10

Chapter 10 – Technology trends and databases

An IBM survey conducted in 2010 revealed different technology trends by 2015. The survey garnered responses from more than 2,000 IT professionals worldwide with expertise in areas such as software testing, system and network administration, software architecture, and enterprise and web application development. There were two main findings from the survey:

- Cloud computing will overtake on-premise computing as the primary way organizations acquire IT resources.
- Mobile application development for devices such as iPhone and Android, and even tablet PCs like iPad and PlayBook, will surpass application development on other platforms.

This chapter explores Cloud computing, mobile application development, and other technology trends, and explains the role that databases play in these technologies. Moreover, the chapter presents you with a real life case scenario where these technologies are used. After completing this chapter, you should have a good understanding of these technologies to take advantage of them for future solutions you develop.

10.1 What is Cloud computing?

Cloud computing is not a new technology, but a new model to deliver IT resources. It gives the illusion that people can have access to an infinite amount of computing resources available on demand. With Cloud computing, you can rent computing power with no commitment. There is no need to buy a server, just pay for what you use. This new model is often compared with how people use and pay for utilities. For example, you only pay for how much water or electricity you consume for a given amount of time.

Cloud computing has drastically altered the way computing resources are obtained, allowing almost everybody, from one-man companies, large enterprises to governments work on projects that could not have been possible before.

Table 10.1 compares the traditional IT model in an enterprise versus the Cloud computing model.

Traditional IT model	Cloud computing model
Capital budget required	Part of operating expense
Large upfront investment	Start at 2 cents / hour
Plan for peak capacity	Scale on demand
120 days for a project to start	Less than 2 hours to have a working system

Table 10.1 - Comparing the traditional IT model to the Cloud computing model.

While in the traditional IT model you need to request budget to acquire hardware, and invest a large amount of money upfront; with the Cloud computing model, your expenses are considered operating expenses to run your business; you pay on demand a small amount per hour for the same resources.

In the traditional IT model, requesting budget, procuring the hardware and software, installing it on a lab or data center, and configuring the software can take a long time. On average we could say a project could take 120 days or more to get started. With Cloud computing, you can have a working and configured system in less than 2 hours!

In the traditional IT model companies need to plan for peak capacity. For example if your company's future workload requires 3 servers for 25 days of the month, but needs 2 more servers to handle the workload of the last 5 days of the month then the company needs to purchase 5 servers, not 3. In the Cloud computing model, the same company could just invest on the 3 servers, and rent 2 more servers for the last 5 days of the month.

10.1.1 Characteristics of the Cloud

Cloud Computing is based on three simple characteristics:

- Standardization.

  Standardization provides the ability to build a large set of homogeneous IT resources mostly from inexpensive components. Standardization is the opposite of customization.

- Virtualization

  Virtualization provides a method for partitioning the large pool of IT resources and allocating them on demand. After use, the resources can be returned to the pool for others to reuse.

- Automation

  Automation allows users on the Cloud to have control on the resources they provision without having to wait for administrator to handle their requests. This is important in a large Cloud environment.

If you think about it, you probably have used some sort of Cloud service in the past. Facebook, Yahoo, and Gmail, for example, deliver services that are standardized, virtual and automated. You can create an account and start using their services right away.

10.1.2 Cloud computing service models

There are three Cloud computing service models:

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

Infrastructure as a Service providers take care of your infrastructure (Data center, hardware, operating system) so you don't need to worry about these.

Platform as a Services providers take care of your application platform or middleware. For example, in the case of IBM middleware products, a PaaS would take care of providing for a DB2 server, a WebSphere application server, and so on.

Software as a Service providers take care of the application that you need to run. You can think of them as "application stores" where you go and rent applications you need by the hour. A typical example of a SaaS is Salesforce.com.

10.1.3 Cloud providers

There are several Cloud computing providers in the market today such as IBM with its "Smart Business Development and Test on the IBM Cloud" offering; Amazon with its Amazon Web Services (AWS) offerings, Rackspace, and so on.

10.1.3.1 IBM Smart Business Development and Test on the IBM Cloud

The IBM Smart Business Development and Test on the IBM Cloud, or ***IBM developer cloud,*** provides Cloud services specifically tailored for development and test. This Cloud has data centers in the US and Europe, and allows you to provision Intel-based 32-bit and 64-bit ***instances*** using RedHat or SUSE Linux, or Microsoft Windows. You can think of an instance as a virtual server.

The IBM Developer Cloud can be accessed at https://www-147.ibm.com/cloud/enterprise

Figure 10.1 shows the IBM Developer Cloud dashboard. It shows three different instances running, the operating system used and the IP address.

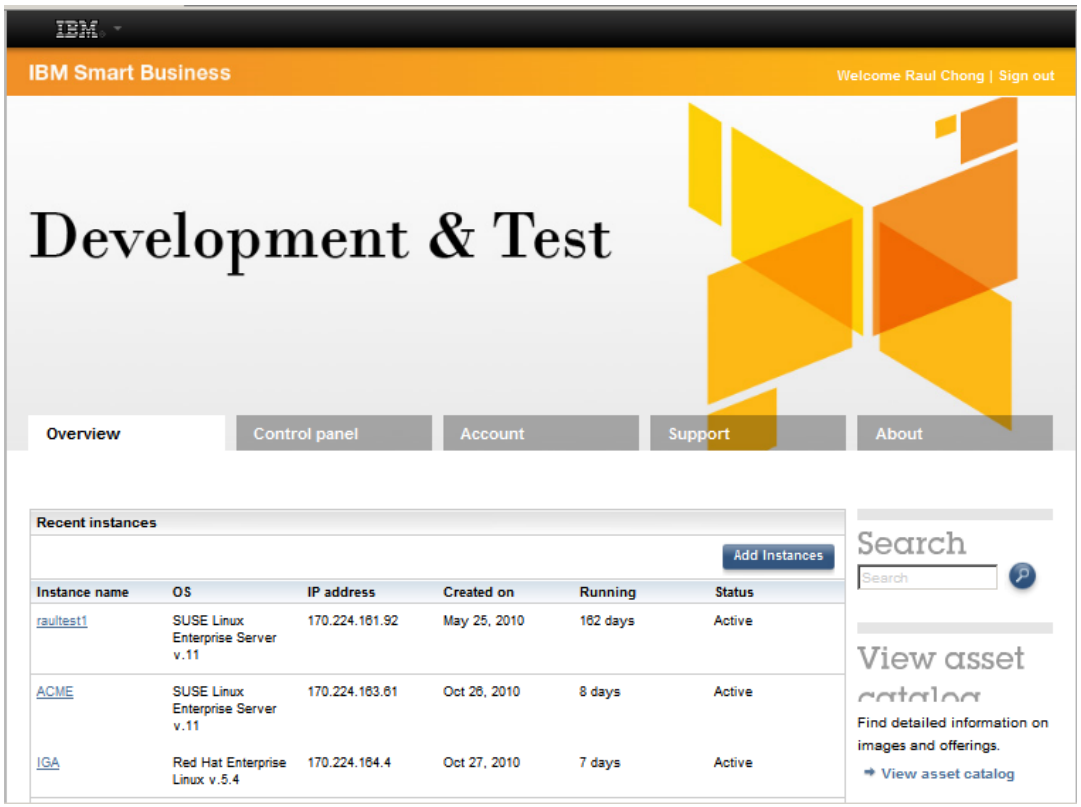

Figure 10.1 - The IBM Developer Cloud dashboard

Getting a system up and running can take just a few minutes. Simply click on the "Add instances" button to start. Next, choose from one of the many instances available as shown in *Figure 10.2*.

