**Mục lục**

[CHƯƠNG 1: TỔNG QUAN 3](#_Toc311060384)

[1.1. Đặt vấn đề 3](#_Toc311060385)

[1.2. Mục tiêu và phạm vi khóa luận 4](#_Toc311060386)

[1.2.1. Mục tiêu 4](#_Toc311060387)

[1.2.2. Phạm vi 4](#_Toc311060388)

[1.3. Kết quả dự kiến 4](#_Toc311060389)

[1.4. Cấu trúc khóa luận 4](#_Toc311060390)

[CHƯƠNG 2: ĐIỆN TOÁN ĐÁM MÂY 6](#_Toc311060391)

[2.1. Tổng quan về điện toán đám mây 6](#_Toc311060392)

[2.1.1. Tổng quan 6](#_Toc311060393)

[2.1.2. Ưu điểm của điện toán đám mây 8](#_Toc311060394)

[2.1.3. Nhược điểm của điện toán đám mây 9](#_Toc311060395)

[2.1.4. So sánh với kiến trúc client/server truyền thống 10](#_Toc311060396)

[2.2. Dịch vụ của điện toán đám mây 12](#_Toc311060397)

[2.2.1. Hạ tầng hướng dịch vụ (Iaas) 13](#_Toc311060398)

[2.2.2. Nền tảng hướng dịch vụ (PaaS) 13](#_Toc311060399)

[2.2.3. Phần mềm hướng dịch vụ (SaaS) 15](#_Toc311060400)

[2.3. Hình thức triển khai điện toán đám mây 16](#_Toc311060401)

[2.3.1. Đám mây công cộng (public cloud) 16](#_Toc311060402)

[2.3.2. Đám mây cá nhân (private cloud) 17](#_Toc311060403)

[2.3.3. Đám mây lai (hybrid cloud) 17](#_Toc311060404)

[2.3.4. Đám mây cộng đồng (community cloud) 18](#_Toc311060405)

[2.4. Windows Azure 19](#_Toc311060406)

[2.4.1. Kiến trúc tổng quan 19](#_Toc311060407)

[2.4.2. Dịch vụ tính toán (compute service) 20](#_Toc311060408)

[2.4.3. Dịch vụ lưu trữ (storage service) 22](#_Toc311060409)

[2.4.3.1. Blob 24](#_Toc311060410)

[2.4.3.2. Table 30](#_Toc311060411)

[2.4.3.3. Queue 35](#_Toc311060412)

[2.4.4. Fabric 36](#_Toc311060413)

[2.4.5. SQL Azure 39](#_Toc311060414)

[2.4.4.1.Tổng quan kiến trúc SQL Azure 40](#_Toc311060415)

[2.4.4.2. Ứng dụng của SQL Azure 42](#_Toc311060416)

[2.4.5. Các kịch bản sử dụng Windows Azure 42](#_Toc311060417)

[2.5. Ứng dụng trên điện toán đám mây 48](#_Toc311060418)

[2.6. Xu hướng phát triển của điện toán đám mây 49](#_Toc311060419)

[CHƯƠNG 3. SỔ LIÊN LẠC TRỰC TUYẾN 51](#_Toc311060420)

[3.1. Giới thiệu tổng quan 51](#_Toc311060421)

[3.2. Lí do chọn đề tài sổ liên lạc trực tuyến 52](#_Toc311060422)

[3.3. Tính năng chính 52](#_Toc311060423)

[3.4. Phân tích thiết kế 53](#_Toc311060424)

[3.4.1. Mô hình use case 53](#_Toc311060425)

[3.4.2. Sơ đồ class 57](#_Toc311060426)

[3.4.3. Thiết kế màn hình 73](#_Toc311060427)

[3.4.4. Thiết kế cơ sở dữ liệu 81](#_Toc311060428)

[3.4.4.1. Sơ đồ quan hệ 81](#_Toc311060429)

[3.4.4.2. Danh sách bảng 84](#_Toc311060430)

[3.4.4.3. Mô tả chi tiết bảng 85](#_Toc311060431)

[CHƯƠNG 4: HIỆN THỰC HỆ THỐNG 108](#_Toc311060432)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 108](#_Toc311060433)

[5.1. Kết luận 108](#_Toc311060434)

[5.2. Hướng phát triển 109](#_Toc311060435)

[TÀI LIỆU THAM KHẢO 109](#_Toc311060436)

[PHỤ LỤC 109](#_Toc311060437)

# CHƯƠNG 1: TỔNG QUAN

## 1.1. Đặt vấn đề

Ngày nay, đối với các công ty, doanh nghiệp, việc quản lý tốt và hiệu quả dữ liệu của riêng công ty cũng như dữ liệu khách hàng và đối tác là một trong những bài toán được ưu tiên hàng đầu và đang không ngừng gây khó khăn cho họ. Để có thể quản lý được nguồn dữ liệu đó, ban đầu các doanh nghiệp phải đầu tư, tính toán rất nhiều loại chi phí như chi phí cho phần cứng, phần mềm, mạng, chi phí cho quản trị viên, chi phí bảo trì, sửa chữa,… Ngoài ra họ còn phải tính toán khả năng mở rộng, nâng cấp thiết bị, phải kiểm soát việc bảo mật dữ liệu cũng như tính sẵn sàng cao của dữ liệu.

Từ một bài toán điển hình như vậy, chúng ta thấy được rằng nếu có một nơi tin cậy giúp các doanh nghiệp quản lý tốt nguồn dữ liệu đó, các doanh nghiệp sẽ không còn quan tâm đến cơ sở hạ tầng, công nghệ mà chỉ tập trung chính vào công việc kinh doanh của họ thì sẽ mang lại cho họ hiệu quả và lợi nhuận ngày càng cao hơn.

Thuật ngữ “điện toán đám mây” (“cloud computing”) ra đời bắt nguồn từ một trong những hoàn cảnh như vậy.

Thuật ngữ “điện toán đám mây” còn được bắt nguồn từ ý tưởng đưa tất cả mọi thứ như dữ liệu, phần mềm, tính toán, … lên trên mạng Internet. Chúng ta sẽ không còn trông thấy các máy tính cá nhân, máy chủ của riêng các doanh nghiệp để lưu trữ dữ liệu, phần mềm nữa mà chỉ còn một số các “máy chủ ảo” tập trung ở trên mạng. Các máy chủ ảo sẽ cung cấp các dịch vụ giúp cho doanh nghiệp có thể quản lý dữ liệu dễ dàng hơn, họ sẽ chỉ trả chi phí cho lượng sử dụng dịch vụ của họ, mà không cần phải đầu tư nhiều vào cơ sở hạ tầng cũng như quan tâm nhiều đến công nghệ. Xu hướng này sẽ giúp nhiều cho các công ty, doanh nghiệp vừa và nhỏ mà không có cơ sở hạ tầng mạng, máy chủ để lưu trữ và quản lý tốt dữ liệu.

Vậy “điện toán đám mây” là gì ? Nó có thể giải quyết bài toán trên như thế nào và có những đặc điểm nổi bật gì? Và tại sao các chuyên gia công nghệ thông tin trên thế giới lại cho rằng điện toán đám mây là một xu thế hiện đại và là cuộc cách mạng lần thứ ba của ngành công nghiệp công nghệ thông tin, sau cuộc cách mạng của máy tính cá nhân (thập kỉ 80) và cuộc cách mạng internet (thập kỉ 90)?

Mong muốn làm rõ các vấn đề trên và xây dựng một ứng dụng minh họa, đó chính là lí do mà nhóm chúng tôi quyết định chọn và thực hiện đề tài cho khóa luận tốt nghiệp là điện toán đám mây và ứng dụng sổ liên lạc trực tuyến.

## 1.2. Mục tiêu và phạm vi khóa luận

### 1.2.1. Mục tiêu

- Mục tiêu của khóa luận là tìm hiểu về điện toán đám mây bao gồm kiến trúc tổng quan, lợi ích mà điện toán đám mây mang đến và cách thức vận hành của nó.

- Các dịch vụ mà điện toán đám mây cung cấp và cách thức triển khai của điện toán đám mây.

- Từ các kiến thức trên, nhóm chúng tôi áp dụng vào việc xây dựng ứng dụng triển khai trên đám mây.

### 1.2.2. Phạm vi

- Phạm vi của khóa luận là trình bày lí thuyết về kiến thức tổng quan của điện toán đám mây, các dịch vụ và các mô hình triển khai điện toán đám mây.

- Trình bày lí thuyết tổng quan về hệ điều hành đám mây Windows Azure của Microsoft, cũng như các thành phần cơ bản của nó.

- Xây dựng ứng dụng sổ liên lạc trực tuyến để minh họa cho một trong các dịch vụ của điện toán đám mây là phần mềm hướng dịch vụ (Software as a Service, viết tắt là SaaS).

## 1.3. Kết quả dự kiến

- Trình bày kiến thức tổng quan của điện toán đám mây nói chung và hệ điều hành đám mây Windows Azure nói riêng.

- Xây dựng thành công ứng dụng sổ liên lạc trực tuyến và triển khai nó lên hệ điều hành Windows Azure của Microsoft để áp dụng cho các trường học.

## 1.4. Cấu trúc khóa luận

Khóa luận bao gồm các phần sau:

Chương 1, đặt vấn đề và nói lên lí do chọn đề tài khóa luận, trình bày mục đích, phạm vi của khóa luận và kết quả dự kiến đạt được.

Chương 2, trình bày kiến thức về điện toán đám mây. Nội dung chương này bao gồm trình bày kiến thức tổng quan về điện toán đám mây, các dịch vụ mà điện toán đám mây cung cấp, các hình thức triển khai điện toán đám mây, tổng quan hệ điều hành đám mây Windows Azure của Microsoft cũng như các thành phần cơ bản của nó là dịch vụ tính toán, các dịch vụ lưư trữ, Fabric và SQL Azure.

Chương 3, trình bày ứng dụng minh họa sổ liên lạc trực tuyến. Nội dung cụ thể chương này bao gồm: phát biểu bài toán, lí do chọn ứng dụng sổ liên lạc trực tuyến để minh họa và nội dung phân tích thiết kế của ứng dụng.

Chương 4, trình bày hiện thực hệ thống, bao gồm kiến trúc hệ thống, hiện thực và triển khai ứng dụng.

Chương 5, kết luận đề tài và hướng phát triển của đề tài. Nội dung kết luận bao gồm kết luận về chương trình và kinh nghiệm đạt được từ việc xây dựng hệ thống.

# CHƯƠNG 2: ĐIỆN TOÁN ĐÁM MÂY

## 2.1. Tổng quan về điện toán đám mây

### 2.1.1. Tổng quan

Theo Wikipedia:

*“*Điện toán đám mây là một mô hình điện toán có khả năng co giãn linh động và các tài nguyên thường được ảo hóa để cung cấp như một dịch vụ trên mạng Internet*”*.1

Theo Ian Foster:

*“*Điện toán đám mây là một mô hình điện toán phân tán có tính co giãn lớn hướng về mặt kinh tế, là nơi chứa các sức mạnh tính toán, kho lưu trữ, các nền tảng và các dịch vụ được trực quan, ảo hóa và co giãn linh động, sẽ được phân phối theo nhu cầu cho các khách hàng bên ngoài thông qua Internet”.

“Điện toán đám mây là một dạng thức điện toán cung cấp các tài nguyên ảo hóa và có quy mô dưới dạng dịch vụ qua mạng Internet. Người dùng không cần tới những kiến thức chuyên môn để quản lý hạ tầng công nghệ này bởi phần việc đó là dành cho các nhà cung cấp dịch vụ”.

“Điện toán đám mây là sự kết hợp giữa các khái niệm: Hạ tầng hướng dịch vụ (IaaS), Nền tảng hướng dịch vụ (PaaS), Phần mềm hướng dịch vụ (SaaS) và một số khái niệm công nghệ mới. Dịch vụ điện toán đám mây thường cung cấp các trực tuyến ứng dụng doanh nghiệp thông dụng, có thể truy xuất qua trình duyệt web trong khi phần mềm và dữ liệu được lưu trữ trên máy chủ của nhà cung cấp*”.*2

1 <http://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_to%C3%A1n_%C4%91%C3%A1m_m%C3%A2y>

2 http://laocai.gov.vn/sites/sotttt/tintucsukien/tinhoatdong/Trang/20111108084216.aspx



Hình 2.1 - Mọi thứ đều tập trung vào đám mây

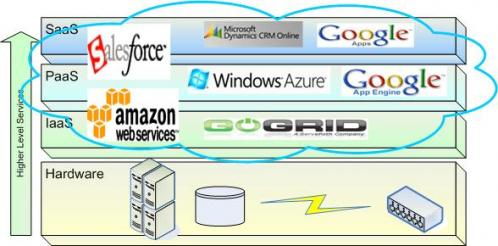
**Lịch sử xuất hiện của điện toán đám mây:**

Thuật ngữ điện toán đám mây xuất hiện bắt nguồn từ ứng dụng điện toán lưới (grid computing) trong thập niên 1980, tiếp theo là điện toán theo nhu cầu (utility computing) và phần mềm dịch vụ (SaaS).

Điện toán lưới đặt trọng tâm vào việc di chuyển một tải công việc (workload) đến địa điểm của các tài nguyên điện toán cần thiết để sử dụng. Một lưới là một nhóm máy chủ mà trên đó nhiệm vụ lớn được chia thành những tác vụ nhỏ để chạy song song, được xem là một máy chủ ảo.

Với điện toán đám mây, các tài nguyên điện toán như máy chủ có thể được định hình động hoặc cắt nhỏ từ cơ sở hạ tầng phần cứng nền và trở nên sẵn sàng thực hiện nhiệm vụ, hỗ trợ những môi trường không phải là điện toán lưới như Web ba lớp chạy các ứng dụng truyền thống hay ứng dụng Web 2.0.

**Kiến trúc tổng quát của điện toán đám mây:**

****

**Hình 2.2 – Kiến trúc tổng quan điện toán đám mây**

Kiến trúc của điện toán đám mây bao gồm 3 tầng :

* Hệ thống hướng dịch vụ ( Infrastructure as Service ).
* Nền tảng hướng dịch vụ ( Platform as Service).
* Phần mềm hướng dịch vụ ( Software as Service).

Các hình thức triển khai của điện toán đám mây: dựa vào nhu cầu của người dùng cũng như của phía nhà cung cấp mà hình thành nên bốn phương thức triển khai điện toán đám mây trong thực tế :

* Đám mây công cộng ( Public Cloud)
* Đám mây cá nhân ( Private Cloud)
* Đám mây lại ( Hybrid Cloud)
* Đám mây cộng đồng ( Community Cloud)

Điện toán đám mây đang được phát động bới nhiều nhà cung cấp, trong đó có Amazon, Google, DataSynapse, và Salesforce cũng như những nhà cung cấp truyền thống như Sun Microsystems, HP, IBM, Intel và Microsoft. Nó đang được nhiều người dùng cá nhân cho đến những công ty lớn như General Electric, L’Oréal, Procter & Gamble và Valeo chấp nhận và sử dụng 3.

### 2.1.2. Ưu điểm của điện toán đám mây

**-** Tiết kiệm và **giảm chi phí**: chi phí đầu tư ban đầu về cơ sở hạ tầng, máy móc và nguồn nhân lựa của người sử dụng điện toán đám mây được giảm đến mức thấp nhất. Khi khách hàng ngừng sử dụng dịch vụ thì tài nguyên sẽ được nhà cung cấp giải phóng và cho khách hàng khác thuê lại. Khách hàng sẽ tận dụng được nguồn vốn dùng để đầu từ cho thiết bị cho các công việc kinh doanh khác.

- Tốc độ xử lý nhanh: người dùng của dịch vụ điện toán đám mây sẽ tận dụng được sức mạnh mà các siêu máy tính của nhà cung cấp mang lại.

- Đa phương tiện: không còn phụ thuộc vào thiết bị và vị trí địa lý, điện toán đám mây phép người dùng truy cập và sử dụng hệ thống thông qua trình duyệt web ở bất kỳ đâu và trên bất kỳ thiết bị nào mà họ sử dụng (chẳng hạn là PC hoặc là điện thoại di động, v.v....)

- Chia sẻ tài nguyên và chi phí trên một địa bàn rộng lớn, mang lại các lợi ích cho người dùng như:

* Công suất xử lý nhanh hơn do tài nguyên được tập trung. Ngoài ra, người dùng không cần phải đầu tư về nguồn nhân lực quản lý hệ thống.

3 http://www.hoituso.net

* Khả năng khai thác và hiệu suất được cài thiện hơn 10-20% so với hệ thống máy tính cá nhân thông thường.

**-** Độ tin cậy cao: điện toán đám mây không chỉ giành cho người dùng phổ thông, mà còn phù hợp với các yêu cầu cao và liên tục của các công ty kinh doanh và các nghiên cứu khoa học. Tuy nhiên, một vài dịch vụ lớn của điện toán đám mây đôi khi rơi vào trạng thái quá tải, khiến hoạt động bị ngưng trệ. Khi rơi vào trạng thái này, người dùng không có khả năng để xử lý các sự cố mà phải nhờ vào các chuyên gia từ đám mây tiến hành xử lý.

**-** Tính co giãn: khả năng mở rộng, co giãn các tài nguyên giúp khách hàng dễ dàng cơ cấu lại hoạt động của mình khi có sự thay đổi về quy mô cũng như phương thức hoạt động.

- Bảo mật: khả năng bảo mật được cài thiện do sự tập trung về dữ liệu.

- Bảo trì và sửa chữa: các ứng dụng của điện toán đám mây dễ dàng để sửa chữa hơn bởi lẽ chúng không được cài đặt cố định trên một máy tính nào.

- Thống kê tài nguyên: tài nguyên sử dụng của điện toán đám mây luôn được quản lý và thống kê trên từng khách hàng và ứng dụng, theo từng ngày, từng tuần, từng tháng. Điều này đảm bảo cho việc định lượng giá cả của mỗi dịch vụ do điện toán đám mây cung cấp để người dùng có thể lựa chọn phù hợp.

### 2.1.3. Nhược điểm của điện toán đám mây

- Phụ thuộc vào nha cung cấp:

Ngày nay, khả năng tương tác giữa các nền tảng khác nhau của các phần mềm đã được cải thiện, nhưng các hàm API (Application Programming Interface) của Cloud Computing vẫn chưa được chuẩn hóa nên nếu một người dùng viết một ứng dụng trên nền tảng của nhà cung cấp dịch vụ thì ứng dụng đó không thể chạy được trên nền tảng của nhà cung cấp dịch vụ khác. Như vậy người dùng phụ thuộc nhà cung cấp dịch vụ là điều bất lợi.

Nếu các các nhà cung cấp Cloud Computing Service cùng nhau chuẩn hóa API, người dùng có thể phát triển ứng dụng trên nền tảng của nhiều nhà cung cấp dịch vụ. Nếu hệ thống cung cấp dịch vụ nào đó gặp sự cố thì dữ liệu người dùng không mất vì nó đã nằm đâu đó trên hệ thống của các nhà cung cấp dịch vụ khác. Nếu như cách này được thực hiện, chắc chắn sẽ dẫn đến cuộc cạnh tranh về giá, đến đây khi lựa chọn dịch vụ người ta phải cân nhắc. Thứ nhất, chất lượng dịch vụ tương xứng với giá mà người sử dụng trả cho nhà cung cấp dịch vụ, dù đắt người ta vẫn dùng, tức là “đắt xắt ra miếng”. Thứ hai, phụ thuộc vào nhà cung cấp và chuẩn hóa các API sẽ dẫn đến khả năng: “cơ sở hạ tầng, phần mềm có thể chạy trên private cloud hoặc public cloud”.

Khi người dùng lưu trữ dữ liệu trên hệ thống của nhà cung cấp dịch vụ, câu hỏi đặt ra là: có gì đảm bảo dữ liệu an toàn, không rò rỉ? Về mặt công nghệ, hiện nay vẫn chưa có cách nào hiệu quả để giải quyết vấn đề trên, do đó người dùng thường chỉ lựa chọn những nhà cung cấp lơn và có uy tín.

**-** Bảo mật và kiểm tra dữ liệu:

Như đã nói ở trên, dữ liệu lưu trên đám mây có an toàn không? Nhưng chắc chắn xác suất bị người khác khác truy xuất rất cao, đây thực sự là một thách thức trong bảo mật dữ liệu. Trước hết người dùng phải mã hóa dữ liệu trước khi đưa lên đám mây để lưu trữ, khi sử dụng tất nhiên phải giải mã trên PC của họ. Người dùng ghi nhận thông tin hệ thống đã xử lý cùng với sử dụng các hệ điều hành ảo khi cung cấp dịch vụ IaaS sẽ làm cho ứng dụng của mình khó bị tấn công hơn.

**-** Tắc nghẽn trên đường truyền dữ liệu và hiệu quả PC:

Có những ứng dụng khi bắt đầu chạy thì dữ liệu ít, càng về sau dữ liệu càng nhiều, có ứng dụng chạy trên Cloud và có thể lưu ở các vị trí khác nhau, khi chạy ứng dụng này phát sinh “vận chuyển dữ liệu giữa các data center”. Người dùng phải trả phí vận chuyển dữ liệu giữa các data center, ứng dụng chạy càng về sau thì chi phí này càng tăng lên, đây là điều phải cân nhắc.

Khi nhiều máy tính ảo cùng chạy, thì vấn đề chia sẽ về CPU hay bộ nhớ đạt hiệu quả cao, nhưng vấn đề giao tiếp IO của các máy ảo này gây ra nhiều vấn đề liên quan đến hiệu suất máy tính.

**-** Nhu cầu lưu trữ người dùng:

Mặc dù điện toán đám mây đáp ứng linh hoạt nhu cầu lưu trữ của người nhưng lại gây khó khăn trong quản lý hệ thống lưu trữ, chẳng hạn một người sử dụng mua một khoảng dung lượng thì phải cung cấp cho người đó bao nhiêu là tối ưu, vừa đủ cho người dùng hay nhiều hơn yêu cầu, nếu cơ chế quản lý không phù hợp sẽ dẫn đến tình trạng phân mảnh dữ liệu, dẫn đến quá trình truy xuất sẽ khó khắn hơn.Như vậy, làm sao tạo ra một hệ thống lưu trữ tiện lợi, đáp ứng vụ nhu cầu và khả năng lưu trữ của người sử dụng đang là vấn đề phức tạp phải giải quyết của các nhà cung cấp Cloud Computing Service.

### 2.1.4. So sánh với kiến trúc client/server truyền thống

Điện toán đám mây "mở" hơn, quan trọng hơn, giá rẻ hơn,.v.v.. Như vậy rất tiện lợi cho các doanh nghiệp và tổ chức hoạt động vì giảm chi phí. Để trả lời ngắn gọn câu hỏi: “Điện toán đám mây có thật sự hữu ích cho một tổ chức, công ty hay không?” chúng ta phải xem xét: quy mô, hình thức hoạt động,v.vv...Đặc biệt nếu dữ liệu là vô cùng quan trọng đối với họ, chắc hắn người ta sẽ xây dựng một hệ thống lưu trữ riêng và nó được vận hành như hệ thống lưu trữ của các ngân hàng. Đối với các doanh nghiệp nhỏ mà dữ liệu không quan trọng lắm, điện toán đám mây sẽ là giải pháp tối ưu do họ không phải chi phí đầu tư, quản lý, bảo trì, vận hành hệ thống lưu trữ.

