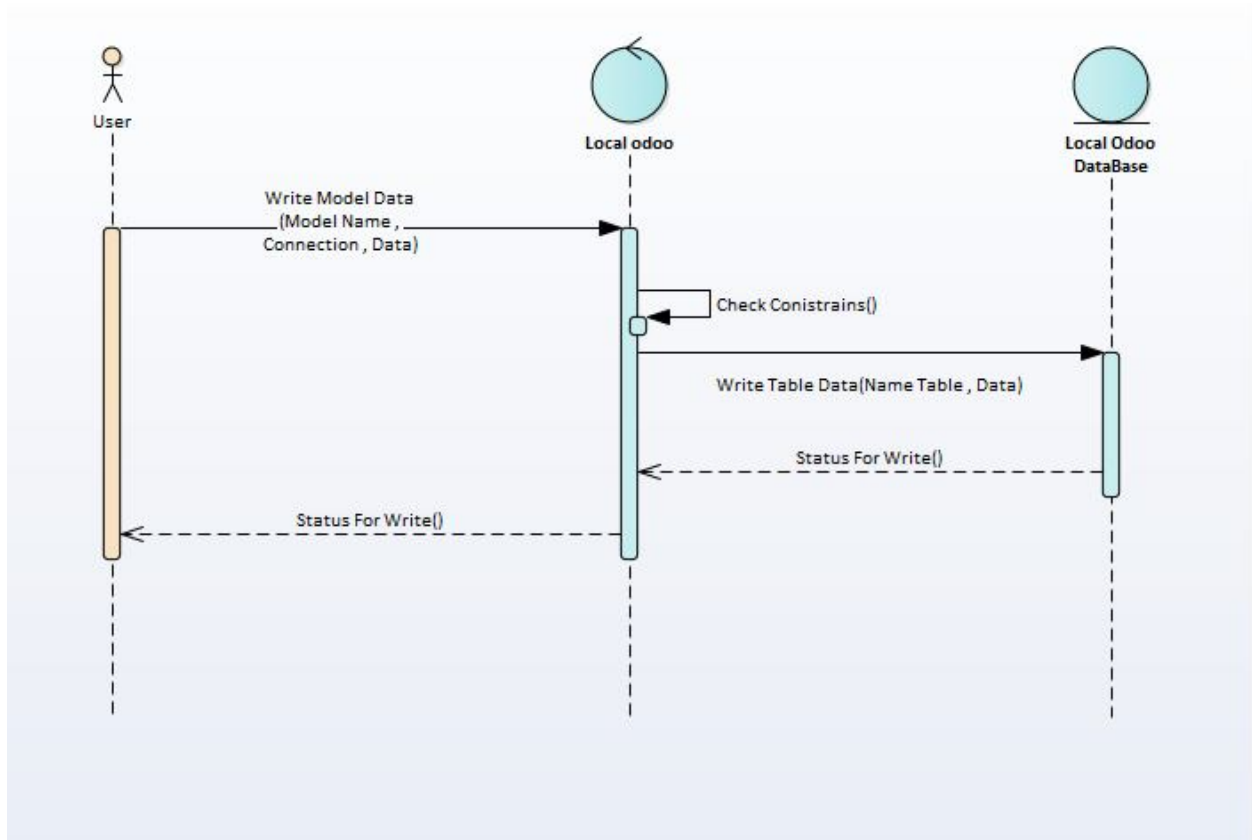
تتم هذه العملية بعد عملية الاتصال وفيها يتم جلب البيانات المراد إستسآخها من قاعدة بيانات خادم اودو الي خادم اودو اخر وتبدأ هذه العملية بتحديد الجداول الخاصة بالنظام الفرعي المراد جلب البيانات منها ، ثم يتم التحقق من القيود اللازمة لعملية القراءة مثل (تاريخ تعديل او اضافة البيانات المراد قراءتها) . ثم يقوم خادم الاودو بترجمة القيود وتحويلها الي إستعلامات (Sql Query) وارسالها الي نظام ادارة قواعد البيانات ، الذي بدوره يقوم بإرجاع البيانات التي تم طلبها ثم يقوم خادم الاودو بتحويل تلك البيانات الي صورة مناسبة لنقلها عبر الشبكة (JSON) واِعادتها إلي الخادم الذي طلبها لتتم عملية الكتابة .