Figure 10.2 - Choosing the instance to use on the IBM Developer Cloud

For example, if you would like to work with IBM DB2 Express-C 9.7.1 - PAYG (Pay as you go) on SUSE Linux, simply select that option. Once selected, click next, and you will be taken to a screen as shown in *Figure 10.3*

In this panel you can choose the type of instance you want (32-bit, 64-bit) and how many CPU cores to use classified by name (copper, bronze, silver, gold). For example, bronze on 32-bit typically gives you 1 CPU running at 1.25GHz, 2GB of memory, and 175GB of local storage.

Figure 10.3 also shows that you need to input a string, which will be used to generate a key that you need to download to your own computer so that later you can SSH to the instance created using ssh software like the free *putty*.

Figure 10.3 - Configuring the instance

After you click *Next*, your instance will be provisioned, which means that virtual CPUs and memory are allocated, the operating system is installed, and the software in the image (such as DB2 Express-C database server in this example) are also installed. This process can take a few minutes. Later on you can also add storage to attach to your instances.

10.1.3.2 Amazon Web Services

Amazon Web Services, or ***AWS,*** is the leading provider of public cloud infrastructure. AWS has data centers in four regions: US-East, US-West, Europe and Asia Pacific. Each region has multiple availability zones for improved business continuity.

Similarly to the IBM developer cloud, with AWS you can select virtual servers, called **Elastic Cloud compute (EC2) instances.** These instances are Intel-based (32 and 64-bit), and can run Windows or Linux (numerous distributions) operating systems. Pick from the different pre-defined types of instances based on your need for CPU cores, memory and local storage. *Figure 10.4* summarizes the different AWS EC2 instance types. Each type has a different price per hour as documented at aws.amazon.com.

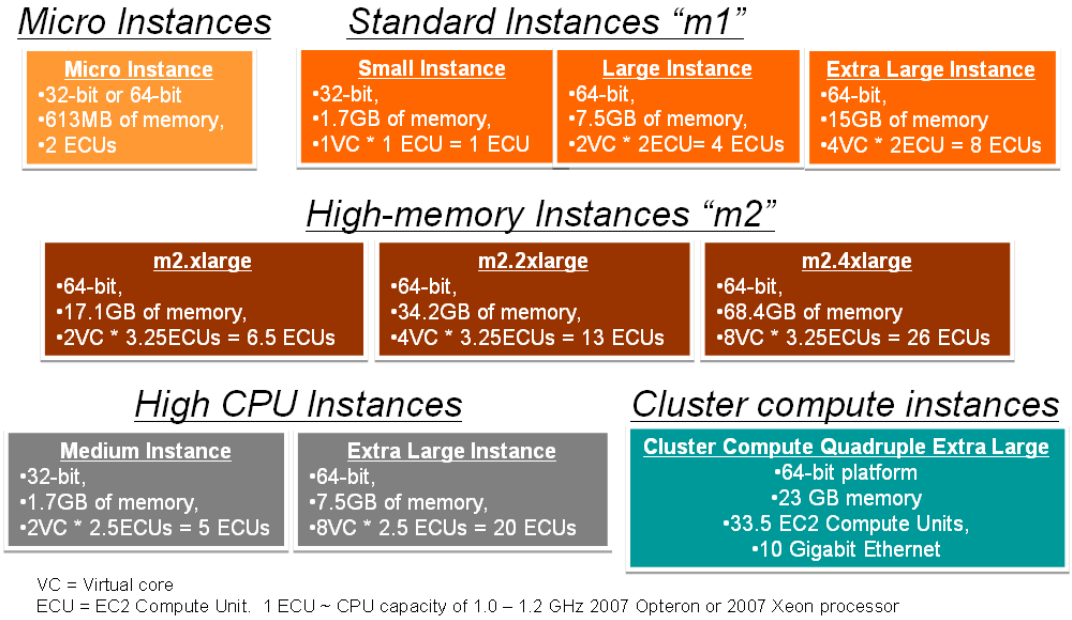

Figure 10.4 - AWS EC2 instance types

You can also add storage to your instance. AWS has three choices:

- Instance storage:

  Instance storage is included with your instance at no extra cost; however, data is not persistent which means that it would disappear if the instance crashes, or if you terminate it.

- Simple Storage Service (S3).

  S3 behaves like a file-based storage organized into buckets. You interact with it using http put and get requests.

- Elastic Block Storage (EBS).

  EBS volumes can be treated as regular disks on a computer. It allows for persistent storage, and is ideal for databases.

10.1.4 Handling security on the Cloud

Security ranks high when discussing the reasons why a company may not want to work on the public Cloud. The idea of having confidential data held by a third party Cloud provider is often seen as a security and privacy risk.

While these concerns may be valid, Cloud computing has evolved and keeps evolving rapidly. Private clouds provide a way to reassure customers that their data is held safely on-premises. Hardware and software such as IBM Cloudburst™ and IBM WebSphere Cloudburst Appliance work hand-in-hand to let companies develop their own cloud.

Companies such as Amazon and IBM offer virtual private cloud (VPC) services where servers are still located in the cloud provider's data centers yet they are not accessible to the internet; security can be completely managed by the company's own security infrastructure and processes.

Companies can also work with ***hybrid clouds*** where they can keep critical data in their private cloud, while data used for development or testing can be stored on the public cloud.

10.1.5 Databases and the Cloud

Cloud Computing is a new delivery method for IT resources including databases. IBM DB2 data server is Cloud-ready in terms of licensing and features.

Different DB2 editions for production, development and test are available on AWS and the IBM developer cloud. DB2 images are also available for private clouds using VMware or WebSphere Cloudburst appliance. *Figure 10.5* summarizes the DB2 images available on the private, hybrid and public clouds.