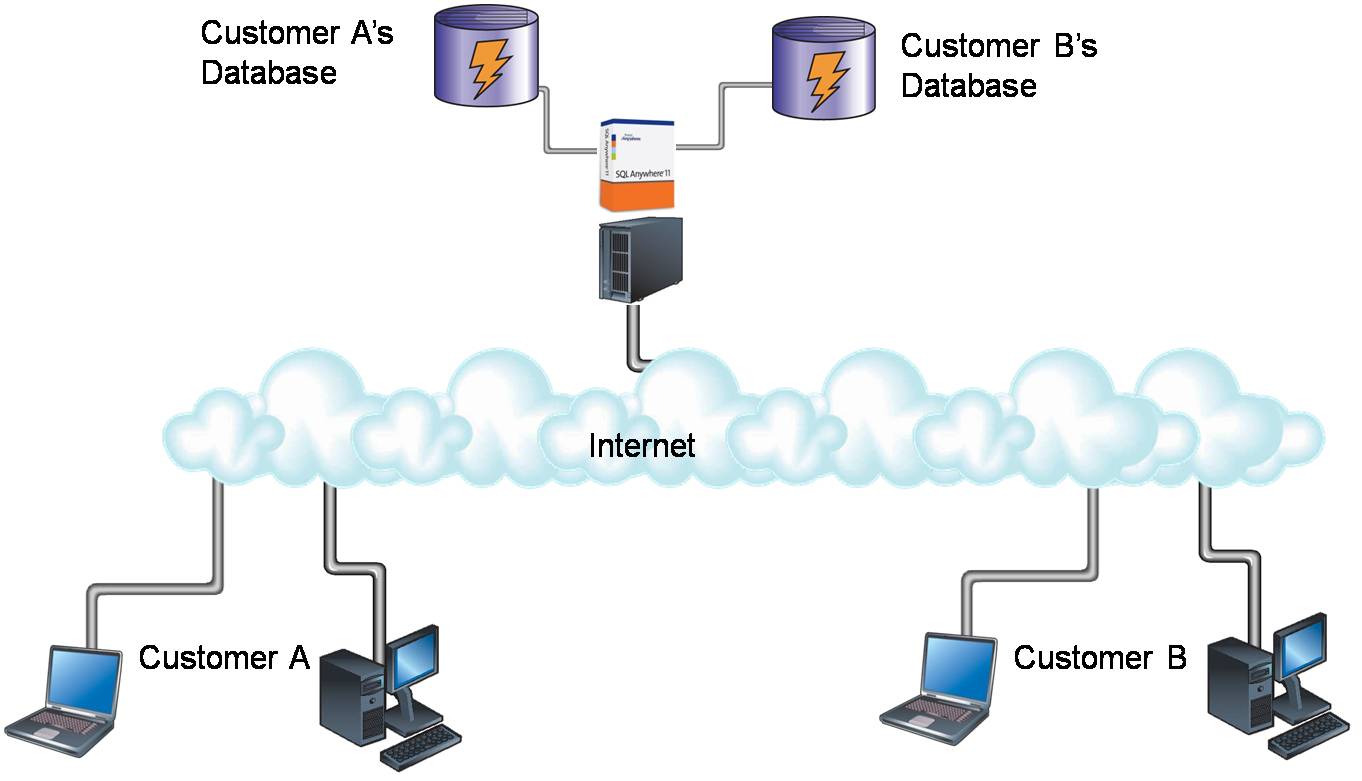
Điện toán đám mây có năm tính chất nổi bật so với mô hình truyền thống:

**-** Khả năng co giãn (Rapid elasticity): Một đặc tính nổi bật của điện toán đám mây là khả năng tự động mở rộng hoặc thu nhỏ hệ thống theo yêu cầu người dùng (hệ thống sẽ tự mở rộng hoặc thu hẹp bằng cách thêm hoặc giảm bớt tài nguyên). Một người dùng ký hợp đồng thuê một Server gồm 4 CPU. Nếu lượng truy cập thấp chỉ cần 1 CPU là đủ, khi đó hệ thống quản lý của nhà cung cấp dịch vụ sẽ tự ngắt bớt 3 CPU, người dùng không phải trả phí cho 3 CPU nói trên và chúng được đưa sang phục vụ người dùng khác. Đến khi nhu cầu tăng tức là lượng truy cập tăng, hệ thống ngay lạp tức sẽ tự động thêm CPU vào, nếu nhu cầu vượt quá 4 CPU thì người dùng trả phí theo hợp đồng đã ký với nhà cung cấp dịch vụ điện toán đám mây. Khả năng co giãn nhanh và linh hoạt giúp cho nhà cung cấp dịch vụ điện toán đám mây tận dụng tài nguyên dư thừa phục vụ được nhiều khách hàng, người dùng giảm chi phí vì họ chỉ phải trả tiền cho những tài nguyên thực sự dùng.

**-** Dịch vụ theo nhu cầu (On-demand self-service): Người dùng gửi yêu cầu thông qua trang web cung cấp dịch vụ, hệ thống của nhà cung cấp sẽ đáp ứng để người dùng có thể tự phục vụ như: tăng – giảm thời gian sử dụng server và dung lượng lưu trữ, v.v.. mà không cần phải trực tiếp yêu cầu nhà cung cấp dịch vụ, tức là mọi nhu cầu khách hàng đều được xử lý trên internet.

**-** Truy xuất diện rộng: dịch vụ điện toán đám mây (cloud computing service) là tập hợp các dịch vụ công nghệ thông tin được cung cấp thông qua môi trường internet, ở đó người dùng thích dịch vụ gì thì dùng dịch vụ ấy, dùng bao nhiêu trả bấy nhiêu, được lựa chọn những dịch vụ tốt nhất ở bất cứ đâu vào bất cứ lúc nào. Như vậy người dùng có kết nối internet là có thể sử dụng dịch vụ, dịch vụ điện toán đám mây không yêu cầu người dùng phải có khả năng xử lý cao, người dùng có thể truy xuất bằng các thiết bị di dộng như điện thoại, PDA và laptop.

**-** Dùng chung tài nguyên và điều tiết dịch vụ: nhà cung cấp dịch vụ cho phép người dùng dùng chung tài nguyên do họ cung cấp dựa trên mô hình “multi-tenant”, tài nguyên được phân phát rất linh hoạt tùy theo nhu cầu của người dùng. Khi nhu cầu của một người dùng nào đó giảm xuống, lập tức phần tài nguyên dư thừa sẽ được phục vụ cho người dùng khác. Nếu một người dùng 4 CPU từ 7 - đến 11 giờ hàng ngày, một người dùng khác thuê 4 CPU tương tự 13 giờ đến 17 giờ hàng ngày thì họ có thể dùng chung 4 CPU đó. Cloud Computing Service dựa trên công nghệ ảo hóa, tài nguyên ở đây đa phần là tài nguyên ảo, chúng được cấp phát linh hoạt tùy theo nhu cầu (động) của từng người dùng khác nhau, nhà cung cấp dịch vụ có thể phục vụ nhiều người dùng hơn so với cách cấp phát tài nguyên (tĩnh) truyền thống. Hệ thống Cloud Computing Service tự động kiểm soát và tối ưu hóa sử dụng tài nguyên bao gồm: dung lượng lưu trữ, đơn vị xử lý, băng thông,.. Lượng tài nguyên sử dụng có thể được theo dõi, kiểm soát và báo cáo một cách minh bạch cho cả hai phía nhà cung cấp dịch vụ và người dùng.



**Hình 2.3 - Mô phỏng cách quản lý tài nguyên của người sử dụng**

## 2.2. Dịch vụ của điện toán đám mây

Hình dưới đây cho thấy những dịch vụ trên đám mây được phân thành ba nhóm chính.



**Hình 2.4 – Kiến trúc tổng quát của đám mây**

### 2.2.1. Hạ tầng hướng dịch vụ (Iaas)

Hạ tầng hướng dịch vụ cung cấp cơ sở hạ tầng máy tính (thường là môi trường ảo) như là một dịch vụ. Thay vì phải mua máy chủ, phần mềm hay thiết bị mạng, khách hàng có thể mua các tài nguyên như là một dịch vụ bên ngoài.

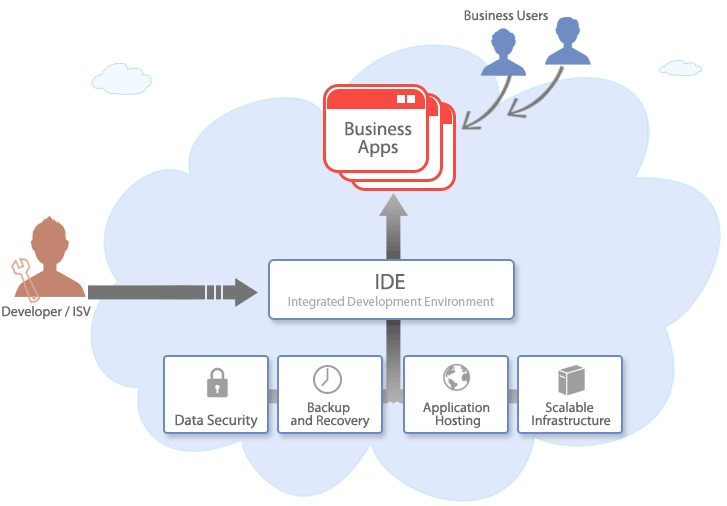
Các lợi ích mà dịch vụ này mang lại:

* Cung cấp tài nguyên như là dịch vụ: bao gồm cả máy chủ, thiết bị mạng, bộ nhớ, CPU, không gian đĩa cứng, trang thiết bị trung tâm dữ liệu.
* Khả năng mở rộng linh hoạt.
* Chi phí thay đổi tùy theo thực tế.
* Nhiều người thuê có thể cùng dùng chung trên một tài nguyên.
* Về cấp độ doanh nghiệp: đem lại lợi ích cho công ty bởi một nguồn tài nguyên tích toán tổng hợp.

### 2.2.2. Nền tảng hướng dịch vụ (PaaS)

Nền tảng hướng dịch vụ hỗ trợ việc triển khai ứng dụng mà không quan tâm đến chi phí hay sự phức tạp của việc trang bị và quản lý các lớp phần cứng và phần mềm bên dưới, cung cấp tất cả các tính năng cần thiết để hỗ trợ chu trình sống đầy đủ của việc xây dựng và cung cấp một ứng dụng và dịch vụ web sẵn sàng trên internet mà không cần bất kì thao tác tải hay cài đặt phần mềm cho những người phát triển, quản lý tin học hay người dùng cuối.

Khi PaaS có sẵn như một dịch vụ, những người phát triển phần mềm có thể kiểm soát toàn bộ việc phát triển và triển khai ứng dụng. PaaS cho phép những người phát triển phần mềm tạo ra các ứng dụng web tùy chỉnh và phát hành nó một cách nhanh chóng, khi nhiều rắc rối như việc thiết lập hosting, servers, databases, quá trình tương tác người dùng và những frameworks được đóng gói.



**Hình 2.5 – Platform as Service**

Một số đặc trưng của PaaS bao gồm:

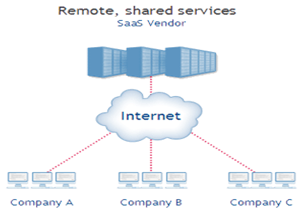
* Phục vụ cho việc phát triển, kiểm thử, triển khai và vận hành ứng dụng giống như là môi trường phát triển tích hợp
* Cung cấp các công cụ khởi tạo với giao diện trên nền web
* Có kiến trúc đồng nhất
* Tích hợp dịch vụ web và cơ sở dữ liệu
* Hỗ trợ cộng tác nhóm phát triển
* Cung cấp các công cụ hỗ trợ tiện tích khác

Việc sử dụng dịch vụ này mang lại một số lợi ích:

* Ưu điểm trong những dự án tập hợp những công việc nhóm có sự phân tán về địa lý.
* Khả năng tích hợp nhiều nguồn của dịch vụ web.
* Giảm chi phí ngoài lề khi tích hợp các dịch vụ về bảo mật, khả năng mở rộng, kiểm soát lỗiv.vv...
* Giảm chi phí khi trừu tượng hóa công việc lập trình ở mức cao để tạo dịch vụ, giao diện người dùng và các yếu tố ứng dụng khác.
* Hướng việc sử dụng công nghệ để đạt được mục đích tạo điều kiện dễ dàng hơn cho việc phát triển ứng dụng đa người dùng cho những người không chỉ trong nhóm lập trình mà có thể kết hợp nhiều nhóm cùng làm việc.

### 2.2.3. Phần mềm hướng dịch vụ (SaaS)

Phần mềm hướng dịch vụ là một mô hình triển khai ứng dụng mà ở đó nhà cung cấp cho phép người dụng sử dụng dịch vụ theo yêu cầu. Những nhà cung cấp SaaS có thể lưu trữ ứng dụng trên máy chủ của họ hoặc tải ứng dụng xuống thiết bị khách hàng, vô hiệu hóa nó sau khi kết thúc thời hạn. Các chức năng theo yêu cầu có thể được kiểm soát bên trong để chia sẻ bản quyền của một nhà cung cấp ứng dụng thứ ba.



**Hình 2.6 – Software as a Service**

Một số ứng dụng sử dụng dịch vụ này:

* Quản lí quan hệ khách hàng (Customer Relationship Management, viết tắt là CRM). Đây là một phương pháp giúp các doanh nghiệp tiếp cận và giao tiếp với khách hàng một cách có hệ thống và hiệu quả, quản lý các thông tin của khách hàng như thông tin về tài khoản, nhu cầu, liên lạcv.vv... nhằm phục vụ khách hàng tốt hơn.
* Dịch vụ hội thảo trực tuyến ( Video Conferencing )
* Kế toán
* Hệ quản trị nội dung web DotNetNuke4

Lợi ích lớn nhất của dịch vụ này mang lại là chi phí thấp. Nhà cung cấp dịch vụ có thể đưa ra các ứng dụng rẻ hơn và đáng tin hơn.

Ngoài ra còn một số lợi ích khác như:

* Sử dụng ít nhân viên

4 <http://webdong.com/detail.aspx?id=44>

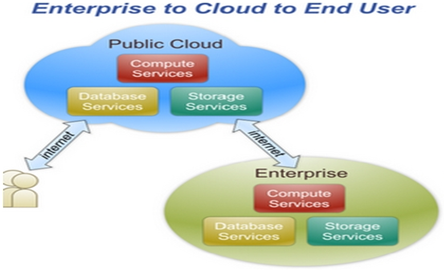
* Sự tùy chỉnh: những ứng dụng trước đây rất khó tùy chỉnh và đòi hỏi hải cập nhật các bản vá lỗi. Ứng dụng SaaS dễ dàng tùy chỉnh và có thể đáp ứng chính xác yêu cầu của tổ chức
* Bảo mật: SSL (Secure Sockets Layer) được sử dụng rộng rãi và tin cậy

## 2.3. Hình thức triển khai điện toán đám mây

### 2.3.1. Đám mây công cộng (public cloud)

Là các dịch vụ điện toán đám mây được một bên thứ ba cung cấp. Chúng tồn tại ngoài tường lửa công ty và được lưu trữ đầy đủ và được nhà cung cấp đám mây quản lý. Các đám mây công cộng cố gắng cung cấp cho người dùng với các phần tử công nghệ thông tin tốt nhất. Cho dù đó là phần mềm, cơ sở hạ tầng ứng dụng hoặc cơ sở hạ tầng vật lý, nhà cung cấp đám mây chịu trách nhiệm về cài đặt, quản lý, cung cấp và bảo trì. Khách hàng chỉ chịu phí cho các tài nguyên nào mà họ sử dụng, vì thế cái chưa sử dụng được loại bỏ.

Tất nhiên điều này liên quan đến chi phí. Các dịch vụ này thường được cung cấp với quy ước về cấu hình, nghĩa là chúng được phân phối với ý tưởng cung cấp các trường hợp sử dụng phổ biến nhất. Các tùy chọn cấu hình thường là một tập hợp con nhỏ hơn so với những gì mà chúng đã có nếu nguồn tài nguyên đã được người tiêu dùng kiểm soát trực tiếp. Một điều khác cần lưu ý là kể từ khi người tiêu dùng có quyền kiểm soát một chút trên cơ sở hạ tầng, các quy trình đòi hỏi an ninh chặt chẽ và tuân thủ quy định dưới luật không phải lúc nào cũng thích hợp cho các đám mây chung.

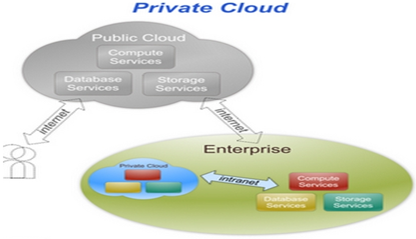


**Hình 2.7 – Đám mây công cộng**

### 2.3.2. Đám mây cá nhân (private cloud)

Là các dịch vụ đám mây được cung cấp trong doanh nghiệp. Những đám mây này tồn tại bên trong tường lửa công ty và chúng được doanh nghiệp quản lý. Các đám mây riêng đưa ra nhiều lợi ích giống như các đám mây chung thực hiện với sự khác biệt chính: doanh nghiệp có trách nhiệm thiết lập và bảo trì đám mây này.

Sự khó khăn và chi phí của việc thiết lập một đám mây bên trong đôi khi có thể có chiều hướng ngăn cản việc sử dụng và chi phí hoạt động liên tục của đám mây có thể vượt quá chi phí của việc sử dụng một đám mây chung.Các đám mâycá nhân đưa ra nhiều lợi thế hơn so với loại cộng cộng. Việc kiểm soát chi tiết hơn trên các tài nguyên khác nhau đang tạo thành một đám mây mang lại cho công ty tất cả các tùy chọn cấu hình có sẵn. Ngoài ra, các đám mây riêng là lý tưởng khi các kiểu công việc đang được thực hiện không thiết thực cho một đám mây công cộng, do đúng với các mối quan tâm về an ninh và về quản lý.



**Hình 2.8 – Đám mây cá nhân**

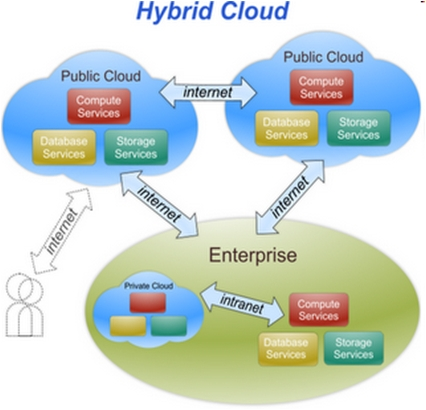
### 2.3.3. Đám mây lai (hybrid cloud)

Là một sự kết hợp của các đám mây công cộng và cá nhân. Những đám mây này thường do doanh nghiệp tạo ra và các trách nhiệm quản lý sẽ được phân chia giữa doanh nghiệp và nhà cung cấp đám mây công cộng. Đám mây lai sử dụng các dịch vụ có trong cả không gian công cộng và cá nhân.

Các đám mây lai là câu trả lời khi một công ty cần sử dụng các dịch vụ của cả hai đám mây riêng và công cộng. Theo hướng này, một công ty có thể phác thảo các mục tiêu và nhu cầu của các dịch vụ và nhận được chúng từ đám mây công cộng hay cá nhân, khi thích hợp.

Một đám mây lai được xây dựng tốt có thể phục vụ các quy trình nhiệm vụ-tới hạn, an toàn, như nhận các khoản thanh toán của khách hàng, cũng như những thứ là không quan trọng bằng kinh doanh, như xử lý bảng lương nhân viên.

Hạn chế chính với đám mây này là sự khó khăn trong việc tạo ra và quản lý có hiệu quả một giải pháp như vậy. Phải có thể nhận được và cung cấp các dịch vụ lấy từ các nguồn khác nhau như thể chúng có nguồn gốc từ một chỗ và tương tác giữa các thành phần riêng và chung có thể làm cho việc thực hiện thậm chí phức tạp hơn nhiều. Do đây là một khái niệm kiến trúc tương đối mới trong điện toán đám mây, nên cách thực hành và các công cụ tốt nhất về loại này tiếp tục nổi lên và bất đắc dĩ chấp nhận mô hình này cho đến khi hiểu rõ hơn.

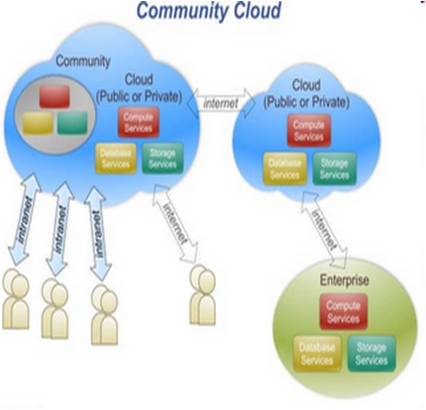


**Hình 2.9 – Đám mây lai**

### 2.3.4. Đám mây cộng đồng (community cloud)

Là các đám mây được chia sẻ bởi một số tổ chức và hỗ trợ một cộng đồng cụ thể có mối quan tâm chung (ví dụ như chung sứ mệnh, yêu cầu an ninh, chính sách, v.v... ). Nó có thể được quản lý bởi các tổ chức hoặc một bên thứ ba.

Một đám mây cộng đồng có thể được thiết lập bởi một số tổ chức có yêu cầu tương tự và tìm cách chia sẻ cơ sở hạ tầng để thực hiện một số lợi ích của điện toán đám mây. Tùy chọn này là tốn kém hơn nhưng có thể đáp ứng về sự riêng tư, an ninh hoặc tuân thủ các chính sách tốt hơn.



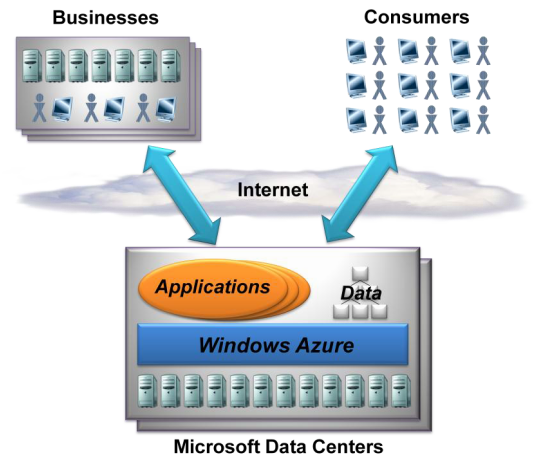
**Hình 2.10 – Đám mây cộng đồng**

## 2.4. Windows Azure

### 2.4.1. Kiến trúc tổng quan

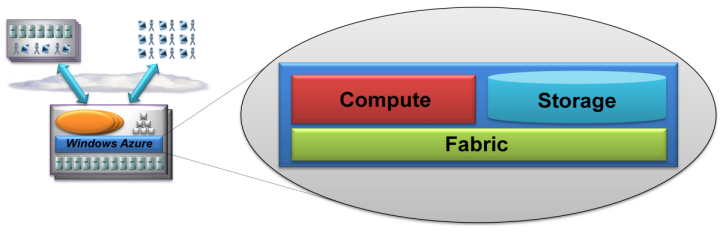
Với xu hướng phát triển của điện toán đám mây hiện này, dĩ nhiên là Microsoft không đứng sau trào lưu này. Microsoft đã cho ra đời Windows Azure Platform nhằm mang đến cho cộng đồng các nhà phát triển ứng dụng cơ hội được xây dựng và cung cấp các dịch vụ trực tuyến trên nền tảng cơ sở hạ tầng Windows.

Nhìn một cách tổng quan, Windows Azure là một hệ điều hành dùng để chạy các ứng dụng Windows và lưu dữ liệu của nó trên đám mây. Nhưng khác với một hệ điều hành bình thường, người dùng phải cài đặt và chạy trên máy tính của mình, Windows Azure là một dịch vụ, tức là khách hàng dùng nó để chạy ứng dụng, lưu trữ dữ liệu trên các máy ở trung tâm dữ liệu Microsoft và có thể truy cập qua internet. Các ứng dụng này có thể cung cấp dịch vụ cho doanh nghiệp, khách hàng hay cả hai.



Hình 2.11 - Ứng dụng Windows Azure chạy trên trung tâm dữ liệu của Microsoft

Windows Azure gồm các thành phần ba thành phần cơ bản: dịch vụ tính toán (computer service), dịch vụ lưu trữ (storage service) và Fabric.



Hình 2.12 - Các thành phần của Windows Azure: Compute Service, Storage Service và Fabric

Giống như tên của nó, dịch vụ tính toán (compute service) sẽ chạy ứng dụng trong khi dịch vụ lưu trữ (storage service) lưu dữ liệu. Thành phần thứ ba, Windows Azure Fabric, cung cấp cách thức thông dụng để quản lý và theo dõi các ứng dụng sử dụng nển tảng đám mây này.

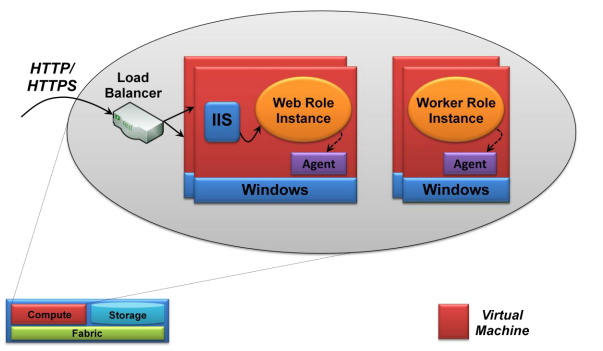
### 2.4.2. Dịch vụ tính toán (compute service)

Dịch vụ tính toán Windows Azure có thể chạy nhiều kiểu ứng dụng khác nhau. Mục tiêu chính của kiến trúc này là hỗ trợ các ứng dụng có lượng người sử dụng truy cập đồng thời cực lớn. Có thể đạt được mục tiêu này bằng cách tăng cường sử dụng nhiều máy chủ lớn hơn. Nhưng thay vì như vậy, Windows Azure được thiết kế để hỗ trợ ứng dụng giảm xuống, chạy nhiều bản sao của cùng một mã nguồn trên nhiều máy chủ khác nhau.

