

Università degli Studi di Modena e Reggio Emilia

Facoltà di Ingegneria di Modena

Corso di Laurea in Ingegneria Informatica

**MIGRAZIONE PIATTAFORMA DATI
DA MICROSOFT SQL SERVER 2000
A MICROSOFT SQL SERVER 2008
IL CASO VIDIERRE**

Relatore:

Chiar.mo Prof. Sonia Bergamaschi

Candidato:

Maurizio Rota

INDICE

1. INTRODUZIONE	7
2. PIATTAFORMA DATI LEGACY VIDIERRE	9
2.1. Presentazione VIDIERRE	9
2.2. Il sistema informativo aziendale	11
2.3. Piattaforma produttiva in Microsoft SQL Server 2000	17
3. ANALISI COMPATIBILITA' DI DATABASE E SOFTWARE APPLICATIVI IN MICROSOFT SQL SERVER 2008	18
3.1. Introduzione alle Virtual Machines con VMWARE	18
3.2. Le VM in VIDIERRE	21
3.3. Installazione VM per test di compatibilità	23
3.4. Microsoft SQL Server Upgrade Advisor 2008	25
3.5. Confronto tra le varie versioni di SQL Server 2008	29
3.6. Installazione e configurazione della versione scelta	31
3.7. Importazione database da SQL Server 2000	34
3.7.1. Backup e Restore dei database	36
3.8. Comparazione strutturale e prestazionale	38
3.8.1. SQL Server Comparison Tool	38
3.8.2. Valutazione prestazionale con query test	42
3.8.3. Beta testing applicativi aziendali su SQL 2008	46
4. MIGRAZIONE DELLA PIATTAFORMA DATI VIDIERRE A MICROSOFT SQL SERVER 2008	52
4.1. Installazione e configurazione nuovo SQL Server	52
4.1.1. Migrazione degli account utente	53
4.1.2. Utilizzo di AWE per la memoria server	58
4.1.3. Impostazione manuale delle opzioni di memoria	59
4.2. Problematiche di migrazione	62
4.2.1. Sp_compatibility_level	62
4.2.2. Da DTS a DTSX	63
4.2.3. Uso restrittivo di apici in campi testo	67
4.2.4. Viste strutturate in Management Studio 2008	69
4.2.5. Reimpostazione Server Jobs	71
5. DISCUSSIONE E NOTE CONCLUSIVE	73
6. BIBLIOGRAFIA	77

1. INTRODUZIONE

L’obiettivo del progetto di tesi è la migrazione di una piattaforma dati da *Microsoft SQL Server 2000* a *Microsoft SQL Server 2008*. Si vogliono verificare effettivi miglioramenti prestazionali ed effettiva convenienza nel caso di un utilizzo fatto da un’Azienda reale, nello specifico VIDIERRE, circa il rapporto tra eventuali e auspicabili miglioramenti e relativi investimenti economici.

Nel caso specifico, si valuteranno prestazioni, numero e gravità dei problemi riscontrati nella migrazione, corretto funzionamento delle stored procedures già presenti, effettive richieste hardware e assenza di perdite dati per *Microsoft SQL Server 2008*.

Il lavoro di tesi volgerà alla produzione di dati specifici, di riferimenti reali e di confronti prestazionali tra le 2 piattaforme in esame per permettere all’Azienda una valutazione obiettiva sulla reale convenienza di una migrazione ad una piattaforma più recente, in termini di guadagni prestazionali e investimenti economici.

Il lavoro di tesi partirà con l’installazione su una nuova macchina fisica (destinata ad andare in produzione) di una macchina virtuale rispecchiante le medesime caratteristiche della macchina fisica attualmente in produzione, e di un’altra macchina virtuale che abbia, al contrario, le stesse caratteristiche del server fisico che verrebbe usato in produzione e che effettivamente funge da host per quella virtuale. L’utilizzo di una macchina virtuale si rende necessario visto il continuo utilizzo 24/7 in produzione dell’intera piattaforma dati (\approx 90 GB) e visto l’obiettivo stesso della tesi da sviluppare su piattaforme differenti.

Si procederà, come sarà trattato nel Capitolo 3, all’installazione sulle macchine virtuali del *Microsoft SQL Server 2008* e si passerà, sulla nuova piattaforma, uno dei database in produzione. Per la realizzazione di tale passaggio, si adotterà una procedura comprensiva di 2 step:

- backup del database in *Microsoft SQL Server 2000* in esecuzione sulla macchina fisica in produzione

- restore del database in *Microsoft SQL Server 2008*, a partire dal 2000, utilizzando apposito tool presente in tutte le versioni del motore relazionale di casa *Microsoft*

Al termine del processo di passaggio dati, si prevede una forte presenza di errori di compatibilità, che andranno prontamente affrontati e risolti. Si procederà, infine, all'effettiva valutazione prestazionale in parallelo delle differenti piattaforme attraverso l'utilizzo di procedure comunemente usate nel lavoro giornaliero (ad es. la chiamata di stored procedures o l'utilizzo di software applicativi).

La fase cruciale del progetto di tesi prevede la restituzione finale di risultati comparativi tra le differenti piattaforme testate sia a livello economico che prestazionale. Sarà inoltre effettuata una comparazione tra vantaggi e svantaggi (in termini di tempi e risorse utilizzate) dei processi migratori, così da consentire all'Azienda la valutazione finale sulla convenienza reale di un upgrade definitivo a *Microsoft SQL Server 2008*.

La verifica circa la convenienza dell'upgrade, ha portato alla decisione da parte dell'Azienda di procedere con il processo di aggiornamento. Nel capitolo 4 sarà quindi illustrata la procedura di installazione sulla nuova macchina fisica della nuova piattaforma, l'effettivo passaggio di tutti i database, la verifica effettiva dell'assenza di problemi o della loro pronta risoluzione (in base all'esperienza derivante dai test effettuati sulle macchine virtuali). Saranno inoltre illustrate le soluzioni alle problematiche non individuabili in fase di test e, infine, la conclusiva messa in funzione a regime dell'intero sistema.

Nell'ultimo capitolo, “Discussione e note conclusive”, si esporranno infine i vantaggi derivanti dalla migrazione alla nuova piattaforma dati e, conseguentemente, le motivazioni che hanno spinto l'Azienda all'investimento per l'upgrading a *Microsoft SQL Server 2008*.

2. PIATTAFORMA DATI LEGACY VIDIERRE

L'intero progetto di tesi è stato pianificato, studiato ed eseguito all'interno dell'Azienda VIDIERRE di Reggio Emilia. La possibilità di operare "sul campo" ha fornito strumenti e opportunità sperimentali che ai fini del progetto di tesi hanno ampliato l'argomentazione, altrimenti puramente teorica, portando esempi, vantaggi e problematiche reali di applicazione degli argomenti trattati direttamente su una piattaforma dati legacy in produzione. La piattaforma dati VIDIERRE, benché basata su un sistema che rispetta le linee di un tipico "*legacy system*" [1] dal punto di vista delle sue caratteristiche "*mission-critical*" (cioè supporta il core delle operazioni aziendali interne ed è costantemente utilizzato, quindi fondamentale per l'operatività dell'organizzazione), "*investment*" (su tale sistema l'Azienda ha pesantemente investito risorse economiche e umane nel corso degli anni con features sviluppate in base alle esigenze resesi necessarie con il passare degli anni) e "*dimension*" (si tratta di un sistema basato su migliaia di files e milioni di linee di codice, distribuite su numerosi software applicativi proprietari), può essere considerato all'avanguardia rispetto a un tipico sistema legacy, grazie alla sua scalabilità e alla sua flessibilità, dovute anche ai forti programmi di ammodernamento applicati negli anni sia al nucleo del sistema stesso, sia agli applicativi che vi si appoggiano per il loro funzionamento. Proprio questa "*atipicità*" rispetto a un sistema legacy standard ha permesso, durante il processo di tesi, di separare le attività progettuali e operare in maniera modulare su ciascun ambito di applicazione dell'intero progetto di tesi.

2.1 Presentazione VIDIERRE

VIDIERRE, società del gruppo Assist, è un'Azienda indipendente, certificata ISO 9001:2008, leader in Europa nel monitoraggio e nell'analisi dei media.



Figura 2.1: Logo VIDIERRE – Assist Group

Dal 1993 a oggi VIDIERRE ha registrato, analizzato e catalogato le apparizioni di marchi, claim, prodotti e personaggi su televisione, stampa e web, per oltre 600 clienti nazionali e internazionali. Per la sua produttività, VIDIERRE utilizza sistema e software proprietari per la rilevazione e archiviazione delle apparizioni tv e mette a disposizione di aziende, istituzioni, organi d'informazione, personaggi e testimonial dati completi e oggettivi sulla visibilità mediatica dei soggetti monitorati, in Italia e nel mondo. VIDIERRE è acronimo di Video Data Recording: il sistema, infatti, registra 365 giorni all'anno, 24/7, i palinsesti di un panel composto da 26 emittenti italiane, 139 emittenti presenti sul territorio europeo, 282 emittenti nel mondo, e cataloga i contenuti visivi, verbali e testuali (tracce video e audio) del palinsesto completo di ogni emittente monitorata.

Il sistema VIDIERRE si basa sull'azione integrata di *rilevazione digitale* (video scanner digitale - VSD) e *personale tecnico specializzato*, composto da ingegneri, analisti e addetti alla rilevazione, e viene applicato sui palinsesti effettivamente trasmessi dai broadcaster che vengono registrati integralmente, archiviati e indicizzati. I risultati di 18 anni di attività sono tutti archiviati in un Database storico unico, fra i più estesi esistenti sul mercato, contenente i dati di rilevazione, i risultati delle analisi e le registrazioni dei passaggi televisivi realizzati per oltre 600 Clienti nazionali e internazionali. Oltre 35 terabyte online e 150 terabyte offline di dati puri e tracce audio-video, che permettono agli analisti di VIDIERRE di realizzare comparazioni approfondite su archi temporali diversi, avendo sempre disponibili online i palinsesti completi di tutte le emittenti monitorate negli ultimi 24 mesi. Da questo Database storico nascono prodotti di ricerca specifici, quali l'Analisi Preventiva (che permette di calcolare la brand exposure di un marchio sponsor prima che la sponsorizzazione abbia inizio), e servizi di supporto alle decisioni, quali l'Osservatorio Sportivo (che offre una reportistica dettagliata degli sport in tv), realizzati con l'occhio attento di chi "legge" al secondo, come VIDIERRE, e non al minuto come Auditel. VIDIERRE, a differenza di Auditel, analizza anche immagini/audio interni ai cosiddetti contenitori come i telegiornali. La prerogativa di VIDIERRE è quindi la lettura al secondo, agganciata successivamente alla rilevazione al minuto di Auditel. Da sottolineare che tutto ciò è *parte integrante*, ma non costituisce *l'intero servizio sponsoring* fornito da VIDIERRE.

VIDIERRE, inoltre, benché abbia lo sponsoring e il monitoraggio media come core-businesses, prevede numerosi altri servizi in produzione, tra i quali, a livello puramente informativo, si citano il *MIP* (Monitoraggio Investimenti Pubblicitari, adottato dalle più grandi case automobilistiche a livello mondiale), *l'Eye Tracking* (collocato nell'ambito del neuro-marketing, è il metodo attraverso il quale è possibile rilevare e analizzare il modo in cui le persone osservano un'immagine statica o in movimento, (c.d. stimolo visivo), attraverso il tracciamento del movimento della pupilla umana – durante la visione di un frame televisivo, ad esempio – e l'analisi del battito cardiaco per capire se le reazioni a un determinato impulso visivo sono positive o negative, con la successiva elaborazione al millesimo di secondo dei riscontri pervenuti) ed il *Product Placement* (il cui monitoraggio costante consente di pianificare al meglio gli investimenti in comunicazione e monitorare il collocamento di marchi/prodotti all'interno di programmi e contenitori televisivi).

2.2 Il sistema informativo aziendale

Il reparto Information Technology di VIDIERRE opera per tutte le società del gruppo Assist, dalla progettazione e consulenza all'assistenza. Con il passare del tempo e l'evolversi delle richieste le mansioni sono progressivamente aumentate. Al momento, il reparto IT si occupa delle seguenti aree operative:

1. Registrazione digitale
2. Elaborazione dati
3. Servizi internet
4. Servizio di assistenza esterno
5. Servizi di assistenza interno

Le aree 1, 2 e 5 sono ad esclusivo interesse interno, mentre le aree 3 e 4 si rivolgono a tutte le aziende del gruppo Assist, nelle varie sedi in cui quest'ultimo opera. Mentre che per le ultime due aree, la 4 e la 5, il servizio si avvale quasi completamente delle conoscenze e delle capacità personali del team IT di VIDIERRE, per le aree di cui ai punti 1, 2 e 3 è stato necessario progettare una struttura ad hoc che si va ad illustrare nel presente paragrafo.

Si rende doverosa un'annotazione: per comodità interna, vengono “impropriamente” definite *server* tutte le macchine che, 24/24h, restano attive ed a disposizione degli utenti, dedicate a

servizi particolari, benché su alcune di quelle macchine non sia installato un sistema operativo propriamente denominato *server*.

Registrazione Digitale

VIDIERRE ha un sistema integrato di registrazione e di storage delle immagini che permette di visualizzare su ogni computer della sede tutto ciò che è trasmesso dai broadcaster, 24/24h, all'interno del panel delle emittenti monitorate. Il segnale video digitale satellitare e digitale terrestre è “registrato” mediante le macchine di encoding, per poi passare alla fase di “muxing” delle relative tracce audio e video, unendole in file unici. Il file prodotto è successivamente depositato all'interno delle NAS (network area storage) e contemporaneamente catalogato ed inserito nel database delle immagini. Il file è infine caricato nelle postazioni di lavoro, attraverso l'utilizzo di applicativi proprietari, ogni qualvolta ve ne sia bisogno, per l'analisi dell'evento e per la rilevazione dei marchi presenti all'interno dei singoli frames e successiva elaborazione con i sistemi di VIDIERRE.

Con un totale di 35 terabyte di spazio su disco rigido, sono immediatamente disponibili online, quasi 2160 giorni di programmazione televisiva, corrispondenti a circa 90 giorni di programmazione completa del panel di emittenti televisive monitorate. I file più vecchi di tre mesi continuano ad essere considerati online, in quanto, grazie a procedure automatizzate, sono archiviati all'interno dei nastri LTO Ultrium 4 e contenuti in una libreria robotizzata che a sua volta vanta una capacità di 64 terabyte su nastro. Qualora sia richiesto un file presente all'interno di uno dei succitati nastri, il sistema provvede automaticamente al suo restore, copiandolo su una NAS e rendendolo accessibile ai software di visualizzazione. Di seguito lo schema dell'hardware necessario ed utilizzato per questa fase del processo produttivo.

Nome	Modello	Sistema Operativo	Processore	Storage	Funzione/Servizio
Encoder1	HP WORKSTATION XW6400	Windows XP Professional	Intel Xeon 2.33 GHz	48 + 508 GB	Registrazione digitale
Encoder2	HP WORKSTATION XW6400	Windows XP Professional	Intel Xeon 2.33 GHz	48 + 508 GB	Registrazione digitale
Encoder3	HP WORKSTATION XW6400	Windows XP Professional	Intel Xeon 2.33 GHz	48 + 508 GB	Registrazione digitale
ImportDati	HP PROLIANT ML350 G5	Microsoft Windows Server 2003 R2 Standard Edition SP2	Intel Xeon 5130 2.00 GHz	2x36 GB SAS	Servizio spostamento su Nas22tb + Backup di ImportVideo
ImportVideo	HP PROLIANT ML350 G5	Microsoft Windows Server 2003 Standard Edition SP2	Intel Xeon 2.00 GHz	2x36 GB SAS	Muxer
ML6030	DELL ML6030	Firmware Dell			LIBRERIA LTO 4
NAS	SUPERMICRO NAS	E-Open		16 TB	NAS
Nas22tb	SUPERMICRO NAS	E-Open		22 TB	NAS

Elaborazione dati

Per ciò che concerne la raccolta e l'elaborazione dei dati, come si analizzerà in seguito e in maniera più approfondita in questo progetto di tesi, si utilizza un database poggiato su una piattaforma *Microsoft SQL Server*. All'interno di questo database, tra tutte le informazioni raccolte, si citano all'uopo della presente descrizione i singoli passaggi dei marchi, i dati Auditel di audience delle trasmissioni ed i listini pubblicitari. Questo richiede una ingente potenza di calcolo, necessità che per VIDIERRE cresce di anno in anno per i seguenti motivi:

- il database assume dimensioni maggiori con incrementi sostanziali, dell'ordine di oltre 1 GB l'anno, e ciò comporta l'aumento della complessità e della tempistica di esecuzione per l'aggregazione e l'elaborazione dei dati necessari agli applicativi per la produzione di report personalizzati per ciascun Cliente di VIDIERRE;
- il numero di report generati aumenta di stagione in stagione. Da parte delle Aziende Clienti la gestione delle sponsorizzazioni diventa oggetto di analisi sempre più rigorosa, per un numero crescente di aziende. Ad esempio: un'Azienda che 5 anni fa investiva in sponsorizzazioni una percentuale del proprio budget comunicazione scegliendo la squadra o il testimonial della propria città, oggi è diventata giocoforza più sensibile alle performance del proprio investimento, cercando con costanza di ottenere il massimo dal capitale investito. Per tali ragioni, non è raro che ciascuna Azienda Cliente richieda benchmark della visibilità con i periodi precedenti e accompagnati, in caso di performance negative, dalle motivazioni che hanno portato il marchio in questione ad “esser meno visto” nelle apparizioni. La crisi economica del 2009 ha accentuato questa tendenza.
- i Clienti di VIDIERRE hanno bisogno di risposte sempre più veloci: spesso i marketing manager chiedono focus particolari, indagini e dati alcuni giorni o addirittura solo alcune ore antecedenti l'incontro con i loro Clienti. Anche questo fenomeno è in costante crescita.

Per questi motivi VIDIERRE è propensa periodicamente (ogni 18 mesi) a potenziare o rinnovare il server di produzione, investendo su hardware più performante o su soluzioni che consentano notevoli miglioramenti nei tempi e conseguentemente nei servizi forniti ai Clienti. Di seguito lo schema dell'hardware necessario ed utilizzato per questa fase del processo

produttivo, composto oltre al server di produzione *King*, anche da macchine (macchine virtuali, VM – Cfr. Par. 3.2), dedicate a svariati servizi, che accompagnano il lavoro quotidiano degli analisti.

Nome Macchina	Caratteristiche hardware	Sistema operativo	Funzione/servizio
KING	Modello: DELL POWEREDGE R710 Processori: 2 x Intel Quad-Core Xeon E5520 2.26 GHz RAM Installata: 32 GB	Microsoft Windows Server 2003 Enterprise Edition 64 bit	SQL SERVER
Knight (VM)		Microsoft Windows Server 2003	SQL Server TEST
Melkor (VM)	Modello: DELL POWEREDGE 2950 III	Microsoft Windows 2000 Advanced Server	Terminal server per inserimento dati da utenti fuori sede
Odin (VM)	Processori: 2 x Intel Quad-Core Xeon E5405 2.00 GHz	Microsoft Windows 2000 Advanced Server	Domain Controller + DNS Server e DHCP
Query (VM)		Microsoft Windows XP Professional	Dedicata al lancio query di Access
Tank (VM)	RAM Installata: 16 GB	Microsoft Windows XP Professional	Dedicata alla produzione dei prospetti

Elaborazione dati (Postazioni Di Lavoro, PDL)

All'interno della sede VIDIERRE in Via Honoré de Balzac a Reggio Emilia, sono presenti 24 postazioni di lavoro, con 19 utenti reali. La maggior parte delle postazioni è munita di un doppio monitor: vista la necessità di lavorare principalmente sulle immagini video, un monitor è dedicato ai filmati ed uno all'applicativo proprietario di VIDIERRE di rilevazione marchi ed inserimento od elaborazione dati. Da sottolineare, inoltre, l'attenzione dell'Azienda all'innovazione delle macchine di ciascuna postazione di lavoro: VIDIERRE, infatti, prevede ogni anno un budget riservato per il potenziamento di ciascuna macchina, così da migliorarne le performance seguendo il graduale aumento di risorse hardware richieste dai vari software con il passare degli anni.

Il massiccio utilizzo di filmati provoca uno scambio molto intenso di informazioni fra la sala server e le PDL durante tutta la giornata: per questo motivo l'intera sede VIDIERRE è interconnessa con una rete LAN interamente a Gigabit, e cablata con cavi di Cat. 6.

Di seguito lo schema dell'hardware dei server (macchine virtuali, VM – Cfr. Par. 3.2), che permettono il lavoro degli utenti in questa fase del processo produttivo, in aggiunta a tutte le macchine descritte nei punti precedenti.

Caratteristiche hardware	Nome Macchina	Sistema operativo	Funzione/servizio
Modello: HP PROLIANT DL380 G3 Processori: 2 x Intel Xeon 2.80 GHz RAM Installata: 2 GB	MYSTIQUE	Microsoft Windows Server 2003 Standard Edition	File server e Backup server su Unità nastro Ultrium 2
	SpotTV (VM)	Microsoft Windows XP Professional	Macchina dedicata al software per l'analisi del database costi spot
	Palinsesti (VM)	Microsoft Windows XP Professional	Macchina dedicata al software per l'analisi dei database palinsesti
Modello: HP PROLIANT DL380 G5 Processori: 1 x Intel Quad-Core Xeon 5130 2.00 GHz RAM Installata: 8 GB	Morpheus (VM)	Microsoft Windows XP Professional	Dedicata alla produzione dei prospetti
	Plate (VM)	Linux CentOS	Web Server + Forum, VPN e Blog
	Radio01 (VM)	Microsoft Windows XP Professional	Macchina dedicata alla registrazione delle emittenti radiofoniche
	Radio02 (VM)	Microsoft Windows XP Professional	Macchina dedicata alla registrazione delle emittenti radiofoniche
	Streaming02 (VM)	Ubuntu 8.04 LTS	Servizio fornitura immagini in streaming (test)
	Gordon (VM)	Microsoft Windows Server 2000	SQL TEST con piattaforma Microsoft SQL Server 2000
Modello: HP PROLIANT DL380 G5 Processori: 1 x Intel Quad-Core Xeon 5130 2.00 GHz RAM Installata: 8 GB	Pawn (VM)	Linux CentOS	Proxy e Antivirus Server
	Streaming01 (VM)	Ubuntu 8.04 LTS	Servizio fornitura immagini in streaming (test)
	Valmar (VM)	Microsoft Windows 2000 Advanced Server	Domain controller, DNS Interno, DHCP
	WMRecorder (VM)	Microsoft Windows XP Professional	Macchina dedicata al servizio di registrazione filmati su web
	Zeus (VM)	Microsoft Windows Server 2000	Macchina di backup per il lancio di prospetti e query access

Servizio Internet: DMZ

Per DMZ (Demilitarized Zone) si intende un'area della LAN isolata, raggiungibile sia da reti interne (LAN) che esterne (Internet). La DMZ VIDIERRE si pone, dal punto di vista funzionale, interposta tra la LAN e la rete Internet. Ogni server operante in DMZ è separato sia dalla LAN che da Internet per mezzo di Firewall Cisco. Pertanto questi server sono severamente soggetti ad accessi limitati e regolamentati, sia verso una direzione che verso l'altra.

I servizi forniti dai server operanti all'interno della DMZ sono tutti “internet oriented”, e nello specifico forniscono funzionalità di FTP server, Web server, DNS ed Email server. In figura 2.2 è rappresentata schematicamente la struttura della DMZ e della rete VIDIERRE.

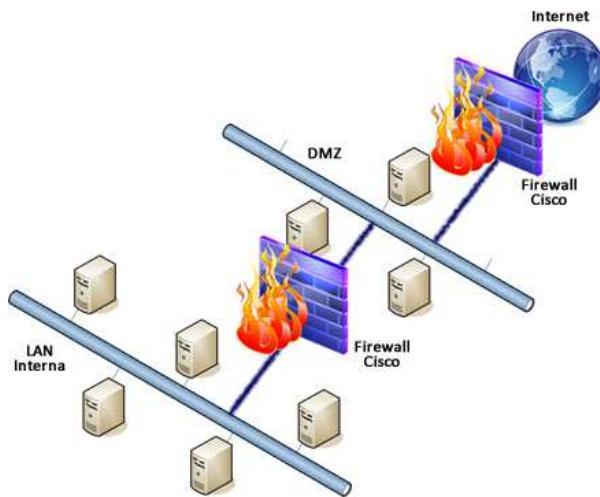


Figura 2.2: Struttura schematica LAN e DMZ di VIDIERRE

Per effettuare hosting di diverse tipologie di siti internet (a tal riguardo si riporta che VIDIERRE è al momento un mantainer registrato presso il NIC), l'Azienda ha deciso di poggiare questo tipo di produttività sull'utilizzo di più server web, ognuno con piattaforme ed ambienti diversi (sia Linux, sia Microsoft). Di seguito lo schema dei server (macchine virtuali, VM – Cfr. Par. 3.2) operanti all'interno della DMZ di VIDIERRE.