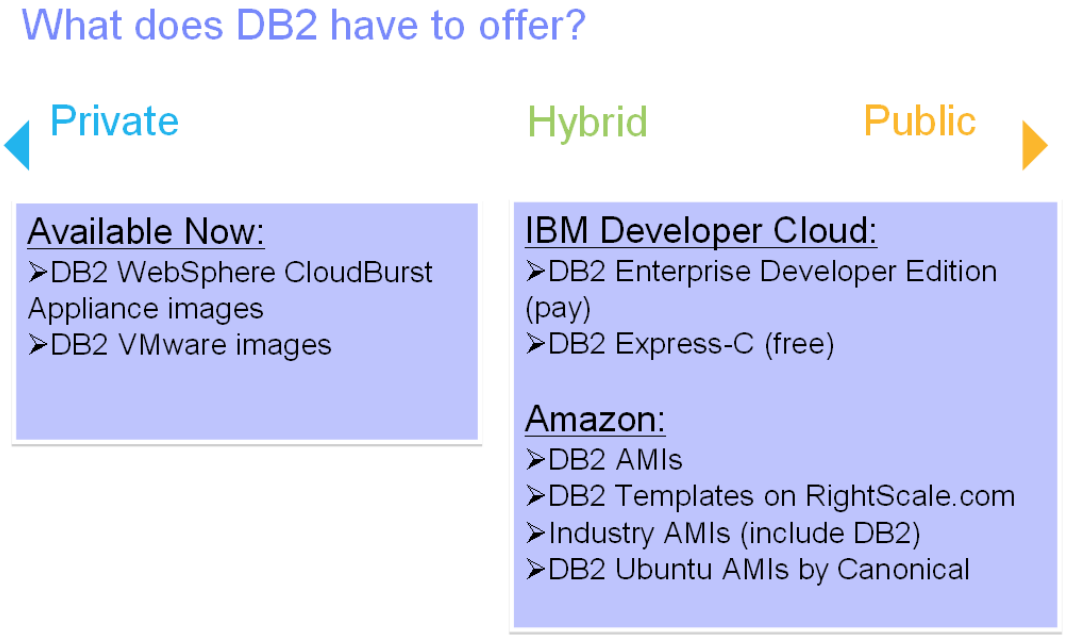

Figure 10.5 - DB2 images available on the private, hybrid and public clouds

In terms of licensing, you can use these methods:

- Bring your own license (BYOL) allows you to use your existing DB2 licenses on the cloud
- Pay as you go (PAYG) allows you to pay for what you use

You can always use DB2 Express-C on the Cloud at no charge, though you may have to pay the Cloud provider for using its infrastructure.

In terms of DB2 features that can be particularly advantageous in a Cloud environment we have:

- Database Partitioning Feature (DPF)
- High Availability Disaster Recovery (HADR)
- Compression

DPF is based on the shared-nothing architecture that fits well into the Cloud provisioning model. DPF is ideal for Data Warehousing environments where there is a large amount of data that you need to query to obtain business intelligence reports. With DPF, queries are automatically parallelized across a cluster of servers; this is done transparently to your application. You can add servers on demand on the Cloud which would bring their own CPU, memory and storage. DB2 would then automatically rebalance the data, and the overall performance would improve almost linearly.

HADR is a robust feature of DB2 with more than 10 years in the market. HADR works with two servers, a primary and a stand-by. When both servers are placed on the same location the HADR feature provides for *high availability*. When one server is placed in one location, and the other is in another (typically a different state or country), then HADR provides for *disaster recovery*. Disaster Recovery (DR) is one of the most needed and most expensive IT areas. It is expensive because companies need to pay for space in another location, as well as for the IT resources (servers, storage, networking) required. Cloud Computing addresses all of these needs very well. In the Cloud you "rent space" in distributed data centre(s) but do not pay for the space, electricity, cooling, security, and so on. You also can get all the IT resources without capital budget. Accessing your "DR site" on the Cloud can be achieved from anywhere securely using a web browser and ssh. With HADR you can place your primary server on-premise, and your stand-by on the Cloud. For added security, you can place the stand-by on a virtual private cloud. HADR allows your system to be up and running in less than 15 seconds should your primary server crash.

If you really want to save money to the last penny, you can use DB2's compression feature on the Cloud to save on space. DB2's compression, depending on your workload, can also increase overall performance.

There are many other features of DB2 that are not discussed in this book but could be applicable to this topic.

10.2 Mobile application development

Mobile applications are software programs that run on a mobile device. Typical enterprise mobile applications utilize mobile computing technology to retrieve information from a main computer server to the mobile device at anytime and anywhere. Moreover, mobile applications using the Cloud as the backend are becoming more and more popular.

The design and implementation of mobile applications is not as straightforward as desktop PC application development. It is very important for mobile application developers to consider the context in which the application will be used. From a business point of view, it

is difficult for project managers to estimate the risk associated with a mobile application project due to the difficulties in comparing the various capabilities of existing mobile platforms.

Nowadays, the early mobile platforms like Symbian, Microsoft Windows Mobile, Linux and BlackBerry OS have been joined by Apple's OS X iPhone, Android and recently Palm's Web OS, adding a layer of complexity for developers. *Figure 10.6* illustrates the typical architecture of mobile application development.

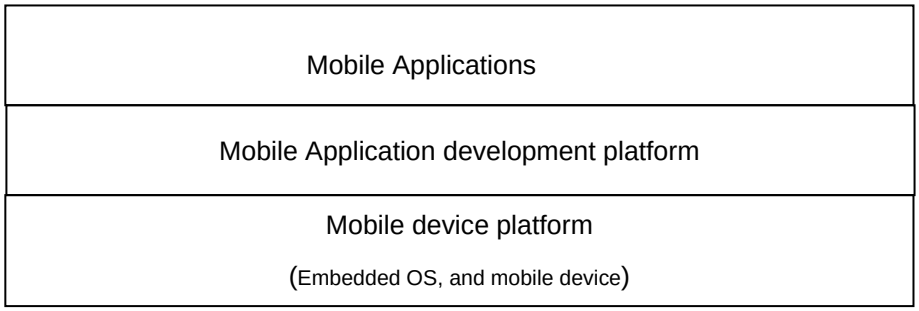

Figure 10.6 - Mobile Application Development: The big picture

In *Figure 10.6*, we show how mobile applications sit on top of two layers:

Mobile device platform refers to the mobile hardware device and its operating system and/or integrated software. For example, Microsoft Windows Mobile 5.0 OS is installed on a Dell AXIM X51v device.

