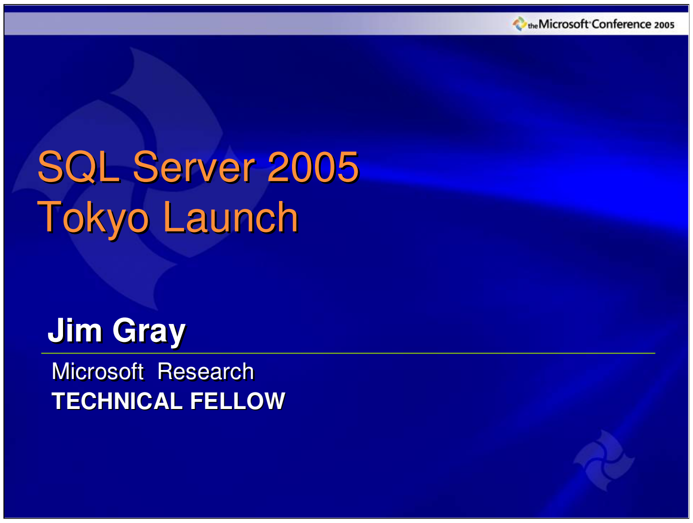
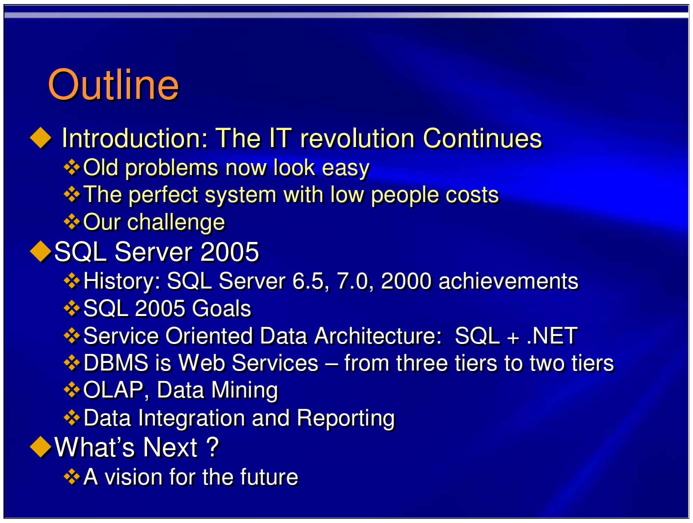


the Microsoft Conference 2005

SQL Server 2005
Tokyo Launch

Jim Gray

Microsoft Research
TECHNICAL FELLOW



Outline

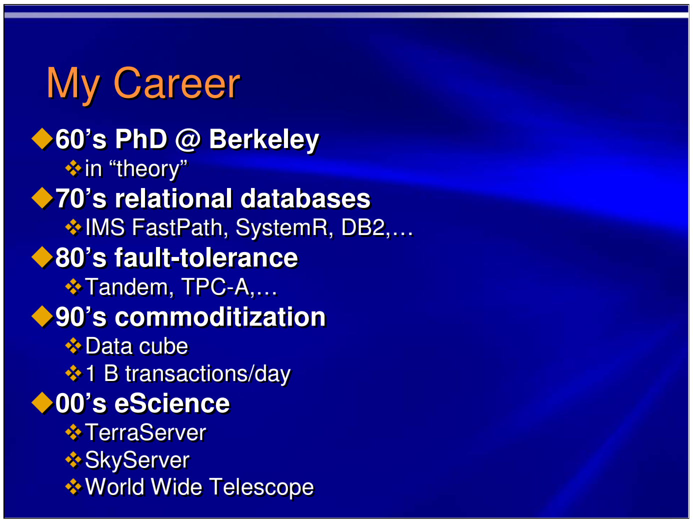
- Introduction: The IT revolution Continues
  - Old problems now look easy
  - The perfect system with low people costs
  - Our challenge
- SQL Server 2005
  - History: SQL Server 6.5, 7.0, 2000 achievements
  - SQL 2005 Goals
  - Service Oriented Data Architecture: SQL + .NET
  - DBMS is Web Services – from three tiers to two tiers
  - OLAP, Data Mining
  - Data Integration and Reporting
- What's Next ?
  - A vision for the future

Thank you very much for coming today.

I am proud to be here today.

I've worked with the SQL team for the last decade.

Today is the milestone we have worked towards for the last 5 years.



My Career

- **60’s PhD @ Berkeley**
  - in “theory”
- **70’s relational databases**
  - IMS FastPath, SystemR, DB2,…
- **80’s fault-tolerance**
  - Tandem, TPC-A,…
- **90’s commoditization**
  - Data cube
  - 1 B transactions/day
- **00’s eScience**
  - TerraServer
  - SkyServer
  - World Wide Telescope

You can see an outline of my career on this slide.

The common theme is that I have been trying to organize information and make it easy to access: ***Information at your fingertips.***

Lately I have been working to get all

science literature and data

integrated and online

and easily accessible.

SQL 2005 is my main tool.

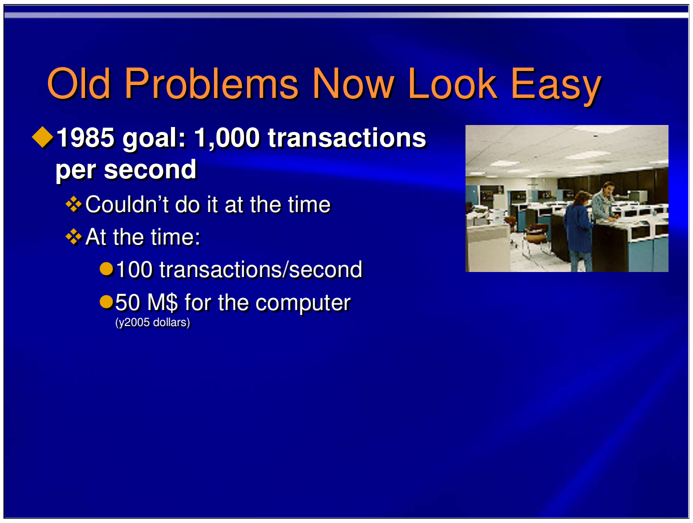
I have been using it for the last 5 years to build the world-wide telescope
and I am proud to have contributed some code to the product.

But mostly proud to be associated with the stellar team that built this breakthrough.

My goal in the next hour is to put SQL 2005 in perspective,

tell you about some of the things that I find most impressive,

and hint at what happens next.



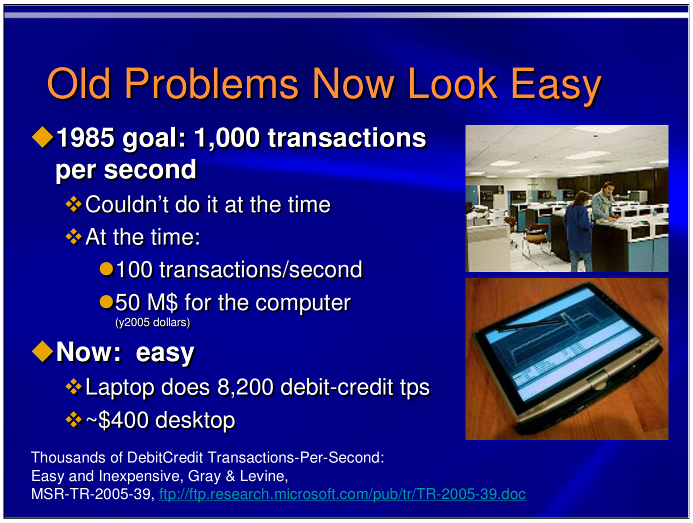
Old Problems Now Look Easy

- **1985 goal: 1,000 transactions per second**
  - Couldn't do it at the time
  - At the time:
    - 100 transactions/second
    - 50 M$ for the computer
      (y2005 dollars)

Old problems now look easy.

In 1985 the goal was to do a thousand transactions per second.

At the time, the biggest systems were only one-tenth that powerful.



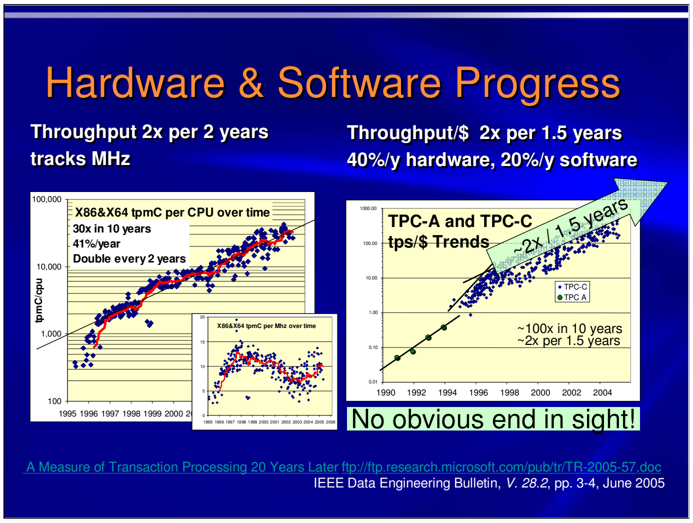
Old Problems Now Look Easy

- **1985 goal: 1,000 transactions per second**
  - Couldn’t do it at the time
  - At the time:
    - 100 transactions/second
    - 50 M$ for the computer (y2005 dollars)
- **Now: easy**
  - Laptop does 8,200 debit-credit tps
  - ~$400 desktop

Thousands of DebitCredit Transactions-Per-Second:
Easy and Inexpensive, Gray & Levine,
MSR-TR-2005-39, ftp://ftp.research.microsoft.com/pub/tr/TR-2005-39.doc

Now, 20 years later, my laptop can do 8 times more than the goal.

It could run the transaction load of all the 1980 US financial institutions.



Hardware & Software Progress

Throughput 2x per 2 years
tracks MHz

Throughput/$ 2x per 1.5 years
40%/y hardware, 20%/y software

No obvious end in sight!

A Measure of Transaction Processing 20 Years Later ftp://ftp.research.microsoft.com/pub/tr/TR-2005-57.doc
IEEE Data Engineering Bulletin, *V. 28.2*, pp. 3-4, June 2005

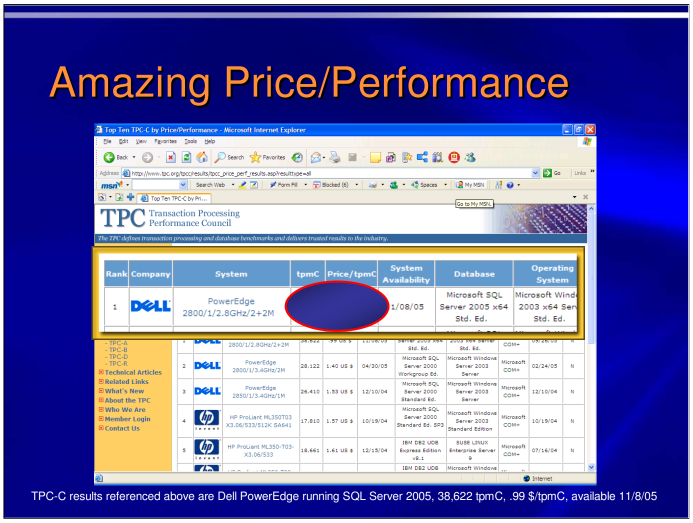
If you look at the progress over the last 25 years,
throughput has doubled every 18 months.

