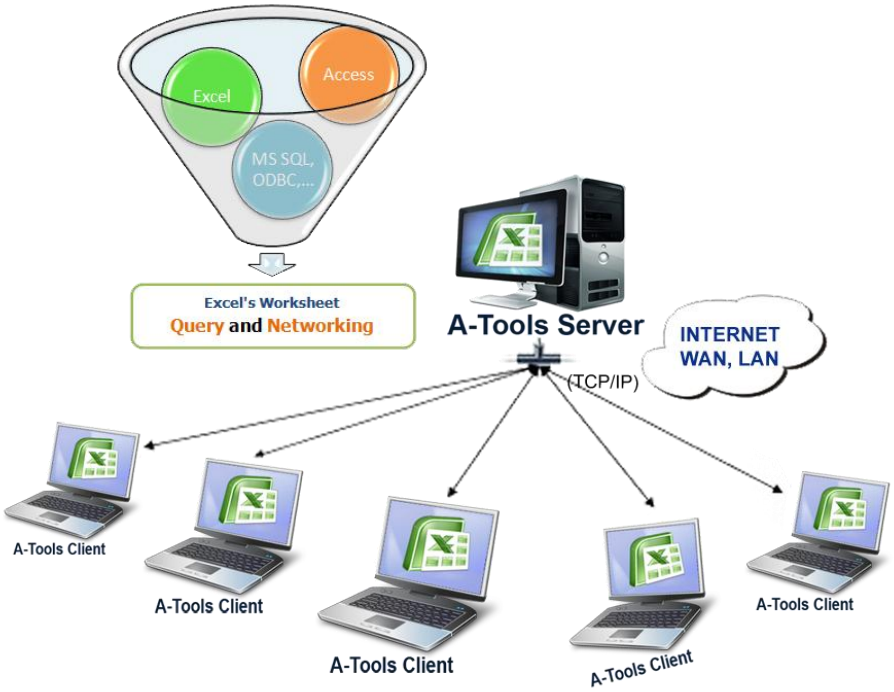
Công ty Cổ phần BLUESOFTS



Add-in A-Tools

Trích lọc dữ liệu và tạo báo cáo động trong Microsoft Excel

Tác giả: ThS. Nguyễn Duy Tuân

MỤC LỤC

Giới thiệu **7**
Cài đặt và chạy Add-in A-Tools **8**
Màn hình chức năng Add-in A-Tools **8**
Tổng quan về trích lọc dữ liệu trong Excel và Add-in A-Tools **9**
Lọc dữ liệu trong Excel 9
Add-in A-Tools là công cụ mạnh cho việc trích lọc dữ liệu và làm báo cáo động 9
Tại sao làm báo cáo trong Excel bằng Add-in A-Tools có thể chạy nhanh và động hơn bất kỳ cách lập công thức Excel thông thường nào? 11
Chuẩn hóa bảng dữ liệu nguồn trước khi sử dụng hàm BS_SQL **13**
+ Bảng dữ liệu/Table từ dòng tiêu đề đến các dòng dữ liệu không được có ô nào ở trạng thái Merge 13
+ Tiêu đề của bảng dữ liệu nên được đặt tên ngắn gọn, viết liền nhau, chỉ nên dùng các ký tự A-Z, _ , 0-9. Số luôn dứng sau chữ 13
+ Định dạng kiểu dữ liệu trong các cột của bảng trước khi nhập dữ liệu 13
+ Đặt tên vùng dữ liệu trong bảng tính Excel (worksheet) 15
Tạo công thức BS_SQL bằng "SQL Builder" **16**
Hướng dẫn sử dụng hàm BS_SQL **19**
Download/Tải tập tin ví dụ "atools_exambles.zip" 19
Mở tập tin dữ liệu mẫu "Examble.xls" 19
B1. Mở tập tin có dữ liệu ví dụ "Examble.xls" 19
B2. Mở sheet "Thực hành" để nhập công thức 20
Nguyên tắc làm việc của hàm BS_SQL trong tập tin chia sẻ qua mạng 20
Cấu trúc hàm BS_SQL 21
Phương pháp trích lọc dữ liệu bằng hàm BS_SQL 22
Tìm hiểu ngôn ngữ T-SQL 22
Sử dụng từ khóa SELECT 22
Sử dụng từ khóa WHERE 24
Sử dụng từ khóa FROM 28
Ghép dữ liệu từ nhiều bảng 28
Sử dụng từ khóa GROUP BY 34
Sử dụng từ khóa HAVING 36
Sử dụng từ khóa ORDER BY 37
Các hàm chuẩn trong ngôn ngữ T-SQL 38
Khai báo tham số OPTIONS trong hàm BS_SQL 38
Tham số INSERT 39
Tham số HR 39
Tham số NAME 40
Tham số AUTONAMES 41

Tham số DBKEY – Liên kết với CSDL bên ngoài vào bảng tính Excel.....42
Mô hình kết nối các CSDL ngoài qua DBKEY vào bảng tính Excel 44
Phương pháp làm báo cáo cho nhiều doanh nghiệp cùng mô hình 44
Tham số SERVERSOURCE: Truy vấn dữ liệu từ máy chủ 45
Tham số chạy thủ tục sự kiện Macro/VBA 45
Các hàm quan trọng phối hợp với hàm BS_SQL trong lập báo 49
Hàm RECNO: Tạo số thứ tự.....49
Hàm CELL: Lấy giá trị tại một ô hoặc NAME trong bảng tính 50
Hàm RANGE: Lấy giá trị tại một ô hoặc NAME trong bảng tính 51
Hàm DATA: Lấy giá trị tại một vị trí của mảng kết quả.....52
Hàm MIN, MAX kết hợp hàm DATA 54
Hàm GETONCE: Lấy giá trị của biểu thức một lần ở dòng đầu trong mảng 54
Hàm FORMULA: Lấy giá trị của công thức Excel 55
Hàm FIELDIF, IIF 56
Hàm VTC: Tạo báo các dạng Cross Tab.....58
Hàm SQL: Cập nhật nguồn dữ liệu có hàm BS_SQL 60
Phương pháp tạo báo cáo động trong Excel với hàm BS_SQL.....61
Bước 1: Thiết kế cấu trúc của báo cáo.....62
Tạo mục chọn mã dạng Validation - List 62
Yêu cầu với khu vực chứa dữ liệu của báo cáo 63
Bước 2: Lập công thức với hàm BS_SQL để lấy dữ liệu vào báo cáo 64
Video hướng dẫn làm báo cáo nhanh với SQL Builder trong Add-in A-Tools 65
Các hàm CSDL: Tìm kiếm và thống kê có nhiều điều kiện 65
Hàm BS_DVLOOKUP: Tìm kiếm nhiều điều kiện 65
Hàm BS_DSUM: Tính tổng với nhiều điều kiện 67
Hàm BS_DCOUNT: Đếm giá trị với nhiều điều kiện.....67
Hàm BS_DMIN: Tính giá trị nhỏ nhất với nhiều điều kiện 67
Hàm BS_DMAX: Tính giá trị lớn nhất với nhiều điều kiện.....67
Hàm BS_DAVG: Tính trung bình với nhiều điều kiện 67
Hàm BS_DVAR: Tính độ lệch chuẩn với nhiều điều kiện 67
Hàm BS_DFUNC: Tính nhiều điều kiện theo tham số là tên hàm.....67
Hàm BS_TABLE: Trích lọc dữ liệu - Cách làm khác của BS_SQL 67
Xem video hướng dẫn Add-in A-Tools.....69
Xử lý lỗi khi Add-in A-Tools không khởi động cùng Microsoft Excel.....69

DANH MỤC VÍ DỤ

Ví dụ 1: Chọn (SELECT) tất cả các cột từ (FROM) sổ KHO 22
Ví dụ 2: Chọn 10 dòng đầu của tất cả các cột từ sổ KHO 22
Ví dụ 3: Chọn danh sách SỐ chứng từ duy nhất (DISTINCT) từ sổ KHO 23
Ví dụ 4: lấy ra các cột dữ liệu MA_VLSPHH, SLG, DON_GIA, THANH_TIEN trong bảng KHO 23
Ví dụ 5: lấy ra các cột dữ liệu MA_VLSPHH, SLG, DON_GIA, THANH_TIEN trong bảng KHO 23
Ví dụ 6: Có thể thực thi các phép toán, mở thêm cột tính toán sau từ khóa SELECT 23
Ví dụ 7: Lấy ra tất cả các cột từ bảng KHO ở đó mã khách hàng (MA_KH) là KH001.....24
Ví dụ 8: Lấy ra các cột SO_CT, NGAY_CT, DIEN_GIAI, THANH_TIEN từ sổ NKC ở đó ký tự đầu của NOTK là 6.....24
Ví dụ 9: Lấy ra các cột SO_CT, NGAY_CT, DIEN_GIAI, THANH_TIEN từ sổ NKC ở đó ký tự đầu của cột NOTK là 11 và COTK có ký tự đầu là 5 24
Ví dụ 10: Chọn tất cả các cột dữ liệu (*) từ sổ KHO ở đó mã hàng (MA_VLSPHH) thuộc danh sách các mã HH001, HH002, HH003 và LOAI_PHIEU là nhập (N) 25
Ví dụ 11: Lấy ra tất cả các cột dữ liệu (*) từ sổ KHO ở đó mã hàng (MA_VLSPHH) có trong danh sách mã có loại (DMVLSPHH.LOAI) là VL và và LOAI_PHIEU là nhập (N). 25
Ví dụ 12: Lấy ra tất cả các cột dữ liệu (*) từ sổ KHO ở đó THANH_TIEN >= 19000000 ..26
Ví dụ 13: Lấy ra tất cả các cột dữ liệu (*) từ sổ KHO ở đó THANH_TIEN trong khoảng từ 19000000 đến 25000000.....27
Ví dụ 14: Lấy ra tất cả các cột dữ liệu (*) từ sổ KHO ở đó ngày chứng từ (NAY_CT) lớn hơn 24/06/2006 27
Ví dụ 15: Lấy ra các cột thông tin mã hàng, tên hàng, số lượng, đơn giá, thành tiền từ sổ KHO. 31
Ví dụ 16: Lập sổ gồm các cột: Mã khách hàng, Tên khách hàng, Số tiền. Dữ liệu được lấy từ 2 sổ KHO và DMKH. 33
Ví dụ 17: lấy ra dữ liệu gồm các cột: MA_KH, Tổng tiền tròn sổ NKC, điều kiện NOTK là 131 (tổng hợp số tiền cho từng khách hàng mà phát sinh Nợ TK =131) 34
Ví dụ 18: Lập sổ gồm các cột: MA_VLSPHH, Tổng số lượng hàng, điều kiện với LOAI_PHIEU là nhập (N) 34
Ví dụ 19: Lập sổ gồm các cột: MA_VLSPHH, Tên hàng, Tổng số lượng hàng, điều kiện LOAI_PHIEU là nhập (N) 35
Ví dụ 20: Lập danh sách gồm mã khách hàng, tên khách hàng với số tiền phát sinh lớn nhất mỗi khách hàng trong sổ KHO, với loại phiếu xuất (X). Sắp xếp theo số tiền giảm dần. 35
Ví dụ 21: Lập danh sách các mã hàng với tổng giá trị nhập về > 60000000.....36
Ví dụ 22: Lập danh sách các mã hàng có số lần nhập > 4 37
Ví dụ 23: Lấy ra các cột từ sổ KHO sắp xếp theo NGAY_CT và SO_CT 37
Ví dụ 24: Chọn tất cả các cột từ sổ KHO, cột THANH_TIEN sắp xếp giảm dần.....38

Ví dụ 25: Lập danh sách gồm mã khách hàng, tên khách hàng với số tiền phát sinh lớn nhất mỗi khách hàng trong sổ KHO, với loại phiếu xuất (X). Sắp xếp theo số tiền giảm dần. 38

Ví dụ 26: Lấy ra dữ liệu gồm các cột: MA_VLSPHH, SLG, DON_GIA, THANH_TIEN với mã hàng là HH001. Dùng thuộc tính chèn dòng. 39

Ví dụ 27: lấy ra mã hàng duy nhất trong bảng KHO. Bảng kết quả không có dòng tiêu đề 39

Ví dụ 28: lấy ra mã hàng duy nhất trong bảng KHO. Bảng kết quả không có dòng tiêu đề. Đặt tên bảng là MAHH 40

Ví dụ 29: Lấy ra các cột NGAY_CT, MA_VLSPHH, THANH_TIEN từ sổ KHO. Bảng kết quả được đặt tên là "DULIEU", các cột trong bảng kết quả được đặt tên với nhóm ký tự đầu là "DL_".....41

Ví dụ 30: Kết nối với CSDL bên ngoài. Mở tập tin "C:\A-Tools\DATA_DEMO\Access Databases\Examble.mdb" bằng MS Access.....43

Ví dụ 31: Đánh số thứ tự liên tiếp 1,2,… 49

Ví dụ 32: Đánh số thứ tự từ 1, bước nhảy là 2. Kết quả là 1,3,5,.....50

Ví dụ 33: Đánh số thứ tự liên tiếp từ 100.....50

Ví dụ 34: Hàm CELL – Điều kiện tham chiếu tới ô chứa giá trị văn bản trong bảng tính...50

Ví dụ 35: Hàm CELL – Lấy dữ liệu từ sổ KHO mà MA_VLSPHH là mã trong ô B10 VÀ số lượng > số lượng ở ô B11. Giả thiết B11 chứa giá trị 2.....51

Ví dụ 36: Hàm CELL – Lấy dữ liệu từ sổ KHO mà MA_VLSPHH là mã trong ô B10 VÀ số lượng > số lượng ở ô B11 VÀ ngày>ngày ở ô B12. Giả thiết B11 chứa giá trị 2, B12 chứa giá trị 25/06/2005.....51

Ví dụ 37: Hàm RANGE – Điều kiện tham chiếu tới ô chứa giá trị văn bản trong bảng tính51

Ví dụ 38: Lập sổ chi tiết hàng hóa gồm SO_CT, NGAY_CT, SLG, DON_GIA,THANH_TIEN, "Lũy kế". Cột "Lũy kế" được tính bởi công thức 52

Ví dụ 39: Hàm MIN, MAX kết hợp hàm DATA 54

Ví dụ 40: Lập sổ chi tiết hàng hóa gồm SO_CT, NGAY_CT, SLG, DON_GIA,THANH_TIEN, "Lũy kế". Cột "Lũy kế" được tính bởi công thức 54

Ví dụ 41: Lập sổ chi tiết hàng hóa gồm SO_CT, NGAY_CT, SLG, DON_GIA,THANH_TIEN, "Lũy kế". Cột "Lũy kế" được tính bởi công thức 55

Ví dụ 42: Lập sổ chi tiết nhập-xuất tồn hàng hóa. Mã hàng nằm ở ô C10. Giá trị tồn đầu ở F13. "Tồn" = Tồn trước + Nhập – Xuất 56

Ví dụ 43: Lập sổ tổng hợp nhập-xuất tồn hàng hóa. 57

Ví dụ 44: Lập sổ tổng hợp số lượng hàng xuất cho các khách hàng. Gồm các cột MA_VLSPHH, các cột còn lại là tổng số lượng hàng xuất cho từng mã khách hàng. 58

Ví dụ 45: Lập sổ tổng hợp số lượng hàng xuất cho các khách hàng. Gồm các cột MA_VLSPHHSTT, Mã hàng, Tên hàng, ĐVT, Tổng xuất, Tên các khách hàng với số lượng xuất mỗi mặt hàng tương ứng. 59

Ví dụ 46: Lập bảng tổng hợp số lượng các hàng hoá xuất kho theo từng ngày. Dữ liệu được lấy từ sổ KHO.....60

Ví dụ 47: Cập nhật dữ liệu nguồn trước khi truy vấn dữ liệu. 61

Ví dụ 48: Lập sổ chi tiết vật tư hàng hóa. Sổ này sẽ lọc ra các dòng dữ liệu liên quan đến loại phiếu được chọn (LOAI_PHIEU) và của mã hàng được chọn (MA_VLSPHH). 61

Ví dụ 49: Tìm một mã hàng mà trong bảng KHO có số lượng nhập kho là 10.....66

Ví dụ 50: Lấy ra các mã hàng có số lượng nhập > 5 66

Ví dụ 51: Tính tổng số lượng nhập kho mà mã hàng là HH001 67

Ví dụ 52: Tính tổng số lượng nhập kho mà mã hàng là HH001 67

Ví dụ 53: Lập một danh sách gồm các cột: MA_VLSPHH, SLG từ sổ KHO mà loại phiếu là loại nhập (LOAI_PHIEU='N'). 68

Ví dụ 54: Lập một danh sách tổng hợp SLG nhập gồm các cột: MA_VLSPHH, SLG từ sổ KHO mà loại phiếu là loại nhập (LOAI_PHIEU='N'), sắp xếp cột SLG tăng dần. 68

Giới thiệu

Add-in A-Tools là một công cụ (add-in) chạy trong môi trường Microsoft Excel. Add-in A-Tools có các chức năng chính sau:

Chia sẻ tập tin Excel qua mạng LAN, Internet cho nhiều người dùng chung: Để chia sẻ tệp dữ liệu Excel cho nhiều người dùng trong mạng, nếu không có công cụ thì chỉ có một phương pháp là share thư mục chứa file Excel đó (cách của Excel). Theo cách này, các máy trong mạng đều có thể truy cập vào copy và thậm chí xoá hẳn tệp dữ liệu này. Người quản lý tập tin Excel này chỉ có thể gửi trọn cả tập tin mà rất khó che dấu những sheets hay vùng dữ liệu riêng của mình. Các vấn đề khác như cho người dùng truy cập vào vùng dữ liệu nhưng có các quyền: chỉ đọc, soạn thảo, giới hạn bao nhiêu người được kết nối,.... Excel không thể làm được.

Với Add-in A-Tools cho phép soạn thảo và quản trị dữ liệu Excel qua mạng. Công cụ này cho phép người dùng quản trị dữ liệu Excel qua mạng một cách mạnh mẽ, có thể kết nối các bảng tính, vùng dữ liệu qua TCP/IP (không phải share full thư mục), quản trị các kết nối tới vùng dữ liệu, phân quyền cho từng nhóm người dùng, khi dữ liệu tại một máy thay đổi thì các máy khác trong mạng được cập nhật ngay lập tức (sau khoảng ~ 1 mili giây).

Tìm hiểu chi tiết chức năng này đọc tài liệu:
C:\A-Tools\Documents\A-Tools-Help.pdf

Trích lọc dữ liệu và tạo báo cáo trong bảng tính Excel: Tạo bảng dữ liệu động bởi việc sử dụng các hàm với ngôn ngữ truy vấn cơ sở dữ liệu (CSDL) T-SQL. Cho phép kết nối với các CSDL bên trong và bên ngoài như Excel, Access, Foxpro, MS SQL, MySQL, Firebird,… vào bảng tính Excel. Với công cụ SQL Builder và hàm BS_SQL sẽ giúp người dùng tạo công thức truy vấn dữ liệu một cách dễ dàng và hiệu quả. Người dùng có thể tạo ra những báo cáo trong bảng tính Excel có độ phức tạp cao, được lọc theo nhiều điều kiện, liên kết với các nguồn dữ liệu khác nhau, đảm bảo độ chính xác, ổn định và tốc độ xử lý nhanh.

Lập trình cùng Add-in A-Tools: Ngoài giao diện người dùng, Add-in A-Tools còn hỗ trợ các hàm API và các đối tượng COM như **BSNetwork, BSFormulaArray** để những người phát triển ứng dụng lập trình kết hợp cùng Add-in A-Tools nhằm khai thác tối đa năng lực của Add-in A-Tools, tăng sức mạnh cho ứng dụng của mình.
Nếu bạn muốn lập trình VBA cùng A-Tools hãy đọc tài liệu tại file "A-Tools Programming.pdf"

Các chức năng phụ:

- Tạo công thức thống kê kết hợp với nhiều điều kiện, công thức mảng.
- Chuyển đổi mã văn bản từ TCVN3, VNI sang UNICODE và ngược lại.
- Đọc số thành chữ tiếng Việt, tiếng Anh
- …

Cài đặt và chạy Add-in A-Tools

Người dùng cần download/tải Add-in A-Tools theo địa chỉ dưới đây:

- **A-Tools Pro** – Phiên bản thương mại:
  http://www.bluesofts.net/Products/AddIns/A-Tools/A-Tools_Pro.exe
- **A-Tools Free** – Phiên bản miễn phí:
  http://www.bluesofts.net/Products/AddIns/A-Tools/A-Tools_Free.exe

Sau khi tải bộ cài đặt A-Tools về máy, Người dùng nhấp đúp chuột vào tệp tin và cài đặt bình thường.

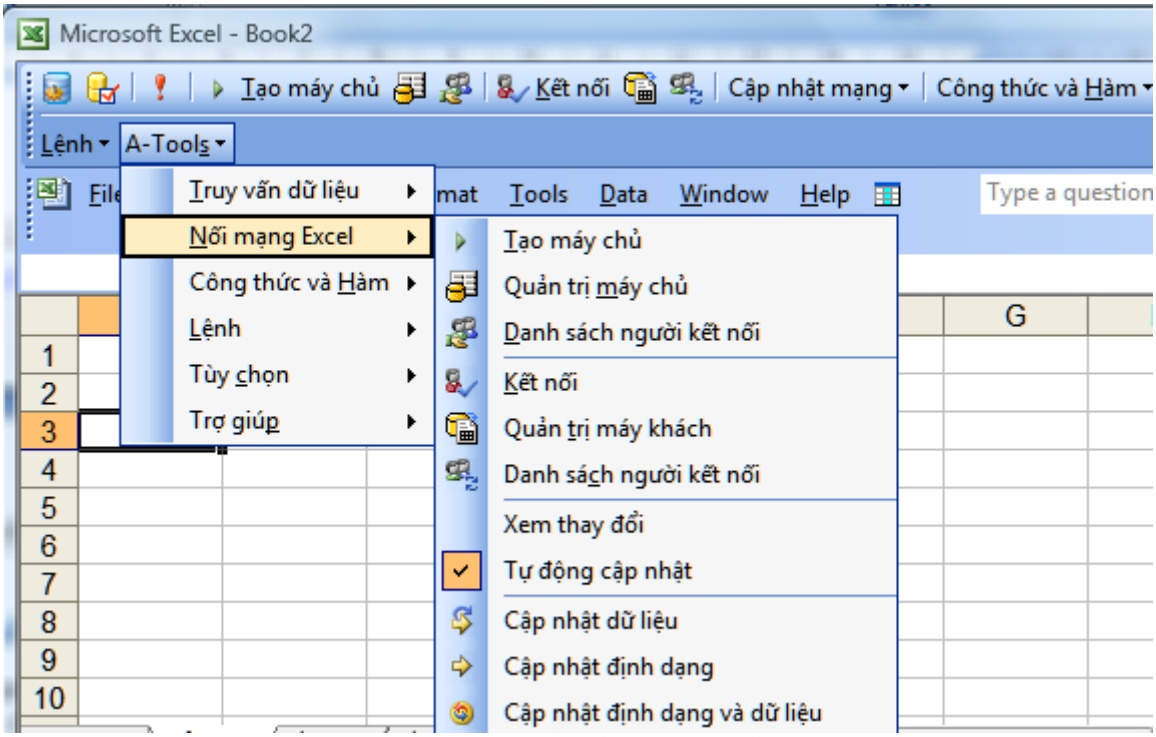
Lưu ý: người dùng phải đăng nhập Windows với quyền quản trị (Administrator). Khi cài đặt thành công, tất cả các loại user đều chạy được A-Tools.

Để chạy A-Tools, người dùng mở Microsoft Excel, khi chương trình mở xong, người dùng sẽ nhìn thấy menu A-Tools nằm gần cuối các menu ngang của Microsoft Excel.

Màn hình chức năng Add-in A-Tools

Mở Microsoft Excel 2007. Chọn menu A-Tools người dùng sẽ thấy các chức năng của A-Tools như hình dưới đây.

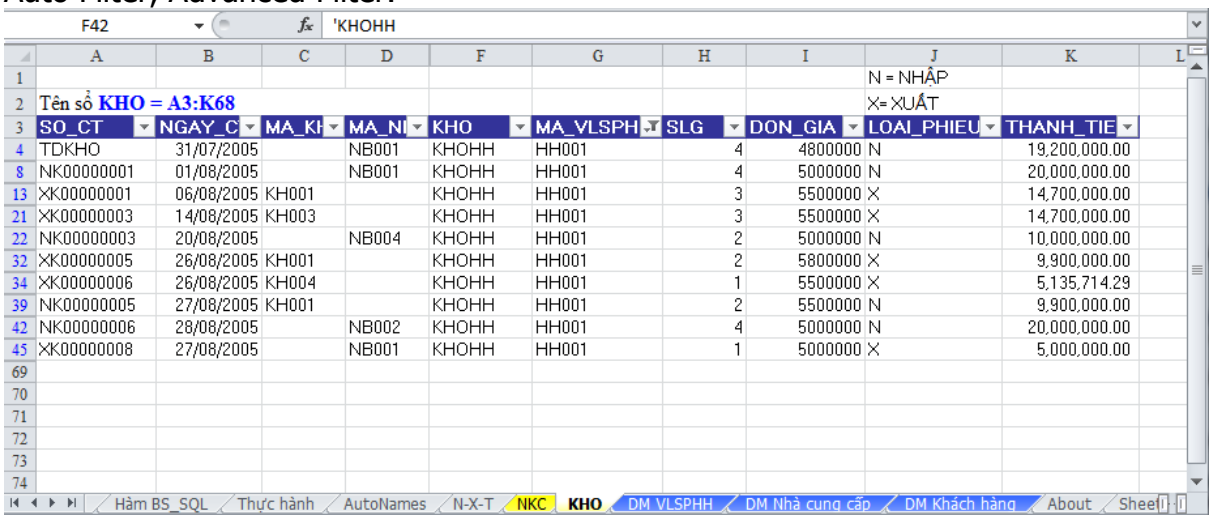
Giao diện trên Microsoft Excel 2003 hoặc các phiên bản thấp hơn



Tổng quan về trích lọc dữ liệu trong Excel và Add-in A-Tools

Lọc dữ liệu trong Excel

Để thực hiện các việc trích lọc dữ liệu ta có thể dùng các chức năng sẵn có như: Auto Filter, Advanced Filter.



SO_CT	NGAY_C	MA_KH	MA_NI	KHO	MA_VLSPH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIE
TDKHO	31/07/2005		NB001	KHOHH	HH001	4	4800000	N	19,200,000.00
NK00000001	01/08/2005		NB001	KHOHH	HH001	4	5000000	N	20,000,000.00
XK00000001	06/08/2005	KH001		KHOHH	HH001	3	5500000	X	14,700,000.00
XK00000003	14/08/2005	KH003		KHOHH	HH001	3	5500000	X	14,700,000.00
NK00000003	20/08/2005		NB004	KHOHH	HH001	2	5000000	N	10,000,000.00
XK00000005	26/08/2005	KH001		KHOHH	HH001	2	5800000	X	9,900,000.00
XK00000006	26/08/2005	KH004		KHOHH	HH001	1	5500000	X	5,135,714.29
NK00000005	27/08/2005	KH001		KHOHH	HH001	2	5500000	N	9,900,000.00
NK00000006	28/08/2005		NB002	KHOHH	HH001	4	5000000	N	20,000,000.00
XK00000008	27/08/2005		NB001	KHOHH	HH001	1	5000000	X	5,000,000.00

Auto Filter: việc trích lọc dữ liệu được thực hiện ngay trong chính sheet/bảng dữ liệu nguồn. Công việc Filter này nhằm giúp người sử dụng nắm bắt thông tin nhanh để có cơ sở đối chiếu và so sánh với các dữ liệu khác.

Advanced Filter: Lọc dữ liệu dựa vào vùng điều kiện - Criteria Range. Điều kiện lập trong vùng này khá phong phú. Dữ liệu trích lọc được có thể đưa sang một địa chỉ khác.

Điểm chung giữa Auto Filter và Advanced Filter:

+ Dữ liệu lọc ra vẫn mang nguyên cả cấu trúc bảng, tức bảng gốc có 10 cột thì bảng kết quả lọc được cũng là 10 cột.

+ Dữ liệu lọc ra chỉ có thể đóng vai trò trung gian làm cơ sở so sánh, đối chiếu với dữ liệu khác, người dùng khó có thể tạo ra một báo cáo từ Auto Filter hoặc Advanced Filter. Trừ trường hợp Filter được dùng trên một báo cáo đã có và có cột công thức làm tiêu thức lọc để ẩn các dòng dữ liệu không cần thiết.