Caratteristiche hardware	Nome Macchina	Sistema operativo	Funzione/servizio
Modello: HP PROLIANT DL380 G4	Artemisia (VM)	Microsoft Windows Server 2003	Web server con applicativo IBM LOTUS DOMINO
Processori: 1 x Intel Xeon 3.40 GHz	Baal (VM)	Linux CentOS	Mail server del gruppo
RAM Installata: 6 GB	Bishop (VM)	Linux CentOS	DNS primario esterno
	Freya (VM)	Microsoft Windows Server 2003	Web server, FTP server
	VidierreWeb (VM)	Linux CentOS	Web server

2.3 Piattaforma produttiva in Microsoft SQL Server 2000

Facendo riferimento a tutti i settori produttivi che permettono a VIDIERRE di operare sul mercato (Cfr. Par. 2.1), è fondamentale considerare come il server in produzione, o meglio la piattaforma produttiva basata su *Microsoft SQL Server 2000 Enterprise Edition*, su cui esistono e lavorano i vari database collegati a ciascun settore produttivo aziendale, sia di importanza vitale per l’Azienda stessa. Basti pensare che il server *Gordon*, che all’inizio del progetto di tesi rappresentava l’unico server in produzione, è stato riavviato a fini manutentivi, nel corso dell’anno 2009, solamente 2 volte, per un downtime complessivo di 65 minuti. All’uopo della stabilità, dunque, va spezzata una lancia in favore del motore *SQL Server 2000* di casa Microsoft, in quanto ha sempre svolto egregiamente le sue funzioni, mantenendo online tutti i database al suo interno.

La situazione che si presentava all’inizio del progetto di tesi, dunque, era parecchio complessa e strutturata, forse anche inquietante in alcuni dettagli se si considera il profondo processo di cambiamento cui si è programmato lo studio e cui andava incontro l’intera piattaforma dati. Nello specifico, qui di seguito sono specificati i database residenti sul *SQL Server 2000*, tutti in produzione ed operativi, alla base dei diversi settori produttivi aziendali:

- *vidierre2k*: database dedicato allo sponsoring ed al monitoraggio media;
- *MIP*: database dedicato all’omonimo business interno all’operatività di VIDIERRE;
- *areaclienti*: database su cui si poggia l’intera Private Area di VIDIERRE (piattaforma di comunicazione one-to-one con i Clienti finali);
- *audience*: database contenente lo storico dei dati audience;
- *presenze*: database su cui si poggia il sistema di Marca Presenze interno a VIDIERRE;
- *12 ulteriori database*, di cui non si entra nel merito per sensibilità dati aziendali.

Ciascuno dei 17 succitati database sarà coinvolto nella migrazione a *Microsoft SQL Server 2008*. Oltre ai database, costituiscono parte integrante della piattaforma dati anche i 18 processi tra server jobs e piani di manutenzione attualmente impostati ed operativi in *SQL Server 2000*, che unitamente ai pacchetti DTS (Data Transformation Service), di cui si parlerà in seguito, saranno soggetti a migrazione. La prerogativa fondamentale che il processo di migrazione dovrà tenere in dovuta considerazione, dunque, riguarda appunto la stabilità e la totale compatibilità dei database e degli oggetti esistenti sul *SQL Server 2000* con il risultato della migrazione stessa, in modo da preservare la produttività continua di VIDIERRE.

3. ANALISI COMPATIBILITA' DATABASE E SOFTWARE APPLICATIVI IN MICROSOFT SQL SERVER 2008

L'intera piattaforma dati di VIDIERRE è in produzione per 24/24h, 365/365gg, e non può mai essere interrotta, in nessun caso, poiché in ogni momento della giornata ci sono software, risorse umane ed procedure che agiscono sui database. Per queste fondamentali ragioni, si è reso necessario utilizzare la virtualizzazione, percorso sul quale VIDIERRE già da tempo ha investito risorse economiche ed umane, ottenendo molteplici vantaggi. La virtualizzazione, infatti, ha permesso durante il progetto di tesi di poter operare senza alcun rischio e poter procedere in maniera approfondita all'analisi di compatibilità dei database e dei software applicativi cui si riferiscono.

3.1 Introduzione alle Virtual Machines con VMWARE

In passato, nei sistemi informativi aziendali, ci si è sempre basati su un modello “one server, one application”, prevedendo n macchine che svolgessero n servizi o che fossero incaricate di eseguire n applicativi. Analizzando la struttura di un qualsiasi sistema informativo del passato, si poteva infatti scorgere un server che fungeva da Domain Controller, un altro server che si occupava dello sharing di files, un ulteriore server che si occupava, magari, di hosting web o mail server, ed infine un server, in genere il più potente, che fungeva da piattaforma per un motore database. Tutto questo, ovviamente, ha sempre richiesto quello che col senno di poi e con l'occhio al futuro può essere definito uno “spreco” di risorse. Generalmente, infatti, la maggior parte del tempo impiegato da un responsabile IT aziendale è sempre stata focalizzata alla gestione dei server esistenti, piuttosto che al loro rinnovo ed all'innovazione tecnologica. Addirittura, circa il 70% dei budget tipicamente riservati ai reparti IT delle aziende “non virtualizzate” è impiegato per il “mantenimento” delle infrastrutture esistenti, con un residuo di budget a volte inesistente destinato alla ricerca ed all'innovazione [2].

In questo contesto, l'introduzione del concetto di “*virtual machine*” (VM) risulta essere una pietra miliare per la ristrutturazione delle infrastrutture e la conseguente gestione delle risorse e dei budget. In origine, il termine *virtual machine* era usato per indicare la creazione di una molteplicità di ambienti di esecuzione identici in un unico computer, ciascuno con il proprio sistema operativo. Il significato più comune oggi è quello di un programma che emula un

calcolatore (di solito un calcolatore astratto, cioè un calcolatore a cui non corrisponde un calcolatore reale). I programmi applicativi sono scritti in un linguaggio che viene compilato per questo calcolatore immaginario (cioè tradotti nelle sue istruzioni native) e, una volta compilati, sono eseguiti sulla macchina virtuale software, che può agire o come interprete o come compilatore “al volo” (compilazione just in time). Poichè si possono scrivere diverse macchine virtuali per diverse piattaforme, il programma compilato può “girare” su qualsiasi piattaforma su cui “gira” la macchina virtuale [3].

L'*Hypervisor* è il componente chiave per un sistema basato sulla virtualizzazione, in quanto è il software che permette la “divisione” di un unico sistema fisico in più sistemi virtuali. Risulta evidente quanto sia fondamentale che l’Hypervisor stesso abbia una sua comprovata affidabilità, ed una struttura che gli permetta di svolgere la sua attività senza “pesare” sul funzionamento e sulle prestazioni delle macchine virtuali che amministra. In figura 3.1 si può visualizzare schematicamente il concetto di funzionamento delle VM.

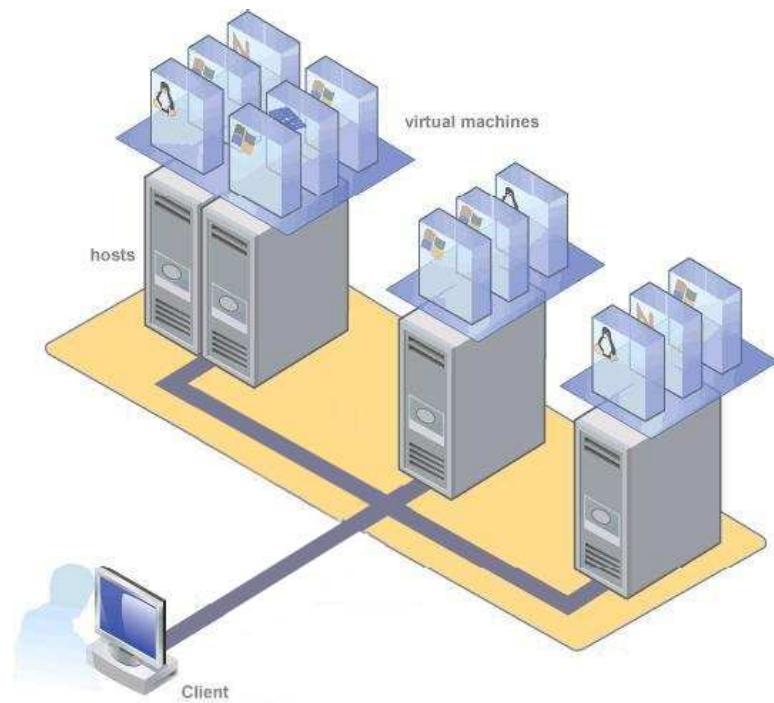


Figura 3.1: Schema concettuale di funzionamento delle Virtual Machines

Nel corso degli anni, una delle Aziende che più si è specializzata nello sviluppo di software per la realizzazione di VM per sistemi x86 e x86-64, raggiungendo traguardi importanti e rilasciando software a mio giudizio alquanto stabili e con grande usabilità, è la VMWare Inc. di Palo Alto, California. Anche VIDIERRE ha scelto di utilizzare le soluzioni software per la virtualizzazione proposte dalla VMware, dopo attente analisi e numerosi test confrontandole con numerose altre piattaforme software.

La piattaforma VMware per la virtualizzazione, utilizzata attualmente da VIDIERRE, è la *VMware Server ESXi*, giunta alla versione 4.1.0, Build 260247. Tale prodotto, per la configurazione fisica strutturale del sistema informativo di VIDIERRE e conseguentemente dei server fisici su cui è stato montato il server *ESXi* di VMware, è realmente molto vantaggioso dal punto di vista del licensing e delle limitazioni che comporta.

Come indicato sul sito ufficiale della VMware, vi è una duplice possibilità:

- il download di una versione trial (60 gg) del *VMware Server*, comprendente tutte le funzioni di ESX. Il periodo di prova di 60 giorni ha inizio non appena si “accende” per la prima volta la macchina ESX, dopo l’installazione. Tale versione “trial” non prevede alcun tipo di limitazione, ed al termine del periodo di prova, qualora non si fornisca un numero di licenza valido ottenuto dopo l’acquisto della licenza stessa, la versione installata “switcha” automaticamente ad ESXi (vedi punto seguente);
- il download di una versione free del server di virtualizzazione, chiamato *VMware ESXi*. Per l’installazione del prodotto, è necessaria una registrazione gratuita, al termine della quale si riceve un numero di licenza personale da impostare all’interno del programma. La versione *ESXi*, essendo gratuita, prevede alcune limitazioni, tra le quali l’impossibilità di utilizzo di più di 4 core per ogni VM installata (quindi un limite alla potenza massima di calcolo di cui ciascuna VM potrà usufruire).

La versione di *VMware ESXi* – dunque gratuita – utilizzata da VIDIERRE, tuttavia, non provoca disagi alla produttività aziendale, in quanto il server più potente su cui è montata è composto da 2 cpu quad-core, e ciò permette alle VM che vi girano di sfruttare nel complesso tutta la potenza disponibile.

3.2 Le VM in VIDIERRE

In passato, anche VIDIERRE si è sempre poggiata sul modello organizzativo “one server, one application”, prevedendo n macchine che svolgessero n servizi. La Sala Server aziendale, infatti, era stipata di server fisici nell’ordine di trenta macchine¹, oltre alla macchina in produzione *Gordon* su cui era montato il *SQL Server*. Ciò ovviamente comportava altissimi costi di manutenzione, in termini di spese economiche ed impiego di risorse umane che ne gestissero le esigenze, oltre ovviamente a problemi di surriscaldamento, cablatura e non ultimo spazio. In figura 3.2 è possibile individuare schematicamente il sovraffollamento di macchine fisiche presenti in VIDIERRE in passato.

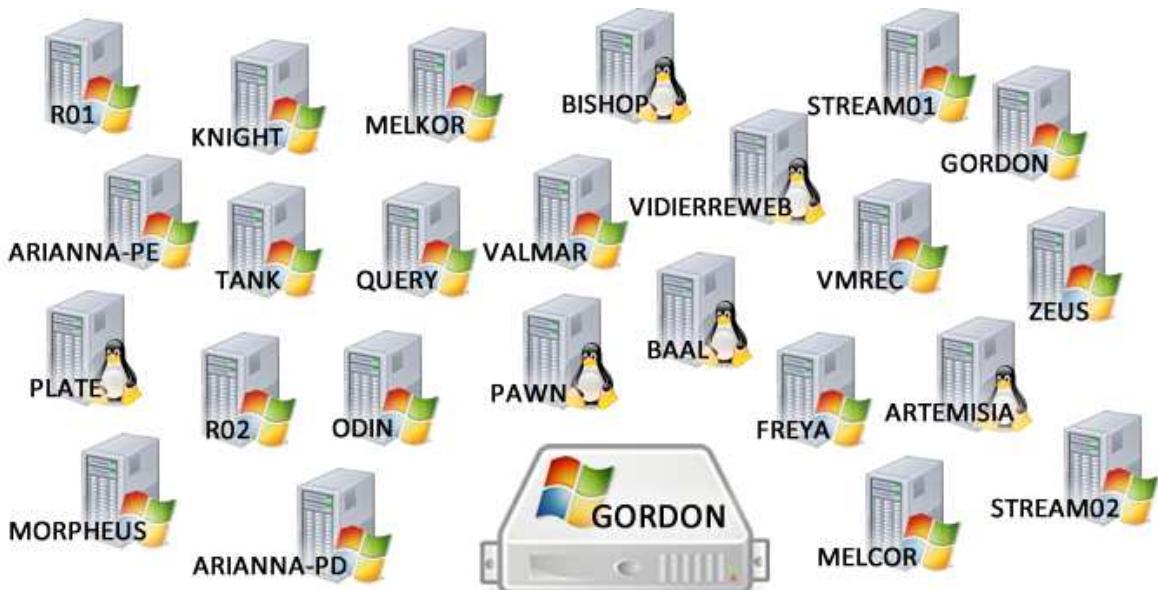


Figura 3.2: Struttura concettuale “VM-LESS” adottata sino al 2009 da VIDIERRE

Dal 2009, si è intrapreso un progetto di innovazione concettuale radicale e completa della struttura dell’intero sistema informativo aziendale. Le nuove strategie mirano ad un utilizzo profondo ed intenso della virtualizzazione, riducendo al minimo le macchine fisiche, senza per questo scendere a compromessi con capacità di calcolo e potenza di ciascuna macchina. Dal punto di vista economico, inoltre, è da considerare come, a parità di budget allocato negli anni, alla drastica diminuzione dei costi di manutenzione, di sostituzione componenti non più

¹ Si noti che nell’indicazione di “30 macchine fisiche”, o nella seguente di “5 macchine fisiche”, non si è tenuto conto dei dispositivi di storage e di altri server utili ai fini della core production di VIDIERRE, che nel presente progetto di tesi non sono stati coinvolti.

in grado di funzionare, e di ore lavorative spese a riparare frequenti danneggiamenti delle numerose macchine fisiche presenti, è corrisposta una più elevata disponibilità economica da riservare all'acquisto di nuovi server di ultima generazione (con architettura x86-64, ad esempio) che fungessero da host sempre più potenti per le varie VM guest. In figura 3.3 è possibile visualizzare il radicale cambiamento della situazione infrastrutturale in VIDIERRE con l'avvento della virtualizzazione.

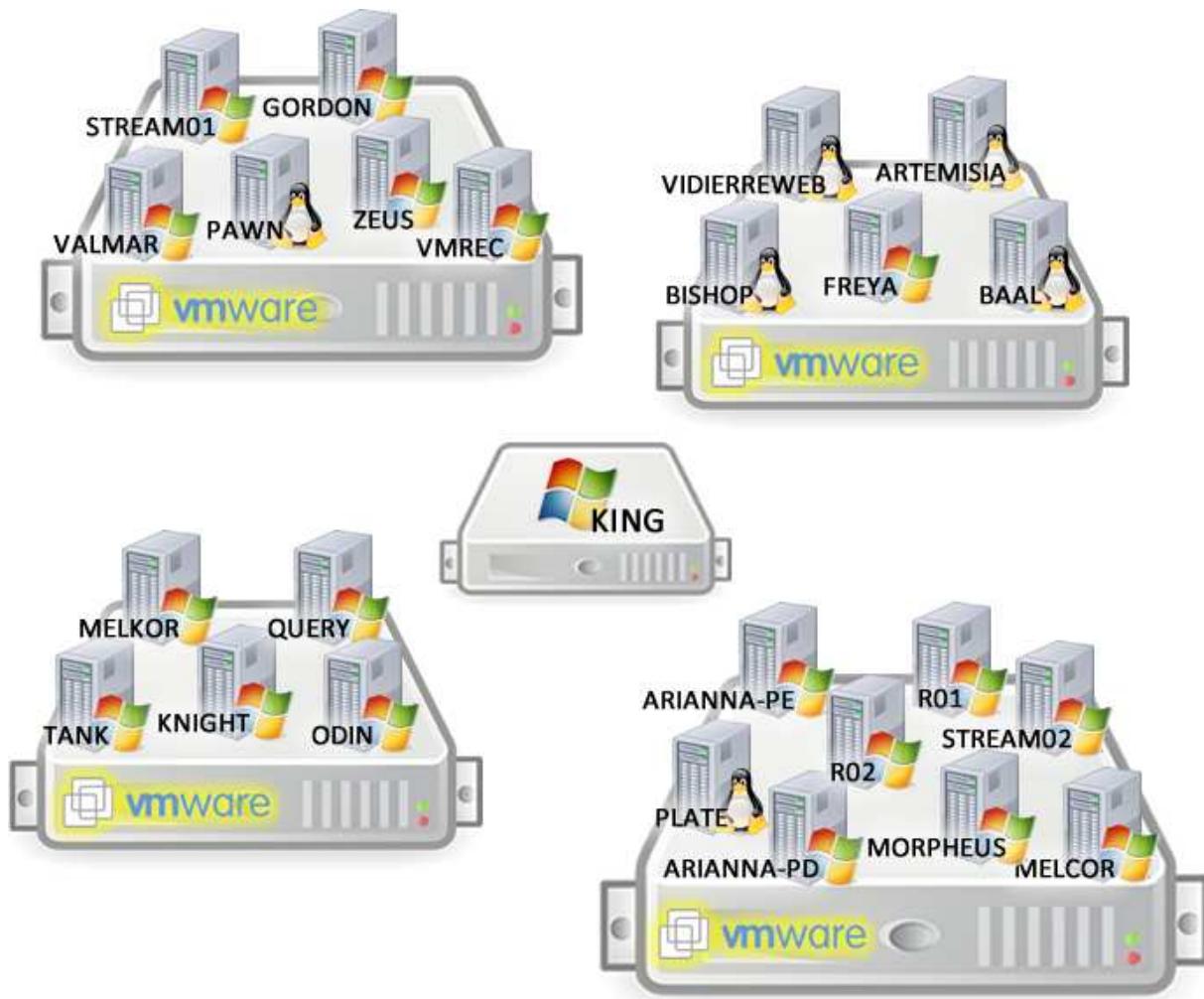


Figura 3.3: Struttura concettuale “VM-BASED” intrapresa da VIDIERRE

I server fisici, al momento, sono scesi al numero di cinque¹, su cui uno, *King*, è montato *SQL Server*, e gli altri quattro che, grazie a *VMware ESXi 4.1.0*, riescono ad ospitare numerose VM, anche con OS differenti (*Windows Advanced Server 2003*, *Windows Server 2000*, versioni *Ubuntu* e altri OS su base *UNIX*).

3.3 Installazione VM per test di compatibilità

Al momento in cui si è intrapreso il progetto di tesi, in VIDIERRE era presente un unico server in produzione, *Gordon*, su cui girava l’intera piattaforma dati basata su *Microsoft SQL Server 2000*. Il server che sarebbe stato denominato *King*, destinato a diventare il futuro unico server in produzione, al contrario era un host di VM su cui era già stata precedentemente installata una VM, chiamata *Queen*, che era l’esatta copia di *Gordon*, sia dal punto di vista hardware (identità architetturale e prestazionale rispetto al server *Gordon* fisico reale), sia dal punto di vista software. *Queen* è stato utilizzato per effettuare la totalità di test di compatibilità previsti nel progetto di tesi, e l’utilizzo di una VM ha evitato il possibile intralcio dell’attività lavorativa in qualsiasi momento ed occasione all’interno dei test, in quanto non è mai stato necessario basare le operazioni effettuate sul server in produzione.

Come prima procedura, si è installato il software *VMware VSphere Client*: si tratta di un applicativo client, appunto, incluso nella versione free utilizzata da VIDIERRE, che permette di accedere e comunicare con gli *Host Server*. Una volta installato il client, si è proceduto, dunque, all’installazione di una nuova macchina virtuale, chiamata *Knight*, avente le stesse caratteristiche hardware di *Queen* (e dunque di *Gordon*): in questo modo, la VM *Knight*, che andava ad ospitare il nuovo *Microsoft SQL Server 2008*, avrebbe garantito la piena uguaglianza di piattaforma in fase di test, ed avrebbe impedito che questi ultimi potessero essere “falsati” dall’esecuzione di procedure, sul *SQL Server 2008*, poggiandosi su una macchina più potente rispetto a quella dove era presente il *SQL Server 2000*.

Il processo di creazione di una nuova Virtual Machine si è rivelato parecchio intuitivo e scorrevole, grazie anche al wizard automatico presente nel software di installazione, che passo dopo passo ha agevolato le operazioni. Una delle numerose funzionalità che rendono all’avanguardia un prodotto come *VMware ESXi*, è la possibilità di scegliere, direttamente in fase di installazione, quale sistema operativo si andrà ad utilizzare sulla nuova VM che si sta configurando, permettendo la scelta tra tutti i sistemi operativi attualmente in commercio².

² Resta inteso che il prodotto *VMware ESXi* non comprende al suo interno i file di installazione dei sistemi operativi che permettono di selezionare in fase di configurazione/installazione di una nuova macchina virtuale. Per proseguire con l’installazione, dunque, è necessario possedere un supporto con all’interno il sistema operativo selezionato, oltre che ovviamente i necessari codici di attivazione e relativi Product Key (nel caso in esame).

Nel caso in esame, si è selezionato *Microsoft Windows Server 2003 Enterprise Edition 64bit*, poiché è il sistema operativo montato sulla maggior parte dei server su base Windows presenti in VIDIERRE, e perché è lo stesso sistema operativo previsto sul futuro *King*. In figura 3.4 è riportato uno screenshot di quanto appena esposto.

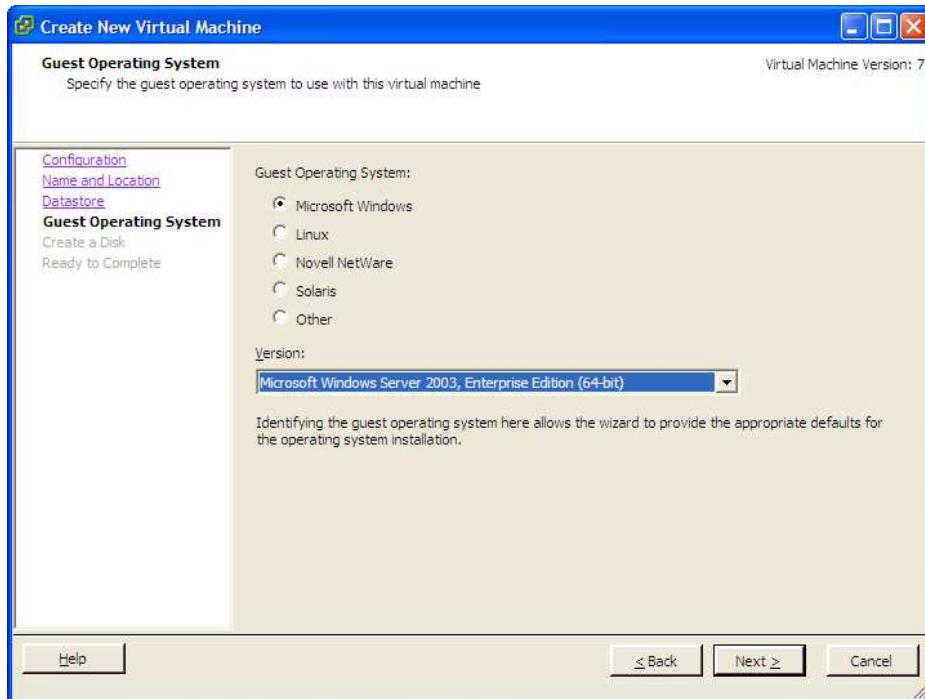


Figura 3.4: Scelta del Sistema Operativo in fase di installazione di una nuova VM

Durante l'installazione della nuova VM, il software *VMware ESXi* permette (ed obbliga) il settaggio di tutte le caratteristiche hardware che la nuova VM dovrà avere: memoria RAM, numero di CPU (core) che verranno utilizzate, socket VMCI (Virtual Machine Communication Interface), scheda video, unità CD/DVD/Floppy, unità HDD e loro grandezza³, controller SCSI e il Network Interface Controller (NIC). Nel nostro caso, sono state impostate sulla nuova VM (*Knight*) le stesse caratteristiche del server fisico *Gordon* su cui era in esecuzione il *SQL Server 2000*: 4 CPU, 16GB di RAM Memory, 50 GB di HDD primario e 250 GB di HDD secondario, un'unità CD/DVD ed un controller SCSI.