Much of this improvement has been due to hardware improvements,

but better software accounts for 20%/year.

There is no obvious end in sight.

In 1995 the peak performance was 20,000 tpmC
using 120 computers costing fifteen million dollars.



Amazing Price/Performance

Top Ten TPC-C by Price/Performance - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://www.tpc.org/tpcc/results/tpcc_price_perf_results.asp?resulttype=all
Go
Links
msn
Search Web
Form Fill
Blocked (6)
Spaces
My MSN
Top Ten TPC-C by Pri...
Go to My MSN.
TPC Transaction Processing Performance Council
The TPC defines transaction processing and database benchmarks and delivers trusted results to the industry.
- TPC-A
- TPC-B
- TPC-D
- TPC-R
⊞ Technical Articles
⊞ Related Links
⊞ What's New
⊞ About the TPC
⊞ Who We Are
⊞ Member Login
⊞ Contact Us
Internet

Rank	Company	System	tpmC	Price/tpmC	System Availability	Database	Operating System
1	DELL	PowerEdge 2800/1/2.8GHz/2+2M			1/08/05	Microsoft SQL Server 2005 x64 Std. Ed.	Microsoft Wind 2003 x64 Serv Std. Ed.

1	DELL	2800/1/2.8GHz/2+2M	38,622	.99 US $	11/08/05	Server 2005 x64 Std. Ed.	2003 x64 Server Std. Ed.	COM+	09/26/05	N
2	DELL	PowerEdge 2800/1/3.4GHz/2M	28,122	1.40 US $	04/30/05	Microsoft SQL Server 2000 Workgroup Ed.	Microsoft Windows Server 2003 Server	Microsoft COM+	02/24/05	N
3	DELL	PowerEdge 2850/1/3.4GHz/1M	26,410	1.53 US $	12/10/04	Microsoft SQL Server 2000 Standard Ed.	Microsoft Windows Server 2003 Server	Microsoft COM+	12/10/04	N
4	hp invent	HP ProLiant ML350T03 X3.06/533/512K SA641	17,810	1.57 US $	10/19/04	Microsoft SQL Server 2000 Standard Ed. SP3	Microsoft Windows Server 2003 Standard Edition	Microsoft COM+	10/19/04	N
5	hp invent	HP ProLiant ML350-T03-X3.06/533	18,661	1.61 US $	12/15/04	IBM DB2 UDB Express Edition v8.1	SUSE LINUX Enterprise Server 9	Microsoft COM+	07/16/04	N
	hp					IBM DB2 UDB	Microsoft Windows			

TPC-C results referenced above are Dell PowerEdge running SQL Server 2005, 38,622 tpmC, .99 $/tpmC, available 11/8/05

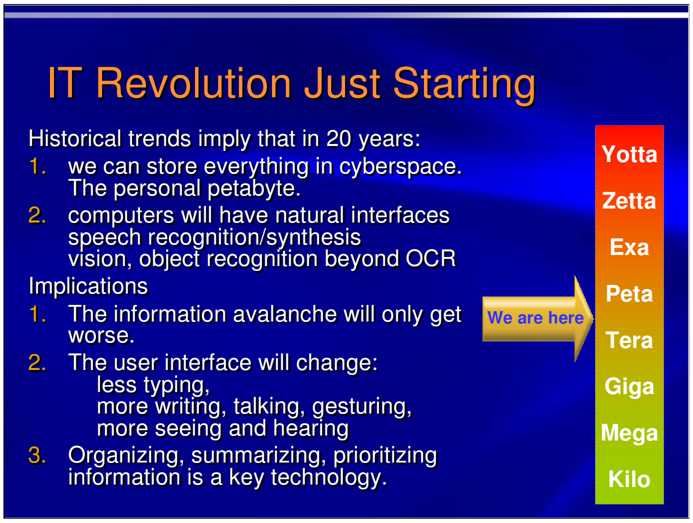
Last Month, SQL Server 2005 and Dell delivered the next point in that graph.

Breaking the 1$ per tpmC.

In ten years 1 yen per tpmc.

The numbers are astronomical.

Few companies run a million transactions per minute,
yet that is what SQL Server on an HP Superdome can deliver.



IT Revolution Just Starting

Historical trends imply that in 20 years:

1. we can store everything in cyberspace.
   The personal petabyte.
2. computers will have natural interfaces
   speech recognition/synthesis
   vision, object recognition beyond OCR

Implications

1. The information avalanche will only get worse.
2. The user interface will change:
   less typing,
   more writing, talking, gesturing,
   more seeing and hearing
3. Organizing, summarizing, prioritizing information is a key technology.

We are here

Yotta
Zetta
Exa
Peta
Tera
Giga
Mega
Kilo

So, old problems look easy.

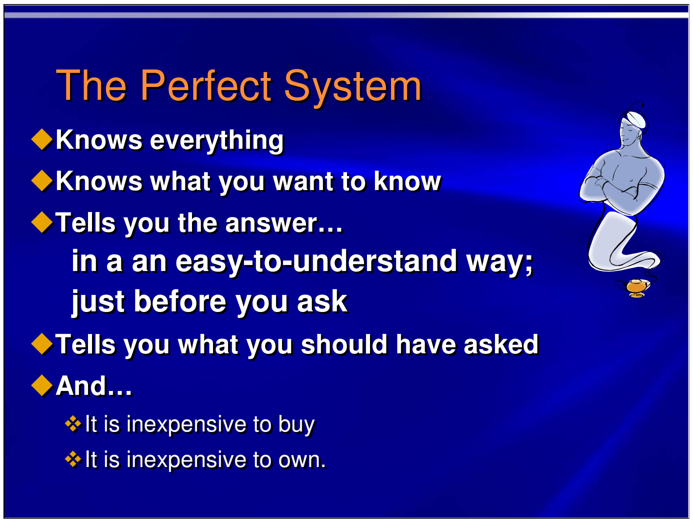
Now things are starting to get interesting.

Now we can tackle the really hard problems.

Organizing and analyzing information –
augmenting human intelligence;

making us smarter,

making organizations more agile.



The Perfect System

- Knows everything
- Knows what you want to know
- Tells you the answer…
  in a an easy-to-understand way;
  just before you ask
- Tells you what you should have asked
- And…
  - It is inexpensive to buy
  - It is inexpensive to own.

We want to build information systems that

know everything

know what we want to know

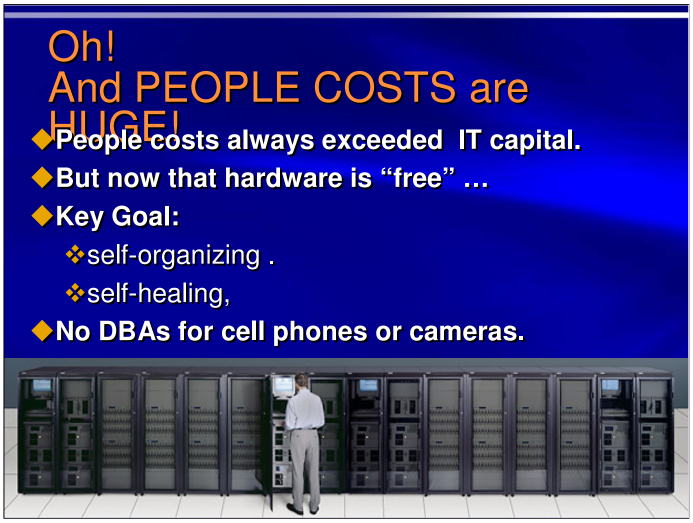
tell us the answer in an understandable way

just before we ask the question.

Indeed, we want the system to tell us

the question we should have asked

and the answer.



Oh!
And PEOPLE COSTS are HUGE!

- People costs always exceeded IT capital.
- But now that hardware is “free” …
- Key Goal:
  - self-organizing .
  - self-healing,
- No DBAs for cell phones or cameras.

Oh! And we want the system to be inexpensive to buy and to own.

Which means that it has to be

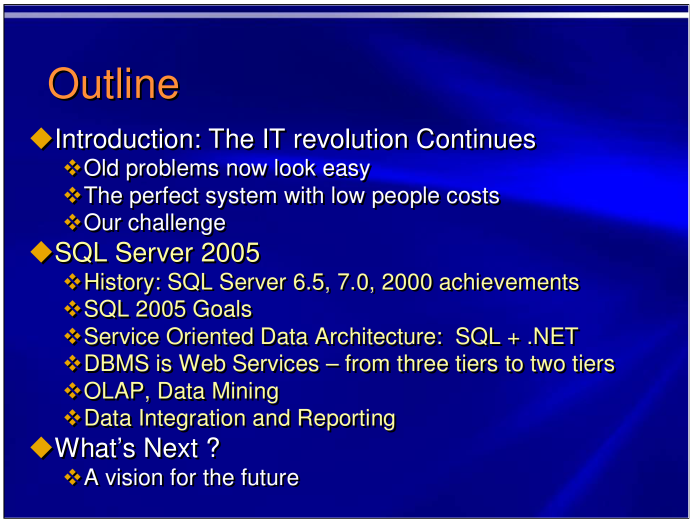
self-organizing and

self-healing.

There has been a lot of progress,

But we are a long way from such self-managing mega-serves.

So, the revolution is just starting

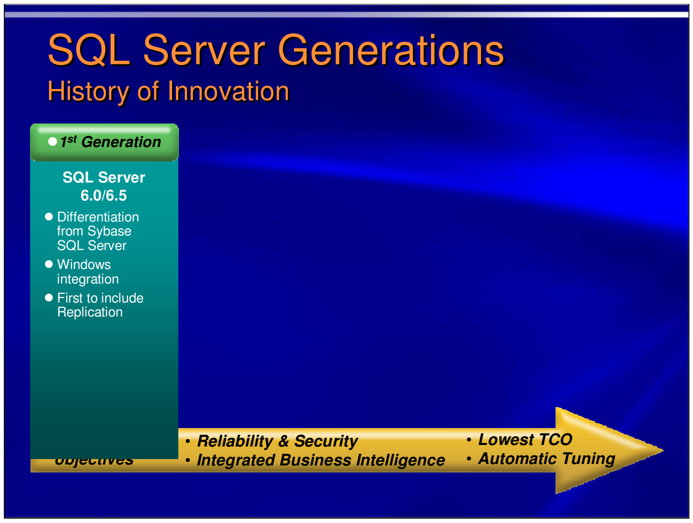


Outline

- Introduction: The IT revolution Continues
  - Old problems now look easy
  - The perfect system with low people costs
  - Our challenge
- SQL Server 2005
  - History: SQL Server 6.5, 7.0, 2000 achievements
  - SQL 2005 Goals
  - Service Oriented Data Architecture: SQL + .NET
  - DBMS is Web Services – from three tiers to two tiers
  - OLAP, Data Mining
  - Data Integration and Reporting
- What’s Next ?
  - A vision for the future

But, lets get real.