Để đạt được điều này, ứng dụng Windows Azure có thể có nhiều thể hiện (instance), mỗi thể hiện (instance) được thực thi trên một máy ảo.

Để chạy một ứng dụng, lập trình viên truy cập vào Windows Azure portal5 thông qua trình duyệt, đăng nhập với tài khoản Windows Live ID. Sau đó, lập trình viên tạo ra một tài khoản hosting để chạy ứng dụng, hoặc một tài khoản lưu trữ (storage) để lưu trữ dữ liệu hoặc cả hai. Một khi lập trình viên có tài khoản hosting thì có thể upload ứng dụng của mình, chỉ ra bao nhiêu thể hiện mà ứng dụng cần, cũng như cấu hình của máy ảo. Windows Azure sẽ tạo ra các máy ảo tương ứng để chạy ứng dụng. Lập trình viên, chỉ có thể thấy được trạng trái của ứng dụng được triển khai, thông qua Windows Azure portal. Một khi ứng dụng được triển khai, nó hoàn toàn được quản lý bởi Windows Azure. Điều duy nhất lập trình viên phải làm là chỉ ra các thông số sử dụng cho ứng dụng. Còn lại, việc triển khai, tính mở rộng, tính sẵn sàng, nâng cấp, chuản bị phần cứng server đều được thực hiện bởi Windows Azure cho các ứng dụng đám mây.

Compute service hỗ trợ 2 loại thể hiện, một loại gọi là Web role và một loại gọi là Worker role.



Hình 2.13: Ứng dụng Windows Azure có thể chứa Web role hoặc Worker role hoặc cả hai.

5 http://windows.azure.com

Một thể hiện Web role có thể chấp nhận một request HTTP/HTTPS. Để cho phép điều này, nó chạy trên một máy ảo có Internet Information Services (IIS) 7. Lập trình viên có thể tạo ra Web role bằng ASP.NET, WCF hay bất kì kĩ thuật .NET nào có thể hoạt động được với IIS 7. Ngoài ra, lập trình viên có thể viết các ứng dụng với native code - việc sử dụng .NET Framework là không yêu cầu. Có nghĩa là có thể upload và chạy các ứng dụng sử dụng kĩ thuật khác, ví dụ PHP và Java. Khi một request được gửi đến Web role, nó sẽ được truyền qua bộ cân bằng tải đến các thể hiện của Web role trong cùng một ứng dụng. Do đó, không đảm bảo rằng, các yêu cầu từ một người dùng có thể được gởi đến cùng một thể hiện của ứng dụng.

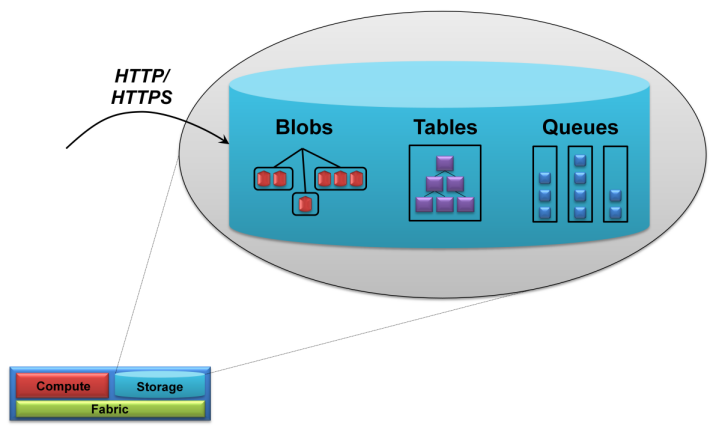
Một thể hiện Worker role không giống như Web role, nó không chấp nhận request từ bên ngoài, các máy ảo của nó không chạy IIS. Một Worker role cho bạn khả năng để chạy các xử lý ngầm liên tục trên đám mây. Một Worker role có thể làm việc với queue, table, blob trong dịch vụ lưu trữ. Nó chạy hoàn toàn độc lập với thể hiện Web role, mặc dù có thể cùng thuộc một phần của dịch vụ. Việc liên lạc giữa Web role và Worker role có thể thông qua queue của dịch vụ lưu trữ.

Lập trình viên có thể chỉ sử dụng thể hiện Web role hay Worker role hoặc kết hợp cả hai để tạo ra ứng dụng Windows Azure. Có thể sử dụng Windows Azure portal để thay đổi số lượng thể hiện của Web role, Worker role tùy theo yêu cầu của ứng dụng.

Khi chạy các thể hiện Web role hay Worker role, các máy ảo cũng chạy đồng thời các tác nhân (Windows Azure agent). Các tác nhân để phục vụ cho sự tương tác hệ giữa các thể hiện với Windows Azure Fabric. Các agent này trình bày các API được định nghĩa để các thể hiện có thể làm một số việc như: ghi chép, tìm thư mục gốc của tài nguyên lưu trữ cục bộ trên máy ảo của nó.

### 2.4.3. Dịch vụ lưu trữ (storage service)

Dịch vụ lưu trữ trong Windows Azure hỗ trợ ba kiểu dịch vụ: blob, table và queue.



Hình 2.14 - Bộ lưu trữ Windows Azure

Những kiểu dịch vụ này hỗ trợ cục bộ cũng như truy cập trực tiếp thông qua REST API. Bảng so sánh giữa các kiểu lưu trữ này:

|  |  |  |  |
| --- | --- | --- | --- |
| **Đặc trưng** | **Blob** | **Queue** | **Table** |
| **URL schema** | http://[Storage  Account].blob.core.wi  ndows.net/[Container  Name]/[Blob Name] | http://[Storage  Account].queue.core.w  indows.net/[Queue  Name] | http://[Storage  Account].table.core.windows.  net/[Table  Name]?$filter=[Query] |
| **Kích thước tối đa** | 50GB | 8K (String) | Thiết kế cho Terabytes dữ liệu |
| **Sử dụng** | Lưu trữ dữ liệu binary lớn | Giao tiếp giữa các dịch vụ | Lưu trữ dữ liệu có cấu trúc kích thước nhỏ |
| **Tham khảo API** | <http://msdn.microsoft.com/en-us/library/dd135733.aspx> | <http://msdn.microsoft.com/en-us/library/dd179363.aspx> | <http://msdn.microsoft.com/en-us/library/dd179423.aspx> |

**Bảng 2.1 - So sánh các kiểu lưu trữ**

Cách đơn giản nhất để lưu trữ dữ liệu trong Windows Azure storage là sử dụng Blob. Một blob chứa dữ liệu nhị phân. Blob cấu trúc lưu trữ đơn giản như sau: mỗi tài khoản lưu trữ có một hoặc nhiều container, mỗi container chứa một hoặc nhiều blob. Kích thước blob có thể lớn đến 50GB và chúng có thể chứa thêm metadata, ví dụ như nơi chụp của tấm ảnh hay ca sĩ thể hiện bài hát trong file MP3.

Bộ lưu trữ Windows Azure cũng cung cấp Table. Tuy nhiên, nó không phải là bảng quan hệ như trong SQL. Thực tế, dữ liệu lưu trữ bên trong nó là một hệ thống các thực thể với các thuộc tính. Hơn cả việc sử dụng SQL, một ứng dụng có thể truy cập dữ liệu của Table bằng ADO.NET data Service hoặc LINQ. Một bảng có thể sẽ rất lớn, với hàng tỉ thực thể chứa hàng terabyte dữ liệu. Và bộ lưu trữ Windows Azure có thể phân vùng cho nó qua nhiều máy chủ khác nhau để tăng hiệu suất.

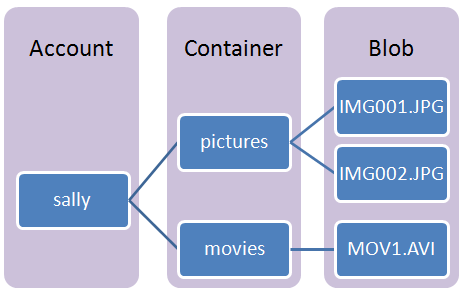
Cả blob và table đều phục vụ cho mục đích lưu trữ và truy xuất dữ liệu. Trong khi đó, queue nhằm phục vụ mục đích khác - đó là sự liên lạc giữa Web role và Worker role. Trong đó, Web role khi nhận được yêu cầu từ người dùng, sẽ ghi thông điệp vào một hàng đợi, mô tả công việc cho Worker role. Một Worker role chờ thông điệp này, lấy thông điệp, và thực hiện tác vụ yêu cầu.

Bộ lưu trữ Windows Azure có thể được truy cập từ một ứng dụng Windows Azure hoặc từ một ứng dụng khác. Trong cả 3 trường hợp, cả ba cách lưu trữ của dịch vụ lưu trữ Windows Azure đều có thể sử dụng REST để truy xuất dữ liệu. Mọi thứ đều được đặt tên qua URL và được truy xuất thông qua các thao tác HTTP chuẩn. Ngoài ra, còn hỗ trợ ADO.NET Data Service, nhưng chỉ có thể sử dụng cho các ứng dụng .NET, còn các ứng dụng khác ví dụ Java thì chỉ có thể sử dụng REST.

#### 2.4.3.1. Blob

Các thành phần của Windows Azure Blob.

**Tài khoản lưu trữ** – Tất cả truy xuất đến bộ lưu trữ Windows Azure được thực hiện thông qua tài khoản lưu trữ. Một tài khoản có thể có nhiều container.



Hình 2.15 - Bộ lưu trữ Blob

* **Blob Container** – Một container bao gồm nhiều blob.
* Chính sách chia sẻ được thiết lập ở cấp độ container. Hiện tại hai thuộc tính "Public READ" và "Private" được hỗ trợ. Khi một container có thuộc tính là “Public READ”,thì tất cả nội dung của nó có thể được đọc bởi bất kì người dùng mà không yêu cầu chứng thực. Khi một container có thuộc tính là “Private”, thì chỉ có người sở hữu tương ứng mới có thể truy xuất blob trong container với truy xuất được chứng thực.
* Một container có thể có metadata kết hợp với chúng. Metadata ở dạng cặp <tên, giá trị>, và chúng lên đến 8KB kích thước với mỗi container.
* **Blob** – Blob được lưu trữ trong Container. Kích thước mỗi blob có thể lên đến 50GB. Một blob có một tên duy nhất trong một container. Blob có thể có metadata kết hợp với chúng, cặp <tên, giá trị>, và chúng lên đến 8KB kích thước cho mỗi blob.

Đường dẫn URI cho một blob được cấu trúc như sau:

http://<account>.**blob**.core.windows.net/<container>/<blobname>

Tất cả các giao tiếp đến Windows Azure Blob được thực hiện thông qua giao tiếp HTTP REST PUT/GET/DELETE.

Các lệnh HTTP/REST được hỗ trợ để thực thi các thao tác blob bao gồm :

* PUT Blob – Thêm một blob mới hoặc ghi đè blob đã tồn tại.
* GET Blob – Lấy một thực thể blob, hoặc lấy một dãy các byte trong blob sử dụng thao tác GET chuẩn HTTP.
* DELETE Blob – Xóa một blob đã tồn tại.
* Copy Blob – Sao chép một blob từ blob nguồn đến blob đích trong cùng tài khoản lưu trữ. Nó sẽ sao chép toàn bộ blob, bao gồm metadata, thuộc tính, và blocklist. Bạn có thể sử dụng CopyBlob cùng với DeleteBlob để đổi tên một blob, để di chuyển một blob giữa các containers, hoặc tạo bảng sao dự phòng cho blob đang tồn tại của bạn.
* Get Block List - Lấy danh sách các block đã được tải lên như một phần của blob. Có hai loại danh sách block:
  + Danh sách block đã commit – Đây là danh sách block đã được commit thành công qua PutBlockList vào một blob.
  + Danh sách block chưa commit – Đây là danh sách các block đã được tải lên cho một blob những chưa được commit vào blob.

Bạn có thể tải một blob có kích thước lên đến 64MB chỉ sử dụng một request PUT blob . Để tải lên các blob có kích thước lên đến 50GB, bạn cần phải ghép các block lại với nhau.

Một trong những chiến lược chính cho Windows Azure Blob là cho phép tải lên các blob có kích thước lên đến nhiều GB. Điều này được cung cấp bởi Windows Azure Blob qua những bước sau:

* Cắt Blob cần tải lên thành nhiều khối liên tiếp nhau. Vd, Một phim 10GB có thẻ cắt thành 2500 khối, mỗi khối có kích thước 4MB, khối đầu tiên đại diện cho byte thứ 1 đến 4194304, khối thứ 2 đại diện cho byte 4194305 đến 8388608.
* Đặt mỗi khối một Tên/ ID duy nhất. ID duy nhất này được giới hạn bởi tên blob được tải lên. Ví dụ, block đầu tiên được gọi là “Block 0001”, block thứ hai “Block 0002”, v.v...
* PUT mỗi block vào đám mây.
* Sau khi tất cả các block đã được lưu trữ trong bộ lưu trữ Windows Azure, sau đó chúng ta commit danh sách các block đã tải lên vào blob mà chúng thuộc về. Điều này được thực hiện với một lênh PUT chỉ định URL ở trên với truy vấn xác định rằng đây là lệnh Blocklist. Sau đó, header HTTP chứa danh sách các block đã được commit cho khối này. Khi thao tác này thực hiện thành công, danh sách các block, theo thứ tự mà chúng được liệt kê, bây giờ đại diện cho phiên bản có thể đọc được của các blob. Blob sau đó có thể được đọc bằng cách sử dụng lệnh GET blob.

Hình dưới đây kết hợp các khối vào khái niệm dữ liệu Windows Azure Blob.

Hình 2.16 - Khái niệm lưu trữ Blob – Thêm Block

Như mô tả trước đó, blob có thể được truy xuất qua PUT và GET bằng cách sử dụng URL sau:

http://<account>.**blob**.core.windows.net/<container>/<blobname>

Ví dụ hình 2, một lệnh PUT có thể được sử dụng để đặt các ảnh theo URL sau:

http://sally.**blob**.core.windows.net/pictures/IMG001.JPG

http://sally.**blob**.core.windows.net/pictures/IMG002.JPG

Khi sử dụng một lệnh PUT đơn, block có kích thước lên đến 64MB có thể được lưu trữ. Để lưu trữ các blob lớn hơn 64MB và lên đến 50GB, đầu tiên cần PUT tất cả các block, và sau đó PUT blocklist để hợp thành phiên bản có thể đọc được của blob này. Trong hình 2 ở trên,chỉ sau khi các block đã được put và commit thành blob, thì blob có thể được đọc theo URL sau:

http://sally.**blob**.core.windows.net/pictures/MOV1.AVI

**Trừu tượng hóa dữ liệu Block**

Mỗi block được xác định bởi Block ID và nó được giới hạn bởi tên blob. Vì thế những blob khác nhau có thể có các block với cùng ID. Mỗi block có kích thước lên đến 4MB, và các block trong cùng blob có thể có kích thước khác nhau. Windows Azure Blob cung cấp các thao tác cấp độ Block như sau:

* PUT block – tải một block vào một blob. Chú ý một block được tải lên thành công với thao tác PUT block không trở thành một phần của blob chỉ khi nào nó được commit vào blocklist với thao tác PUT blocklist.
* PUT blocklist – commit một blob bằng cách xác định danh sách các ID block tạo nên blob. Các block được xác định trong hoạt động này phải được tải lên thành công bằng cách sử dụng lời gọi PUT block. Thứ tự các block trong thao tác PUT blocklist, sẽ hợp thành phiên bản có thể đọc của blob.
* GET blocklist – lấy danh sách block đã được commit trước đó cho blob bằng thao tác PUT blocklist. Danh sách block được trả ra định rõ ID và kích thước của mỗi block. Chức năng này có thể được sử dụng để lấy các danh sách block chưa được commit.

Chú ý : ID Block có thể được thấy qua một phần của metadata mà bạn có thể lưu vết cho mỗi block.

Một ví dụ tận dụng ID block là có được ID block ID qua giá trị băm của nội dung dữ liệu qua mỗi block. Trong phương pháp này, đây là một trong những cách có thể kiểm tra tính toàn vẹn dữ liệu của mỗi block dữ liệu nhận được từ hệ thống.

**Các chiến lược tải blok lên**

Tải lên một blob theo danh sách các block chúng ta có được các lợi ích sau:

* Sự nối tiếp – khi mỗi block được tải lên thành công điều này có thể chắc chắn thành công của khối này và thử lại khối nếu có thất bại và tiếp tục từ điểm đó.
* Tải lên song song – điều này có thể tải các block lên song song làm giảm thời gian tải của các blob rất lớn.
* Tải lên không theo thứ tự - Bạn có thể tải các block lên mà không theo thứ tự.
* Kết hợp BlockID như Metadata cho mỗi block.

**PutBlock**(BlockId1);

**PutBlock**(BlockId3);

**PutBlock**(BlockId4);

**PutBlock**(BlockId2);

**PutBlock**(BlockId4);

**PutBlockList**(BlockId2, BlockId3, BlockId4);

Hình 2.17 - Chiến lược tải block lên

Ví dụ hình 4 sẽ được sử dụng để giải thích các chiến lược khác nhau có thể gặp khi sử dụng giao tiếp block để tải lên các blob. Ở đó là:

* Tải lên các block với cùng Block ID – Khi các block với cùng ID được tải lên cùng blob, block được úp lên gần nhất cùng với ID sẽ được sử dụng khi commit blob với PUT blocklist. Trong ví dụ trên, hai block với ID = BlockId4 được tải lên và block sau cùng sẽ được sử dụng trong block list commit cho blob.
* Tải các block lên không theo thứ tự - Block có thể được tải lên theo thứ tự khác nhau như trong danh sách block tải lên blob. Trong ví dụ ở trên, danh sách block được commit cuối cùng là những block theo thứ tự BlockId2, BlockId3, BlockId4, nhưng những block này được upload theo thứ tự khác nhau. Dữ liệu blob có thể đọc được (qua GET) được sắp xếp liên quan đến danh sách được đặt tả trong PUT blocklist. Do đó, nếu bạn lấy một blob và đọc blob từ lúc bắt đầu đến khi kết thúc bạn sẽ thấy nội dung của BlockId2, theo sau la BlockId3, sau nữa là Block ID 4.

Các Block không sử dụng – Hơn nữa, một vài block có thể không bao giờ lấy được commit vào danh sách block cuối cùng trong blob. Các block này sẽ được dọn dẹp bởi hệ thống. BlockId1 và Block với ID BlockId4 đầu tiên trong ví dụ sẽ được thu dọn.

#### 2.4.3.2. Table

Windows Azure Table là bộ lưu trữ có cấu trúc được cung cấp bởi nền tảng Windows Azure. Nó hỗ trợ các bảng có khả năng mở rộng cực lớn trên đám mây, nó có thể chứa hàng tỉ thực thể và terabyte dữ liệu. Hệ thống sẽ tự động mở rộng một cách hiệu quả đến hàng ngàn máy chủ.

Bộ lưu trữ có cấu trúc được cung cấp dưới dạng bảng, mỗi bảng chứa một tập các thực thể, mỗi thực thể chứ một tập các thuộc tính được đặt tên. Một số điểm nổi bật của Windows Azure Table :

* Hỗ trợ LINQ, dịch vụ dữ liệu ADO .NET và REST.
* Tập hợp các kiểu dữ liệu phong phú cho các giá trị thuộc tính.
* Hỗ trợ số lượng bảng và thực thể không giới hạn và không bị giới hạn về kích thước bảng.
* Dùng mô hình optimistic concurrency cho những thao tác cập nhật và xóa.
* Đối với các truy vấn lấy ra số lượng lớn các kết quả hoặc truy vấn bị ngắt quãng. Từng phần kết quả được trả về với một cái thẻ nối tiếp cho phép truy vấn lại tiếp tục tại nơi mà nó đã dừng lại.

**Mô hình dữ liệu bảng**

Sau đây tóm tắt các mô hình dữ liệu cho Windows Azure Table:

* **Storage Account (Tài khoản lưu trữ) -** Ứng dụng phải sử dụng một tài khoản hợp lệ để truy xuất bộ lưu trữ Windows Azure. Bạn có thể tạo một tài khoản mới qua trang web của Windows Azure. Người dùng sẽ nhận được một khóa bí mật 256-bit cho mỗi tài khoản được tạo. Khóa bí mật này dùng để chứng thực những truy vấn từ người dùng đến hệ thống lưu trữ. Đặc biệt, một chữ kí HMAC SHA256 cho mỗi request sử dụng khóa bí mật này. Chữ ký được chuyển qua với mỗi request để xác thực các request của người dùng.

Tên tài khoản là một phần của hostname trên đường dẫn URL. Hostname để truy xuất các bảng là <accountName>.table.core.windows.net.

* **Table (Bảng)** – chứa một bộ các thực thể. Một ứng dụng có thể tạo ra nhiều bảng chỉ với một tài khoản lưu trữ.
* **Entity (Thực thể)** – Thực thể (một thực thể tương tự như một dòng) là mẫu dữ liệu cơ bản được lưu trữ trong một bảng. Một thực thể chứa một tập các thuộc tính. Mỗi bảng có 2 thuộc tính tên là “PartitionKey” và “RowKey” tạo ra khóa duy nhất cho thực thể.
* **Property (Thuộc tính)** – Nó đại diện cho một giá trị duy nhất trong thực thể. Tên thuộc tính phân biệt hoa, thường. Một tập hợp phong phú các kiểu dữ liệu được hỗ trợ cho giá trị thuộc tính.
* **PartitionKey** – Đây là thuộc tính khóa đầu tiên của mỗi bảng. Hệ thống sử dụng khóa này để tự động cấp phát các thực thể của bảng qua nhiều nút lưu trữ.
* **RowKey** – là thuộc tính khóa thứ hai của bảng. Đây là ID duy nhất của thực thể trong phân vùng mà nó thuộc về. PartitionKey kết hợp với RowKey xác định duy nhất một thực thể trong một bảng.
* **Timestamp** – là thuộc tính có giá trị **Datetime**, được lưu giữ bởi hệ thống để ghi lại thời điểm thực thể thay đổi cuối cùng.
* **Partition** – Một tập hợp các thực thể trong một bảng với cùng giá trị khóa PartitionKey.
* **Sort Order (Thứ tự sắp xếp)** – Các thực thể trong bảng được sắp xếp theo PartitionKey vào sau đó là theo RowKey. Điều này có nghĩa là các truy vấn xác định theo các khóa này sẽ hiệu quả hơn, và tất cả các kết quả được trả về được sắp xếp theo PartitionKey và sau đó theo RowKey.

Một bảng có một giản đồ linh hoạt. Windows Azure Table lưu giữ tên và kiểu dữ liệu cho mỗi thuộc tính trong một thực thể.

Sau đâu là một số thông tin chi tiết bổ sung cho Bảng, Thực thể và Thuộc tính :

* Bảng
  + Tên bảng chỉ chứa các kí tự và số.
  + Tên bảng không bắt đầu với một kí tự số.
  + Tên bảng phân biệt hoa – thường.
  + Tên bảng phải có độ dài từ 3 đến 63 kí tự.
* Tên thuộc tính

Chỉ được chứa các kí tự chữ cái và dấu ‘\_’.

* Một thực thể có thể có tối đa 255 thuộc tính bao gồm các thuộc tính bắt buộc - PartitionKey, RowKey và Timestamp. Tất cả các thuộc tính khác trong thực thể có tên được định nghĩa theo nhu cầu ứng dụng.
* PartitionKey và RowKey có kiểu chuỗi, và mỗi khóa bị giới hạn trong kích thước 1 KB.
* Timestamp là thuộc tính chỉ đọc.
* Không có giản đồ cố định – Không có giản đồ được lưu trữ bởi Windows Azure Table, vì thế tất cả các thuộc tính được lưu trữ theo cặp <tên, kiểu dữ liệu>. Điều này có nghĩa rằng 2 thực thể trong cùng một bảng có thể có thuộc tính khác nhau. Một bảng có thể có hai thực thể với tên thuộc tính giống nhau, nhưng có kiểu dữ liệu khác nhau cho giá trị thuộc tính. Tuy nhiên, tên thuộc tính phải là duy nhất với mỗi thực thể.

• Kết hợp kích thước của tất cả các dữ liệu trong một thực thể không thể vượt quá 1MB. Kích thước này bao gồm kích thước của tên thuộc tính cũng như kích thước của giá trị thuộc tính hoặc kiểu của chúng, bao gồm cả hai thuộc tính khoá bắt buộc (PartitionKey và RowKey).

* Hỗ trợ các kiểu thuộc tính: Binary, Bool, DateTime, Double, GUID, Int, Int64, String.