Add-in A-Tools là công cụ mạnh cho việc trích lọc dữ liệu và làm báo cáo động

Để hiểu qua về trích lọc dữ liệu trong Add-in A-Tools bạn hãy tạo một công thức đơn giản theo các bước sau:

B1. Mở tập tin có dữ liệu ví dụ tên là "Examble.xls" tại đường dẫn "C:\A-Tools\DATA_DEMO\"

Hoặc download từ địa chỉ

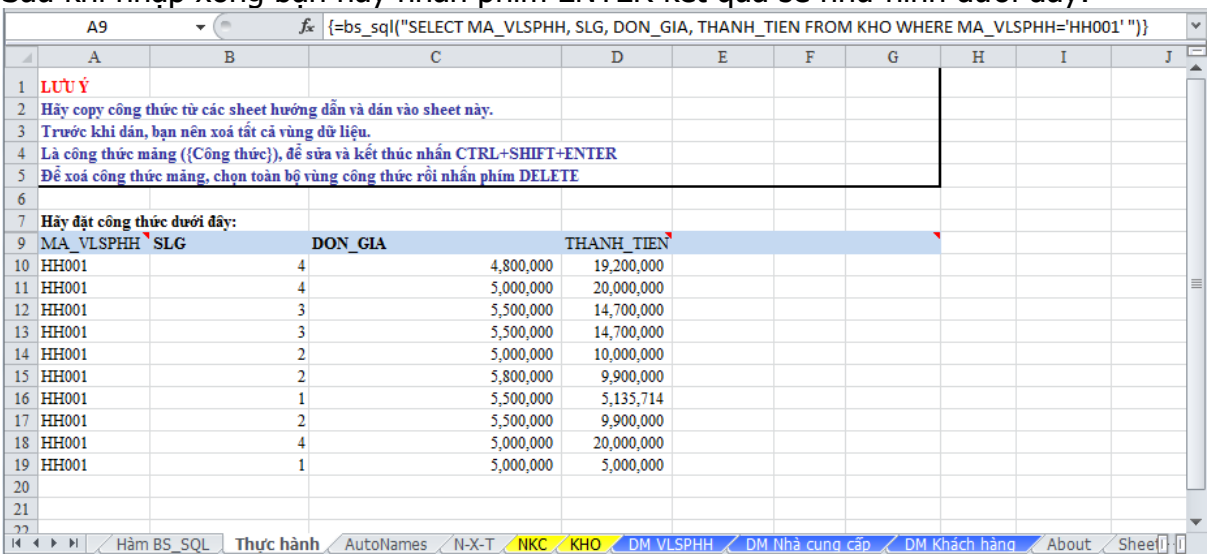
http://bluesofts.net/Products/AddIns/A-Tools/Help/DEMO/Examble.xls

B2. Sheet "KHO" chứa các phiếu nhập, phiếu xuất, vùng A3:K68 đã được đặt tên là "KHO". **Mở sheet "Thực hành"** để nhập công thức

B3. Lập công thức lọc ra các dòng dữ liệu với các cột thông tin MA_VLSPHH, SLG, DON_GIA, THANH_TIEN trong vùng dữ liệu "KHO", điều kiện lọc là mã hàng là HH001 (WHERE MA_VLSPHH= 'HH001') . Công thức như sau

```
=bs_sql("SELECT MA_VLSPHH, SLG, DON_GIA, THANH_TIEN FROM KHO WHERE MA_VLSPHH='HH001' ")
```

Sau khi nhập xong bạn hãy nhấn phím ENTER kết quả sẽ như hình dưới đây:



Lưu ý:

- ➔ Khi sửa công thức phải đảm bảo chỉ có một ô được chọn.
- ➔ Nếu sửa lại công thức, khi kết thúc phải nhấn CTRL+SHIFT+ENTER vì BS_SQL là công thức mảng.
- ➔ Nếu không muốn thực hiện nhấn phím ESC để hủy không sửa.
- ➔ Muốn xóa thì chọn cả bảng rồi xóa.

Với công thức trên điều kiện lọc nằm sau từ khóa WHERE, nếu thay HH001 thành HH002 thì kết quả lọc ra là của mã hàng HH002. Phần sau tài liệu này sẽ mô tả chi tiết cách lập công thức trên.

+ Add-in A-Tools cung cấp hàm BS_SQL sử dụng ngôn ngữ T-SQL (SELECT * FROM…WHERE…) để truy vấn dữ liệu. Ngôn ngữ T-SQL ngày nay được phổ biến cho các phần mềm quản trị CSDL, nó cho phép trích lọc dữ liệu với khối lượng lớn mà vẫn đảm bảo tốc độ nhanh. Các điều kiện lọc được thực hiện rất linh hoạt. Người sử dụng có thể học nó một cách dễ dàng.

+ Báo cáo tạo ra thường là sự ghép nối dữ liệu từ một hay nhiều bảng dữ liệu khác. Ví dụ báo cáo nhập xuất tồn được lấy ra từ hai sổ: sổ kho (nơi ghi nhận các phiếu nhập phiếu xuất, mã hàng mã nhà cung cấp, mã khách hàng,…) và sổ danh mục hàng hóa. Ngôn ngữ T-SQL cho phép kết nối các bảng dữ liệu lại với nhau để chọn ra những trường thông tin nào tùy ý.

+ Add-in A-Tools cung cấp các hàm hỗ trợ cho hàm BS_SQL và ngôn ngữ T-SQL để có thể tạo ra báo cáo phức tạp, dữ liệu kết nối linh hoạt. Các hàm như: RECNO() – Tạo số thứ tự; CELL(), RANGE để lấy giá trị trên bảng tính; FORMULA() – Tính toán công thức trong câu lệnh SELECT; VTC() – Làm báo cáo theo mô hình CROSS; DATA() giúp cho việc làm báo cáo với việc tính chỉ tiêu lũy kế dễ dàng;,…. Chính các hàm bổ sung này cùng với ngôn ngữ T-SQL mới là sức mạnh thực sự của A-Tools trong việc trích lọc dữ liệu và làm báo cáo động.

+ Add-in A-Tools với hàm BS_SQL sử dụng DBKEY cho phép kết nối với nhiều loại CSDL bên ngoài như MS Access, SQL Server, MySQL, Foxpro, Excel,…. Để liên kết dữ liệu, làm báo cáo trong bảng tính Excel. Khi dữ liệu trong các CSDL thay đổi thì báo cáo trên Excel cũng thay đổi (khi được cập nhật). Với tính năng này người dùng có thể tạo ra báo cáo trên Excel mà dữ liệu lấy từ phần mềm nào đó đang chạy trong

máy tính hoặc trong mạng. Các phần mềm không làm hết cho ta các loại báo cáo, vậy ta có thể tự tạo ra chúng dễ dàng bởi Add-in A-Tools.
+ Hàm BS_SQL kết hợp với chức năng chia sẻ tập tin Excel qua mạng LAN, Internet cho phép báo cáo trên máy khách mà dữ liệu gốc nằm ở máy chủ. Với phương pháp thiết kế này thì đảm bảo tập tin làm báo cáo nhẹ vì chỉ có công thức, an toàn dữ liệu vì dữ liệu góc nằm tại máy chủ.

Tại sao làm báo cáo trong Excel bằng Add-in A-Tools có thể chạy nhanh và động hơn bất kỳ cách lập công thức Excel thông thường nào?

+ Với cách làm Excel thông thường

A86 | =IF(NKC!A79>=NGAYDK,IF(NKC!E79='SO CT'!$C$5,IF(OR(NKC!L79='SO CT'!$C$4,NKC!M79

SỔ KẾ TOÁN CHI TIẾT

Từ ngày 01/01/12 đến 31/12/12

Tài khoản: 131 Phải trả cho người bán

Chi tiết: KH002

Đơn vị tính: đồng

	A	B	C	D	E	F	G	H	I	J
7	Ngày ghi sổ	Chứng từ		Diễn giải	TK đối ứng	Số phát sinh		Số dư		
8		Số	Ngày							
9						Nợ	Có	Nợ	Có	LOC
10				Tồn đầu kỳ				550,000	0	1
86	00/01/1900		0			0	0			0
87	00/01/1900		0			0	0			0
88	00/01/1900		0			0	0			0
89	25/06/2002	0	37432	Bán hàng	511	2,400,000	0			1
90	25/06/2002	0	37432	Thuế	333	240,000	0			1
91	00/01/1900		0			0	0			0
92	00/01/1900		0			0	0			0
93	00/01/1900		0			0	0			0
94	00/01/1900		0			0	0			0
95	00/01/1900		0			0	0			0
96	26/06/2002	0	37433	Khách hàng thanh toán	111	0	650,000			1
97	00/01/1900		0			0	0			0

Cần phải sử dụng các hàm IF, Vlookup, SumIf, SumProduct,OffSet, Index,… Công thức được lập ra để lấy dữ liệu theo điều kiện của báo cáo. Trong báo cáo công thức được copy thừa ra nhiều dòng, cho nhiều mã đối tượng để bao quát các trường hợp. Khi cần thu gọn lại dùng Auto Filter để lọc ra cái cần xem và in. Chính phương pháp này có nhược điểm rất lớn là TẠO CÔNG THỨC THỪA rất nhiều, nó ép cho Excel phải tính toán những cái lẽ ra không phải tính. Ví dụ trong danh mục hàng có 100 mã hàng, nhưng trong sổ kho nhập, xuất mới chỉ 10 mã nhưng công thức vẫn phải copy cho cả 100 mã – thừa. Hay theo như hình của "Sổ kế toán chi tiết" bên trên. Nếu số NKC (dữ liệu nguồn) mà có 20.000 dòng thì trong sổ này cũng phải lập công thức cho 20.000 dòng, trong khi với dữ liệu cần hiển thị cho TK 131, mã khách hàng là "KH002 thì số dòng có thể chỉ là 20 dòng. Vậy rõ ràng chúng ta đã làm cho Excel phải luôn tính thừa những đối tượng không cần thiết.
Vấn đề thứ hai là vì đặc tính công thức Excel, khi dữ liệu nguồn thay đổi thì kết quả các công thức liên kết đến nó cũng thay đổi. Hiểu một cách khác là toàn bộ các công thức đang tham chiếu tới vùng dữ liệu thay đổi bị tính toán lại – Đây chính là lý

do bảng tính chạy rất chậm khi người dùng nhập dữ liệu trong các vùng có công thức tham chiếu đến.

+ Với cách làm bởi BS_SQL trong Add-in A-Tools

A15 | {=bs_sql(A11,"HR=NO;INSERT=YES")}

CÔNG TY CỔ PHẦN BLUESOFTS — **Mẫu số xx-DN**

32/106 Ngõ 79 đường Cầu Giấy - Yên Hòa - Cầu Giấy - Hà Nội — (Ban hành theo QĐ số 48/QĐ-BTC ngày 14/09/2006 của Bộ trưởng BTC)

THẺ KHO

Từ ngày 01/01/07 đến 31/12/07

Tại kho: KHOHH Kho hàng hóa
Tên hàng : HH001 **Máy tính ĐNA L1**
Đơn vị tính : Chiếc

STT	Chứng từ		Diễn giải	Số lượng			Ghi chú
	Số hiệu	Ngày tháng		Nhập	Xuất	Tồn	
			Số dư đầu kỳ			**10**	
1	NK00001	01/01/2007	Nhập kho	6		16	
2	XK00001	04/01/2007	Xuất kho		4	12	
3	XK00004	10/01/2007	Xuất kho		1	11	
4	XK00005	20/01/2007	Xuất kho		1	10	
			Cộng phát sinh	**6**	**6**		
			Số dư cuối kỳ			**10**	

Tự co giãn dòng và công

Hà Nội, ngày 18 tháng 02 năm 2014

Người lập	**Kế toán trưởng**	**Giám đốc**
(Ký, họ tên)	(Ký, họ tên)	(Ký, họ tên, đóng dấu)

Việc trích lọc dữ liệu chỉ cần một hàm chủ đạo là BS_SQL, hàm này sẽ trả về cả bảng dữ liệu cần cho báo cáo sau khi thực thi câu lệch SQL. Hàm BS_SQL chỉ tính toán, lọc trả về dữ liệu theo đúng điều kiện lọc của báo cáo. Không tính thừa dữ liệu như phương pháp lập công thức Excel bên trên. Các dòng trong báo cáo tự co giãn để phù hợp khi báo cáo tăng hay giảm dòng dữ liệu (nhờ khai báo INSERT=YES trong tham số OPTIONS của hàm BS_SQL) . Các dữ liệu dưới chân báo cáo – Footer được đẩy theo linh hoạt. Báo cáo chỉ cập nhật lại khi thực hiện lệnh cập nhật hoặc khi mở tập tin. Việc lấy ra danh sách mã duy nhất: mã hàng, mã nhà cung cấp, khách hàng,… trong sổ phát sinh được thực hiện dễ dàng, kết quả trích lọc được đặt NAME tự động theo công thức BS_SQL. Các NAME này được đưa vào làm Validation hay các công thức khác của Excel sẽ tạo cho bạn một báo cáo chuyên nghiệp và linh động. Add-in A-Tools được lập trình trong Delphi với các thuật toán tối ưu tiết kiệm bộ nhớ, được biên dịch ra mã máy nên công thức chạy rất nhanh.

- ➔ Phương pháp làm báo cáo của Excel là phải làm NHIỀU công thức gộp lại rồi copy.
- ➔ Phương thức làm báo của Add-in A-Tools là chỉ cần làm MỘT công thức có hàm BS_SQL tạo ra một báo cáo. Vì một vài lý do khác nữa mà báo cáo tạo bởi hàm BS_SQL trong Add-in A-Tools làm dễ, chạy nhanh và linh hoạt hơn cách làm Excel thông thường.

Chuẩn hóa bảng dữ liệu nguồn trước khi sử dụng hàm BS_SQL

+ Bảng dữ liệu/Table từ dòng tiêu đề đến các dòng dữ liệu không được có ô nào ở trạng thái Merge

Là một bảng dữ liệu được định nghĩa bởi cột và dòng, giao của cột và dòng là ô. Việc Merge nhiều ô thành một ô sẽ làm việc truy vấn dữ liệu gặp lỗi.

Trong bảng ví dụ dưới đây ô B10 bị lỗi vì có trạng thái Merge.

	A	B	C	D
1				
2	Tên sổ KHO = A3:K68			
3	SO_CT	NGAY_CT	MA_KH	MA_NB
4	TDKHO	31/07/2005		NB001
5	TDKHO	31/07/2005		NB002
6	TDKHO	31/07/2005		NB004
7	TDKHO	31/07/2005		NB004
8	NK00000001	01/08/2005		NB001
9	NK00000001	01/08/2005		NB001
10	NK00000002	03/08/2005		NB002
11	NK00000002			NB002
12	NK00000002			NB002
13	XK00000001	06/08/2005	KH001	
14	XK00000001	06/08/2005	KH001	

+ Tiêu đề của bảng dữ liệu nên được đặt tên ngắn gọn, viết liền nhau, chỉ nên dùng các ký tự A-Z, _ , 0-9. Số luôn đứng sau chữ

SỐ CT	NGÀY CT	MÃ KH	MÃ NB	MÃ NV	KHO	MÃ VLSPHH	SỐ LƯỢNG	ĐƠN GIÁ	LOẠI PHIẾU	THÀNH TIỀN
TDKHO	31/07/2005		NB001		KHOHH	HH001	4	4800000	N	19,200,000.00
TDKHO	31/07/2005		NB002		KHOHH	HH002	6	4500000	N	27,000,000.00
TDKHO	31/07/2005		NB004		KHOHH	HH003	6	2800000	N	16,800,000.00
TDKHO	31/07/2005		NB004		KHOHH	HH004	3	4200000	N	12,600,000.00

Với bảng dữ liệu trên, công thức phải làm như sau

```
=bs_sql("select [SỐ CT], [MÃ VLSPHH], [SỐ LƯỢNG],[ĐƠN GIÁ],[THÀNH TIỀN] from [KHO$A3:K10]")
```

Tiêu đề các cột có dấu dẫn đến khi làm công thức dễ bị sai do nhầm lẫn, lập công thức mất thời gian vì việc soạn thảo nhiều.

Tiêu đề các cột của bảng dữ liệu nên đặt tên lại như sau

SO_CT	NGAY_CT	MA_KH	MA_NB	MA_NV	KHO	MA_VLSPHH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
TDKHO	31/07/2005		NB001		KHOHH	HH001	4	4800000	N	19,200,000.00
TDKHO	31/07/2005		NB002		KHOHH	HH002	6	4500000	N	27,000,000.00
TDKHO	31/07/2005		NB004		KHOHH	HH003	6	2800000	N	16,800,000.00
TDKHO	31/07/2005		NB004		KHOHH	HH004	3	4200000	N	12,600,000.00

Khi đó công thức là

```
=bs_sql("SELECT SO_CT, MA_VLSPHH, SLG,DON_GIA ,THANH_TIEN FROM [KHO$A3:K10]")
```

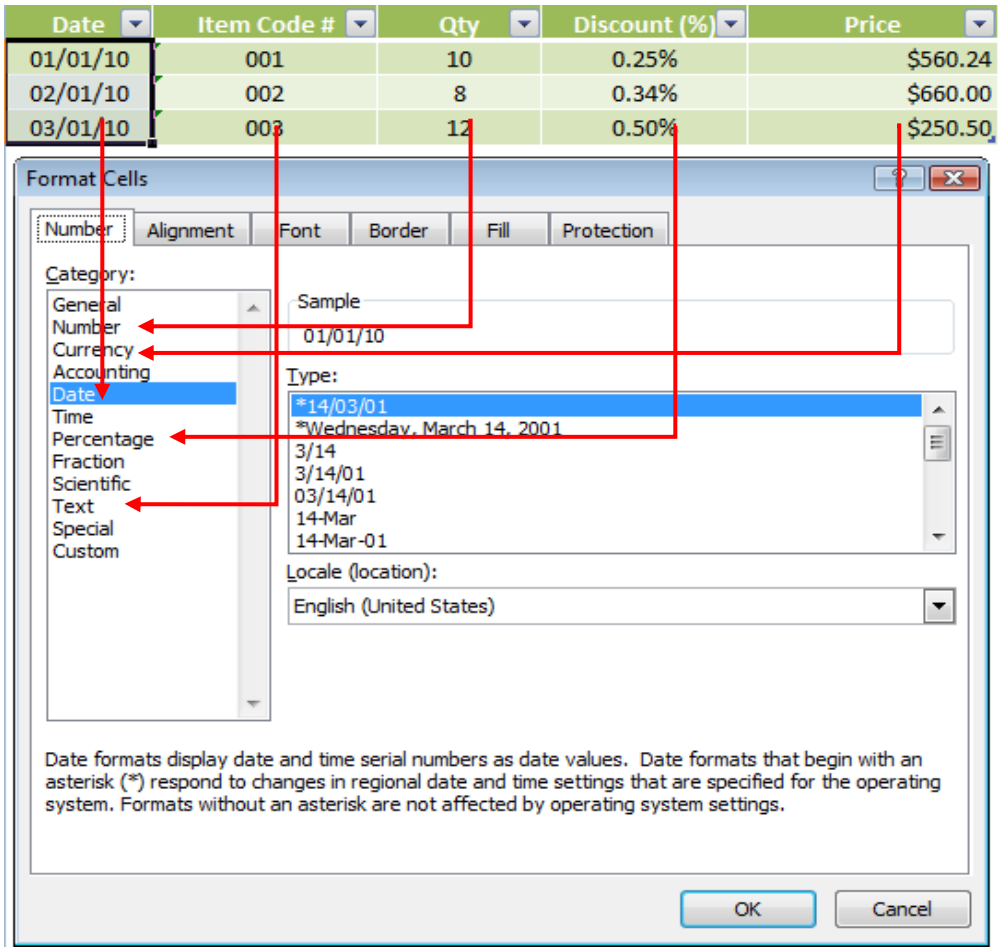
Không cần phải dùng ký tự bao [ ], soạn thảo chữ không dấu sẽ nhanh và giảm lỗi soạn thảo hơn.

+ Định dạng kiểu dữ liệu trong các cột của bảng trước khi nhập dữ liệu

Trong bảng tính Excel, các ô ngầm định đặt kiểu General, kiểu dữ liệu được xác định khi dữ liệu được nhập vào. Trong một hoàn cảnh nào đó, dữ liệu nhập vào

khó để cho A-Tools hay đối tượng truy vấn dữ liệu ADO xác định rõ kiểu dữ liệu. Như là số CMND, Mã hàng,… Giả xử ta nhập giá trị là 1111 Excel lại hiểu là kiểu số (Number), mục đích của người sử dụng phải là kiểu TEXT và trong công thức dùng phép so sánh các giá trị TEXT dẫn đến sai. Ví dụ MA_VLSPHH = '1111' sẽ sai vì Excel đang coi MA_VLSPHH là số - Number. Vậy cần phải định dạng cột MA_VLSPHH về dạng TEXT trước sau đó mới nhập 1111 lúc này công thức so sánh MA_VLSPHH = '1111' là đúng vì cột MA_VLSPHH được Excel hiểu là TEXT. *Nếu giá trị 1111 nhập trước sau đó mới định dạng thì không có ý nghĩa. Vẫn cần phải nhập lại sau khi định dạng. Nếu không muốn nhập lại, bạn chạy chức năng "**Số thành văn bản**" trong menu **A-Tools**.*

Hình dưới đây là gợi ý về định dạng các cột trong bảng trước khi đưa vào nhập dữ liệu.



Trong bảng dữ liệu nguồn nên có dòng đầu tiên để nhập giá trị giả định mà các giá trị này không ảnh hưởng tới kết quả của báo cáo.

Các cột là dạng văn bản (TEXT) nhập giá trị ` (nháy đơn). Các cột có các giá trị ngày tháng, số lượng, số tiền, phần trăm, tỷ lệ (số) thì nhập số **0**.

Việc nhập giá trị giả định như trên giúp cho A-Tools hay đối tượng ADO hiểu được kiểu dữ liệu của các cột dữ liệu trong bảng được rõ ràng hơn.

A4 fx '

	A	B	C	D	E	F	G	H	I	J	K
3	SO_CT	NGAY_CT	MA_KH	MA_NB	MA_NV	KHO	MA_VLSPHH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
4		00/01/1900						0	0		0.00
5	TDKHO	31/07/2005		NB001		KHOHH	HH001	4	4800000	N	19,200,000.00
6	TDKHO	31/07/2005		NB002		KHOHH	HH002	6	4500000	N	27,000,000.00
7	TDKHO	31/07/2005		NB004		KHOHH	HH003	6	2800000	N	16,800,000.00
8	TDKHO	31/07/2005		NB004		KHOHH	HH004	3	4200000	N	12,600,000.00

+ Đặt tên vùng dữ liệu trong bảng tính Excel (worksheet)

Trong bảng tính có nhiều vùng dữ liệu, chỉ một khu vực nào đó đảm bảo đó là bảng dữ liệu – Table. Vậy để A-Tools hiểu khu vực đó là bảng thì người dùng cần đặt tên – NAME cho vùng dữ liệu này là bảng.

Ví dụ lọc lấy các dòng dữ liệu trong sheet "KHO", vùng dữ liệu A3:K68.

Đặt công thức:

```
=BS_SQL("SELECT * FROM [KHO$A3:K68]")
```

Nếu vùng dữ liệu được đặt tên là **KHO** (=KHO!A3:K68) thì công thức như sau.

```
=BS_SQL("SELECT * FROM KHO")
```

Việc đặt tên - NAME đem lại hiệu quả sử dụng rất cao trong các công thức Excel đồng thời đảm bảo cho A-Tools biết đó là khu vực bảng. Với A-Tools, tên của một vùng KHO được coi như tên của một bảng (table) trong CSDL Excel.

Cách tạo tên cho vùng dữ liệu

Người dùng làm theo trình tự sau:

+ **1:** Chọn vùng dữ liệu. Vùng dữ liệu phải từ dòng tiêu đề của bảng tới dòng cuối cùng. Ví dụ trong sheet "KHO", bôi đen từ A3:K68

+ **2:** Chọn menu "Formula" (Với Excel 2007 hoặc cao hơn)

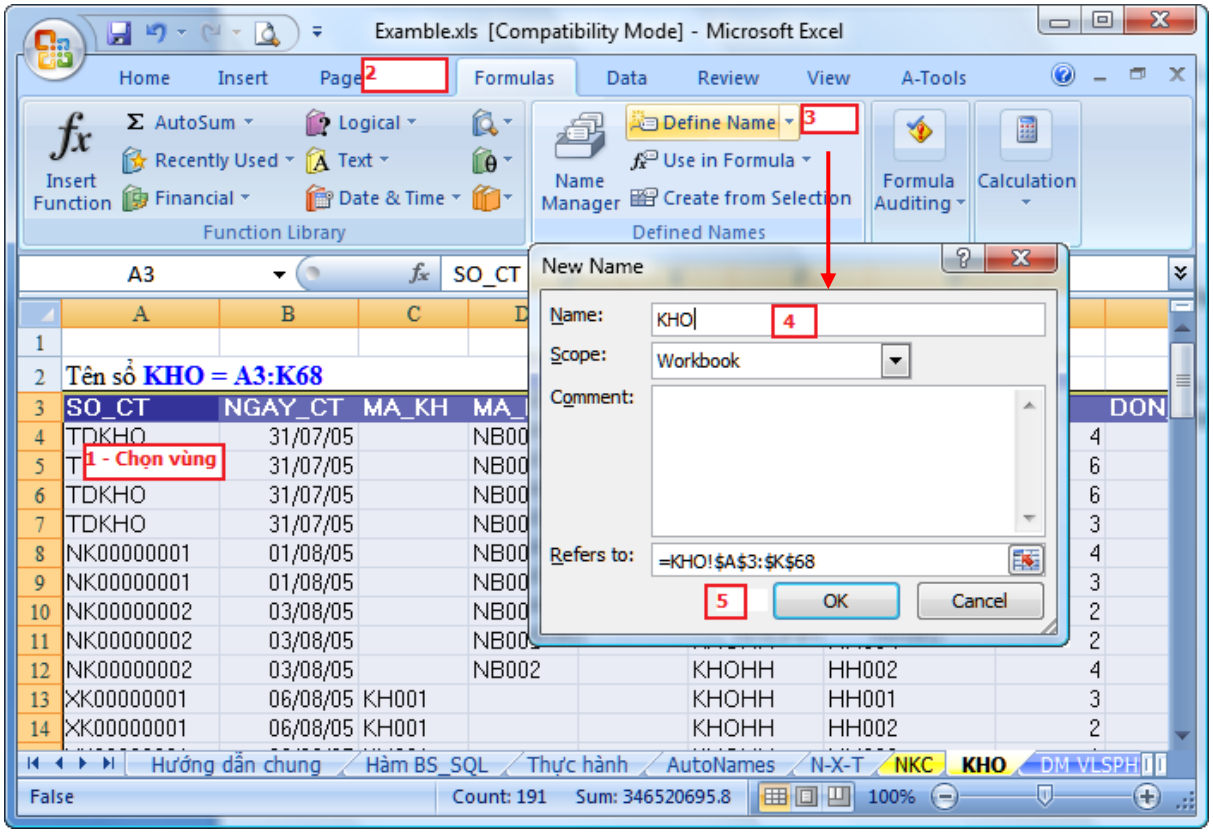
+ **3:** Chọn menu "Insert->Name" (Với Excel 2003 hoặc thấp hơn)

+ **4:** Chọn "Define Name". Đặt tên tại mục "Name:"

+ **5:** Chọn [Ok] để đồng ý

Ở bước 2,3 phím tắt là CTRL+F3

Ta được màn hình đặt tên - NAME dưới đây



+ Name: đặt tên vùng **KHO**
+ Refers to: địa chỉ tham chiếu. Excel tự điền từ vùng ta chọn bước 1.

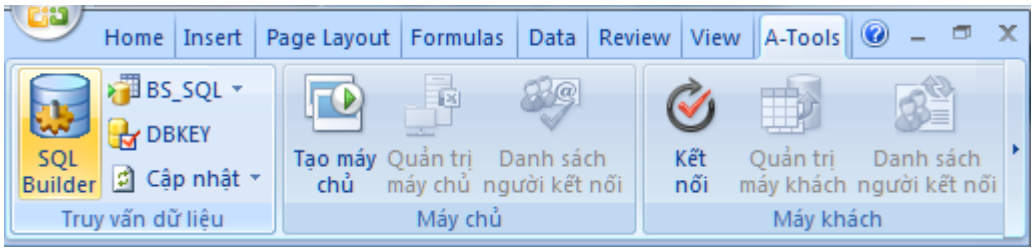
Sau khi nhập tên xong, nhấn nút "Ok" để hoàn tất. Nếu máy báo tên đã tồn tại thì vào "Name Manager" chọn Edit để sửa lại.