We are here to talk about SQL Server 2005.



SQL Server Generations

History of Innovation

- 1st Generation

SQL Server 6.0/6.5

- Differentiation from Sybase SQL Server
- Windows integration
- First to include Replication

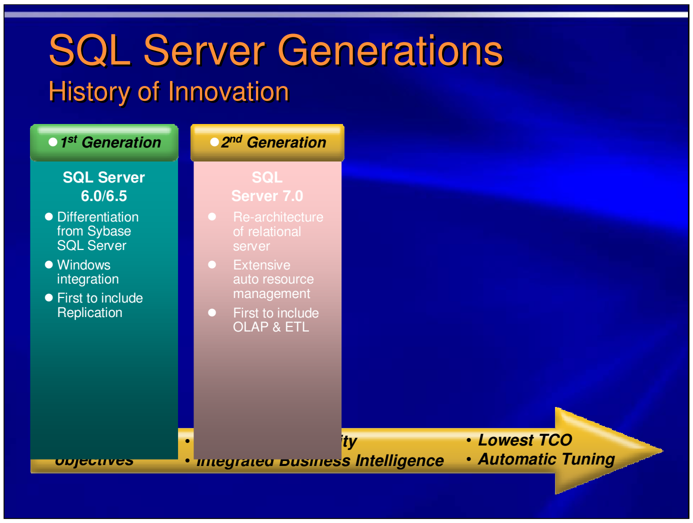
objectives

- Reliability & Security
- Integrated Business Intelligence
- Lowest TCO
- Automatic Tuning

It’s best to start with a historical perspective.

In the mid-90’s SQL Server focused on
cleaning up the code base
integrating with Windows Security, threads and management,

It was first to have replication built into the product
and continues to have the easiest database replication.



SQL Server Generations

History of Innovation

1st Generation

SQL Server 6.0/6.5

- Differentiation from Sybase SQL Server
- Windows integration
- First to include Replication

2nd Generation

SQL Server 7.0

- Re-architecture of relational server
- Extensive auto resource management
- First to include OLAP & ETL

objectives

- *ity*
- *Integrated Business Intelligence*

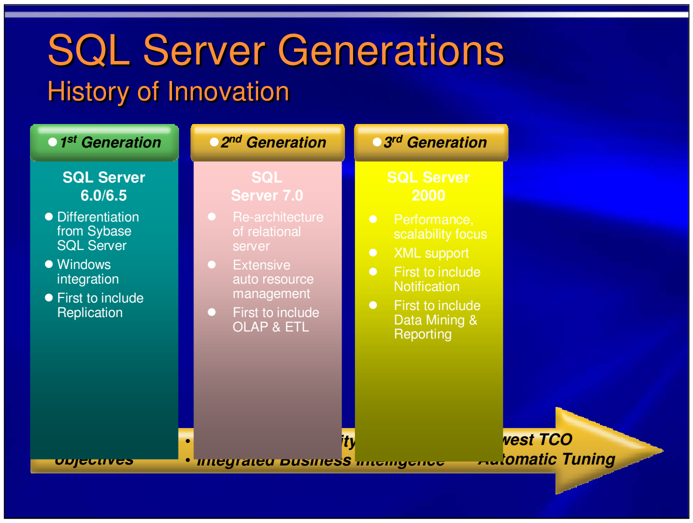
- *Lowest TCO*
- *Automatic Tuning*

Then came SQL 7.0 with a re-architected kernel.

It introduced OLAP tools
with a great data cube implementation,

had auto-administration features,

and included Extract-Transform-Load tools.



SQL Server Generations

History of Innovation

- 1st Generation

SQL Server 6.0/6.5

- Differentiation from Sybase SQL Server
- Windows integration
- First to include Replication

- 2nd Generation

SQL Server 7.0

- Re-architecture of relational server
- Extensive auto resource management
- First to include OLAP & ETL

- 3rd Generation

SQL Server 2000

- Performance, scalability focus
- XML support
- First to include Notification
- First to include Data Mining & Reporting

objectives

- *ity*
- *Integrated Business Intelligence*

west TCO
Automatic Tuning

Then came SQL 2000.

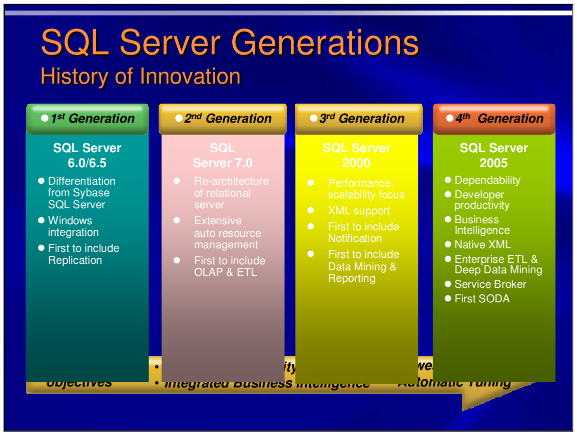
It improved performance with a new lock and log design.

It was our initial installment on XML support.
The XML support evolved with several web releases since then.

It included notification server that makes SQL Server an “active database” that triggers external actions when the database changes.

These were all great things,
but the real breakthrough in my opinion was
the data mining and reporting built into the product.

This was just a down payment, but the Business Intelligence tools have been a huge asset to our customers.



SQL Server Generations

History of Innovation

1st Generation

SQL Server 6.0/6.5

- Differentiation from Sybase SQL Server
- Windows integration
- First to include Replication

2nd Generation

SQL Server 7.0

- Re-architecture of relational server
- Extensive auto resource management
- First to include OLAP & ETL

3rd Generation

SQL Server 2000

- Performance, scalability focus
- XML support
- First to include Notification
- First to include Data Mining & Reporting

4th Generation

SQL Server 2005

- Dependability
- Developer productivity
- Business Intelligence
- Native XML
- Enterprise ETL & Deep Data Mining
- Service Broker
- First SODA

objectives

- Integrated Business Intelligence

Automatic Tuning

Now comes SQL 2005.

This is our first SQL release since Bill Gate's
Trustworthy Computing memo.

That memo stopped ALL development in Microsoft for several months while we all got trained on building dependable systems.

Then the Slammer virus hit.

Bill Gates' memo and 2 months of training was good for us.
Slammer was a nightmare – and it transformed us.

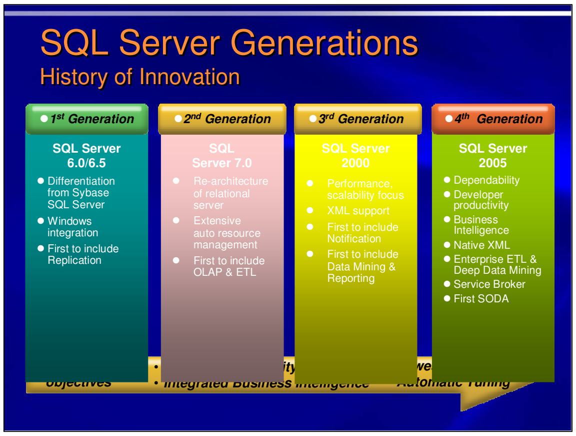
Now "dependable by design, by default, and by deployment" is our mantra.

SECURITY now dominates all our design decisions.

This is also the first SQL release to embody the .NET architecture,

as you will see, everything is now built around XML and web services.

We have been working for 5 years to make SQL Server 2005 more dependable.
We have also been working to make it really cool.
The integration of SQL with Visual Studio that you heard about from Steve Ballmer
is a cool thing for us developers.   I love it.



SQL Server Generations

History of Innovation

- 1st Generation
  SQL Server 6.0/6.5
  - Differentiation from Sybase SQL Server
  - Windows integration
  - First to include Replication
- 2nd Generation
  SQL Server 7.0
  - Re-architecture of relational server
  - Extensive auto resource management
  - First to include OLAP & ETL
- 3rd Generation
  SQL Server 2000
  - Performance, scalability focus
  - XML support
  - First to include Notification
  - First to include Data Mining & Reporting
- 4th Generation
  SQL Server 2005
  - Dependability
  - Developer productivity
  - Business Intelligence
  - Native XML
  - Enterprise ETL & Deep Data Mining
  - Service Broker
  - First SODA

objectives • Integrated Business Intelligence Automatic Tuning

The Business intelligence tools,
the many new data mining algorithms,
the unified dimension model, and the ETL engine
are must-haves for knowledge workers.

XML and semi-structured data is a key part of SQL 2005.
XML is now a native data type.
The XQUERY language is integrated with SQL.
You can mix SQL data, XML data, and text data all in one statement.

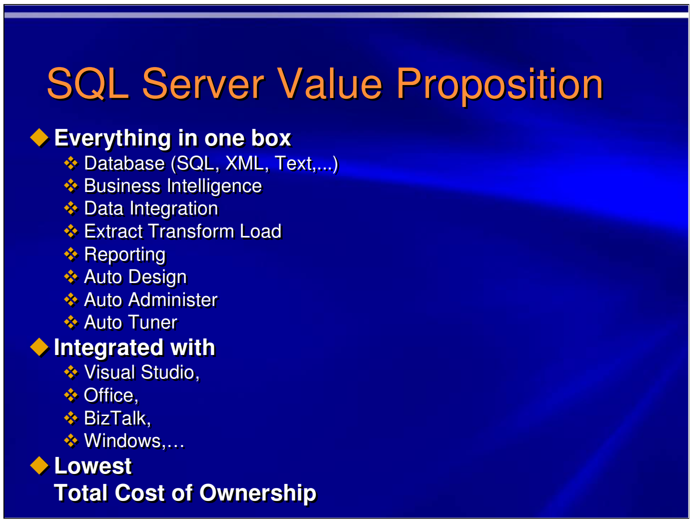
A new reporting service that lets you build data dashboards for everyone in the organization,
Now they can see what they want in the very graphical way they want to see it.

SQL 2005 is a web service container that lets you easily publish web services.
A built-in queue manager that revolutionizes the way you build web services.

We call this new application architecture: Service Oriented Data Architectures or SODA.
It’s a breakthrough in how we conceptualize, build and deploy applications.

SQL 2005 is introducing it TODAY.

Throughout all this, the goal has been lowest TCO.