Mobile application development platform refers to the combination of programming languages, general development APIs, and the runtime environment for that language on the mobile device. The mobile application development platform runs on top of the mobile device platform and provides an extra layer of software as a protection barrier to limit the damage for erroneous or malicious software. For example, the typical development software for mobile applications for Windows Mobile 6.0 is Microsoft Visual Studio 2005/2008 plus Windows Mobile 6.0 SDK.

Since there are various types of mobile device platforms and mobile application development platforms to choose from, ask yourself the following question before assembling a list of application platforms: Are you developing an application for a specific device or an application development platform?

10.2.1 Developing for a specific device

When writing an application to a specific device (or set of devices) or device OS, the vendor of that device platform often provides an SDK, emulator, and other tools that can help developers write applications specific to that platform.

Table 10.2 lists some of these mobile device platforms and what they offer in the way of SDKs, tools, and so on.

Operating System (OS)	Language/OS options	SDKs/Tools	Developer Web Site
Symbian OS	C++, Java, Ruby, Python, Perl, OPL, Flash Lite, .NET	Carbide C++, IDE(Nokia) – C++, and Java SDKs, per device	developer.symbian.com
Windows Mobile	Java, C++	Visual Studio, Platform Builder, Embedded Visual C++ (eVC), Free Pascal, and Lazarus	Microsoft Windows Embedded Developer Center
iPhone OS	Objective-C	Xcode	www.apple.com/iphone
Android	Java	Eclipse, Android SDK, Android Development Tool Plugin, JDK 5	www.android.com
Palm OS	C, C++, Java	Palm OS SDK, Java development with IBM Websphere EveryPlace Micro Environment, Palm Windows Mobile SDK for Palm products on Windows Mobile platform	www.palm.com/us/developer

Table 10.2 - Device platforms and tools

10.2.2 Developing for an application platform

Everyone has beliefs in a particular programming language and development environment. Why do you prefer that development language and application development environment?

Mobile application development raises this question time and again, and you may have to re-evaluate your answer in light of your mobile application .Java, .NET (C#, VB), C, C++, and other languages are all options available in mobile application development. In some ways, the same criteria that are used to evaluate these languages on the desktop or server can also be applied to mobile applications. However, mobile application development does

bring considerations, which distort the old desktop/server application development environment discussions.

For example, Java's strength has always been portability. Java is a widely available platform on many mobile devices. However, the exact same Java is not on all those devices. In order for Java to fit on so many different styles and shapes of platforms, the Java architecture has been broken into a set of configurations and profiles. The configurations and profiles differ according to device and device capabilities. Thus, writing to this platform may not get you to the customers you want to capture.

So how will you look at the software platforms for mobile devices? The next two sections explore several more prevalent mobile device platform and mobile application platform choices and looks at ways to compare and contrast these platforms.

10.2.3 Mobile device platform

Some of the most widely used mobile device platforms are described below.

10.2.3.1 Windows Mobile

Windows Mobile, derived from Windows CE (originally released in 1996), provides a subset of the Microsoft Windows APIs on a compact real-time operating system designed for mobile computing devices and wireless terminals. After several false starts on stand-alone PDAs, Windows Mobile has enjoyed reasonable success on a number of PDAs and handsets, perhaps most notably the Motorola Q and some Treo devices.

Microsoft provides the ActiveSync technology for synchronization for both personal and enterprise users. Moreover, the current version of Microsoft Windows Mobile is a versatile platform for developers, permitting application development in C or C++ as well as supporting managed code with the .NET Compact Framework.

10.2.3.2 Symbian OS

Symbian OS is a proprietary operating system designed for mobile devices, with associated libraries, user interface, frameworks and reference implementations of common tools, developed by Symbian Ltd. Symbian began its life as EPOC Release 5, the last software release of EPOC by Psion with a joint venture company of the same name. Symbian is a very powerful platform programmable in C++ using Symbian’s frameworks for applications and services. The Symbian platform itself runs Java ME, so Symbian phones can run both native Symbian applications as well as Java ME applications.

10.2.3.3 iPhone

The iPhone is an internet-connected, multimedia smartphone designed and marketed by Apple Inc. It is a closed platform, permitting mobile applications only through its robust WebKit-based Web Browser, Safari. Since its minimal hardware interface lacks a physical keyboard, the multi-touch screen renders a virtual keyboard when necessary.

The iPhone and iPod Touch SDK uses Objective C, based on the C programming language. The available IDE for iPhone application development is Xcode, which is a suite of tools for developing software on Mac OS X developed by Apple.

10.2.3.4 Android

Android is an open and free software stack for mobile devices that includes operating system, middleware, and key applications. The development language for Android is Java.

Android application programming is exclusively done in Java. You need the Android specific Java SDK which includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator (based on QEMU), documentation, sample code, and tutorials. Currently supported development platforms include x86-architecture computers running Linux (any modern desktop Linux Distribution), Mac OS X 10.4.8 or later, Windows XP or Vista. Requirements also include Java Development Kit, Apache Ant, and Python 2.2 or later. The officially supported integrated development environment (IDE) is Eclipse (3.2 or later) using the Android Development Tools (ADT) Plug-in, though developers may use any text editor to edit Java and XML files then use command line tools to create, build and debug Android applications.

10.2.4 Mobile application development platform

In this section, we will introduce the application development platforms to implement the mobile application on top of the mobile device.

10.2.4.1 Java Micro Edition (Java ME)

Java ME, formerly called ***Java 2 Micro Edition*** (***J2ME***), is a platform spanning lightweight mobile devices and other nontraditional computing devices including set-top boxes and media devices. Once promising “write once, run anywhere” to software developers, Java ME skills remain perhaps the most transferable from device to device, because lower-level languages such as C or C++ on other platforms require intimate knowledge of platform-specific APIs such as those found in Windows Mobile or Symbian OS.

There are actually different implementations of Java for different devices for the immense variety in devices supported by Java ME.

10.2.4.2 .NET Compact Framework

This platform is a version of the .NET Framework for Windows CE platforms. It is a subset of the standard .NET framework, but also includes some additional classes that address specific mobile device needs.

Currently there are .NET Compact Framework V1.0, V2.0 and V3.5 available. You might be wondering which version of the .NET Compact Framework should you choose as target? "The latest version" might be the obvious answer, but, as with many things concerning mobile devices, it is not quite that simple! As the mobile application developer, you must choose a version of the .NET Compact Framework on which to build your application. If you choose version 1.0, you can be reasonably confident that your application will run on all devices because versions 2.0 and later of the .NET Compact Framework runtime run applications that were built to run on an earlier version. However, if you write code that uses features only available in .NET Compact Framework 2.0, that version of the .NET Compact Framework runtime must be installed on your target device for your application to operate.

10.2.4.3 Native C++

C++ applications can run natively on the device, making the applications less portable, but this usually means you have more speed and more control over the device.