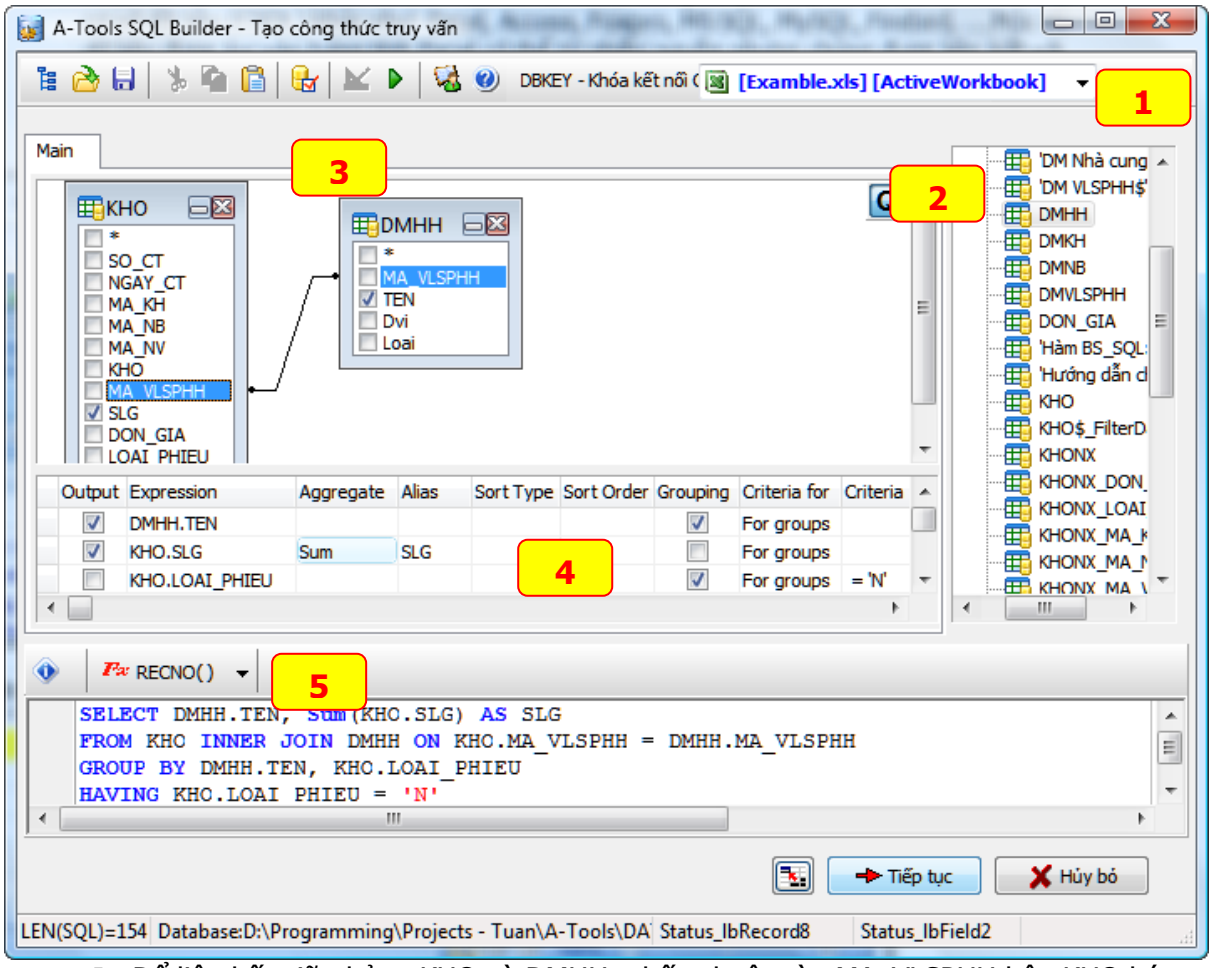
➔ Lần đầu đặt tên các bảng dữ liệu bạn nên lưu lại và đóng tập tin này sau đó mở lại thì Excel hiểu cấu trúc các vùng dữ liệu được đặt tên theo dạng bảng/table..

Tạo công thức BS_SQL bằng "SQL Builder"

Từ menu "A-Tools" chọn "SQL Builder"

Với Excel 2003 vào menu "A-Tools"->"Truy vấn dữ liệu" chọn "SQL Builder"





➔ Để liên kết giữa bảng KHO và DMHH, nhấp chuột vào MA_VLSPHH bên KHO kéo sang MA_VLSPHH của bảng DMHH rồi nhả chuột.

Các khu vực đánh số được giải thích dưới đây:

(1) DBKEY – Khóa kết nối: chọn CSDL hoặc DBKEY (mã kết nối với CSDL bên ngoài). Để kết nối với CSDL bên ngoài, người dùng phải tạo DBKEY trước đó, cách tạo sẽ hướng dẫn ở trang sau. Ngầm định A-Tools kết nối tới tập tin Excel đang mở hiện tại. Ví dụ nếu bạn đang mở file "Examble.xls" thì mục DBKEY sẽ hiện "Examble.xls" có nghĩa là A-Tools đang kết nối tới tập tin"Examble.xls".

(2) Danh sách các tên vùng hoặc tên bảng (table, query) dữ liệu trong CSDL chọn ở (1). Người dùng hãy nhấp đúp chuột hoặc kéo tên vùng dữ liệu từ (2) sang (3) để làm nguồn dữ liệu.

(3) Nối các khóa quan hệ giữa các bảng (table) nếu có (trong hình vẽ là "MA_VLSPHH"). Chọn các trường dữ liệu để hiển thị và làm điều kiện.

(4) + Output: nếu chọn, trường dữ liệu đó sẽ hiển thị

+ Expression: tên trường dữ liệu hoặc biểu thức tính

+ Aggregate: chọn tên hàm thống kê: Sum; Count; Min; Max;…

+ Alias: đặt tên cho trường hoặc biểu thức ở "Expression"

+ Sort Type: chọn kiểu sắp xếp tăng (Ascending) hay giảm (Descending) dần

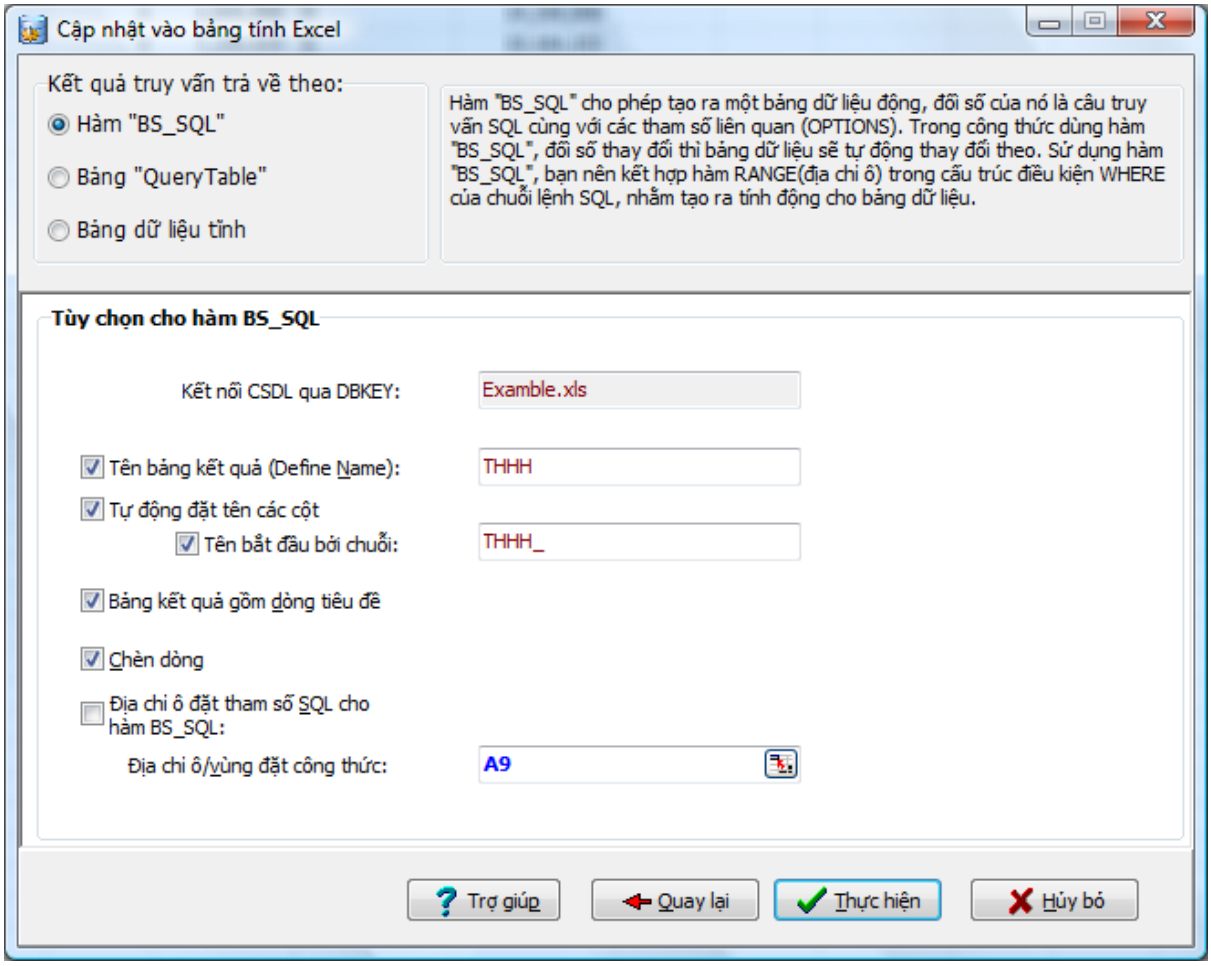
+ Sort Order: đặt ưu tiên trường sắp xếp

+ Grouping: chọn các trường dữ liệu được nhóm theo. Mục này được sử dụng khi người dùng chỉ định hàm thống kê ở "Aggregate". Tất cả các trường nếu không có chỉ định hàm thống kê thì ẩn được chọn "Grouping".

Criteria: mục đặt điều kiện lọc. Người dùng có thể dùng chuột để chọn dữ liệu trong một ô của bảng tính làm điều kiện động, thực chất là sử dụng hàm RANGE(..).

(5) Sau khi thực hiện 4 mục trên, A-Tools sẽ tự động tạo câu lệnh T-SQL cho hàm BS_SQL.

Sau khi thiết lập xong, người dùng chọn nút "Tiếp tục" sẽ xuất hiện màn hình dưới đây



+ Tên bảng kết quả: A-Tools sẽ tự động tạo tên cho vùng kết quả. Bạn nên đặt tên trong trường hợp bảng dữ liệu kết quả trả về lại được dùng cho một công thức khác.

+ Tự động tạo tên các cột: A-Tools tự đặt tên các vùng theo mỗi cột trong vùng kết quả. Những tên được tạo sẽ được sử dụng trong các công thức của Excel.

+ Chèn dòng: nếu được chọn, bảng kết quả sẽ chèn các dòng dữ liệu sau con trỏ. Nếu không, dữ liệu dễ bị ghi đè lên. Khi làm báo cáo bạn nên chọn mục này.

+ Địa chỉ đặt tham số SQL: ô chứa câu lệnh SQL. Cần thiết sử dụng khi chuỗi lệnh SQL dài (số ký tự > 128 ký tự).
+ Địa chỉ ô/vùng đặt công thức: ô đầu tiên trong bảng tính chứa giá trị của bảng kết quả.

Sau khi hoàn tất, chọn nút "Thực hiện" để A-Tools chạy hàm BS_SQL trong bảng tính.

Công thức có hàm BS_SQL được tạo ra bởi "SQL Builder" như sau:

```
=BS_SQL("SELECT DMHH.TEN, Sum(KHO.SLG) AS SLG
        FROM DMHH INNER JOIN KHO ON KHO.MA_VLSPHH = DMHH.MA_VLSPHH
        GROUP BY DMHH.TEN, KHO.LOAI_PHIEU
        HAVING KHO.LOAI_PHIEU = 'N'" , "INSERT=YES; AutoNames(THHH);")
```

Như vậy, với công cụ SQL Builder bạn có thể không cần phải hiểu rõ ngôn ngữ T-SQL mới lọc được dữ liệu mà chỉ bằng vài cú nhấp chuột. Việc công cụ tự sinh mã T-SQL bạn hoàn toàn tự học được ngôn ngữ này ở mức cơ bản.

Bạn xem video hướng dẫn tại địa chỉ sau:

http://bluesofts.net/giai-phap-phan-mem/a-tools-truy-van-loc-soan-thao-va-quan-tri-du-lieu-excel-qua-mang/117-lam-bao-cao-tren-excel-de-dang-bang-a-tools-.html

Hướng dẫn sử dụng hàm BS_SQL

Hàm BS_SQL trong A-Tools là một hàm đa năng và rất đặc biệt. Kết quả trả về của hàm là một bảng dữ liệu được lọc theo các điều kiện phức tạp, dữ liệu được móc nối với nhau mà nguồn dữ liệu có thể từ nhiều loại CSDL khác như đã trình bày ở phần trên. Chỉ cần một hàm BS_SQL, người dùng có thể tạo ra các loại báo cáo động trong Excel với tốc độ nhanh dù dữ liệu nhiều, chính xác cao.

Trước khi học tuần tự cách sử dụng hàm BS_SQL bạn hãy làm một ví dụ sau để hiểu qua hàm BS_SQL.

Download/Tải tập tin ví dụ "atools_exambles.zip"

http://bluesofts.net/Products/AddIns/A-Tools/Help/atools_exambles.zip

Mở tập tin dữ liệu mẫu "Examble.xls"

B1. Mở tập tin có dữ liệu ví dụ "Examble.xls"

Tên tập tin là "Examble.xls" lấy từ tập tin download hoặc trong bộ cài Add-in A-Tools tại đường dẫn "C:\A-Tools\DATA_DEMO\"

Tập tin "Examble.xls" sẽ luôn được mở và làm ví dụ, thực hành các công thức trong toàn bộ tài liệu này.

Sheet "KHO" chứa các phiếu nhập, phiếu xuất, vùng A3:K68 đã được đặt tên là "KHO".

	A	B	C	D	F	G	H	I	J	K
1									N = NHẬP	
2	Tên sổ **KHO = A3:K68**								X= XUẤT	
3	SO_CT	NGAY_CT	MA_KH	MA_NB	KHO	MA_VLSPHH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
52	XK00000010	09/08/2006	KH002		KHOHH	HH006	2	1000	X	$2,000.00
53	NK00000009	09/08/2006		NB002	KHOHH	HH006	3	10000	N	$30,000.00
54	XK00000011	09/08/2006	KH003		KHOHH	HH005	8	1000	X	$80,000.00
55	XK00000011	09/08/2006	KH003		KHOHH	HH006	4	1000	X	$4,000.00
56	XK00000012	09/08/2006	KH004		KHOHH	HH005	4	1000	X	$40,000.00
57	XK00000012	09/08/2006	KH004		KHOHH	HH006	8	1000	X	$8,000.00
58	NK00000010	24/10/2006		NB001	KHOHH	BM	10	64000	N	$640,000.00
59	NK00000010	24/10/2006		NB001	KHOHH	BE1	5	57000	N	$285,000.00
60	XK00000013	24/10/2006	KH001		KHOHH	BO	2	35000	X	$70,000.00
61	XK00000013	24/10/2006	KH001		KHOHH	BO	1	45000	X	$45,000.00
62	XK00000013	24/10/2006	KH001		KHOHH	BO	1	55000	X	$55,000.00
63	XK00000013	24/10/2006	KH001		KHOHH	LB	1	100000	X	$100,000.00
64	XK00000013	24/10/2006	KH001		KHOHH	LB	1	120000	X	$120,000.00
65	XK00000013	24/10/2006	KH001		KHOHH	BE	1	40000	X	$40,000.00
66	XK00000013	24/10/2006	KH001		KHOHH	BE	1	45000	X	$45,000.00
67	XK00000013	24/10/2006	KH001		KHOHH	BTC	1	40000	X	$40,000.00
68	XK00000013	24/10/2006	KH001		KHOHH	BTC	1	35000	X	$35,000.00

(Vùng A3:K68 được đặt tên (Define Name) là KHO)

Sheet "NKC" chứa các chứng từ kế toán, định khoản nợ, có. Vùng A3:K112 đã được đặt tên là "NKC".

Nghiệp vụ kinh tế phát sinh (SỔ NHẬT KÝ)

Tên sổ **NKC = A3:K112**

SO_CT	NGAY_CT	DIEN_GIAI	MA_KH	MA_NB	LOAI_THUE	NOTK	COTK	THANH_TIEN
BH001	31/07/2005	Bán hàng	KH001		VAT10	131		2,750,000.00
BH001	31/07/2005	Bán hàng	KH001		VAT10		3331	250,000.00
BH002	31/07/2005	Bán hàng	KH002		VAT10	131		3,300,000.00
BH002	31/07/2005	Bán hàng	KH002		VAT10		3331	300,000.00
BH003	31/07/2005	Bán hàng	KH003		VAT10	131		4,400,000.00
BH003	31/07/2005	Bán hàng	KH003		VAT10		3331	400,000.00

B2. Mở sheet "Thực hành" để nhập công thức

Lọc ra các dòng dữ liệu với các cột thông tin MA_VLSPHH, SLG, DON_GIA, THANH_TIEN trong vùng dữ liệu "KHO", điều kiện lọc là mã hàng là HH001 (WHERE MA_VLSPHH= 'HH001') . Công thức như dưới đây.

```
=bs_sql("SELECT MA_VLSPHH, SLG, DON_GIA, THANH_TIEN FROM KHO WHERE MA_VLSPHH='HH001' ")
```

Sau khi nhập xong bạn hãy nhấn phím ENTER kết quả sẽ như hình dưới đây:

A9 | {=bs_sql("SELECT MA_VLSPHH, SLG, DON_GIA, THANH_TIEN FROM KHO WHERE MA_VLSPHH='HH001' ")}

LƯU Ý
Hãy copy công thức từ các sheet hướng dẫn và dán vào sheet này.
Trước khi dán, bạn nên xoá tất cả vùng dữ liệu.
Là công thức mảng ({Công thức}), để sửa và kết thúc nhấn CTRL+SHIFT+ENTER
Để xoá công thức mảng, chọn toàn bộ vùng công thức rồi nhấn phím DELETE

Hãy đặt công thức dưới đây:

MA_VLSPHH	SLG	DON_GIA	THANH_TIEN
HH001	4	4,800,000	19,200,000
HH001	4	5,000,000	20,000,000
HH001	3	5,500,000	14,700,000
HH001	3	5,500,000	14,700,000
HH001	2	5,000,000	10,000,000
HH001	2	5,800,000	9,900,000
HH001	1	5,500,000	5,135,714
HH001	2	5,500,000	9,900,000
HH001	4	5,000,000	20,000,000
HH001	1	5,000,000	5,000,000

Hàm BS_SQL | Thực hành | AutoNames | N-X-T | NKC | KHO | DM_VLSPHH | DM Nhà cung cấp | DM Khách hàng | About | Sheet

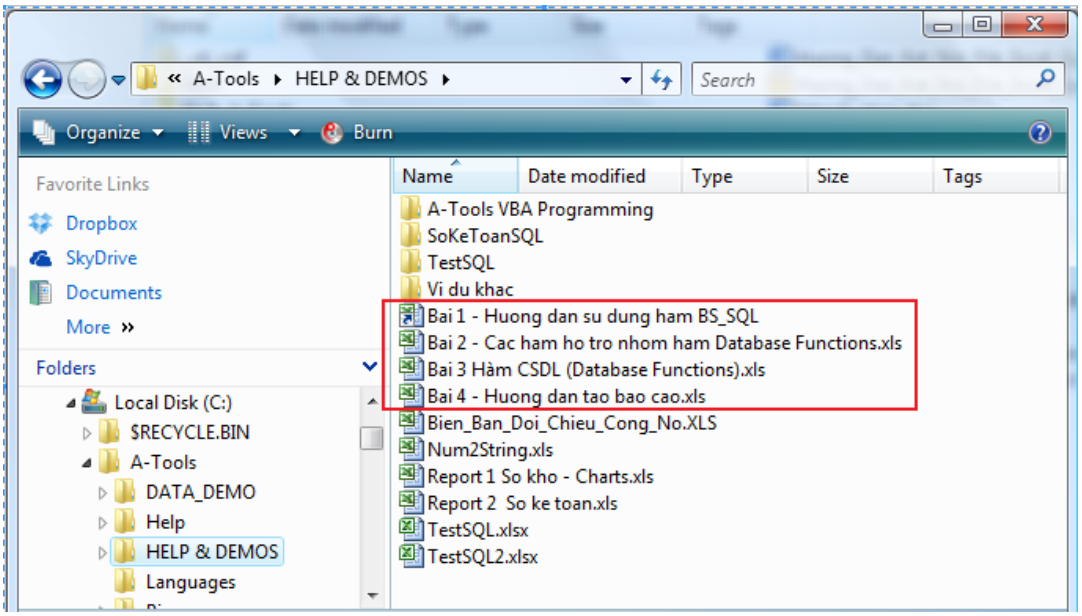
Lưu ý:

- ➔ Nếu sửa thì chỉ được chọn duy nhất một ô
- ➔ Là công thức mảng ({=Công thức}), để chạy công thức nhấn CTRL+SHIFT+ENTER
- ➔ Để xoá công thức mảng, chọn toàn bộ vùng công thức rồi nhấn phím DELETE
- ➔ Nếu không muốn thực hiện nhấn phím ESC để hủy việc sửa.

Nguyên tắc làm việc của hàm BS_SQL trong tập tin chia sẻ qua mạng

Hàm BS_SQL trong tập tin Excel được chia sẻ qua mạng có nguyên tắc như sau:
+ Tại máy chủ: không tự động chạy, ở chế độ đóng băng. Đây là đặc điểm giúp cho máy chủ tiết kiệm bộ nhớ và chạy nhanh hơn. Bạn có thể tự cập nhật bằng cách đặt con trỏ vào vùng công thức nhấn lệnh "Cập nhật" trên menu A-Tools hoặc nhấn tổ hợp phím CTRL+SHIFT+A.
+ Tại máy khách: chạy và cập nhật bình thường. Dữ liệu truy vấn là tại máy chủ.

Các tập tin Excel có dữ liệu mẫu cùng các ví dụ để học các hàm trong tài liệu này



Cấu trúc hàm BS_SQL

BS_SQL(SQL [,OPTIONS])

+ **SQL**: Là một chuỗi khai báo câu lệnh truy vấn dữ liệu theo ngôn ngữ T-SQL. T-SQL là một ngôn ngữ chuẩn truy vấn CSDL mà các hệ quản trị CSDL thường dùng như: MS Access, Foxpro, dBASE, SQLServer, Oracle, MySQL, Excel,…

Câu lệnh SQL trong hàm BS_SQL được phối hợp thêm các hàm A-Tools tạo thêm: RECNO, CELL, RANGE, DATA, GetOnce, FORMULA, FieldIf, VTC,…Nhờ phối hợp các hàm này trong khai báo SQL, hàm BS_SQL có thể trả về kết quả rất linh động, đáp ứng các yêu cầu phức tạo trong làm báo cáo chi tiết, báo cáo tổng hợp trong Excel.

+ **OPTIONS**: Là tham số tuỳ chọn (có thể không có), tham số này là một chuỗi, cho phép khai báo thêm các tham số cho hàm, các tham số được ngăn cách nhau bởi dấu chấm phảy (;). Nếu không có tham số này, hàm BS_SQL sẽ truy vấn dư liệu trong chính tập tin Excel đang mở hiện tại và trả về bảng kết quả lọc.

Các tham số sử dụng trong OPTIONS bao gồm: DBKEY, HR, NAME, AUTONAMES, INSERT, SERVERSOURCE.

Có thể lập trình các macro trong VBA để chạy trong các sự kiện của hàm BS_SQL, khai báo chạy các macro thực hiện trong tham số OPTIONS. Các tham số gồm: OnBeforeUpdate, OnAfterUpdate, OnDblClick, OnSelectionChange, OnGetValue

Cách sử dụng các tham số được trình bày chi tiết phần sau.

Phương pháp trích lọc dữ liệu bằng hàm BS_SQL

Tìm hiểu ngôn ngữ T-SQL

```
SELECT [DISTINCT] [TOP n] select_list FROM table_source [ WHERE search_condition ]
        [ GROUP BY group_by_expression ] [ HAVING search_condition ] [ ORDER BY
order_expression [ ASC | DESC ] ]
```

Đọc thêm: Có rất nhiều các website hướng dẫn về ngôn ngữ này, các bạn có thể tham khảo tài các website sau:

http://www.w3schools.com/sql/sql_select.asp
http://www.w3schools.com/sql/sql_distinct.asp
http://www.w3schools.com/sql/sql_where.asp
http://www.w3schools.com/sql/sql_orderby.asp
http://www.w3schools.com/sql/sql_like.asp
http://www.w3schools.com/sql/sql_join.asp

Các từ có màu xanh là từ khoá, khi viết câu lệnh SQL phải đảm bảo thứ tự trước sau của các từ khoá.

Những khai báo nằm trong ngoặc vuông ([ ]) có thể dùng hoặc không - tuỳ chọn.

Sử dụng từ khóa SELECT

SELECT select_list

select_list: Danh sách các cột trong bảng dữ liệu "table_source", cũng có thể là danh sách các giá trị.

Ví dụ lấy ra các cột dữ liệu MA_VLSPHH, SLG, DON_GIA, THANH_TIEN

```
SELECT MA_VLSPHH, SLG, DON_GIA, THANH_TIEN
```

Ví dụ lấy ra các cột dữ liệu MA_VLSPHH, SLG, DON_GIA, THANH_TIEN nhưng các cột lấy ra được đặt lại tên

```
SELECT MA_VLSPHH AS [Mã Hàng], SLG AS [Số Lượng], DON_GIA AS [Đơn
Giá], THANH_TIEN AS [Thành Tiền]
```

---> Để đổi tên cột, đặt AS [Tên cột] ngay sau cột muốn đổi

---> Nếu lấy tất cả các cột trong "table_source" dùng ký tự *

table_source: là một bảng hay danh sách bảng dữ liệu có quan hệ với nhau. Nếu là danh sách thì các bảng cách nhau bởi dấu phảy (,) hoặc kết nối với nhau bởi từ khóa JOIN.