³ La capacità dell'unità HDD che si imposta, dimensiona conseguentemente il file .vmdk (Virtual Disk). Ciò significa che se al momento dell'installazione si imposta una qualsiasi unità disco con una capacità ad esempio di 100 GB, il software *VMware ESXi* crea un Virtual Disk, in un file .vmdk, che avrà esattamente quella dimensione, in quanto viene immediatamente allocata (e quindi sottratta alla reale disponibilità fisica n , che diventerà subito dopo $n-100 [GB]$) la quantità di spazio richiesta dal disco della nuova VM.

In figura 3.5 è visibile la finestra di configurazione delle proprietà “fisiche” della nuova VM in fase di installazione.

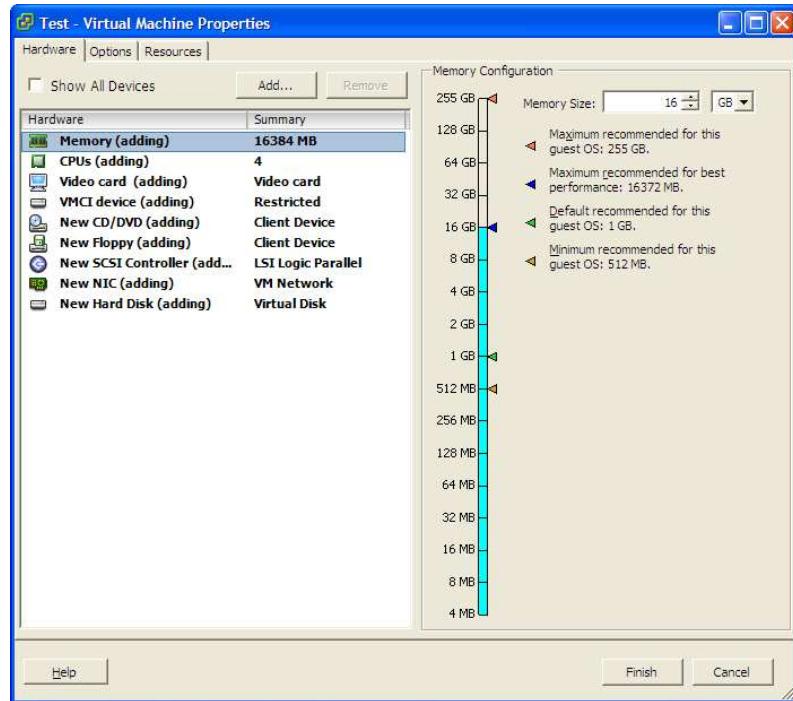


Figura 3.5: Scelta delle proprietà “fisiche” della nuova VM in fase di installazione

Un’altra particolare peculiarità del software *VMware ESXi*, è la possibilità di associare, direttamente in fase di installazione, un’immagine ISO all’unità disco ottico. In questo modo, la VM appena creata, al primo avvio, carica in primis la iso che “vede” nell’unità ottica. Nel nostro caso, si è associato il file ISO del sistema operativo *Microsoft Windows Server 2003 Enterprise Edition 64bit* che si andava ad installare, e non appena avviata la nuova VM, l’installazione del sistema operativo è stata lanciata in automatico.

3.4 Microsoft SQL Server Upgrade Advisor 2008

Microsoft, nell’ambito della migrazione tra due piattaforme diverse di SQL Server, mette a disposizione un potente tool, chiamato *SQL Server Upgrade Advisor*, che permette di simulare la migrazione tra differenti piattaforme evidenziando eventuali problematiche. Nel nostro caso, si è utilizzata la versione 2008. *SQL Server Upgrade Advisor* analizza istanze di SQL Server in preparazione per l’aggiornamento a *SQL Server 2008*. L’esecuzione di *SQL*

Server Upgrade Advisor consente di identificare le modifiche alla configurazione e alle funzionalità che potrebbero influire sull'aggiornamento e offre collegamenti a documentazione in cui sono descritti i problemi rilevati e le soluzioni disponibili [4]. Questo tool, sottolineato che non modifica in alcun modo i dati o le configurazioni delle istanze analizzate, ha permesso inoltre di riscontrare, per ciascun database in *SQL Server 2000*, quali modifiche e quali problemi sarebbero stati presenti una volta migrato lo stesso database sulla nuova piattaforma *SQL Server 2008*, segnalando per ciascuna problematica in quale momento essa dovesse essere affrontata e risolta (se prima o dopo la migrazione, o in qualsiasi momento). In figura 3.6 è possibile visualizzare il report finale generato dal *SQL Server Upgrade Advisor 2008* inerente al database “vidierre2k” in esecuzione su *Queen*.

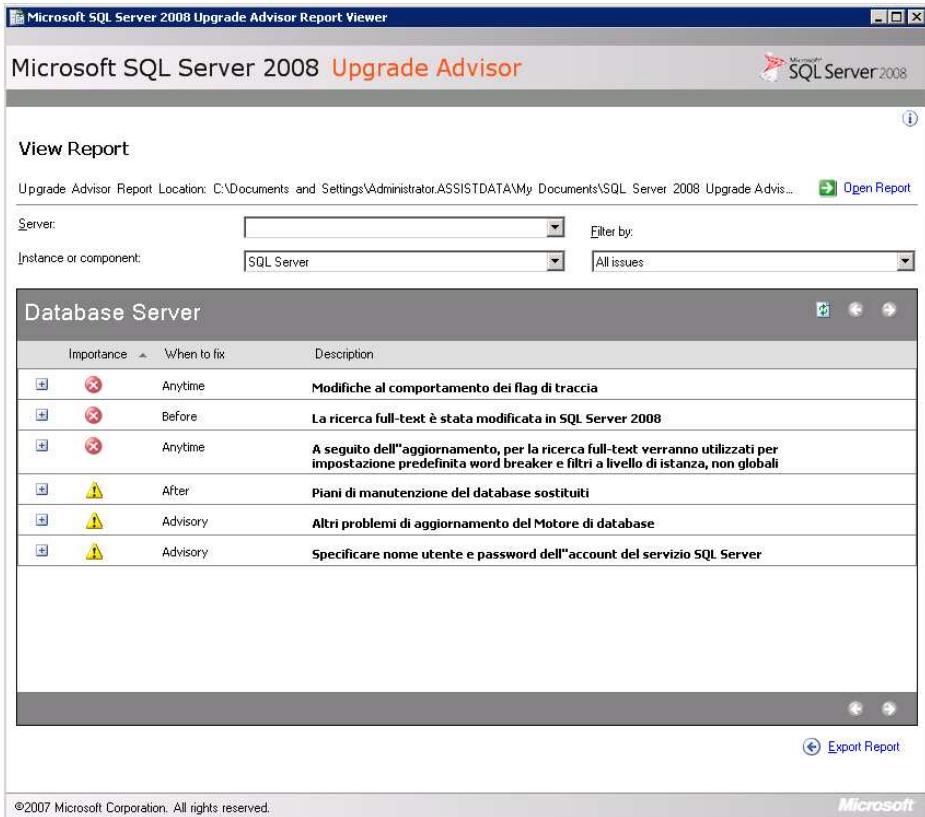


Figura 3.6: Report di aggiornamento di Microsoft SQL Server Upgrade Advisor 2008

Come si evince dalla figura 3.6, *Upgrade Advisor* ha segnalato alcuni problemi critici da risolvere prima della migrazione, ed altri avvisi di compatibilità senza l'indicazione di un momento definito per la loro risoluzione. Il report generato dal tool, riguardante il database

vidierre2k, ha denotato in maniera specifica le seguenti problematiche previste al momento della migrazione:

1. Modifiche al comportamento dei flag di traccia: in *SQL Server 2008*, i flag di traccia globali settati in una sessione, a differenza di *SQL Server 2000*, hanno effetto immediato. In particolare, Upgrade Advisor ha rilevato possibili problemi con il flag di traccia 1204, che restituisce le risorse e i tipi di blocco coinvolti in un deadlock nonché il comando corrente interessato; tuttavia, l'applicazione istantanea del setting per questo flag, si è valutata ininfluente ai fini del corretto funzionamento del database e dei software applicativi che vi si poggiano.
2. La ricerca full-text è stata modificata in SQL Server 2008: è stato rilevato che la ricerca full-text è soggetta ad aggiornamento, e molte opzioni e impostazioni della ricerca full-text sono state modificate. Questa problematica è stata giudicata ininfluente, come meglio specificato nel prossimo punto.
3. A seguito dell'aggiornamento, per la ricerca full-text saranno utilizzati per impostazione predefinita word breaker e filtri a livello di istanza, non globali: il tool ha fornito in aggiunta la seguente motivazione: “*SQL Server provides a way to allow instance-level registration of new word breakers and filters, Component: .css*”. Data l'assenza di qualsiasi componente .css all'interno di applicativi collegati al database in questione, si è valutata ininfluente questa problematica segnalata.
4. Piani di manutenzione del database sostituiti: il tool ha segnalato che, poiché la nuova funzionalità del piano di manutenzione del database non è una sostituzione diretta dei piani di manutenzione, dopo la migrazione sarebbe stato possibile che alcune funzionalità fossero andate perse. Anche questa problematica, nonostante la presenza di 15 piani di manutenzione già configurati ed installati sul *SQL Server 2000*, è stata giudicata affrontabile in un momento successivo: qualora nel processo di migrazione fossero venute meno alcune funzionalità dei suddetti piani, sarebbero stati ricreati ad hoc nella nuova piattaforma *SQL Server 2008*.
5. Altri problemi di aggiornamento del Motore di database: il possibile problema portato alla luce dal tool nel punto in questione riguarda l'AUTOGROW dei file dati e log. Il tool, in particolare, ha indicato: “*Upgrade Advisor detected data or log files that are not set to autogrow. In SQL Server 2008 and later, new and enhanced features require additional disk space for user databases and the tempdb system database. To ensure*

resources can accommodate size increases during upgrade and subsequent production operations, we recommend setting autogrow to ON for all user data and log files and the tempdb data and log files before upgrading to SQL Server 2008". Poiché il settaggio ad ON dell'AUTOGROW è in palese e forte contrasto con le politiche di gestione database di VIDIERRE (che adotta metodi di gestione ad hoc riguardo sia i file DB che i file di LOG), si è deciso di ignorare questa problematica, considerando "sconsigliabile" l'attuazione della risoluzione proposta dal tool. Qualora, al termine della migrazione, fossero sorti grossi problemi relativi al warning segnalato, si sarebbe proceduto all'attuazione di diverse strategie, benché sconsigliabili, come ad esempio il settaggio ad ON dell'AUTOGROW esclusivamente per la migrazione, e il successivo ritorno ad OFF al termine delle procedure.

6. Specificare nome utente e password dell'account del servizio SQL Server: per aggiornare ed eseguire il motore di ricerca full-text Microsoft per *SQL Server 2008*, è necessario immettere un nome account e una password associata a un gruppo di protezione specifico. Si tratta di una nuova funzionalità di protezione di *SQL Server 2008*: quando si esegue l'aggiornamento a *SQL Server 2008*, il motore di ricerca full-text Microsoft è eseguito con un nuovo account con privilegi di basso livello da associare a un gruppo di protezione specifico introdotto in questa versione. Questa problematica è stata giudicata facilmente risolvibile, semplicemente associando al nuovo account di esecuzione del motore di ricerca full-text il nome utente e la password di "sa", già in uso nell'istanza di *SQL Server 2000*, che non sarebbero variati all'atto della migrazione a *SQL Server 2008*.

Il tool *Microsoft SQL Server Upgrade Advisor 2008*, dunque, ha fornito numerose possibili problematiche che potrebbero essersi verificate al momento della migrazione della piattaforma dati. Esse sono state analizzate ed esaminate a fondo, con l'aiuto giudicato molto valido della documentazione on-line fornita da Microsoft stessa, oltre all'appoggio di discussioni presenti in numerosi forum in rete, per giungere alla conclusione che nessuna di queste avrebbe provocato criticità tali da mettere in discussione la prosecuzione del progetto di migrazione. L'esistenza di tale tool si rivela comunque molto importante, in quanto potrebbe portare a numerosi vantaggi derivanti dalla conoscenza a priori di problematiche di migrazione da affrontare e risolvere senza ipoteticamente influire in alcun modo sui tempi di produzione di un'Azienda.

3.5 Confronto tra le varie versioni di SQL Server 2008

La piattaforma dati presente in produzione in VIDIERRE, come già esposto, si basava sul *Microsoft SQL Server 2000 Enterprise Edition* che, al momento del suo acquisto ormai parecchi anni addietro, era considerata la versione più completa e potente del motore di casa Microsoft. Dovendo migrare ad una nuova edizione della stessa piattaforma, si è deciso di pesare la scelta della versione da acquistare, in base alle funzionalità incluse nelle varie versioni realmente necessarie in un'Azienda come VIDIERRE, ed al relativo costo del licensing che ciascuna di queste versioni prevede. Nella tabella seguente sono state riportate le voci considerate maggiormente importanti ai fini della valutazione delle differenze tra le due versioni a confronto.

	Enterprise	Standard
Numero di CPU	8	4
Massima memoria RAM utilizzabile	2 TB	64 GB
Massima dimensione DB	524 PB	524 PB
Numero di nodi di fail-over cluster	16	2
Backup in mirroring	✓	
Database snapshots	✓	
Recovery veloce	✓	
Pubblicazione dati da ORACLE a SQL Server	✓	
PowerPivot per Microsoft Excel	✓	

Come si evince dalla tabella, vi sono alcune funzionalità previste nella versione Enterprise ed assenti in quella Standard, come il backup in mirroring e soprattutto l'innovativa funzione del "database snapshot", cui VIDIERRE non avrebbe rinunciato. Per quanto riguarda il numero di CPU massime supportate dal *SQL Server 2008*, la versione Enterprise è abilitata a supportarne 8, a differenza delle 4 della versione Standard. Attenzione però ad una differenza sostanziale: in questo caso, come spiegatoci dall'account Microsoft che segue VIDIERRE, si parla di CPU fisiche, e non di core. Non vi è, dunque, alcuna limitazione dei core per ciascuna CPU. La situazione delle caratteristiche di ciascuna versione andava ben delineandosi. Bisogna oltretutto considerare che nella tabella precedente sono state evidenziate solo alcune delle differenze tra le due versioni, e cioè quelle che a giudizio di VIDIERRE sono state ritenute

più importanti. Benché fossero presenti le suddette differenze, è stato considerato anche che, tra le due versioni in esame, non vi era alcuna diversità tra le caratteristiche ritenute oltremodo fondamentali anche perché già utilizzate nella versione 2000 del SQL Server, tra le quali:

- i *Management Tools* (di cui fan parte, tra gli altri, il Database Engine Tuning Advisor, il SQL Server Profiler ed il SQL Server Agent);
- l'*Enterprise Security*, che prevede tra le altre funzioni la *Windows Integrated Authentication*, ossia lo sharing di account tra un dominio aziendale e gli utenti di SQL Server;
- i servizi di *database replication*, che permettono la replica di database attraverso una struttura gerarchica in un contesto multi-server;
- il *supporto per la virtualizzazione*, studiato da Microsoft per garantire la migliore integrazione possibile di tutti i servizi di SQL Server in piattaforme montate su sistemi virtualizzati;
- i *development tools*, che prevedono tra le numerose funzionalità la piena integrazione con Microsoft Visual Studio;
- i *servizi di programmabilità*, con un supporto nativo per l'XML;
- i *servizi di analisi* (ad esclusione delle Analytic functions, tra cui le aggregazioni finanziarie, non utili alla produttività di VIDIERRE);
- i *servizi di reporting*, da sempre considerati fondamentali dall'Azienda in fase di analisi a posteriori delle prestazioni e di risoluzione problematiche di vario tipo.

Dal punto di vista economico, la differenza dei costi di licensing tra la versione *Standard* e la versione *Enterprise* è notevole (circa il quadruplo). E' doverosa una precisazione: a differenza del Pricing orientativo che Microsoft pubblica online, l'acquisto delle licenze avviene tramite distributori ufficiali Microsoft che offrono alle Aziende soluzioni personalizzate in base alle configurazioni di ciascuna di esse (ad esempio, licenze per CPU piuttosto che User/Device CAL) e conseguentemente a prezzi riservati.

Alla luce di quanto detto, viste le necessità produttive ed operative di VIDIERRE, valutate le differenze non critiche tra le due versioni, e valutata soprattutto l'importante differenza economica messa in luce dal confronto, l'Azienda ha scelto di acquistare, nel caso la

migrazione fornisse risultati palesemente migliorativi in termini di tempo di calcolo e prestazioni, la versione *Standard* di *Microsoft SQL Server 2008*, anche perché, infine, il downgrade delle versioni (dalla precedente *SQL Server 2000 Enterprise* alla futura *SQL Server 2008 Standard*) non prevede una perdita di funzionalità precedentemente utilizzate, ma al contrario un’impossibilità di utilizzo di nuove funzionalità implementate nella nuova piattaforma (giudicate d’altro canto non utili al core business di VIDIERRE). Di conseguenza si può affermare che la versione *Standard* di *SQL Server 2008*, a livello funzionale, operativo e produttivo, è assimilabile alla versione *Enterprise* di *SQL Server 2000*, nel caso di un’Azienda reale quale VIDIERRE.

3.6 Installazione e configurazione della versione scelta

Una volta compiute le dovute valutazioni del caso sulla scelta della versione di *Microsoft SQL Server 2008* da utilizzare, cioè la versione *Standard*, si è proceduto alla sua installazione sulla VM *Knight*. Il pacchetto di installazione si presenta come un “*Installation Center*” in autorun, da cui è possibile accedere a tutte le funzionalità di pre-installazione (vedi *SQL Server Upgrade Advisor 2008*), installazione, manutenzione ed utilizzo di tool vari (come il *System Configuration Checker*). In figura 3.7 si riporta la schermata iniziale dell’*Installation Center*.

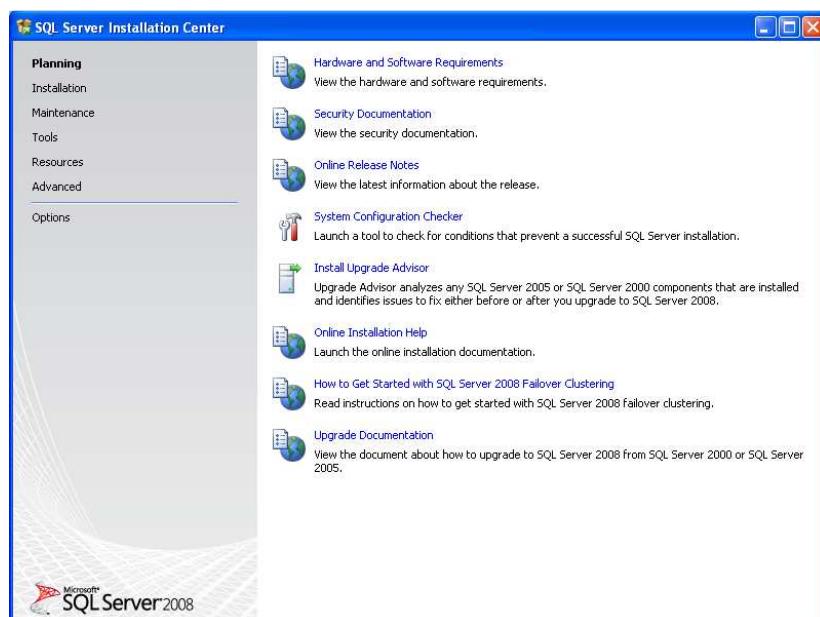


Figura 3.6: Report di aggiornamento di Microsoft SQL Server Upgrade Advisor 2008

Nel nostro caso, data l'intera attività di “pre-installazione” già compiuta, anche attraverso l'utilizzo del *SQL Server Upgrade Advisor 2008* (Cfr. Par. 3.4), si è proceduto direttamente ad una nuova “SQL Server stand-alone installation”. In figura 3.7 è riportato uno screenshot subito antecedente l'installazione finale del *SQL Server 2008* in cui è riportato il riepilogo di tutte le funzionalità che ci si appresta ad installare e le varie configurazioni richieste dall'Installation Center per la nuova istanza di SQL Server.

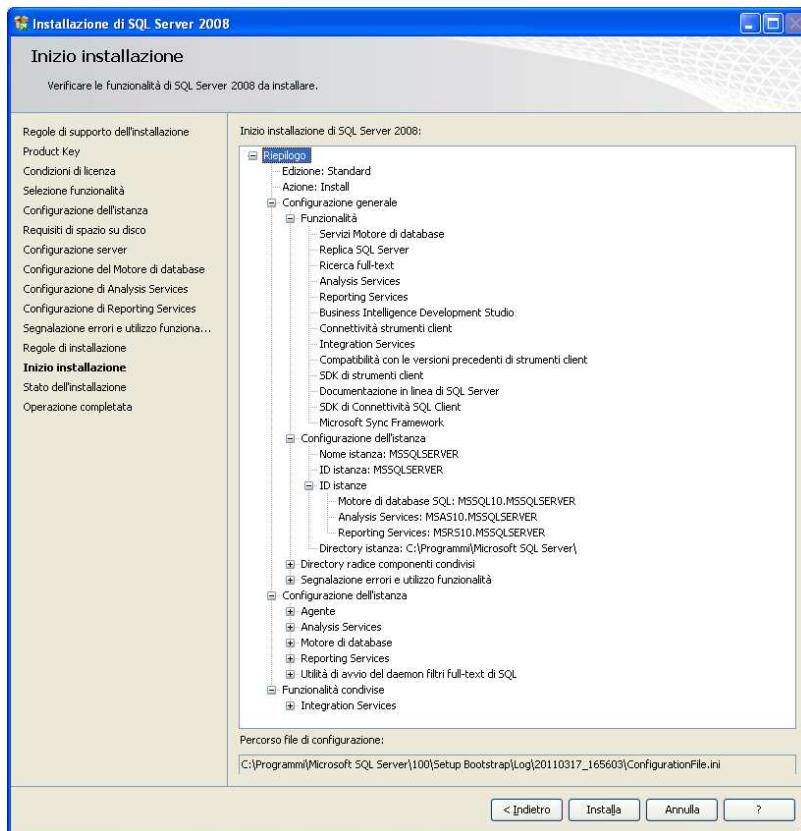


Figura 3.7: Riepilogo caratteristiche e configurazioni di installazione nella nuova istanza

Una piccola “imperfezione” del *SQL Server 2008* che balza subito all’occhio, è l’assenza, nella system tray di Windows, dell’icona del servizio **MSSQLSERVER** già presente in tutte le precedenti edizioni della piattaforma di casa Microsoft, molto utile in quanto spia real-time dello status del servizio stesso. Per ovviare a questa “mancanza”, ci sono dei software esterni che simulano l’icona, ma si è sperimentalmente appurato che, tali software, incredibilmente, rendono instabile il sistema operativo: ciò ha dunque comportato la rinuncia alla presenza della suddetta icona. Nel listato seguente, si possono evincere tutte le impostazioni di configurazione adottate per *SQL Server 2008* appena installato.