Generally, native C++ development needs IDE tools like Microsoft Visual C++ 6/.NET, Eclipse IDE together with specific C++ SDK, or Borland C++ Builder. Especially, for Symbian OS, you could also choose Metrowerks CodeWarrior Studio which also requires corresponding Symbian C++ SDK from Nokia.

10.2.4.4 Binary Runtime Environment for Wireless

The Binary Runtime Environment for Wireless (BREW) is a programming platform developed by Qualcomm for CDMA-based phones. BREW provides SDK and emulator for developing and testing applications written in C or C++.

To develop with BREW, you need development tools like BREW SDK, ARM RealView compilation tools, Visual C++, etc.

10.2.5 The next wave of mobile applications

A wide spectrum of next-generation mobile applications and services have begun to surface anywhere the coverage rate of cell phones PDAs is high. New mobile applications are being driven by mobile service providers seeking new value-added applications and services to take advantage of 3G systems and other wireless data infrastructures. Traditional mobile applications, such as voice, and simple data services, such as Internet surfing, are not enough today; consumers are looking forward to better leverage the mobility supplied by low-power, wireless-capable mobile devices to enjoy content rich entertainment, ubiquitous information access, and agile business operations.

A great many markets are yet to be saturated in the mobile world, such as telematics systems, m-commerce and m-enterprise services, mobile multimedia streaming service, mobile messaging, location based mobile computing, and so forth.

Surely, we will see many exciting technological breakthroughs and innovations of mobile computing in the next several years.

10.2.6 DB2 Everyplace

The DB2 edition for mobile devices is DB2 Everyplace. DB2 Everyplace features a small-footprint relational database and high-performance data synchronization solution that enables enterprise applications and data to be extended securely to mobile devices.

Note:

To learn more about mobile application development, refer to the free eBook *Getting started with mobile application development*, which is part of the DB2 on Campus book series.

10.3 Business intelligence and appliances

Databases are commonly used for online transaction processing such as bank transactions, but they can also be used to derive business intelligence to see trends and patterns that can help businesses make better decisions. Companies today collect huge amounts of information daily. This information is stored in data warehouses that are used to generate reports using software like IBM Cognos®.

Companies struggling to set up their data warehouses are now attracted to the idea of purchasing data warehouse appliances. An appliance consists of hardware and software that has been tightly integrated and tested to perform a given function or functions. ***IBM Smart Analytics System*** is one such appliance for Data Warehousing and Business Intelligence.

Note:

To learn more about data warehousing and business intelligence, refer to the free eBook *Getting started with data warehousing*, which is part of the DB2 on Campus book series.

10.4 db2university.com: Implementing an application on the Cloud (case study)

Advances in technology enable advances in other fields. Thanks to the internet, many universities are offering online degrees. Credits towards a degree are given to students who register and participate in online courses. These courses often consist of recorded webinars, live sessions over the Web, online forums, and more. Pundits argue that in the near future everyone with access to the internet will be able to get excellent education without having to physically attend a university.

Online education gives you the ability to learn at your own pace, in the comfort of your own home or wherever you are. Moreover, thanks to mobile devices you can participate "on the go".

In this section we introduce **db2university.com**, an educational Web site that has been implemented using many of the technologies described in this chapter. *Figure 10.7* illustrates how db2university.com looks like at the time of writing.

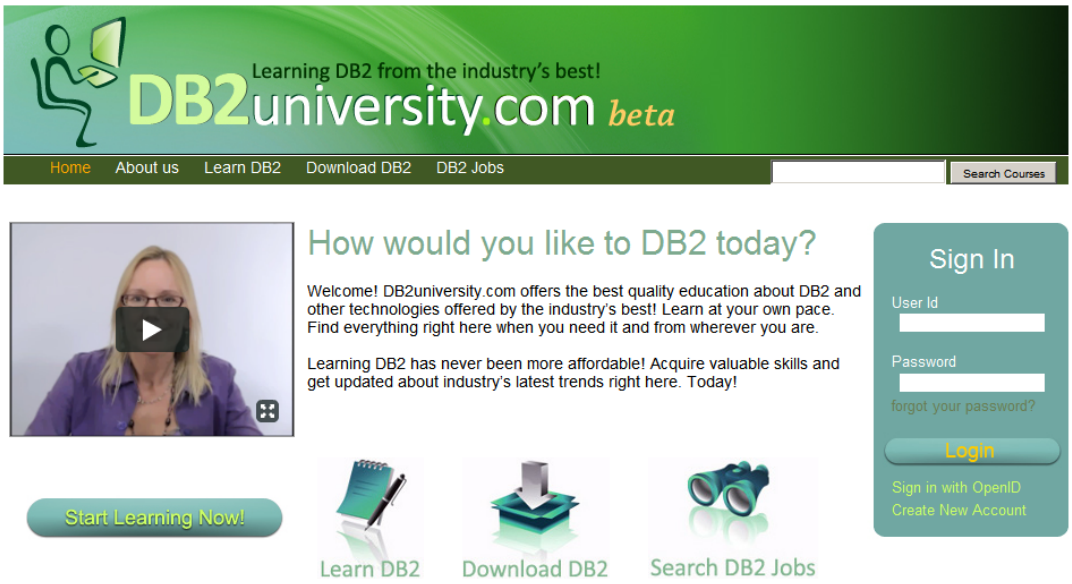

Figure 10.7 - db2university.com home page

10.4.1 Moodle open source course management system

A vast range of materials, from videos to podcasts and eBooks are available right now on the internet for self-study; however, they are often not arranged in a way that is easily consumable. Some of this information may also be obsolete. ***Course management systems*** allow instructors to create and arrange existing materials for easy consumption. At **db2university.com** we use the ***Moodle*** open source course management system (CMS).

Moodle, written in PHP, is one of the best, if not the best, open source CMS in the market. Martin Dougiamas at Curtin University of Technology in Perth, Australia developed the first version in 2001. Today, there are over 52,000 registered Moodle sites and over 950,000 registered users from more than 175 countries around the world on moodle.org.

Our journey with Moodle started late in 2009. Students at California State University, Long Beach (CSULB) were assigned, as part of one of their courses, the task to enable Moodle 1.9.6 to work with DB2 Express-C. Their contribution to the community (provided as a patch) can be found at http://moodle.org/mod/data/view.php?d=13&rid=3100&filter=1.

Early in 2010 we started the enablement of Moodle 2.0 to use DB2 Express-C. Moodle 2.0 is a major revision of the Moodle software, and it has fundamental changes from the previous version in the way it interfaces with databases. At the time of writing, Moodle 2.0 Release Candidate 1 has been made available to the public. This is the version we have enabled to use DB2 Express-C, and the one we are running at **db2university.com**. When the official version of Moodle 2.0 is available, we will be updating db2university.com and committing our contribution to the Moodle community.