Mở tập tin "Examble.xls" vào sheet "Thực hành" làm các ví dụ sau:

Ví dụ 1: Chọn (SELECT) tất cả các cột từ (FROM) sổ KHO

```
=BS_SQL("SELECT * FROM KHO")
```

Ví dụ 2: Chọn 10 dòng đầu của tất cả các cột từ sổ KHO

```
=BS_SQL("SELECT TOP 10 * FROM KHO")
```

Ví dụ 3: Chọn danh sách số chứng từ duy nhất (DISTINCT) từ sổ KHO

=BS_SQL("SELECT DISTINCT SO_CT FROM KHO")

A9 | {=bs_sql("SELECT DISTINCT SO_CT FROM KHO")}

	A	B	C	D	E	F
9	SO_CT					
10	NK00000001					
11	NK00000002					
12	NK00000003					
13	NK00000004					
14	NK00000005					
15	NK00000006					
16	NK00000007					
17	NK00000008					
18	NK00000009					

Ví dụ 4: lấy ra các cột dữ liệu MA_VLSPHH, SLG, DON_GIA, THANH_TIEN trong bảng KHO

=BS_SQL("SELECT MA_VLSPHH, SLG, DON_GIA, THANH_TIEN FROM KHO")

Ví dụ 5: lấy ra các cột dữ liệu MA_VLSPHH, SLG, DON_GIA, THANH_TIEN trong bảng KHO

```
=BS_SQL("SELECT MA_VLSPHH AS [Mã Hàng], SLG AS [Số Lượng], DON_GIA AS
                    [Đơn Giá], THANH_TIEN AS [Thành Tiền]
        FROM KHO")
```

A9 | {=bs_sql("SELECT MA_VLSPHH AS [Mã Hàng],
SLG AS [Số Lượng],
DON_GIA AS [Đơn Giá],
THANH_TIEN AS [Thành Tiền]
FROM KHO")}

	A	B	C	D	E	F	G
9	Mã Hàng	Số Lượng	Đơn Giá	Thành Tiền			
10	HH001	4	4,800,000	19,200,000			
11	HH002	6	4,500,000	27,000,000			
12	HH003	6	2,800,000	16,800,000			
13	HH004	3	4,200,000	12,600,000			
14	HH001	4	5,000,000	20,000,000			
15	HH002	3	4,800,000	14,400,000			
16	HH003	2	3,000,000	6,000,000			
17	HH004	2	4,500,000	9,000,000			

Ví dụ 6: Có thể thực thi các phép toán, mở thêm cột tính toán sau từ khóa SELECT

=bs_sql("SELECT 1+1 AS BT1, MA_VLSPHH, SLG, DON_GIA, SLG*DON_GIA AS [Thành Tiền], '' AS [Ghi chú] FROM KHO")

BT1	MA_VLSPHH	SLG	DON_GIA	Thành Tiền	Ghi chú
2	HH001	4	4,800,000	19200000	
2	HH001	4	5,000,000	20000000	
2	HH001	3	5,500,000	16500000	
2	HH001	3	5,500,000	16500000	

1 + 1 AS BT1 → Cột BT1 được tạo ra, kết quả của nó là phép cộng 1 + 1 = 2

" AS [Ghi chú]→ Cột "Ghi chú" được tạo ra, kết quả của nó là giá trị chuỗi trống. Giá trị chuỗi trong SQL là nằm trong hai dấu nháy đơn 'Chuỗi giá trị'

Các phép toán sử dụng về cơ bản giống như các phép toán bạn lập trong công thức Excel.

Sử dụng từ khóa WHERE

WHERE search_condition

search_condition: là điều kiện – biểu thức logic/so soánh. Các bản ghhi được lọc ra nếu biểu thức so sánh trả về TRUE. Theo T-SQL/A-Tools Nếu giá trị so sánh là văn bản thì phải đặt trong hai dấu nháy đơn (') , trong Excel là nháy kép ".

Các toán tử so sánh là: > , >= , < , <= , <> , = , LIKE , IN, BETWEEN

Khi so sánh tập ký tự bất kỳ ta dùng ký tự % , ký tự này dùng với toán tử là LIKE

Toán tử kết hợp logic: AND , OR , NOT

Ví dụ 7: Lấy ra tất cả các cột từ bảng KHO ở đó mã khách hàng (MA_KH) là KH001

```
=BS_SQL("SELECT * FROM KHO WHERE MA_KH='KH001' ")
```

Ví dụ 8: Lấy ra các cột SO_CT, NGAY_CT, DIEN_GIAI, THANH_TIEN từ sổ NKC ở đó ký tự đầu của NOTK là 6

```
=BS_SQL("SELECT SO_CT, NGAY_CT, DIEN_GIAI, NOTK, COTK, THANH_TIEN
        FROM NKC
        WHERE NOTK LIKE '6%' ")
```

Ký tự % được dùng làm đại diện để so sánh với nhóm ký tự bất kỳ, thường dùng với toán tử LIKE.

A9 {=bs_sql("SELECT SO_CT, NGAY_CT, DIEN_GIAI, NOTK, COTK, THANH_TIEN
FROM NKC
WHERE NOTK LIKE '6%' ")}

	A	B	C	D	E	F	G	H	I
9	SO_CT	NGAY_CT	DIEN_GIAI	NOTK	COTK	THANH			
10	BH00000001	38570	Giá vốn hàn	632	156	35513077			
11	BH00000002	38572	Giá vốn hàn	632	156	20813077			
12	PT00000002	38575	Chiết khấu c	635	131	300000			
13	BH00000003	38578	Giá vốn hàn	632	156	24023077			
14	PT00000003	38580	Chiết khấu c	635	131	80000			

Ví dụ 9: Lấy ra các cột SO_CT, NGAY_CT, DIEN_GIAI, THANH_TIEN từ sổ NKC ở đó ký tự đầu của cột NOTK là 11 và COTK có ký tự đầu là 5

```
=bs_sql("SELECT SO_CT, NGAY_CT, DIEN_GIAI, NOTK, COTK, THANH_TIEN
        FROM NKC
        WHERE NOTK LIKE '11%' AND COTK LIKE '5%' ")
```

A9 fx {=bs_sql("SELECT SO_CT, NGAY_CT, DIEN_GIAI, NOTK, COTK, THANH_TIEN
FROM NKC
WHERE NOTK LIKE '11%' AND COTK LIKE '5%' ")}

	A	B	C	D	E	F	G	H	I
9	SO_CT	NGAY_CT	DIEN_GIAI	NOTK	COTK	THANH			
10	BH00000006	38590	Doanh thu b	1111	5111	0			
11	PT00000004	38590	Doanh thu b	1111	5111	22700000			
12	BH00000007	38591	Doanh thu b	1111	5111	0			
13	PT00000005	38591	Doanh thu b	1111	5111	24200000			

Trong search_condition, bạn có thể sử dụng IN (tập danh sách giá trị) để làm điều kiện.

Sau IN ta có thể là một cột dữ liệu lấy ra bởi SELECT, hoặc danh sách các giá trị cách nhau bởi dấu phảy (,).

Ví dụ 10: Chọn tất cả các cột dữ liệu (*) từ sổ KHO ở đó mã hàng (MA_VLSPHH) thuộc danh sách các mã HH001, HH002, HH003 và LOAI_PHIEU là nhập (N)

```
=bs_sql("SELECT *
        FROM KHO
        WHERE MA_VLSPHH IN ('HH001','HH002','HH003') AND LOAI_PHIEU='N' ")
```

A10 fx {=bs_sql("SELECT *
FROM KHO
WHERE MA_VLSPHH IN ('HH001','HH002','HH003') AND LOAI_PHIEU='N' ")}

	A	B	C	D	F	G	H	I	J	K
9	SO_CT	NGAY_CT	MA_KH	MA_NB	KHO	MA_VLSPH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
10	TDKHO	38564		NB001	KHOHH	HH001	4	4800000	N	19200000
11	TDKHO	38564		NB002	KHOHH	HH002	6	4500000	N	27000000
12	TDKHO	38564		NB004	KHOHH	HH003	6	2800000	N	16800000
13	NK00000001	38565		NB001	KHOHH	HH001	4	5000000	N	20000000

Ví dụ 11: Lấy ra tất cả các cột dữ liệu (*) từ sổ KHO ở đó mã hàng (MA_VLSPHH) có trong danh sách mã có loại (DMVLSPHH.LOAI) là VL và và LOAI_PHIEU là nhập (N).

Trong tập tin "Examble.xls" có sheet "DMVLSPHH" chứa thông tin về danh mục hàng hóa.

Tên sổ **DMVLSPHH = A3:D15**

MA_VLSPHH	TEN	Dvi	Loai
HH001	Máy tính ĐNA L1	Chiếc	HH
HH002	Máy tính ĐNA L2	Chiếc	HH
HH003	Máy in HP 1200	Chiếc	HH
HH004	Máy in HP 1300	Chiếc	HH
HH005	Dầu A	Chai	HH
HH006	Dầu B	Chai	HH
BO	Bò xào nướng	Đĩa	HH
LB	Lẩu bò	Nồi	HH
BE	Bê	Đĩa	HH
BTC	Bê tái chanh	Đĩa	HH
BM	Bò mông	Kg	VL
BE1	Bê	Kg	VL

Hàm BS_SQL | Thực hành | **DM VLSPHH** | AutoNames

(Vùng A3:D15 được đặt tên (Define Name) là "DMVLSPHH")

```
=bs_sql("SELECT *
        FROM KHO
        WHERE MA_VLSPHH IN (SELECT MA_VLSPHH FROM DMVLSPHH WHERE LOAI = 'VL')
                AND LOAI_PHIEU='N' ")
```

Kết quả là:

A9 {=bs_sql("SELECT *
FROM KHO
WHERE MA_VLSPHH IN (SELECT MA_VLSPHH FROM DMVLSPHH WHERE LOAI = 'VL')
AND LOAI_PHIEU='N' ")}

SO_CT	NGAY_CT	MA_KH	MA_NB	KHO	MA_VLSPH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
NK00000010	39014		NB001	KHOHH	BM	10	64000	N	640000
NK00000010	39014		NB001	KHOHH	BE1	5	57000	N	285000

Lệnh sau trên trả về danh sách các mã hàng có loại là 'VL'.

SELECT MA_VLSPHH FROM DMVLSPHH WHERE LOAI = 'VL'

Qua ví dụ ta thấy sự linh hoạt troang ngôn ngữ T-SQL. Có thể làm điều kiện mà dữ liệu làm điều kiện nằm ở sheet khác. Ví dụ trên, dữ liệu lấy ra là sổ KHO nhưng dữ liệu làm điều kiện lấy ở DMVLSPHH so sánh với cột MA_VLSPHH ở sổ KHO.

➔ Tìm hiểu thêm LIKE, IN tại đây:
http://www.w3schools.com/sql/sql_like.asp
http://www.w3schools.com/sql/sql_in.asp

Ví dụ 12: Lấy ra tất cả các cột dữ liệu (*) từ sổ KHO ở đó THANH_TIEN >= 19000000

```
=bs_sql("SELECT *
        FROM KHO
        WHERE THANH_TIEN >= 19000000 ")
```

A9 {=bs_sql("SELECT *
FROM KHO
WHERE THANH_TIEN >= 19000000 ")}

	A	B	C	D	F	G	H	I	J	K
9	SO_CT	NGAY_CT	MA_KH	MA_NB	KHO	MA_VLSPH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
10	TDKHO	38564		NB001	KHOHH	HH001	4	4,800,000	N	19,200,000
11	TDKHO	38564		NB002	KHOHH	HH002	6	4,500,000	N	27,000,000
12	NK00000001	38565		NB001	KHOHH	HH001	4	5,000,000	N	20,000,000
13	NK00000002	38567		NB002	KHOHH	HH002	4	4,800,000	N	19,200,000
14	NK00000003	38584		NB004	KHOHH	HH002	4	4,800,000	N	19,200,000
15	NK00000003	38584		NB004	KHOHH	HH004	6	4,500,000	N	27,000,000
16	XK00000007	38591	KH003		KHOHH	HH002	4	5,200,000	X	19,164,103
17	NK00000006	38592		NB002	KHOHH	HH002	6	4,800,000	N	28,800,000
18	NK00000006	38592		NB002	KHOHH	HH001	4	5,000,000	N	20,000,000
19	NK00000008	38938		NB002	KHOHH	HH002	4	4,800,000	N	19,200,000

Ví dụ 13: Lấy ra tất cả các cột dữ liệu (*) từ sổ KHO ở đó THANH_TIEN trong khoảng từ 19000000 đến 25000000

```
=bs_sql("SELECT *
        FROM KHO
        WHERE THANH_TIEN >= 19000000 AND THANH_TIEN <= 25000000 ")
```

Cách khác là dùng BETWEEN

```
=bs_sql("SELECT *
        FROM KHO
        WHERE THANH_TIEN BETWEEN 19000000 AND 25000000 ")
```

A9 {=bs_sql("SELECT *
FROM KHO
WHERE THANH_TIEN >= 19000000 AND THANH_TIEN <= 25000000")}

	A	B	C	D	F	G	H	I	J	K
9	SO_CT	NGAY_CT	MA_KH	MA_NB	KHO	MA_VLSPH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
10	TDKHO	38564		NB001	KHOHH	HH001	4	4,800,000	N	19,200,000
11	NK00000001	38565		NB001	KHOHH	HH001	4	5,000,000	N	20,000,000
12	NK00000002	38567		NB002	KHOHH	HH002	4	4,800,000	N	19,200,000
13	NK00000003	38584		NB004	KHOHH	HH002	4	4,800,000	N	19,200,000
14	XK00000007	38591	KH003		KHOHH	HH002	4	5,200,000	X	19,164,103
15	NK00000006	38592		NB002	KHOHH	HH001	4	5,000,000	N	20,000,000
16	NK00000008	38938		NB002	KHOHH	HH002	4	4,800,000	N	19,200,000

fieldname BETWEEN a AND b
Tương đương với só sánh
fieldname >= a AND fieldname <= b
Nếu fieldname so sánh trước a và sau b không có dấu bằng (=) thì không được dùng BETWEEN

Ví dụ 14: Lấy ra tất cả các cột dữ liệu (*) từ sổ KHO ở đó ngày chứng từ (NAY_CT) lớn hơn 24/06/2006

```
=bs_sql("SELECT *
        FROM KHO
        WHERE NGAY_CT >= #10/24/2006# ")
```

A9 {=bs_sql("SELECT *
FROM KHO
WHERE NGAY_CT >= #10/24/2006#")}

	A	B	C	D	F	G	H	I	J	K
9	SO_CT	NGAY_CT	MA_KH	MA_NB	KHO	MA_VLSPH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
10	NK00000010	24/10/2006		NB001	KHOHH	BM	10	64,000	N	640,000
11	NK00000010	24/10/2006		NB001	KHOHH	BE1	5	57,000	N	285,000
12	XK00000013	24/10/2006	KH001		KHOHH	BO	2	35,000	X	70,000
13	XK00000013	24/10/2006	KH001		KHOHH	BO	1	45,000	X	45,000

(Trong T-SQL của Excel, giá trị ngày phải nằm trong hai dấu #, theo định dạng MM/dd/yy)

Xin nhắc lại các giá trị tham gia vào biểu thức trong SQL của Excel theo quy định như sau: 'Giá trị văn bản' – Có ký tự nháy đơn (`) bao hai bên #Giá trị ngày tháng# – Có ký tự # bao hai bên Số, phần trăm, tỷ lệ (Number) - Không có ký tự bao hai bên

Sử dụng từ khóa FROM

FROM table_source

table_source: là một hay nhiều bảng dữ liệu nguồn chứa các thông tin cần trích lọc (lấy ra), làm điều kiện lọc. Trong Excel, bảng dữ liệu thường được định nghĩa (Define Name) là một NAME có cấu trúc dạng bảng. Như các ví dụ trên chính là các vùng KHO, DMVLSPHH, NKC. Nó có thể là tên một sheet cụ thể, khi đó cần đặt theo mẫu [Tên sheet$].

Các ví dụ sau:

```
=BS_SQL("SELECT * FROM [KHO$A3:K68]")
```

Nếu vùng dữ liệu KHO!A3:K68 được đặt tên (Define Name) là **KHO** thì công thức như sau

```
=BS_SQL("SELECT * FROM KHO")
```

Nếu lấy dữ liệu theo sheet thì công thức như sau

```
= BS_SQL("SELECT * FROM [KHO$]")
```

Công thức lấy theo sheet [KHO$] tuy chạy được nhưng chúng ta không nên dùng vì có thể sheet chứa nhiều loại dữ liệu khác nhau, không theo cấu trúc bảng/table dẫn đến lỗi trong quá trình truy vấn dữ liệu. Chúng ta thống nhất dùng phương pháp đặt tên vùng dữ liệu có cấu trúc dạng bảng để làm việc với hàm BS_SQL trong Add-in A-Tools.

Ghép dữ liệu từ nhiều bảng

Các ví dụ ở các phần trước giúp chúng ta cũng đã hiểu việc sử dụng từ khóa FROM. Trong phần này tôi xin đi sâu vào các phương pháp ghép các bảng dữ liệu lại với nhau để rồi lấy các cột dữ liệu của chúng ra một bảng kết quả cuối cùng.

Ví dụ chúng ta có hai bảng KHO và DMVLSPHH như dưới đây

SO_CT	NGAY_CT	MA_VLSPHH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
TDKHO	31/07/2005	HH001	4	4800000	N	19,200,000.00
TDKHO	31/07/2005	HH002	6	4500000	N	27,000,000.00
TDKHO	31/07/2005	HH003	6	2800000	N	16,800,000.00
TDKHO	31/07/2005	HH004	3	4200000	N	12,600,000.00
NK00000001	01/08/2005	HH001	4	5000000	N	20,000,000.00
NK00000001	01/08/2005	HH002	3	4800000	N	14,400,000.00

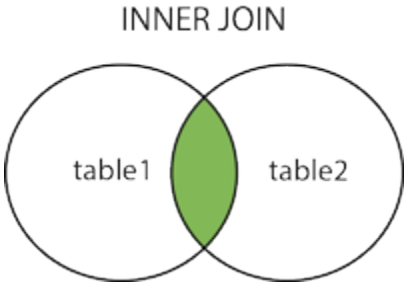
MA_VLSPHH	TEN	Dvi	Loai
HH001	Máy tính ĐNA L1	Chiếc	HH
HH002	Máy tính ĐNA L2	Chiếc	HH
HH003	Máy in HP 1200	Chiếc	HH
HH004	Máy in HP 1300	Chiếc	HH
HH005	Dầu A	Chai	HH
HH006	Dầu B	Chai	HH
BO	Bò xào nướng	Đĩa	HH
LB	Lẩu bò	Nồi	HH
BE	Bê	Đĩa	HH
BTC	Bê tái chanh	Đĩa	HH
BM	Bò mông	Kg	VL
BE1	Bê	Kg	VL

Công thức ghép nối hai bảng

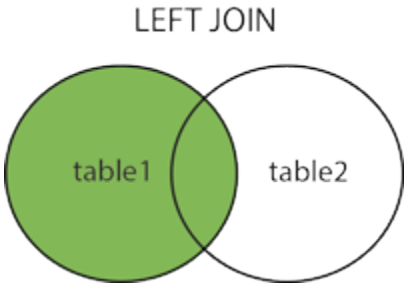
Table1 Phương thức ghép Table2 ON Biểu thức quan hệ (so sánh) Table1 với Table2

Các phương thức ghép

INNER JOIN: ghép ngang bằng. Chỉ dữ liệu nào có cả hai bảng mới hiện ra. Ví dụ bảng 1 có giá trị 1, bảng 2 có giá trị 1,2 thì chỉ giá trị 1 được lấy ra.

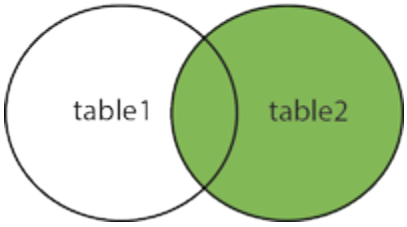


LEFT JOIN: dữ liệu bảng bên trái được lấy ra hết, những dữ liệu bảng bên phải nếu thỏa mãn điều kiện mới lấy ra.



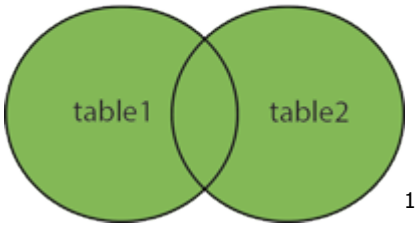
RIGHT JOIN: dữ liệu bảng bên phải được lấy ra hết, những dữ liệu bảng bên trái nếu thỏa mãn điều kiện mới lấy ra.

RIGHT JOIN



FULL JOIN: tất cả dữ liệu các bảng bên trái và bên phải được lấy ra hết. Phương thức này là hợp bởi LEFT, RIGHT

FULL OUTER JOIN



[1]

Bây giờ cần ghép hai bảng KHO và DMVLSPHH lại với nhau. Muốn ghép được chúng ta cần chọn phương thức ghép và chỉ ra biểu thức quan hệ giữa chúng.

Nếu không có lý do đặc biệt ta thường chọn phương thức ghép nối INNER JOIN (1-1)

- Table1 có quan hệ với Table2 nếu chúng cùng chứa một loại thông tin ở cột nào đó.

Bảng KHO và DMVLSPHH thì cột MA_VLSPHH của cả hai bảng này đều lưu một loại thông tin đó là mã hàng. Vậy giữa chúng có quan hệ với nhau bởi cột KHO.MA_VLSPHH và DMVLSPHH.MA_VLSPHH . Tên cột ở các bảng có thể khác nhau nhưng nội dung phải lưu loại thông tin giống nhau.

Biểu thức thể hiện quan hệ là

```
KHO. MA_VLSPHH=DMVLSPHH.MA_ VLSPHH
```

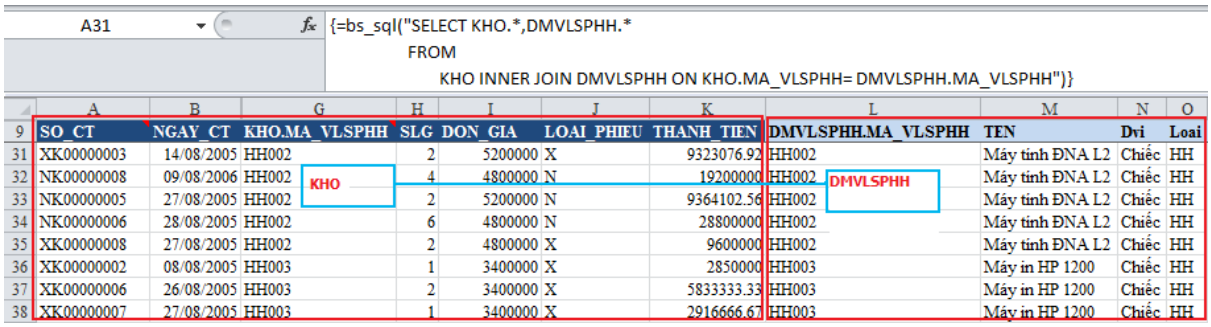
Công thức ghép sổ KHO với sổ DMVLSPHH sẽ là:

```
FROM
        KHO INNER JOIN DMVLSPHH ON KHO.MA_VLSPHH=DMVLSPHH.MA_VLSPHH
```

[1] http://www.w3schools.com/sql/sql_join_inner.asp

Thực hiện công thức lấy toàn bộ dữ liệu trong sổ KHO (KHO.*) và toàn bộ dữ liệu sổ DMVLSPHH(DMVLSPHH.*) có liên quan với nhau

```
=bs_sql("SELECT KHO.*,DMVLSPHH.*
        FROM
          KHO INNER JOIN DMVLSPHH ON KHO.MA_VLSPHH= DMVLSPHH.MA_VLSPHH")
```



	A	B	G	H	I	J	K	L	M	N	O
9	SO_CT	NGAY_CT	KHO.MA_VLSPHH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN	DMVLSPHH.MA_VLSPHH	TEN	Dvi	Loai
31	XK00000003	14/08/2005	HH002	2	5200000	X	9323076.92	HH002	Máy tính ĐNA L2	Chiếc	HH
32	NK00000008	09/08/2006	HH002	4	4800000	N	19200000	HH002	Máy tính ĐNA L2	Chiếc	HH
33	NK00000005	27/08/2005	HH002	2	5200000	N	9364102.56	HH002	Máy tính ĐNA L2	Chiếc	HH
34	NK00000006	28/08/2005	HH002	6	4800000	N	28800000	HH002	Máy tính ĐNA L2	Chiếc	HH
35	XK00000008	27/08/2005	HH002	2	4800000	X	9600000	HH002	Máy tính ĐNA L2	Chiếc	HH
36	XK00000002	08/08/2005	HH003	1	3400000	X	2850000	HH003	Máy in HP 1200	Chiếc	HH
37	XK00000006	26/08/2005	HH003	2	3400000	X	5833333.33	HH003	Máy in HP 1200	Chiếc	HH
38	XK00000007	27/08/2005	HH003	1	3400000	X	2916666.67	HH003	Máy in HP 1200	Chiếc	HH

Bạn cũng có thể ghép hai bảng theo cách sau:

```
=bs_sql("SELECT KHO.*,DMVLSPHH.*
        FROM KHO, DMVLSPHH
        WHERE KHO.MA_VLSPHH= DMVLSPHH.MA_VLSPHH")
```

Khi ghép nối từ hai bảng trở lên, muốn lấy dữ liệu ở bảng nào ta cần làm theo cú pháp

Tên_bảng.Tên_cột

KHO.* là toàn bộ các cột sổ KHO
DMVLSPHH.* là toàn bộ các cột sổ DMVLSPHH

Ví dụ 15: Lấy ra các cột thông tin mã hàng, tên hàng, số lượng, đơn giá, thành tiền từ sổ KHO.

Từ yêu cầu trên ta thấy trong bảng KHO không có tên hàng. Muốn lấy tên hàng thì phải nối bảng KHO với DMVLSPHH vì ở đó mới có tên hàng.

Công thức là:

```
=bs_sql("SELECT KHO.MA_VLSPHH,DMVLSPHH.TEN,KHO.SLG,KHO.DON_GIA,KHO.THANH_TIEN
        FROM
              KHO INNER JOIN DMVLSPHH ON KHO.MA_VLSPHH= DMVLSPHH.MA_VLSPHH")
```

Dùng phương pháp đặt **ALIAS** (tên khác) cho các bảng để rút ngắn câu lệnh. Khi đã đặt Alias ta có thể dùng tên Alias thay cho tên bảng.

Cú pháp:

Tên_bảng Tên_alias

Ví dụ: KHO K (K là tên Alias của KHO)

Công thức trên sửa theo cách đặt Alias như sau:

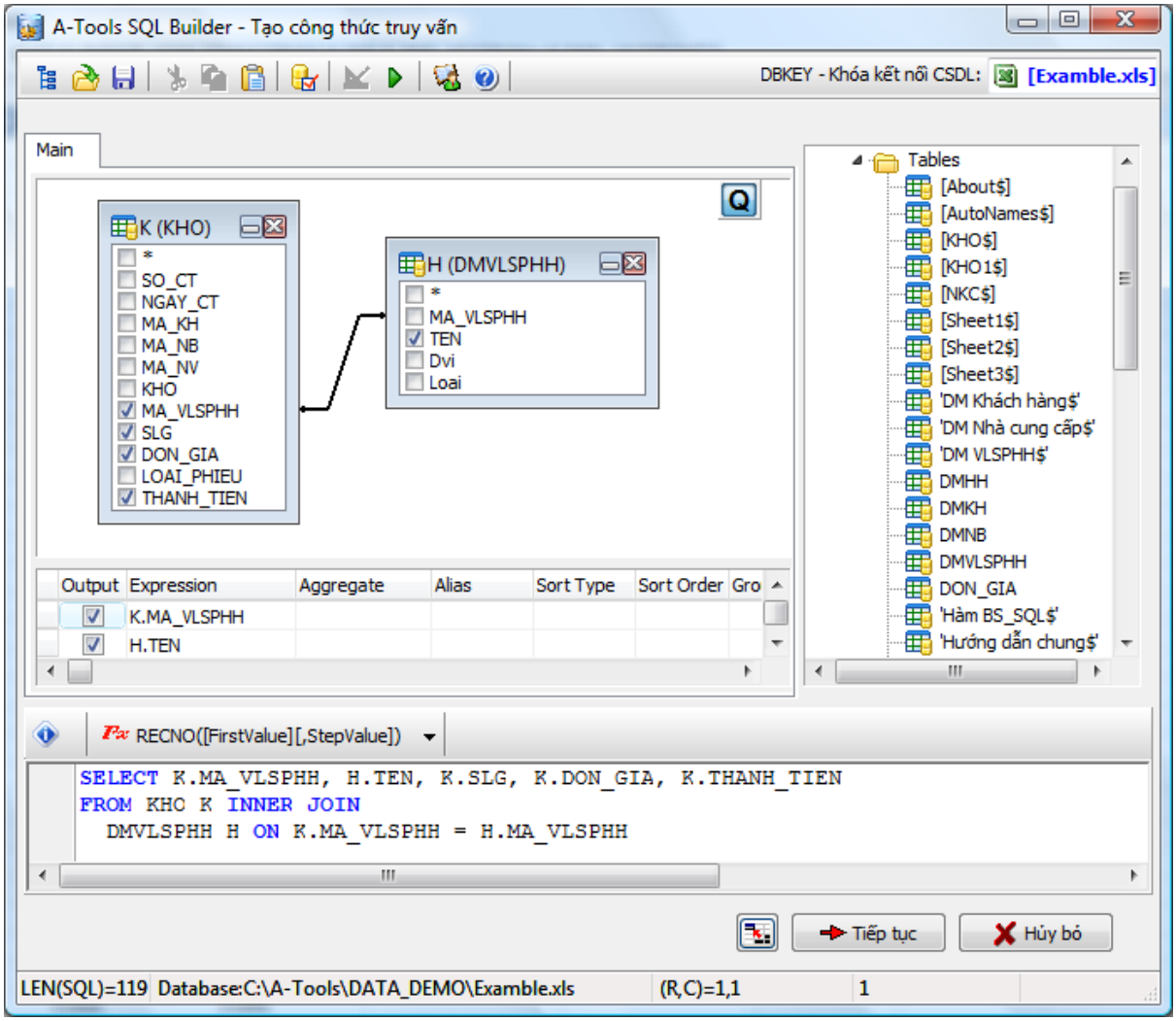
```
=bs_sql("SELECT K.MA_VLSPHH,H.TEN,K.SLG,K.DON_GIA,K.THANH_TIEN
        FROM
              KHO K INNER JOIN DMVLSPHH H ON K.MA_VLSPHH= H.MA_VLSPHH")
```

A9 {=bs_sql("SELECT K.MA_VLSPHH,H.TEN,K.SLG,K.DON_GIA,K.THANH_TIEN
FROM
KHO K INNER JOIN DMVLSPHH H ON K.MA_VLSPHH= H.MA_VLSPHH")}

	A	B	C	D	E	F	G	H	I
9	MA_VLSPHH	TEN	SLG	DON_GIA	THANH_TIEN				
34	HH002	Máy tính ĐNA L2	6	4800000	28800000				
35	HH002	Máy tính ĐNA L2	2	4800000	9600000				
36	HH003	Máy in HP 1200	1	3400000	2850000				
37	HH003	Máy in HP 1200	2	3400000	5833333.33				
38	HH003	Máy in HP 1200	1	3400000	2916666.67				

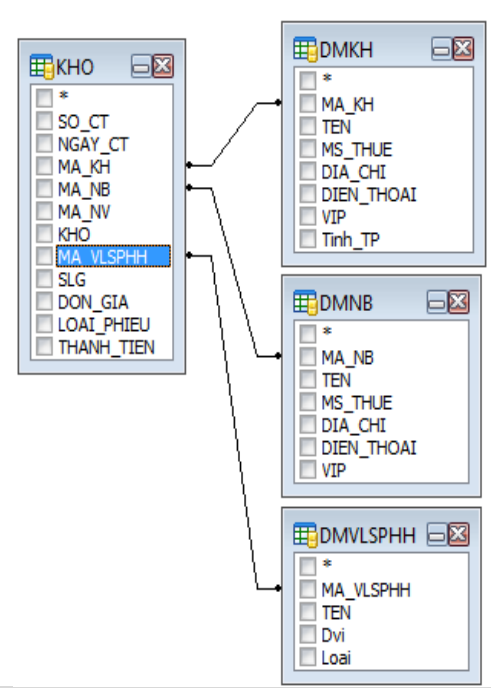
Theo cách làm công thức Excel thông thường, muốn lấy tên phải dùng VLOOKUP tìm mã hàng trong DMVLSPHH rồi copy cho tất cả các dòng còn lại. Với Add-in A-Tools chỉ cần kết nối với sổ danh mục như trên rồi SELECT thông tin cần lấy. Chỉ một công thức BS_SQL duy nhất là hoàn thành cả một bảng dữ liệu báo cáo.