Xem bảng bên dưới để biết giới hạn.

| **Kiểu thuộc tính** |  |
| --- | --- |
| Binary | Kiểu mảng các byte 64 KB. |
| Bool | Kiểu Boolean. |
| DateTime | Kiểu thời gian 64-bit. Miền giá trị được hỗ trợ từ ngày 1/1/1601 đến 12/31/9999. |
| Double | Kiểu số thực 64-bit. |
| GUID | Kiểu dữ liệu để chỉ định ID 128-bit. |
| Int | Kiểu số nguyên 32-bit. |
| Int64 | Kiểu số nguyên 64-bit. |
| String | Kiểu chuỗi. Giá trị chuỗi có thể lên đến 64 KB. |

**Phân vùng bảng**

Windows Azure Table cho phép các bảng mở rộng đến hàng ngàn nút lưu trữ bằng cách phân phối các thực thể trong bảng. Khi phân phối các thực thể, nó mong muốn đảm bảo một tập hợp các thực thể luôn nằm cùng trên một nút lưu trữ. Một ứng dụng điều chỉnh thiết lập này bằng cách chọn giá trị thích hợp cho thuộc tính PartitionKey trong mỗi thực thể.

Hình 2.18 - Ví dụ phân vùng

Hình mình họa mô tả một bảng chứa nhiều phiên bản của nhiều tài liệu khác nhau. Mỗi thực thể trong bảng tương ứng với một phiên bản của một tài liệu cụ thể. Trong ví dụ này, PartitionKey của bảng là tên tài liệu, và RowKey là tên phiên bản. Tên tài liệu cùng với tên phiên bản xác định duy nhất một thực thể trong bảng. Trong ví dụ này, tất cả các phiên bản của cùng một tài liệu tổ chức vào một phân vùng riêng lẻ.

**Tác động của việc phân vùng**

Bây giờ chúng ta mô tả mục đích của các phân vùng và làm thế nào để chọn một PartitionKey.

**Khả năng mở rộng của bảng**

Hệ thống lưu trữ đạt được khả năng mở rộng khá lớn bằng cách phân bố các phân vùng qua nhiều nút lưu trữ. Hệ thống quan sát mô thức sử dụng các phân vùng, và tự động cân bằng các phân vùng này qua tất cả các nút lưu trữ. Điều này cho phép hệ thống và ứng dụng của bạn mở rộng để đáp ứng nhu cầu lưu thông của các bảng. Nghĩa là, nếu có nhiều lưu thông đến một vài phân vùng, hệ thống sẽ tự động lan chúng ra đến nhiều nút lưu trữ, vì thế tải lưu thông sẽ được lan ra qua nhiều máy chủ. Tuy nhiên, một phân vùng ,có nghĩa là với tất cả các thực thể có cùng PartitionKey, sẽ được đáp ứng bởi một nút lưu trữ. Tuy nhiên, lượng dữ liệu được lưu trữ trong một phân vùng không bị giới hạn bởi dung lượng lưu trữ của một nút.

**Giao dịch nhóm thực thể**

Đối với các thực thể được lưu trữ trong cùng một bảng và cùng một phân vùng, ứng dụng có thể thực hiện một giao dịch bao hàm nhiều thực thể. Điều này cho phép giao dịch thực hiện nhiều thao tác Thêm/ Xóa/ Sửa qua nhiều thực thể với chỉ một request batch đến hệ thống lưu trữ, với điều kiện tất cả các thực thể ở cùng phân vùng trong một bảng. Hoặc tất cả các thao tác thực hiện thành công trong một giao dịch hoặc chúng bị thất bại hoàn toàn.

**Miền thực thể**

Các thực thể trong một phân vùng được lưu trữ cùng nhau. Điều này cho phép truy vấn hiệu quả trong một phân vùng.

Trong ví dụ trên, tất cả các phiên bản của cùng một tài liệu tổ chức trong một phân vùng riêng. Do đó, việc truy xuất “tất cả các phiên bản của một tài liệu xác định” sẽ đạt hiệu quả cao, bởi vì chúng ta truy xuất đến một phân vùng duy nhất. Mặt khác, một truy vấn “tất cả các phiên bản của các tài liệu được chỉnh sửa trước 5/30/2007” không bị giới hạn ở một phân vùng duy nhất. Bởi vì truy kiểm tra tất cả các phân vùng trên nhiều nút lưu trữ, một truy vấn như vậy sẽ tốn nhiều chi phí hơn.

**Chọn một PartitionKey**

Việc lựa chọn một PartitionKey đóng vai trò quan trọng để một ứng dụng có khả năng mở rộng tốt. Có một sự đánh đổi ở đây giữa việc cố gắng để được lợi từ miền thực thể, ở đó bạn đạt được những truy vấn hiệu quả qua các thực thể trong cùng một phân vùng, và khả năng mở rộng của các bảng, ở đây càng nhiều phân vùng trong bảng thì càng dễ dàng ở cho Windows Azure Table lan truyền tải qua nhiều máy chủ.

**Giao dịch nhóm thực thể**

Nếu ứng dụng của bạn cần sử dụng giao dịch nhóm thực thể, một PartitionKey cần phải được lựa chọn để qui mô của nó có thể bao trùm hết tất cả các thực thể bạn cần thực hiện giao dịch trên đó. Tùy theo nhu cầu truy vấn của bạn, cần chọn PartitionKey mà nó chỉ nhóm cùng các thực thể cần được nhóm lại với nhau, để thực hiện giao dịch nhóm thực thể trên chúng.

**Truy vấn hiệu quả**

Chúng ta nhận thấy rằng truy vấn có tần suất cao sử dụng PartitionKey như một điều kiện lọc truy vấn. Sử dụng ParititionKey trong bộ lọc truy vấn hạn chế thực thi truy vấn đến một phân vùng đơn hoặc thành phần con của phân cùng (tùy thuộc vào điều kiện sử dụng), do đó cải thiện được hiệu suất truy vấn.

Nếu PartitionKey không phải là một phần của truy vấn, thì truy vấn phải đi qua tất cả các phân vùng trong bảng để tìm ra thực thể cần tìm, điều này thực sự không mang lại hiệu quả.

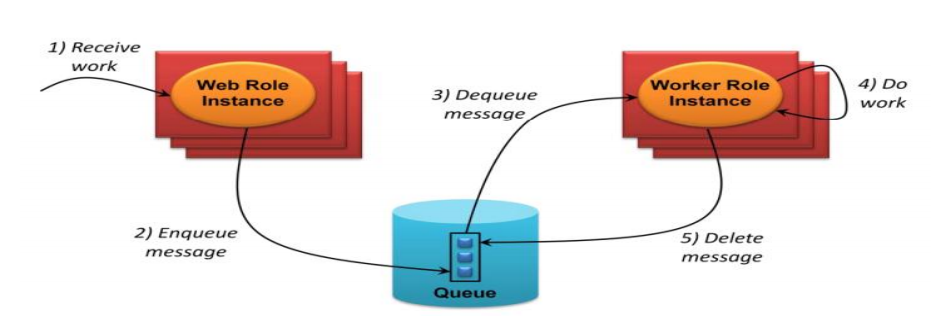
Sau đây là một số hướng dẫn cơ bản và các đề xuất cho cách chọn một PartitionKey để bảng của bạn cho hiệu quả truy vấn cao:

1. Đầu tiên xác định các thuộc tính quan trọng trong bảng của bạn. Đây là những thuộc tính thường xuyên sử dụng như bộ lọc truy vấn.
2. Lựa chọn những khóa có khả năng từ những thuộc tính quan trọng này.
   1. Điều quan trọng là xác định các truy vấn chi phối đến khối lượng công việc của ứng dụng. Từ những truy vấn chi phối này, lựa chọn các thuộc tính được sử dụng trong bộ lọc truy vấn.
   2. Đây là tập hợp các thuộc tính khóa ban đầu của bạn.
   3. Sắp xếp các thuộc tính khóa theo tầm quan trọng trong truy vấn của bạn.
3. Các thuộc tính khóa này xác định độc nhất các thực thể chưa ? Nếu chưa, thêm định danh duy nhất (ID) vào tập hợp các khóa.
4. Nếu bạn chỉ có một thuộc tính khóa, sử dụng nó làm PartitionKey.
5. Nếu bạn chỉ có 2 thuộc tính khóa, sử dụng thuộc tính đầu tiên làm ParitionKey, và thuộc tính thứ 2 làm RowKey.
6. Nếu bạn có hơn 2 thuộc tính khóa, bạn có thể ghép chúng thành hai nhóm – nhóm ghép đầu tiên là PartitionKey, và nhóm thứ hai là RowKey. Với PartitionKey có thể chứa hai khóa cách nhau bởi dấu “-“.

#### 2.4.3.3. Queue

Windows Azure Queue cung cấp một cơ chế phân phối message đáng tin cậy. Nó cung cấp cơ chế điều phối không đồng bộ, được sử dụng để kết nối các thành phần khác nhau của ứng dụng đám mây. Windows Azure Queue có tính sẵn sàng cao, bền và hiệu quả thực thi cao. Lập trình ngữ nghĩa của nó đảm bảo rằng một message có thể được xử lí ít nhất một lần.

Windows Azure Queue cho phép tách các thành phần khác nhau của ứng dụng đám mây, cho phép ứng dụng đám mây được xây dựng một cách dễ dàng với nhiều công nghệ khác nhau và dễ dàng mở rộng theo nhu cầu lưu thông.



**Hình 2.19 – Các thao tác với Queue**

Trong kịch bản điển hình, nhiều thể hiện Web role đang chạy, mỗi thể hiện

nhận công việc từ người dùng (bước 1). Để đẩy công việc đến thể hiện Worker role,

một thể hiện Web role viết một message vào queue(bước 2). Message này, chứa tối

đa 8KB, có thể chứa đường dẫn URI đến một blob hoặc một entity trong table hoặc

những nội dung khác. Thể hiện Worker role đọc message từ queue này(bước 3). Sau đó thực hiện công việc mà queue yêu cầu. Khi thể hiện Worker role đã hoàn thành công việc message yêu cầu, nó phải xóa hẳn message từ queue (bước 5).

**Mô hình dữ liệu Queue :**

Storage Account (Tài khoản lưu trữ).

* Một tài khoản có nhiều queue. Queue (Hàng đợi) – Một queue chứa nhiều message (thông báo).
* Không có giới hạn về số lượng các message được lưu trữ trong queue.
* Một message được lưu trữ nhiều nhất một tuần. Hệ thống sẽ dọn các message đã lưu trữ nhiều hơn một 1 tuần.
* Queue có metadata. Metadata là cặp <tên, giá trị> và chúng có kích thước lên đến 8KB với mỗi queue.

Message (Thông báo) – Message được lưu trữ trong queue. Mỗi message có kích thước lên đến 8KB. Để lưu trữ dữ liệu lớn hơn, chúng ta lưu trữ nó vào bộ lưu trữ Blob hoặc Table, và sau đó lưu trữ tên blob/entity trong message Định nghĩa một vài tham số:

* MessageID: một giá trị GUID để nhận biết message trong queue.
* VisibilityTimeout: một giá trị nguyên định rõ visibility timeout của message. Giá trị mặc định là 30 giây. Giá trị tối đa là 2 giờ.
* PopReceipt: một chuỗi, chuỗi này cùng với MessageID được yêu cầu để xóa một message từ queue.
* MessageTTL: Xác định khoảng thời gian sống cho message. Thời

gian sống tối đa là 7 ngày. Nếu một message không được xóa trong thời gian sống,

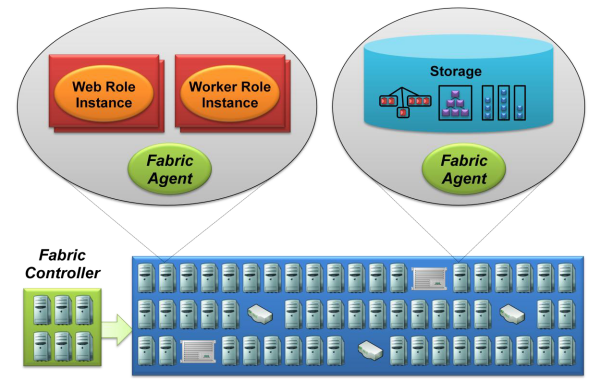
thì sau thời gian sống nó sẽ được dọn và xóa bởi hệ thống lưu trữ.

URI cho một queue cụ thể có cấu trúc như sau:

http://<account>.queue.core.windows.net/<QueueName>

### 2.4.4. Fabric

Tất cả các ứng dụng Windows Azure, và dữ liệu của nó đều tồn tại trên trung tâm dữ liệu của Microsoft. Bên trong trung tâm dữ liệu này, một tập hợp các máy dành cho Windows Azure được tổ chức thành một kết cấu (*fabric*).



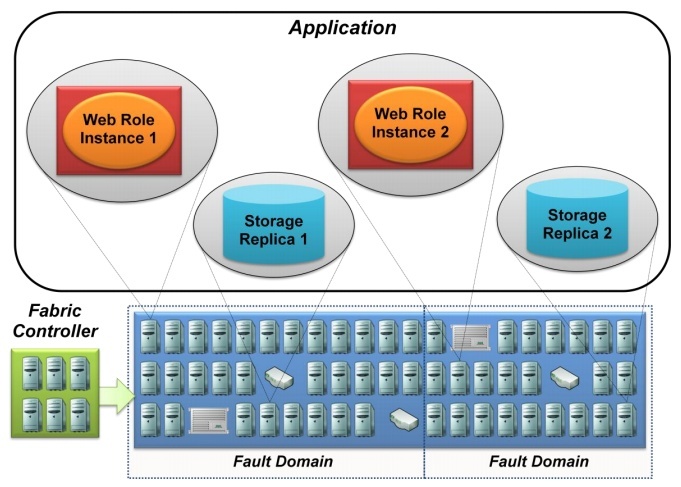
Hình 2.20 - Fabric Controller

Như hình trên, Windows Azure Fabric chứa một một nhóm các máy, tất cả chúng được quản lý bởi một phần mềm gọi là *fabric controller*. Fabric controller được tái tạo qua mỗi nhóm từ 5 đến 7 máy, nó là sở hữu tất cả các tài nguyên: máy tính, switches, bộ cân bằng tải (load balancer),v.v… Bởi vì nó có thể giao tiếp với một *fabric agent* trên mỗi máy tính, nó cũng nhận biết được tất cả các ứng dụng Windows Azure trong kết cấu.

Với các thông tin có được, cho phép *fabric controller* có thể làm được nhiều việc rất hữu ích. Nó theo dõi tất cả các ứng dụng đang chạy. Nó quản lý hệ điều hành, quản lý các việc như vá lỗi cho phiên bản của Windows Service 2008. Nó quyết định khi một ứng dụng mới được upload lên, thì sẽ được chạy trên dịch vụ nào. Để làm được điều này, fabric controller phụ thuộc vào tập tin cấu hình của mỗi ứng dụng được upload lên, trong đó, chỉ ra bao nhiêu thể hiện cần được tạo ra, và kích thước các máy ảo là thế nào. Dựa vào đó, fabric controller tạo ra các máy ảo tương ứng. Khi tạo ra các máy ảo này, fabric controller sẽ theo dõi các ứng dụng đó, Nếu một, ứng dụng, cần có 5 thể hiện, và một trong số đó bị “chết”, fabric controller sẽ tự động khởi tạo một thể hiện mới. Nếu một máy ảo đang chạy bị chết, fabric controller sẽ tự động khởi tạo một thể hiện khác của ứng dụng trên một máy ảo khác, sau đó khởi động lại bộ cân bằng tải nếu cần thiết để chỉ đến máy mới này.

Như đã nói, fabric controller chịu trách nhiệm gán cho các thể hiện của ứng dụng vào một máy vật lý cụ thể. Điều này, rất quan trọng, việc gán thế nào, sẻ ảnh hướng rất nhiều đến tính sẵn sàng phục vụ của ứng dụng. Ví dụ, một ứng dụng yêu cầu 5 thể hiện Web role, 2 thể hiện Worker role, giả sử fabric controller gán các thể hiện này vào một mạng dùng chung một switch, vậy điều gì xẩy ra, nếu như switch này không hoạt động được, ứng dụng sẽ không còn hoạt động được nữa. Đích hướng dến của Windows Azure là tính sẵn sàng cao, do vây, việc để cho ứng dụng phụ thuộc vào một điểm thất bại là không chấp nhận được.

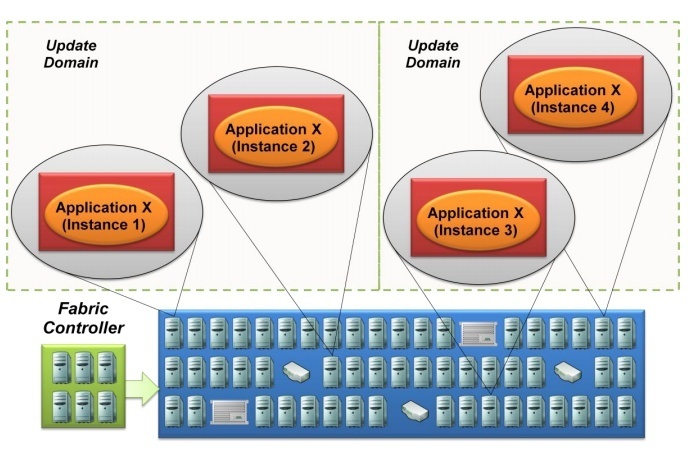
Để khắc phục điều này, fabric controller nhóm các máy nó làm chủ thành một số các miền gọi là *fault domains*. Mỗi miền là một phần của trung tâm dữ liệu, Ví dụ:



**Hình 2.21 -**

Trong hình trên, giả sử ứng dụng cần 2 thể hiện Web role, và trung tâm dữ liệu được chia làm 2 miền lỗi. Khi fabric controller triển khai ứng dụng này, nó sẽ đặt mỗi thể hiện Web role vào một miền, khi đó, khi có lỗi xẩy ra ở một miền nào đó, cũng sẽ không thể làm chết hoàn toàn ứng dụng của bạn.

Điều này vẫn chưa đủ, điều gì xảy ra nếu ứng dụng có nhu cầu cập nhật, việc tắt ứng dụng và khở động lại là không nên. Để tránh điều này, fabric controller nhóm các thể hiện của ứng dụng thành các miền cập nhât “*update domain*”.

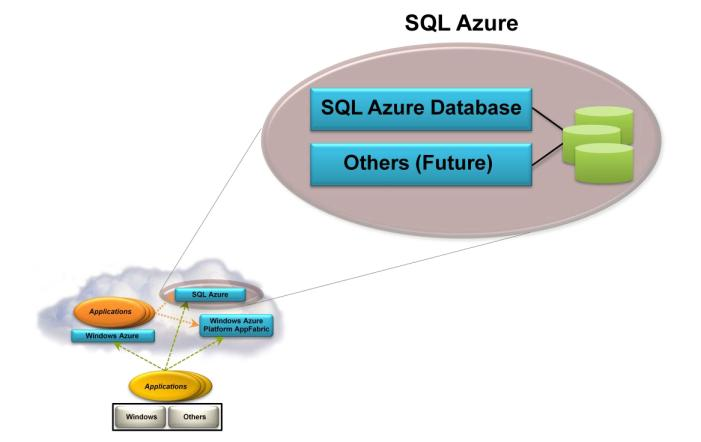


**Hình 2.22 -**

Khi mã của ứng dụng cần cập nhât, fabric controller sẽ thực hiện cập nhật trong từng miền. Ví dụ trong hình trên, fabric controller sẽ tiến hành tắt 2 thể hiện 1 và 2 của ứng dụng trước, cập nhật,và khởi động lại. Sau đó, tiến hành tương tự cho 2 thể hiện ở miền còn lại. Mục đích của việc này là làm cho ứng dụng không bi gián đoạn, khi một ứng dụng đang cập nhật, người dùng vẫn có thể truy cập đến với phiên bản cũ của ứng dụng, cho dến khi nó cập nhật xong.

### 2.4.5. SQL Azure

Mục tiêu của SQL Azure cung cấp c ác dịch vụ dựa trên đ ám mây để lưu trữ và xử lí dữ liệu. Trong khi đó Microsoft nói rằng SQL Azure sẽ bao gồm một loạt các tính năng định hướng dữ liệu, đồng bộ hó a dữ liệu, báo cáo, phân tích dữ liệu và những c hức năng khác.



**Hình 2.22 - SQL Azure cung cấp dịch vụ định hướng dữ liệu trong đám mây**

Cơ sở dữ liệu SQL Azure cung cấp một hệ thống quản lí cơ sở dữ liệu dựa trên đám mây. Công nghệ này cho phép ứng dụng on-premise và ứng dụng đám mây lưu trữ dữ liệu quan hệ và những kiểu dữ liệu khác trên các máy chủ trong trung tâm dữ liệu của Microsoft. Cũng như các công nghệ đám mây khác và người dùng chỉ trả cho những gì họ sử dụng.

Cơ sở dữ liệu SQL Azure được xây dựng dựa trên Microsoft SQL Sever. Công nghệ này cung cấp môi trường SQL Server trong đám mây và bổ sung index.

#### 2.4.4.1.Tổng quan kiến trúc SQL Azure

**Tài khoản Wi ndows Azure Pl atform**: để sử dụng SQL Azure, bạn phải tạo một tài khoản Windows Azure. Sử dụng tài khoản này, bạn có thể truy xuất tất cả chức năng của nền tảng Wi ndows Azure. Tài khoản này được sử dụng để thanh toán việc sử dụng các dịch vụ Windows Azure.

**Máy chủ Wi ndows Azure Pl atform**: mỗi tài khoản Windows Azure có thể chứa nhiều máy chủ SQL Azure. Mỗi máy chủ bao gồm hệ thống đăng nhập và bạn cũng có thể chỉ định vị trí địa lý mà máy chủ của bạn được đặt ở cấp này.

Bạn sử dụng cổng SQL Azure để tạo và quản lí máy chủ cơ sở dữ liệu của bạn. Cổng này cung cấp giao diện dễ dàng sử dụng để bạn có thể tạo đăng nhập và cung cấp cơ sở dữ liệu.

**Cơ s ở dữ li ệu SQL Azure**: mỗi máy chủ Cơ sở dữ liệu SQL Azure có thể chứa nhiều cơ sở dữ liệu. Một máy chủ cơ sở dữ liệu mới có một cơ sở dữ liệu chính. Trong mỗi cơ sở dữ liệu, bạn có thể tạo table, view, stored procedure và các đối tượng cơ sở dữ liệu quen thuộc khác. Bạn có thể sử dụng cổng SQL Azure để tạo cơ sở dữ liệu.

Cơ sở dữ liệu SQL Azure được nhân rộng các phân vùng dữ liệu trên nhiều máy tính vật lý tại một trung tâm dữ liệu SQL Azure. Kiến trúc này cung cấp chuyển đổi dự phò ng tự động và cân bằng tải. Dữ liệu khách hàng được lan truyền qua nhiều máy chủ vật lý trong phạm vi địa điểm địa lý được thiết lập. Bằng cách này, Cơ sở dữ liệu SQL Azure đạt được độ sẵn sàng cao và ổn định cho tất cả các ứng dụng từ nhỏ nhất đến lớn nhất mà không cần nỗ lực quản lí chuyên sâu.

**Mô hình dữ liệu quan hệ:** mục tiêu thiết kế chính cho SQL Azure là cung cấp một môi trường thân thuộc cho lập trình viên cơ sở dữ liệu. Do đó, các đối tượng được tạo ra trong Cơ sở dữ liệu SQL Azure giống như các đối tượng hiện có trong cơ sở dữ liệu SQL Ser ver. Như: Table, Index, View, Stored Procedure, Trigger v.v…

Cả SQL Server và SQL Azure sử dụng ngôn ngữ TSQL để tạo cơ sở dữ liệu và thao tác dữ liệu. Do đó, các lập trình viên cơ sở dữ liệu và người quản trị có thể sử dụng các kỹ năng chuyên môn hiện có của họ cho SQL Azure.

**Kiến trúc truy xuất dữ liệu:** cơ sở dữ liệu SQL Azure expose một endpoint Tabular Data Stream ( TDS) cho cơ sở dữ liệu trên đám mây. Một ứng dụng desktop của client có thể kết nối vào Cơ sở dữ liệu SQL Azure tương tự như cách kết nối một thể hiện SQL Server on- premise. Truy vấn được sử dụng bằng ngôn ngữ TSQL. Secure Sockets Layer(SSL) được yêu cầu khi một ứng dụng client kết nối vào endpoint TDS Cơ sở dữ liệu SQL Azure nhằm đảm bảo bảo mật.