الشكل (3.6) : يوضح عملية قراءة البيانات من خوادم ال Odoo

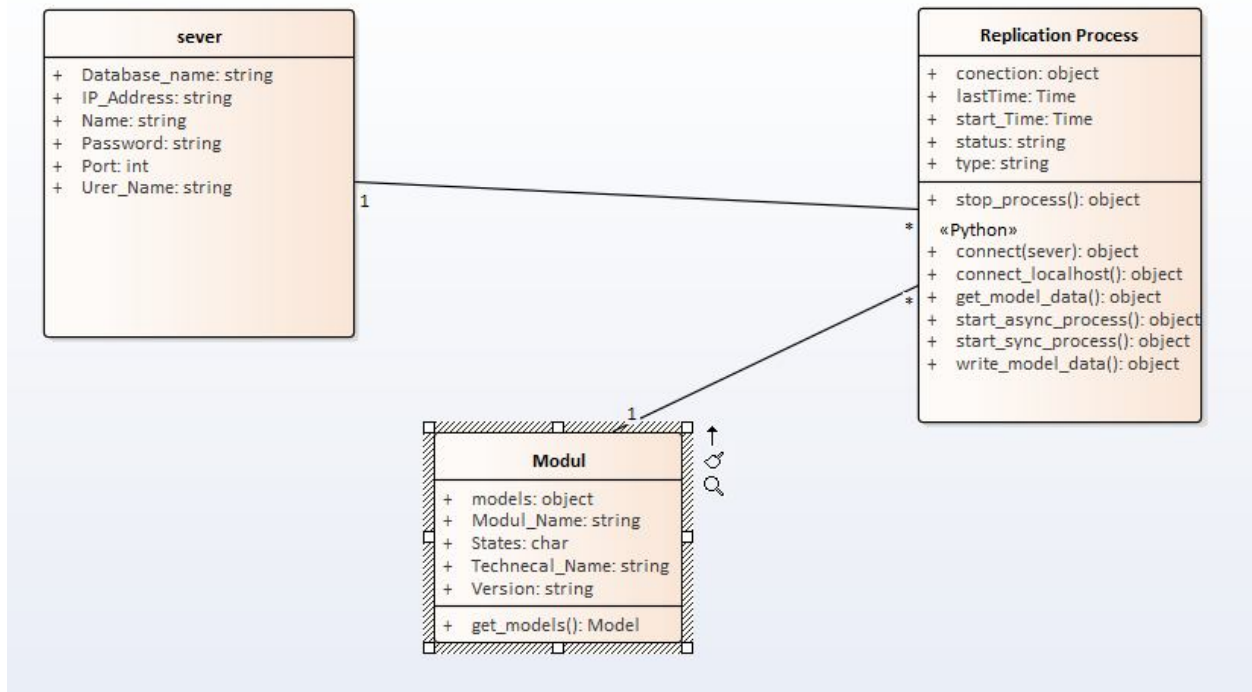
3.3.3.4 عملية الكتابة :

تتم هذه العملية بعد عملية القراءة وفيها يتم تحويل البيانات التي تم إستقبالها من خادم اودو الي صورة تستطيع لغة البرمجة بايثون التعامل معها مثل (List and Dictionary)، ثم يتم التحقق من القيود اللازمة لعملية الكتابة مثل (تطابق الاعمدة الخاصة بكل جدول مع الاعمدة التي تم إستقبالها) . ثم يقوم خادم الودو بترجمة القيود وتحويلها الي إستعلامات (Sql Query) وارسالها الي نظام ادارة قواعد البيانات ، الذي بدوره يقوم بكتابة البيانات .



الشكل (3.7) يوضح عملية الكتابة للبيانات التي تمت قراءتها من الخوادم الاخري

3.3.4 مخطط الفئات (Class Diagram) :



الشكل (3.8) : يوضح الفئات التي يبني عليها النظام

4.1 المقدمة

يتم في هذا الباب عرض رسوم توضيحية لأجزاء النظام وتوضيح كيفية تنفيذ الخطوات التي يتبعها مدير النظام لتنفيذ عملية الاستنساخ .