Với ví dụ trên, nếu làm bằng SQL Builder ta sẽ có màn hình thiết kế như dưới đây.



(Để tạo Alias, nhấp đúp chuột vào tên bảng rồi nhập tên Alias là được.)

Ví dụ quan hệ giữa các bảng trong tập tin (CSDL) Examble.xls dưới đây




```
SELECT ...
FROM
((DMVLSPHH INNER JOIN KHO
            ON KHO.MA_VLSPHH = DMVLSPHH.MA_VLSPHH)
INNER JOIN DMKH ON KHO.MA_KH = DMKH.MA_KH)
INNER JOIN DMNB ON KHO.MA_NB = DMNB.MA_NB
```

Ví dụ 16: Lập sổ gồm các cột: Mã khách hàng, Tên khách hàng, Số tiền. Dữ liệu được lấy từ 2 sổ KHO và DMKH.

Danh mục khách hàng. Tên đặt là "DMKH"

	A	B	C	D	E	F	G
1			ĐƠN VỊ KHÁCH HÀNG				
2		Tên sổ **DMKH**					
3	MA_KH	TEN	MS_THUE	DIA_CHI	DIEN_THOAI	VIP	Tinh_TP
4	KH001	Cty TNHH Hoàng Long	1233456780123	34 - Tây Sơn	8589555		Hà Nội
5	KH002	Cửa hàng Thành Đông	0021545449	55 - Nguyễn Trãi	3622222		Hải Phòng
6	KH003	Cửa hàng máy tính Đông Đô	0120113115	108 - ĐBP	4595126		Hà Nội
7	KH004	Cty IPQ	1560045881	96 - Nhổn	7655411		Hải Phòng

Công thức là:

```
=bs_sql("SELECT K.MA_KH, KH.TEN, K.THANH_TIEN
        FROM KHO K INNER JOIN DMKH KH ON K.MA_KH=KH.MA_KH
        WHERE LOAI_PHIEU='X' ")
```

Hoặc

```
=bs_sql("SELECT K.MA_KH, KH.TEN, K.THANH_TIEN
        FROM KHO K, DMKH KH
        WHERE K.MA_KH=KH.MA_KH AND LOAI_PHIEU='X' ")
```

Kết quả là:

A9 fx {=bs_sql("SELECT K.MA_KH, KH.TEN, K.THANH_TIEN
FROM KHO K INNER JOIN DMKH KH ON K.MA_KH=KH.MA_KH
WHERE LOAI_PHIEU='X' ")}

	A	B	C	D	E	F	G
9	MA_KH	TEN	THANH_TIEN				
13	KH001	Cty TNHH Hoàng Long	8,640,000				
14	KH002	Cửa hàng Thành Đông	9,323,077				
15	KH002	Cửa hàng Thành Đông	2,850,000				
16	KH002	Cửa hàng Thành Đông	8,640,000				
17	KH003	Cửa hàng máy tính Đông Đô	9,323,077				
18	KH003	Cửa hàng máy tính Đông Đô	14,700,000				
19	KH004	Cty IPQ	14,046,154				
20	KH004	Cty IPQ	5750000				
21	KH004	Cty IPQ	17897142.86				
22	KH001	Cty TNHH Hoàng Long	9364102.56				

Sử dụng từ khóa GROUP BY

GROUP BY group_by_expression: Được dùng khi bạn muốn thống kê chỉ tiêu số lượng và nhóm theo một hoặc nhiều đối tượng.

group_by_expression: là danh sách các cột được nhóm, các cột được ngăn cách nhau bởi dấu phảy (,)

➔ Các hàm dùng để thống kê thường dùng gồm: SUM , COUNT , AVG , MIN , MAX

Ví dụ 17: lấy ra dữ liệu gồm các cột: MA_KH, Tổng tiền tròn sổ NKC, điều kiện NOTK là 131 (tổng hợp số tiền cho từng khách hàng mà phát sinh Nợ TK =131)

```
=bs_sql("SELECT MA_KH AS [Mã KH], SUM(THANH_TIEN) AS [Tổng tiền]
        FROM NKC
        WHERE NOTK LIKE '131%'
        GROUP BY MA_KH")
```

Kết quả là:

A9 fx {=bs_sql("SELECT MA_KH AS [Mã KH], SUM(THANH_TIEN) AS [Tổng tiền]
FROM NKC
WHERE NOTK LIKE '131%'
GROUP BY MA_KH")}

	A	B	C	D	E	F	G
9	Mã KH	Tổng tiền					
10	KH001	78,656,600					
11	KH002	29,926,600					
12	KH003	34,003,200					
13	KH004	51,383,200					

Ví dụ 18: Lập sổ gồm các cột: MA_VLSPHH, Tổng số lượng hàng, điều kiện với LOAI_PHIEU là nhập (N)

```
=bs_sql("SELECT MA_VLSPHH AS [Mã hàng], SUM(SLG) AS [Tổng Nhập]
        FROM KHO
        WHERE LOAI_PHIEU='N'
        GROUP BY MA_VLSPHH ")
```

Ví dụ 19: Lập sổ gồm các cột: MA_VLSPHH, Tên hàng, Tổng số lượng hàng, điều kiện LOAI_PHIEU là nhập (N)

```
=bs_sql("SELECT K.MA_VLSPHH AS [Mã hàng],H.TEN AS [Tên hàng],SUM(K.SLG) AS [Tổng Nhập]
        FROM KHO K INNER JOIN DMVLSPHH H ON K.MA_VLSPHH=H.MA_VLSPHH
        WHERE LOAI_PHIEU='N'
        GROUP BY K.MA_VLSPHH,H.TEN ")
```

A10 fx {=bs_sql("SELECT K.MA_VLSPHH AS [Mã hàng], H.TEN AS [Tên hàng], SUM(K.SLG) AS [Tổng Nhập]
FROM KHO K INNER JOIN DMVLSPHH H ON K.MA_VLSPHH=H.MA_VLSPHH
WHERE LOAI_PHIEU='N'
GROUP BY K.MA_VLSPHH,H.TEN ")}

	A	B	C	D	E	F	G	H	I
9	Mã hàng	Tên hàng	Tổng Nhập						
10	BE1	Bê	5						
11	BM	Bò mông	10						
12	HH001	Máy tính ĐNA L1	16						
13	HH002	Máy tính ĐNA L2	33						
14	HH003	Máy in HP 1200	19						
15	HH004	Máy in HP 1300	14						
16	HH005	Dầu A	2						
17	HH006	Dầu B	3						

➔ Lưu ý: gần như tất cả các cột sau từ khóa SELECT ta đưa vào sau GROUP BY, trừ các cột tính toán.

Ví dụ 20: Lập danh sách gồm mã khách hàng, tên khách hàng với số tiền phát sinh lớn nhất mỗi khách hàng trong sổ KHO, với loại phiếu xuất (X). Sắp xếp theo số tiền giảm dần.

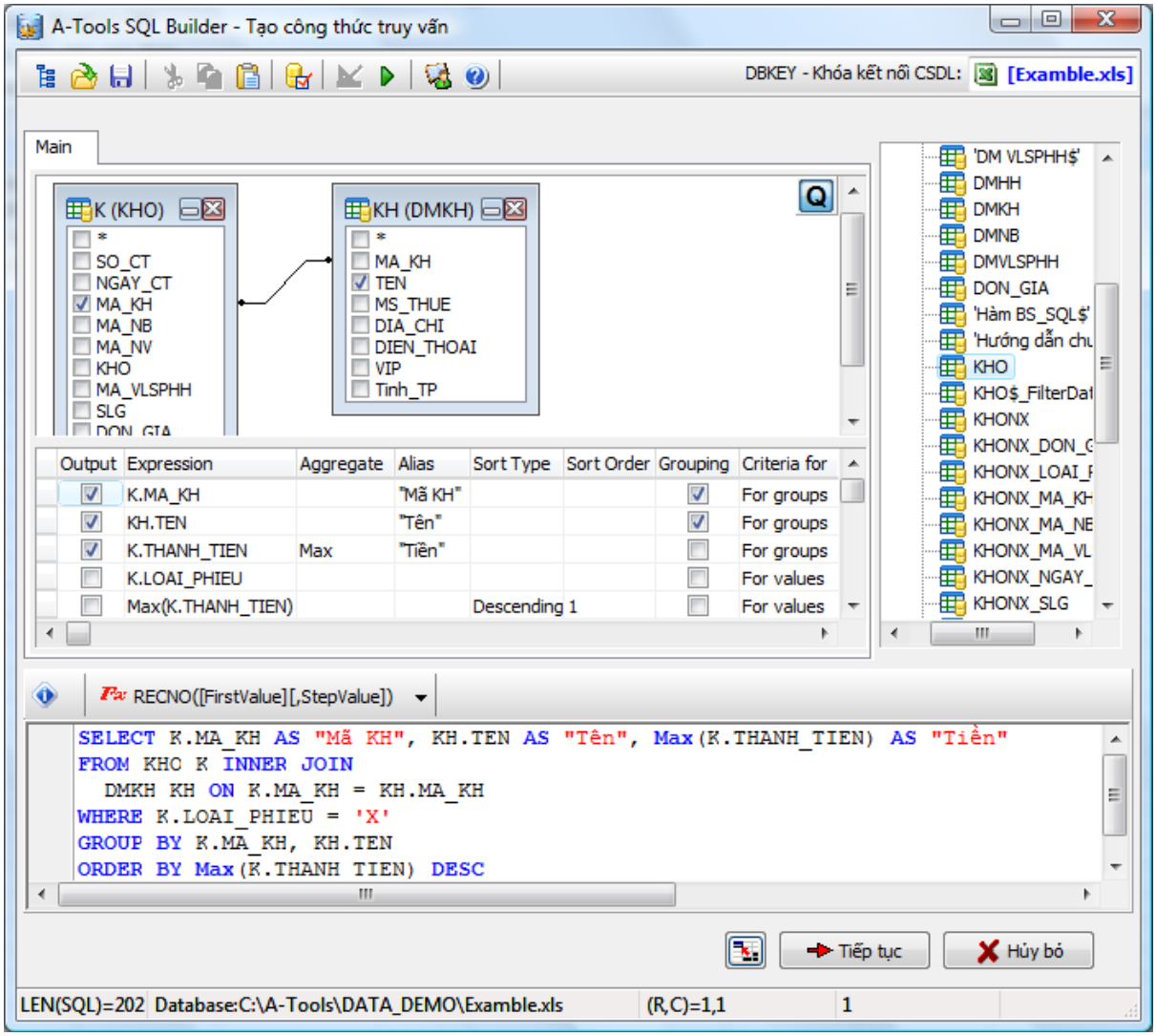
```
=bs_sql("SELECT K.MA_KH AS [Mã KH], KH.TEN AS [Tên], MAX(K.THANH_TIEN) AS [Tiền]
FROM KHO K INNER JOIN DMKH KH ON K.MA_KH = KH.MA_KH
WHERE K.LOAI_PHIEU = 'X'
GROUP BY K.MA_KH, KH.TEN
ORDER BY MAX(K.THANH_TIEN) DESC")
```

➔ ORDER BY MAX(K.THANH_TIEN) DESC → là sắp xếp số tiền phát sinh lớn nhất của mỗi khách hàng giảm dần. Nếu không chỉ ra DESC thì là tăng dần.

A9 fx {=bs_sql("SELECT K.MA_KH AS [Mã KH], KH.TEN AS [Tên], Max(K.THANH_TIEN) AS [Tiền]
FROM KHO K INNER JOIN DMKH KH ON K.MA_KH = KH.MA_KH
WHERE K.LOAI_PHIEU = 'X'
GROUP BY K.MA_KH, KH.TEN
ORDER BY MAX(K.THANH_TIEN) DESC")}

	A	B	C	D	E	F	G	H
9	Mã KH	Tên	Tiền					
10	KH003	Cửa hàng máy tính Đông Đô	19,164,103					
11	KH004	Cty IPQ	17,897,143					
12	KH001	Cty TNHH Hoàng Long	14,700,000					
13	KH002	Cửa hàng Thành Đông	9,323,077					

Tạo trong SQL Builder sẽ là



Sử dụng từ khóa HAVING

HAVING search_condition

search_condition: là điều kiện để truy vấn dữ liệu giống như dùng với từ khoá WHERE nhưng câu điều kiện có sử dụng các hàm thống kê SUM, COUNT, MIN,MAX, AVG.

➔ Sử dụng khai báo HAVING thường đi kế sau với khai báo GROUP BY

Ví dụ 21: Lập danh sách các mã hàng với tổng giá trị nhập về > 60000000

```
=bs_sql("SELECT MA_VLSPHH AS [Mã hàng], SUM(THANH_TIEN)  AS [Tổng tiền]
        FROM KHO
        WHERE LOAI_PHIEU='N'
        GROUP BY MA_VLSPHH
        HAVING SUM(THANH_TIEN)>60000000")
```

A9 fx {=bs_sql("SELECT MA_VLSPHH AS [Mã hàng], SUM(THANH_TIEN) AS [Tổng tiền]
FROM KHO
WHERE LOAI_PHIEU='N'
GROUP BY MA_VLSPHH
HAVING SUM(THANH_TIEN)>60000000")}

	A	B	C	D	E	F	G
9	Mã hàng	Tổng tiền					
10	HH001	79,100,000					
11	HH002	155,564,103					
12	HH004	62,100,000					
13							

Ví dụ 22: Lập danh sách các mã hàng có số lần nhập > 4

```
=bs_sql("SELECT MA_VLSPHH AS [Mã hàng], COUNT(MA_VLSPHH)  AS [Số lần]
        FROM KHO
        WHERE LOAI_PHIEU='N'
        GROUP BY MA_VLSPHH
        HAVING COUNT(MA_VLSPHH)>4")
```

A9 fx {=bs_sql("SELECT MA_VLSPHH AS [Mã hàng], COUNT(MA_VLSPHH) AS [Số lần]
FROM KHO
WHERE LOAI_PHIEU='N'
GROUP BY MA_VLSPHH
HAVING COUNT(MA_VLSPHH)>4")}

	A	B	C	D	E	F	G
9	Mã hàng	Số lần					
10	HH001	5					
11	HH002	8					
12	HH003	5					
13							

Sử dụng từ khóa ORDER BY

ORDER BY order_expression [ASC | DESC]

Từ khóa này cho phép sắp xếp theo các cột tăng dần (ASC) hoặc giảm dần (DESC), ngầm đình là tăng dần.

order_expression: Là danh sách cột được sắp xếp, cột nào đứng trước thì được xếp trước, các cột được cách nhau bởi dấu phảy (,).

Có thể không cần chỉ ra tên cột, thay vào đó ta chỉ vị trí cột trong select_list.

Ví dụ 23: Lấy ra các cột từ sổ KHO sắp xếp theo NGAY_CT và SO_CT

```
=bs_sql("SELECT * FROM KHO ORDER BY NGAY_CT, SO_CT ")
```

G18 fx {=bs_sql("SELECT * FROM KHO ORDER BY NGAY_CT, SO_CT ")}

	A	B	F	G	H	I	J	K
9	SO_CT	NGAY_CT	KHO	MA_VLSPHH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
10	TDKHO	31/07/2005	KHOHH	HH002	6	4,500,000	N	27,000,000
11	TDKHO	31/07/2005	KHOHH	HH003	6	2,800,000	N	16,800,000
12	TDKHO	31/07/2005	KHOHH	HH004	3	4,200,000	N	12,600,000
13	TDKHO	31/07/2005	KHOHH	HH001	4	4,800,000	N	19,200,000
14	NK00000001	01/08/2005	KHOHH	HH001	4	5,000,000	N	20,000,000
15	NK00000001	01/08/2005	KHOHH	HH002	3	4,800,000	N	14,400,000
16	NK00000002	03/08/2005	KHOHH	HH004	2	4,500,000	N	9,000,000
17	NK00000002	03/08/2005	KHOHH	HH002	4	4,800,000	N	19,200,000
18	NK00000002	03/08/2005	KHOHH	HH003	2	3,000,000	N	6,000,000
19	XK00000001	06/08/2005	KHOHH	HH002	2	5,200,000	X	9,323,077
20	XK00000001	06/08/2005	KHOHH	HH003	1	3400000	X	2850000
21	XK00000001	06/08/2005	KHOHH	HH004	2	5200000	X	8640000
22	XK00000001	06/08/2005	KHOHH	HH001	3	5500000	X	14700000
23	XK00000002	08/08/2005	KHOHH	HH004	2	5200000	X	8640000

Ví dụ 24: Chọn tất cả các cột từ sổ KHO, cột THANH_TIEN sắp xếp giảm dần

```
=BS_SQL("SELECT * FROM KHO ORDER BY THANH_TIEN DESC ")
```

Ví dụ 25: Lập danh sách gồm mã khách hàng, tên khách hàng với số tiền phát sinh lớn nhất mỗi khách hàng trong sổ KHO, với loại phiếu xuất (X). Sắp xếp theo số tiền giảm dần.

```
=bs_sql("SELECT K.MA_KH AS [Mã KH], KH.TEN AS [Tên], MAX(K.THANH_TIEN) AS [Tiền]
FROM KHO K INNER JOIN DMKH KH ON K.MA_KH = KH.MA_KH
WHERE K.LOAI_PHIEU = 'X'
GROUP BY K.MA_KH, KH.TEN
ORDER BY MAX(K.THANH_TIEN) DESC")
```

A9 fx {=bs_sql("SELECT K.MA_KH AS [Mã KH], KH.TEN AS [Tên], Max(K.THANH_TIEN) AS [Tiền]
FROM KHO K INNER JOIN DMKH KH ON K.MA_KH = KH.MA_KH
WHERE K.LOAI_PHIEU = 'X'
GROUP BY K.MA_KH, KH.TEN
ORDER BY MAX(K.THANH_TIEN) DESC")}

	A	B	C
9	Mã KH	Tên	Tiền
10	KH003	Cửa hàng máy tính Đông Đô	19,164,103
11	KH004	Cty IPQ	17,897,143
12	KH001	Cty TNHH Hoàng Long	14,700,000
13	KH002	Cửa hàng Thành Đông	9,323,077

Các hàm chuẩn trong ngôn ngữ T-SQL

Các bạn tham khảo tại đây:
http://www.w3schools.com/sql/sql_functions.asp

Khai báo tham số OPTIONS trong hàm BS_SQL

BS_SQL(SQL [,OPTIONS])

Tham số OPTIONS trong hàm BS_SQL có cách khai báo đặc biệt, nó khác với cách truyền tham số thông thường trong các hàm của Excel. OPTIONS là chuỗi khai báo một hay nhiều các tham số cho hàm BS_SQL, các tham số được ngăn cách nhau bởi dấu chấm phẩy (;).

Cách thức khai báo tham số trong OPTIONS:

"PARAM1=value1; PARAM2=value2; PARAM3=value3;…."

Các tham số mà OPTIONS cho phép là:
INSERT, HR, NAME, AUTONAMES(), DBKEY, SERVERSOURCE, OnBeforeUpdate, OnAfterUpdate, OnDblClick, OnSelectionChange, OnGetValue

Tham số INSERT

Cú pháp: INSERT=YES|NO

Nếu là YES, bảng kết quả được chèn vào ô hiện thời (các dữ liệu đứng sau sẽ bị đẩy xuống), NO thì bảng kết quả được ghi đè lên vùng dữ liệu nếu bị tràn.
Nếu không khai báo tham số này, ngầm định hàm BS_SQL đặt INSERT=NO
Khi tạo báo cáo bắt buộc phải đặt INSERT=YES . Mỗi sheet chỉ nên có một báo cáo!

Ví dụ 26: Lấy ra dữ liệu gồm các cột: MA_VLSPHH, SLG, DON_GIA, THANH_TIEN với mã hàng là HH001. Dùng thuộc tính chèn dòng.

```
=bs_sql("SELECT MA_VLSPHH, SLG, DON_GIA, THANH_TIEN FROM KHO WHERE
MA_VLSPHH='HH001'  ", "INSERT=YES")
```

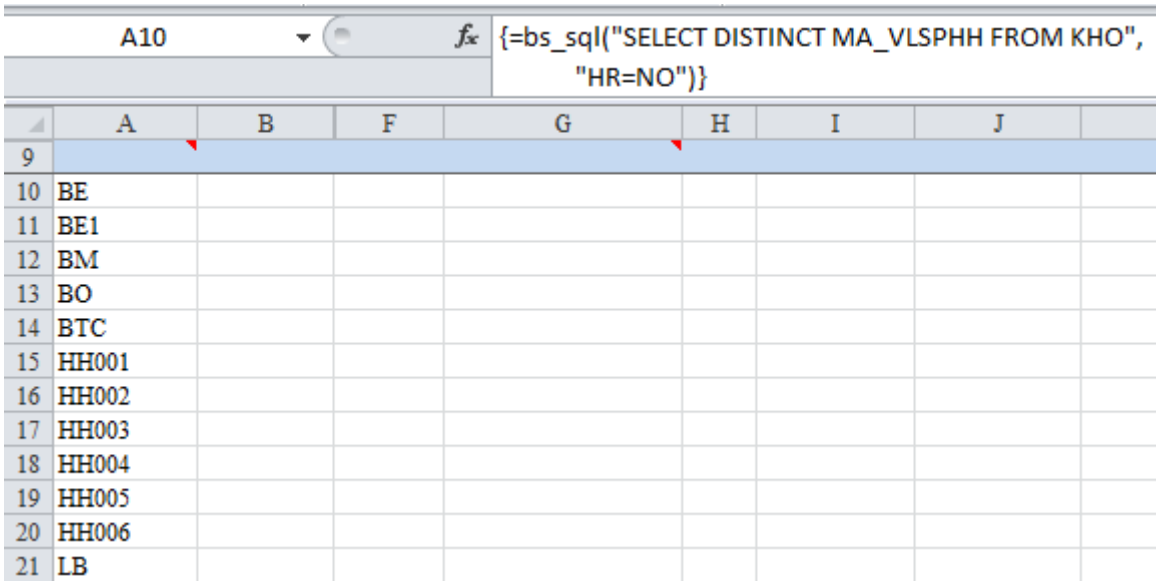
Khi bạn thay đổi HH001 thành HH002 thì dòng kết quả tự co giãn và không ảnh hưởng tới các dòng dữ liệu bên dưới bảng kết quả. Tham số INSERT cần dùng khi làm báo cáo.

Tham số HR

Cú pháp: HR=YES|NO
Nếu là YES (ngầm định), kết quả bảng dữ liệu có dòng tiêu đề, NO thì không có.
Nếu không khai báo tham số này, ngầm định hàm BS_SQL đặt HR=YES

Ví dụ 27: lấy ra mã hàng duy nhất trong bảng KHO. Bảng kết quả không có dòng tiêu đề

```
=bs_sql("SELECT DISTINCT MA_VLSPHH FROM KHO",
        "HR=NO")
```



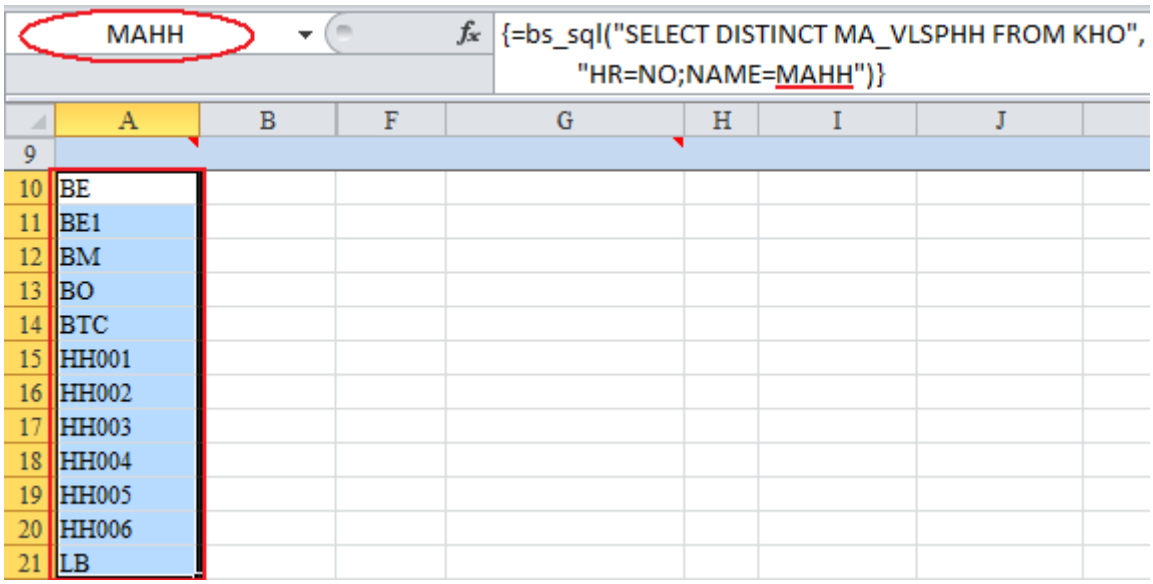
Tham số NAME

Cú pháp: NAME = Tên vùng

Cho phép đặt tên vùng kết quả theo tên được khai báo. Bình thường trong Excel ta phải làm: chọn (bôi đen) vùng, nhấn CTRL+F3 và tạo tên. Add-in A-Tools sẽ tự làm việc này cho bạn. Khi khai báo NAME vùng kết quả sẽ tự được tạo tham chiếu đùng vào vùng dữ liệu của nó.