Figure 10.8 shows different courses available at **db2university.com**. You can reach this page by clicking on the *Learn* tab in the home page. The Learn tab shows our implementation of Moodle. We created a Moodle theme specific for db2university.

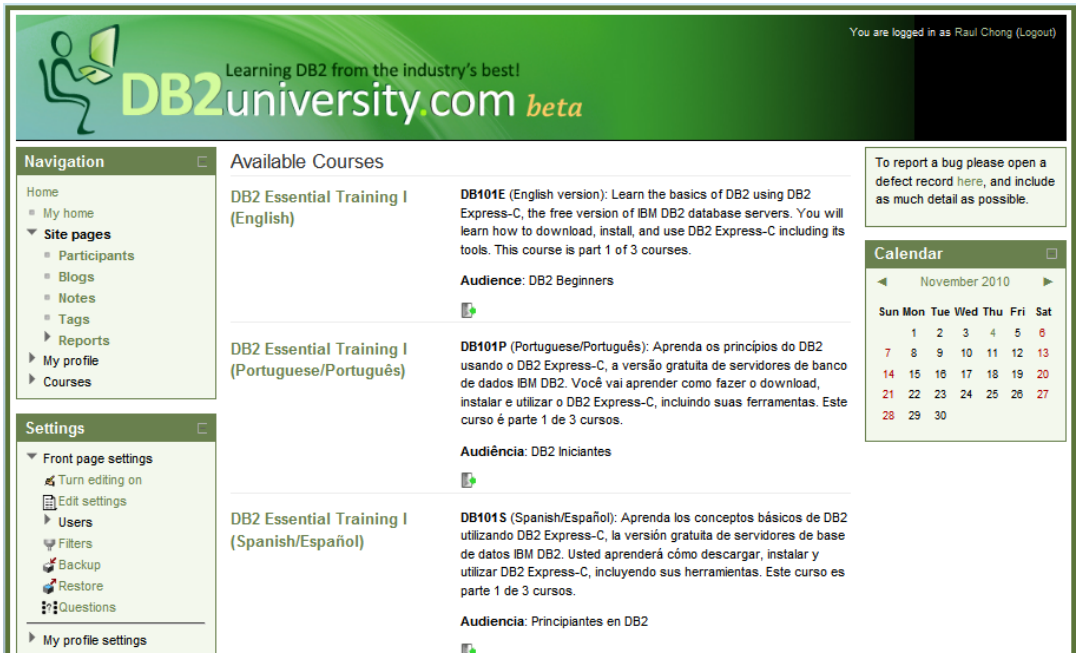

Figure 10.8 - Learn tab takes you to the Moodle implementation in db2university

Courses at **db2university.com** are not just about DB2. They can be about other topics like PHP or Ruby on Rails for example, but any lesson in those courses that requires interaction with a database will be using DB2. Instructors for the courses are members of the DB2 community including university professors, professionals, IBM employees, and students.

There are two main categories for courses at the site: *Free* courses, and *Paid* courses. For *Paid* courses we will use the Paypal plug-in included with Moodle.

Figure 10.9 and *10.10* provide more details about the course "DB2 Essential Training I". We picked this course as an example of what you could do with our implementation of Moodle. Note the course has many videos, transcripts (PDF files), and links to an eBook to use as reference. Not shown in the figure are links to a course forum, Web pages, assignments, and exams.