Overall summary:
 Final result: operazione riuscita
 Exit code (Decimal): 0
 Exit message: Operazione riuscita
 Requested action: Install

Machine Properties:
 Machine name: KNIGHT
 Machine processor count: 4
 OS version: Windows Server 2003 Enterprise Edition
 OS service pack: Service Pack 3
 OS region: Italia
 OS language: italiano (Italia)
 OS architecture: x86
 Process architecture: 64 Bit
 OS clustered: No

Product features discovered:
 Product Edition: SQL Server 2008 Standard Edition
 Instance Version: 10.0.1600.22
 Instance ID Clustered: NO
 Feature: Strumenti di gestione - Di base
 Language: 1040

Package properties:
 Description: SQL Server Database Services 2008
 SQLProductFamilyCode: {628F8F38-600E-493D-9946-F4178F20A8A9}
 ProductName: SQL2008
 Type: RTM
 Version: 10
 SPLevel: 0
 Installation location: E:\x86\setup\
 Installation edition: STANDARD

Impostazioni input utente:
 ACTION: Install
 ADDCURRENTUSERASQLADMIN: False
 AGTSVCACCOUNT: [REDACTED]
 AGTSVCPASSWORD: [REDACTED]
 AGTVCSTARTUPTYPE: Manual
 ASBACKUPDIR: C:\Programmi\Microsoft SQL Server\MSAS10.MSSQLSERVER\OLAP\Backup
 ASCOLLATION: Latin1_General_CI_AS
 ASCONFIGDIR: C:\Programmi\Microsoft SQL Server\MSAS10.MSSQLSERVER\OLAP\Config
 ASDATADIR: C:\Programmi\Microsoft SQL Server\MSAS10.MSSQLSERVER\OLAP\Data
 ASDOMAINGROUP: <vuoto>
 ASLOGDIR: C:\Programmi\Microsoft SQL Server\MSAS10.MSSQLSERVER\OLAP\Log
 ASSPROVIDERSOLAP: 1
 ASSVCACCOUNT: [REDACTED]
 ASSVCPASSWORD: [REDACTED]
 ASSVCSTARTUPTYPE: Automatic
 ASSYADMINACCOUNTS: C:\Programmi\Microsoft SQL Server\MSAS10.MSSQLSERVER\OLAP\Temp
 ASTEMPDIR: Disabiled
 BROWSERSVCSTARTUPTYPE: False
 ENablerANU: False
 ERRORREPORTING: False
 FEATURES: SQLENGINE, REPLICATION, FULLTEXT, AS, RS, BIDS, CONN, IS, BC, SDK, BOL, SNAC_SDK, OCS
 FILESTREAMLEVEL: 0
 FILESTREAMSHARENAME: <vuoto>
 FT SVCACCOUNT: <vuoto>
 FT SVCPASSWORD: [REDACTED]
 HELP: False
 INDICATEPROGRESS: False
 INSTALLSHAREDDIR: c:\Programmi\Microsoft SQL Server\
 INSTALLSHAREDWOWDIR: C:\Programmi\Microsoft SQL Server\
 INSTALLSQLDATADIR: <vuoto>
 INSTANCEDIR: C:\Programmi\Microsoft SQL Server\
 INSTANCEID: MSSQLSERVER
 INSTANCENAME: MSSQLSERVER
 ISSVCACCOUNT: [REDACTED]
 ISSVCPASSWORD: [REDACTED]
 ISSVCSTARTUPTYPE: Automatic
 MEDIASOURCE: E:\
 NPENABLED: 0
 PID: [REDACTED]
 QUIET: False
 QUIETSIMPLE: False
 RSINSTALLMODE: DefaultNativeMode
 RSSVCACCOUNT: [REDACTED]
 RSSVCPASSWORD: [REDACTED]
 RSSVCSTARTUPTYPE: Automatic
 SAPWD: <vuoto>
 SECURITYMODE: <vuoto>
 SQLBACKUPDIR: <vuoto>
 SQLCOLLATION: Latin1_General_CI_AS
 SQLSVCAccount: [REDACTED]
 SQLSVCPASSWORD: [REDACTED]
 SQLVCSTARTUPTYPE: Automatic
 SQLSYSADMINACCOUNTS: <vuoto>
 SQLTEMPDBDIR: <vuoto>
 SQLTEMPDBLOGDIR: <vuoto>
 SQLUSERDBDIR: [REDACTED]
 SQLUSERDBLOGDIR: [REDACTED]
 SQMREPORTING: False
 TCPENABLED: 1

Configuration file: C:\Programmi\Microsoft SQL Server\100\Setup Bootstrap\Log\\ConfigurationFile.ini

Detailed results:	
Feature:	Servizi Motore di database
Status:	Operazione riuscita
Feature:	SDK di Connattività SQL Client
Status:	Operazione riuscita
Feature:	Replica SQL Server
Status:	Operazione riuscita
Feature:	Ricerca full-text
Status:	Operazione riuscita
Feature:	Analysis Services
Status:	Operazione riuscita
Feature:	Reporting Services
Status:	Operazione riuscita
Feature:	Integration Services
Status:	Operazione riuscita
Feature:	Connattività strumenti client
Status:	Operazione riuscita
Feature:	SDK di strumenti client
Status:	Operazione riuscita
Feature:	Compatibilità con le versioni precedenti di strumenti client
Status:	Operazione riuscita
Feature:	Business Intelligence Development Studio
Status:	Operazione riuscita
Feature:	Documentazione in linea di SQL Server
Status:	Operazione riuscita
Feature:	Microsoft Sync Framework
Status:	Operazione riuscita
Rules with failures:	0

3.7 Importazione database da SQL Server 2000

Al termine dell'installazione del nuovo *SQL Server 2008*, e della conseguente configurazione della nuova istanza, si è proceduto con il trasferimento dei database dalla vecchia piattaforma *SQL Server 2000* alla nuova *SQL Server 2008*. Le metodologie di importazione che potevano essere intraprese erano due: la prima consisteva nell'utilizzo della procedura automatica, presente nella suite di tools di *SQL Server 2008*, chiamata “*SQL Server Import and Export Wizard*” presente nelle due versioni a 32 e 64 bit; la seconda consisteva invece nell'effettuazione di un backup di ogni singolo database all'interno del *SQL Server 2000*, e il successivo restore del suddetto database in *SQL Server 2008*, il tutto attraverso procedure automatizzate presenti in entrambe le piattaforme.

Si è proceduto dunque in primis ad importare i database attraverso il *SQL Server Import and Export Wizard* agendo direttamente dal *SQL Server 2008* appena installato sulla VM *Knight*. Il wizard è davvero intuitivo ed abbastanza potente. Dopo aver specificato *Queen* come server sorgente di *SQL Server 2000* e dopo aver selezionato il database *vidierre2k* come database da importare, il wizard ha richiesto il server di destinazione (impostando di default l'istanza di *SQL Server 2008* appena creata su *Knight*) ed il database di destinazione. Si è scelto, dunque,

di creare contestualmente all’importazione un nuovo database, target del trasferimento dati, che sarebbe stata nel nuovo *SQL Server 2008* la copia esatta del database sorgente. Una volta avviata la procedura, tuttavia, il wizard richiedeva di selezionare le tabelle e le viste del database sorgente da importare. Purtroppo, il wizard si è rivelato non in grado di “leggere” il database sorgente, in quanto ha restituito come messaggio di errore “*The source database you have selected contains no visible table or views. Please go back to the Choose a Data Source page and choose a database that has user tables or views*”. In figura 3.8 è riportato lo screenshot dell’errore in questione.



Figura 3.8: Errore in lettura dal database sorgente del wizard di importazione dati

Lo stesso errore era restituito usando entrambe le versioni disponibili del wizard di importazione, sia a 32bit, sia a 64bit. In step successivi, a migrazioni effettuate, si è riprovato ad utilizzare il tool, selezionando come sorgente lo stesso database ma su un server con piattaforma *SQL Server 2008*, e l’errore in questione non è stato più presente. Da ciò si deduce che, purtroppo, il wizard fornito da Microsoft per l’importazione di dati non è stato utile alla causa del progetto di tesi, poiché si è denotata l’impossibilità di comunicazione a livello di contenuti di un database con un server su cui è montato il *SQL Server 2000*. Si è proceduto, dunque, all’utilizzo della seconda metodologia.

3.7.1 Backup e Restore dei database

Come già indicato, la seconda metodologia prevista per la migrazione dei database consiste nell'effettuazione di un backup di ogni singolo database all'interno del *SQL Server 2000*, e il successivo restore di ciascun database in *SQL Server 2008*. Anche in questo caso, l'intera procedura è facilmente eseguibile attraverso l'utilizzo di tool inclusi in qualsiasi piattaforma e versione di SQL Server si vada ad utilizzare. Nel caso in questione, si è proceduto al backup su *Queen*, in *SQL Server 2000*, del database *vidierre2k*. La procedura è raggiungibile dal menù a tendina cliccando con il tasto destro del mouse su un qualsiasi database, e selezionando successivamente *All Tasks* ed infine *Backup Database*. Nella finestra di impostazione del backup, visibile in figura 3.9, si può scegliere il tipo di backup da effettuare (completo, differenziale, esclusivamente del transaction log o di un filegroup), il file di destinazione del backup (attenzione, il risultato finale non sarà un database, ma un file unico contenente dati, log, impostazioni e configurazioni), la possibilità di “append” ad un database esistente o sovrascrittura del file selezionato ed infine la possibilità di scheduling dell'esecuzione procedura (in questo modo si crea automaticamente un Job, ad esecuzione unica, gestito direttamente da *SQL Server Agent*)

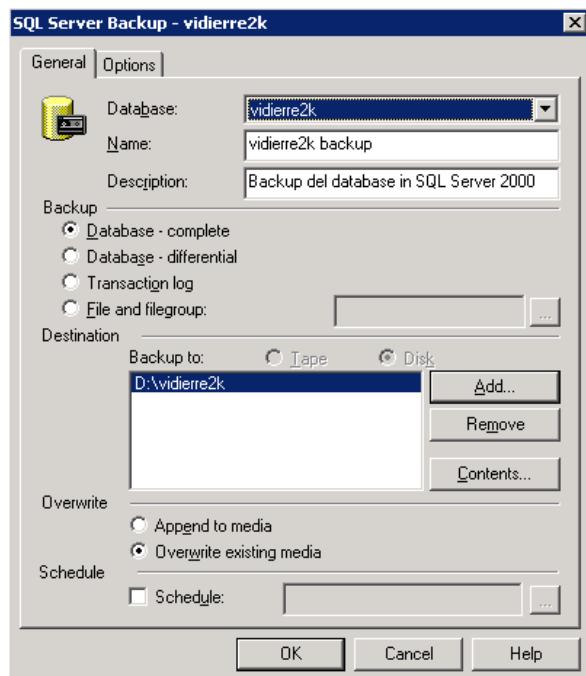


Figura 3.9: Finestra di impostazione del backup di un database in *SQL Server 2000*

Al termine della procedura automatica, si è spostato il file creato da *D:\vidierre2k* al server di destinazione *Knight*, in modo da non dover eseguire il restore del database “al volo” (partendo da un file presente su un’altra macchina in LAN), ma preferendo dunque la presenza del file stesso in locale su *Knight*.

Si è proceduto dunque all’attivazione su *SQL Server 2008* della procedura guidata per il restore dei database di cui si possiede un backup. Tale procedura, dopo aver selezionato il backup “sorgente” da ripristinare, permette l’impostazioni di diversi valori, tra i quali il database di destinazione (che è stato nominato *vidierre2k*), il path di destinazione dei file dati MDF e log LDF, e non ultima la possibilità di scegliere il metodo di restore tra:

- **RESTORE WITH RECOVERY**: lascia il database pronto per l’utilizzo eseguendo il rollback delle transazioni di cui non è stato eseguito il commit;
- **RESTORE WITH NORECOVERY**: lascia il database non operativo, e non esegue il rollback delle transazioni di cui non è stato eseguito il commit;
- **RESTORE WITH STANDBY**: lascia il database in modalità sola lettura, annulla le transazioni di cui non è stato eseguito il commit e salva le azioni di rollback in un file standby in modo che gli effetti del recupero possano essere annullati.

Tra le tre opzioni elencate, si è scelta la prima (RESTORE WITH RECOVERY), in quanto si è appurata l’inesistenza di transazioni di cui non è stato eseguito il commit, e la volontà di avere il database subito operativo nella nuova piattaforma di destinazione. In figura 3.10 è riportato lo screenshot del termine procedura di restore effettuato con successo.

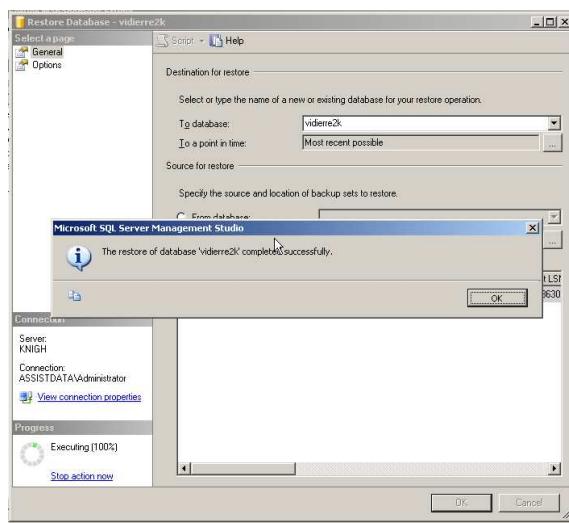


Figura 3.10: Termine della procedura di restore database

Al termine della procedura, svoltasi senza alcun intoppo, il database *vidierre2k* era presente nell'elenco dei database su *SQL Server 2008*, in stato “Attached” e “Online”. La metodologia di migrazione database rivelatasi più efficace, veloce (tempo totale richiesto: 25 minuti per il backup e 20 minuti per il restore) e sicura è stata senza dubbio quella che prevede l'utilizzo di uno strumento, quale il backup/restore, disponibile in ogni piattaforma e capace di creare file leggibili in qualsiasi versione ed edizione, al momento in cui si scrive, di Microsoft SQL Server.

3.8 Comparazione strutturale e prestazionale

Con la presenza del database *vidierre2k* in funzione sulla nuova piattaforma *SQL Server 2008*, si è potuto procedere ad alcuni test, per verificare la perfetta corrispondenza strutturale tra la versione dello stesso database, in produzione, sul *SQL Server 2000* e quella da poco migrata sulla nuova piattaforma. Inoltre si è potuto procedere con comparazioni prestazionali circa l'esecuzione e i tempi di risposta di alcune operazioni su entrambi i database, per poi poterle analizzare e verificarne l'effettivo miglioramento a livello di velocità di calcolo della nuova piattaforma. I succitati test sono stati eseguiti con l'ausilio di software ad hoc, come il *SQL Server Comparison Tool*, attraverso l'uso di query molto complesse ed infine attraverso il beta testing di software applicativi proprietari, eseguendo le operazioni che sulla base dell'esperienza impiegano diversi minuti ad essere completate.

3.8.1 SQL Server Comparison Tool

SQL Server Comparison Tool è un software della AlfaAlfa Software, liberamente scaricabile dal sito internet della società nella sua versione shareware. Questa versione è completamente funzionante, senza alcun tipo di limitazione, per 30 giorni dalla data di prima installazione. Passato il periodo di prova, si può acquistare con diversi tipi di licensing.

Il *SQL Server Comparison Tool* (SCT) è un programma Windows per l'analisi, la comparazione e la documentativa sui database in SQL Server. Con il *SQL Server Comparison Tool* è possibile effettuare la comparazione e l'analisi di database presenti su differenti server, come nel caso del presente progetto di tesi, in una modalità “read-only” che assicura una completa sicurezza circa l'integrità dei dati analizzati [5]. Il software si presenta molto intuitivo e user-friendly, e permette la comparazione dell'intero database, oppure solo di

alcune tabelle, stored procedures, o dati selezionati. In figura 3.11 è visibile la finestra principale del *SQL Server Comparison Tool* in cui selezionare le opzioni di comparazione ed i server/database da comparare.

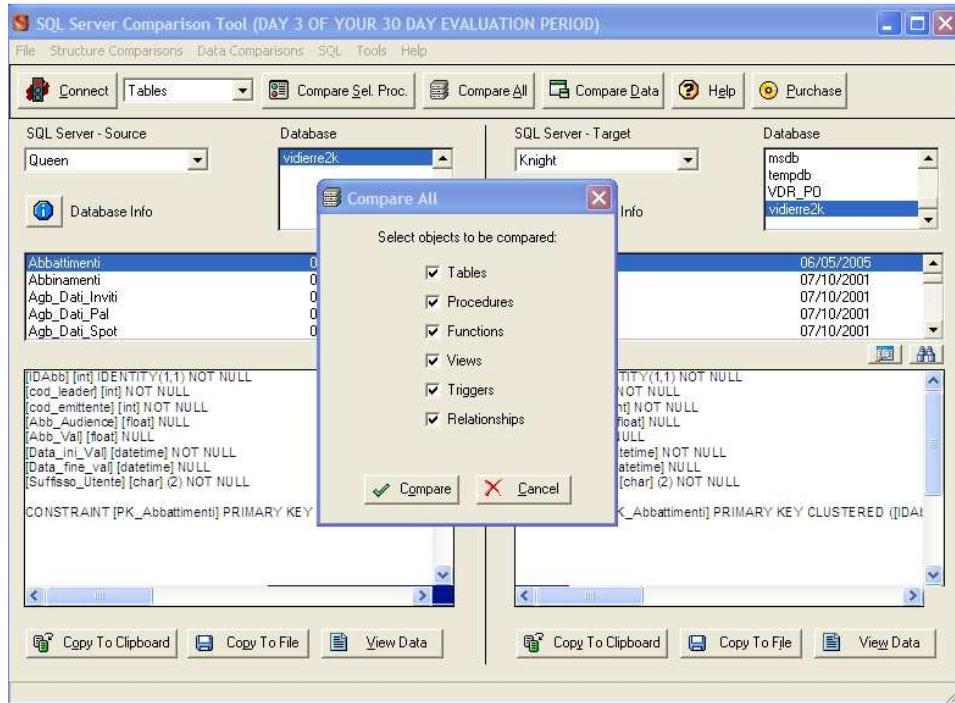


Figura 3.11: Finestra principale di SQL Server Comparison Tool

Al termine della procedura, il software ha restituito un report molto dettagliato sui risultati della comparazione dei due database eseguita su tabelle, procedure, funzioni, viste, triggers e relazioni. Di seguito si è riportata copia del report stesso

```

Comparison of databases
Source database Queen.vidierre2k
Target database Knight.vidierre2k

Comparison of tables
There are 125 tables in Source database.
There are 125 tables in Target database.
List of tables which exist in Target database but not in Source database:
[REDACTED]
(2 tables)
All 123 remaining tables have identical structures:
Abbattimenti
Abbinamenti
Agb_Dati_Inviti
Agb_Dati_Pal
Agb_Dati_Spot
Agb_Desc
Agb_Inviti
Agb_Inviti_Target
Agb_Ottimale
Agb_Pal
Agb_Pal_Ext
Agb_Pal_Target

```

Agb_Sponsor
Agb_Sponsor_Target
Agb_Spot
Agb_Spot_Target
Argomenti
Argomenti_Marchio
Attivita
CancellareTemp
Circuiti
Claposcan
Classi
[REDACTED]
[REDACTED]
[REDACTED]
Dati_Chiusi
Dati_Rilev
Dati_Rilev_Storico
Dati_Rilev_Tipo
dicituraprosp
DicituraProsp_OLD
Dimensioni
dtproperties
[REDACTED]
Emittenti_Arianna
Emittenti_Gruppo
Emittenti_Nomi
Emittenti_Periodi
Emittenti_Viti
Emittenti_Viti_Pal
Emittenti_Zone
[REDACTED]
Fasce_Apparizione
Fasce_Emittente
Fasce_Emittente_Periodi
Fasce_Emittente_Periodi_old
Fasce_Grafici
Fasce_Orarie
Fasce_Publitalia
FolliniPalinsesto
Giornate
Gruppi
Leader_Disiplina
Leader_Emittenti
Leader_Marchi
Leader_sports
Listini_spot
Loc_Ottimale
Localita
Log_op
LogLogon
Mapping_Copie
[REDACTED]
[REDACTED]
Moltiplicatori_Marchi
Moltiplicatori_pos
Naz_Ottimale
Parametri
Periodi_Chiusi
Posizion_Marchio
Posizionamenti
PosizionScanner
PostIt
[REDACTED]
Rapporto_target
Regioni
RIL_CRISTIAN
Rilev_abattute
Rilevazioni_Ext
Rilevazioni_Split
Rilevazioni_Tempi
Rilevazioni_Tempi_Offset
Sat_Ottimale
[REDACTED]
[REDACTED]
[REDACTED]
Sicurezza
[REDACTED]
Sponsor_Desc
Sponsor_Grup
Sponsor_Nome
Sponsor_Old
Sponsor_Ord
Sports

```

Storico
Storico_Rilev
Storico_Rilev_Tempi
t_e
t_fe
t_feo_a
t_r
target_ord
temp_CP30
Temp_pal_arianna
TempTrasfTim
Tipologie
tmp_RetroAttivo
tondelli_vega
[REDACTED]
Trasmissioni_Emissente
Trasmissioni_Emissente_Target
Trasmissioni_Tipologie
TrasmissTemp
ucci
[REDACTED]
universo_target
universo_target_localita
Utenti
validitaMarchi
VdrLog
visione_Oператори

Comparison of procedures
There are 106 procedures in Source database.
There are 106 procedures in Target database.
All procedures in both databases are identical.

Comparison of functions
There are 14 functions in Source database.
There are 14 functions in Target database.
All functions in both databases are identical.

Comparison of views
There are 28 views in Source database.
There are 26 views in Target database.
List of views which exist in Source database but not in Target database:
    sysconstraints
    syssegments
    (2 views)
All views in both databases are identical.

Comparison of triggers
There are 55 triggers in Source database.
There are 55 triggers in Target database.
All triggers in both databases are identical.

Comparison of relationships
There are 4 relationships in Source database.
There are 4 relationships in Target database.
All relationships in both databases are identical.

Time Elapsed: 00:08:35

```

La procedura di comparazione ha riportato le seguenti differenze tra i due database analizzati:

- Due tavole, presenti nel database in *SQL Server 2008*, erano invece assenti nel database in *SQL Server 2000*. In realtà, durante la migrazione, il nome di queste due tavole, ipotizzandolo in TABELLA sul database di origine, è stato rinominato in dbo.TABELLA nel database di destinazione. La struttura ed i dati relativi alle presenti tavole sono rimasti invariati, ma non è stato possibile trovare una spiegazione a tale cambio di nome avvenuto durante la migrazione.

- Nel database in *SQL Server 2008* non esistono 2 viste di sistema presenti nello stesso database su piattaforma *SQL Server 2000*, e cioè sysconstraints e syssegments. Come si apprende da Microsoft, “*Queste tabelle di sistema di SQL Server 2000 sono incluse come viste per una compatibilità con le versioni precedenti. E' raccomandabile l'utilizzo delle viste di sistema correntemente installate sulla vostra versione di SQL Server. Questa funzionalità sarà rimossa in una futura versione di Microsoft SQL Server. Evitate dunque di usare questa funzionalità in nuovi progetti di sviluppo, e pianificate la modifica delle applicazioni che al momento fanno uso di questa funzionalità*” [6].

Per quanto riguarda il resto degli elementi dei due database sottoposti a comparazione (tra cui anche i dati), non è stata rilevata dal *SQL Server Comparison Tool* alcuna differenza. Ne consegue l'assoluta certezza che il processo di migrazione del database a *Microsoft SQL Server 2008*, avvenuto attraverso la modalità di backup/restore (Cfr. Par. 3.7.1), non ha comportato perdite di dati o di funzionalità in alcuna tabella/stored procedure/vista al suo interno.