Trong kịch bản ứng dụng desktop và Cơ sở dữ liệu SQL Azure, bạn phải xem xét độ trễ xảy ra trên đám mây và trong xử lý của client. Độ trễ này chắc chắn cao hơn so với cơ sở dữ liệu được đ ặt trên cơ sở của bạn. Một cách để tránh độ trễ này là tạo ra một giao diện người dùng trên nền Web cho các ứng dụng cơ sở dữ liệu của bạn và máy chủ lưu trữ nó trong Wi ndows Azure. Trong kịch bản này, mã nguồn client và các dữ liệu được lưu trữ tại trung tâm cùng với cơ sở dữ liệu, do đó độ trễ thấp.

Kiến trúc thứ ba mà SQL Azure hỗ trợ, bạn có thể tạo một ứng dụng bằng cách sử dụng ADO.Net và Entity Frame work, chứa nó trên Windows Azure. Sau đó bạn có thể sử dụng ADO.NET Data Services để công bố ứng dụng này như một dịchvụ, sử dụng SOAP, REST hoặc giao tiếp JSON, và xây dựng ứng dụng client nhẹ để sử dụng dữ liệu từ dịch vụ.

**Mô hình bảo mật:** nhiều cơ sở dữ liệu chứa các thông tin nhạy cảm, vì thế nó cần phải kiểm soát truy cập cẩn thận. Trong SQL Azure, bạn có thể sử dụng các cách bảo mật sau để chứng thực truy xuất và bảo vệ dữ liệu của bạn:

- SQL Server Login: được dùng để chứng thực truy xuất vào SQL Azure ở cấp độ máy chủ.

- Database User: được dùng để cấp quyền truy cập vào SQL Azure tại cấp độ cơ sở dữ liệu.

- Database Role: được dùng cho nhóm người dùng , để cấp quyền truy xuất vào SQL Azure tại cấp độ cơ sở dữ liệu.

**Triển khai:** có thể tạo và đặt một cơ sở dữ liệu ho àn to àn bằng cơ sở dữ liệu SQL Azure bằng các h sử dụng Trans act-SQL. Tuy nhiên, trong hầu hết trường hợp, các lập trình viê n hoặc quản trị viên sẽ thiết kế và đặt cơ sở dữ liệu trên máy tính lập trình viê n hoặc máy chủ on-premise. Khi ứng dụng được hoàn thành, cơ sở dữ liệu sẽ được triển khai đến các đ ám mây.

Để triển khai một cơ sở dữ liệu trên SQL Azure, bạn có thể tạo một script TSQL CREATE DATABASE trong Microsoft SQL Server® Management Studio với Generate Script Wizard. Sau đó bạn có thể chạy TSQL trong SQL Azure để tạo cơ sở dữ liệu.

#### 2.4.4.2. Ứng dụng của SQL Azure

Một ứng dụng Windows Azure có thể lưu trữ dữ liệu trong Cơ sở dữ liệu SQL Azure. Trong khi bộ lưu trữ Windows Azure không hỗ trợ các bảng dữ liệu quan hệ, mà nhiều ứng dụng đang tồn tại sử dụng cơ sở dữ liệu quan hệ. Vì vậy, lập trình viê n có thể chuyển ứng dụng đang chạy sang ứng dụng Wi ndows Azure với lưu trữ dữ liệu trong Cơ sở dữ liệu SQL Azure.

Xây dựng một ứng dụng của doanh nghiệp nhỏ hoặc của các phòng ban trong công ty lớn có thể lưu trữ dữ liệu trong Cơ sở dữ liệu SQL Azure.

Giả sử một nhà sản xuất muốn thông ti n sản phẩm có sẵn trên cả mạng lưới đại lý và khách hàng. Đưa dữ liệu này vào Cơ sở dữ liệu SQL Azure để cho nó được truy cập bởi các ứng dụng đang chạy tại các đại lý và ứng dụng Web của khách hàng.

### 2.4.5. Các kịch bản sử dụng Windows Azure

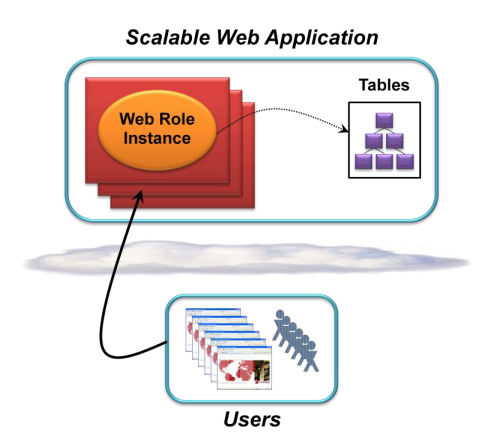
Chúng ta sẽ đi qua 5 kịch bản chính sử dụng Windows Azure:

* Tạo một ứng dụng web có khả năng mở rộng.
* Tạo một ứng dụng xử lí song song.
* Tạo một ứng dụng web với xử lí nền.
* Tạo một ứng dụng web với dữ liệu quan hệ
* Sử dụng lưu trữ đám mây từ ứng dụng on-premise hoặc hosted.

##### Tạo môt ứng dụng web có khả năng mở rộng

Ví dụ, nếu ứng dụng có nhu cầu xử lí nhiều người dùng truy xuất đồng thời, thì cần phải xây dựng nó trên một nền tảng hỗ trợ mở rộng ứng dụng và mở rộng dữ liệu. Hoặc ứng dụng có tải thay đổi đáng kể, thỉnh thoảng có nhu cầu sử dụng tải cực cao. Ví dụ như: bán vé online hoặc xem các tin tức nóng hổi có thể thể hiện mô hình này. Chạy ứng dụng loại này trong một trung tâm dữ liệu thông thường đòi hỏi phải luôn có sẵn đủ máy để xử lý các tải ở đỉnh cao, mặc dù toàn bộ hệ thống không sử dụng hầu hết thời gian. Nếu ứng dụng được xây dựng trên Windows Azure, tổ chức chạy nó có thể mở rộng nhiều thể hiện để sử dụng khi cần thiết, sau đó co lại với số thể hiện ít hơn. Bởi vì Windows Azure tính phí theo nhu cầu sử dụng – điều này sẽ rẻ hơn việc duy trì nhiều máy không sử dụng tới.

Để tạo một ứng dụng web có thể mở rộng trên Windows Azure, lập trình viên có thể sử dụng Web role và table.

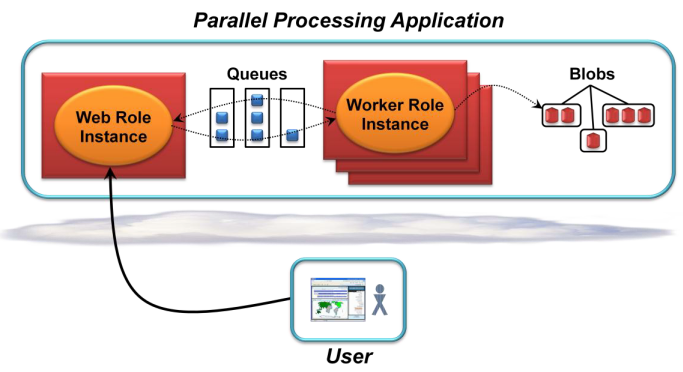


**Hình 2.23 - Ứng dụng Web mở rộng sử dụng Web Role và table**

Trong trường hợp này, lập trình viên xác định số thể hiện mà ứng dụng nên chạy, và Windows Azure fabric controller tạo ra số máy ảo tương ứng. Như đã nói trước đó, fabric controller cũng theo dõi các thể hiện, và đảm bảo rằng số lượng thể hiện yêu cầu là luôn luôn có sẵn. Để lưu trữ dữ liệu, ứng dụng sử dụng Windows Azure Table, nó cung cấp khả năng lưu trữ mở rộng để xử lý một lượng lớn dữ liệu.

##### Tạo môt ứng dụng xử lí song song

Lập trình viên có thể sử dụng nhiều Worker role để tạo ra ứng dụng loại này. Và nó không chỉ là lựa chọn duy nhất, xử lí song song có thể sử dụng một tập lớn dữ liệu,nó có thể được lưu trữ trong Window Azure Blob.



Hình 2.24 - Ứng dụng xử lí song song dùng 1 Web role, nhiều Work role , blob và queue

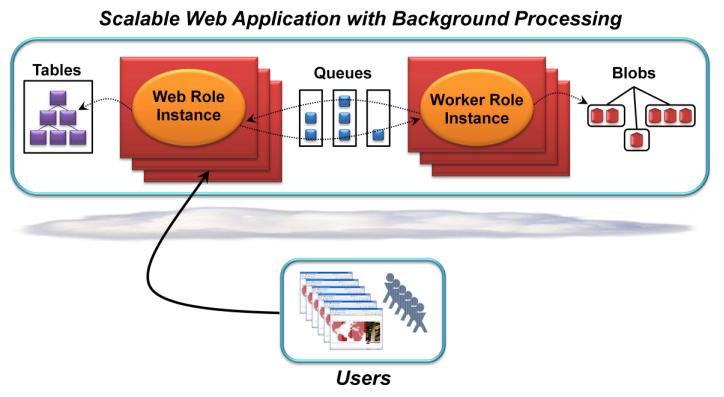
Trong chiến lược ở đây, công việc song song được thực hiện bằng nhiều thể hiện Worker role chạy đồng thời, mỗi thể hiện sử dụng dữ liệu blob. Để tương tác với ứng dụng người dùng dựa vào một thể hiện Web role. Qua giao tiếp này, người dùng có thể quyết định bao nhiêu thể hiện Worker role nên chạy, bắt đầu và dừng lại các thể hiện, lấy kết quả v.v…Liên lạc giữa thể hiện Web role và Worker role dựa trên Windows Azure Storage queue.

##### Tạo một ứng dụng Web mở rộng với xử lí nền

Trong nhiều tình huống phần mềm truy xuất trình duyệt cũng cần thiết lập các công việc chạy nền độc lập với request/ response của ứng dụng.

Ví dụ, ứng dụng web chia sẻ video. Nó cần phải chấp nhận request từ số lượng lớn người dùng đồng thời. Một số request sẽ tải lên video mới, mỗi video phải được xử lý và lưu trữ cho truy cập sau đó. Làm cho người dùng chờ đợi trong khi xử lý này đang thực hiện sẽ làm mất nhiều thời gian. Thay vì vậy, một phần của ứng dụng chấp nhận request có thể thiết lập một công việc nền để thực hiện công việc này.

Nhiều Windows Azure Web role và Worker role được sử dụng cùng nhau trên kịch bản này.

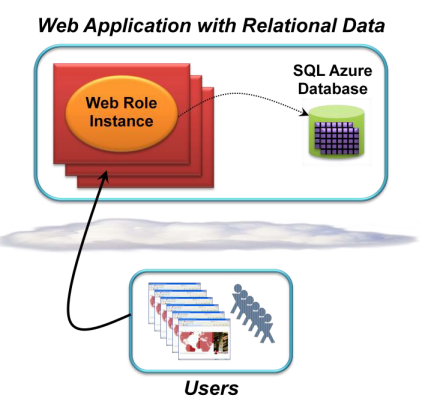


Hình 2.25 - Một ứng dụng Web mở rộng với xử lí nền

Ứng dụng xử dụng nhiều thể hiện Web role để xử lí các request từ người dùng, Để hỗ trợ số lượng lớn các người dùng đồng thời, table được sử dụng để lưu trữ thông tin. Xử lí nền, nó dựa vào các thể hiện Worker role, chuyển các công việc qua queue.

##### Tạo một ứng dụng Web với dữ liệu quan hệ

Blob, table, và queue phù hợp với một số tình huống. Trong những tình huống khác, dữ liệu quan hệ phù hợp hơn. Giả sử một doanh nghiệp muốn chạy một ứng dụng trên Windows Azure. Ứng dụng này không cần thiết phải mở rộng dữ liệu hay thay đổi thành phần dữ liệu như Windows Azure Table hỗ trợ. Thay vào đó, lập trình viên thích sử dụng dữ liệu quan hệ hơn. Trong trường hợp này, ứng dụng có thể sử dụng Windows Azure cùng với cơ sở dữ liệu SQL Azure.

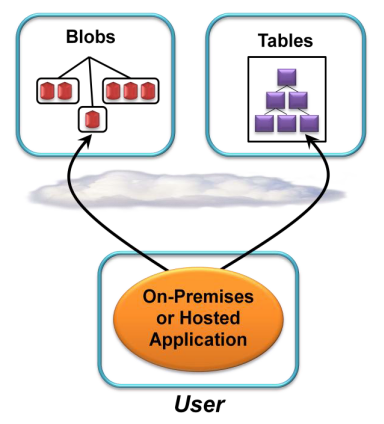


Hình 2.26 - Ứng dụng Windows Azure sử dụng cơ sở dữ liệu SQL Azure.

Cơ sở dữ liệu SQL Azure cung cấp một tập lớn các chức năng SQL Server khi quản lí dịch vụ đám mây. Ứng dụng có thể tạo cơ sở dữ liệu, chạy truy vấn SQL,và hơn thế nữa, nhưng không cần quản lí hệ thống cơ sở dữ liệu hoặc phần cứng – Microsoft sẽ lo việc này. Bởi vì cơ sở dữ liệu SQL Azure là dịch vụ đám mây, việc tính phí dựa trên lưu lượng sử dụng.

##### Sử dụng lưu trữ đám mây từ ứng dụng on-premise hoặc hosted.

Trong khi Windows Azure cung cấp nhiều khả năng, một ứng dụng đôi khi chỉ cần sử dụng một trong số đó. Ví dụ, hãy nghĩ về một ứng dụng on-premises và hosted có nhu cầu để lưu trữ lượng dữ liệu rất lớn. Một trang web tin tức chạy ở một hoster có thể cần một nơi có khả năng mở rộng và truy xuất trên toàn cầu để lưu trữ số lượng lớn văn bản, đồ họa, video, và thông tin hồ sơ về người sử dụng.



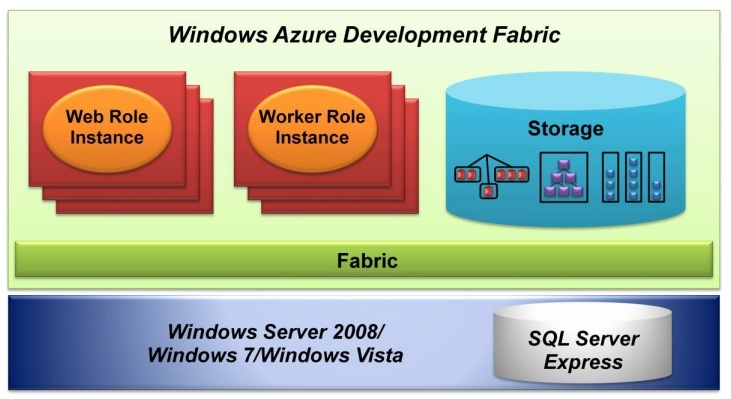
**Hình 2.27 - Một ứng dụng on-premises hoặc hosted dùng Windows Azure blob và table.**

Như hình trên, một ứng dụng on-premise hoặc hosted có thể truy xuất trực tiếp bộ lưu trữ Windows Azure. Với truy cập này có thể sẽ chậm hơn so với truy cập lưu trữ cục bộ, nhưng nó cũng có thể sẽ rẻ hơn, và có khả năng mở rộng lớn hơn. Đối với một số ứng dụng cần phải cân nhắc vấn đề này. Ngoài ra, ứng dụng có thể sử dụng cơ sở dữ liệu SQL Azure.

##### Phát triển ứng dụng Windows Azure

Đối với các lập trình viên, xây dựng ứng dụng Windows Azure cũng giống như xây dựng một ứng dụng Windows truyền thống. Nền tảng Windows Azure hỗ trợ cả ứng dụng .Net và ứng dụng xây dựng với native code. Windows Azure cung cấp các mẫu project trong Visual Studio để tạo Web role, Worker role, và kết hợp cả hai.

Một vấn đề là ứng dụng Windows Azure không chạy cục bộ mà chạy trên đám mây. Vì vậy, để giải quyết vấn đề này, Microsoft cung cấp development fabric, một phiên bản của môi trường Windows Azure để chạy trên máy của lập trình viên.



Hình 2.28. Development fabric cung cấp một bản sao của Windows Azure cho lập trình viên.

Development fabric chạy trên máy đơn Windows Server 2008, Windows 7, hoặc Windows Vista. Nó mô phỏng các chức năng của Windows Azure trên đám mây, với Web role, Worker role, và bộ lưu trữ Windows Azure. Lập trình viên có thể xây dựng một ứng dụng Windows Azure triển khai nó trên development fabric, và chạy nó giống như chạy ứng dụng trên đám mây. Lập trình viên có thể quyết định xem bao nhiêu thể hiện của mỗi role nên chạy và có thể dùng queue để liên lạc giữa các thể hiện với nhau. Một ứng dụng có thể được phát triển và kiểm thử ở cục bộ, lập trình viên có thể upload code và tập tin cấu hình của nó qua cổng Windows Azure, sau đó chạy nó.

## 2.5. Ứng dụng trên điện toán đám mây

Ứng dụng sử dụng điện toán đám mây dễ nhận ra nhất chính là sản phẩm của Google mà chúng ta sử dụng hằng ngày: ChromeBook (www.google.com/chromebook). Chúng ta chỉ cần đăng nhập thông tin của mình bằng gmail ở bất cứ đâu, sau một khoảng thời gian thiết lập, chrome sẽ nhận ra các thói quen sử dụng trình duyệt của chủ nhân (hình nền, các bookmark, v.v... ).

Bên cạnh email thì mạng xã hội cũng là một lĩnh vực mà điện toán đám mây được ứng dụng khá rộng rãi mà dễ thấy nhất là Facebook ([www.facebook.com](http://www.facebook.com)) và MySpace (www.myspace.com). Ý tưởng chính của mạng xã hội là để tìm người bạn đã biết hoặc những người mà chúng ta sẽ muốn biết và chia sẻ thông tin của chúng ta với họ. Tất nhiên, khi chúng ta chia sẻ thông tin của chúng ta với những người này, chúng ta cũng chia sẻ nó với những người chạy dịch vụ.

Những sản phẩm của [Google Docs](http://translate.googleusercontent.com/translate_c?hl=vi&ie=UTF8&prev=_t&rurl=translate.google.com.vn&sl=en&tl=vi&twu=1&u=http://www.google.com/google-d-s/intl/en/tour1.html&usg=ALkJrhhTuMV2aE--cMYnDVMqChpNIFgyWw) (https://docs.google.com), cũng như một số dịch vụ [Zoho Office](http://translate.googleusercontent.com/translate_c?hl=vi&ie=UTF8&prev=_t&rurl=translate.google.com.vn&sl=en&tl=vi&twu=1&u=http://www.zoho.com/&usg=ALkJrhg8FnV6Vary8OQICe23qj1ApwDPHg) (www.zoho.com) tồn tại trên Internet, nó cho phép chúng ta giữ và chỉnh sửa tài liệu của chúng ta trực tuyến. Bằng cách đó, các tài liệu sẽ có thể truy cập bất cứ nơi nào và chúng ta có thể chia sẻ tài liệu và tương tác với chúng. Nhiều người có thể làm việc trong cùng một tài liệu cùng một lúc.

Google và Microsoft cung cấp một phương tiện mà người tiêu dùng có thể tạo ra một hồ sơ sức khỏe trực tuyến cá nhân (Personal health record). Google Health (www.google.com/health) và Microsoft HealthVault ([www.microsoft.com/en-us/healthvault](http://www.microsoft.com/en-us/healthvault)) cho phép người dùng tạo ra, lưu trữ và truy cập hồ sơ sức khỏe trực tuyến cá nhân trên website tìm kiếm cả mình.

Ngay cả khi bạn sử dụng dịch vụ để giữ tất cả các tài liệu và hình ảnh, rất có thể là bạn vẫn còn có dữ liệu trên máy tính cá nhân của bạn. Một trong những vấn đề lớn nhất với máy tính cá nhân có xu hướng bị mất dữ liệu đó nếu máy tính của bạn bị đánh cắp, bị phá hủy, hoặc thiết bị lưu trữ bị hư hỏng.

Các dịch vụ miễn phí như [Syncplicity](http://translate.googleusercontent.com/translate_c?hl=vi&ie=UTF8&prev=_t&rurl=translate.google.com.vn&sl=en&tl=vi&twu=1&u=http://www.syncplicity.com/&usg=ALkJrhi4cuOe_GbWs_Hfi9nz9LuYRpXWVg) và [Dropbox](http://translate.googleusercontent.com/translate_c?hl=vi&ie=UTF8&prev=_t&rurl=translate.google.com.vn&sl=en&tl=vi&twu=1&u=http://www.dropbox.com/&usg=ALkJrhimDPI1_aVwpsPmNumamp-l74E1vg) ([www.dropbox.com](http://www.dropbox.com)) làm cho nó dễ dàng để giữ các bản sao của các tập tin trên nhiều máy tính đồng bộ trong khi vẫn giữ một bản sao trong đám mây. Một số các dịch vụ này thậm chí sẽ tiếp tục các phiên bản trước của tập tin hoặc file bị xóa trong trường hợp bạn xóa một file quan trọng.

([www.syncplicity.com](http://www.syncplicity.com))

## 2.6. Xu hướng phát triển của điện toán đám mây

Dưới đây là những nhận định về xu hướng phát triển của điện toán đám mây trong những năm tới ở thế giới nói chung và Việt Nam nói riêng mà VnExpress thu thập được :

**Sự phát triển của đám mây di động (mobile cloud)**:Apple iCloud, Amazon Cloud và dịch cụ điện toán đám mây trên Windows Phone đang giúp công nghệ này trở nên đại trà. Nhu cầu lưu trữ thông tin trên đám mây và khả năng truy cập bất cứ khi nào họ cần sẽ giảm bớt "gánh nặng" cho thiết bị. Nỗi lo mất điện thoại vì "mọi dữ liệu quan trọng như số liên lạc, ảnh, video,v.v… nằm cả trong đó" sẽ không còn bởi thông tin đã được tự động sao lưu lên đám mây và người sử dụng có thể thoải mái xóa dữ liệu từ xa để tránh tình trạng dữ liệu bí mật, riêng tư rơi vào tay kẻ xấu.

**Sự nở rộ của đám mây lai (hybrid cloud):**Đám mây lai là sự giao thoa của hai hay nhiều mô hình đám mây, như như kết hợp giữa public cloud (các dịch vụ cloud được cung cấp cho mọi người sử dụng rộng rãi và private cloud (cơ sở hạ tầng và các dịch vụ được xây dựng để phục vụ cho một tổ chức, doanh nghiệp duy nhất). Điều này sẽ giúp khai thác những điểm mạnh nhất của từng mô hình, mang đến khả năng bảo vệ dữ liệu an toàn hơn, nhưng cũng linh động và gần gũi hơn với người sử dụng. Năm tới sẽ có ngày càng nhiều doanh nghiệp vừa và lớn sẽ chuyển sang mô hình này. Theo Gartner, tổng giá trị cho các dịch vụ đám mây hiện thời là gần 2,4 tỷ USD và đến năm 2013 sẽ đạt gần 8,1 tỷ USD. ( http://vnexpress.net/gl/vi-tinh/2011/11/cac-xu-huong-dien-toan-dam-may-nam-2012/)

**Sự tiến hóa của bảo mật đám mây:**Bảo mật luôn là đề tài nóng và là một trong những nguyên nhân chính khiến các tổ chức có liên quan đến các dữ liệu nhạy cảm lưỡng lự trong việc đón nhận. Họ lo ngại hacker tìm cách xâm nhập vào kho thông tin nằm trên đám mây, do đó việc liên tục tạo ra những phương pháp bảo mật kiểu mới, an toàn và hiệu quả là mục tiêu hàng đầu của các chuyên gia phát triển trong năm 2012.

**Cuộc cách mạng môi trường làm việc di động:**Tương tự e-mail thay đổi cách con người liên lạc với bạn bè và đồng nghiệp, cloud được cho là đang tạo ra con đường gửi và lưu trưc thông tin nhanh chóng và thông suốt hơn bao giờ hết. Với khả năng truy cập và làm việc từ xa, công việc sẽ được giải quyết dễ dàng hơn mà không bị ngắt quãng.

**Dịch vụ phần mềm (SaaS) mở rộng thành dịch vụ IT (ITaaS):**SaaS sẽ không chỉ giới hạn trong lĩnh vực CRM mà dần có tầm ảnh hưởng đến cả cơ sở hạ tầng IT. "Một lĩnh vực mới đang nổi lên trên thị trường là IT as a Service (dịch vụ IT), trong đó các doanh nghiệp sẽ 'tiêu thụ' IT, biến nó trở thành một dịch vụ trong doanh nghiệp. Bạn có thể hình dung một thế giới mà ở đó việc triển khai các ứng dụng trên toàn cầu chỉ mất 2 tiếng thay vì 2 tháng, các chuyên gia phát triển sẽ sử dụng một nền tảng tự phục vụ (a self-service platform) để cung cấp và triển khai ứng dụng thay vì phải thông qua một quá trình thủ công tốn kém nào đó, hoặc một người có thể quản lý 10.000 server thay vì chỉ 100 server", ông Tyson Dowd, Giám đốc chiến lược thương mại của Microsoft thuộc khu vực châu Á - Thái Bình Dương, nhấn mạnh.