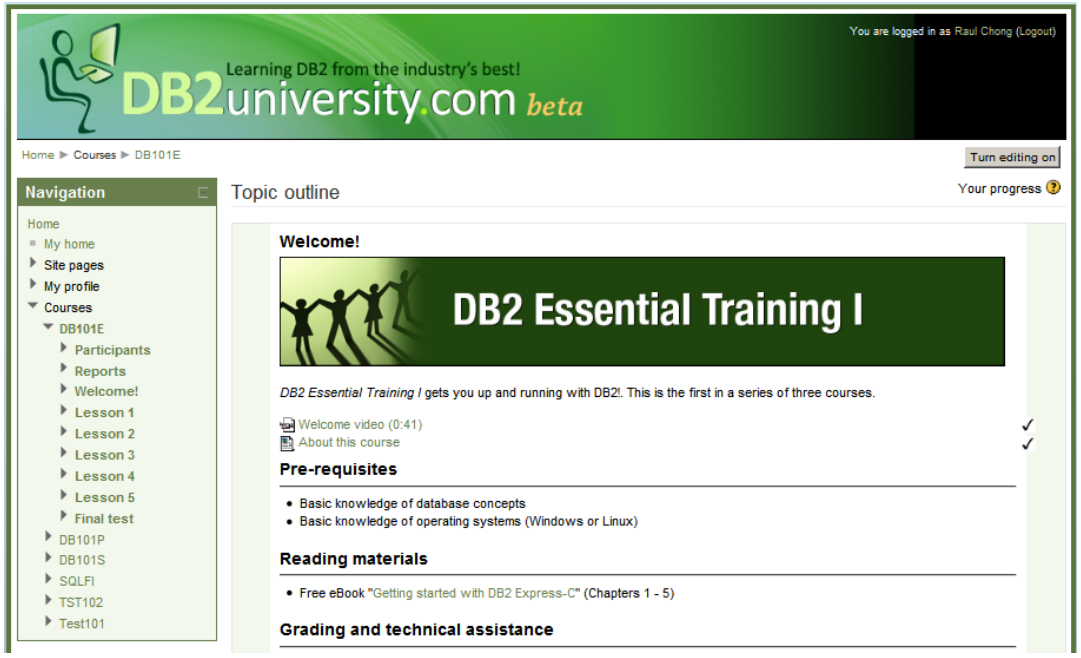

Figure 10.9 - Contents of the DB2 Essential Training I course

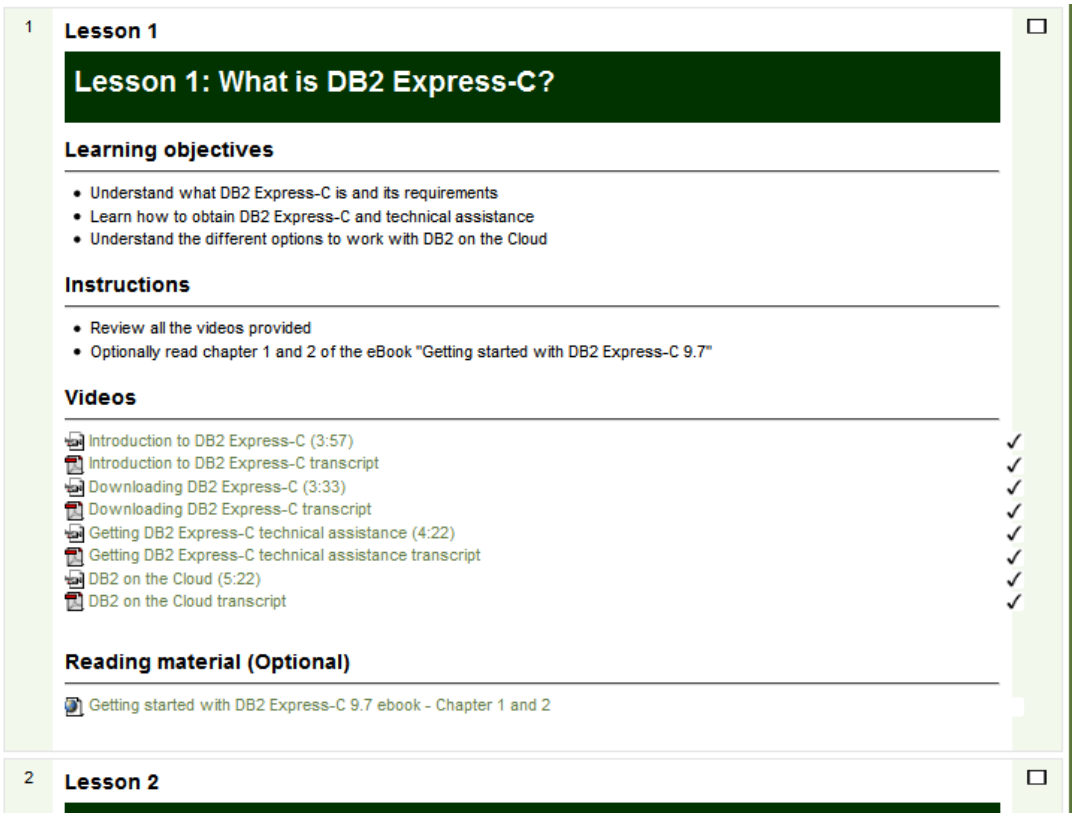

Figure 10.10 - Contents of the DB2 Essential Training I course (continued)

A tutorial is available at IBM developerWorks® to teach you how to create a course in db2university.com.

Note:

To learn more about what it takes to develop open source software, take a look at the free eBook *Getting started with open source development*, which is part of the DB2 on Campus book series.

10.4.2 Enabling openID sign-in

Research shows that people landing on a Web page with a complicated or long registration process tend to walk away from the site quickly. To ensure users stay and register to **db2university.com** we enabled openID sign in, in addition to allowing users to register on the site. This means that people who already have a Facebook, Google, Yahoo, AOL, ChannelDB2, and other accounts need not input registration information. db2university would instead ask those providers -- with your consent -- to share the information. *Figure 10.11* illustrates the process.

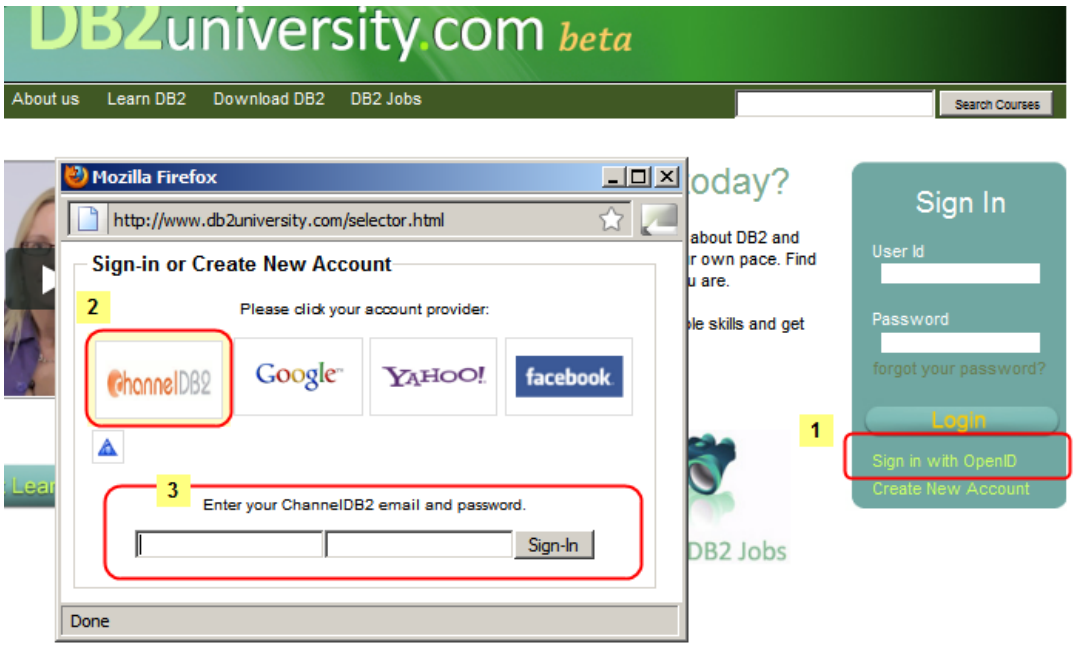

Figure 10.11 - OpenID sign-in process

First you click on the *Sign in with OpenID* link. A window will pop up showing you the different openID providers supported. Then you select one provider. In the example we chose *ChannelDB2*. Finally you input your ChannelDB2 user/email ID and the corresponding password. Only the first time you go through this process will you get another page from your provider asking for your consent to share information between the sites. If you accept, the information is shared, and you will be automatically registered to

db2university!. After this process every time you want to log on to **db2university.com** you follow the same steps.

An article available at IBM developerWorks explains you how the enablement of the openID providers were done.

10.4.3 Running on the Amazon Cloud

db2university.com is hosted on the Amazon cloud. This allows us to take full advantage of Cloud computing benefits, including setting up DB2 HADR for disaster recovery on the Cloud. These are the AWS resources we are using:

- EC2 standard large instance (64-bit, 4 ECUs, 7.5 GB of memory, 850GB instance storage)
- EBS volumes of 50 GB size
- S3 buckets of 3 and 1 GB size
- CloudFront

Three EC2 instances in the US-East region have been provisioned. One of them is used to run the Moodle application and the Web server, and another one to run the DB2 database server (DB2 HADR Primary server). The DB2 HADR stand-by server uses a third EC2 instance in a different availability zone within the US-East region as the disaster recovery site.

In terms of storage, we use instance storage, S3 and EBS volumes. *Figure 10.12* provides the details.