SQL Server Value Proposition

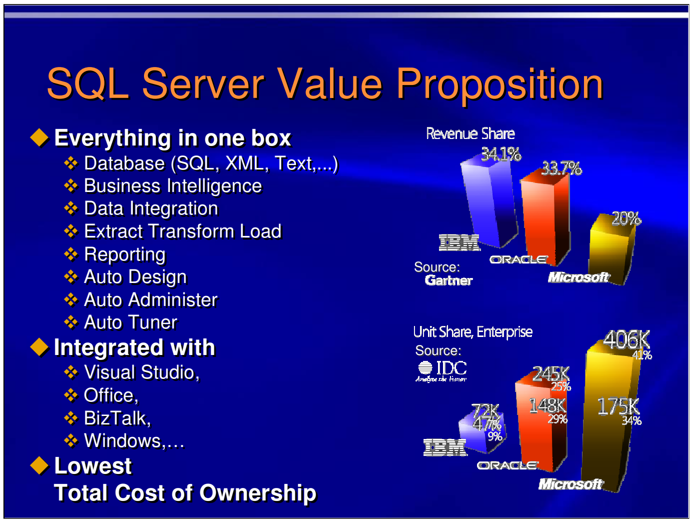
- **Everything in one box**
  - Database (SQL, XML, Text,...)
  - Business Intelligence
  - Data Integration
  - Extract Transform Load
  - Reporting
  - Auto Design
  - Auto Administer
  - Auto Tuner
- **Integrated with**
  - Visual Studio,
  - Office,
  - BizTalk,
  - Windows,…
- **Lowest Total Cost of Ownership**

The SQL Server value proposition is that you get everything in one box – no “add-ins.”

You get SQL, XML, Text and other data engines.
You get OLAP, Cubes,

You get data mining.

You get data integration including extract-transform-load.
You get reporting.
The system has a ton of auto-design, auto-admin, and auto-tuning features.
It is integrated with the rest of the Microsoft products.



SQL Server Value Proposition

- **Everything in one box**
  - Database (SQL, XML, Text,...)
  - Business Intelligence
  - Data Integration
  - Extract Transform Load
  - Reporting
  - Auto Design
  - Auto Administer
  - Auto Tuner
- **Integrated with**
  - Visual Studio,
  - Office,
  - BizTalk,
  - Windows,…
- **Lowest Total Cost of Ownership**

And it has the lowest total cost of ownership.

No extra charge for 64-bit.

With the advent of multi-core chips,
Microsoft is leading the per-socket pricing
rather than the pr-core pricing of some other software venders.

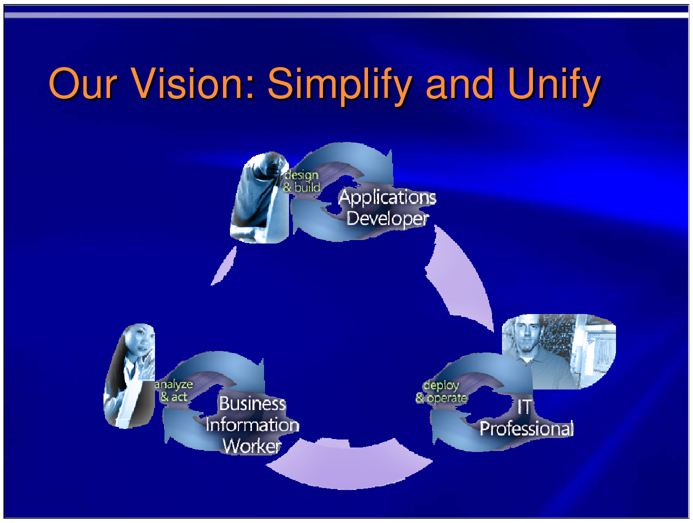
It is hard to quantify TCO.
We have studies that show SQL Server systems are much less expensive to own.

But it ***is*** easy to quantify prices, so let's at least do that.
The graph at the upper right shows that SQLserver gets less than 2/3 the revenue that Oracle or IBM gets.

But the bottom box shows it has 2 to 5 times as many systems
in both the enterprise category and overall.

So, SQL Server delivers MANY MANY more systems
for about ½ the revenue.

That means a LOT of customer savings.



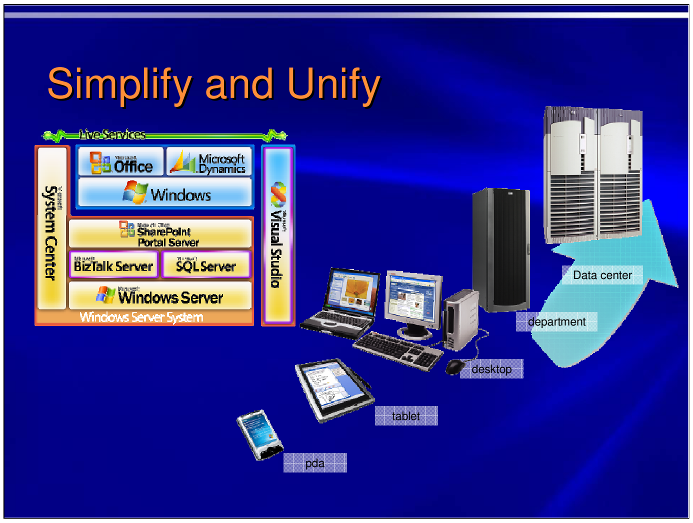
Our Vision: Simplify and Unify

But TCO is the real issue.

The idea is that everyone in the organization can benefit from SQL Server 2005.

It eliminates a lot of the drudgery.

It automates tasks.



Simplify and Unify

Live Services

Office

Microsoft Dynamics

Windows

SharePoint Portal Server

BizTalk Server

SQL Server

Windows Server

Windows Server System

System Center

Visual Studio

Data center

department

desktop

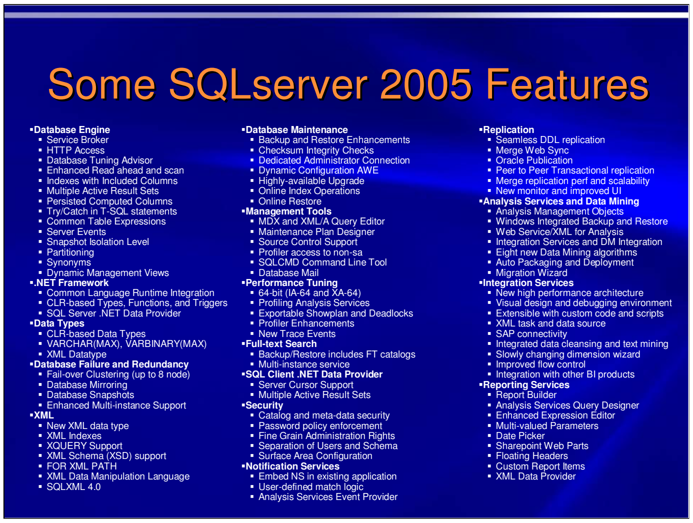
tablet

pda

And it works with all the other Microsoft products, leveraging your skills.

It works on PDAs, on tablets, on departmental servers and in data centers.

It works on Intel and AMD 64-bit platforms
with no change to your programs or data.



Some SQLserver 2005 Features

- **Database Engine**
  - Service Broker
  - HTTP Access
  - Database Tuning Advisor
  - Enhanced Read ahead and scan
  - Indexes with Included Columns
  - Multiple Active Result Sets
  - Persisted Computed Columns
  - Try/Catch in T-SQL statements
  - Common Table Expressions
  - Server Events
  - Snapshot Isolation Level
  - Partitioning
  - Synonyms
  - Dynamic Management Views
- **.NET Framework**
  - Common Language Runtime Integration
  - CLR-based Types, Functions, and Triggers
  - SQL Server .NET Data Provider
- **Data Types**
  - CLR-based Data Types
  - VARCHAR(MAX), VARBINARY(MAX)
  - XML Datatype
- **Database Failure and Redundancy**
  - Fail-over Clustering (up to 8 node)
  - Database Mirroring
  - Database Snapshots
  - Enhanced Multi-instance Support
- **XML**
  - New XML data type
  - XML Indexes
  - XQUERY Support
  - XML Schema (XSD) support
  - FOR XML PATH
  - XML Data Manipulation Language
  - SQLXML 4.0
- **Database Maintenance**
  - Backup and Restore Enhancements
  - Checksum Integrity Checks
  - Dedicated Administrator Connection
  - Dynamic Configuration AWE
  - Highly-available Upgrade
  - Online Index Operations
  - Online Restore
- **Management Tools**
  - MDX and XML/A Query Editor
  - Maintenance Plan Designer
  - Source Control Support
  - Profiler access to non-sa
  - SQLCMD Command Line Tool
  - Database Mail
- **Performance Tuning**
  - 64-bit (IA-64 and XA-64)
  - Profiling Analysis Services
  - Exportable Showplan and Deadlocks
  - Profiler Enhancements
  - New Trace Events
- **Full-text Search**
  - Backup/Restore includes FT catalogs
  - Multi-instance service
- **SQL Client .NET Data Provider**
  - Server Cursor Support
  - Multiple Active Result Sets
- **Security**
  - Catalog and meta-data security
  - Password policy enforcement
  - Fine Grain Administration Rights
  - Separation of Users and Schema
  - Surface Area Configuration
- **Notification Services**
  - Embed NS in existing application
  - User-defined match logic
  - Analysis Services Event Provider
- **Replication**
  - Seamless DDL replication
  - Merge Web Sync
  - Oracle Publication
  - Peer to Peer Transactional replication
  - Merge replication perf and scalability
  - New monitor and improved UI
- **Analysis Services and Data Mining**
  - Analysis Management Objects
  - Windows Integrated Backup and Restore
  - Web Service/XML for Analysis
  - Integration Services and DM Integration
  - Eight new Data Mining algorithms
  - Auto Packaging and Deployment
  - Migration Wizard
- **Integration Services**
  - New high performance architecture
  - Visual design and debugging environment
  - Extensible with custom code and scripts
  - XML task and data source
  - SAP connectivity
  - Integrated data cleansing and text mining
  - Slowly changing dimension wizard
  - Improved flow control
  - Integration with other BI products
- **Reporting Services**
  - Report Builder
  - Analysis Services Query Designer
  - Enhanced Expression Editor
  - Multi-valued Parameters
  - Date Picker
  - Sharepoint Web Parts
  - Floating Headers
  - Custom Report Items
  - XML Data Provider

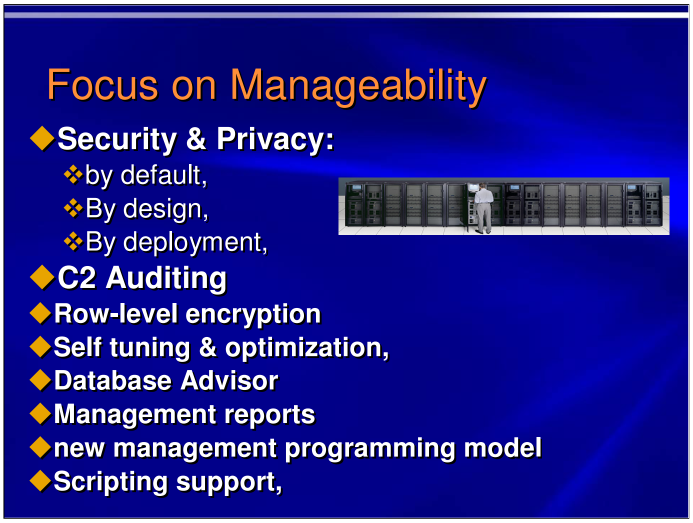
SQL Server represents the effort of over 1,000 developers working for over 5 years.