# CHƯƠNG 3. SỔ LIÊN LẠC TRỰC TUYẾN

## 3.1. Giới thiệu tổng quan

Là cổng kết nối thông tin giữa nhà trường và phụ huynh, hệ thống sổ liên lạc trực tuyến mang những đặc điểm của một cuốn sổ liên lạc truyền thống: thông tin điểm học, nhận xét của giáo viên, thông báo của nhà trường, v.v... Nhưng sổ liên lạc truyền thống có còn hiệu quả không khi chỉ liên lạc hàng tuần, hàng tháng, thậm chí không đến được tay của phụ huynh. Với việc ứng dụng công nghệ thông tin, sổ liên lạc trực tuyến cho phép thông tin đến phụ huynh về tình hình học tập của con em nhanh chóng, chính xác và đầy đủ hơn.

Hệ thống sổ liên lạc trực tuyến được nhóm xây dựng trên mô hình là kênh thông tin giữa nhà trường và phụ huynh qua internet bằng ứng dụng web và email. Qua kênh này, phụ huynh nhanh chóng nhận được:

* Thông tin về lớp học
* Thông tin về điểm của con em mình
* Nhận xét của giáo viên
* Các thông báo của nhà trường như họp phụ huynh, nghỉ lễ, tết, v.v…

Đồng thời, phụ huynh cũng có thể gửi ý kiến phản hồi tới nhà trường, giáo viên.

Với việc áp dụng hệ thống sổ liên lạc trực tuyến này sẽ đem lại nhiều lợi ích cho nhà trường và phụ huynh học sinh như:

* Đối với phụ huynh học sinh:
  + Dễ dàng nắm bắt thông tin nhanh chóng về tình hình học tập hàng ngày của con em cũng như các nhận xét, thông báo, mời họp từ phía nhà trường, kế hoạch của lớp.
* Đối với nhà trường:
  + Nâng cao hình ảnh về một nhà trường hiện đại, chuyên nghiệp
  + Góp phần thúc đẩy hiện đại hóa, tin học hóa trong nhà trường

Ngoài ra sổ liên lạc trực tuyến còn mang tính bảo mật trong việc cung cấp thông tin. Hệ thống sẽ kiểm chứng tin nhắn gửi đến đúng số điện thoại di động đã đăng ký, để đảm bảo không ai khác ngoài phụ huynh có thể nhận thông tin của con em mình.

Với những lợi ích trên, cùng với các chức năng giải pháp mềm dẻo, linh hoạt và hệ thống đã được trải nghiệm tại nhiều trường, nhóm chúng tôi tin tưởng khi hệ thống đưa vào sử dụng sẽ làm nâng cao được chất lượng giảng dạy và làm hài lòng nhà trường và phụ huynh.

## 3.2. Lí do chọn đề tài sổ liên lạc trực tuyến

Qua nghiên cứu các lợi ích mà điện toán đám mây mang đến, nhóm chúng tôi chọn xây dựng hệ thống sổ liên lạc trực tuyến bởi vì điện toán đám mây mang đến các giải pháp cho hệ thống sổ liên lạc trực tuyến như sau:

**-** Giải quyết được bài toán đa người dùng: Hệ thống sổ liên lạc trực tuyến được xây dựng để áp dụng cho nhiều trường học, do đó lượng người sử dụng hệ thống là rất lớn. Với sự hỗ trợ về khả năng tính toán, điện toán đám mây là giải pháp tốt cho bài toán này.

- Chi phí sử dụng: Sổ liên lạc trực tuyến tuy không phải là hình thức liên lạc mới nhưng tính chất phổ biến ở các trường học chưa cao. Việc tính toán, ước lượng lượng người sử dụng hệ thống ở mỗi trường không đồng nhất và gặp nhiều khó khăn. Đế áp dụng hình thức liên lạc trực tuyến này, không phải tất cả các trường đều có khả năng kinh tế để xây dựng hệ thống cho riêng mình hoặc không phải tất cả nhà cung cấp dịch vụ có khả năng xây dựng hệ thống dành cho các trường theo mô hình client/server truyền thống. Do đó, với giải pháp tính toán chi phí theo lưu lượng sử dụng, điện toán đám mây là lựa chọn tối ưu để xây dựng hệ thống sổ liên lạc trực tuyến.

- Chi phí bảo trì: Nếu xây dựng theo mô hình client/server truyền thống, nhà trường phải tự mình hoặc nhà cung cấp dịch vụ phải tự quản lí hạ tầng của hệ thống, giải quyết các vấn đề liên quan đến mạng, v.v…Nhưng với điện toán đám mây, phía trường hoặc nhà cung cấp dịch vụ chỉ cần quan tâm đến nghiệp vụ của sổ liên lạc. Bởi vì điện toán đám mây sẽ tự giải quyết các vấn để liên quan đến hạ tầng.

## 3.3. Tính năng chính

Các tính năng chính của hệ thống sổ liên lạc trực tuyến:

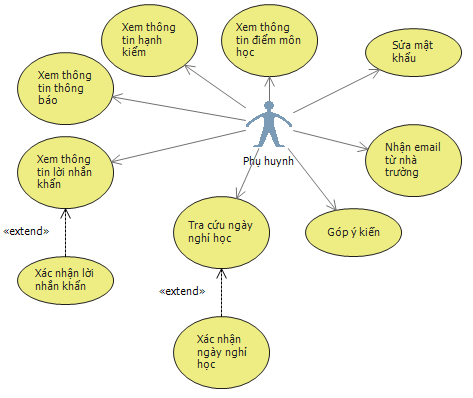
* Quản lí thông tin học sinh:
  + Quản lí danh sách học sinh của toàn trường. Hệ thống cho phép người dùng thêm, xóa, sửa thông tin của từng học sinh.
  + Quản lí điểm học tập theo từng môn, từng học kỳ trong năm học của học sinh.
  + Xếp loại hạnh kiểm cho học sinh và tự động tính toán học lực của học sinh.
  + Quản lí thông tin nghỉ học của học sinh. Hệ thống cho phép người dùng phía nhà trường thêm, sửa, xóa ngày nghỉ học và cho phép người dùng phụ huynh xác nhận ngày nghỉ học.
  + Quản lí các hoạt động hằng ngày của học sinh. Hệ thống cung cấp chức năng thêm, sửa , xóa hoạt động của học sinh, đồng thời đánh giá thái độ của học sinh khi tham gia hoạt động.
* Quản lí lớp học:
  + Quản lí thông tin lớp học của toàn trường.
  + Quản lí thông tin giáo viên chủ nhiệm của lớp. Hệ thống cho phép chỉ định giáo viên của trường làm giáo viên chủ nhiệm lớp.
  + Sắp xếp thời khóa biểu lớp. Hệ thông cung cấp chức năng thêm, sửa, xóa thời khóa biểu của lớp theo ngày, buổi và tiết học.
* Quản lí lời nhắn khẩn đến phụ huynh:
  + Hệ thống sổ liên lạc trực tuyến cho phép nhà trường gửi thông báo khẩn đến phụ huynh của học sinh chỉ định hoặc của tất cả học sinh toàn trường.
* Phụ huynh góp ý:
  + Qua hệ thống, phụ huynh có thể gửi lời góp ý đến nhà trường và thông tin này sẽ được nhà trường phản hồi.
* Quản lí danh mục:
  + Hệ thống cung cấp khả năng tùy biến với các danh mục ngành học, khối lớp, môn học, tiết học, loại điểm, hạnh kiểm, học lực.
* Quản lí người dùng:
  + Hệ thống sổ liên lạc trực tuyến cung cấp 4 nhóm người dùng cơ bản: quản trị, phụ huynh, giáo viên bộ môn và giáo viên chủ nhiệm. Mỗi nhóm có khả năng truy cập vào hệ thống tùy theo quyền được thiết lập. Ngoài ra, hệ thống cũng cho phép thêm các nhóm người dùng tự định nghĩa.
  + Quản lí tài khoản người dùng: Mỗi người dùng được cung cấp tài khoản để sử dụng hệ thống. Ngoài ra, hệ thống quản lí thời hạn sử dụng của tài khoản phụ huynh dựa vào thông tin đăng kí dịch vụ.
  + Phân quyền chức năng nhóm người dùng: Nhằm tăng khả năng tùy biến tối đa, hệ thống cho phép người dùng quản trị có khả năng thiết lập khả năng truy cập của từng nhóm người dùng cụ thể.
* Báo cáo thống kê:
  + Hệ thống sổ liên lạc trực tuyến cung cấp chức năng xuất file báo cáo về thông tin điểm số, kết quả học tập của học sinh, v.v…

## 3.4. Phân tích thiết kế

### 3.4.1. Mô hình use case

Hệ thống sổ liên lạc trực tuyến cung cấp cho phụ huynh các chức năng cơ bản sau:

* Xem thông tin điểm môn học của học sinh con em mình.
* Xem thông tin hạnh kiểm, học lực và danh hiệu mà học sinh đạt được trong học kì, năm học.
* Nhận các thông báo, lời nhắn khẩn từ phía nhà trường và xác nhận thông tin này.
* Xem thông tin ngày nghỉ học của học sinh và xác nhận thông tin này.
* Đóng góp ý kiến đến nhà trường.
* Ngoài thao tác truy vấn thông tin trên ứng dụng web, phụ huynh còn thể nhận các thông tin qua email.
* Thay đổi mật khẩu.



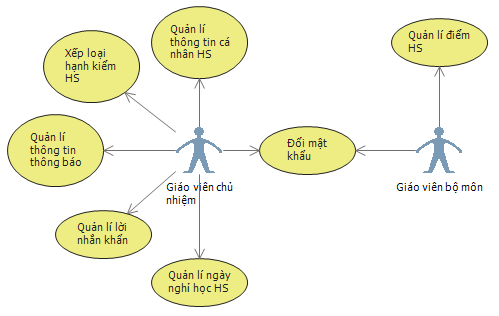
**Hình 3.1 – Use case diagram của actor Phụ huynh**

Hệ thống sổ liên lạc trực tuyến cung cấp cho giáo viên bộ môn các chức năng cơ bản sau đối với các học sinh thuộc lớp do giáo viên này chủ nhiệm:

* Quản lí thông tin cá nhân của học sinh.
* Xếp loại hạnh kiểm học sinh.
* Thông báo đến phụ huynh thông tin về học sinh và lớp học.
* Thông báo đến phụ huynh tình hình nghỉ học của học sinh.
* Quản lí tình hình tham gia các hoạt động của học sinh
* Thay đổi mật khẩu

Đối với giáo viên bộ môn, hệ thống sổ liên lạc trực tuyến cung cấp các chức năng cơ bản sau đối với các học sinh thuộc lớp do giáo viên này giảng dạy:

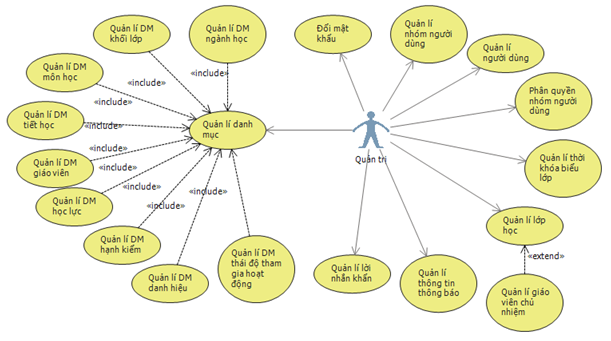
* Quản lí điểm số của học sinh.
* Thay đổi mật khẩu



**Hình 3.2 – Use case diagram của actor Giáo viên**

Hệ thống sổ liên lạc trực tuyến cung cấp cho người quản trị hệ thống các chức năng cơ bản sau:

* Quản lí danh mục, bao gồm danh mục ngành học, khối lớp, môn học, loại điểm, tiết học, hạnh kiểm, học lực, danh hiệu, thái độ tham gia hoạt động và giáo viên.
* Quản lí danh sách lớp học
* Phân công giáo viên chủ nhiệm
* Sắp xếp thời khóa biểu lớp
* Quản lí danh sách người dùng, nhóm người dùng
* Phân quyền truy cập vào hệ thống của nhóm người dùng.

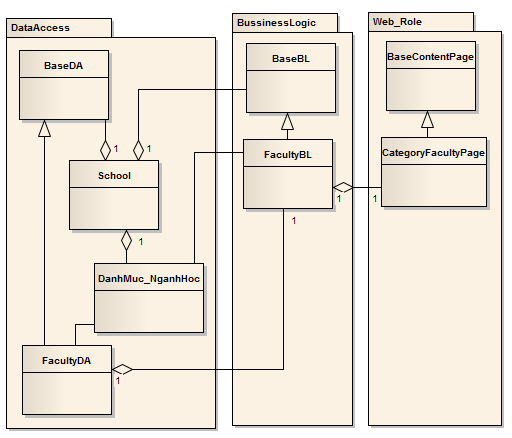


**Hình 3.3 – Use case diagram của actor Quản trị**

### 3.4.2. Sơ đồ class

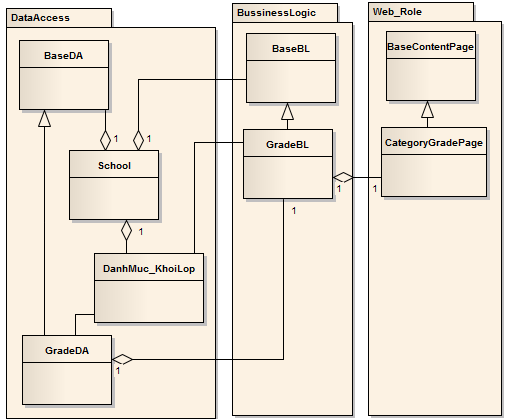
##### (Bổ sung sau)