Ví dụ 28: lấy ra mã hàng duy nhất trong bảng KHO. Bảng kết quả không có dòng tiêu đề. Đặt tên bảng là MAHH

```
=bs_sql("SELECT DISTINCT MA_VLSPHH FROM KHO",
        "HR=NO; NAME=MAHH")
```



Với cách thức tạo danh sách duy nhất đồng thời tạo NAME "như MAHH như trên rất tiện cho việc tạo Validation dạng List. Nó phục vụ cho thiết kế các mục chọn mã trong báo cáo.

Tham khảo cách tạo Validation List trong Excel tại đây:

http://handung107-giaiphapexcel.blogspot.com/2008/11/excel-hack-17.html

Tham số AUTONAMES

Cú pháp hàm: AUTONAMES([Tên vùng] [,Chuỗi mẫu])
Hàm AUTONAMES tự động tạo Name (Tên vùng) cho các cột dữ liệu trong bảng kết quả.
Tên vùng: là tên bảng kết quả sẽ được đặt tên. Tham số này thể khai báo hoặc không.
Chuỗi mẫu: tên các cột dữ liệu sẽ được đặt với tên bắt đầu bởi "Chuỗi mẫu". Tham số này có thể khai báo hoặc không.

Ví dụ 29: Lấy ra các cột NGAY_CT, MA_VLSPHH, THANH_TIEN từ sổ KHO. Bảng kết quả được đặt tên là "DULIEU", các cột trong bảng kết quả được đặt tên với nhóm ký tự đầu là "DL_"

các cột được đặt tự động với tên bắt đầu là "DL_" .

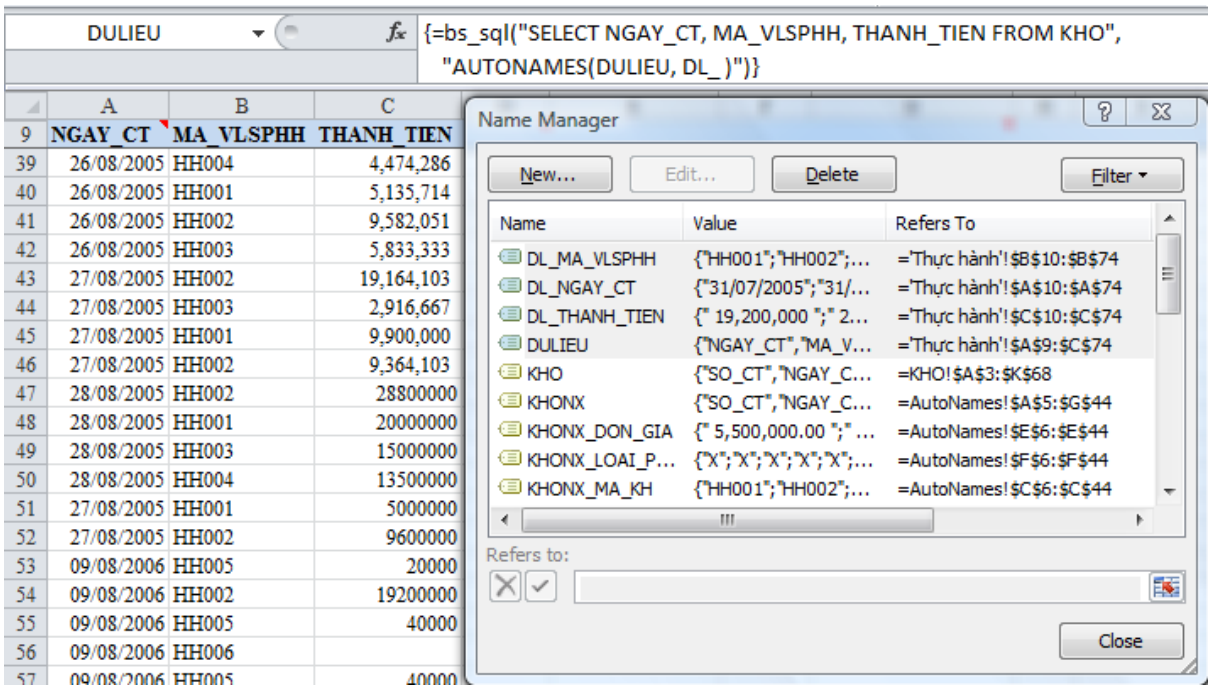
```
=bs_sql("SELECT NGAY_CT, MA_VLSPHH, THANH_TIEN FROM KHO",
        "AUTONAMES(DULIEU, DL_ )")
```

Sau khi chạy công thức trên, A-Tools sẽ tạo các NAME:

DULIEU
DL_NGAY_CT
DL_MA_VLSPHH
DL_THANH_TIEN

Bạn có thể sử dụng các NAME tạo như trên cho các mục đích khác. Ví dụ tạo công thức tính tổng vùng thành tiền của cột dữ liệu vừa trả về bởi công thức trên .
=SUM(DL_THANH_TIEN)

Như vậy nếu sử dụng AUTONAMES(DULIEU) thì không cần khai báo NAME=DULIEU



Vẫn ví dụ trên nhưng tên các cột được đặt tự động. Bảng dữ liệu không được đặt tên.

```
=bs_sql("SELECT NGAY_CT, MA_VLSPHH, THANH_TIEN FROM KHO",
        "AUTONAMES()")
```

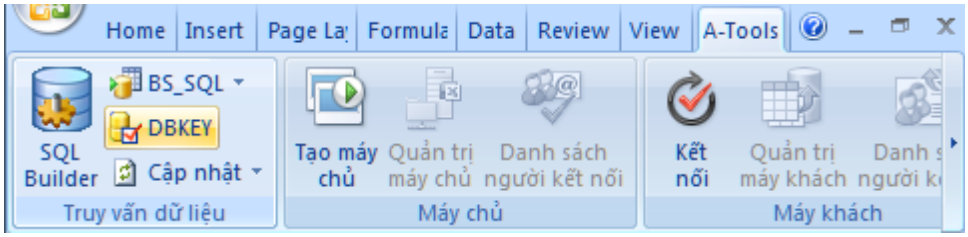
Tham số DBKEY – Liên kết với CSDL bên ngoài vào bảng tính Excel

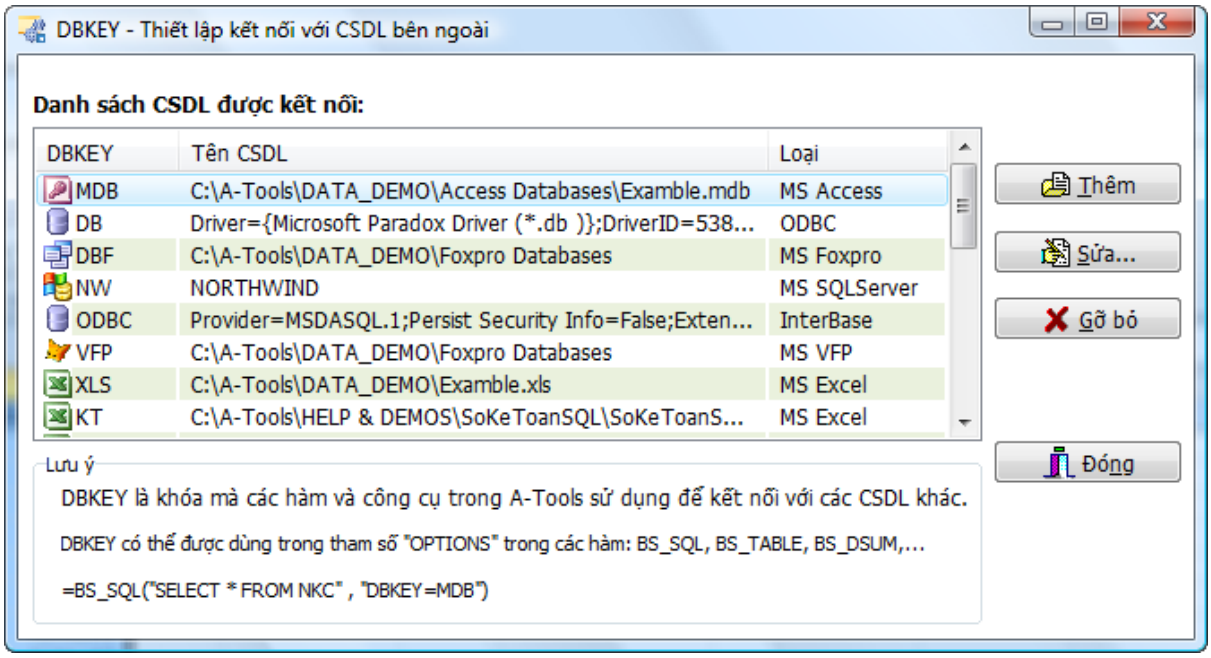
Add-in A-Tools cho phép kết nối với các CSDL ngoài: MS Access, Foxpro, MS SQL, MySQL,…Để kết nối vào bảng tính Excel thông qua mã kết nối DBKEY.

Cách tạo DBKEY

Với Excel 2003 vào menu "A-Tools"->"Truy vấn dữ liệu" chọn "DBKEY - Thiết lập các kết nối với CSDL bên ngoài".

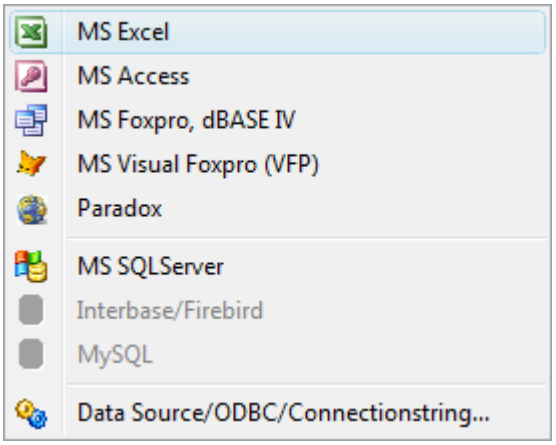
Với Excel 2007 hoặc cao hơn làm theo menu dưới đây:





+ DBKEY: mã của kết nối. Tên DBKEY được sử dụng trong các hàm của A-Tools như BS_SQL, BS_TABLE, BS_DSUM,…

+ Thêm: tạo thêm DBKEY (thêm kết nối)



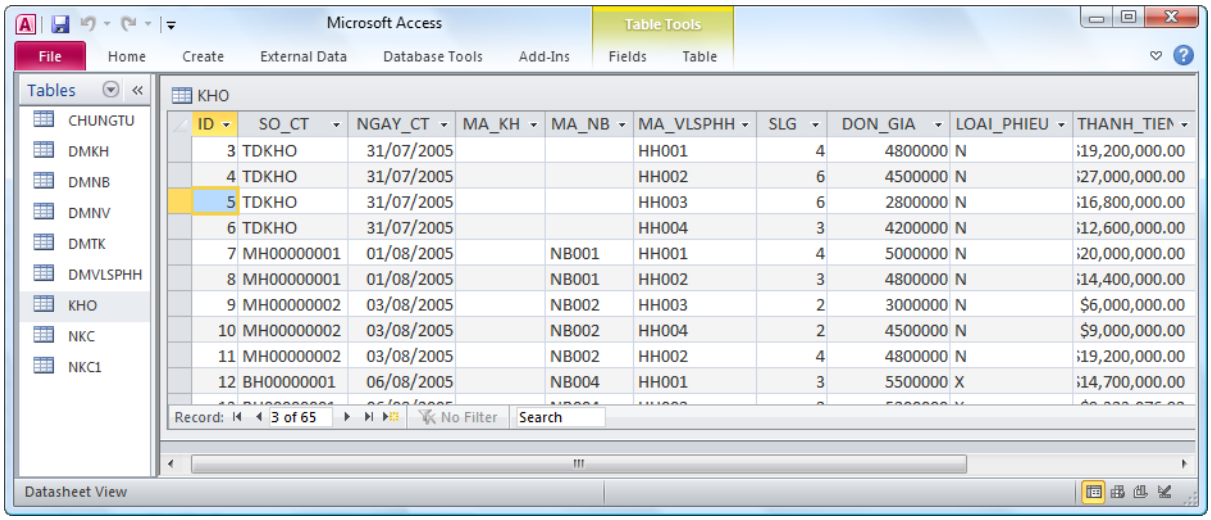
Tùy vào loại CSDL mà ta chọn các mục tạo tương ứng. Riêng mục "ODBC/ConnectionString" ta có thể tạo mã DBKEY kết nối tới bất kỳ CSDL nào mà Windows cho phép.

Nhìn trong bảng quản trị DBKEY trên, ta thấy có mã DBKEY là "MDB" kết nối tới tập tin CSDL Access "Examble.mdb". Ta sẽ có ví dụ kết nối tới CSDL này sau.

+ Sửa: sửa lại các thong số thiết lập DBKEY

+ Gỡ bỏ: gỡ bỏ DBKEY khỏi A-Tools. Nếu DBKEY bị gỡ bỏ, các công thức sử dụng tới DBKEY này đều bị lỗi.

Ví dụ 30: Kết nối với CSDL bên ngoài. Mở tập tin "C:\A-Tools\DATA_DEMO\Access Databases\Examble.mdb" bằng MS Access. Mở table KHO ta có màn hình bên dưới



ID	SO_CT	NGAY_CT	MA_KH	MA_NB	MA_VLSPHH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
3	TDKHO	31/07/2005			HH001	4	4800000	N	19,200,000.00
4	TDKHO	31/07/2005			HH002	6	4500000	N	27,000,000.00
5	TDKHO	31/07/2005			HH003	6	2800000	N	16,800,000.00
6	TDKHO	31/07/2005			HH004	3	4200000	N	12,600,000.00
7	MH00000001	01/08/2005		NB001	HH001	4	5000000	N	20,000,000.00
8	MH00000001	01/08/2005		NB001	HH002	3	4800000	N	14,400,000.00
9	MH00000002	03/08/2005		NB002	HH003	2	3000000	N	$6,000,000.00
10	MH00000002	03/08/2005		NB002	HH004	2	4500000	N	$9,000,000.00
11	MH00000002	03/08/2005		NB002	HH002	4	4800000	N	19,200,000.00
12	BH00000001	06/08/2005		NB004	HH001	3	5500000	X	14,700,000.00

Một mã DBKET là "MDB" đã được tạo và kết nối tới tập tin "Examble.mdb". Ta có thể truy vấn dữ liệu từ tập tin này qua DBKEY=MDB. Điều kiện truy vấn là lấy toàn bộ dữ liệu trong table KHO, với loại chứng từ là nhập 'N'.

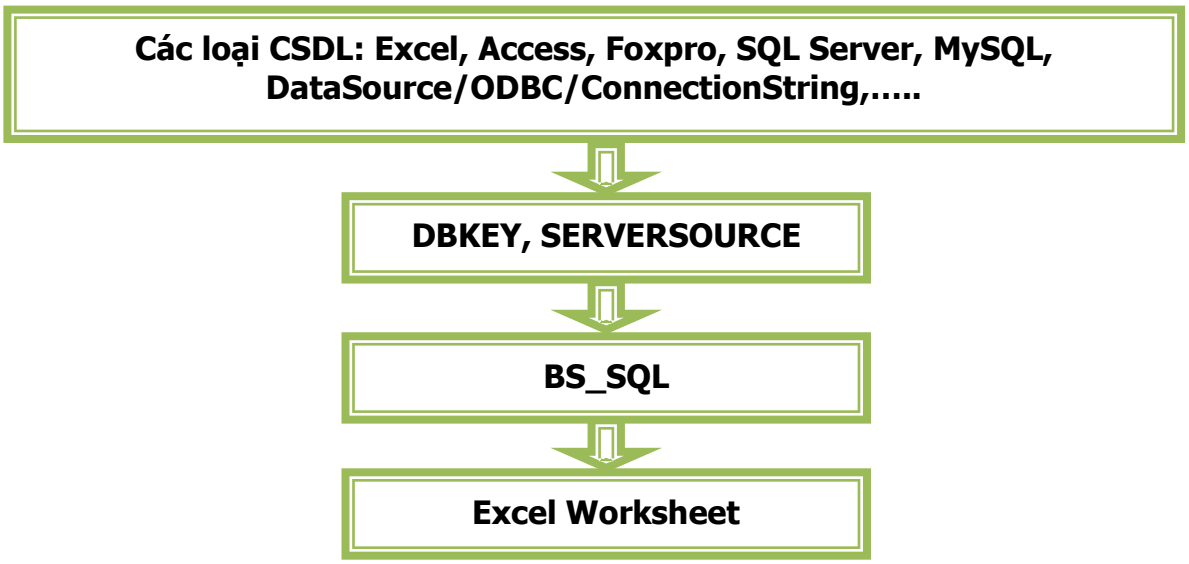
```
=BS_SQL("SELECT * FROM KHO WHERE LOAI_PHIEU='N' " , "DBKEY=MDB")
```

Kết quả trong bảng tính Excel được như sau:

A4 {=bs_sql("SELECT * FROM KHO WHERE LOAI_PHIEU='N' ", "DBKEY=MDB")}

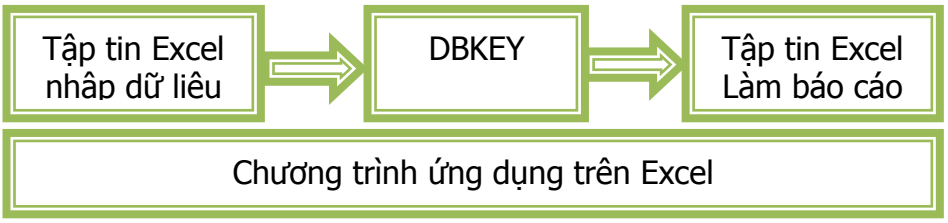
	A	B	C	D	E	G	H	I	J	K	L	N
1	ID	SO_CT	NGAY_CT	MA_KH	MA_NB	MA_KHO	MA_VLSPI	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN	
2	3	TDKHO	38564			KHOHH	HH001	4	4800000	N	19200000	
3	4	TDKHO	38564			KHOHH	HH002	6	4500000	N	27000000	
4	5	TDKHO	38564			KHOHH	HH003	6	2800000	N	16800000	
5	6	TDKHO	38564			KHOHH	HH004	3	4200000	N	12600000	
6	7	MH000000	38565		NB001	KHOHH	HH001	4	5000000	N	20000000	
7	8	MH000000	38565		NB001	KHOHH	HH002	3	4800000	N	14400000	
8	9	MH000000	38567		NB002	KHOHH	HH003	2	3000000	N	6000000	

Mô hình kết nối các CSDL ngoài qua DBKEY vào bảng tính Excel



Xem thêm video tạo DBKEY
http://bluesofts.net/giai-phap-phan-mem/a-tools-truy-van-loc-soan-thao-va-quan-tri-du-lieu-excel-qua-mang/180-tao-dbkey-ket-noi-du-lieu.html

Khi người dùng đã nắm được cách tạo và kết nối với DBKEY thì có thể có những thiết kế tập tin báo cáo Excel chuyên nghiệp hơn. Có thể thiết kế tập tin nhập dữ liệu riêng. Tập tin này được kết nối vào một DBKEY. Một tập tin Excel khác để làm báo cáo bởi hàm BS_SQL và kết nối qua DBKEY tới tập tin Excel dữ liệu. Với phương pháp này tập tin Excel sẽ có dung lượng nhẹ và chạy rất nhanh.



Phương pháp làm báo cáo cho nhiều doanh nghiệp cùng mô hình

Ví dụ bạn làm kế toán cho 3 công ty, mô hình các công ty này giống nhau nên cấu trúc CSDL giống nhau. Mỗi công ty bạn tạo một tập tin CSDL Excel riêng. Mỗi tập tin được kết nối vào một DBKEY, ví dụ các DBKEY: CT1, CT2, CT3. Bạn tạo

một tập tin Excel để làm báo cáo. Công thức trong tập tin báo cáo này dùng hàm BS_SQL kết nối tới các CSDL ngoài bởi DBKEY. Chọn một ô trong một sheet nào đó để khai báo mã DBKEY. Như là lấy ô A1 để khai báo ";DBKEY=CT1". Vậy các công thức làm báo cáo làm theo mẫu dưới đây:

=BS_SQL("SELECT …FROM…", "Các tham số;" & A1)

A1 chứa giá trị ";DBKEY=CT1" vì thế công thức luôn lấy dữ liệu của CT1 – Công ty 1. Nếu cần mở báo cáo cho CT2 – Công ty 2 chỉ cần thay **A1** là ";DBKEY=CT2".

Tham số SERVERSOURCE: Truy vấn dữ liệu từ máy chủ

Để chạy tham số này thì máy chủ A-Tools phải đang mở. Nếu máy chủ A-Tools chưa chạy, vào menu A-Tools chạy "Start Server"/"Tạo máy chủ". Thực hiện các việc nạp CSDL Excel để chia sẻ và thiết lập các tham số cần thiết.

Xem thêm video:

http://bluesofts.net/giai-phap-phan-mem/a-tools-truy-van-loc-soan-thao-va-quan-tri-du-lieu-excel-qua-mang/184-huong-dan-chia-se-file-excel-qua-mang-lan-internet-phan-1-.html

Cú pháp: SERVERSOURCE=YES|NO

Từ máy khách dùng hàm BS_SQL với việc gán tham số **SERVERSOURCE=YES** khi đó dữ liệu sẽ được lấy từ máy chủ trả về cho máy khách. Nếu không khai báo thì tương đương với SERVERSOURCE=NO, dữ liệu sẽ lấy của máy tính hiện tại đang chạy (localhost).

Giả sử máy chủ đang chạy và đang mở một workbook có vùng dữ liệu tên là "KHO". Tại máy khách, để lấy dữ liệu trong máy chủ thì công thức như sau:

```
=BS_SQL("SELECT * FROM KHO" , "SERVERSOURCE=YES")
```

Nếu máy chủ đang có mã DBKEY là "XLS" kết nối tới tập tin Excel "Examble.xls". Tại máy khách, để lấy dữ liệu trong máy chủ thì công thức như sau:

```
=BS_SQL("SELECT * FROM KHO" , "DBKEY=XLS; SERVERSOURCE=YES")
```

- ➔ Nếu hàm BS_SQL lập trong một tập tin và tập tin này được chia sẻ từ máy chủ. Khi máy khách kết nối vào máy chủ rồi mở tập tin này thì hàm BS_SQL tự động truy vấn dữ liệu của máy chủ và trả kết quả về máy khách mặc dù không cần khai báo SERVERSOURCE=YES.

Các tham số chạy thủ tục sự kiện Macro/VBA

Các tham số là:
OnBeforeUpdate, OnAfterUpdate, OnDblClick, OnSelectionChange, OnGetValue

- ➔ Nếu bạn không biết lập trình VBA thì bỏ qua các tham số này.
- ➔ Xem các ví dụ trong tập tin Excel C:\A-Tools\HELP & DEMOS\A-Tools VBA Programming\Report 3 So ke toan A-Tools VBA.xls

Có thể lập trình các macro trong VBA để chạy trong các sự kiện của hàm BS_SQL, khai báo chạy các macro thực hiện trong tham số OPTIONS. Các tham số được khai báo như sau:
+ Cho phép chạy một macro trước khi cập nhật dữ liệu vào bảng tính

OnBeforeUpdate = YourMacro
+ Cho phép chạy một macro sau khi dữ liệu được cập nhật
OnAfterUpdate = YourMacro
+ Cho phép chạy một macro khi nhấp đúp chuột vào vùng dữ liệu (có hàm BS_SQL)
OnDblClick = YourMacro
+ Cho phép chạy một macro khi con trỏ ô di chuyển
OnSelectionChange = YourMacro
+ Cho phép chạy một macro (hàm) để thay đổi giá trị trong quá trình nhận dữ liệu từ kết quả truy vấn SQL.
OnGetValue = YourMacro

Để viết macro, bạn phải tạo Module trong VBE (nếu chưa có). Phần đầu module cần đưa khai báo hai hàm API của Add-in A-Tools là GetFieldNames, SetDataValue để can thiệp vào phần tử mảng kết quả trả về của hàm BS_SQL. Khai báo như sau:

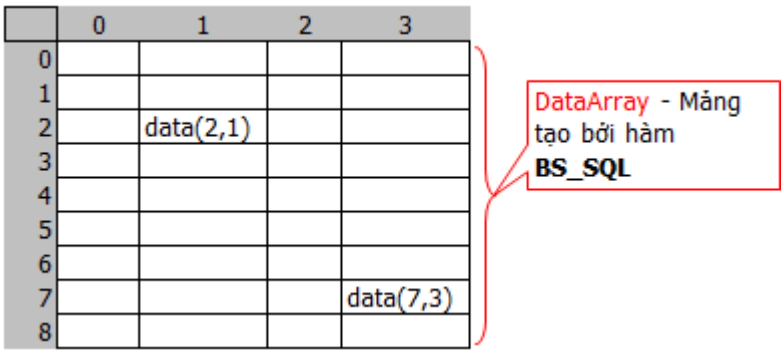
```
#If VBA7 Then
Declare PtrSafe Function GetFieldNames Lib "AddinATools.dll" (ByRef FieldNames) As Long
Declare PtrSafe Function SetDataValue Lib "AddinATools.dll" (ByVal Row As Long, ByVal Column As Long, ByVal Value As Variant) As Long
#Else
Declare Function GetFieldNames Lib "AddinATools.dll" (ByRef FieldNames) As Long
Declare Function SetDataValue Lib "AddinATools.dll" (ByVal Row As Long, ByVal Column As Long, ByVal Value As Variant) As Long
#End If
```

Cách khai báo và viết macro theo từng dạng thủ tục sự kiện

<u>Cho phép chạy một macro trước khi cập nhật dữ liệu vào bảng tính</u>

Khai báo trong tham số OPTIONS của hàm BS_SQL**: OnBeforeUpdate** = DoBeforeUpdate
=BS_SQL("chuỗi khai báo SQL", "…;**OnBeforeUpdate** = DoBeforeUpdate)
Trong môi trường lập trình (VBE), viết thủ tục "DoBeforeUpdate" trong một module theo cấu trúc sau

```
Sub DoBeforeUpdate(ByVal OldDataTable As Range, ByVal NewDataTable As Range, ByVal DataArray)
    'Có thể đổi tên thủ tục
    '+ OldDataTable: là đối tượng quản lý vùng dữ liệu cũ (trước khi hàm BS_SQL chạy)
    '+ NewDataTable: là đối tượng quản lý vùng dữ liệu mới (sau khi hàm BS_SQL chạy)
    '+ DataArray: la mảng 2 chiều. Các phần tử dòng và cột nhận giá trị từ 0
    'Giá trị tại dòng 1 cột 2 của mảng là DataArray(1,2) , trên bảng tính Excel thì là giá trị của dòng 2 cột 3.
End Sub
```



Ví dụ:

```
Sub DoBeforeUpdate(ByVal OldDataTable As Range, ByVal NewDataTable As Range, ByVal DataArray)
  Const Column = 6
  Dim Row As Long
  MsgBox "OldDataTable:" & OldDataTable.Address & Chr(13) & "NewDataTable:" & NewDataTable.Address, , "Options: OnBeforeUpdate=DoBeforeUpdate"
  For Row = LBound(DataArray, 1) To UBound(DataArray, 1)
    If Row = 0 Then
      DataArray(Row, Column) = DataArray(Row, 5)
    Else
      DataArray(Row, Column) = DataArray(Row - 1, Column) + DataArray(Row, 5)
    End If
    SetDataValue Row, Column, DataArray(Row, Column)
  Next Row
End Sub
```

Cho phép chạy một macro sau khi cập nhật dữ liệu vào bảng tính

Khai báo trong tham số OPTIONS của hàm BS_SQL**: OnAfterUpdate** = DoAfterUpdate

=BS_SQL("chuỗi khai báo SQL", "…;**OnAfterUpdate** = DoAfterUpdate)

Trong môi trường lập trình (VBE), viết thủ tục "DoAfterUpdate" trong một module theo cấu trúc sau

```
Sub DoAfterUpdate(ByVal DataTable As Range)
      'Có thể đổi tên thủ tục
     `+ DataTable tương tự như NewDataTable trong DoBeforeUpdate, là vùng dữ liệu trong bảng tính Excel có kết quả trả về của hàm BS_SQL.
End Sub
```

Cho phép chạy một macro khi nhấp đúp chuột trong vùng công thức

Khai báo trong tham số OPTIONS của hàm BS_SQL**: OnDblClick** = DoDblClick

=BS_SQL("chuỗi khai báo SQL", "…;**OnDblClick** = DoDblClick)