Figure 10.12 - DB2 storage on the Amazon cloud

In the figure:

- EBS volumes are used to store DB2 data and DB2 recovery logs. We chose 50GB to satisfy our needs. Optionally we are considering using RAID.
- A different small EBS volume of 2GB is used to store the DB2 code and the DB2 configuration files.
- Instance storage is used to temporarily store DB2 backup images. Later we move these backups to S3 for longer retention period.
- S3 storage is used for storing backup images and log archives for long retention periods. EBS volume snapshots are taken often, and they are stored in S3.

AWS CloudFront is used to allow users to download course videos and materials from an Amazon "edge" server that is closer to the location where the user lives. The files to download are first copied to a S3 bucket, and are later replicated to the edge servers.

In terms of software, this is the set up used:

- Ubuntu Linux 10.04.1 LTS
- DB2 Express 9.7.2 for Linux 64-bit
- Moodle 2.0 Release Candidate 1
- PHP 5.3.3

DB2 Express is used, as opposed to DB2 Express-C, because we are using the DB2 HADR feature, which is only available starting with the DB2 Express edition.

RightScale is ideal to manage AWS resources. RightScale is a partner of IBM, and its software gives the ability to build environments quickly using templates and scripts.

An article available at IBM developerWorks explains the AWS details used for db2university.

10.4.4 Using an Android phone to retrieve course marks

Moodle allows students to see their marks for assignments and exams on the Web; however, we have also developed a mobile application for the Android phone using App Inventor.

This simple application is illustrated in *Figure 10.13*. First you need to input a *mobile access code* which is a unique keyword that each student can obtain from their profile in Moodle. Next, they can choose the course they took or are currently taken, and display all their marks.

In order for this application to work, coding on the Client side using App Inventor, and on the server side (using a Web service created with Ruby/Sinatra) is required. An article available at IBM developerWorks explains you the details.

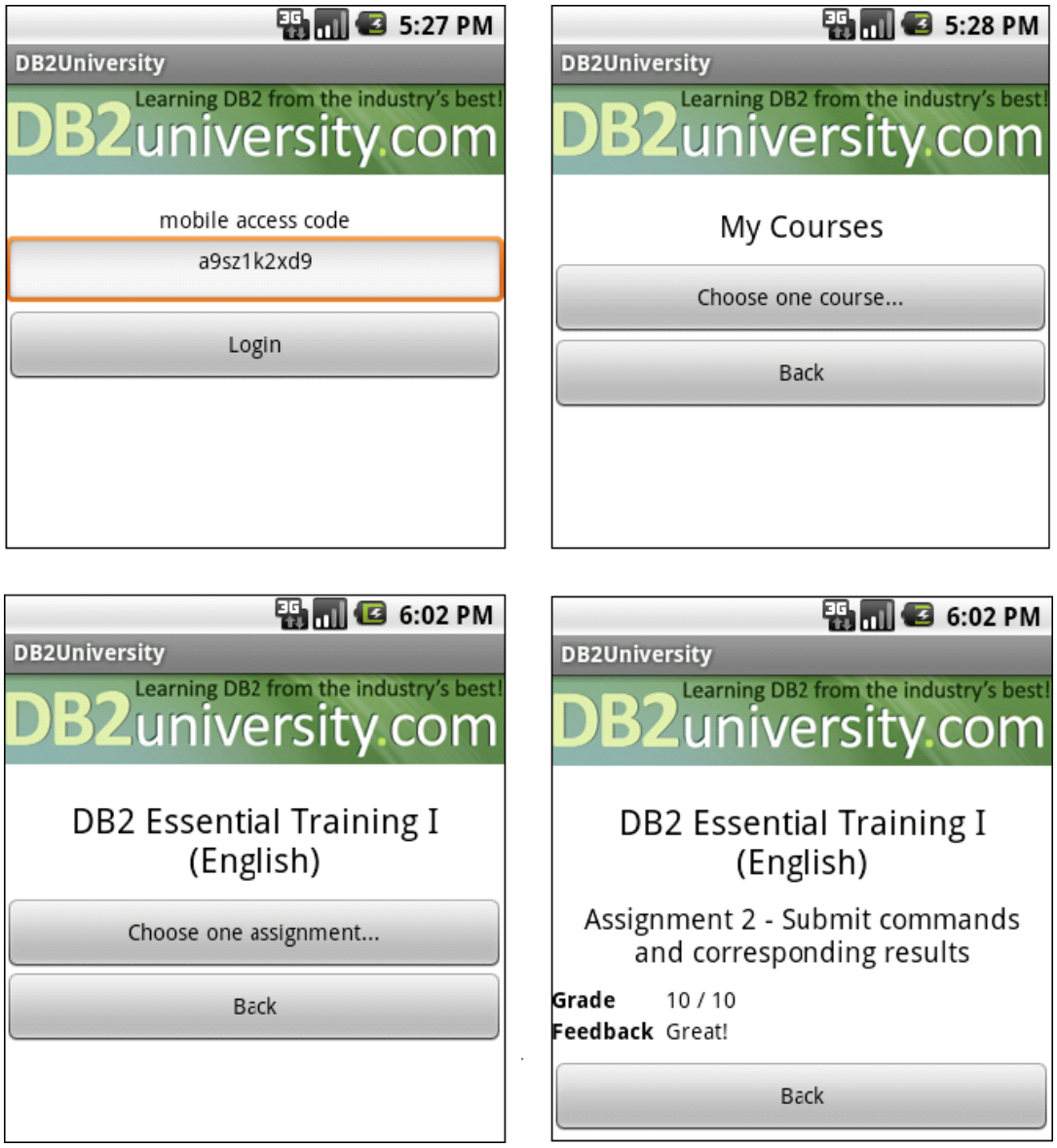

Figure 10.13 - App Inventor for Android application to access db2university

10.5 Summary

This chapter discussed important technology trends that will be realized by 2015, and highlighted the role of databases in these technologies. Cloud computing is at the top of the list, and is currently the hottest IT topic. Cloud computing is a new delivery method for IT resources allowing companies and individuals access to practically any amount of computer resources on-demand. Cloud computing is cost-effective since you only need to pay for what you consume.

Later in the chapter we talked about mobile applications. Mobile applications is another area of enormous growth. The chapter introduced you to different mobile device platforms and platforms for development.

The chapter then briefly discussed about business intelligence and appliances. Companies want to gain intelligence from the data they store to make better decisions for their businesses. At the same time, they do not want to have their in-house IT department spend large amount of time setting up their warehouses and configuring them. A data warehouse and business intelligence appliance such as IBM Smart Analytics system can help solve these issues.

Finally the chapter talked about **db2university.com** as a case study where many of the technologies described in the chapter were used.