##### Sơ đồ class danh mục ngành học:

****

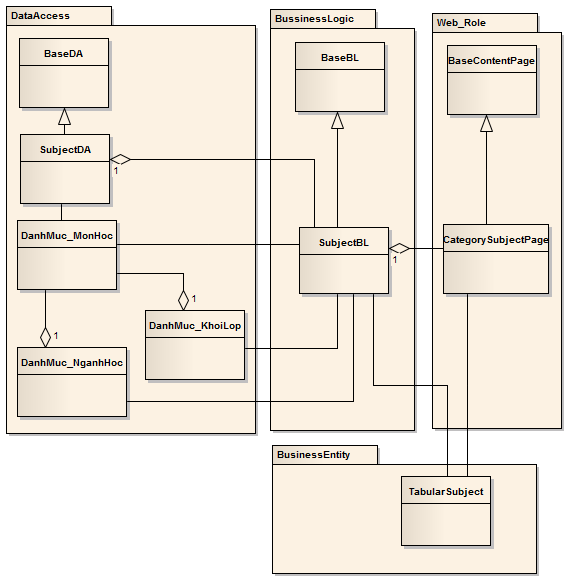
**Danh mục Ngành học**

##### 3.4.2.2. Danh mục Khối Lớp

****

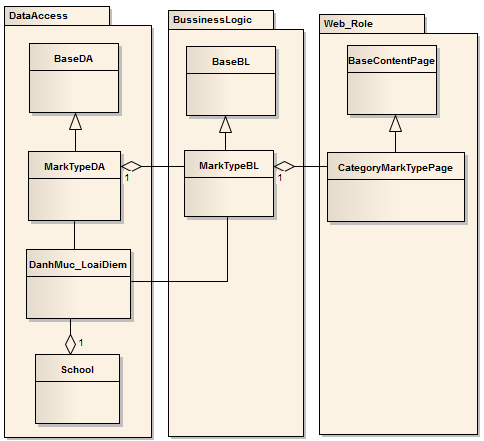
**Danh mục Khối Lớp**

##### Danh mục Môn học:

****

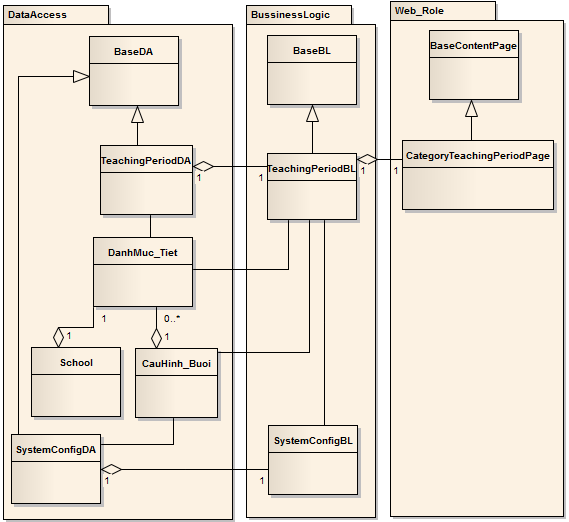
**Danh mục Môn học**

##### Danh mục Loại điểm:

****

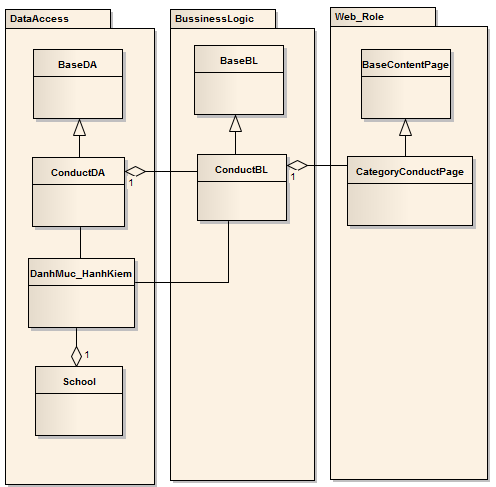
**Danh mục Loại điểm**

##### Danh mục Tiết học:

****

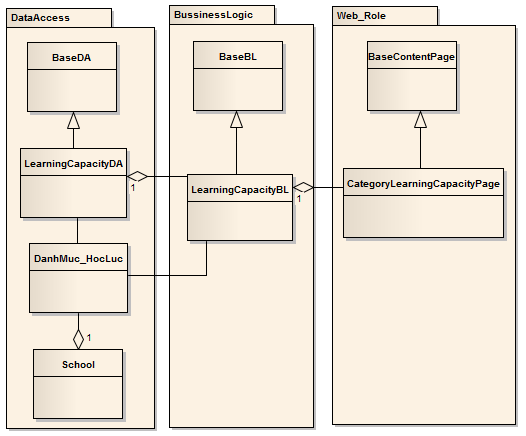
**Danh mục tiết học**

##### Danh mục Hạnh kiểm:

****

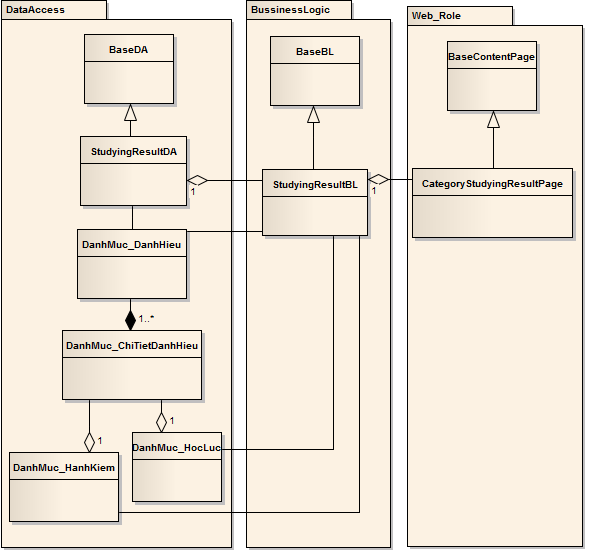
**Danh mục Hạnh kiểm**

##### Danh mục học lực:

****

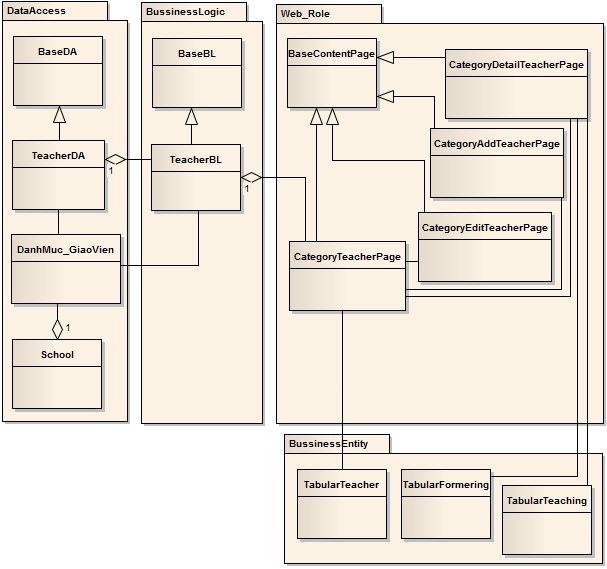
**Danh mục Học lực**

##### Danh mục Danh hiệu:

****

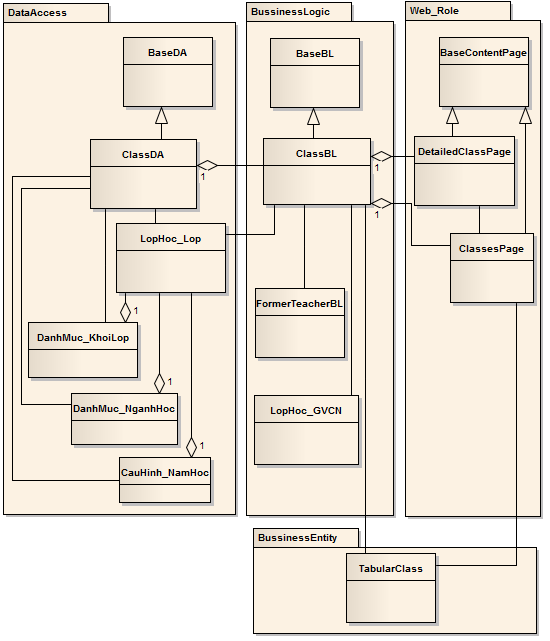
**Danh mục Danh hiệu**

##### Danh mục giáo viên:

****

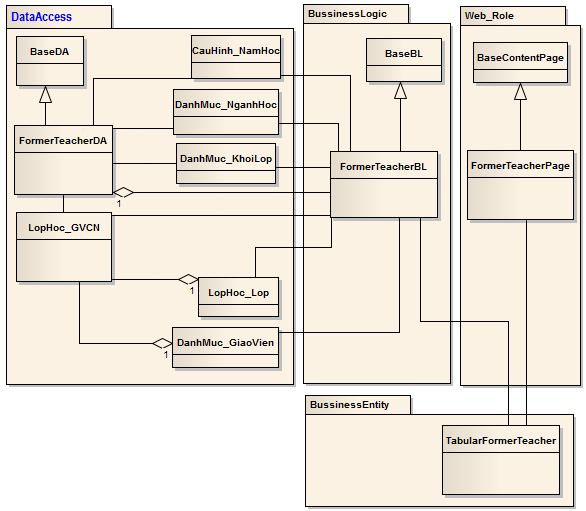
**Danh mục giáo viên**

##### Lớp học:

****

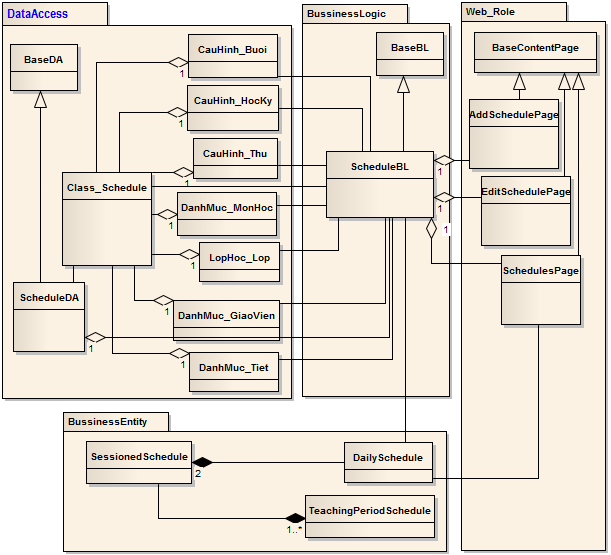
**Hình Lớp học**

##### Giáo viên chủ nhiệm:

****

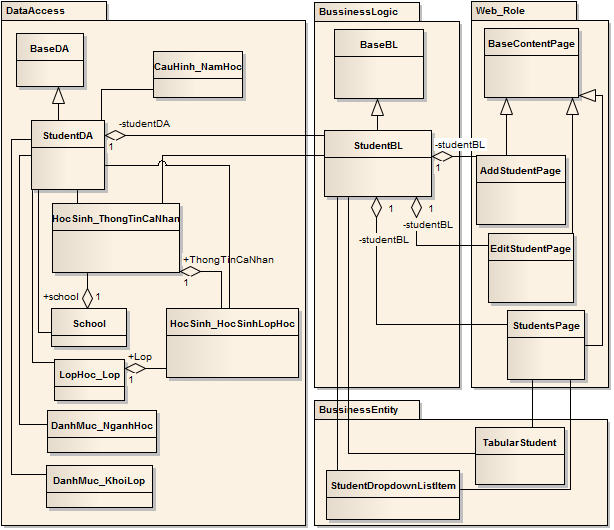
**Giáo viên chủ nhiệm**

##### Thời khóa biểu:

****

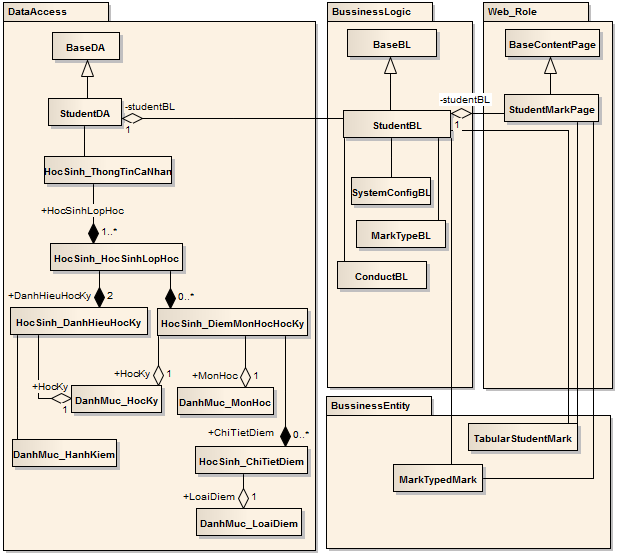
**Thời khóa biểu**

##### Học sinh:

****

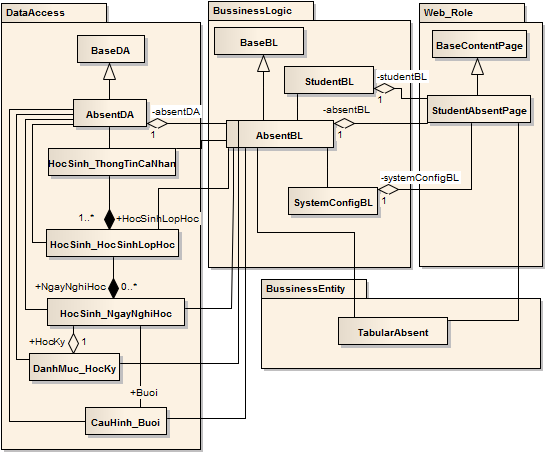
**Học sinh**

##### Điểm học sinh:

****

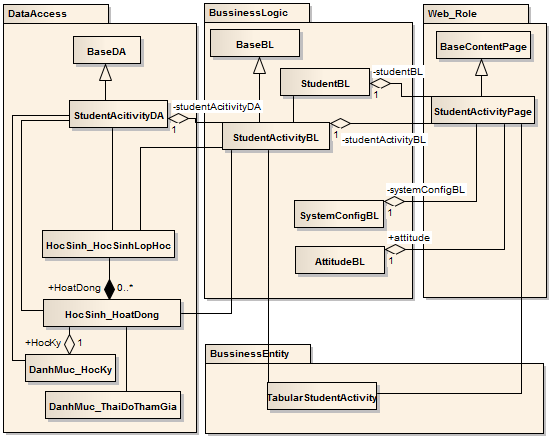
**Điểm học sinh**

##### Ngày nghỉ của học sinh:

****

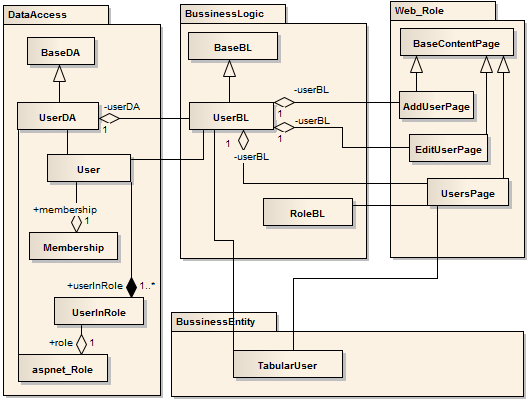
**Ngày nghỉ học của Học sinh**

##### Hoạt động của Học sinh:

****

**Hoạt động của Học sinh**

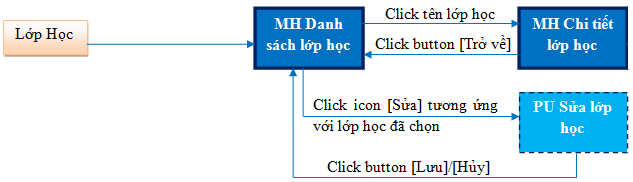
##### Người dùng:

****

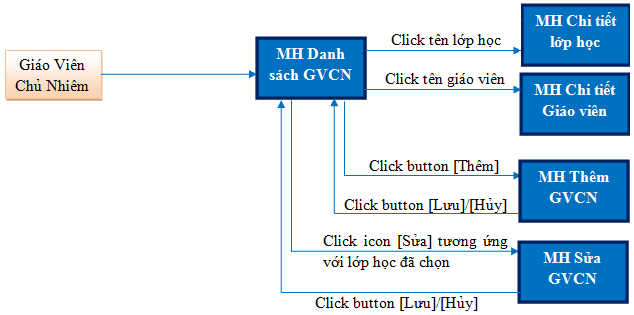
**Người dùng**

### 3.4.3. Thiết kế màn hình

(Bổ sung sau)  
Quản lí lớp học:



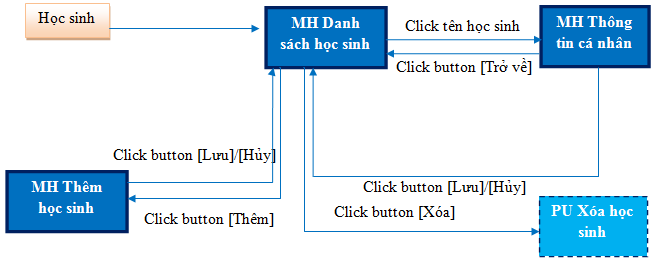
Quản lí giáo viên chủ nhiệm:



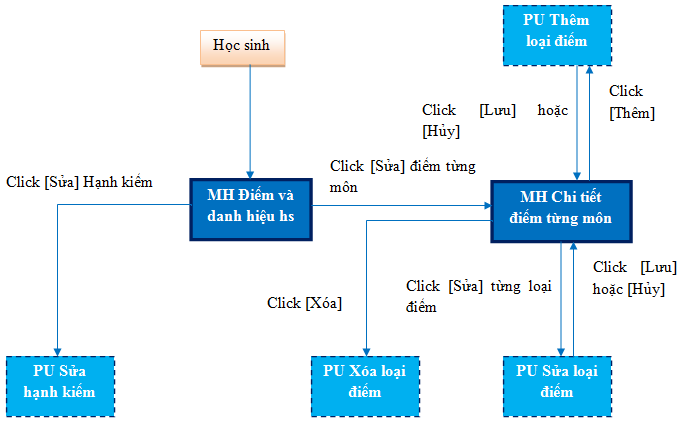
Quản lí thời khóa biểu lớp:



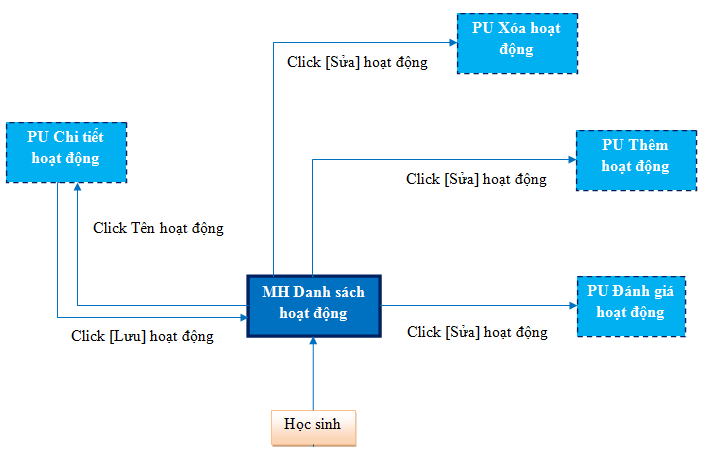
Quản lí học sinh:



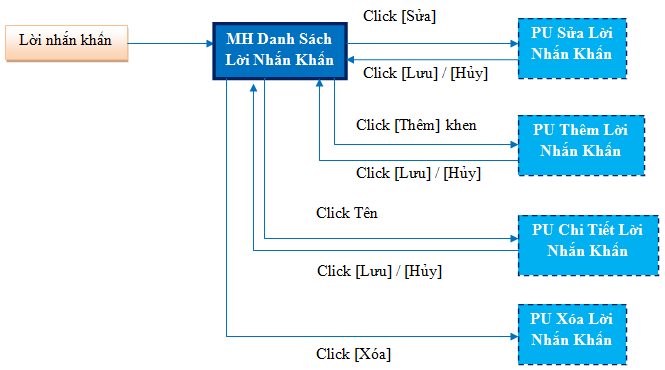
Quản lí điểm học sinh:



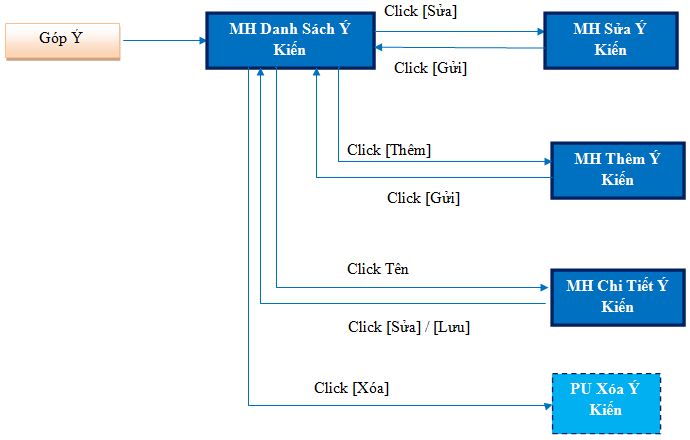
Quản lí hoạt động của học sinh:



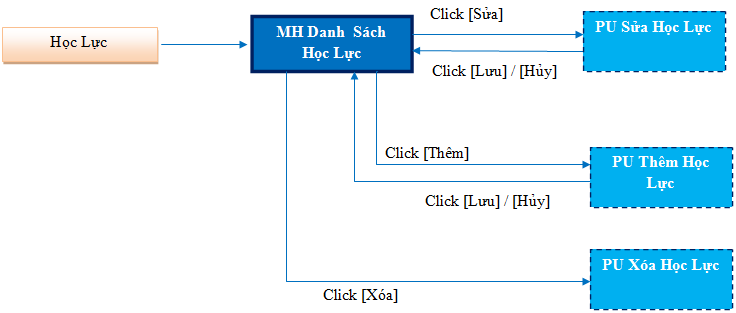
Quản lí lời nhắn khẩn:



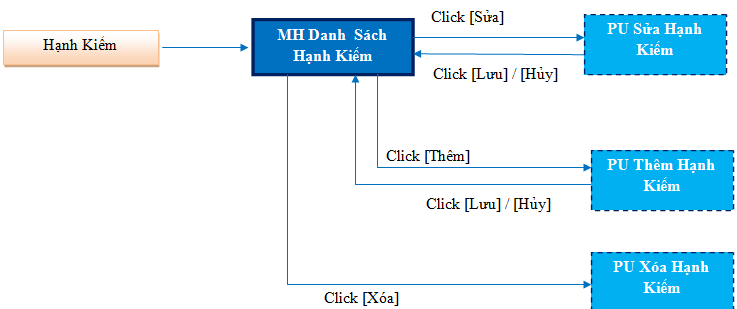
Quản lí góp ý của phụ huynh:



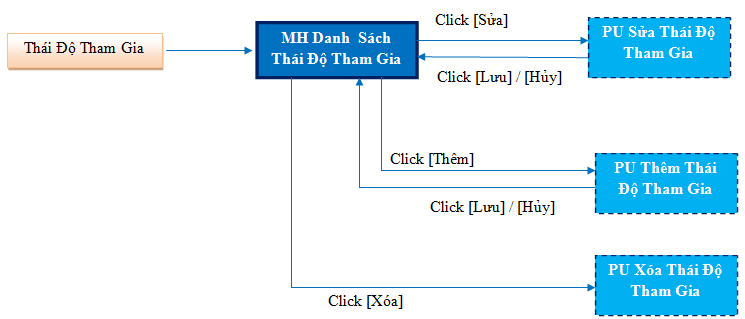
Danh mục học lực:



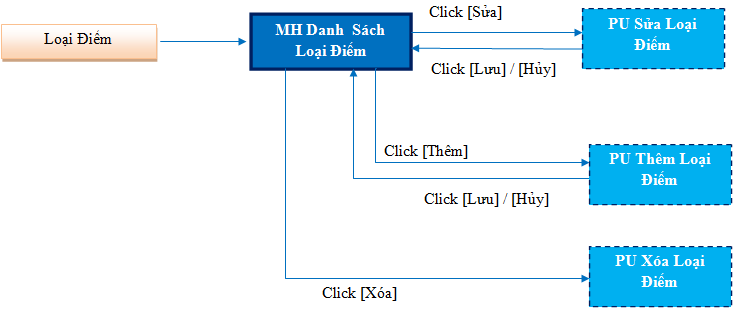
Danh mục hạnh kiểm:



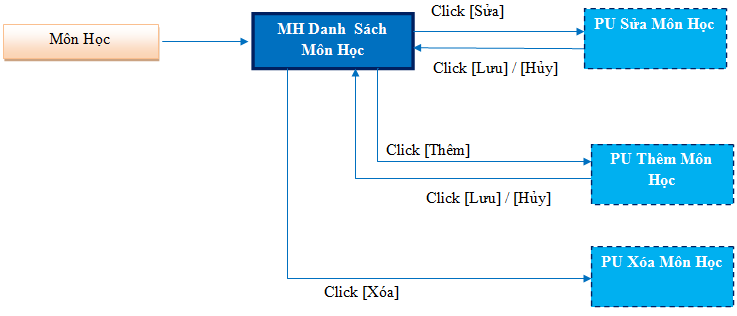
Danh mục thái độ tham gia hoạt động:



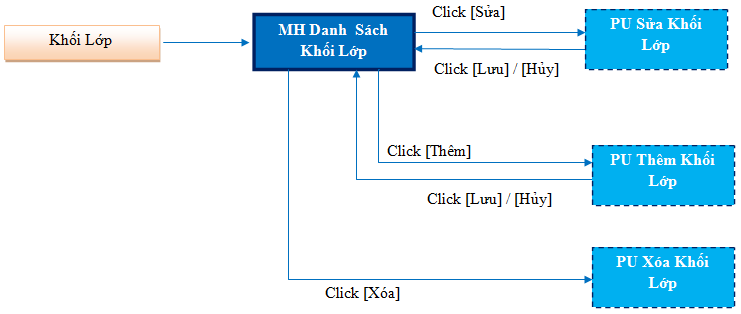
Danh mục loại điểm:



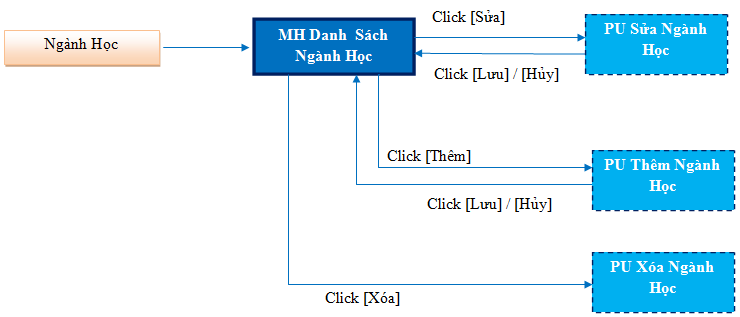
Danh mục môn học:



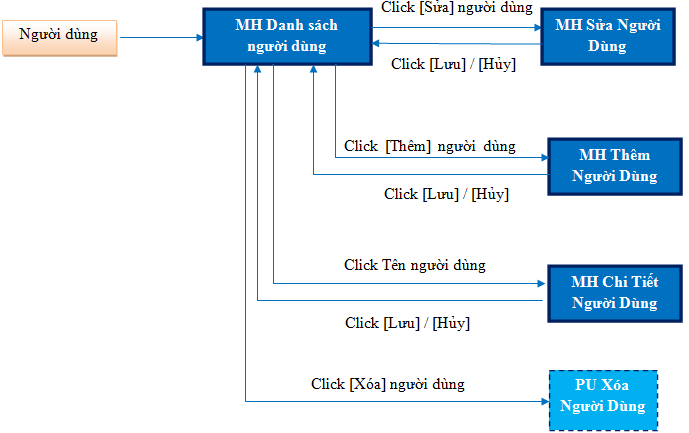
Danh mục khối lớp:



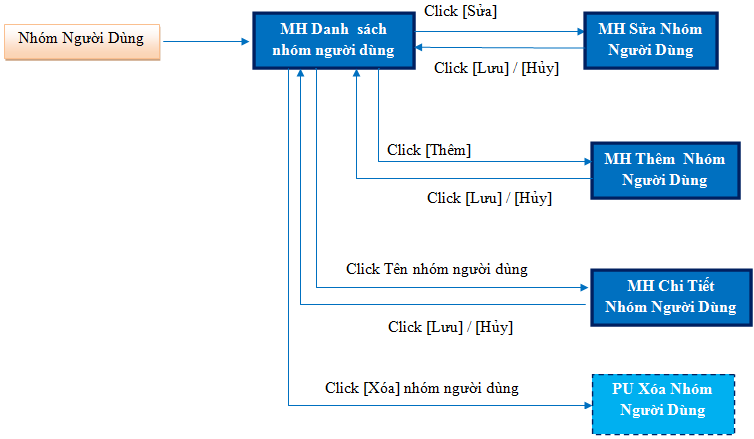
Danh mục ngành học:



Quản lí người dùng:



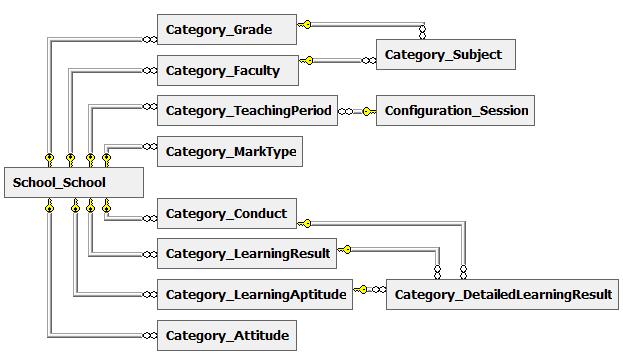
Quản lí nhóm người dùng:



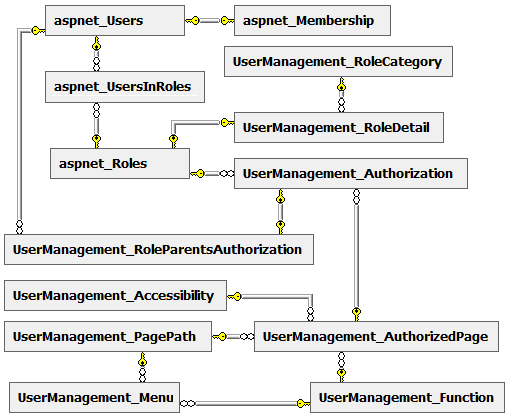
### 3.4.4. Thiết kế cơ sở dữ liệu

#### 3.4.4.1. Sơ đồ quan hệ

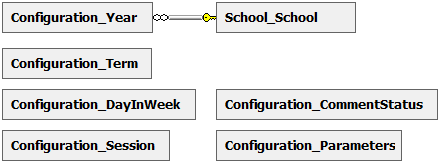
Dưới đây là sơ đồ quan hệ các bảng trong hệ thống sổ liên lạc trực tuyến:

****

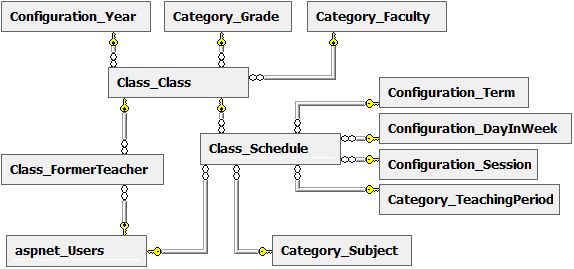
**Hình - Sơ đồ quan hệ các bảng Danh mục**

****

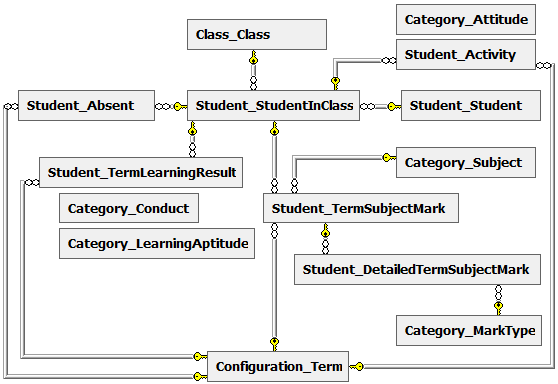
**Hình - Sơ đồ quan hệ các bảng module Người dùng**

****

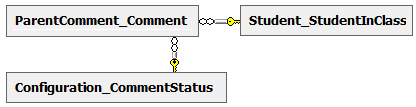
**Hình - Sơ đồ quan hệ các bảng lưu thông tin cấu hình**

****

**Hình - Sơ đồ quan hệ các bảng module lớp học**

****

**Hình - Sơ đồ quan hệ các bảng module học sinh**

****

**Hình - Sơ đồ quan hệ các bảng module Góp ý của phụ huynh**

****

**Hình - Sơ đồ quan hệ các bảng module Lời nhắn đến phụ huynh**

#### 3.4.4.2. Danh sách bảng

Danh sách các bảng thuộc module danh mục:

- Category\_Grade: lưu thông tin danh mục khối lớp của trường học.

- Category\_Faculty: lưu thông tin danh mục ngành học của trường học.

- Category\_Subject: lưu thông tin danh mục môn học của trường học.

- Category\_TeachingPeriod: lưu danh mục thông tin tiết học của trường học.

- Category\_MarkType: lưu thông tin danh mục loại điểm của trường học.

- Category\_Conduct: lưu thông tin danh mục hạnh kiểm của trường học.

- Category\_LearningAptitude: lưu thông tin danh mục học lực của trường học.

- Category\_LearningResult: lưu thông tin danh mục danh hiệu học sinh của trường học.

- Category\_DetailedLearningResult: lưu thông tin chi tiết danh hiệu học sinh của trường học.

- Category\_Attitude: lưu thông tin danh mục thái độ tham gia hoạt động của học sinh của trường học.

Danh sách các bảng thuộc module người dùng:

- aspnet\_Roles: lưu thông tin nhóm người dùng trong hệ thống.

- UserManagement\_RoleDetail: lưu thông tin chi tiết của nhóm người dùng.

- UserManagement\_RoleCategory: lưu thông tin phân loại nhóm người dùng.

- aspnet\_Users: lưu thông tin người dùng trong hệ thống.

- aspnet\_Membership: lưu thông tin chi tiết của người dùng.

- aspnet\_UsersInRoles: lưu thông tin quan hệ giữa người dùng và nhóm người dùng, cho biết người dùng nào thuộc nhóm người dùng nào.

- UserManagement\_Function: lưu thông tin chức năng được cung cấp trong hệ thống.

- UserManagement\_PagePath: lưu thông tin các trang .aspx được sử dụng để hiển thị.

- UserManagement\_Accessibility: lưu thông tin các khả năng truy cập vào hệ thống.

- UserManagement\_AuthorizedPage: lưu thông tin cho biết khả năng truy cập chức năng đối với trang .aspx.

- UserManagement\_Authorization: lưu thông tin phân quyền của nhóm người dùng.

- UserManagement\_Menu: lưu thông tin menu cả hệ thống.

- UserManagement\_RoleParentsAuthorization: lưu thông tin phân quyền của người dùng thuộc nhóm “Phụ huynh”.