So, you can imagine that there many new features.

Here are a hundred of them.

I’d love to cover them all in the next ½ hour.

But, I have to focus on just a few.



Focus on Manageability

- **Security & Privacy:**
  - by default,
  - By design,
  - By deployment,
- **C2 Auditing**
- **Row-level encryption**
- **Self tuning & optimization,**
- **Database Advisor**
- **Management reports**
- **new management programming model**
- **Scripting support,**

SQL Server has the lowest TCO, but we want to do better.

As I said before, it is secure by design, by default, and by deployment.

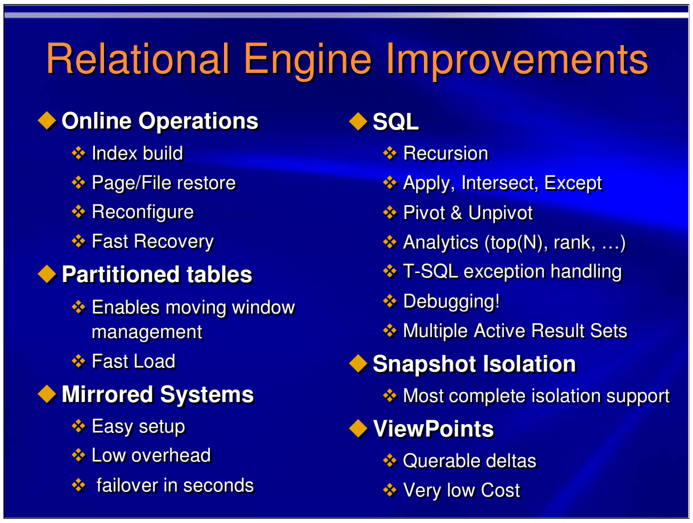
It allows C2 auditing of all data accesses.

It offers row-level encryption.

It self-tunes and can generate many reports.

It has a more complete management object model,

that lets your scripts automate standard tasks.



Relational Engine Improvements

- **Online Operations**
  - Index build
  - Page/File restore
  - Reconfigure
  - Fast Recovery
- **Partitioned tables**
  - Enables moving window management
  - Fast Load
- **Mirrored Systems**
  - Easy setup
  - Low overhead
  - failover in seconds
- **SQL**
  - Recursion
  - Apply, Intersect, Except
  - Pivot & Unpivot
  - Analytics (top(N), rank, …)
  - T-SQL exception handling
  - Debugging!
  - Multiple Active Result Sets
- **Snapshot Isolation**
  - Most complete isolation support
- **ViewPoints**
  - Querable deltas
  - Very low Cost

Just to highlight a few points in the core SQL Server engine.

Traditionally many operations were online
but this release focused on the main causes of operations downtime.

All the DDL operations are now online –
including index build and page or file granularity recovery.

Tables can be partitioned by a key;
you can load a partition offline and then add it to the table.
And you can remove a partition from the table in less than a second.

This allows easy “moving-window” data management.

Mirrored systems let you geo-plex your database for very high availability with failover in a few seconds.

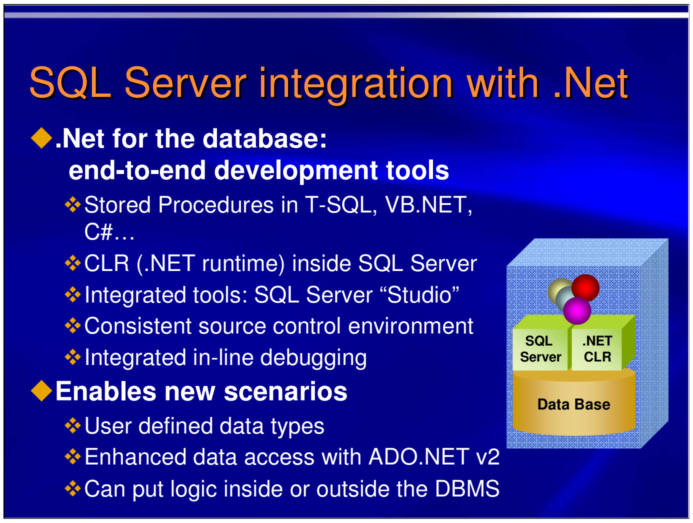
It adds snapshot isolation (something Oracle calls Consistent Read)

so now SQL Server has the most complete isolation support of any product.

It also supports point in time snapshots, called viewpoints, for reporting.

The SQL language has many important features added.

The online debugger works across all languages.



SQL Server integration with .Net

- **.Net for the database: end-to-end development tools**
  - Stored Procedures in T-SQL, VB.NET, C#…
  - CLR (.NET runtime) inside SQL Server
  - Integrated tools: SQL Server “Studio”
  - Consistent source control environment
  - Integrated in-line debugging
- **Enables new scenarios**
  - User defined data types
  - Enhanced data access with ADO.NET v2
  - Can put logic inside or outside the DBMS

The BIG event for SQL server users is the integration with
the .NET common language runtime and with Visual Studio.

Database developers have been plagued
by the “impedance mismatch” between programming languages and databases.

This gave rise many “object oriented” databases and to Object Relational databases.

SQL Server 2005 is the first system to integrate the programming language runtime with the database runtime.

Now there is no inside outside dichotomy.

SQL datatypes are .NET datatypes are SQL datatypes.

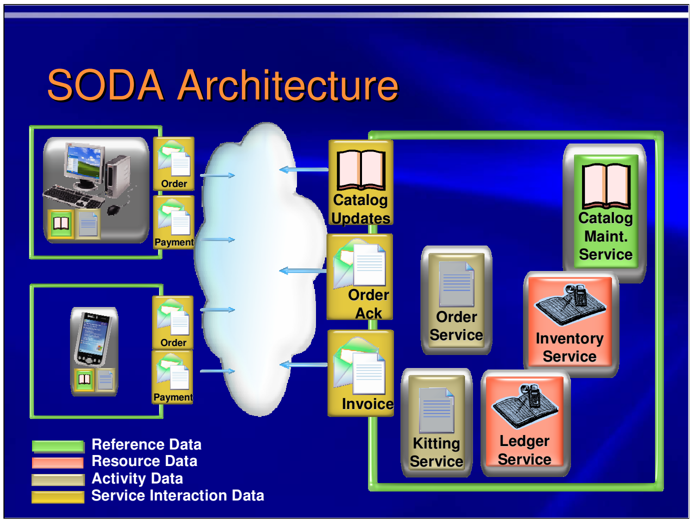
You can write your stored procedures in any .NET language – not just transactSQL, but also VB or C++ or C# or whatever.

And, Transact SQL has been improved as a language with many new features, including exception handling.

You get all the great VisualStudio tools like
intelisense,
integrated debugging,
source code control, and
project management

You can define your own types and classes
and you can define new aggregate functions using any language you like.

SODA Architecture



🟩 Reference Data
🟥 Resource Data
▨ Activity Data
🟨 Service Interaction Data

SQL Server CLR integration revolutionizes the programming model;

but the larger revolution is Service Oriented Data Architectures.

CLR lets you encapsulate data.

But SQL 2005 lets you publish that encapsulated data as a Web Service
-- a Service Oriented Architecture (SOA).

But this is not just SOA – it is Service Oriented ***Data*** Architecture.

SODA postulates four data classes:

Resource data that is the “master data” for the enterprise.

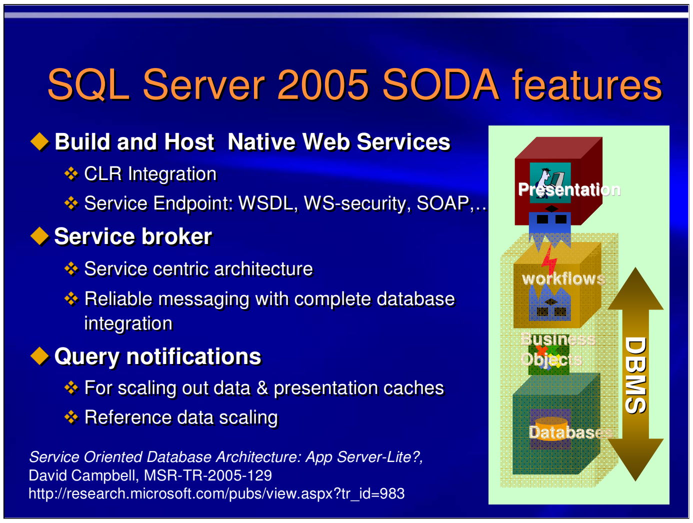
Reference data that is a copy of the resource data and widely distributed in caches.

Service data are the messages exchanged among clients and services

Activity data records what happened

SODA uses this data model to design federated data services that are
Event Driven
Loosely coupled
and Heterogeneous

All communication is via Service Data Messages



Presentation
workflows
Business Objects
Databases
DBMS

SQL Server 2005 SODA features

- **Build and Host Native Web Services**
  - CLR Integration
  - Service Endpoint: WSDL, WS-security, SOAP,…
- **Service broker**
  - Service centric architecture
  - Reliable messaging with complete database integration
- **Query notifications**
  - For scaling out data & presentation caches
  - Reference data scaling

Service Oriented Database Architecture: App Server-Lite?,
David Campbell, MSR-TR-2005-129
http://research.microsoft.com/pubs/view.aspx?tr_id=983

SODA is an Asynchronous, Distributed, Decoupled programming environment.

SQL 2005 integrated SODA into

the database engine
the programming model,
and the management interfaces.

The main features are:

SQL 2005 is a **web service container** that fully understands SOAP and WSDL and WS-Security

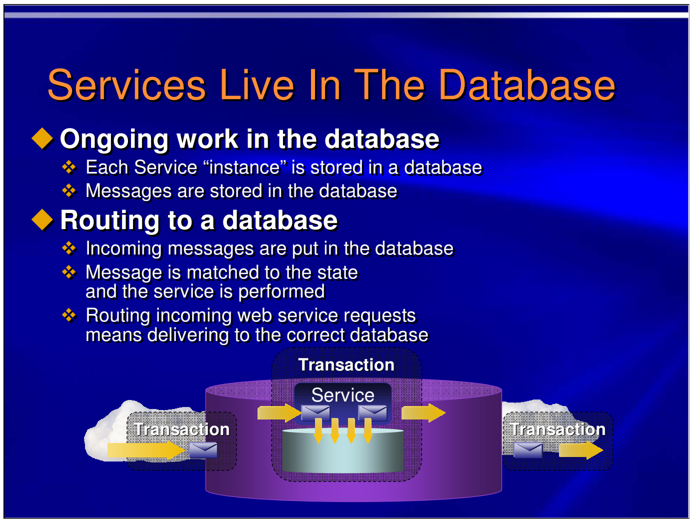
Service Broker makes it easy to write message-oriented loosely coupled applications.

Notification Service makes it easy to manage caches of reference data.

The next two slides walk thru two scenarios.

But I really encourage you to look at David Campbell's excellent paper describing SODA concepts and how SQL Server 2005 implements them.