3.8.2 Valutazione prestazionale con query test

Dopo aver appurato la consistenza dei dati all'interno del nuovo database su *SQL Server 2008*, ed aver verificato l'identità strutturale rispetto allo stesso database sorgente di *SQL Server 2000*, si è proceduto all'effettuazione di un'analisi prestazionale attraverso l'utilizzo di una query molto complessa, che tocca gran parte delle tabelle e delle stored procedures presenti nel database *vidierre2k*. La query utilizzata, cui segue spiegazione circa la sua funzione, è la seguente:

```
SET STATISTICS TIME ON

DECLARE @start_time DATETIME
SELECT @start_time = GETDATE()

select t4.*, utl.individui, ad.desc_target, l.nome_localita,
tgtord.cod_ord, sn.desc_sponsor, so.ord_sponsor, e.indic_estere From (
SELECT chiave, cod_leader, nome_leader, cod_marchio, nome_marchio,
alias_marchio, cod_macroarg, desc_macroarg, cod_argomento, desc_argomento,
cod_circuito, cod_emittente, nome_emittente, prog_ord, cod_trasmis,
nome_trasmis, id_pal_agb, data_trasmis, ora_inizio_trasmis,
durata_trasmis, durata_servizio, prog_inizio_tape, prog_fine_tape,
```

```

numero_tape_1, numero_tape_2, nome_servizio, categ_servizio,
tipo_servizio, audience_effet, numero_servizi, numero_rilev, tempo_totale,
convert(float, valutazione_totale)/1936.27 as valutazione_totale,
indic_circuito, indic_tv_satellite, indic_agb, indic_nazionale,
flag_stato, cod_pos, sigla_pos, CASE WHEN cod_localita is null THEN 135
ELSE cod_localita END as cod_localita, CASE WHEN TET_AUDM IS NULL THEN
APT_AUDM ELSE TET_AUDM END As AUDIENCE_MEDIO, CASE WHEN TET_CONT IS NULL
THEN APT_CONT ELSE TET_CONT END As contatti, CASE WHEN tet_pen is null
THEN apt_pen ELSE tet_pen END As Penetrazione, CASE WHEN TET_TARGET IS
NULL THEN apt_target ELSE tet_target END As id_target, case WHEN
tet_share is null THEN apt_share ELSE tet_share END as share From (
SELECT
TT.* , TET.AUDIENCE_MEDIO AS TET_AUDM, TET.CONTATTI AS TET_CONT, TET.ID_TARGET
AS TET_TARGET, tet.penetrazione as tet_pen, tet.share as tet_share From (
SELECT chiave, cod_leader, nome_leader, cod_marchio, nome_marchio,
alias_marchio, cod_macroarg, desc_macroarg, cod_argomento, desc_argomento,
cod_circuito, R.cod_emittente, R.nome_emittente, R.prog_ord, cod_trasmis,
nome_trasmis, R.id_pal_agb, data_trasmis, ora_inizio_trasmis,
durata_trasmis, case WHEN durata_servizio=0 THEN 1 ELSE durata_servizio END
as durata_servizio, prog_inizio_tape, prog_fine_tape, numero_tape_1,
numero_tape_2, nome_servizio, categ_servizio, tipo_servizio,
audience_effet, numero_servizi, SUM(numero_rilev) AS NUMERO_RILEV,
SUM(tempo_totale) AS TEMPO_TOTALE, SUM(valutazione_totale) AS
VALUTAZIONE_TOTALE, R.indic_circuito, R.indic_tv_satellite, R.indic_agb,
R.indic_nazionale, flag_stato, cod_pos, sigla_pos, APT.AUDIENCE_MEDIO AS
APT_AUDM, APT.CONTATTI AS APT_CONT, APT.ID_TARGET AS
APT_TARGET, apt.penetrazione as apt_pen, apt.share as apt_share,
E.COD_LOCALITA From (
select * From dati_rilev (nolock) WHERE
cod_leader=1668 and chiave='T' and data_trasmis between '05/01/2010' and
'05/31/2010' and indic_circuito=0 and not (cod_marchio=1904 and
cod_macroarg=3417 and cod_argomento=750 and sigla_pos='0') and not
...
...
... omissis per sensibilità dati aziendali ...
...
and not (cod_marchio=146 and cod_macroarg=3417 and cod_argomento=753 and
sigla_pos='PATCH MAGLIA') and not (cod_marchio=436 and cod_macroarg=3417
and cod_argomento=753 and sigla_pos='PALLONE') and not (cod_marchio=146
and cod_macroarg=3417 and cod_argomento=2108 and sigla_pos='4° UOMO') and
not (cod_marchio=146 and cod_macroarg=3417 and cod_argomento=2108 and
sigla_pos='CAMPO') and not (cod_marchio=146 and cod_macroarg=3417 and
cod_argomento=2108 and sigla_pos='INTERVISTE TIM') and not (
cod_marchio=146 and cod_macroarg=3417 and cod_argomento=2108 and
sigla_pos='SIGLA FINALE') and not (cod_marchio=146 and cod_macroarg=3417
and cod_argomento=2108 and sigla_pos='SIGLA INIZIALE') and not (
cod_marchio=146 and cod_macroarg=3417 and cod_argomento=2108 and
sigla_pos='SIGLE') and not (cod_marchio=146 and cod_macroarg=3417 and
cod_argomento=2108 and sigla_pos='STRISCIONE PREPARTITA') and not (
cod_marchio=1812 and cod_macroarg=3417 and cod_argomento=2108 and
sigla_pos='RETROPORTA DX 1a Fila-LED') and not (cod_marchio=1812 and
cod_macroarg=3417 and cod_argomento=2108 and sigla_pos='RETROPORTA SX 1a
Fila-LED') and not (cod_marchio=1812 and cod_macroarg=3417 and
cod_argomento=2108 and sigla_pos='RETTILINEO 1a Fila-LED') and not (
cod_marchio=1818 and cod_macroarg=3417 and cod_argomento=2108 and
sigla_pos='RETROPORTA DX 1a Fila-LED') and not (cod_marchio=1818 and
cod_macroarg=3417 and cod_argomento=2108 and sigla_pos='RETROPORTA SX 1a
Fila-LED') and not (cod_marchio=1818 and cod_macroarg=3417 and
cod_argomento=2108 and sigla_pos='RETTILINEO 1a Fila-LED') and not (
cod_marchio=1821 and cod_macroarg=3417 and cod_argomento=2108 and
sigla_pos='RETROPORTA DX 1a Fila-LED') and not (cod_marchio=1821 and
cod_macroarg=3417 and cod_argomento=2108 and sigla_pos='RETROPORTA SX 1a
Fila-LED')

```

```

Fila-LED') and not ( cod_marchio=1821 and cod_macroarg=3417 and
cod_argomento=2108 and sigla_pos='RETTILINEO 1a Fila-LED') and not (
cod_marchio=1822 and cod_macroarg=3417 and cod_argomento=2108 and
sigla_pos='RETROPORTA DX 1a Fila-LED') and not ( cod_marchio=1822 and
cod_macroarg=3417 and cod_argomento=2108 and sigla_pos='RETROPORTA SX 1a
Fila-LED')

...
... omissis per sensibilità dati aziendali ...

as r  INNER JOIN AGB_PAL_TARGET APT (NOLOCK)ON R.ID_PAL_AGB=APT.ID_PAL_AGB
AND APT.id_target=2 LEFT OUTER JOIN EMITTENTI_zone E (NOLOCK) ON
R.COD_EMITTENTE=E.COD_EMITTENTE GROUP BY chiave , cod_leader, nome_leader,
cod_marchio, nome_marchio, alias_marchio, cod_macroarg, desc_macroarg,
cod_argomento, desc_argomento, cod_circuito, R.cod_emittente,
R.nome_emittente, R.prog_ord, cod_trasmis, nome_trasmis, R.id_pal_agb,
data_trasmis, ora_inizio_trasmis, durata_trasmis, durata_servizio,
prog_inizio_tape, prog_fine_tape, numero_tape_1, numero_tape_2,
nome_servizio, categ_servizio, tipo_servizio, audience_effet,
numero_servizi, R.indic_circuito, R.indic_tv_satellite, R.indic_agb,
R.indic_nazionale, flag_stato, cod_pos, sigla_pos, apt.AUDIENCE_MEDIO,
apt.contatti, apt.ID_TARGET, apt.penetrazione, apt.share, e.COD_LOCALITA )
as TT LEFT OUTER JOIN TRASMISSIONI_EMITTENTE_TARGET TET (NOLOCK) ON
TET.COD_TRASMIS=TT.COD_TRASMIS ) as ttt ) as t4  LEFT OUTER JOIN
universo_target_localita utl (NOLOCK) on t4.id_target=utl.id_target and
t4.cod_localita=utl.cod_localita and t4.data_trasmis between data_inizio
and data_fine  LEFT OUTER JOIN agb_desc ad (NOLOCK) on
t4.id_target=ad.id_target  LEFT OUTER JOIN localita l on
t4.cod_localita=l.cod_localita  LEFT OUTER JOIN target_ord tgtord (nolock)
on t4.id_target=tgtord.id_target  LEFT OUTER JOIN XXXXXXXXX e on
t4.cod_emittente=e.cod_emittente  LEFT OUTER JOIN sponsor s (NOLOCK) on
t4.cod_leader= s.cod_leader and t4.cod_marchio=s.cod_marchio and
t4.cod_macroarg=s.cod_macroarg  LEFT OUTER JOIN sponsor_nome sn (NOLOCK) on
s.cod_sponsor=sn.cod_sponsor  LEFT OUTER JOIN sponsor_ord so (NOLOCK) on
s.cod_sponsor=so.cod_sponsor and s.grup_sponsor=so.grup_sponsor ORDER BY
t4.nome_emittente,DATA_trasmis,ORA_INIZIO_TRASMIS

SELECT 'Elapsed time, ms' = DATEDIFF ( ms, @start_time, GETDATE() )

SET STATISTICS TIME OFF

```

La query precedente, utilizzata all'interno di un applicativo proprietario di VIDIERRE, permette la generazione di recordset relativi al monitoraggio di un marchio per un Cliente all'interno di tutte le manifestazioni sportive, relativamente a tutte le posizioni in cui è visibile tale marchio, a tutte le emittenti televisive monitorate ed al target “individui” dei palinsesti AGB. Tali recordset vengono poi trasferiti, e ulteriormente elaborati e raggruppati dall'applicativo di VIDIERRE, che, attraverso il *Crystal Report*, genera documenti PDF finali di elaborazione per il marchio monitorato richiesto. In alcune occasioni, in cui le query devono essere ottimizzate perché lavorano su grandi quantità di dati, è necessario verificarne i tempi di esecuzione. Transact-SQL mette a disposizione il comando "SET STATISTICS TIME ON", usato nella precedente query, che permette di avere un report riepilogativo dei

tempi dettagliati per l'interpretazione, la compilazione e l'esecuzione di una interrogazione. Tuttavia, proprio perché il report è dettagliato, in alcuni casi come ad esempio le stored procedure (che al loro interno contengono più interrogazioni), è necessario avere i tempi totali dell'esecuzione e non quelli dettagliati. Per fare fronte a questa esigenza, e monitorare i tempi di esecuzione della precedente query su entrambe le piattaforme di SQL Server, e poterli quindi confrontare tra loro, è stato scritto, all'inizio della query, il codice:

```
SET STATISTICS TIME ON  
  
DECLARE @start_time DATETIME  
SELECT @start_time = GETDATE()
```

mentre al termine della stessa query è stato posto il codice:

```
SELECT 'Elapsed time, ms' = DATEDIFF ( ms, @start_time, GETDATE() )  
  
SET STATISTICS TIME OFF
```

In questo modo, si è salvato nella variabile `@start_time` l'ora di inizio dell'elaborazione per la query in esame, ed al termine è stata effettuata una differenza tra l'ora di fine e l'ora di inizio dell'elaborazione, con il risultato salvato in un recordset nella colonna '`Elapsed time, ms`'. Il risultato è stato sorprendente: la stessa query, eseguita su *Microsoft SQL Server 2000*, ha restituito un Elapsed Time pari a 345.912 [ms] mentre eseguendola su *Microsoft SQL Server 2008* ha restituito un Elapsed Time pari a 185.325 [ms]. Ciò equivale ad un risparmio di tempo di circa il **46,4 %** nell'esecuzione di una query molto frequente per il lavoro produttivo di VIDIERRE, che nella sua esecuzione coinvolge tabelle per un numero complessivo di record nell'ordine dei milioni.

Ad onor di cronaca, su query molto più semplici (ad esempio query di SELECT di dati in tabella, senza l'uso di JOIN, UNION ed altri operatori di algebra relazionale), nonostante le righe restituite al termine dell'elaborazione fossero contenute in un numero dell'ordine di centinaia di migliaia, l'Elapsed Time minore ottenuto con le due piattaforme è sempre stato quello ottenuto al termine dell'elaborazione su *SQL Server 2008*, ma la differenza in questi casi è di poche centinaia di millisecondi. Tralasciabile dunque ai fini valutativi circa miglioramenti a livello di tempi di esecuzione in processi produttivi.

3.8.3 Beta testing applicativi aziendali su SQL Server 2008

Appurato il considerevole miglioramento in termini di tempi elaborativi di query molto complesse, ci si è concentrati sull'utilizzo massivo e stressante degli applicativi aziendali, connessi non più a *Queen* con *SQL Server 2000*, ma a *Knight* con *SQL Server 2008*. Nel presente progetto di tesi, si è focalizzata l'attenzione su un aspetto fondamentale per la produttività di VIDIERRE: la stampa di prospetti riepilogativi di monitoraggio. Per ciò che concerne la migrazione a *SQL Server 2008*, ci si è concentrati tantissimo, oltre che sulla tempistica di esecuzione delle elaborazioni, anche sulla consistenza ed egualianza dei dati finali tra la nuova e la vecchia piattaforma. Il core business di VIDIERRE, infatti, non permette in alcun modo la possibilità di variazioni, anche se ipoteticamente minime, di valutazioni circa l'apparizione di marchi, le sponsorizzazioni, ed i valori complessivi di campagne pubblicitarie piuttosto che di supporti utilizzati. Era necessario, dunque, che qualsiasi dato finale prodotto da VIDIERRE sulla base di una piattaforma rodata ed in produzione come *SQL Server 2000* e consegnato al Cliente, fosse riprodotto identicamente anche sulla nuova piattaforma in fase di testing, il *SQL Server 2008*. Nel presente progetto di tesi, si è scelto di riportare come esempio la stampa di prospetti riepilogativi per Cliente circa la valutazione delle apparizioni di un marchio sponsor, nell'arco di un periodo di tempo stabilito. Nelle pagine che seguono, vi è il risultato dell'elaborazione di stampa di un esempio di quanto appena esplicato.

VIDEERRI

PROSPETTO M

Marchio: XXX Maglia

TARGET: Individui

- RIEPILOGATIVO PER MACROARGOMENTO, ARGOMENTO E POSIZIONE RIFERITO AD UN MARCHIO -

Cliente: YY

PERIODO: 01/04/2010 - 30/04/2010

EMITTENTI:Nazionali

MACROARG.	ARGOMENTO	POSIZIONE	SERVIZI	DURATA TOT SERVIZI (min)	SEC APP.	SUM. APPARIZ.	AUDIENCE EFFETTIVA	CONTATTI (x 100)	'GRP	VALORE
ALLENAMENTI	BACKDROP			1	10	7	2	228.419	1.099	0,40
ALLENAMENTI	BACKDROP			1	10	7	2	228.419	1.099	0,40
ALLENAMENTI	DINAMICA	DIVISA	21	273	203	47	13.218.213	24.099	23,04	€ 8.727,61
ALLENAMENTI	DINAMICA	GIACCA	29	91	71	38	10.452.390	22.800	18,22	€ 3.612,10
ALLENAMENTI	DINAMICA	TUTA	217	712	1.828	752	134.940.787	244.127	235,20	€ 110.146,31
ALLENAMENTI	DINAMICA		267	1.076	2.102	837	156.611.390	291.026	276,46	€ 122.486,02
ALLENAMENTI	STATICA			1	1	1	1	723.401	1.482	1,26
ALLENAMENTI	STATICA			1	1	1	1	723.401	1.482	1,26
ALLENAMENTI			269	1.087	2.110	840	159.563.210	293.607	278,12	€ 123.484,70
AMICHEVOLI	DINAMICA	TUTA	5	16	20	5	6.686.621	12.771	11,65	€ 3.213,39
AMICHEVOLI	DINAMICA		5	16	20	5	6.686.621	12.771	11,65	€ 3.213,39
AMICHEVOLI			5	16	20	5	6.686.621	12.771	11,65	€ 3.213,39
CAMPIONATO	BACKDROP			88	1.328	8.102	381	43.755.007	89.941	76,27
CAMPIONATO	BACKDROP		88	1.328	8.102	381	43.755.007	89.941	76,27	€ 571.778,21
CAMPIONATO	DINAMICA	BORSE	11	22	22	11	313.338		0,55	€ 89,35
CAMPIONATO	DINAMICA	DIVISA	1.262	14.985	15.834	5.988	474.842.230	882.310	827,66	€ 432.729,42
CAMPIONATO	DINAMICA	FRATINI	44	928	162	58	19.353.480	36.532	33,73	€ 5.886,90
CAMPIONATO	DINAMICA	GIACCA	118	889	450	180	29.590.938	58.507	51,58	€ 11.031,00
CAMPIONATO	DINAMICA	PALLONE	37	1.781	250	107	3.612.706	1.120	6,30	€ 4.035,59
CAMPIONATO	DINAMICA	TUTA	134	1.590	1.233	338	44.888.515	106.543	78,24	€ 23.669,22
CAMPIONATO	DINAMICA		1.606	20.195	17.951	6.682	572.601.767	1.085.012	998,06	€ 477.481,40
CAMPIONATO	STATICA	CARTELLO CORNER BASSO DX	68	4.524	715	246	22.809.723	45.956	39,76	€ 24.209,43
CAMPIONATO	STATICA	CARTELLO CORNER BASSO SX	55	4.396	619	244	8.417.918	11.423	14,67	€ 14.605,92
CAMPIONATO	STATICA	CORNICE MAXI SCHERMO	37	1.778	227	66	18.761.973	50.534	32,70	€ 33.226,25
CAMPIONATO	STATICA	CORNICE TUNNEL	94	1.075	690	117	8.744.381	3.063	15,24	€ 21.474,28

VIDEERRI

PROSPETTO M

Marchio: XXX Maglia

TARGET: Individui

- RIEPILOGATIVO PER MACROARGOMENTO, ARGOMENTO E POSIZIONE RIFERITO AD UN MARCHIO -

Cliente: YY

PERIODO: 01/04/2010 - 30/04/2010

EMITTENTI:Nazionali

MACROARG.	ARGOMENTO	POSIZIONE	SERVIZI	DURATA TOT SERVIZI (min)	SEC APP.	NUM. APPARIZ.	AUDIENCE EFFETTIVA	CONTATTI (x 100)	'GRP	VALORE
CAMPIONATO	STATICA	MAXISCHERMO	52	2.108	317	90	7.745.320	14.389	13,50	€ 7.795,92
CAMPIONATO	STATICA	MULTISPONSOR 1a Fila	111	5.314	2.147	517	22.795.484	52.743	39,73	€ 63.923,94
CAMPIONATO	STATICA	MULTISPONSOR 2a Fila	69	4.700	818	237	10.712.386	14.120	18,67	€ 22.244,83
CAMPIONATO	STATICA	RETREPORTA DX 1a Fila+LED	121	4.843	540	254	49.753.054	85.643	86,72	€ 28.817,26
CAMPIONATO	STATICA	RETREPORTA DX 2a Fila	74	4.108	641	273	51.310.717	96.339	89,44	€ 29.115,26
CAMPIONATO	STATICA	RETREPORTA SX 1a Fila+LED	67	4.724	602	273	10.433.789	7.882	18,19	€ 16.243,60
CAMPIONATO	STATICA	RETREPORTA SX 2a Fila	75	4.513	1.217	427	51.285.849	100.009	89,39	€ 48.211,77
CAMPIONATO	STATICA	RETTILINEO 1a Fila+LED	119	5.025	652	248	45.837.366	73.907	79,90	€ 26.260,80
CAMPIONATO	STATICA	RETTILINEO 2a Fila	130	5.140	3.130	547	75.117.918	136.720	130,93	€ 108.315,98
CAMPIONATO	STATICA	SOTTOTRIBUNA	51	4.360	750	219	5.499.288		9,59	€ 21.963,55
CAMPIONATO	STATICA		1.124	56.608	13.065	3.758	389.226.026	692.798	678,43	€ 466.398,80
			2.818	78.131	39.118	10.821	1.005.582.800	1.867.751	1.752,75	€ 1.515.628,50
CAMPIONATO	DINAMICA	DIVISA	1.278	17.169	29.108	11.091	€15.961.092	1.037.955	1.073,63	€ 2.123.403,76
CAMPIONATO	DINAMICA	GIACCA	151	3.940	1.272	454	27.490.811	56.703	47,92	€ 34.587,63
CAMPIONATO	DINAMICA	TUTA	182	2.568	2.814	470	113.779.260	207.001	198,32	€ 101.277,71
CAMPIONATO	DINAMICA		1.611	23.677	33.194	12.015	757.231.163	1.301.659	1.319,87	€ 2.259.289,11
CAMPIONATO	DINAMICA	DIVISA	1.611	23.677	33.194	12.015	757.231.163	1.301.659	1.319,87	€ 2.259.289,11
TIM CUP	DINAMICA	GIACCA	115	438	669	274	99.079.604	168.351	172,70	€ 206.490,31
TIM CUP	DINAMICA	PALLONE	6	66	197	9	8.474.183	23.972	14,77	€ 101.293,72
TIM CUP	DINAMICA		124	507	872	286	107.639.800	192.333	181,62	€ 307.856,20
TIM CUP	STATICA	CARTELLO CORNER BASSO SX	2	13	4	3	749.293	1.315	1,31	€ 237,57
TIM CUP	STATICA		2	13	4	3	749.293	1.315	1,31	€ 237,57
			126	520	876	289	108.389.003	193.638	188,92	€ 308.073,77
VARIE	DINAMICA	MAGLIA BAMBINI	54	488	214	84	10.807.125	10.362	18,84	€ 4.612,48
VARIE	DINAMICA	MAGLIA GIOCATORI	19	238	304	57	615.557		1,07	€ 1.280,30

VIDIERE

PROSPETTO M - RIEPILOGATIVO PER MACROARGOMENTO, ARGOMENTO E POSIZIONE RIFERITO AD UN MARCHIO -

Marchio: XXX

Cliente: YY

TARGET: Individui

PERIODO: 01/04/2010 - 30/04/2010

EMITTENTI:Nazionali

MACROARG.	ARGOMENTO	POSIZIONE	SERVIZI	DURATA TOT SERVIZI (min)	SEC APP.	NUM. APPARIZ.	AUDIENZE EFFETTIVA	CONTATTI (x 100)	*3GP	VALORE
VARIE	DINAMICA	MAGLIA TIFOSI	43	441	162	70	29.099.360	56.919	50,72	€ 17.279,61
VARIE	DINAMICA	PULLMAN	9	25	21	16	4.111.362	8.833	7,17	€ 2.487,26
VARIE	DINAMICA		125	1.192	701	227	44.633.704	76.114	77,80	€ 25.659,64
VARIE	STATICA	SITO INTERNET	28	61	158	29	15.066.065	28.137	26,26	€ 16.384,72
VARIE	STATICA		28	61	158	29	15.066.065	28.137	26,26	€ 16.384,72
VARIE			153	1.253	859	256	59.699.769	104.251	104,06	€ 42.644,36
Totali Emittenti	Nazionali		4.982	104.684	76.177	24.226	2.097.152.636	3.773.677	3.655,38	€ 4.252.293,84
Totale Marchio:	XXX		4.982	104.684	76.177	24.226	2.097.152.636	3.773.677	3.655,38	€ 4.252.293,84

VIDIERI

PROSPETTO M - RIEPILOGATIVO PER MACROARGOMENTO, ARGOMENTO E POSIZIONE RIFERITO AD UN MARCHIO -

Marchio: XXX Gold Partner

Cliente: YY

TARGET: Individui

PERIODO: 01/04/2010 - 30/04/2010

EMITTENTI:Nazionali

MACROARG.	ARGOMENTO	POSIZIONE	SERVIZI	DURATA TOT SERVIZI (min)	SEC APP.	NUM. APPARIZ.	AUDIENZE EFFETTIVA	CONTATTI (x 100)	*3GP	VALORE
ALLENAMENTI	BACKDROP			7	52	45	8	251.632	1.099	0,44
ALLENAMENTI	BACKDROP			7	52	45	8	251.632	1.099	0,44
CAMPIONATO	STATICA	CARTELLO CORNER ALTO DX		681	8.932	13.559	3.434	238.830.803	455.952	416.29
CAMPIONATO	STATICA	MULTISPONSOR 1a Fila		55	3.443	659	171	9.087.124	7.171	15.84
CAMPIONATO	STATICA	MULTISPONSOR 2a Fila		62	4.609	557	161	6.707.820	2.617	11.69
CAMPIONATO	STATICA	RETREPORTA DX 1a Fila+LED		491	8.516	7.049	1.462	187.781.282	353.106	327.31
CAMPIONATO	STATICA	RETREPORTA SX 1a Fila+LED		313	7.340	5.414	1.305	89.125.127	172.064	155.35
CAMPIONATO	STATICA	RETTILINEO 1a Fila+LED		482	8.139	16.902	2.845	166.052.776	340.066	289.43
CAMPIONATO	STATICA	RETTILINEO 1a Fila+LED		2.084	40.978	44.140	9.378	897.584.932	1.330.976	1.215.91
CAMPIONATO				2.084	40.978	44.140	9.378	897.584.932	1.330.976	1.215.91
TIM CUP	STATICA	RETTILINEO 1a Fila+LED		5	11	16	5	5.953.570	11.634	10.38
TIM CUP	STATICA			5	11	16	5	5.953.570	11.634	10.38
TIM CUP				5	11	16	5	5.953.570	11.634	10.38
VARIE	STATICA	SITO INTERNET		11	21	52	11	4.345.151	8.385	7.57
VARIE	STATICA			11	21	52	11	4.345.151	8.385	7.57
VARIE				11	21	52	11	4.345.151	8.385	7.57
Totali Emissori Nazionali				2.107	41.063	44.253	9.402	708.135.295	1.352.094	1.234.30
Totale Marchio: XXX				2.107	41.063	44.253	9.402	708.135.295	1.352.094	1.234.30
										€ 1.464.684,16

Dal punto di vista della consistenza dei dati, il prospetto precedente, ottenuto dall'applicativo VIDIERRE elaborando dati utilizzando la piattaforma *SQL Server 2008*, risulta identico in ogni campo, riga e colonna allo stesso prospetto ottenuto poggiandosi sulla vecchia piattaforma. Qualsiasi difficoltà di utilizzo di campi di solito molto problematici, come data, valuta, int o float, è stato scongiurato. Tutte le stored procedures utilizzate dall'applicativo non hanno restituito alcun tipo di errore, ed i raggruppamenti vari utilizzati per l'ottenimento del prospetto di cui sopra non hanno generato alcuna differenza valutativa.