Danh sách các bảng thuộc module lớp học:

- Class\_Class: lưu thông tin các lớp học.

- Class\_FormerTeacher: Lưu thông tin giáo viên chủ nhiệm lớp.

- Class\_Schedule: Lưu thông tin thời khóa biểu.

Danh sách các bảng thuộc module học sinh:

- Student\_Student: lưu thông tin cá nhân của học sinh.

- Student\_StudentInClass: Lưu thông tin biểu thị mối quan hệ giữa học sinh và lớp học.

- Student\_Absent: Lưu thông tin ngày nghỉ học của học sinh.

- Student\_Activity: Lưu thông tin hoạt động của học sinh.

- Student\_TermLearningResult: Lưu thông tin kết quả học tập trong học kì của học sinh.

- Student\_TermSubjectMark: Lưu thông tin điểm của môn học trong học kì của học sinh.

- Student\_DetailedTermSubjectMark: Lưu thông tin chi tiết các điểm của môn học trong học kì của học sinh.

Danh sách các bảng thuộc module góp ý của phụ huynh:

- ParentComment\_Comment: lưu thông tin góp ý của học sinh.

Danh sách các bảng thuộc module lời nhắn đến phụ huynh:

- MessageToParents\_Message: lưu thông tin lời nhắn đến phụ huynh.

Danh sách các bảng thuộc lưu thông tin cấu hình:

- Configuration\_Year: lưu thông tin năm học của trường học.

- Configuration\_Term: lưu thông tin cấu hình học kì.

- Configuration\_DayInWeek: lưu thông tin cấu hình các ngày trong tuần.

- Configuration\_Session: lưu thông tin cấu hình các buổi trong ngày.

- Configuration\_CommentStatus: lưu thông tin cấu hình trạng thái của góp ý của phụ huynh.

- Configuration\_Parameters: lưu thông tin cấu hình các giá trị mặc định.

- School\_School: lưu thông tin trường học sử dụng hệ thống.

#### 3.4.4.3. Mô tả chi tiết bảng

Các bảng dưới đây mô tả chi tiết các bảng thuộc module danh mục.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| GradeId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| GradeName | nvarchar(50) | Không |  | Tên khối lớp.  Tên này là duy nhất trong dữ liệu khối lớp của từng trường học. |
| DisplayedOrder | smallint | Không | 1 | Thứ tự hiển thị của khối lớp trong dữ liệu khối lớp của từng trường học. |
| SchoolId | int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết khối lớp này thuộc trường học nào. |

**Bảng - Thông tin bảng Category\_Grade**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| FacultyId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| FacultyName | nvarchar(100) | Không |  | Tên ngành học.  Tên này là duy nhất trong dữ liệu ngành học của từng trường học. |
| Description | nvarchar(200) | Có |  | Mô tả ngành học. |
| SchoolId | Int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết ngành học này thuộc trường học nào. |

**Bảng - Thông tin bảng Category\_Faculty**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| SubjectId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| SubjectName | nvarchar(50) | Không |  | Tên môn học.  Tên này là duy nhất trong dữ liệu môn học của từng trường học. |
| FacultyId | int | Không |  | Khóa ngoại.  Tham chiếu đến bảng Category\_Faculty.  Cho biết môn học này thuộc ngành học nào trong trường học. |
| GradeId | Int | Không |  | Khóa ngoại.  Tham chiếu đến bảng Category\_Grade.  Cho biết môn học này thuộc khối lớp nào trong trường học. |
| MarkRatio | Float | Không | 1 | Hệ số điểm của môn học. |

**Bảng - Thông tin bảng Category\_Subject**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| TeachingPeriodId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| TeachingPeriodName | nvarchar(50) | Không |  | Tên tiết học.  Tên này là duy nhất trong dữ liệu tiết học của từng trường học. |
| TeachingPeriodOrder | Int | Không |  | Thứ tự của tiết học trong dữ liệu tiết học của từng trường học. |
| BeginTime | Datetime | Không |  | Thời điểm bắt đầu tiết học |
| EndTime | Datetime | Không |  | Thời điểm kết thúc tiết học |
| SessionId | Int | Không |  | Khóa ngoại.  Tham chiếu đến bảng Configuration\_Session.  Cho biết tiết học này thuộc buổi nào trong ngày. |
| SchoolId | Int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết tiết học này thuộc trường học nào. |

**Bảng - Thông tin bảng Category\_TeachingPeriod**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| MarkTypeId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| MarkTypeName | nvarchar(50) | Không |  | Tên loại điểm.  Tên này là duy nhất trong dữ liệu loại điểm của từng trường học. |
| MarkRatio | Float | Không | 1 | Hệ số của loại điểm. |
| MaxQuantity | Smallint | Không |  | Số lượng điểm tối đa của loại điểm trong một học kì |
| IsUsedForCalculatingAvg | Bit | Không |  | Cho biết loại điểm này có được áp dụng để tính điểm trung bình hay không |
| SchoolId | Int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết loại điểm này thuộc trường học nào. |

**Bảng - Thông tin bảng Category\_MarkType**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| ConductId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| ConductName | nvarchar(50) | Không |  | Tên hạnh kiểm.  Tên này là duy nhất trong dữ liệu danh mục hạnh kiểm của từng trường học. |
| SchoolId | int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết danh mục hạnh kiểm này thuộc trường học nào. |

**Bảng - Thông tin bảng Category\_Conduct**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| LearningAptitudeId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| LearningAptitudeName | nvarchar(50) | Không |  | Tên học lực.  Tên này là duy nhất trong dữ liệu danh mục học lực của từng trường học. |
| BeginAverageMark | float | Không |  | Điểm trung bình bắt đầu trong đoạn điểm trung bình dùng đế xác định học lực. |
| EndAverageMark | Float | Không |  | Điểm trung bình kết thúc trong đoạn điểm trung bình dùng đế xác định học lực. |
| SchoolId | int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết học lực này thuộc trường học nào. |

**Bảng - Thông tin bảng Category\_LearningAptitude**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| LearningResultId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| LearningResultName | nvarchar(50) | Không |  | Tên danh hiệu.  Tên này là duy nhất trong dữ liệu danh mục danh hiệu của từng trường học. |
| SchoolId | int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết danh mục danh hiệu này thuộc trường học nào. |

**Bảng - Thông tin bảng Category\_LearningResult**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| LearningResultId | int | Không |  | Khóa chính.  Khóa ngoại tham chiếu đến bảng Category\_LearningResult.  Cho biết chi tiết danh hiệu của danh hiệu. |
| LearningAptitudeId | int | Không |  | Khóa chính.  Khóa ngoại tham chiếu đến bảng Category\_LearningAptitude.  Cho biết chi tiết danh hiệu thuộc học lực nào. |
| ConductId | int | Không |  | Khóa chính.  Khóa ngoại tham chiếu đến bảng Category\_Conduct.  Cho biết chi tiết danh hiệu thuộc hạnh kiểm nào. |

**Bảng - Thông tin bảng Category\_DetailedLearningResult**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| AttitudeId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| AttitudeName | nvarchar(50) | Không |  | Tên thái độ tham gia hoạt động.  Tên này là duy nhất trong dữ liệu danh mục thái độ tham gia hoạt động của từng trường học. |
| SchoolId | int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết danh mục thái độ tham gia hoạt động này thuộc trường học nào. |

**Bảng - Thông tin bảng Category\_Attitude**

Các bảng dưới đây mô tả chi tiết các bảng thuộc module người dùng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| RoleId | uniqueidentifier | Không | newid() | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| RoleName | nvarchar(256) | Không |  | Tên nhóm người dùng.  Tên này là duy nhất trong dữ liệu nhóm người dùng. |
| Description | nvarchar(256) | Không |  | Mô tả nhóm người dùng. |

**Bảng - Thông tin bảng aspnet\_Roles**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| RoleId | uniqueidentifier | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| DisplayedName | nvarchar(100) | Không |  | Tên hiển thị của nhóm người dùng.  Tên này là duy nhất trong dữ liệu nhóm người dùng của từng trường học. |
| IsDeletable | Bit | Không |  | Biểu thị nhóm người dùng có được phép xóa hay không. |
| ParentRoleId | Uniqueidentifier | Có |  | Mã nhóm người dùng cha. |
| RoleCategoryId | nvarchar(50) | Không |  | Khóa ngoại.  Tham chiếu đến bảng UserManagement\_RoleCategory.  Biểu thị loại nhóm người dùng. |
| SchoolId | Int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết khối lớp này thuộc trường học nào. |

**Bảng - Thông tin bảng UserManagement\_RoleDetail**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| RoleCategoryId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| FunctionFlag | nvarchar(100) | Không |  | Định nghĩa các giá trị FunctionFlag trong bảng UserManagement\_Function. |

**Bảng - Thông tin bảng UserManagement\_RoleCategory**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| UserId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần.  Mã người dùng. |
| UserName | nvarchar(50) | Không |  | Tên người dùng.  Tên này là duy nhất trong dữ liệu người dùng. |

**Bảng - Thông tin bảng aspnet\_Users**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| UserId | Uniqueidentifier | Không | 1 | Khóa chính.  Khóa ngoại tham chiếu tới bảng aspnet\_Users. |
| Password | nvarchar(128) | Có |  | Mật khẩu. Mật khẩu này được mã hóa theo giá trị của PasswordFormat. |
| PasswordFormat | Int | Không | 1 | Định dạng mã hóa mật khẩu. |
| Email | nvarchar(256) | Có |  | Email của người dùng. |
| CreateDate | Datetime | Không |  | Ngày tạo người dùng. |
| LastLoginDate | Datetime | Không |  | Ngày gần nhất người dùng đăng nhập vào hệ thống. |
| LastPasswordChangedDate | Datetime | Không |  | Ngày gần nhất người dùng đổi mật khẩu. |
| IsDeletable | Bit | Có |  | Biểu thị người dùng này có khả năng bị xóa hay không. |
| FullName | nvarchar(200) | Có |  | Tên đầy đủ của người dùng. |
| Birthday | Datetime | Có |  | Ngày sinh. |
| Gender | Bit | Có |  | Giới tính. |
| Phone | nvarchar(50) | Có |  | Điện thoại liên lạc. |
| Photo | Image | Có |  | Hình ảnh. |
| Address | nvarchar(200) | Có |  | Địa chỉ liên lạc. |
| IsTeacher | Bit | Có |  | Biểu thị người dùng này có phải là “Giáo viên” hay không. |
| SchoolId | Int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết khối lớp này thuộc trường học nào. |

**Bảng - Thông tin bảng aspnet\_Membership**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| UserId | Uniqueidentifier | Không | 1 | Khóa chính.  Khóa ngoại tham chiếu đến bảng aspnet\_Users.  Biểu thị mối quan hệ với người dùng. |
| RoleId | Uniqueidentifier | Không |  | Khóa chính.  Khóa ngoại tham chiếu đến bảng aspnet\_Roles.  Biểu thị mối quan hệ với nhóm người dùng. |

**Bảng - Thông tin bảng aspnet\_UsersInRoles**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| FunctionId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| FunctionName | nvarchar(100) | Không |  | Tên chức năng. |
| FunctionCategory | nvarchar(50) | Không |  | Phân loại chức năng.  Dùng để xác định nhóm chức năng mà chức năng này thuộc. |
| FunctionFlag | nvarchar(50) | Không |  | Dùng để xác định chức năng sẽ dùng được cung cấp cho nhóm người dùng nào. |
| ParentFunctionId | Int | Có |  | Mã chức năng cha. |

**Bảng - Thông tin bảng UserManagement\_Function**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| AccessibilityId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| AccessibilityName | nvarchar(50) | Không |  | Tên khả năng truy cập chức năng. |

**Bảng - Thông tin bảng UserManagement\_Accessibility**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| PagePathId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| PageTitle | nvarchar(100) | Không |  | Tiêu đề trang .aspx. |
| PhysicalPath | nvarchar(200) | Không |  | Đường dẫn vật lí của trang .aspx. |

**Bảng - Thông tin bảng UserManagement\_PagePath**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| AuthorizedPageId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| FunctionId | int | Không |  | Khóa ngoại.  Tham chiếu đến bảng UserManagement\_Function.  Biểu thị mối quan hệ với chức năng. |
| AccessibilityId | int | Không |  | Khóa ngoại.  Tham chiếu đến bảng UserManagement\_Accessibility.  Biểu thị mối quan hệ với khả năng truy cập. |
| PagePathId | Int | Không |  | Khóa ngoại.  Tham chiếu đến bảng UserManagement\_PagePath.  Biểu thị mối quan hệ với trang .aspx. |
| Description | nvarchar(200) | Có |  | Mô tả |

**Bảng - Thông tin bảng UserManagement\_AuthorizedPage**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| AuthorizationId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| RoleId | Uniqueidentifier | Không |  | Khóa ngoại.  Tham chiếu tới bảng aspnet\_Roles. |
| AuthorizedPagePathId | int | Không |  | Khóa ngoại.  Tham chiếu tới bảng UserManagement\_AuthorizedPage.  Biểu thị mối quan hệ với trang .aspx được phân khả năng truy cập vào chức năng cụ thể. |

**Bảng - Thông tin bảng UserManagement\_Authorization**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| RoleParentAuthorizationId | int | Không | 1 | Khóa ngoại.  Tham chiếu tới bảng UserManagement\_Authorization.  Biểu thị mối quan hệ với các quyền của nhóm người dùng “Phụ huynh”. |
| UserParentId | Uniqueidentifier | Không |  | Khóa ngoại.  Tham chiếu tới bảng aspnet\_Users.  Biểu thị mối quan hệ với người dùng “Phụ huynh”. |

**Bảng - Thông tin bảng UserManagement\_RoleParentsAuthorization**

Các bảng dưới đây mô tả chi tiết các bảng thuộc module lớp học.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| ClassId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| ClassName | Nvarchar(10) | Không |  | Tên lớp học.  Tên này là duy nhất trong dữ liệu lớp học của từng trường học. |
| StudentQuantity | Int | Không | 0 | Sỉ số học sinh của lớp. |
| YearId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Configuration\_Year.  Biểu thị năm học của lớp học. |
| FacultyId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Category\_Faculty.  Biểu thị ngành học của lớp học. |
| GradeId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Category\_Grade.  Biểu thị khối lớp của lớp học. |
| SchoolId | Int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết ngành học này thuộc trường học nào. |

**Bảng - Thông tin bảng Class\_Class**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| FormerTeacherId | int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| ClassId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Class\_Class.  Biểu thị lớp học mà giáo viên chủ nhiệm. |
| TeacherId | Uniqueidentifier | Không |  | Khóa ngoại.  Tham chiếu tới bảng aspnet\_Users.  Biểu thị giáo viên được phân công chủ nhiệm. |

**Bảng - Thông tin bảng Class\_FormerTeacher**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| ScheduleId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| SubjectId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Class\_Class.  Biểu thị lớp học mà giáo viên chủ nhiệm. |
| TeacherId | Uniqueidentifier | Không |  | Khóa ngoại.  Tham chiếu tới bảng Class\_Class.  Biểu thị lớp học mà giáo viên chủ nhiệm. |
| ClassId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Class\_Class.  Biểu thị lớp học được sắp xếp thời khóa biểu. |
| TermId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Configuration\_Term.  Biểu thị học kì của thông tin thời khóa biểu |
| DayInWeekId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Configuration\_DayInWeek.  Biểu thị ngày trong tuần của thông tin thời khóa biểu. |
| SessionId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Configuration\_Session.  Biểu thị buổi trong ngày của thông tin thời khóa biểu. |
| TeachingPeriodId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Category\_TeachingPeriod.  Biểu thị tiết học của thông tin thời khóa biểu. |

**Bảng - Thông tin bảng Class\_Schedule**

Danh sách các bảng thuộc module học sinh:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| StudentId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| StudentCode | Nvarchar(20) | Không |  | Mã hiển thị của học sinh.  Mã này là duy nhất trong dữ liệu học sinh của từng trường học. |
| FullName | Nvarchar(100) | Không |  | Tên đầy đủ |
| StudentBirthday | Datetime | Không |  | Ngày sinh của học sinh. |
| Gender | Bit | Không |  | Giới tính của học sinh. |
| Birthplace | Nvarchar(200) | Có |  | Nơi sinh của học sinh. |
| Address | Nvarchar(200) | Không |  | Địa chỉ liên lạc. |
| ContactPhone | Nvarchar(20) | Có |  | Địa thoại liên lạc. |
| Photo | Image | Có |  | Hình ảnh của học sinh. |
| FatherName | Nvarchar(100) | Có |  | Tên cha của học sinh. |
| FatherBirthday | Datetime | Có |  | Ngày sinh của cha của học sinh. |
| FatherJob | Nvarchar(100) | Có |  | Nghề nghiệp của cha của học sinh. |
| MotherName | Nvarchar(100) | Có |  | Tên mẹ của học sinh. |
| MotherBirthday | Datetime | Có |  | Ngày sinh của mẹ của học sinh. |
| MotherJob | Nvarchar(100) | Có |  | Nghề nghiệp của mẹ của học sinh. |
| PatronName | Nvarchar(100) | Có |  | Tên người đỡ đầu của học sinh. |
| PatronBirthday | Datetime | Có |  | Ngày sinh của người đỡ đầu của học sinh. |
| PatronJob | Nvarchar(100) | Có |  | Nghề nghiệp người đỡ đầu của học sinh. |
| SchoolId | int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết học sinh này thuộc trường học nào. |

**Bảng - Thông tin bảng Student\_Student**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| StudentInClassId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| StudentId | Int | Không |  | Khóa ngoại.  Tham chiếu đến bảng Student\_Student.  Biểu thị mối quan hệ với học sinh. |
| ClassId | Int | Không |  | Khóa ngoại.  Tham chiếu đến bảng Class\_Class.  Biểu thị mối quan hệ với lớp học. |

**Bảng - Thông tin bảng Student\_StudentInClass**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| AbsentId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| Date | datetime | Không |  | Ngày học sinh nghỉ học. |
| SessionId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Configuration\_Session.  Cho biết buổi mà học sinh này nghỉ. |
| IsAsked | Bit | Không |  | Biểu thị ngày nghỉ có được xin phép hay chưa. |
| Reason | Nvarchar(300) | Có |  | Lí do nghỉ học. |
| IsConfirmed | Bit | Không |  | Biểu thị phụ huynh có xác nhận ngày nghỉ hay chưa. |
| StudentInClass | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Student\_ StudentInClass.  Cho biết thông tin học sinh nghỉ học. |
| TermId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Configuration\_Term  Cho biết học kì mà học sinh này nghỉ. |

**Bảng - Thông tin bảng Student\_Absent**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| ActivityId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| Title | Nvarchar(100) | Không |  | Tiêu đề hoạt động. |
| ActivityContent | Nvarchar(300) | Có |  | Nội dung hoạt động |
| Date | Datetime | Không |  | Ngày hoạt động |
| AttitudeId | Int | Có |  | Khóa ngoại.  Tham chiếu tới bảng Category\_Attitude.  Cho biết thái độ tham gia của học sinh. |
| StudentInClassId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Student\_StudentInClass  Cho biết thái độ tham gia của học sinh. |
| TermId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Configuration\_Term.  Cho biết học kì mà học sinh này tham gia hoạt động. |

**Bảng - Thông tin bảng Student\_Activity**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| TermLearningResultId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| TermId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Configuration\_Term.  Cho biết học kì học tập. |
| StudentInClassId | Int |  |  | Khóa ngoại.  Tham chiếu tới bảng Student\_ StudentInClass.  Cho biết thông tin học sinh học tập. |
| TermAverageMark | Float | Không | -1 | Điểm trung bình học kì |
| TermLearningAptitudeId | Int | Có |  | Có giá trị của LearningAptitudeId của Category\_ LearningAptitude.  Cho biết học lực trong học kì của học sinh. |
| TermConductId | Int | Có |  | Có giá trị của Conduct Id của Category\_ Conduct.  Cho biết hạnh kiểm trong học kì của học sinh. |

**Bảng - Thông tin bảng Student\_TermLearningResult**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| TermSubjectMarkId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| StudentInClassId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Student\_ StudentInClass.  Cho biết thông tin học sinh học tập. |
| SubjectId | Uniqueidentifier | Không |  | Khóa ngoại.  Tham chiếu tới bảng Category\_ Subject.  Cho biết thông tin môn học. |
| TermId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Configuration\_ Term.  Cho biết thông tin học kì. |
| AverageMark | Float | Không | -1 | Điểm trung bình môn học trong học kì của học sinh. |

**Bảng - Thông tin bảng Student\_TermSubjectMark**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| DetailedTermSubjectMark | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| TermSubjectMarkId | Int | Không |  | Tên khối lớp.  Tên này là duy nhất trong dữ liệu khối lớp của từng trường học. |
| MarkTypeId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Category\_ MarkType.  Biểu thị mối quan hệ với loại điểm. |
| MarkValue | float | Không |  | Giá trị điểm. |

**Bảng - Thông tin bảng Student\_DetailedTermSubjectMark**

Danh sách các bảng thuộc module góp ý của phụ huynh:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| CommentId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| Title | Int | Không |  | Tiêu đề góp ý. |
| CommentContent | Int | Không |  | Nội dung góp ý. |
| Feedback | float | Không |  | Nội dung phản hồi từ nhà trường |
| Date | Datetime | Không |  | Ngày gửi góp ý. |
| CommentStatusId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Configuration\_CommentStatus.  Biểu thị trạng thái của lời góp ý. |
| StudentInClassId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Student\_StudentInClass.  Biểu thị góp ý được gửi từ phụ huynh học sinh này. |

**Bảng - Thông tin bảng ParentComment\_Comment**

Danh sách các bảng thuộc module lời nhắn đến phụ huynh:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| MessageId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| Title | Nvarchar(100) | Không |  | Tiêu đề lời nhắn. |
| MessageContent | Nvarchar(300) | Có |  | Nội dung lời nhắn. |
| Date | Datetime | Không |  | Ngày gửi lời nhắn. |
| IsConfirmed | Bit | Không |  | Biểu thị phụ huynh có xác nhận lời nhắn hay chưa. |
| StudentInClassId | Int | Không |  | Khóa ngoại.  Tham chiếu tới bảng Student\_StudentInClass.  Biểu thị lời nhắn sẽ được gửi cho phụ huynh học sinh này. |

**Bảng - Thông tin bảng MessageToParents\_Message**

Danh sách các bảng thuộc lưu thông tin cấu hình:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| YearId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| YearName | nvarchar(20) | Không |  | Tên năm học.  Tên này là duy nhất trong dữ liệu năm học của từng trường học. |
| BeginYear | Nvarchar(100) | Không |  | Năm bắt đầu năm học. |
| EndYear | Nvarchar(300) | Không |  | Năm kết thúc năm học. |
| SchoolId | int | Không |  | Khóa ngoại.  Tham chiếu đến bảng School\_School.  Cho biết khối lớp này thuộc trường học nào. |

**Bảng - Thông tin bảng Configuration\_Year**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| TermId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| TermName | Nvarchar(50) | Không |  | Tên học kì. |

**Bảng - Thông tin bảng Configuration\_Term**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| DayInWeekId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| DayInWeekName | Nvarchar(20) | Không |  | Tên ngày trong tuần. |

**Bảng - Thông tin bảng Configuration\_DayInWeek**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| SessionId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| SessionName | Nvarchar(20) | Không |  | Tên buổi trong ngày. |

**Bảng - Thông tin bảng Configuration\_Session**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| CommentStatusId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| CommentStatusName | Nvarchar(50) | Không |  | Tên trạng thái góp ý. |

**Bảng - Thông tin bảng Configuration\_CommentStatus**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| HomePageFunctionId | Int | Không |  | Mã chức năng của trang chủ |

**Bảng - Thông tin bảng Configuration\_Parameters**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Cho phép giá trị null | Giá trị mặc định | Mô tả |
| SchoolId | Int | Không | 1 | Khóa chính.  Giá trị được sinh tự động và tăng dần. |
| SchoolName | Nvarchar(200) | Không |  | Tên trường học.  Tên này là duy nhất trong dữ liệu trường học. |
| Status | Bit | Có | True | Trạng thái kích hoạt sử dụng hệ thống của trường học. |
| TotalOfUsers | int | Không | 0 | Tổng số người dùng trong một trường học. |

**Bảng - Thông tin bảng School\_School**

CHƯƠNG 4: HIỆN THỰC HỆ THỐNG  
1. Kiến trúc hệ thống

(Bổ sung sau)  
2. Hiện thực và triển khai  
(Bổ sung sau)

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận   
- Về chương trình, về hệ thống đã được gì  
- Về kinh nghiệm về hệ thống

5.2. Hướng phát triển  
Hướng phát triển của hệ thống.

# TÀI LIỆU THAM KHẢO

# PHỤ LỤC