A Japanese translation of the paper available later today



Services Live In The Database

- Ongoing work in the database
  - Each Service “instance” is stored in a database
  - Messages are stored in the database
- Routing to a database
  - Incoming messages are put in the database
  - Message is matched to the state and the service is performed
  - Routing incoming web service requests means delivering to the correct database

Transaction
Service
Transaction
Transaction

Let’s first look at a simple web service that lives inside the database.

SOAP messages arrive described by an XML Schema contract

SQL Server manages the interface contract,

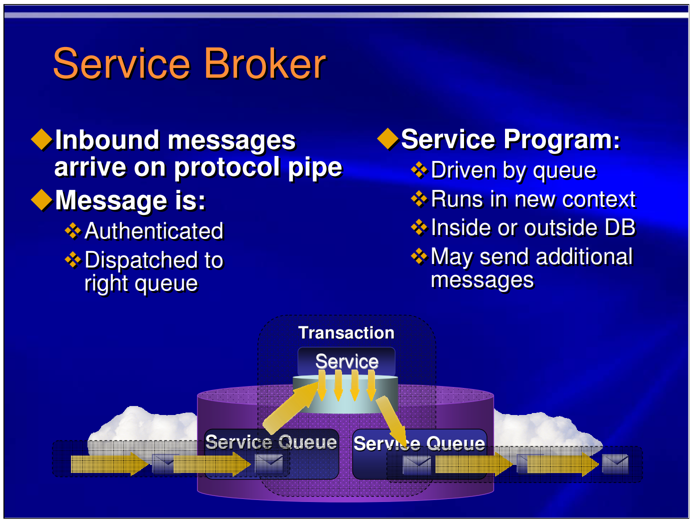
does the WS-security checks,
does the XML schema validation,

and places the message in a queue.

Now the message is processed by a service transaction
and produces one or more SOAP messages
that are routed to the original requestor or to other services.

The delivery of each of these messages is a separate transaction.

Some of these messages may be Notification Services messages.



Service Broker

- Inbound messages arrive on protocol pipe
- Message is:
  - Authenticated
  - Dispatched to right queue
- Service Program:
  - Driven by queue
  - Runs in new context
  - Inside or outside DB
  - May send additional messages

Service Broker enforces an XML service contract.

It reliably delivers messages and it manages message flows.

It's the basis for building very sophisticated workflows.

Click
Here we see a message arrive.

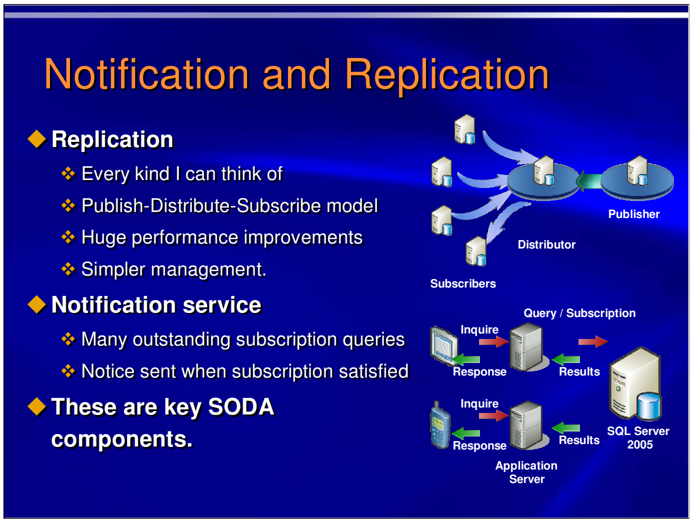
It sits in a queue till Service Broker dispatches the service.

Click
The service could be a stored procedure as in the previous slide
or it could be a process or service outside SQL Server.

Click

Typically, the service produces new messages which are delivered by service broker.

Again, I can only give you a brief introduction to SODA here –
do read Dave Campbell's article get the full story.



Notification and Replication

- **Replication**
  - Every kind I can think of
  - Publish-Distribute-Subscribe model
  - Huge performance improvements
  - Simpler management.
- **Notification service**
  - Many outstanding subscription queries
  - Notice sent when subscription satisfied
- **These are key SODA components.**

SQL Server has almost every kind of data replication I know of.

It has merge replication, transactional replication, log shipping, and so on.

It replicates data and schema changes.

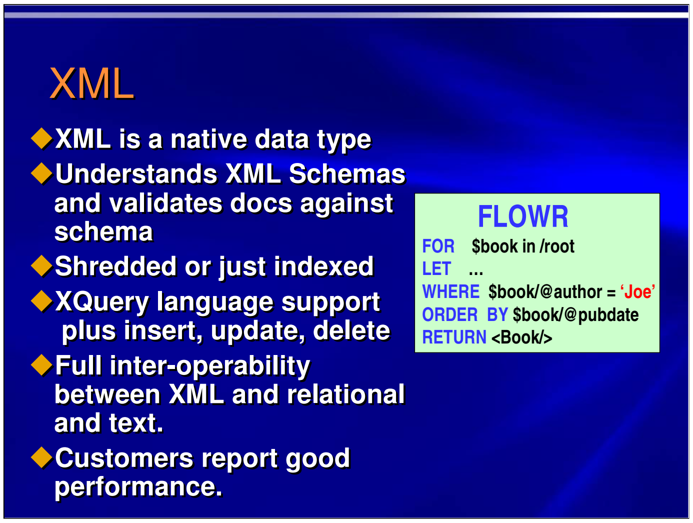
SQL 2005 adds Secure Socket replication for mobile devices over the internet.

It replicates from Oracle and other data engines.

The distinguishing characteristic is how easy it is to configure and manage.

SQL Server also has a built-in Notification service.
You can register a “standing Query” against the database.
When data changes satisfy this query, you get a notification.

This is a key component of maintaining reference data in a SODA architecture, and we used the same mechanisms to maintain our BI data cubes and caches.



XML

- XML is a native data type
- Understands XML Schemas and validates docs against schema
- Shredded or just indexed
- XQuery language support plus insert, update, delete
- Full inter-operability between XML and relational and text.
- Customers report good performance.

FLOWR

```
FOR    $book in /root
LET    ...
WHERE  $book/@author = 'Joe'
ORDER  BY $book/@pubdate
RETURN <Book/>
```

ALL these external interfaces use XML and SODA is based on XML.

There is no doubt that XML is the lingua-franca for data on the outside.

But, SQL Server also assumes that most data will be semi structured.

SQL Sever 2000 had basic support for XML data as a blob.

But, SQL 2005 supports XML as a ***native*** datatype.

If the data has a schema, SQL Server will store the data in a very efficient format.

This makes element and attribute searches and updates very fast.

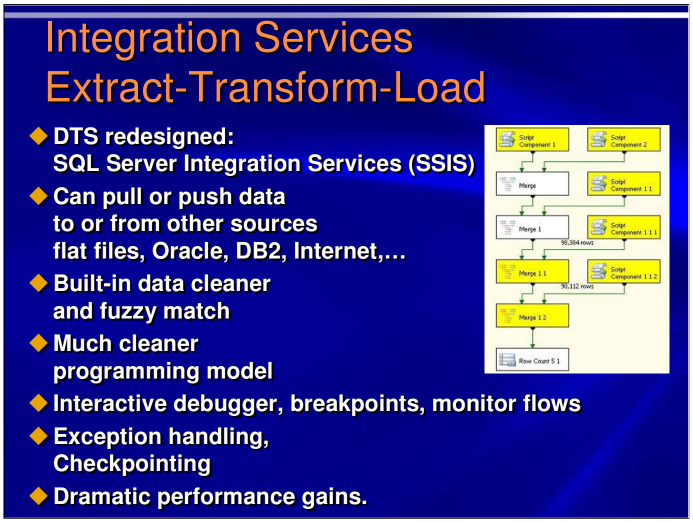
SQL 2005 integrates XQUERY with the SQL.
XQuery has been extended to include inserts, updates, and deletes.

There is full inter-operability between xml data, xml data definition, xml query and the corresponding non-xml data and access methods.

SQL Server 2005 uses XML extensively to externalize its own semi-structured data like query plans, configuration data, and trace files.

I have talked to many groups who have used these XML facilities and who are absolutely delighted with the functionality and performance.

There is a lot of buzz about this.



Integration Services Extract-Transform-Load

- DTS redesigned: SQL Server Integration Services (SSIS)
- Can pull or push data to or from other sources flat files, Oracle, DB2, Internet,…
- Built-in data cleaner and fuzzy match
- Much cleaner programming model
- Interactive debugger, breakpoints, monitor flows
- Exception handling, Checkpointing
- Dramatic performance gains.

XML is just one of the many datatypes you have to deal with in doing an application.

Most data comes from elsewhere.

SQL Server Integration Services can pull data
from production systems,
from files,
from foreign databases,
from legacy systems,
and from the web and web services.

It gives you a GUI and programmatic interface
to build data pipelines that scrub the data,
to reorganize it,
and then to insert it in your data warehouse.

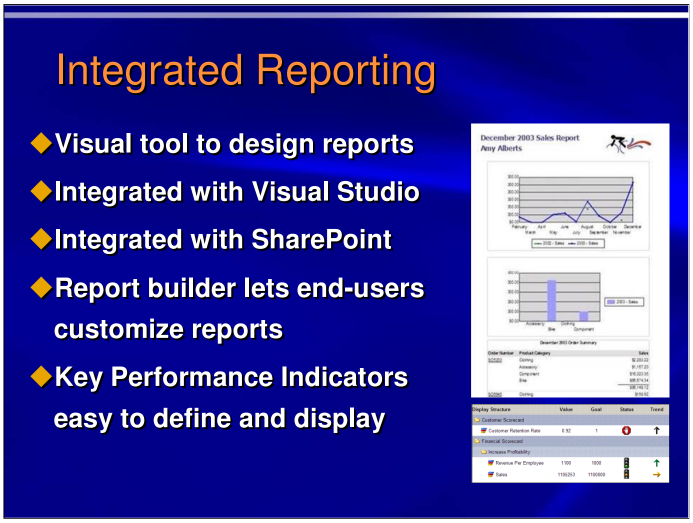
It has many tools for these data cleaning tasks.

It also has an elegant debugging environment, integrated with SQL Workbench,

that lets you set breakpoints on the code and lets you watch data flow through the pipeline.

It has checkpoint-restart and exception handling which are essential for big jobs.

And, at least for my applications,
its built-in parallelism is ten times faster than the DTS equivalent pipeline.



Integrated Reporting

- ◆Visual tool to design reports
- ◆Integrated with Visual Studio
- ◆Integrated with SharePoint
- ◆Report builder lets end-users customize reports
- ◆Key Performance Indicators easy to define and display

December 2003 Sales Report
Amy Alberts

December 2003 Order Summary

Order Number	Product Category	Sales
SO5202	Clothing	$2,283.22
	Accessory	$1,167.20
	Component	$16,023.38
	Bike	$98,674.34
		$98,148.12
SO5565	Clothing	$158.92

Display Structure	Value	Goal	Status	Trend
Customer Scorecard				
Customer Retention Rate	0.92	1	●	↑
Financial Scorecard				
Increase Profitability				
Revenue Per Employee	1100	1000	●	↑
Sales	1105253	1100000	●	→

SQL Server includes a reporting service with design tools
integrated with the rest of Visual Studio and the SQL Server tools.