Dal punto di vista della tempistica di generazione del prospetto precedente, si è notato nuovamente un guadagno considerevole: si è passato da un tempo di 17m51s per il *SQL Server 2000* a un tempo di 9m47s per *SQL Server 2008*, equivalente ad un risparmio di tempo in secondi di circa il **45,2 %**. Ciò che deve far riflettere è come tale risparmio in termini di tempistica, sia riferito ad un unico marchio per un unico Cliente; il lavoro di VIDIERRE, con riferimento unicamente allo sponsoring, si basa su n Clienti, ciascuno con n marchi, ciascuno con n esigenze diverse. Visto il tutto in un'ottica più generalmente ampliata, si comprende come, un'eventuale migrazione definitiva a *SQL Server 2008*, potesse prevedere che ogni ipotetico minuto di tempo risparmiato per l'elaborazione di un marchio, si sarebbe riversato in un'ipotetica ora risparmiata per un Cliente cui si riferiscono 60 marchi, e così via...

4. MIGRAZIONE DELLA PIATTAFORMA DATI VIDIERRE A MICROSOFT SQL SERVER 2008

Il processo di migrazione della piattaforma dati dell’Azienda VIDIERRE a questo punto è entrato nel vivo. Dopo aver analizzato nei capitoli precedenti la compatibilità strutturale di uno stesso database sulle due differenti piattaforme, dopo aver effettuato numerosi test prestazionali con l’utilizzo di query più o meno complesse, dopo aver verificato la consistenza dei dati prodotti da applicativi proprietari di VIDIERRE con la nuova piattaforma ed averne quindi constatato la totale assenza di errori, si è ora giunti al momento cruciale del progetto di tesi: l’effettiva migrazione dell’intera piattaforma dati di VIDIERRE a *Microsoft SQL Server 2008*. Tutto ciò consci del fatto che l’installazione di un nuovo server in produzione, e la successiva migrazione, avrebbe creato problematiche non prevedibili e dunque testabili attraverso l’utilizzo di Virtual Machines, ma forti delle esperienze che i test avevano alimentato e pronti a fronteggiare qualsiasi tipo di problematica fosse venuta alla luce.

4.1 Installazione e configurazione nuovo SQL Server

Come previsto in fase di progettazione dell’intero intervento di migrazione della piattaforma dati, l’utilizzo delle VM a questo punto non è più indispensabile. Dopo aver eseguito tutti i test necessari, ed aver verificato l’assenza di errori critici nel processo di migrazione, il server *King*, fino a questo momento utilizzato come Host di *VMware ESXi*, è stato interamente formattato, e susseguentemente è stato equipaggiato con il sistema operativo *Microsoft Windows Server 2003 Enterprise Edition 64bit*. Al termine della configurazione del nuovo server (aggiunta al dominio aziendale, installazione Network Team a 4 dispositivi LAN, verifica driver e configurazione servizi server principali), si è proceduto con l’installazione del *SQL Server 2008*, seguendo le stesse procedure adottate in passato per l’installazione della piattaforma sulla VM *Knight* (Cfr. Par. 3.6).

Al termine dell’installazione e successiva configurazione dell’istanza di *SQL Server 2008*, si è proceduto, con la metodologia scelta (Cfr. Par. 3.7.1) all’effettiva migrazione di tutti i database dal vecchio server *Gordon* (*SQL Server 2000*) al nuovo server *King*, e cioè al

restore/backup di ogni singolo database. Data la criticità dell’operazione, che avrebbe comunque comportato uno stop produttivo di circa 40 minuti complessivi, si è preferito effettuare tutte le procedure del caso durante la notte, in modo da limitare al minimo i disagi dovuti al downtime della piattaforma produttiva. Il restore ed il successivo backup di ciascun database non hanno generato alcun tipo di problematica.

4.1.1 Migrazione degli account utente

In fase di testing della nuova piattaforma *SQL Server 2008*, utilizzando sempre l’utente “sa” per connettersi al SQL Server, non si era resa necessaria la presenza di ulteriori utenti all’interno del server. Va fatta una dovuta considerazione: all’interno del database *vidierre2k*, uno dei tanti database in produzione, vi è la presenza della tabella “Utenti”, peraltro correttamente migrata; tale tabella, tuttavia, non garantisce come preventivato l’accesso e dunque la connessione tramite ODBC o stringa di connessione proprietaria al SQL Server, ma è esclusivamente utilizzata a livello di restrizioni di accesso/utilizzo del database stesso. Una volta connessi al SQL Server, vi è la presenza di una directory interna all’istanza creata, chiamata “Protezione \ Account di accesso”, al cui interno sono memorizzati tutti gli utenti in grado di accedere e connettersi all’istanza stessa. Subito dopo la migrazione, tale directory conteneva esclusivamente gli utenti di sistema, tra cui appunto “sa”, ma non tutti gli account creati ed in funzione nella vecchia piattaforma di *SQL Server 2000*. Si è reso dunque necessario trovare un modo per trasferire gli account utente da una piattaforma all’altra. Nei forum e nelle community vi sono parecchie discussioni su questo argomento, e si è potuto facilmente trovare il metodo risolutivo necessario [7]. All’interno di *SQL Server 2000*, si è eseguito il seguente listato:

```
USE master
GO
IF OBJECT_ID ('sp_hexadecimal') IS NOT NULL
DROP PROCEDURE sp_hexadecimal
GO
CREATE PROCEDURE sp_hexadecimal
@binvalue varbinary(256),
@hexvalue varchar(256) OUTPUT
AS
DECLARE @charvalue varchar(256)
DECLARE @i int
DECLARE @length int
DECLARE @hexstring char(16)
SELECT @charvalue = '0x'
SELECT @i = 1
SELECT @length = DATALENGTH (@binvalue)
```

```

SELECT @hexstring = '0123456789ABCDEF'
WHILE (@i <= @length)
BEGIN
DECLARE @tempint int
DECLARE @firstint int
DECLARE @secondint int
SELECT @tempint = CONVERT(int, SUBSTRING(@binvalue,@i,1))
SELECT @firstint = FLOOR(@tempint/16)
SELECT @secondint = @tempint - (@firstint*16)
SELECT @charvalue = @charvalue +
SUBSTRING(@hexstring, @firstint+1, 1) +
SUBSTRING(@hexstring, @secondint+1, 1)
SELECT @i = @i + 1
END
SELECT @hexvalue = @charvalue
GO

IF OBJECT_ID ('sp_help_revlogin_2000_to_2008') IS NOT NULL
DROP PROCEDURE sp_help_revlogin_2000_to_2008
GO
CREATE PROCEDURE sp_help_revlogin_2000_to_2008

@login_name sysname = NULL,
@include_db bit = 0,
@include_role bit = 0

AS
DECLARE @name sysname
DECLARE @xstatus int
DECLARE @binpwd varbinary (256)
DECLARE @dfltdb varchar (256)
DECLARE @txtpwd sysname
DECLARE @tmpstr varchar (256)
DECLARE @SID_varbinary varbinary(85)
DECLARE @SID_string varchar(256)

IF (@login_name IS NULL)
DECLARE login_curs CURSOR STATIC FOR
SELECT sid, [name], xstatus, password, isnull(db_name(dbid), 'master')
FROM master.dbo.sysxlogins
WHERE srvid IS NULL AND
[name] <> 'sa'
ELSE
DECLARE login_curs CURSOR FOR
SELECT sid, [name], xstatus, password, isnull(db_name(dbid), 'master')
FROM master.dbo.sysxlogins
WHERE srvid IS NULL AND
[name] = @login_name

OPEN login_curs

FETCH NEXT FROM login_curs INTO @SID_varbinary, @name, @xstatus, @binpwd, @dfltdb

IF (@@fetch_status = -1)
BEGIN
PRINT 'No login(s) found.'
CLOSE login_curs
DEALLOCATE login_curs
RETURN -1
END

SET @tmpstr = /* sp_help_revlogin script '
PRINT @tmpstr
SET @tmpstr = '** Generated '
+ CONVERT (varchar, GETDATE()) + ' on ' + @@SERVERNAME + ' */'
PRINT @tmpstr
PRINT ''
PRINT ''
PRINT ''
PRINT '/***** CREATE LOGINS *****/'

WHILE @@fetch_status = 0
BEGIN
PRINT ''

```

```

SET @tmpstr = '-- Login: ' + @name
PRINT @tmpstr

IF (@xstatus & 4) = 4
BEGIN -- NT authenticated account/group
IF (@xstatus & 1) = 1
BEGIN -- NT login is denied access
SET @tmpstr = '' --'EXEC master..sp_denylogin ''' + @name + '''
PRINT @tmpstr
END
ELSE
BEGIN -- NT login has access
SET @tmpstr = 'IF NOT EXISTS (SELECT * FROM sys.server_principals WHERE [name] = ''' + @name +
''')'
PRINT @tmpstr
SET @tmpstr = CHAR(9) + 'CREATE LOGIN [' + @name + '] FROM WINDOWS'
PRINT @tmpstr
END
END
ELSE
BEGIN -- SQL Server authentication
EXEC sp_hexadecimal @SID_varbinary, @SID_string OUT

IF (@binpwd IS NOT NULL)
BEGIN -- Non-null password
EXEC sp_hexadecimal @binpwd, @txtpwd OUT
SET @tmpstr = 'CREATE LOGIN [' + @name + '] WITH PASSWORD=' + @txtpwd + ' HASHED'
END
ELSE
BEGIN -- Null password
SET @tmpstr = 'CREATE LOGIN [' + @name + '] WITH PASSWORD=''''
END

SET @tmpstr = @tmpstr + ', CHECK_POLICY=OFF, SID=' + @SID_string
PRINT @tmpstr
END

FETCH NEXT FROM login_curs INTO @SID_varbinary, @name, @xstatus, @binpwd, @dfldb
END

IF @include_db = 1
BEGIN
PRINT ''
PRINT ''
PRINT ''
PRINT '/***** SET DEFAULT DATABASES *****/'

FETCH FIRST FROM login_curs INTO @SID_varbinary, @name, @xstatus, @binpwd, @dfldb

WHILE @@fetch_status = 0
BEGIN
PRINT ''
SET @tmpstr = '-- Login: ' + @name
PRINT @tmpstr

SET @tmpstr = 'ALTER LOGIN [' + @name + '] WITH DEFAULT_DATABASE=[' + @dfldb + ']'
PRINT @tmpstr

FETCH NEXT FROM login_curs INTO @SID_varbinary, @name, @xstatus, @binpwd, @dfldb
END
END

IF @include_role = 1
BEGIN
PRINT ''
PRINT ''
PRINT ''
PRINT '/***** SET SERVER ROLES *****/'

FETCH FIRST FROM login_curs INTO @SID_varbinary, @name, @xstatus, @binpwd, @dfldb

WHILE @@fetch_status = 0
BEGIN
PRINT ''

```

```

SET @tmpstr = '-- Login: ' + @name
PRINT @tmpstr

IF @xstatus &16 = 16 -- sysadmin
BEGIN
SET @tmpstr = 'exec master.dbo.sp_addsrvrolemember @loginame=''' + @name + ''',
@rolename='sysadmin''
PRINT @tmpstr
END

IF @xstatus &32 = 32 -- securityadmin
BEGIN
SET @tmpstr = 'exec master.dbo.sp_addsrvrolemember @loginame=''' + @name + ''',
@rolename='securityadmin''
PRINT @tmpstr
END

IF @xstatus &64 = 64 -- serveradmin
BEGIN
SET @tmpstr = 'exec master.dbo.sp_addsrvrolemember @loginame=''' + @name + ''',
@rolename='serveradmin''
PRINT @tmpstr
END

IF @xstatus &128 = 128 -- setupadmin
BEGIN
SET @tmpstr = 'exec master.dbo.sp_addsrvrolemember @loginame=''' + @name + ''',
@rolename='setupadmin''
PRINT @tmpstr
END

IF @xstatus &256 = 256 --processadmin
BEGIN
SET @tmpstr = 'exec master.dbo.sp_addsrvrolemember @loginame=''' + @name + ''',
@rolename='processadmin''
PRINT @tmpstr
END

IF @xstatus &512 = 512 -- diskadmin
BEGIN
SET @tmpstr = 'exec master.dbo.sp_addsrvrolemember @loginame=''' + @name + ''',
@rolename='diskadmin''
PRINT @tmpstr
END

IF @xstatus &1024 = 1024 -- dbcreator
BEGIN
SET @tmpstr = 'exec master.dbo.sp_addsrvrolemember @loginame=''' + @name + ''',
@rolename='dbcreator''
PRINT @tmpstr
END

IF @xstatus &4096 = 4096 -- bulkadmin
BEGIN
SET @tmpstr = 'exec master.dbo.sp_addsrvrolemember @loginame=''' + @name + ''',
@rolename='bulkadmin''
PRINT @tmpstr
END

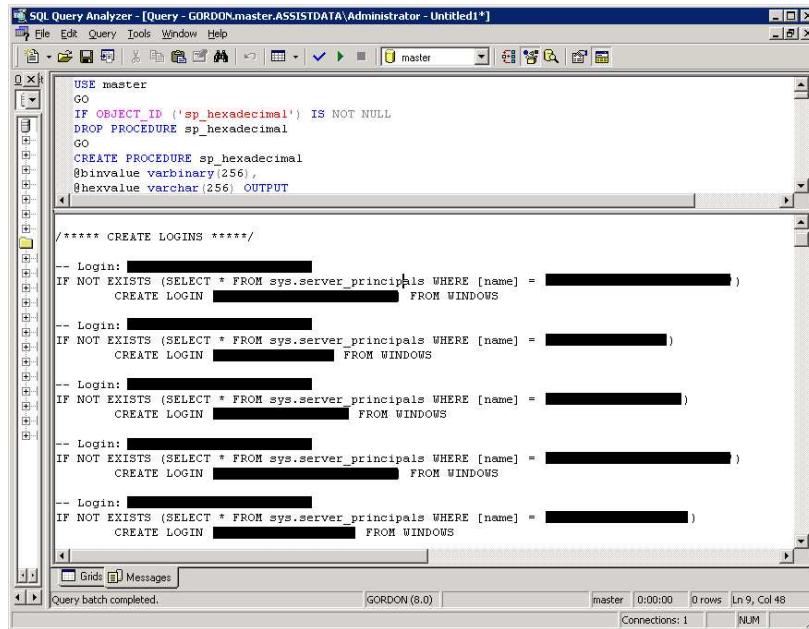
FETCH NEXT FROM login_curs INTO @SID_varbinary, @name, @xstatus, @binpwd, @dfltmdb
END
END

CLOSE login_curs
DEALLOCATE login_curs
RETURN 0
GO

exec sp_help_revlogin_2000_to_2005 @login_name=NULL, @include_db=1, @include_role=1
GO

```

Al termine dell'esecuzione del listato precedente, il *SQL Server 2000* ha fornito un altro listato, visibile in figura 4.1, da eseguire in *SQL Server 2008 Management Studio*.



```

USE master
GO
IF OBJECT_ID ('sp_hexadecimal') IS NOT NULL
DROP PROCEDURE sp_hexadecimal
GO
CREATE PROCEDURE sp_hexadecimal
@binvalue varbinary(256),
@hexvalue varchar(256) OUTPUT
AS
-- This stored procedure converts a binary value to its
-- hexagonal representation. It is used by the sp_hexvalue
-- stored procedure.

-- **** CREATE LOGINS ****

-- Login: [REDACTED]
IF NOT EXISTS (SELECT * FROM sys.server_principals WHERE [name] = [REDACTED])
CREATE LOGIN [REDACTED] FROM WINDOWS

-- Login: [REDACTED]
IF NOT EXISTS (SELECT * FROM sys.server_principals WHERE [name] = [REDACTED])
CREATE LOGIN [REDACTED] FROM WINDOWS

-- Login: [REDACTED]
IF NOT EXISTS (SELECT * FROM sys.server_principals WHERE [name] = [REDACTED])
CREATE LOGIN [REDACTED] FROM WINDOWS

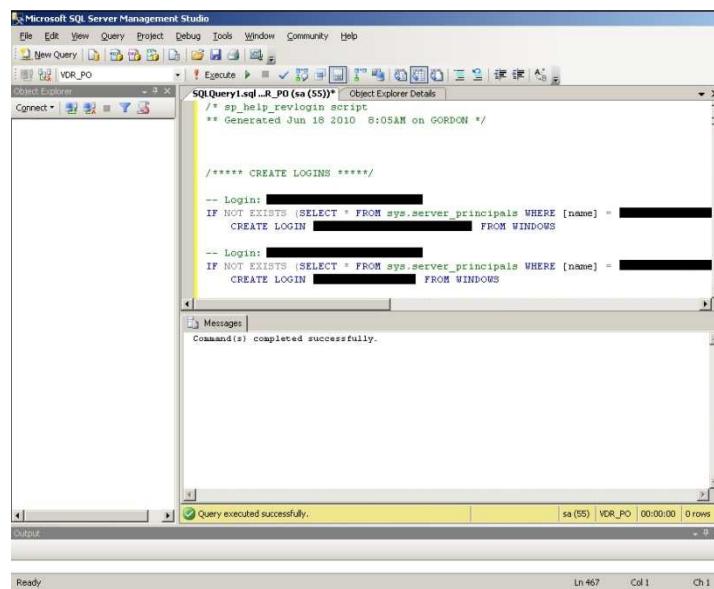
-- Login: [REDACTED]
IF NOT EXISTS (SELECT * FROM sys.server_principals WHERE [name] = [REDACTED])
CREATE LOGIN [REDACTED] FROM WINDOWS

-- Login: [REDACTED]
IF NOT EXISTS (SELECT * FROM sys.server_principals WHERE [name] = [REDACTED])
CREATE LOGIN [REDACTED] FROM WINDOWS

```

Figura 4.1: Listato fornito come risultato in *SQL Server 2000* dopo l'esecuzione dello script di migrazione utenti

Una volta copiato ed incollato il codice ottenuto dal *SQL Server 2000* nel *SQL Server 2008*, e lanciatane l'esecuzione, come risultato, visibile in Figura 4.2, si è ottenuta la creazione nella nuova piattaforma di tutti gli utenti presenti nella vecchia.



```

Microsoft SQL Server Management Studio - Query [VDR_PO] [sa (55)] [Object Explorer]
File Edit View Project Debug Tools Window Community Help
New Query Object Explorer Object Explorer Details Execute
/* sp_help_revlogin script
** Generated Jun 18 2010 8:05AM on GORDON */

-- **** CREATE LOGINS ****

-- Login: [REDACTED]
IF NOT EXISTS (SELECT * FROM sys.server_principals WHERE [name] = [REDACTED])
CREATE LOGIN [REDACTED] FROM WINDOWS

-- Login: [REDACTED]
IF NOT EXISTS (SELECT * FROM sys.server_principals WHERE [name] = [REDACTED])
CREATE LOGIN [REDACTED] FROM WINDOWS

```

Messages

Command(s) completed successfully.

Figura 4.2: Risultato dell'esecuzione dello script per la creazione degli utenti in *SQL Server 2008*

La procedura eseguita ha permesso non solo la completa migrazione di tutti gli account utente, ma contestualmente, per ciascun utente, ha automaticamente impostato la password, il database di default associato, e tramite l'esecuzione della stored procedure master.dbo.sp_addsrvrolemember ha impostato i server role (come ad esempio dbcreator, serveradmin o sysadmin) di ciascun utente.

4.1.2 Utilizzo di AWE per la memoria server

Il server di produzione *King*, contestualmente con la migrazione della piattaforma, è stato potenziato con l'aggiunta di moduli RAM fino al raggiungimento della Memoria RAM totale disponibile di 32 GB. Si è dunque creata un'altra esigenza a livello di configurazione del *SQL Server 2008*: l'utilizzo di AWE, un'opzione abilitata sul *SQL Server 2000* in esecuzione su *Gordon*. In *Microsoft SQL Server 2008* è possibile utilizzare l'API AWE (Address Windowing Extensions) per consentire l'accesso alla memoria fisica oltre i limiti impostati sulla memoria virtuale configurata. La quantità di memoria utilizzabile dipende dalla configurazione hardware e dal supporto fornito dal sistema operativo. Con l'introduzione dei sistemi operativi Microsoft Windows Server 2003, la quantità di memoria fisica supportata è aumentata. La memoria fisica alla quale AWE ha accesso dipende dal sistema operativo in uso [7]. Nel caso di VIDIERRE, avendo la versione Enterprise Edition di Windows Server 2003, la memoria fisica massima supportata è pari a 32 GB: esattamente la stessa quantità di memoria RAM raggiunta dopo il potenziamento del server *King*. L'opzione AWE ENABLED estende le funzionalità delle applicazioni, consentendo l'accesso alla memoria fisica disponibile oltre i limiti impostati nel relativo spazio degli indirizzi della memoria virtuale. Durante l'avvio, infatti, SQL Server riserva solo una piccola parte della memoria mappata AWE. Man mano che si rende necessaria memoria mappata AWE aggiuntiva, il sistema operativo la alloca in modo dinamico a SQL Server. Analogamente, se sono necessarie meno risorse, SQL Server può restituire memoria mappata AWE al sistema operativo, per l'utilizzo da parte di altri processi o applicazioni. Il bilanciamento tra SQL Server e il sistema operativo è soggetto ai vincoli dei parametri min server memory e max server memory [8]. Dopo un'accurata ricerca, conseguente a test con esito negativo su *King* per verificare che la memoria mappata AWE fornisse effettivi vantaggi e miglioramenti della velocità e della metodologia di gestione della memoria, si è appreso che l'opzione AWE non solo non è

necessaria, ma non può essere configurata con sistemi operativi a 64 bit. Sarebbe stato utile, di conseguenza, un alert o un link ad un help online od offline direttamente in *SQL Server 2008*, al momento dell'impostazione AWE ENABLED, che fornisse linee guida e soprattutto tracciasse i limiti e l'impossibilità di funzionamento di tale opzione in sistemi operativi a 64 bit, come nel caso di VIDIERRE.

4.1.3 Impostazione manuale delle opzioni di memoria

Già in *SQL Server 2000*, nella configurazione delle opzioni di memoria da destinare alle istanze di SQL Server, era presente la possibilità di indicare un limite minimo “fixed” alla memoria RAM da allocare al momento dell'attivazione del servizio SQL Server. Inoltre, era possibile specificare, in maniera dinamica, quale dovessero essere i valori della memoria RAM minima e massima riservata per il servizio SQL Server. Su *Gordon*, come impostazione di memoria per il *SQL Server 2000*, si era scelto di utilizzare l'opzione della memoria allocata staticamente, e si era impostato il valore pari a 11.155 MB (pari circa al 75% dei 15.355 MB di RAM disponibile). In figura utenti 4.3 è visibile uno screenshot della finestra di impostazione della memoria in *SQL Server 2000*, con i valori impostati come appena esposto.