Trong môi trường lập trình (VBE), viết thủ tục "DoAfterUpdate" trong một module theo cấu trúc sau

```
Sub DoDblClick(ByVal DataTable As Range, ByVal Row As Integer, ByVal Column As Integer)
      'Có thể đổi tên thủ tục
      'Row, Column là trị số dòng, cột trong vùng DataTable
End Sub
```

Ví dụ: Nhấp đúp chuột vào vùng công thức, tự động mở sheet "Demo" và trong sheet Demo, gán ô C4=DataTable(1, Column), C5=DataTable(Row, 1)

```
Sub DoDblClick(ByVal DataTable As Range, ByVal Row As Integer, ByVal Column As Integer)
  Dim WS As Worksheet
  If Row = 1 Or Column < 3 Then
    Exit Sub
  End If
  Set WS = Sheets("Demo")
  WS.Select
On Error GoTo Done:
  Application.Calculation = xlCalculationManual
  Application.EnableEvents = False
  WS.Range("C4").Value = DataTable(1, Column)
  WS.Range("C5").Value = DataTable(Row, 1)
  WS.Range("C5").Select
Done:
  Application.EnableEvents = True
  Application.Calculation = xlCalculationAutomatic
End Sub
```

Cho phép chạy một macro khi di chuyển ô chọn trong vùng công thức

Khai báo trong tham số OPTIONS của hàm BS_SQL**: OnSelectionChange** = DoSelectionChange
=BS_SQL("chuỗi khai báo SQL", "…;**OnSelectionChange** = DoSelectionChange)
Trong môi trường lập trình (VBE), viết thủ tục "DoSelectionChange" trong một module theo cấu trúc sau

```
Sub DoSelectionChange(ByVal DataTable As Range, ByVal Row As Integer, ByVal Column
As Integer)
      'Có thể đổi tên thủ tục
End Sub
```

Ví dụ:

```
Sub DoSelectionChange(ByVal DataTable As Range, ByVal Row As Integer, ByVal Column As
Integer)
  Application.Caption = Row & ":" & Column & " = " & DataTable(Row, Column)  'ActiveCell.Value
End Sub
```

Cho phép chạy một macro (hàm) khi nhận giá trị cho từng phần tử của mảng kết quả trả về

Khai báo trong tham số OPTIONS của hàm BS_SQL**: OnGetValue**=GetValue
=BS_SQL("chuỗi khai báo SQL", "…;**OnGetValue**=GetValue)
Trong môi trường lập trình (VBE), viết thủ tục "GetValue" trong một module theo cấu trúc sau

```
Function GetValue(ByVal DataArray, ByVal Row As Integer, ByVal Column As Integer, ByVal
Value As Variant)
      'Có thể đổi tên hàm
      'GetValue = NewValue
End Function
```

Ví dụ: Tại cột 2 của mảng giá trị được thêm số dòng và chuỗi www.bluesofts.net, cột 6 thì tính lũy kế

```
Function GetValue(ByVal DataArray, ByVal Row As Integer, ByVal Column As Integer, ByVal Value As
```

```
Variant) As Variant
  If Column = 2 Then
    GetValue = Row & " " & "[www.Bluesofts.net] " & Value
  End If
  If Column = 6 Then
    If Row = 0 Then
      GetValue = DataArray(Row, 5)
    Else
      GetValue = DataArray(Row - 1, Column) + DataArray(Row, 5)
    End If
  End If
End Function
```

Giải thích các tham số trong các thủ tục sự kiện:

+ DataTable: Là vùng dữ liệu trên sheet, ở đó chứa kết quả trả về của hàm BS_SQL, dòng (Row) và cột (Column) nhận giá trị từ 1
+ DataArray: Là mảng (2 chiều R:C) , ở đó chứa giá trị (lưu trong bộ nhớ) mà hàm BS_SQL trả về bảng tính. Trong bộ nhớ, dòng (Row) và cột (Column) của mảng nhận giá trị từ 0
+ Row, Column: Là dòng, cột của DataTable (giá trị nhỏ nhất là 1) hoặc DataArray (giá trị nhỏ nhất là 0)
+ OldDataTable: Là vùng dữ liệu cũ trên sheet (trước khi chạy hàm BS_SQL)
+ NewDataTable: Là vùng dữ liệu mới trên sheet (sau khi chạy hàm BS_SQL)

Các hàm quan trọng phối hợp với hàm BS_SQL trong lập báo

Để tạo một báo cáo hoàn chỉnh Add-in A-Tools cung cấp thêm một số hàm quan trọng phối hợp với hàm BS_SQL để trích lọc dữ liệu và làm báo cáo. Các hàm là: RECNO, CELL, RANGE, DATA, GETONCE, FORMULA, FIELDIF, VTC, SQL.

➔ Trước khi đọc phần này các bạn hãy mở tập tin C:\A-Tools\HELP & DEMOS\Bai 2 - Cac ham ho tro nhom ham Database Functions.xls

Hàm RECNO: Tạo số thứ tự

Cấu trúc: RECNO([FirstValue][,StepValue])

FirstValue: là một số nguyên được gán cho dòng đầu tiên, tham số này nếu không nhập giá trị ngầm định là 1.

StepValue: là một số tạo bước nhảy, tham số này giá trị ngầm định là 1.

Ví dụ 31: Đánh số thứ tự liên tiếp 1,2,…

```
=bs_sql("SELECT RECNO() AS [STT],* FROM DMVLSPHH")
```

B10 | fx {=bs_sql("SELECT RECNO() AS [STT],* FROM DMVLSPHH")}

	A	B	C	D	E	F	G
10		STT	MA_VLSPHH	Ten	Dvi	Loai	
11		1	BE	Bê	Đĩa	HH	
12		2	BE1	Bê	Kg	VL	
13		3	BM	Bò mông	Kg	VL	
14		4	BO	Bò xào nướng	Đĩa	HH	
15		5	BTC	Bê tái chanh	Đĩa	HH	
16		6	HH001	Máy tính ĐNA L1	Chiếc	HH	
17		7	HH002	Máy tính ĐNA L2	Chiếc	HH	
18		8	HH003	Máy in HP 1200	Chiếc	HH	
19		9	HH004	Máy in HP 1300	Chiếc	HH	

Ví dụ 32: Đánh số thứ tự từ 1, bước nhảy là 2. Kết quả là 1,3,5,…

```
=bs_sql("SELECT RECNO(1,2) AS [STT],* FROM DMVLSPHH")
```

Ví dụ 33: Đánh số thứ tự liên tiếp từ 100

```
=bs_sql("SELECT RECNO(100) AS [STT],* FROM DMVLSPHH")
```

Hàm CELL: Lấy giá trị tại một ô hoặc NAME trong bảng tính

Hàm CELL lấy giá trị tại địa chỉ ô chỉ định. Hàm CELL giúp ta xây dựng công thức SQL tham chiếu tới giá trị của một ô trong bảng tính, đây là điều kiện làm cho báo cáo động. Khi thay giá trị ở ô được tham chiếu, đặt con trỏ tại vùng công thức BS_SQL nhấn CTRL+SHIFT+A để cập nhật.

Hàm CELL có hai cấu trúc:

+ Cấu trúc 1: CELL(AddressCell [,ValueType [,DateFmt]])

Nhận giá trị tại địa chỉ "AddressCell" trong bảng tính

+ Cấu trúc 2: CELL(R,C [,ValueType [,DateFmt]])

AddressCell: địa chỉ ô

R, C: là các giá trị của dòng và cột ở đó hàm CELL nhận giá trị.

ValueType: nhận một trong các giá trị

T - Nếu xác định giá trị là kiểu văn bản (Text)

D - Nếu xác định giá trị là kiểu ngày (Date)

N - Nếu xác định giá trị là kiểu số (Number)

Nếu bỏ qua, A-Tools tự động xác định kiểu giá trị (theo Format Cell)

DateFmt: Là định dạng ngày hệ thống sử dụng làm chuẩn, ngầm định nhận giá trị MM/dd/yy . Dù bạn đang sử dụng kiểu ngày tháng dd/MM/yy cũng không nên thay đổi, chỉ thay đổi nếu thực sự cần thiết. Tham số này chỉ sử dụng khi ValueType là D (xác định giá trị trong AddressCell là kiểu Date)

Ví dụ 34: Hàm CELL – Điều kiện tham chiếu tới ô chứa giá trị văn bản trong bảng tính

Giả thiết B10 chứa giá trị HH001. Cần lấy tất cả dữ liệu từ bảng KHO mà mã hàng tại ô B10.

```
=bs_sql("SELECT * FROM KHO WHERE MA_VLSPHH=CELL(B10)")
```

A16 | {=bs_sql("SELECT * FROM KHO WHERE MA_VLSPHH=CELL(B10)")}

	A	B	C	D	E	G	H	I	J	K	L
10	Mã hàng	HH001									
16	ID	SO_CT	NGAY_CT	MA_KH	MA_NB	MA_KHO	MA_VLSPH	SLG	DON_GIA	LOAI_PHIE	THANH_TIEN
17	3	TDKHO	31/07/2005			KHOHH	HH001	4	4800000	N	19200000
18	7	MH00000001	01/08/2005		NB001	KHOHH	HH001	4	5000000	N	20000000
19	12	BH00000001	06/08/2005		NB004	KHOHH	HH001	3	5500000	X	14700000
20	20	BH00000003	14/08/2005		NB002	KHOHH	HH001	3	5500000	X	14700000
21	21	MH00000003	20/08/2005		NB004	KHOHH	HH001	2	5000000	N	10000000
22	31	BH00000005	26/08/2005	KH001		KHOHH	HH001	2	5800000	X	9900000
23	33	BHTL000000	27/08/2005	KH001		KHOHH	HH001	2	5500000	N	9900000
24	36	MH00000005	28/08/2005		NB002	KHOHH	HH001	4	5000000	N	20000000
25	39	BH00000006	26/08/2005	KH004		KHOHH	HH001	1	5500000	X	5112500
26	44	MHTL000001	27/08/2005		NB001	KHOHH	HH001	1	5000000	X	5112500

Có thể thay CELL(B10) thành CELL(B10,T) để A-Tools hiểu chắc chắn B10 là văn bản. Vì có trường hợp là văn bản nhưng người dùng định dạng B10 là General và nhập số.

Có thể thay CELL(B10) thành CELL(10,2) – dòng 10, cột 2, theo cấu trúc 2.

Ví dụ 35: Hàm CELL – Lấy dữ liệu từ sổ KHO mà MA_VLSPHH là mã trong ô B10 VÀ số lượng > số lượng ở ô B11. Giả thiết B11 chứa giá trị 2.

```
=bs_sql("SELECT * FROM KHO WHERE MA_VLSPHH=CELL(B10) AND SLG>CELL(B11)")
```

Có thể thay CELL(B11) thành CELL(B11,N) để ép về kiểu số.

Ví dụ 36: Hàm CELL – Lấy dữ liệu từ sổ KHO mà MA_VLSPHH là mã trong ô B10 VÀ số lượng > số lượng ở ô B11 VÀ ngày>ngày ở ô B12. Giả thiết B11 chứa giá trị 2, B12 chứa giá trị 25/06/2005.

```
=bs_sql("SELECT * FROM KHO WHERE MA_VLSPHH=CELL(B10) AND SLG>CELL(B11) AND
NGAY_CT>CELL(B12)")
```

Có thể thay CELL(B11) thành CELL(B11,N) để ép về kiểu số, CELL(B12) thành CELL(B12,D) để ép về kiểu Date.
Kết quả là:

A16 fx {=bs_sql("SELECT *
FROM KHO
WHERE MA_VLSPHH=CELL(B10) AND SLG>CELL(B11) AND NGAY_CT>CELL(B12)")}

	A	B	C	D	E	G	H	I	J	K	L
10	Mã hàng	HH001									
11	Số lượng>	2									
12	Ngày>	25/06/2005									
16	ID	SO_CT	NGAY_CT	MA_KH	MA_NB	MA_KHO	MA_VLSPH	SLG	DON_GIA	LOAI_PHIE	THANH_TIEN
17	3	TDKHO	31/07/2005			KHOHH	HH001	4	4800000	N	19200000
18	7	MH00000001	01/08/2005		NB001	KHOHH	HH001	4	5000000	N	20000000
19	12	BH00000001	06/08/2005		NB004	KHOHH	HH001	3	5500000	X	14700000
20	20	BH00000003	14/08/2005		NB002	KHOHH	HH001	3	5500000	X	14700000
21	36	MH00000005	28/08/2005		NB002	KHOHH	HH001	4	5000000	N	20000000

Hàm RANGE: Lấy giá trị tại một ô hoặc NAME trong bảng tính

Hàm RANGE lấy giá trị tại địa chỉ ô chỉ định. Hàm RANGE giúp ta xây dựng công thức SQL tham chiếu tới địa chỉ của một ô trong bảng tính, đây là điều kiện làm cho báo cáo động. Hàm RANGE được chuyển hóa thành công thức ghép chuỗi trong Excel. Khi thay giá trị ở ô được tham chiếu thì vùng công thức BS_SQL tự động cập nhật. Hàm RANGE cách dùng giống hàm CELL, nhưng hàm RANGE thì bị chuyển hóa và làm cho công thức BS_SQL tự động cập nhật.

Cấu trúc: CELL(AddressCell [,ValueType [,DateFmt]])
Nhận giá trị tại địa chỉ "AddressCell" trong bảng tính

AddressCell: địa chỉ ô
ValueType: nhận một trong các giá trị
- T - Nếu xác định giá trị là kiểu văn bản (Text)
- D - Nếu xác định giá trị là kiểu ngày (Date)
- N - Nếu xác định giá trị là kiểu số (Number)
- Nếu bỏ qua, A-Tools tự động xác định kiểu giá trị (theo Format Cell)

DateFmt: Là định dạng ngày hệ thống sử dụng làm chuẩn, ngầm định nhận giá trị MM/dd/yy . Dù bạn đang sử dụng kiểu ngày tháng dd/MM/yy cũng không nên thay đổi, chỉ thay đổi nếu thực sự cần thiết. Tham số này chỉ sử dụng khi ValueType là D (xác định giá trị trong AddressCell là kiểu Date)

Ví dụ 37: Hàm RANGE – Điều kiện tham chiếu tới ô chứa giá trị văn bản trong bảng tính

Giả thiết B10 chứa giá trị HH001. Cần lấy tất cả dữ liệu từ bảng KHO mà mã hàng tại ô B10.

```
=bs_sql("SELECT * FROM KHO WHERE MA_VLSPHH=RANGE(B10)")
```

Hàm DATA: Lấy giá trị tại một vị trí của mảng kết quả

Hàm **DATA** dùng sau từ khóa SELECT. Nhận giá trị tại vị trí dòng, cột của mảng hai chiều (2D), mảng này là mảng kết quả trả về của hàm BS_SQL. Số cột của nó tùy vào danh sách biểu thức liệt sau từ khóa SELECT.

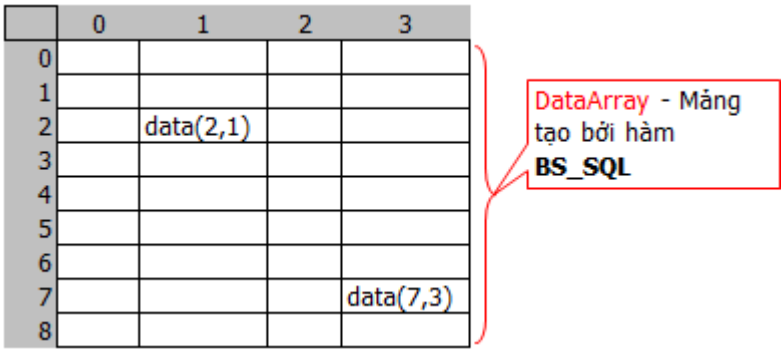
Cấu trúc hàm: **data**(Row,Col)

Trả về giá trị tại dòng (Row) và cột (Col) trong mảng giá trị.

Row, Col: là số nguyên dương (>=0) là giá trị dòng, cột hoặc các biến hệ thống dưới đây:

+ PrevRow: Trả về dòng liền trước
+ Row: Trả về dòng đang hoạt động
+ PrevCol: Trả về cột liền trước
+ Col: Trả về cột đang hoạt động

Cấu trúc mảng kết quả trả về bởi hàm BS_SQL. Thứ tự dòng và cột trong mảng bắt đầu từ vị trí 0.



Ví dụ 38: Lập sổ chi tiết hàng hóa gồm SO_CT, NGAY_CT, SLG, DON_GIA,THANH_TIEN, "Lũy kế". Cột "Lũy kế" được tính bởi công thức

Slg luỹ kế cột j = Slg liền trước của cột j + Giá trị mới tại cột Slg

Theo như yêu cầu trên thì bảng kết quả sẽ có 6 cột: SO_CT, NGAY_CT, SLG, DON_GIA,THANH_TIEN, "Lũy kế". Mảng kết quả sẽ lưu giá trị tại các cột có vị trí từ 0->5. Vị trí cột SLG là 2, cột "Lũy kế" là 5.

Slg luỹ kế = Slg liền trước của cột + Giá trị mới tại cột Slg (thứ tự cột là 2)

Hàm DATA thể hiện việc tính lũy kế như sau:

data(PrevRow,Col)+data(Row,2) AS [Luỹ kế]

Cách làm:

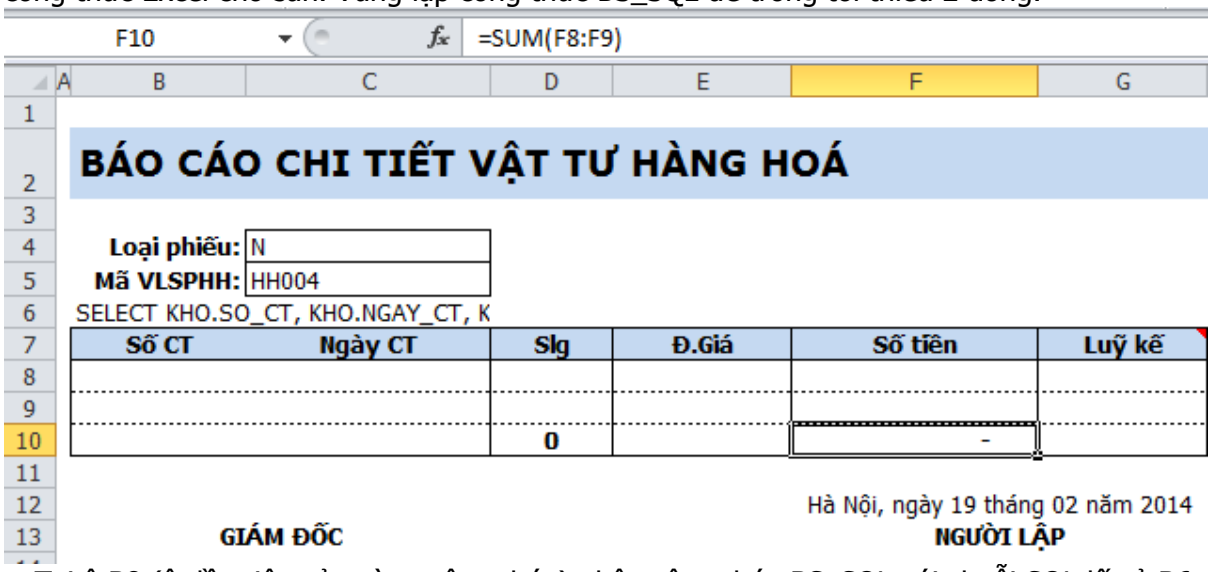
+ Tại ô B6 đặt công thức

```
="SELECT KHO.SO_CT, KHO.NGAY_CT, KHO.SLG, KHO.DON_GIA, KHO.THANH_TIEN,
        data(PrevRow,5)+data(Row,2) AS [Luỹ kế]
FROM KHO
WHERE KHO.LOAI_PHIEU = CELL(C4) AND KHO.MA_VLSPHH = RANGE(C5)"
```

➔ Dùng RANGE(C5) (không dùng CELL(C5) vì muốn khi thay đổi mã hàng tại ô C5 thì bảng dữ liệu tự động cập nhật lại.

➔ Khi chuỗi khai báo SQL dài chúng ta không nên đưa trực tiếp vào hàm BS_SQL mà nên đưa vào một ô sau đó dùng địa chỉ ô này đưa vào hàm BS_SQL.

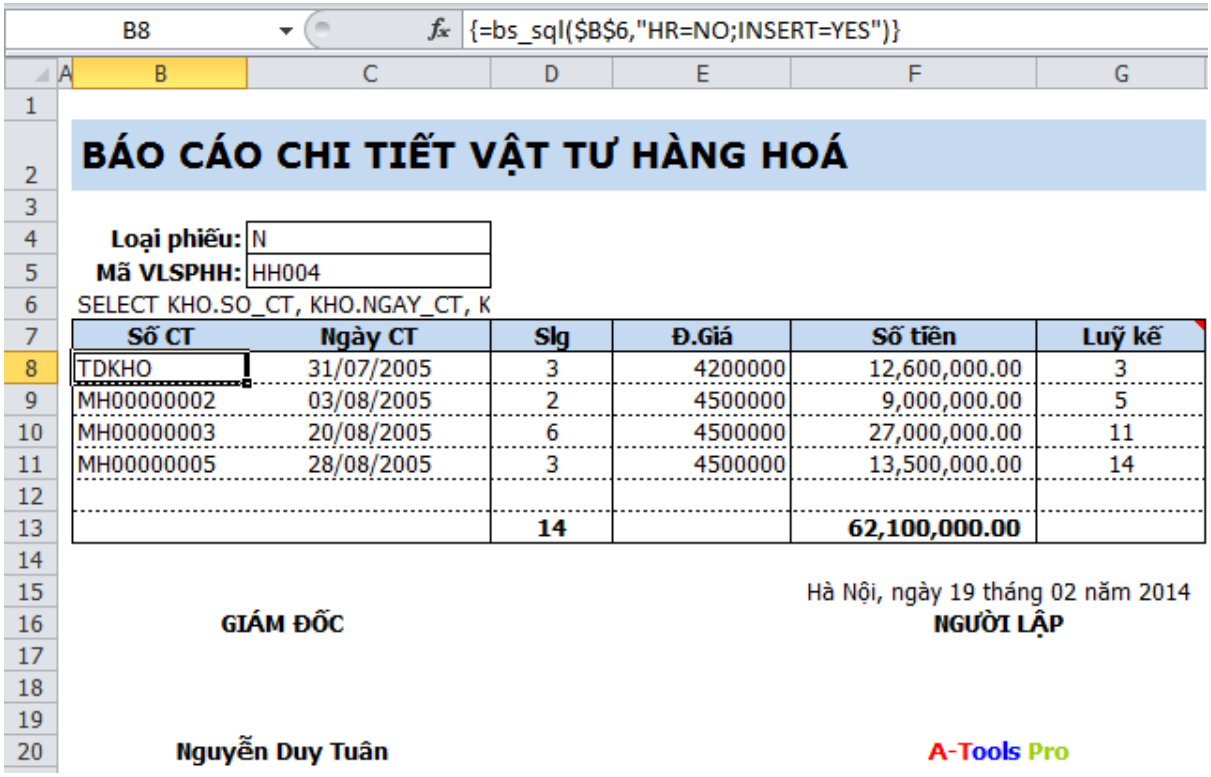
+ Tại ô B7 nhập các dòng tiêu đề có dấu để tiện cho in ấn. Thiết kế khung báo cáo. Lập các công thức Excel chờ sẵn. Vùng lập công thức BS_SQL để tróng tối thiểu 2 dòng.



+ Tại ô B8 (ô đầu tiên của vùng công thức) nhập công thức BS_SQL với chuỗi SQL lấy ở B6, sử dụng tham số chèn dòng, không hiện dòng tiêu đề của bảng kết quả vì dòng 7 đã tự nhập rồi.

```
=bs_sql($B$6,"HR=NO;INSERT=YES")
```

+ Kết quả ta được như hình dưới đây



Hàm RANGE lấy giá trị tại địa chỉ ô chỉ định. Hàm RANGE giúp ta xây dựng công thức SQL tham chiếu tới địa chỉ của một ô trong bảng tính, đây là điều kiện làm cho báo cáo động. Hàm RANGE được chuyển hóa thành công thức ghép chuỗi trong Excel. Khi thay giá trị ở ô được tham chiếu C5 thì vùng công thức BS_SQL tự động cập nhật, các dòng dữ liệu tự động co giãn.

Hàm MIN, MAX kết hợp hàm DATA

Cấu trúc: MIN(value1, value2)

Nếu value1<value2 thì hàm MIN trả về value1

Cấu trúc: MAX(value1, value2)

Nếu value1<value2 thì hàm MAX trả về value2

Ví dụ 39: Hàm MIN, MAX kết hợp hàm DATA

```
=bs_sql("SELECT MA_VLSPHH,SLG, Max(DATA( Row, 1),  8) AS [MAX],
                                Min(DATA( Row, 1), 4) AS [MIN]
FROM KHO
WHERE LOAI_PHIEU = 'N' ")
```

A9 {=bs_sql("SELECT MA_VLSPHH,SLG, Max(DATA(Row, 1), 8) AS [MAX], Min(DATA(Row, 1), 4) AS [MIN] FROM KHO WHERE LOAI_PHIEU = 'N' ")}

	A	B	C	D	E	F	G	H	I
9	MA_VLSPHH	SLG	MAX	MIN					
25	HH001	2	8	2					
26	HH002	2	8	2					
27	HH002	6	8	4					
28	HH001	4	8	4					
29	HH003	5	8	4					
30	HH004	3	8	3					
31	HH005	2	8	2					
32	HH002	4	8	4					
33	HH006	3	8	3					
34	BM	10	10	4					
35	BE1	5	8	4					

- Ở cột MAX, công thức Max(DATA(Row, 1), 8): Giá trị SLG so sánh với 8. Ví dụ dòng 25, SLG=2 vì vậy cột MAX trả về 8 (MAX(2,8)=8)
- Ở cột MIN, công thức Min(DATA(Row, 1), 4): Giá trị SLG so sánh với 4. Ví dụ dòng 27, SLG=6 vì vậy cột MIN trả về 4 (MIN(6,4)=4).

Hàm GETONCE: Lấy giá trị của biểu thức một lần ở dòng đầu trong mảng

Hàm **GetOnce** dùng sau từ khóa SELECT. Nhận giá trị của biểu thức Expr một lần cho dòng đầu tiên của mảng giá trị.

Cấu trúc hàm: **GetOnce**(Expr) Nhận giá trị trong biểu thức Expr. Nó chỉ nhận ở dòng đầu tiên trong mảng kết quả trả về bởi BS_SQL.

Ví dụ 40: Lập sổ chi tiết hàng hóa gồm SO_CT, NGAY_CT, SLG, DON_GIA,THANH_TIEN, "Lũy kế". Cột "Lũy kế" được tính bởi công thức

Slg luỹ kế cột j = Tồn đầu + Slg liền trước của cột j + Giá trị mới tại cột Slg

Tồn đầu là 20 nằm ở G6.

Theo như yêu cầu trên thì bảng kết quả sẽ có 6 cột: SO_CT, NGAY_CT, SLG, DON_GIA,THANH_TIEN, "Lũy kế". Mảng kết quả sẽ lưu giá trị tại các cột có vị trí từ 0->5. Vị trí cột SLG là 2, cột "Lũy kế" là 5.