4.2 مهام النظام الرئيسية

4.2.1 إدارة معلومات الخوادم

يوضح الشكل (4.1) كيفية إضافة معلومات الخادم (Add Servers Information) حيث يقوم مدير النظام بإدخال اسم الخادم (Server Name) ، عنوان الخادم (IP Address) و رقم المنفذ (Port Number) . وبعد عملية ادخال البيانات تتم عملية الاتصال بالخادم ومن ثم يتم إدخال اسم قاعدة البيانات (Database Name) والتأكد من صحتها عن طريق الاتصال وإضافة اسم المستخدم (User Name) وكلمة السر (Password) ثم يتم حفظ البيانات .

server name	localhost	ip address	127.0.0.1
port number	8,069	database name	replica
username	odoo	password	****

الشكل (4.1) يوضح كيفية إضافة معلومات الخادم

يوضح الشكل (4.2) قائمة بأسماء الخوادم الموجودة في النظام ، حيث يتم فيها إستعراض الخوادم المدرجة في النظام والتي تعتمد عليها عملية الاستساح . يمكن عن طريق هذه القائمة إضافة ، او حذف ، او تعديل معلومات اي خادم .

server name	ip address	port number	database name	username	password
hasabo server	192.168.137.71	8,069	replica	odoo	****
localhost	127.0.0.1	8,069	replication	odoo1	****
localhost test	127.0.0.2	8,069	replica	odoo	****

الشكل (4.2) يوضح قائمة بأسماء الخوادم الموجودة في النظام .

4.2.2 إدارة عمليات الاستنساخ

يوضح الشكل (4.3) كيفية إضافة معلومات عملية الاستنساخ ، حيث ان مدير النظام يقوم إضافة إسم الاتصال (Connection Name) ، وتحديد الخادم الذي سوف تتم منه عملية القراءة (Remote Server) ، و الخادم الذي سوف تتم عليه عملية الكتابة (Local Server) ، وذلك اعتماداً على نوع الاستنساخ (Replication Type) ، الذي يقوم مدير النظام بتحديدده في هذه الشاشة كما يمكنه إضافة نظام فرعي او اكثر من نظام (Module)

The screenshot shows a web-based configuration interface for creating a new replication process. The title bar reads 'Process / New'. On the left, there is a sidebar with a navigation menu containing 'Process Information', 'Replication Process' (which is highlighted), 'Servers Information', and 'Servers'. The main content area contains the following fields and controls:

- connection name:** A text input field.
- Remote Server:** A dropdown menu.
- Local Server:** A dropdown menu.
- Start time:** A dropdown menu.
- Last time:** A dropdown menu.
- Replication Type:** A dropdown menu.
- Status:** A text field showing 'Stopped'.
- Modules:** A section with a table header 'Module' and a button 'Add an item' below it.
- Auto Run:** A dropdown menu.

At the bottom of the form, there are two buttons: 'Start Process' and 'Stop'.

الشكل (4.3) يوضح كيفية إضافة معلومات عملية الاستنساخ

يوضح الشكل (4.4) قائمة ببيانات العمليات التي تم إضافتها حيث انه يمكن معرفة العمليات التي تمت إضافتها وزمن كل عمله ونوع الاستنساخ و اخر زمن تمت فيها عملية الاستنساخ ، و حالة كل عملية إذا كانت تعمل حالياً او متوقفه (Status) ،ويمكن عن طريق هذه القائمة إضافة ، او حذف ، او تعديل معلومات اي عملية .

connection name	Modules	Start time	Last time	Status	Replication Type
abdo	1 record	10/14/2018 16:22:37	10/14/2018 16:22:40	Stopped	Master

شكل (4.4) يوضح قائمة عمليات الاستنساخ

يوضح الشكل (4.5) كيفية إضافة بيانات جدولة عملية الاستساخ حيث يمكن ان يتم الاستساخ تلقائياً اعتماداً علي البيانات المدخلة وتحديد التاريخ والوقت وعدد مرات التكرار للاستساخ ،كما يمكن إيقاف او تشغيل عملية الاستساخ المجدولة .