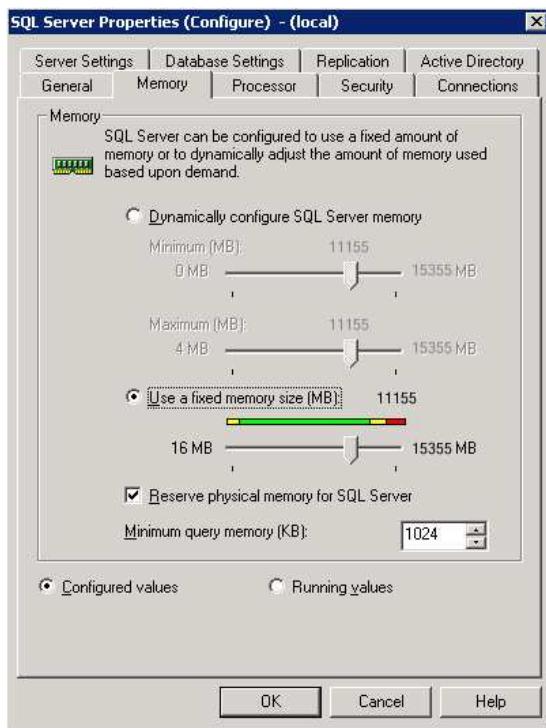


Figura 4.3: Impostazione memoria in *SQL Server 2000*

La prima differenza sostanziale in *SQL Server 2008*, è l'assenza di un'opzione che consenta di impostare “staticamente” la memoria destinata esclusivamente al servizio di SQL Server. Sono presenti, tuttavia, le opzioni di impostazione dei valori minimi e massimi della memoria, chiamate rispettivamente *min server memory* e *max server memory*. Qualora si ritenesse necessario, per impostare la quantità fissa di memoria da allocare al pool di buffer di *SQL Server 2008* basterà impostare i valori di *min server memory* e *max server memory* sulla stessa quantità. L'impostazione dei due parametri in esame, tuttavia, è parecchio delicata ed ha necessitato un attento studio sul da farsi, in base alla configurazione fisica della macchina – *King* – su cui era in esecuzione *SQL Server 2008*. Come si è appreso, per impostazione predefinita, in SQL Server i requisiti di memoria possono variare dinamicamente in base alle risorse di sistema disponibili. L'impostazione predefinita di *min server memory* è 0, mentre l'impostazione predefinita di *max server memory* è 2147483647. La quantità minima di memoria che è possibile specificare per *max server memory* è 16 MB. Quando SQL Server utilizza la memoria in modo dinamico, esegue verifiche periodiche del sistema per determinare la quantità di memoria fisica libera disponibile. In Windows Server 2003, SQL Server utilizza l'API di notifica relativa alla memoria *QueryMemoryResourceNotification* per determinare quando il pool di buffer è in grado di allocare e rilasciare memoria [9]. Si è appreso inoltre che sarebbe consigliabile consentire a SQL Server di utilizzare dinamicamente la memoria. È tuttavia possibile impostare manualmente le opzioni relative alla memoria e limitare la quantità di memoria accessibile a SQL Server. Prima di impostare la quantità di memoria per SQL Server, è utile determinare l'impostazione di memoria appropriata, sottraendo alla memoria fisica totale la quantità necessaria per Windows Server 2003 e per eventuali altre istanze di SQL Server, nonché per altri utilizzi se il sistema non è dedicato esclusivamente a SQL Server. La differenza così ottenuta rappresenta la quantità di memoria massima assegnabile a SQL Server [9]. Applicando quanto appreso, e ricordando che il server in produzione *King* disponeva di 31,9 GB di RAM realmente disponibile, si è calcolato il valore della memoria fisica totale da destinare al sistema operativo Windows Server 2003 ed alle altre applicazioni in 1,4 GB totali. In tal modo, avendo scelto di non permettere a *SQL Server 2008* di allocare la memoria minima in maniera dinamica, ma di permettere questa feature solo sul valore di memoria massima, si è proceduto all'impostazione manuale dei parametri *min server memory* e *max server memory*. Il primo, per far sì che SQL allocasse al minimo l'intera quantità di RAM ad esclusione del valore calcolato come riservato per il

sistema operativo e le altre applicazioni, è stato impostato a 31.232 MB (risultato della differenza tra valore RAM fisica installata di 32.665 MB e 1.433 MB di memoria riservata al sistema operativo). Il secondo, è stato invece impostato seguendo il valore massimo indicato da Microsoft affinché il SQL Server gestisca dinamicamente la quantità di memoria massima, e cioè 2.147.483.647 MB. In questo modo, si è ottenuto un livello prestazionale discretamente alto, considerato anche che, allocando sulla memoria fisica la gran parte delle risorse richieste, l'accesso al disco per l'utilizzo del file di paging sarebbe stato di gran lunga poco frequente durante le elaborazioni dati effettuate dal *SQL Server 2008*. In figura 4.4 è visibile la finestra di impostazione dei valori di memoria per *SQL Server 2008*, con i valori attualmente configurati.

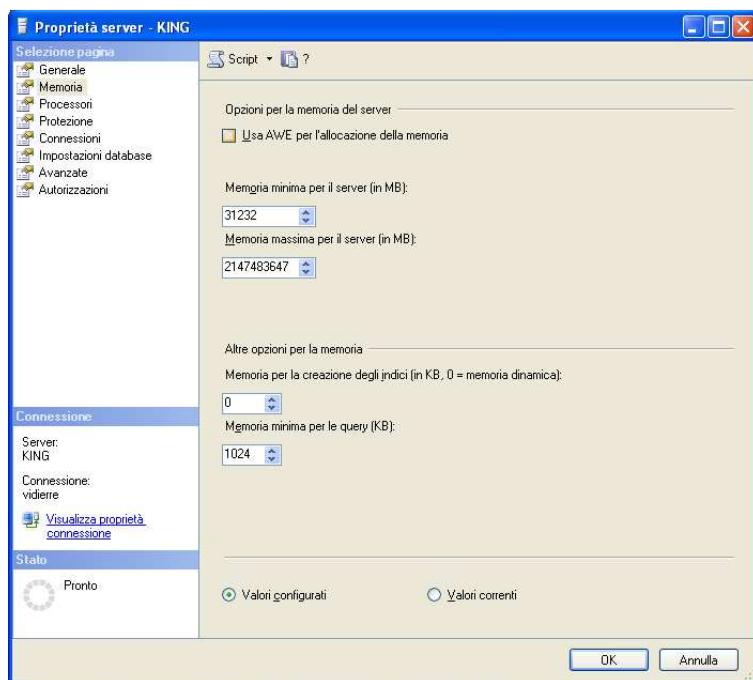


Figura 4.4: Impostazione memoria in *SQL Server 2008*

Da notare, come esposto in precedenza, che sul *SQL Server 2000* di *Gordon* il valore della memoria riservata a SQL Server era impostato a circa il 75% della memoria fisicamente disponibile. Sul *SQL Server 2008* di *King*, al contrario, il valore in percentuale è salito al 95%: questo perché la quantità di memoria riservata al sistema operativo ed alle altre applicazioni non è sostanzialmente cambiata rispetto al sistema di *Gordon*, ma è raddoppiata tuttavia la memoria fisica totale disponibile su *King*.

4.2 Problematiche di migrazione

Come previsto in fase progettuale, al momento dell’effettiva migrazione dell’intera piattaforma dati si sono verificate diverse problematiche dovute alla migrazione stessa, non prevedibili in fase di testing sia del nuovo *SQL Server 2008*, sia delle applicazioni che vi si poggiano (Cfr. Par. 3.8.3). Ciò è causato, sostanzialmente, dalla differenza che permane tra un sistema fisico ed una macchina virtuale: per quanto le VM rispecchino al massimo la configurazione di una macchina fisica, esse non potranno mai essere la loro copia perfetta, a livello di configurazioni e driver dell’hardware, a livello di configurazioni del sistema operativo, ed a livello di numerose altre variabili di cui non è possibile prevederne il comportamento. Di seguito si son volute indicare le problematiche giudicate, tra le altre, le più importanti derivanti dalla migrazione di una piattaforma dati da *SQL Server 2000* a *SQL Server 2008*.

4.2.1 sp_compatibility_level

La prima problematica derivante dalla migrazione è stata il “compatibility level”, e cioè l’impostazione di aspetti specifici del funzionamento del database in modo che risultino compatibili con la versione specificata di SQL Server. La sintassi ALTER DATABASE seguente, che rappresenta una novità di *SQL Server 2008*, sostituisce la stored procedure *sp_dbcmptlevel* presente sul *SQL Server 2000* per l’impostazione del livello di compatibilità del database [10]:

```
ALTER DATABASE database_name  
SET COMPATIBILITY_LEVEL = { 80 | 90 | 100 }
```

Il parametro COMPATIBILITY_LEVEL { 80 | 90 | 100 } specifica la versione di SQL Server con cui il database deve risultare compatibile. Il valore deve essere uno dei seguenti:

80 = SQL Server 2000

90 = SQL Server 2005

100 = SQL Server 2008

Al momento del restore di ciascun database sul nuovo *SQL Server 2008*, il valore che il sistema ha associato automaticamente al COMPATIBILITY_LEVEL è stato 100. Ciò ha comportato numerosi errori in fase di stampa dei prospetti, tra i quali ad esempio l’alert di

assenza completa di dati all'interno di intere colonne dei recordset da cui attingere per la creazione del prospetto. La causa di questo specifico problema è stata individuata nella nuova prerogativa di *SQL Server 2008*, che prevede il tipo del risultato di un'operazione UNION derivato in modo indipendente da INSERT SELECT. Per ogni ramo di UNION si esegue il cast nel tipo del risultato di UNION e quindi è eseguito il cast nel tipo della colonna di destinazione di INSERT. Se sono presenti tipi incompatibili nell'operazione UNION, il primo cast causa un errore che non permette la prosecuzione e il riempimento del recordset necessario all'applicativo di VIDIERRE. In *SQL Server 2000*, al contrario, anche se l'unione aveva esito negativo a causa di conversioni di tipo non compatibili, se l'operazione era eseguita singolarmente, all'interno di INSERT SELECT l'operazione UNION era sempre completata perché il ramo per il tipo del risultato dell'operazione UNION non era mai convertito [10].

Per tali ragioni, si è deciso di eseguire un ALTER DATABASE e modificare, o meglio, ripristinare il compatibility level a “80”, poiché la rivisitazione dell'intero applicativo, per ogni singola procedura che prevede questo tipo di operazione UNION, avrebbe richiesto risorse non preventive e non disponibili, soprattutto in virtù di una visione di prossima innovazione intrapresa da VIDIERRE riguardante i suoi applicativi proprietari. Non appena eseguita la procedura di ALTER DATABASE di cui sopra, l'applicativo ha restituito correttamente i risultati ed i prospetti richiesti, ripristinando dunque la funzionalità produttiva.

4.2.2 Da DTS a DTSX

Il DTS (Data Transformation Service) è un pacchetto, o meglio un set di oggetti correlati tra di loro (connessioni, task, flussi, etc.) che possono essere usati per l'accesso, la trasformazione e la manipolazione dei dati reperibili in un ampio raggio di possibilità, inclusi file di testo e database relazionali. Il DTS è stato introdotto a partire dalla release di SQL Server 7.0 ed è stato in seguito, vista la sua immensa popolarità e percentuale di utilizzo, sviluppato e migrato nelle successive versioni di SQL Server. Anche VIDIERRE, all'interno dei suoi applicativi, adotta un utilizzo massiccio di questo strumento, soprattutto nella lettura di files di testo e conseguente manipolazione ed inserimento dei dati in essi contenuti all'interno dei propri database.

A partire dal SQL Server 2005, i pacchetti DTS, ricalcando le orme di sviluppo multi-software di Microsoft già protagoniste in altre famose suite di programmi della casa di Redmond, sono stati inglobati all'interno del *Microsoft SQL Server Integration Service* (SSIS), assumendo il nome di DTSx. I nuovi DTSx, al contrario del *SQL Server 2000* che permetteva la progettazione visiva di ciascun DTS attraverso un software incluso nelle release di *SQL Server 2000* (il DTS Designer), possono essere creati solamente attraverso una versione .NET dell'ambiente *Visual Studio*. Le funzionalità dei nuovi DTSx rimangono immutate rispetto ai loro predecessori, mantenendo le prerogative del multi-threading che li hanno resi performanti e necessari allo stesso tempo.

All'interno di *SQL Server 2008* è presente un tool per l'importazione in SSIS dei “vecchi” pacchetti DTS di *SQL Server 2000*, il quale dovrebbe prevedere la conversione automatica di tutti gli oggetti, le connessioni, i flussi, etc., nel nuovo formato. Il condizionale “dovrebbe” è d'obbligo, perché nel caso specifico di VIDIERRE non è stato così. Uno degli applicativi di VIDIERRE si poggia sull'esecuzione di DTS per l'importazione giornaliera dei “CP30” (costi spot a 30 secondi), partendo da file di testo formattati, dai quale estrae sulla base di colonne a larghezza stabilita da caratteri speciali (\$) delle righe contenenti i dati necessari. Segue un breve esempio del contenuto di un file al cui interno vi sono valori dei CP30.

```
Date$Start Time$Channel$Spot Type$Prog Typology$roduc\Variables$Duration$Investment$GRP abs.$Reach$-
24/10/2009$02:00:01$Canale 5$AUTOPUB$telegiornale$MEDIASET PREMIUM TV DIG.TERR.$00:00:31$4'600,00 € $404'559$404'559$-
24/10/2009$02:00:12$Rai Uno$PROMO$anteprima cinema tv$RAI UNO$00:00:30$520,00 € $228'713$228'713$-
24/10/2009$02:00:24$Italia 1$BETWEEN$motociclismo su pista$POKERSTARS SITO INTERNET$00:00:16$2'620,00 € $365'111$365'111$-
24/10/2009$02:00:32$Canale 5$WITHIN$telegiornale$STYLE BOUTIQUE VIDEOGIOCOS$00:00:30$4'600,00 € $405'038$405'038$-
24/10/2009$02:00:40$Italia 1$BETWEEN$motociclismo su pista$FOR MEN MAGAZINE MENSILE$00:00:10$2'240,00 € $365'111$365'111$-
24/10/2009$02:00:42$Rai Uno$AUTOPUB$anteprima cinema tv$RAI COMUNICATO DIGIT.TERR.$00:00:33$624,00 € $224'195$224'195$-
24/10/2009$02:00:50$Italia 1$BETWEEN$motociclismo su pista$FORD FIESTA GPL-BENZ.$00:00:15$2'710,00 € $363'392$363'392$-
24/10/2009$02:01:02$Canale 5$WITHIN$telegiornale$MEDIA SHOP MINIPUB SPILL.BIRRA$00:01:00$11'930,00 € $412'515$412'515$-
24/10/2009$02:01:05$Italia 1$PROMO$motociclismo su pista$ITALIA 1$00:00:27$3'730,00 € $359'954$359'954$-
```

Al momento dell'importazione effettiva dei CP30, non appena eseguita la migrazione dell'intera piattaforma dati a *SQL Server 2008*, l'applicativo di VIDIERRE ha generato un errore in lettura del file, nonostante i DTS che si occupano di tale procedura fossero stati importati con successo attraverso il wizard incluso in *SQL Server 2008*. Il funzionamento di tale procedura prevede l'utilizzo di due differenti pacchetti DTS: il primo, si occupa di leggere il contenuto di una directory, alla ricerca di nuovi file di testo contenenti CP30 a partire dall'ultima importazione eseguita con successo; a seguire, tale DTS passa il nome del

nuovo file di testo contenente i nuovi CP30, all'interno di una variabile globale, al secondo pacchetto DTS, il quale si occupa di aprire il file, leggerne il contenuto, elaborarlo e scriverlo nel database con procedure di cui si omette la specifica. I due file ex-DTS, ora DTSx, che dovevano occuparsi di questi procedimenti, come già accennato, provocavano eccezioni ed errori in fase di lettura e decodifica dei files di testo contenenti i CP30. Si è proceduto, dunque, al “tentativo” di modifica dei due DTSx attraverso il *Visual Studio .NET*. Da notare che uno degli oggetti ora disponibili per la programmazione dei pacchetti DTSx è “Execute DTS”: ciò avrebbe dovuto permettere, benché all'interno di SSIS, l'esecuzione del vecchio pacchetto DTS come se fosse un “eseguibile esterno”, ma così non è stato. Il DTSx restituiva gli stessi errori. A questo punto, per mantenere inalterata la prerogativa di produttività di VIDIERRE, si è deciso di ri-programmare ex-novo entrambi i pacchetti DTSx che permettono l'importazione dei CP30, direttamente dal *Visual Studio*. All'interno del presente progetto di tesi, si è volutamente omessa l'attività di programmazione e di creazione dei nuovi pacchetti, in quanto si è ritenuta non fosse prettamente pertinente al progetto stesso. Ci si è limitati, in figura 4.5, a riportare la struttura del nuovo pacchetto DTSx creato ed implementato in SSIS.

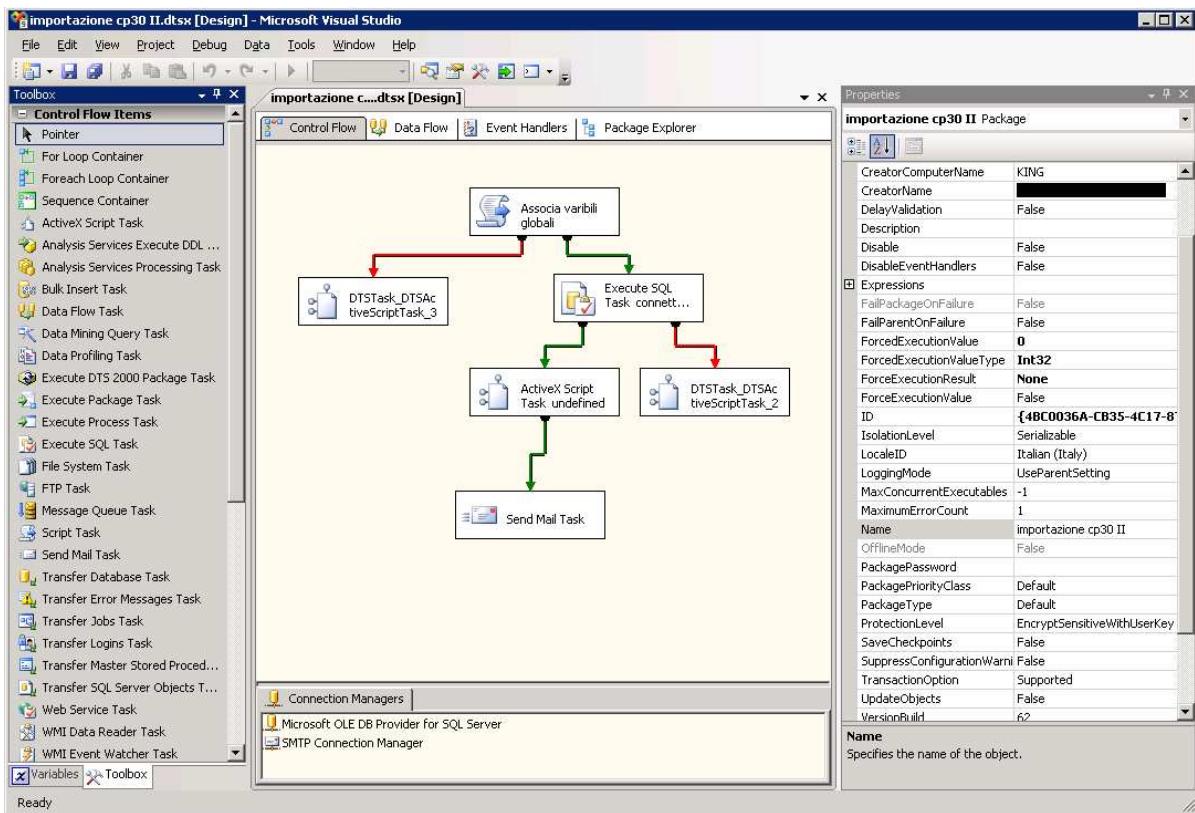


Figura 4.5: Struttura nuovo pacchetto DTSx per l'importazione nel database di CP30 a partire da file di testo

Gli applicativi di VIDIERRE, come già accennato, compiono un utilizzo massiccio dei DTS, e, di tutti i DTS funzionanti in *SQL Server 2000*, attraverso le procedure di importazione/migrazione in SSIS a pacchetti DTSx, nessuno di questi ha funzionato. Si è dovuto dunque procedere, benché nel presente progetto di tesi si sia posta l’attenzione esclusivamente su due di questi, alla ri-programmazione di tutti i pacchetti in tempi brevissimi per ripristinare la produttività di tutti i settori di VIDIERRE.

In conclusione, la migrazione a *SQL Server 2008*, dal punto di vista dei pacchetti DTS perfettamente funzionanti con la precedente versione del motore di casa Microsoft, ha causato non pochi disagi all’azienda VIDIERRE. Sarebbe utile che Microsoft ponesse maggiore attenzione sulla compatibilità di questi pacchetti, in quanto sono a volte alla base di parecchi processi produttivi vitali, visto che le premesse seguenti fornite a livello informativo da Microsoft risultano al contrario molto confortanti:

“Quando si esegue la migrazione di pacchetti DTS al formato di Integration Services, la struttura dei pacchetti migrati e la completezza della migrazione stessa dipendono dal contenuto del pacchetto originale. La migrazione dei pacchetti può avere gli esiti seguenti:

- *La migrazione viene completata senza problemi, dato che i pacchetti contengono attività e utilizzano funzionalità che vengono associate solo ad attività e funzionalità di Integration Services.*
- *Benché i pacchetti contengano attività e funzionalità DTS di cui non è possibile eseguire la migrazione direttamente ad attività e funzionalità di IS, la funzionalità dei pacchetti viene mantenuta tramite l'utilizzo di pacchetti DTS incapsulati.*
- *In alcuni casi, ad esempio quando il nome del pacchetto contiene i caratteri non validi (/ \ : [] . =) o quando il pacchetto contiene oggetti non registrati, la Migrazione guidata pacchetti non è in grado di eseguire la migrazione finché non vengono risolti questi problemi.” [11]*

Nel caso di VIDIERRE, appunto, la migrazione è stata completata senza problemi, ma la funzionalità dei pacchetti migrati non è stata in alcun modo ripristinata all’interno di SSIS; al contrario, si è dovuto procedere come già detto alla completa ri-programmazione ex-novo di ciascun pacchetto coinvolto, con un notevole dispendio di tempi e risorse.

4.2.3 Uso restrittivo di apici in campi testo

Un'altra grave problematica riscontrata al termine della migrazione della piattaforma dati di VIDIERRE a *Microsoft SQL Server 2008* è venuta alla luce attraverso l'applicativo che opera su un altro settore aziendale, il MIP (Monitoraggio Investimenti Pubblicitari). Durante un'interrogazione circa una pratica di un gruppo, attraverso l'applicativo proprietario, quest'ultimo restituiva un errore di run-time con il seguente messaggio: “*Server: messaggio 245, livello 16, stato 1, riga 1. Conversion failed when converting the nvarchar value 'ea160' to data type int.*”. Per risalire alla base del problema, si è analizzato il codice VB inerente alla funzionalità dell'applicativo che restituiva questo tipo di errore, nello specifico si è cercato all'interno del *frmFatturaSEnew*, nella Sub *TvwPraticheQuadr_NodeClick*. Si è partiti dal presupposto che il campo *Cod_Organizzato* della tabella *Protocollo* è *nvarchar(5)*, mentre all'interno del codice VB la variabile *Cod_Organizzato* è di tipo *String*. Di seguito si riporta il listato che all'atto dell'esecuzione generava il problema:

```
'RIEMPIMENTO DELLA TREEVIEW DEGLI ALLEGATI

strsql = "SELECT pr.cod_pratica, pr.cod_gruppo, ga.Desc_Gruppo, a1.mese, a2.costo_riportato,
a2.cod_alleg_2, na2.cod_tip, na2.Desc_tipologia"
strsql = strsql + " FROM Periodo pe"
strsql = strsql + " JOIN Protocollo pr on pe.mese = pr.mese"
strsql = strsql + " JOIN Gruppi_Auto ga on pr.Cod_Gruppo = ga.Cod_Gruppo "
strsql = strsql + " JOIN Allegato_1 al on pr.cod_pratica = al.Pratica"
strsql = strsql + " JOIN Dett_Allegato_1_AU d1 on al.Anno = d1.Anno AND al.Mese = d1.Mese AND
a1.Service = d1.Service AND a1.Cod_Gruppo = d1.Cod_Gruppo AND a1.Cod_Organizzato =
d1.Cod_Organizzato"
strsql = strsql + " JOIN allegato_2_vwau a2 ON d1.cod_alleg_2 = a2.cod_alleg_2"
strsql = strsql + " JOIN natura_alleg_2_vwau na2 ON a2.cod_tip = na2.cod_tip"
strsql = strsql + " WHERE pr.cod_gruppo in ('SE','SX')"
strsql = strsql + " AND pr.cod_organizzato =" + Cod_Organizzato
strsql = strsql + " AND pr.Anno = " + str(AnnoPratica)
strsql = strsql + " AND pe.quadrimestre=' " + Node.text + " '"
strsql = strsql + " ORDER BY pe.num_mese"
```

Nel precedente listato, è stata indicata in rosso la riga che assemblava in maniera errata la stringa SQL con cui l'applicativo andava ad interrogare il database. La stringa SQL generata era la seguente:

```
SELECT pr.cod_pratica, pr.cod_gruppo, ga.Desc_Gruppo, a1.mese, a2.costo_riportato,
a2.cod_alleg_2, na2.cod_tip, na2.Desc_tipologia FROM Periodo pe JOIN Protocollo pr on pe.mese
= pr.mese JOIN Gruppi_Auto ga on pr.Cod_Gruppo = ga.Cod_Gruppo JOIN Allegato_1 al on
pr.cod_pratica = al.Pratica JOIN Dett_Allegato_1_AU d1 on al.Anno = d1.Anno AND al.Mese =
d1.Mese AND al.Service = d1.Service AND a1.Cod_Gruppo = d1.Cod_Gruppo AND a1.Cod_Organizzato =
d1.Cod_Organizzato JOIN allegato_2_vwau a2 ON d1.cod_alleg_2 = a2.cod_alleg_2 JOIN
natura_alleg_2_vwau na2 ON a2.cod_tip = na2.cod_tip WHERE pr.cod_gruppo in ('SE','SX') AND
pr.cod_organizzato = 80555 AND pr.Anno = 2010 AND pe.quadrimestre='Aprile - Giugno' ORDER BY
pe.num_mese
```

Il problema si è scoperto dunque essere generato dalla diversa gestione, da parte di *SQL Server 2008*, delle conversioni tra campi di tipo *nvarchar* e campi di tipo *int*. Eseguendo la query precedente su *SQL Server 2000*, infatti, il risultato era il seguente:

cod_pratica	cod_gruppo	Desc_Gruppo	mese	costo_riportato	cod_alleg_2	cod_tip	Desc_tipologia
485333	SE	SEAT	Aprile-Giugno	8.740,00	34.112	26	Investimenti vari