It can generate reports from all the SQL Server data sources
-- tables, cubes, data mining, XML, web services, or any data source.

End users can customize these reports as a digital dashboard.

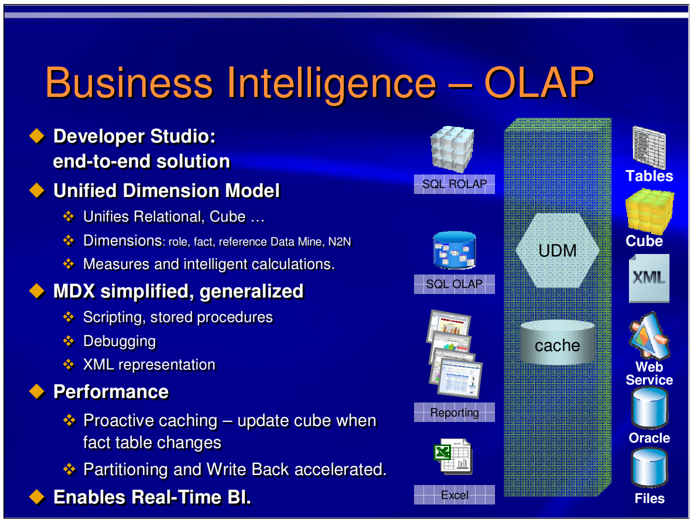
Something that is very popular is the ability to define
key performance indicators and icons or colors to highlight them.

And, these are not static reports, they are fully interactive and real-time –
on the intra-net and on the Internet.

You saw some of that in Steve Blamers' talk.

This reporting service is used extensively within SQL Server’s management
interface to show system dynamics.

It is also being used by applications like SharePoint and Office 12.



Business Intelligence – OLAP

- **Developer Studio: end-to-end solution**
- **Unified Dimension Model**
  - Unifies Relational, Cube …
  - Dimensions: role, fact, reference Data Mine, N2N
  - Measures and intelligent calculations.
- **MDX simplified, generalized**
  - Scripting, stored procedures
  - Debugging
  - XML representation
- **Performance**
  - Proactive caching – update cube when fact table changes
  - Partitioning and Write Back accelerated.
- **Enables Real-Time BI.**

SQL Server Analysis services covers both data analysis and data mining.

It introduces a Business Intelligence workbench that organizes both data access and workflows.

SQL 2005 introduces a Unified Dimension Model
that makes it much easier to think about and manipulate multi-dimensional data.

UDM makes a clear distinction between **measures** and **dimensions**.
It unifies the ROLAP, OLAP, and MOLAP worlds –
It lets you design a single schema that pulls data from many sources
and to ask queries that aggregate measures along any of the dimensions.

Combined with the new MDX query language,
it is much simpler and more powerful than the previous interface.

MDX now has a scripting language and also an interactive debugger built into the workbench.

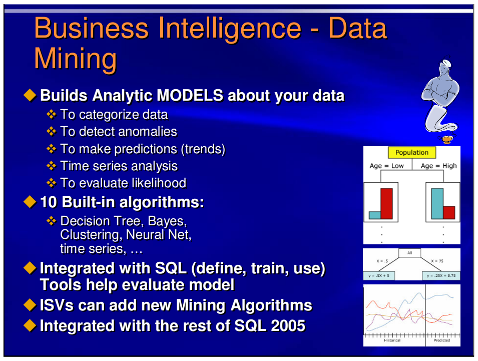
The UDM includes “intelligent types” that understand standard dimensions like Time,
and standard measures like money and the associated calculations and aggregations.

Of course, when dealing with these huge data volumes it’s essential to have good algorithms.
SQL 2005 introduces **“proactive caching** that lets you specify how “fresh” the cubes should be.
If you like, SQL will update the cube as the underlying fact data changes.

This and many other features make SQL2005’s
Cube implementation the most scalable in the industry.
They let you navigate terabyte-sized fact tables.

And all this has an XML flavor that makes it easy for tool builders to add new ways of navigating the data.

But the real excitement in the knowledge discovery features of SQL 2005



Business Intelligence - Data Mining

- Builds Analytic MODELS about your data
  - To categorize data
  - To detect anomalies
  - To make predictions (trends)
  - Time series analysis
  - To evaluate likelihood
- 10 Built-in algorithms:
  - Decision Tree, Bayes, Clustering, Neural Net, time series, …
- Integrated with SQL (define, train, use) Tools help evaluate model
- ISVs can add new Mining Algorithms
- Integrated with the rest of SQL 2005

The progression is that data leads to information
that leads to knowledge
that leads to wisdom.

Most database systems are still at the data stage.

SQL Server has gotten data to the information level
with things like cubes and reporting services.

Now we get to the exciting stuff --- the real future of databases –
the knowledge discovery features of SQL 2005

SQL Server 2000 was the first to integrate data mining with a SQL system.

Doing this required a fundamental change in the way we think about answers.

Data mining answers are not exact – rather they are estimates.

Each answer comes with a ***Support*** and **Confidence**.

Support says how much evidence there is for this prediction.
Confidence estimates how reliable the prediction is.

For example, if we have only 3 cases and 2 of them are cured,

The cure prediction is 66% confidence but not does not have much support.
If we have 1,000 cases and 600 of them are cured
then we have much more support for our prediction.

This approximate reasoning is needed for text search and for data mining.
It is a different mind set.

SQL 2005 can build analytic models about your data to summarize it,
to categorize it, to find trends and anomalies and to make predictions based on it.

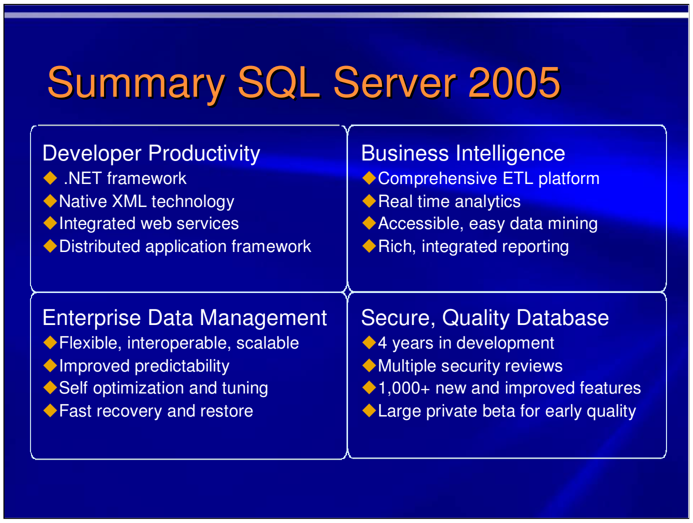
It has 10 built-in data mining algorithms,
wizards to help you use them,
and a very intuitive, define-train-predict interface.

It can mine numeric, text, and temporal data.

It has a framework that lets you or others add new mining algorithms.

In the past, data mining was an academic domain,
but these tools are accessible to business professionals.

Now you can see and understand your data in new ways.



Summary SQL Server 2005

Developer Productivity
- .NET framework
- Native XML technology
- Integrated web services
- Distributed application framework

Business Intelligence
- Comprehensive ETL platform
- Real time analytics
- Accessible, easy data mining
- Rich, integrated reporting

Enterprise Data Management
- Flexible, interoperable, scalable
- Improved predictability
- Self optimization and tuning
- Fast recovery and restore

Secure, Quality Database
- 4 years in development
- Multiple security reviews
- 1,000+ new and improved features
- Large private beta for early quality

To Summarize, SQL Server

delivers a unified storage foundation that organizes all your data,

including structured and unstructured data,

delivers secure, highly available operations and scalability,

At the lowest cost.

The innovations in,

development tools,

data analytics,

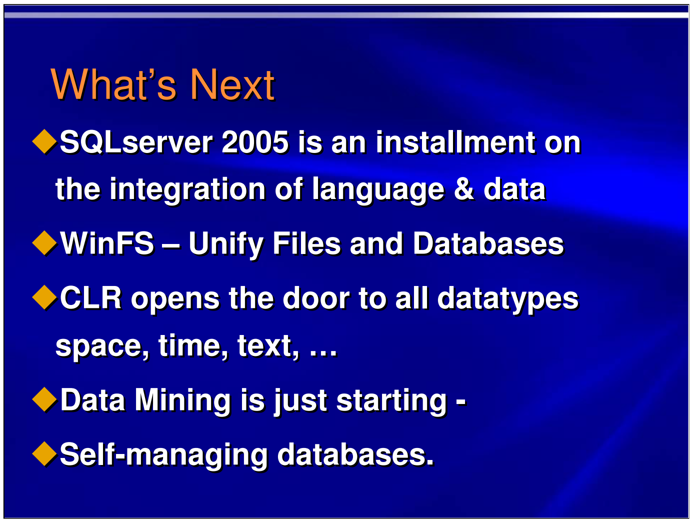
data transformation,

and reporting

Puts information in the hands of all employees

to make better decisions faster.

Arid, it is the first SODA system.



What's Next

- SQLserver 2005 is an installment on the integration of language & data
- WinFS – Unify Files and Databases
- CLR opens the door to all datatypes space, time, text, …
- Data Mining is just starting -
- Self-managing databases.

That's revolutionary!

And we are very proud of it,

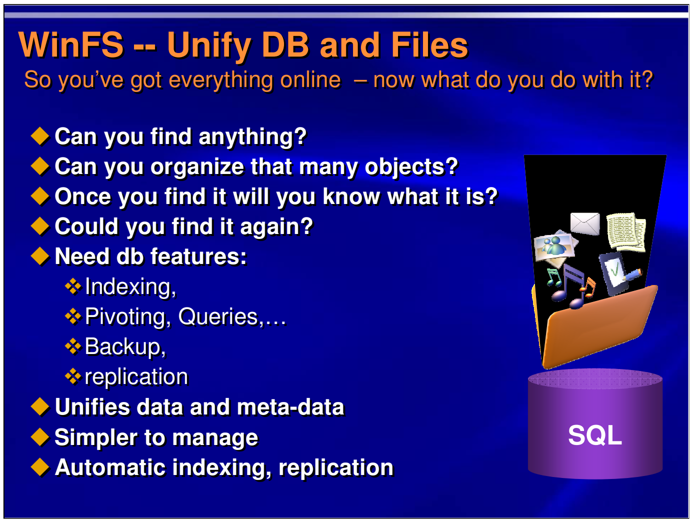
but… what next?

SQL Server 2005 is just a big step for us – and the basis for our future products.

CLR opens the door to integrating all datatypes.

The BI framework opens the door to new data mining algorithms.

And we continue to make the system more self managing.



WinFS -- Unify DB and Files

So you've got everything online – now what do you do with it?