Process Information
Replication Process
Servers Information
Servers

Create: Auto Run

Run Manually

Action Name
Replication Timer On

Model
repmainrep Scheduler User Administrator

Action To Do
Execute Python Code

Execute Every
1 Days

Next Execution Date
10/20/2018 12:49:07

Number of Calls
1

Priority
5

Repeat Missed

Python Code Help

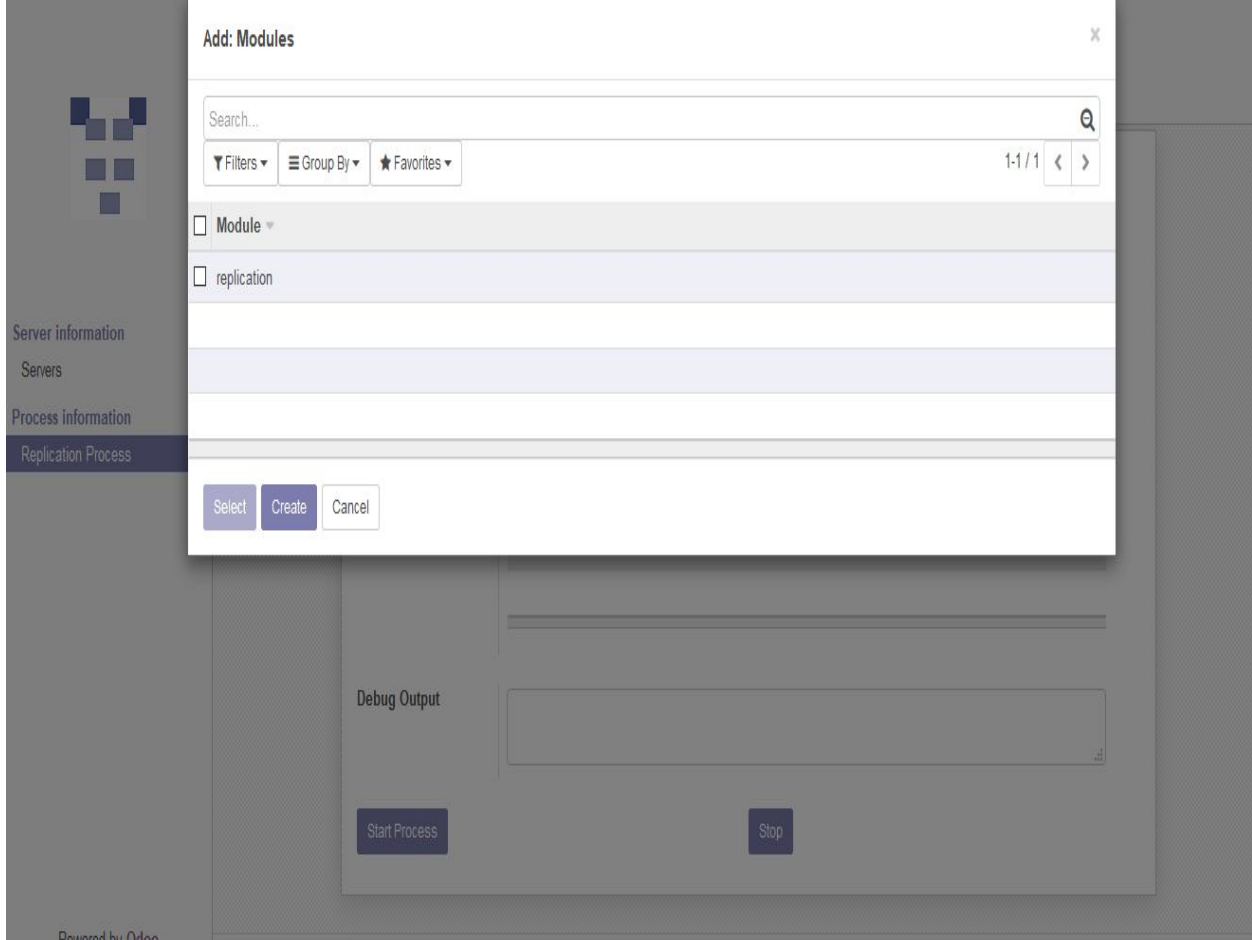
```
1 # Available variables:  
2 # - env: Odoo Environment on which the action is triggered  
3 # - model: Odoo Model of the record on which the action is triggered; is a void recordset  
4 # - record: record on which the action is triggered; may be void  
5 # - records: recordset of all records on which the action is triggered in multi-mode; may be void
```