La stessa query, eseguita su *SQL Server 2008*, generava l'errore di “*conversion failed*” di cui sopra. Il problema è stato risolto semplicemente racchiudendo il valore della variabile *Cod_Organizzato* tra apici, come si può vedere nel listato seguente:

'RIEMPIMENTO DELLA TREEVIEW DEGLI ALLEGATI

```
strsql = "SELECT pr.cod_pratica, pr.cod_gruppo, ga.Desc_Gruppo, al.mese, a2.costo_riportato,
a2.cod_alleg_2, na2.cod_tip, na2.Desc_tipologia"
strsql = strsql + " FROM Periodo pe"
strsql = strsql + " JOIN Protocollo pr on pe.mese = pr.mese"
strsql = strsql + " JOIN Gruppi_Auto ga on pr.Cod_Gruppo = ga.Cod_Gruppo "
strsql = strsql + " JOIN Allegato_1 al on pr.cod_pratica = al.Pratica"
strsql = strsql + " JOIN Dett_Allegato_1_AU d1 on al.Anno = d1.Anno AND al.Mese = d1.Mese AND
al.Service = d1.Service AND al.Cod_Gruppo = d1.Cod_Gruppo AND al.Cod_Organizzato =
d1.Cod_Organizzato"
strsql = strsql + " JOIN allegato_2_vwau a2 ON d1.cod_alleg_2 = a2.cod_alleg_2"
strsql = strsql + " JOIN natura_alleg_2_vwau na2 ON a2.cod_tip = na2.cod_tip"
strsql = strsql + " WHERE pr.cod_gruppo in ('SE','SX')"
strsql = strsql + " AND pr.cod_organizzato = '" + Trim(Cod_Organizzato) + "'"
strsql = strsql + " AND pr.Anno = " + str(AnnoPratica)
strsql = strsql + " AND pe.quadrimestre=''' + Node.text + '''"
strsql = strsql + " ORDER BY pe.num_mese"
```

In verde è visibile la riga cui sono state apportate le modifiche. La stringa SQL generata da tale riga, dopo la modifica, è diventata la seguente:

```
SELECT pr.cod_pratica, pr.cod_gruppo, ga.Desc_Gruppo, al.mese, a2.costo_riportato,
a2.cod_alleg_2, na2.cod_tip, na2.Desc_tipologia FROM Periodo pe JOIN Protocollo pr on pe.mese
= pr.mese JOIN Gruppi_Auto ga on pr.Cod_Gruppo = ga.Cod_Gruppo JOIN Allegato_1 al on
pr.cod_pratica = al.Pratica JOIN Dett_Allegato_1_AU d1 on al.Anno = d1.Anno AND al.Mese =
d1.Mese AND al.Service = d1.Service AND al.Cod_Gruppo = d1.Cod_Gruppo AND al.Cod_Organizzato =
d1.Cod_Organizzato JOIN allegato_2_vwau a2 ON d1.cod_alleg_2 = a2.cod_alleg_2 JOIN
natura_alleg_2_vwau na2 ON a2.cod_tip = na2.cod_tip WHERE pr.cod_gruppo in ('SE','SX') AND
pr.cod_organizzato = '80555' AND pr.Anno = 2010 AND pe.quadrimestre='Aprile - Giugno' ORDER BY
pe.num_mese
```

La problematica, dunque, è stata dovuta alla differente gestione di un tipo di dati in un campo: in *SQL Server 2000*, passando un valore *int* in un campo impostato come tipo di dato *varchar*, non c’è bisogno di racchiudere il valore stesso tra apici; in *SQL Server 2008*, al contrario, la gestione è molto più restrittiva: se il campo è indicato come un *nvarchar*, nel codice esecutivo bisogna obbligatoriamente racchiudere il valore tra apici, anche se il valore stesso è un *int*. Diversamente, *SQL Server 2008* genera errore di conversione tra tipi di dato. Una volta

analizzato e risolto il problema, dunque, si è passati alla modifica in blocco di tutto il codice sorgente dell'applicativo di VIDIERRE in questione, modificando tutti i frammenti di codice interessati dalla presente problematica di migrazione.

4.2.4 Viste strutturate in Management Studio 2008

Alcuni operatori specializzati di VIDIERRE, nel corso del loro lavoro giornaliero, hanno compiuto negli ultimi anni un uso intensivo delle viste strutturate per modificare “al volo” alcune rilevazioni o alcune informazioni su record salvati nel database.

In *Microsoft SQL Server 2000* cioè era reso molto semplice attraverso l'utilizzo di SQL Server Enterprise Manager, che permetteva una veloce costruzione di una query attraverso l'uso di diagrammi. Ciascun operatore ne avesse necessità e permessi, infatti, costruiva la sua vista strutturata scegliendo per ciascuna tabella i campi da visualizzare nei risultati, e a colpo d'occhio erano evidenziate le eventuali relazioni tra tali campi. Poteva inoltre impostare range di date in cui racchiudere la ricerca, o comunque scrivere parti di query direttamente in SQL attraverso l'editor dedicato presente tra le opzioni di costruzione query in Enterprise Manager. Il vantaggio assoluto che si traeva da questi procedimenti, era la possibilità di modificare, direttamente nella tabella dei risultati della query, ciascun campo in ciascuna riga, nonostante facessero parte di tabelle diverse. Si “aggirava” così il limite del tempo impiegato ad ottenere lo stesso risultato utilizzando maschere e funzioni dell'applicativo VIDIERRE.

In *Microsoft SQL Server 2008*, al contrario, attraverso l'utilizzo del SQL Server Management Studio, ciò non è più possibile. Il programma offre le stesse possibilità di costruzione delle viste offerte già dal *SQL Server 2000*, ma con una differenza sostanziale: non è più possibile modificare i campi presenti nelle varie righe, in quanto i dati e le celle che li contengono risultano in sola lettura. In figura 4.6, pagina seguente, è rappresentata l'interfaccia del Management Studio, utilizzata per le viste strutturate, in modalità costruzione query con diagrammi, in cui è stato cerchiato il limite delle celle restituite in sola lettura.

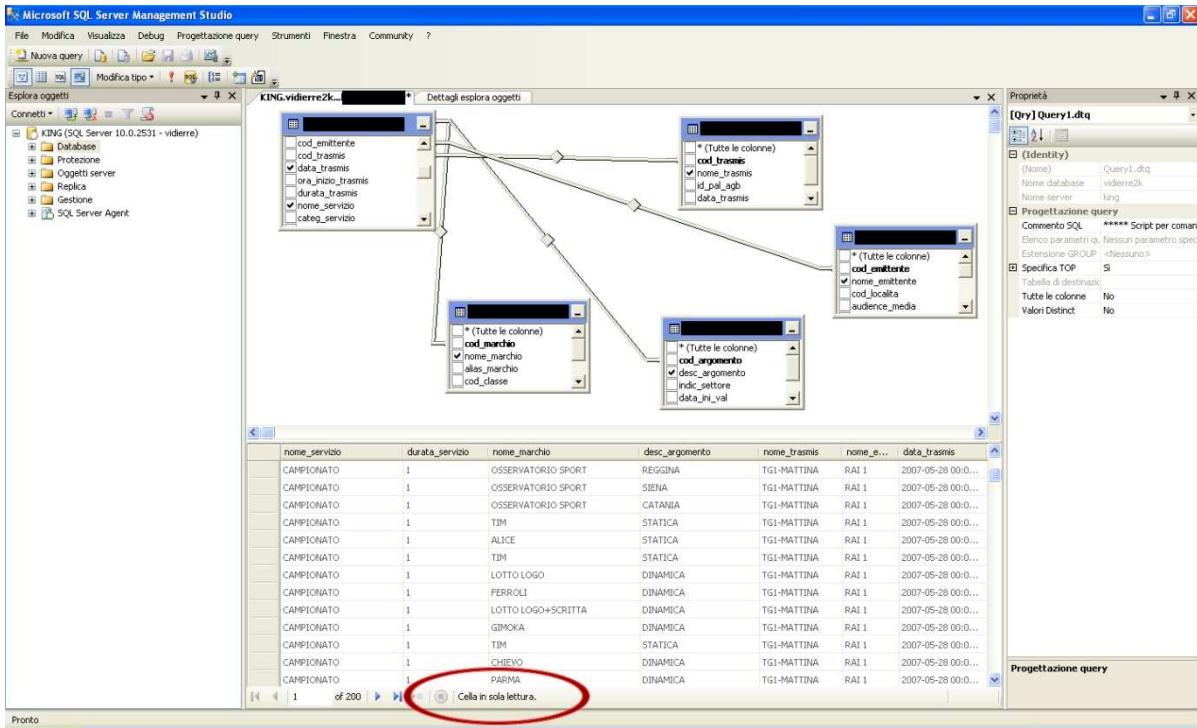


Figura 4.6: Viste strutturate in Microsoft SQL Server Management Studio

Questo limite, purtroppo, si è rivelato essere un “grosso” limite. Parecchi operatori specializzati utilizzavano le procedure appena enunciate per la modifica massiva “al volo” di informazioni all’interno del database. Dopo l’avvenuta migrazione dell’intera piattaforma dati a *SQL Server 2008*, l’unica maniera per modificare i record ottenuti con criteri di selezione da più tabelle è seguire le normali procedure previste nell’applicativo VIDIERRE.

Si è altresì provato ad aggirare tale limite, cercando di utilizzare il *SQL Server 2000 Enterprise Manager* registrando il nuovo server *King* in modo che si potesse avere accesso al nuovo server utilizzando un precedente software di gestione SQL Server. Purtroppo, anche questo tentativo si è rivelato vano: all’atto della registrazione del nuovo server, era restituito il messaggio di errore “*KING – To connect this server you must use SQL Server Management Studio*”.

4.2.5 Reimpostazione Server Jobs

L'ultima problematica nel gruppo di quelle definite "importanti" all'inizio del presente capitolo, ha riguardato i server jobs ed i maintenance plans. Al termine della migrazione definitiva a *Microsoft SQL Server 2008*, si è dovuta ripristinare interamente la situazione dei job e dei piani manutenzione precedentemente attivi sul *SQL Server 2000*. Si riporta, a titolo di esempio, la creazione attraverso il wizard contenuto in *SQL Server Management Studio* del piano di manutenzione per un database, volto a creare giornalmente, ad un orario stabilito, il backup differenziale del database stesso.

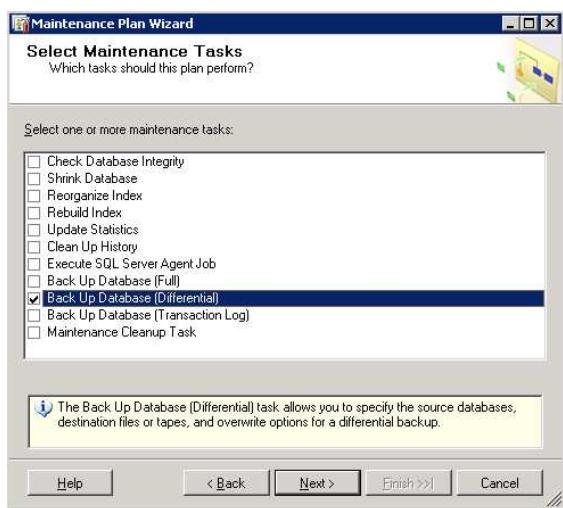


Figura 4.7: Wizard di configurazione dei Maintenance Plans

In figura 4.7 è visibile uno screenshot dei primi passaggi proposti dal wizard di configurazione dei Maintenance Plans di SQL Server. Come si nota, la scelta è ampia e maggiore rispetto alla precedente versione di SQL Server. Nel caso in esame, si è scelto di configurare un backup differenziale. Un backup differenziale è basato sull'ultimo backup completo dei dati. In un backup differenziale sono acquisiti solo i dati che hanno subito modifiche dopo il backup completo. Tali dati sono definiti *base* del backup differenziale. Un backup differenziale include solo i dati che sono stati modificati dalla base differenziale. In *SQL Server 2008* i backup differenziali del file sono estremamente rapidi in quanto il motore di database di *SQL Server 2008*, a differenza del predecessore, tiene traccia delle modifiche apportate dalla creazione della base differenziale [12]. La politica manutentiva di VIDIERRE prevede, per ogni database, un backup differenziale a cadenza giornaliera ed un backup

completo a cadenza settimanale. Attraverso il wizard di configurazione, dunque, si è proseguito con la scelta del database di cui eseguire il backup differenziale, con lo scheduling di tale operazione, e l'indicazione di altri parametri come il nome del file di backup e la relativa posizione, e l'overwrite di altri media esistenti aventi lo stesso nome file (al posto dell'append, in quanto giudicato non ottimale ai fini del concetto stesso di backup differenziale).

Al termine delle procedure di impostazione del maintenance plan, il *SQL Server Management Studio* ha non solo creato il piano manutenzione all'interno delle proprietà di management dell'istanza di SQL Server in uso, ma anche creato il relativo Job di esecuzione all'interno del SQL Agent, avente come opzioni e tempi di esecuzione gli stessi valori impostati nel wizard di creazione come opzioni di scheduling. La procedura, dunque si è rivelata alquanto semplice ed intuitiva, benché lunga sotto certi aspetti. In una realtà come VIDIERRE, in cui sono attualmente impostati ed in funzione 18 processi tra piani di manutenzione e server jobs, la procedura di ri-creazione di ciascun Job così come era precedentemente impostato in *SQL Server 2000* non è stata nello specifico ottimale, in quanto si è dovuto “leggere” ogni singola opzione ed ogni più piccola impostazione di scheduling dal vecchio piano per poi re-impostarle sul nuovo. Il tutto moltiplicato per 18.

In un'Azienda con necessità più consistenti di VIDIERRE, questa procedura, in caso di una stessa migrazione come da presente progetto di tesi, potrebbe causare non poche problematiche. Sarebbe utile, in conclusione, lo sviluppo da parte di Microsoft stessa di un tool o di una procedura in parte automatizzata che permetta, assieme a tutte le componenti già previste, la migrazione anche dei piani di manutenzione o più in generale dei server jobs già impostati in una precedente versione di SQL Server.

5. DISCUSSIONE E NOTE CONCLUSIVE

Il presente progetto di tesi è partito con lo studio dell'intera piattaforma dati di un'Azienda reale, come VIDIERRE, per la quale la piattaforma stessa è alla base del suo business produttivo, e si è poi concentrato sulla fattibilità della migrazione dell'intera piattaforma dati da *Microsoft SQL Server 2000* a *Microsoft SQL Server 2008*. Ci si è concentrati, in particolare, sulla compatibilità strutturale dei database durante il processo di migrazione, sull'integrità dei dati al loro interno dopo l'avvenuto processo, sulle prestazioni esecutive garantite dalla nuova piattaforma, sui costi che tale attività progettuale ha comportato per l'acquisto delle necessarie licenze di *SQL Server 2008*, sul corretto funzionamento di tutti gli applicativi proprietari di VIDIERRE che si poggiano ai differenti database e, non per ultima, sulla comparazione tra gli effettivi miglioramenti ottenuti e le problematiche riscontrate in fase di migrazione.

Fondamentale per tutta l'attività progettuale è stato l'utilizzo delle Virtual Machines, in quanto ha permesso la corretta esecuzione di test e comparazioni su una macchina virtuale (copia perfetta del server in produzione) senza effettivamente coinvolgere in alcun modo il server fisico e, di conseguenza, la produzione stessa non ha risentito di alcun tipo di contraccolpo. L'utilità estrema delle macchine virtuali, con tutti i vantaggi che ne conseguono a livello economico e di utilizzo di risorse umane, avvalora la strada intrapresa nel 2009 da VIDIERRE, con la graduale sostituzione di tutti i server fisici con la loro "copia virtuale" utilizzando la piattaforma di virtualizzazione "VMware ESXi".

L'Azienda VIDIERRE ha sempre utilizzato, dal momento della sua distribuzione, la versione *Enterprise* di *SQL Server 2000*, acquistandone le dovute licenze a costi parecchio elevati. Si è proceduto, per questo motivo, ad un'attenta valutazione delle differenti versioni rilasciate da Microsoft del *SQL Server 2008*, per scegliere la versione più adatta alle esigenze di VIDIERRE senza tralasciare il fattore economico, anch'esso punto fondamentale in fase di scelta. Al termine dell'analisi, si è optato per la versione *Standard* per due fondamentali ragioni: la prima consiste nel fatto che il downgrade delle versioni (dalla precedente *SQL Server 2000 Enterprise* alla futura *SQL Server 2008 Standard*) non prevede una perdita di funzionalità precedentemente utilizzate, ma al contrario un'impossibilità di utilizzo di nuove

funzionalità implementate nella nuova piattaforma (giudicate d’altro canto non utili al core business di VIDIERRE); la seconda, e assolutamente importante, è la differenza dal punto di vista economico delle versioni *Standard* ed *Enterprise* di *SQL Server 2008* (si parla, in questo caso, di circa il triplo del costo).

La comparazione strutturale effettuata sui database migrati alla nuova piattaforma, installata e configurata su una VM appositamente creata, non ha rilevato alcun tipo di problematica. L’attenzione, dunque, si è concentrata sulla comparazione prestazionale tra le due differenti piattaforme, attraverso l’utilizzo di query più o meno complicate (Cfr. Par. 3.8.2) e l’utilizzo dei software applicativi per monitorare i tempi produttivi di prospetti finali utilizzati da VIDIERRE (Cfr. Par. 3.8.3). Più in generale, qui di seguito sono riportate le più importanti comparazioni migliorative in termini di tempi di restituzione dei risultati ottenute utilizzando la nuova piattaforma *SQL Server 2008*:

- una query abbastanza complessa (Cfr. Par. 3.8.2), eseguita su *Microsoft SQL Server 2000*, ha restituito un Elapsed Time pari a 345.912 [ms] mentre eseguita su *Microsoft SQL Server 2008* ha restituito un Elapsed Time pari a 185.325 [ms]. Ciò equivale ad un risparmio di tempo di circa il **46,4 %** nell’esecuzione di una query all’ordine del giorno nel lavoro di VIDIERRE;
- nella produzione di un prospetto riepilogativo di valorizzazione delle apparizioni di marchi sponsor per un Cliente, a parametri di uso comune, si è passati da un tempo di esecuzione di 17m51s per il *SQL Server 2000* ad un tempo di esecuzione di 9m47s per *SQL Server 2008*, equivalente ad un risparmio di tempo in secondi di circa il **45,2 %**. Ciò che deve far riflettere è come tale risparmio in termini di tempistica è riferito ad un unico marchio per un unico Cliente; il lavoro di VIDIERRE, con riferimento unicamente allo sponsoring, si basa su n clienti, ciascuno con n marchi, ciascuno con n esigenze diverse;
- nel processo di importazione dei palinsesti per tutte le emittenti televisive monitorate da VIDIERRE, attraverso l’utilizzo di un DTSx che legge, elabora, e salva i dati a partire da un file di testo formattato a separatore, si è passati da un tempo di esecuzione medio di 5 minuti per il *SQL Server 2000* ad un tempo di esecuzione medio di meno di 2 minuti. Si parla, ovviamente, di tempo di esecuzione medio perché il contenuto di ciascun file di testo contenente i palinsesti è variabile in base al numero

di trasmissioni televisive trasmesse dal broadcaster (si pensi, per esempio, ai canali *SKY CALCIO*: durante i weekend in cui sono previsti incontri di calcio di varie manifestazioni sportive, i palinsesti che li riguardano hanno numerose trasmissioni; al contrario, durante la settimana, i palinsesti sono composti da uniche trasmissioni “cartello”). Anche in questo caso, benché si valuti un tempo di esecuzione medio, si è ottenuto un risparmio di tempo in minuti di circa il **70 %**;

- per l’attribuzione di un valore economico ad un’apparizione di un marchio rilevata, basato su precisi algoritmi matematici certificati proprietari di VIDIERRE che comprendono l’uso di numerosi parametri, il software applicativo utilizza delle procedure alquanto complesse di matching, verifica, elaborazione e scrittura di ciascun dato. Il ricalcolo, dunque, è un procedimento che richiede parecchie risorse alla piattaforma in uso. Eseguendo la procedura, utilizzando gli stessi parametri (tra cui l’impostazione del ricalcolo su 1000 rilevazioni), su entrambe le piattaforme, il risultato è stato quantificabile in un tempo di esecuzione di 8m12s per il *SQL Server 2000* contro i 58s per *SQL Server 2008*, equivalente ad un risparmio di tempo di circa il **88,2 %**.

Alla luce dei risultati dei confronti tra le differenti piattaforme testate, a livello economico, prestazionale e non ultimo comparativo tra vantaggi e svantaggi (in termini di tempi, risorse utilizzate e problematiche affrontate) dei processi migratori, VIDIERRE ha valutato con esito positivo la convenienza reale di un upgrade definitivo a *Microsoft SQL Server 2008*.

Al termine del processo di migrazione, in corrispondenza della fase cruciale del processo stesso, e cioè l’installazione del nuovo *SQL Server 2008* sul nuovo server fisico, si sono verificate problematiche in parte complesse, su cui si è concentrata immediatamente l’attenzione e lo sforzo risolutivo, per non incidere in alcun modo sul lavoro produttivo giornaliero dell’Azienda. Si è inoltre dedicato uno studio particolare alle impostazioni e le configurazioni per le opzioni di memoria server da applicare alla nuova istanza di *SQL Server 2008* appena entrata in produzione, in modo da ottimizzare al meglio la potenza fisica espressa dal server e sfruttare appieno tutte le nuove features implementate dalla casa di Redmond sul nuovo *SQL Server 2008*.

In conclusione, al termine del progetto di tesi, ed alla luce di quanto emerso durante tutto il corso applicativo del progetto stesso, è doveroso sottolineare il vantaggio considerevole in termini di tempi di esecuzione che si è ottenuto con il nuovo *Microsoft SQL Server 2008*.

Tutto ciò, in aggiunta alla quasi assenza di problematiche strutturali tra i database migrati, alla velocità/facilità di migrazione attraverso la modalità di backup/restore di ciascun database prevista ed implementata con tool automatizzati all'interno di *SQL Server 2008*, alla presenza di problematiche più o meno onerose al termine della migrazione (peraltro prontamente affrontate e risolte), alle numerose opzioni di configurazione che la nuova piattaforma permette per lo sfruttamento intensivo della potenza fisica di ciascuna macchina, e tenuto altresì conto dei costi non altamente proibitivi sostenuti dall'Azienda per l'acquisto delle nuove licenze, porta a sostenere con forza l'utilità e la convenienza del passaggio da una vecchia versione di Microsoft SQL Server al nuovo *Microsoft SQL Server 2008*.

Nel caso specifico di VIDIERRE, il progetto di tesi ha permesso un nuovo passo in avanti nel progetto di innovazione strutturale intrapreso anni fa, consentendo all'Azienda di allinearsi ed aggiornarsi al mercato dei motori relazionali, nell'ambito della stessa casa produttrice, effettuando un upgrade da una versione ormai obsoleta all'ultima in commercio, senza scendere a compromessi con paventati downtime dei database o ipotetici fermi produttivi.

6. BIBLIOGRAFIA

- [1] Dayani – Fard, Müller, Mylopoulos, “Legacy Software Systems: Issues, Progress, and Challenges” – IBM Technical Report: TR-74.165-k, Aprile 1999,
<https://www-927.ibm.com/ibm/cas/publications/TR-74.165/k/legacy.pdf>
- [2] Virtualization. *VMware*. [Online] <http://www.vmware.com/virtualization/what-is-virtualization.html>.
- [3] Macchina Virtuale. *Wikipedia*. [Online] http://it.wikipedia.org/wiki/Virtual_machine.
- [4] Download center. *Microsoft*. [Online] <http://www.microsoft.com/downloads/it-it/details.aspx?FamilyID=f5a6c5e9-4cd9-4e42-a21c-7291e7f0f852>.
- [5] Sql Server Comparison Tool. [Online] AlfaAlfa. <http://www.sql-server-tool.com/>.
- [6] sys.sysconstraints (Transact-SQL). *MSDN Community*. [Online] <http://msdn.microsoft.com/en-us/library/ms190365.aspx>.
- [7] How to transfer logins and password between instances of SQL Server. *Microsoft Support*. [Online] <http://support.microsoft.com/kb/246133/en-us?p=1>.
- [8] Opzione AWE. *MSDN Community*. [Online] <http://msdn.microsoft.com/it-it/library/ms190731%28v=SQL.100%29.aspx>.
- [9] Server memory options. *MSDN Community*. [Online] <http://msdn.microsoft.com/it-it/library/ms178067%28v=SQL.100%29.aspx>.
- [10] Compatibility_level. *MSDN Community*. [Online] <http://msdn.microsoft.com/it-it/library/bb510680.aspx>.
- [11] Migrazione DTS. *MSDN Community*. [Online] <http://msdn.microsoft.com/it-it/library/ms143501.aspx>.
- [12] Backup differenziali. *Microsoft TechNet*. [Online] <http://technet.microsoft.com/it-it/library/ms175526%28SQL.90%29.aspx>.