- Can you find anything?
- Can you organize that many objects?
- Once you find it will you know what it is?
- Could you find it again?
- Need db features:
  - Indexing,
  - Pivoting, Queries,…
  - Backup,
  - replication
- Unifies data and meta-data
- Simpler to manage
- Automatic indexing, replication

WinFS tries to solve a BIG problem.

We have too much information

in too many formats

and in too many repositories.

How can you find anything?

What does it mean?

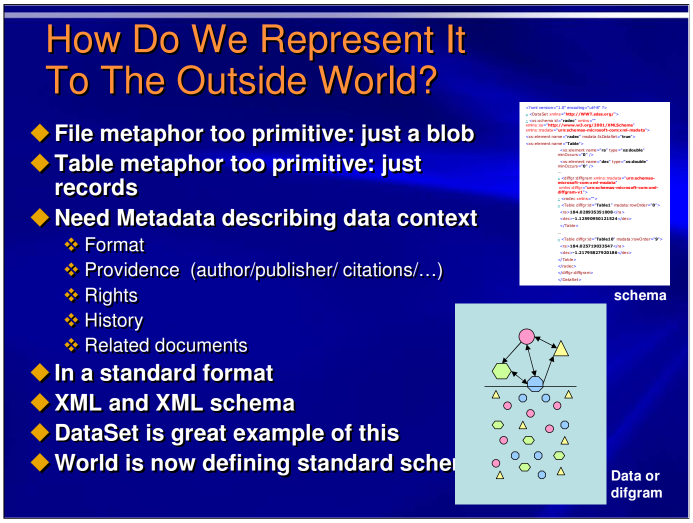
Organizations have this problem, but so do individuals.

It is easier to explain the story for individuals:
you have mail and photos and music and financial records
All spread across many different stores.

The next version of SQL Server hopes to
unify all these stores as part of WinFS.

WinFS unifies files and databases by keeping the metadata in the database and the large data streams in the file system.

The second beta of WinFS was released two months ago and
the next version of SQL Server will include the "real" WinFS.



How Do We Represent It To The Outside World?

- File metaphor too primitive: just a blob
- Table metaphor too primitive: just records
- Need Metadata describing data context
  - Format
  - Providence (author/publisher/ citations/…)
  - Rights
  - History
  - Related documents
- In a standard format
- XML and XML schema
- DataSet is great example of this
- World is now defining standard scher

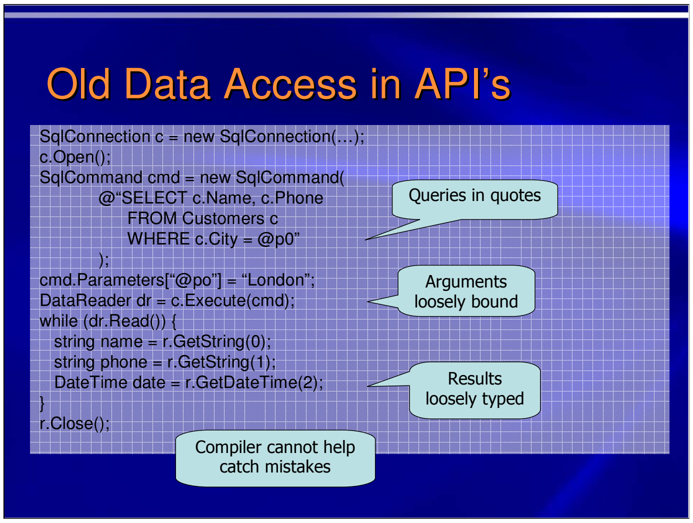
```
<?xml version="1.0" encoding="utf-8" ?>
- <DataSet xmlns="http://WWT.sdss.org/">
- <xs:schema id="radec" xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
<xs:element name="radec" msdata:IsDataSet="true">
<xs:element name="Table">
            <xs:element name="ra" type="xs:double"
           minOccurs="0" />
            <xs:element name="dec" type="xs:double"
           minOccurs="0" />
           …
           - <diffgr:diffgram xmlns:msdata="urn:schemas-
           microsoft-com:xml-msdata"
            xmlns:diffgr="urn:schemas-microsoft-com:xml-
           diffgram-v1">
           - <radec xmlns="">
           - <Table diffgr:id="Table1" msdata:rowOrder="0">
            <ra>184.028935351008</ra>
            <dec>-1.12590950121524</dec>
            </Table>
           …
           - <Table diffgr:id="Table10" msdata:rowOrder="9">
            <ra>184.025719033547</ra>
            <dec>-1.21795827920186</dec>
           </Table>
           </radec>
           </diffgr:diffgram>
           </DataSet>
```

In order to exchange information we need standard data formats.

Microsoft and others are working on standard XML definitions for concepts like document, calendar, event, person, and so on.

Others are standardizing other concepts using XML Schema so that we can exchange information.

Serializers for these classes will all be built into .NET, SQL and products from other vendors.



Old Data Access in API’s

```
SqlConnection c = new SqlConnection(…);
c.Open();
SqlCommand cmd = new SqlCommand(
        @“SELECT c.Name, c.Phone
            FROM Customers c
            WHERE c.City = @p0”
        );
cmd.Parameters[“@po”] = “London”;
DataReader dr = c.Execute(cmd);
while (dr.Read()) {
  string name = r.GetString(0);
  string phone = r.GetString(1);
  DateTime date = r.GetDateTime(2);
}
r.Close();
```

Queries in quotes

Arguments loosely bound

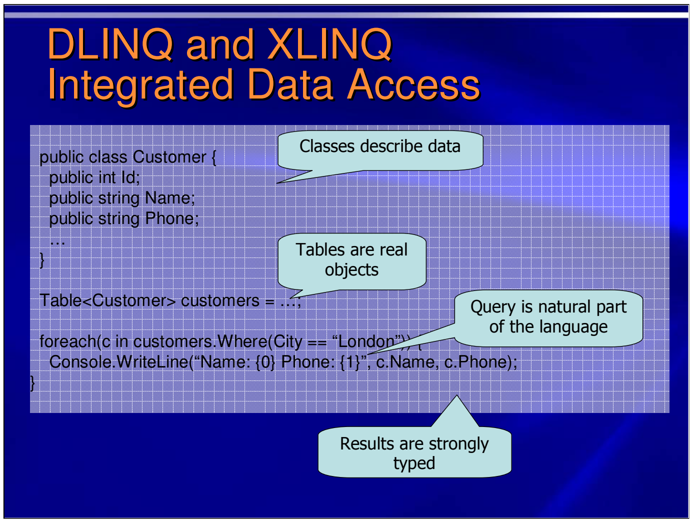
Results loosely typed

Compiler cannot help catch mistakes

It is still to difficult to write data access applications.

Look at this program that just wants to get London customers.

It’s embarrassing.



DLINQ and XLINQ
Integrated Data Access

```
public class Customer {
  public int Id;
  public string Name;
  public string Phone;
  …
}

Table<Customer> customers = …;

foreach(c in customers.Where(City == “London”)) {
  Console.WriteLine(“Name: {0} Phone: {1}”, c.Name, c.Phone);
}
```

Classes describe data
Tables are real objects
Query is natural part of the language
Results are strongly typed

Anders Hejlsberg, the architect of C#, and his colleagues

Have a breakthrough in programming data access.

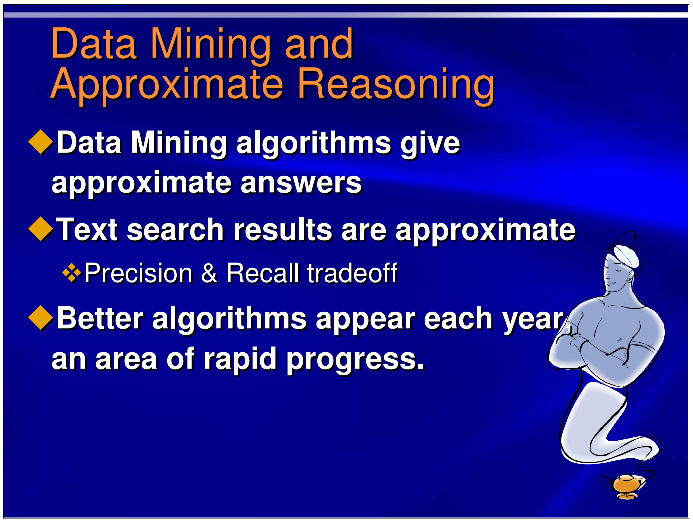
It is called LINQ (DLINQ for records and XLINQ for XML).

It vastly simplifies writing data access programs.

You can download it for C#   today.

Bindings for VB and other languages are coming.

So, there is LOTS of progress in
making it easier to write Data access programs
and to write more reliable programs.



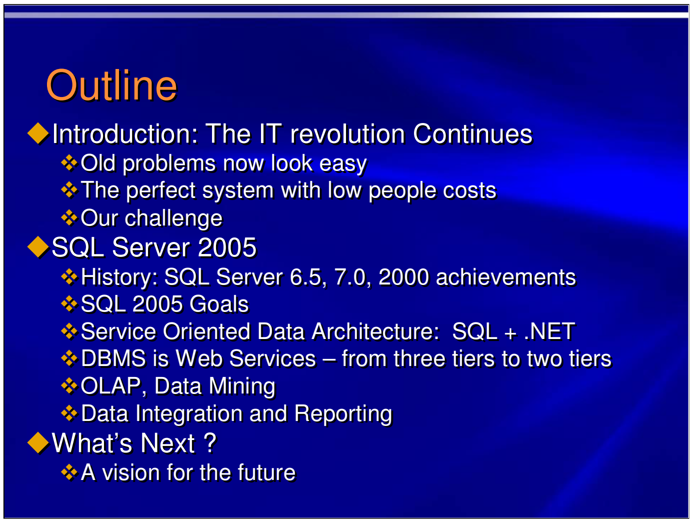
Data Mining and Approximate Reasoning

- Data Mining algorithms give approximate answers
- Text search results are approximate
  - Precision & Recall tradeoff
- Better algorithms appear each year, an area of rapid progress.

And, if we talk about the future, I have to come back to the main idea.

Data mining and knowledge discovery is just starting.

Huge progress will be made in this area with new ideas and new algorithms.



Outline

- Introduction: The IT revolution Continues
  - Old problems now look easy
  - The perfect system with low people costs
  - Our challenge
- SQL Server 2005
  - History: SQL Server 6.5, 7.0, 2000 achievements
  - SQL 2005 Goals
  - Service Oriented Data Architecture: SQL + .NET
  - DBMS is Web Services – from three tiers to two tiers
  - OLAP, Data Mining
  - Data Integration and Reporting
- What's Next ?
  - A vision for the future

It's and exciting time.

Old problems now look easy.

SQL 2005 is a huge advance,

Bringing many features together
and integrating with Visual Studio, BizTalk,
and Office, and Windows.

It is the SODA platform.

Like you, I have been waiting for it for a long time.

But, now the wait is over.

Now we can use it for our applications.

Thank you very much for your attention,

There are in-depth presentations on SQL this afternoon.

I hope you find the rest of the day informative and helpful.