Save Discard

الشكل (4.5) يوضح كيفية جدولة عملية الاستساخ

4.2.3 إدارة معلومات الانظمة الفرعية

يوضح الشكل (4.6) كيفية اختيار نظام فرعي (Module) او اكثر من الانظمة الفرعية المستخدمة مسبقاً في إحدى العمليات الاخرى وذلك لتنفيذ عملية الاستنساخ عليه ، وفي حالة عدم وجوده في القائمة يقوم مدير النظام بعملية الاضافة وذلك بالضغط علي الزر (Create).



الشكل (4.6) يوضح قائمة لاختيار الانظمة الفرعية

يوضح الشكل (4.7) عملية إضافة نظام فرعي (Module) جديد ، وذلك بعد ان يقوم مدير النظام بالضغط علي زر (Create) ، حيث يقوم المدير بأختيار النظام الفرعي (Module) من قائمة الانظمة الموجود في النظام الرئيسي وذلك إعتياداً علي عملية الاتصال بالخادم وقاعدة البيانات لجلب البيانات الخاصة بالنظام الفرعي ، وبعد اختيار النظام الفرعي تظهر للمستخدم كل الجداول (Models List) الخاصة بالنظام الفرعي الذي تم إختياره بصورة تلقائية ويمكن له إضافة جدول من نظام فرعي اخر كما موضح في الشكل (4.8) ، او حذف اي جدول كما يمكنه عرض تفاصيل كل جدول كما موضح في الشكل (4.10) ويمكنه أيضاً إضافة شرط او مجموعة من الشروط التي تعتمد عليها عملية القراءة لكل جدول (Models Filters).

الشكل (4.7) يوضح عملية إضافة النظام الفرعي (Module)

يوضح الشكل (4.8) عملية اختيار الجدول (model) او الجداول من الخاصة بكل نظام فرعي من قائمة الجداول لجميع الانظمة .

Add: Models List x

Search... Q

Filters Group By Favorites 1-80 / 167 < >

<input type="checkbox"/> Model	Model Description	Type	Transient Model
<input type="checkbox"/> _unknown	_unknown	Base Object	<input type="checkbox"/>
<input type="checkbox"/> barcode.nomenclature	barcode.nomenclature	Base Object	<input type="checkbox"/>
<input type="checkbox"/> barcode.rule	barcode.rule	Base Object	<input type="checkbox"/>
<input type="checkbox"/> barcodes.barcode_events_mixin	barcodes.barcode_events_mixin	Base Object	<input type="checkbox"/>
<input type="checkbox"/> base	base	Base Object	<input type="checkbox"/>
<input type="checkbox"/> base.language.export	base.language.export	Base Object	<input checked="" type="checkbox"/>
<input type="checkbox"/> base.language.import	Language Import	Base Object	<input checked="" type="checkbox"/>
<input type="checkbox"/> base.language.install	Install Language	Base Object	<input checked="" type="checkbox"/>
<input type="checkbox"/> base.module.uninstall	Module Uninstallation	Base Object	<input checked="" type="checkbox"/>
<input type="checkbox"/> base.module.update	Update Module	Base Object	<input checked="" type="checkbox"/>
<input type="checkbox"/> base.module.upgrade	Module Upgrade	Base Object	<input checked="" type="checkbox"/>

Select Create Cancel

الشكل (4.8) يوضح عملية اختيار الجداول (models)

يوضح الشكل (4.9) كيفية إضافة شرط لجدول (Model) ، حيث يقوم المستخدم بتحديد الجدول (Model) و الشرط الذي يحدد مدى البيانات (Domain) كما يمكنه إيقاف شرط معين عن العمل (Active) او تحديد الشرط كشرط رئيسي (Default Filter).

The screenshot shows a dialog box titled "Add: Models Filters" with a close button in the top right corner. The dialog is divided into several sections for configuring a filter:

- Filter Name:** A text input field.
- Available for User:** A dropdown menu currently showing "Administrator".
- Model:** A dropdown menu.
- Default Filter:** A checkbox that is currently unchecked.
- Action:** A dropdown menu.
- Active:** A checkbox that is currently checked.
- Domain:** A text input field with the placeholder text "Select a model to add a filter."
- Context:** A text input field.
- Sort:** A text input field.

At the bottom of the dialog, there are three buttons: "Save & Close", "Save & New", and "Discard".

الشكل (4.9) يوضح عملية إضافة شرط لجدول (Model) .

يوضح الشكل (4.10) التفاصيل الخاصة لكل جدول (Model) ، حيث يوضح الحقول (Fields) وانواعها والقيود الخاصة بكل حقل .

Field Name	Field Label	Field Type	Required	Readonly
__last_update	Last Modified on	datetime	<input type="checkbox"/>	<input checked="" type="checkbox"/>
create_date	Created on	datetime	<input type="checkbox"/>	<input type="checkbox"/>
create_uid	Created by	many2one	<input type="checkbox"/>	<input type="checkbox"/>
display_name	Display Name	char	<input type="checkbox"/>	<input checked="" type="checkbox"/>
id	ID	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>
models_filters	Models Filters	many2many	<input type="checkbox"/>	<input type="checkbox"/>
models_list	Models List	many2many	<input type="checkbox"/>	<input type="checkbox"/>
module_name	Module	many2one	<input type="checkbox"/>	<input type="checkbox"/>

الشكل (4.10) يوضح التفاصيل الخاصة لكل جدول (Model)

النتائج

النتائج التي تم الوصول اليها هي تنفيذ مفهوم الاستنساخ غير المتزامن (Asynchronous) بين نظامين

الاولوالذي ترتب عليه:-

- فصل الانظمة الفرعية عن النظام المركزي وإمكانية عمل الانظمة الفرعية دون الاتصال بالنظام المركزي .
- زيادة إتاحة البيانات .
- خفض الضغط على الشبكة .
- زيادة إعتمادية النظام حيث يمكن للنظام الاستمرار في العمل اذا توقف اي موقع عن العمل

التوصيات

- بعد تطبيق النظام المقترح والحصول على النتائج نوصي بالاتي :
- اضافة نوع الاستنساخ اللحظي (synchronous Replication)
 - إضافة إمكانيات التقسيم الجزئي للجداول (Fragmentation)

الخاتمة

تم بحمد الله تطبيق مفهوم الاستتساخ علي نظام تخطيط الموارد البشرية عن طريق إنشاء اداة كمنظام فرعي

للاودو (Module) تقوم بعملية إستتساخ بيانات الانظمة الفرعية الخاصة بالاودو حسب الجداول الخاصة بكل

نظام فرعي ومزامنة هذه البيانات مع النظام الرئيسي

المراجع

المرجع	الترميز
Fundamentals of Database Systems Sixth Edition Ramez Elmasri Shamkant B. Navthe	[1]
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 11, November 2014 Savyasachi received Amit Kumar Nirala	[2]
https://bucardo.org/Bucardo/	[3]
https://wiki.postgresql.org/wiki/PgCluster	[4]
https://wiki.postgresql.org/wiki/Pgpool-II	[5]
https://wiki.postgresql.org/wiki/Slony	[6]
Learning Python 3rd Edition / Covers Python 2.5 By Mark Lutz	[7]
https://searchmicroservices.techtarget.com/definition/XML-Extensible-Markup-Language	[8]
https://www.postgresql.org/about/	[9]
https://odoo-community.org	[10]
https://pythonhosted.org/OdooRPC/#introduction	[11]
Wireshark User's Guide Version 2.9.0 https://www.wireshark.org/download/docs/user-guide.pdf	[12]
Smartdraw.com : https://www.smartdraw.com/uml-diagram/	[13]