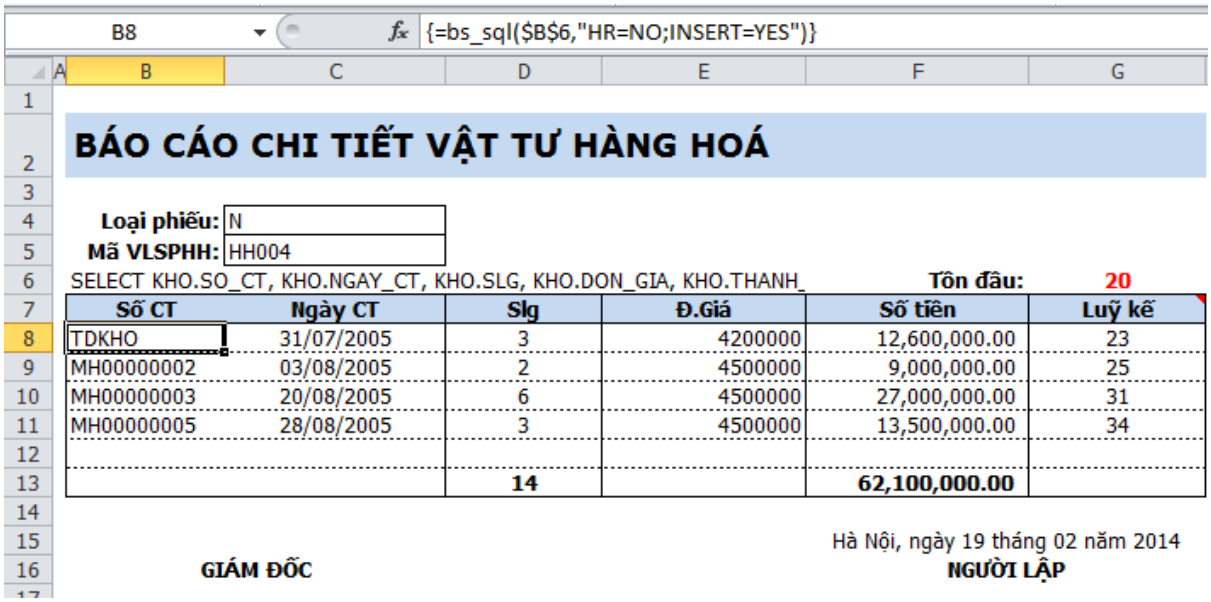
Cách làm:

+ Tại ô B6 đặt công thức

```
="SELECT KHO.SO_CT, KHO.NGAY_CT, KHO.SLG, KHO.DON_GIA, KHO.THANH_TIEN,
        GetOnce(Cell(G6))+data(PrevRow,5)+data(Row,2) AS [Luỹ kế]
FROM KHO
WHERE KHO.LOAI_PHIEU = CELL(C4) AND KHO.MA_VLSPHH = RANGE(C5)"
```

+ Tại ô B8 (ô đầu tiên của vùng công thức) nhập công thức BS_SQL với chuỗi SQL lấy ở B6, sử dụng tham số chèn dòng, không hiện dòng tiêu đề của bảng kết quả vì dòng 7 đã tự nhập rồi.

```
=bs_sql($B$6,"HR=NO;INSERT=YES")
```



Add-in A-Tools thực hiện việc tính toán trong mảng kết quả như sau:

+ Dòng 0 của mảng (dòng 8 của Excel) công thức là

GetOnce(Cell(G6))+data(PrevRow,5)+data(Row,2) AS [Luỹ kế]

20 + 3 = 23

+ Dòng 1 của mảng (dòng 9 của Excel) công thức là

data(PrevRow,5)+data(Row,2) AS [Luỹ kế]

23 + 2 = 25

Hàm FORMULA: Lấy giá trị của công thức Excel

Hàm **FORMULA** dùng để nhận kết quả của công thức Excel

Cấu trúc hàm: **FORMULA**(XlFml) - Nhận kết quả từ biểu thức XlFml

XlFml: Là một công thức Excel.

Ví dụ 41: Lập sổ chi tiết hàng hóa gồm SO_CT, NGAY_CT, SLG, DON_GIA,THANH_TIEN, "Lũy kế". Cột "Lũy kế" được tính bởi công thức

Slg luỹ kế cột j = Tồn đầu + Slg liền trước của cột j + Giá trị mới tại cột Slg

Tồn đầu là công thức Excel: 10+SUM(I3:I4)+G6

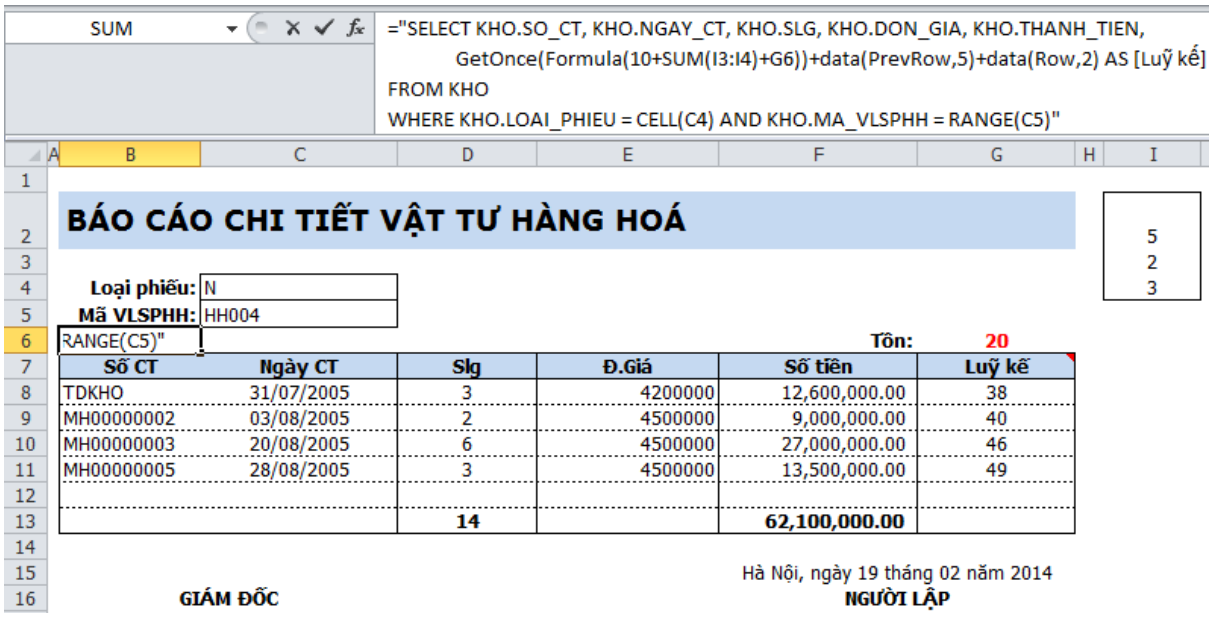
+ Tại ô B6 đặt công thức

```
="SELECT KHO.SO_CT, KHO.NGAY_CT, KHO.SLG, KHO.DON_GIA, KHO.THANH_TIEN,
        GetOnce(Formula(10+SUM(I3:I4)+G6))+data(PrevRow,5)+data(Row,2) AS [Luỹ kế]
FROM KHO
WHERE KHO.LOAI_PHIEU = CELL(C4) AND KHO.MA_VLSPHH = RANGE(C5)"
```

+ Tại ô B8 (ô đầu tiên của vùng công thức) nhập công thức BS_SQL với chuỗi SQL lấy ở B6, sử dụng tham số chèn dòng, không hiện dòng tiêu đề của bảng kết quả vì dòng 7 đã tự nhập rồi.

```
=bs_sql($B$6,"HR=NO;INSERT=YES")
```

+ Kết quả



Hàm FIELDIF, IIF

Cấu trúc:

FieldIf(search_condition,ValueTrue,[ValueFalse],[Func])

search_condition: Là biểu thức logic, cách dùng như sau mệnh đề WHERE

ValueTrue:Là giá trị nếu kết quả của search_condition là đúng (True), ValueTrue có thể là tên cột trong table_source.

ValueFalse:Là giá trị nếu kết quả của search_condition là sai (False), ValueFalse có thể là tên cột trong table_source.

Func: Tên hàm dùng để tính (SUM, COUNT, MIN, MAX, AVG)

Ví dụ 42: Lập sổ chi tiết nhập-xuất tồn hàng hóa. Mã hàng nằm ở ô C10. Giá trị tồn đầu ở F13. "Tồn" = Tồn trước + Nhập – Xuất

+ Tại ô B6 đặt công thức

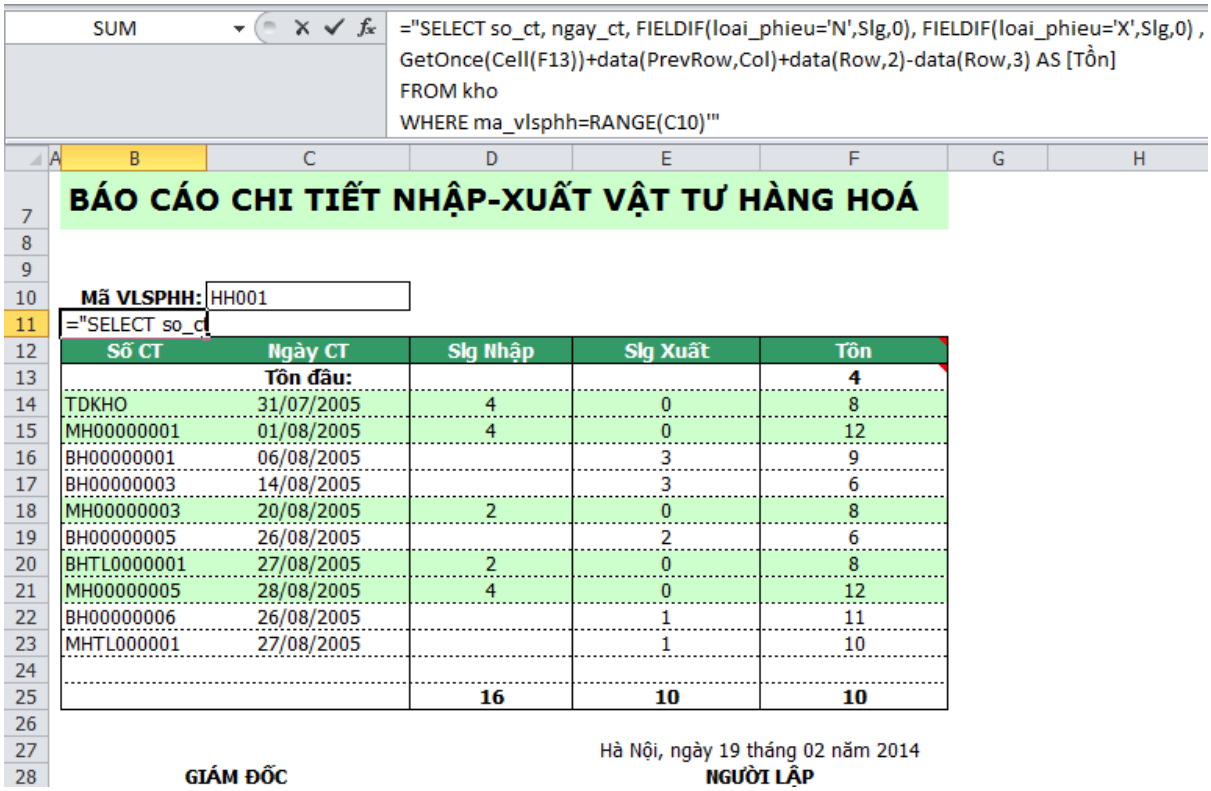
```
="SELECT so_ct, ngay_ct, FIELDIF(loai_phieu='N',Slg,0), FIELDIF(loai_phieu='X',Slg,0) ,
        GetOnce(Cell(F13))+data(PrevRow,Col)+data(Row,2)-data(Row,3) AS [Tồn]
```

```
FROM kho
WHERE ma_vlsphh=RANGE(C10)'"
```

+ Tại ô B14 (ô đầu tiên của vùng công thức) nhập công thức BS_SQL với chuỗi SQL lấy ở B11, sử dụng tham số chèn dòng, không hiện dòng tiêu đề của bảng kết quả vì dòng 12 đã tự nhập rồi.

```
=bs_sql(B6,"HR=NO;INSERT=YES")
```

+ Kết quả



Ví dụ 43: Lập sổ tổng hợp nhập-xuất tồn hàng hóa.

+ Tại ô B6 đặt công thức

```
SELECT DMVLSPHH.MA_VLSPHH,DMVLSPHH.Ten, DMVLSPHH.Dvi,
        FIELDIF(loai_phieu='N',SLG,0,SUM) as [N],
        FIELDIF(loai_phieu='X',SLG,0,SUM) as [X],
        N-X as [Tồn]
FROM DMVLSPHH LEFT JOIN
 KHO ON DMVLSPHH.MA_VLSPHH = KHO.MA_VLSPHH
GROUP BY DMVLSPHH.MA_VLSPHH, DMVLSPHH.Ten, DMVLSPHH.Dvi
```

+ Kết quả

B8 {=bs_sql(B6,"HR=NO;INSERT=YES")}

BÁO CÁO TH NHẬP-XUẤT-TỒN VẬT TƯ HÀNG HOÁ

SELECT DMVLSPHH.MA_VLSPHH, DMVLSPHH.Ten, DMVLSPHH.Dvi, FIELDIF(loai_phieu='N',SLG,0,SL

Mã HH	Tên HH	ĐVT	Slg Nhập	Slg Xuất	Tồn
BE	Bê	Đĩa		2	-2
BE1	Bê	Kg	5	0	5
BM	Bò mông	Kg	10		10
BO	Bò xào nướng	Đĩa	0	4	-4
BTC	Bê tái chanh	Đĩa		2	-2
HH001	Máy tính ĐNA L1	Chiếc	16	10	6
HH002	Máy tính ĐNA L2	Chiếc	33	19	14
HH003	Máy in HP 1200	Chiếc	19	7	12
HH004	Máy in HP 1300	Chiếc	14	9	5
HH005	Dầu A	Chai	2	20	-18
HH006	Dầu B	Chai	3	16	-13
LB	Lẩu bò	Nồi	0	2	-2

Hà Nội, ngày 19 tháng 02 năm 2014

GIÁM ĐỐC NGƯỜI LẬP

Ta có thể dùng hàm IIF thay cho FIELDIF

```
SELECT DMVLSPHH.MA_VLSPHH, DMVLSPHH.Ten, DMVLSPHH.Dvi,
        SUM(IIF(loai_phieu='N',SLG,0) as [N],
        SUM(IIF(loai_phieu='X',SLG,0) as [X],
        N-X as [Tồn]
FROM DMVLSPHH LEFT JOIN
  KHO ON DMVLSPHH.MA_VLSPHH = KHO.MA_VLSPHH
GROUP BY DMVLSPHH.MA_VLSPHH, DMVLSPHH.Ten, DMVLSPHH.Dvi
```

Hàm VTC: Tạo báo các dạng Cross Tab

Cấu trúc: VTC(ColumnName,Func,ColumnCalc,[FormatDate])

ColumnName: Tên cột mà các giá trị của nó làm tiêu thức, điều kiện để tính

Func: Tên hàm dùng để tính (SUM, COUNT, MIN, MAX, AVG)

ColumnCalc:Tên cột được tính giá trị theo hàm khai báo ở Func, cột này phải là kiểu số.

FormatDate: Là định dạng ngày. Có thể bỏ qua tham số này, nếu kiểu của ColumnName không phải là kiểu Date, hoặc là kiểu Date nhưng dữ liệu đang ở dạng mm/dd/yy. Nếu ColumnName là kiểu Date mà dữ liệu nhập không phải ở dạng mm/dd/yy thì FormatDate phải là mm/dd/yy.

Hàm VTC sẽ liệt kê giá trị duy nhất trong cột ColumnName thành các cột. Trong mỗi cột tạo ra được thống kê giá trị ở cột ColumnCalc theo một trong các hàm SUM, COUNT, MIN, MAX, AVG.

Ví dụ 44: Lập sổ tổng hợp số lượng hàng xuất cho các khách hàng. Gồm các cột MA_VLSPHH, các cột còn lại là tổng số lượng hàng xuất cho từng mã khách hàng.

+ Tại ô B14 đặt công thức

```
SELECT MA_VLSPHH, VTC(MA_KH, SUM, SLG)
FROM KHO WHERE LOAI_PHIEU='X'
GROUP BY MA_VLSPHH
```

B15 đặt công thức sử dụng SQL ở B14

=bs_sql($B$14,"INSERT=YES")

+ Kết quả như dưới đây

B15 {=bs_sql($B$14,"INSERT=YES")}

BÁO CÁO TỔNG HỢP HÀNG XUẤT KHÁCH HÀNG

SELECT MA_VLSPHH, VTC(MA_KH, SUM,SLG)
FROM KHO WHERE LOAI_PHIEU='X'
GROUP BY MA_VLSPHH

MA_VLSPHH	KH001	KH002	KH003	KH004
BE	2			
BO	4			
BTC	2			
HH001	2			1
HH002	2		4	2
HH003			1	2
HH004	1			4
HH005	4	4	8	4
HH006	2	2	4	8
LB	2			

- ➔ VTC(MA_KH, SUM, SLG) : sẽ liệt kê các mã khách hàng duy nhất thành các cột, trong mỗi cột được tính tổng số lượng mỗi mặt hàng.
- ➔ GROUP BY MA_VLSPHH : A-Tools sẽ thống kê sô liệu và nhóm theo mỗi mặt hàng.

Ví dụ 45: Lập sổ tổng hợp số lượng hàng xuất cho các khách hàng. Gồm các cột MA_VLSPHHSTT, Mã hàng, Tên hàng, ĐVT, Tổng xuất, Tên các khách hàng với số lượng xuất mỗi mặt hàng tương ứng.

+ Tại ô A1 đặt công thức

```
SELECT RECNO() AS STT,K.MA_VLSPHH AS [Mã], HH.TEN AS [Tên], HH.Dvi AS [ĐVT],
                                SUM(K.SLG) AS [Tổng xuất], VTC(KH.TEN,SUM,K.SLG)
 FROM KHO K, DMKH KH, DMVLSPHH HH
 WHERE K.MA_KH=KH.MA_KH AND K.MA_VLSPHH=HH.MA_VLSPHH AND K.LOAI_PHIEU='X'
 GROUP BY K.MA_VLSPHH, HH.TEN, HH.Dvi
```

A2 đặt công thức sử dụng SQL ở A1

=bs_sql(A1,"INSERT=YES")

+ Kết quả như dưới đây

A2 fx {=bs_sql(A1, "INSERT=YES")}

```
SELECT RECNO() AS STT,K.MA_VLSPHH AS [Mã], HH.TEN AS [Tên], HH.Dvi AS [ĐVT], SUM(K.SLG) AS [Tổng xuất],
VTC(KH.TEN,SUM,K.SLG)
 FROM KHO K, DMKH KH, DMVLSPHH HH
 WHERE K.MA_KH=KH.MA_KH AND K.MA_VLSPHH=HH.MA_VLSPHH AND K.LOAI_PHIEU='X'
 GROUP BY K.MA_VLSPHH, HH.TEN, HH.Dvi
```

STT	Mã	Tên	ĐVT	Tổng xuất	Cty IPQ	Cty TNHH Hoàng Long	Cửa hàng Thành Đông	Cửa hàng máy tính Đông Đô
1	BE	Bê	Đĩa	**2**		2		
2	BO	Bò xào nướng	Đĩa	**4**		4		
3	BTC	Bê tái chanh	Đĩa	**2**		2		
4	HH001	Máy tính ĐNA L1	Chiếc	**9**	1	5		3
5	HH002	Máy tính ĐNA L2	Chiếc	**17**	5	4	2	6
6	HH003	Máy in HP 1200	Chiếc	**7**	4	1	1	1
7	HH004	Máy in HP 1300	Chiếc	**9**	4	3	2	
8	HH005	Dầu A	Chai	**20**	4	4	4	8
9	HH006	Dầu B	Chai	**16**	8	2	2	4
10	LB	Lẩu bò	Nồi	**2**		2		

Ví dụ 46: Lập bảng tổng hợp số lượng các hàng hoá xuất kho theo từng ngày. Dữ liệu được lấy từ sổ KHO.

+ Tại ô A1 đặt công thức

```
SELECT MA_VLSPHH,SUM(SLG) AS [TỔNG XUẤT], VTC(NGAY_CT,SUM,SLG,MM/DD/YY)
 FROM KHO
 WHERE LOAI_PHIEU='X'
                                GROUP BY MA_VLSPHH
```

A2 fx {=bs_sql(A1, "INSERT=YES")}

```
SELECT MA_VLSPHH,SUM(SLG) AS [TỔNG XUẤT], VTC(NGAY_CT,SUM,SLG,MM/DD/YY)
 FROM KHO
 WHERE LOAI_PHIEU='X'
 GROUP BY MA_VLSPHH
```

MA_VLSPHH	TỔNG XUẤT	06/08/2005	08/08/2005	14/08/2005	26/08/2005	27/08/2005	09/08/2006	24/10/2006
BE	**2**							2
BO	**4**							4
BTC	**2**							2
HH001	**10**	3		3	3	1		
HH002	**19**	2	2	2	7	6		
HH003	**7**	1	1		4	1		
HH004	**9**	2	2		5			
HH005	**20**						20	
HH006	**16**						16	
LB	**2**							2

Hàm SQL: Cập nhật nguồn dữ liệu có hàm BS_SQL

Cấu trúc**:** SQL(table_source[, **u**pdate])

table_source: là tên của bảng(table) hay tên của một bảng được truy vấn bởi hàm BS_SQL.

update: Nếu là TRUE bảng table_source sẽ luôn được cập nhật lại khi công thức BS_SQL gọi nó thực hiện. Là FALSE (ngầm định) hoặc bỏ trống thì hàm BS_SQL chỉ cập nhật bảng table_source nếu nó đang ở trạng thái #N/A (chưa cập nhật).

- ➔ Bạn nên dùng tham số với giá trị theo ngầm định.
- ➔ Hàm SQL cần sử dụng khi công thức table2 dựa vào nguồn dữ liệu của table1, table1 được tạo bởi hàm BS_SQL.

Ví dụ 47: Cập nhật dữ liệu nguồn trước khi truy vấn dữ liệu.

+ Giả xử ta lập công thức lấy toàn bộ dữ liệu trong sổ kho của mã hàng HH001, bảng kết quả được đặt tên là DLHH1.

Công thức là:

```
=bs_sql("SELECT * FROM KHO WHERE MA_VLSPHH='HH001' ", "NAME=DLHH1")
```

DLHH1 | {=bs_sql("SELECT * FROM KHO WHERE MA_VLSPHH='HH001' ", "NAME=DLHH1")}

	A	B	C	D	G	H	I	J	K	L
9	SO_CT	NGAY_CT	MA_KH	MA_NB	MA_VLSPH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN	
10	TDKHO	31/07/2005		NB001	HH001	4	4,800,000	N	19,200,000	
11	NK00000001	01/08/2005		NB001	HH001	4	5,000,000	N	20,000,000	
12	XK00000001	06/08/2005	KH001		HH001	3	5,500,000	X	14,700,000	
13	XK00000003	14/08/2005	KH003		HH001	3	5,500,000	X	14,700,000	
14	NK00000003	20/08/2005		NB004	HH001	2	5,000,000	N	10,000,000	
15	XK00000005	26/08/2005	KH001		HH001	2	5,800,000	X	9,900,000	
16	XK00000006	26/08/2005	KH004		HH001	1	5,500,000	X	5,135,714	
17	NK00000005	27/08/2005	KH001		HH001	2	5,500,000	N	9,900,000	
18	NK00000006	28/08/2005		NB002	HH001	4	5,000,000	N	20,000,000	
19	XK00000008	27/08/2005		NB001	HH001	1	5,000,000	X	5,000,000	

NAME=DLHH1: A-Tools sẽ tạo vùng dữ liệu A9:K19 với NAME là "DLHH1".

+ Cần lấy tất cả dữ liệu từ bảng DLHH1 tạo ra ở trên với phiếu nhập
Công thức là:

```
=bs_sql("SELECT * FROM SQL(DLHH1) WHERE LOAI_PHIEU='N' ")
```

A21 | {=bs_sql("SELECT * FROM SQL(DLHH1) WHERE LOAI_PHIEU='N' ")}

	A	B	C	D	G	H	I	J	K
21	SO_CT	NGAY_CT	MA_KH	MA_NB	MA_VLSPH	SLG	DON_GIA	LOAI_PHIEU	THANH_TIEN
22	TDKHO	31/07/2005		NB001	HH001	4	4,800,000	N	19,200,000
23	NK00000001	01/08/2005		NB001	HH001	4	5,000,000	N	20,000,000
24	NK00000003	20/08/2005		NB004	HH001	2	5,000,000	N	10,000,000
25	NK00000005	27/08/2005	KH001		HH001	2	5,500,000	N	9,900,000
26	NK00000006	28/08/2005		NB002	HH001	4	5,000,000	N	20,000,000

Phương pháp tạo báo cáo động trong Excel với hàm BS_SQL

Phương pháp làm báo cáo của Excel là phải làm NHIỀU công thức gộp lại rồi copy.

Phương thức làm báo của Add-in A-Tools là chỉ cần làm MỘT công thức có hàm BS_SQL tạo ra một báo cáo. Vì một vài lý do khác nữa mà báo cáo tạo bởi hàm BS_SQL trong Add-in A-Tools chạy nhanh và linh hoạt hơn cách làm Excel thông thường.

Để học và nắm được cách làm ta giả thiết cần lập sổ chi tiết vật tư hàng hóa.

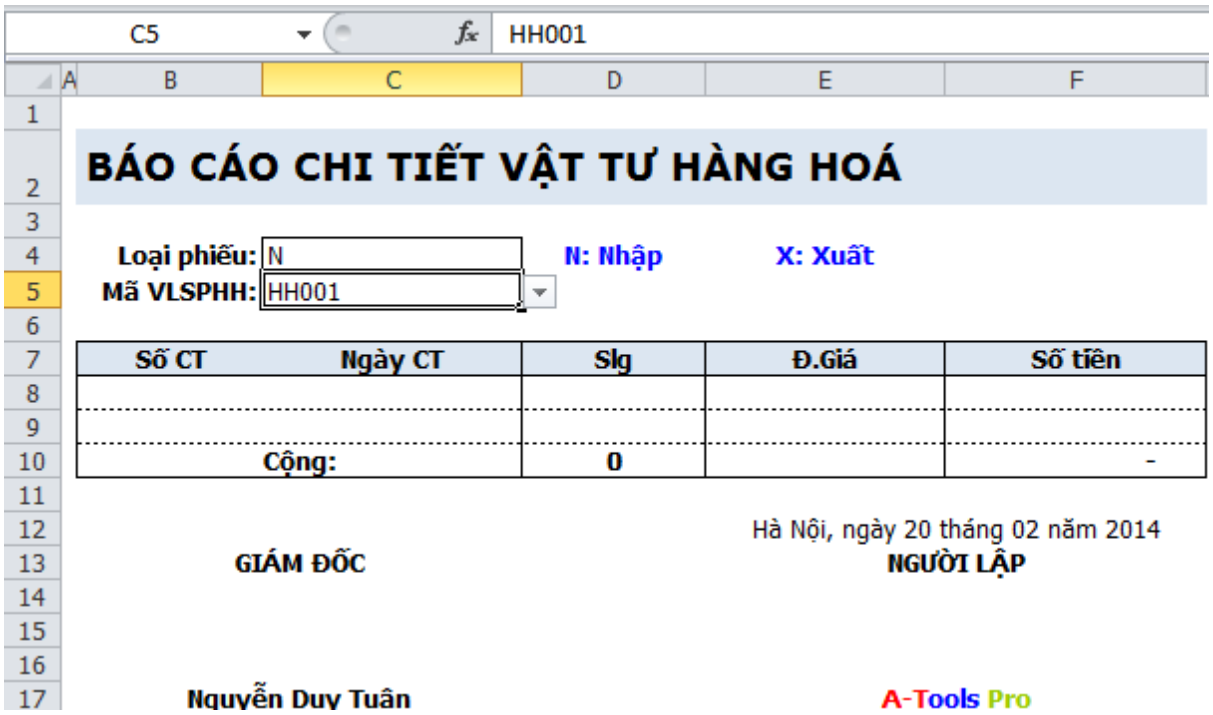
Ví dụ 48: Lập sổ chi tiết vật tư hàng hóa. Sổ này sẽ lọc ra các dòng dữ liệu liên quan đến loại phiếu được chọn (LOAI_PHIEU) và của mã hàng được chọn (MA_VLSPHH).

Mở tập tin có dữ liệu ví dụ "Examble.xls" có bảng dữ liệu nguồn KHO.

Việc thiết kế được thực hiện theo các bước sau:

Bước 1: Thiết kế cấu trúc của báo cáo

Tạo sheet "CTHH" trong tập tin "Examble.xls" để thiết kế cấu trúc báo cáo theo hình dưới đây.

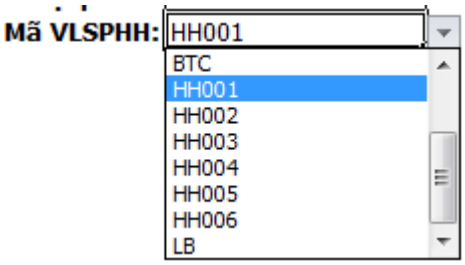


Tạo mục chọn mã dạng Validation - List

Trong báo cáo có 2 mục chọn:

+ "Loại phiếu" tại ô C4 cần chọn loại phiếu "N", "X" từ danh sách:

+ "Mã VLSPHH" tại ô C5 cần chọn từ danh sách:



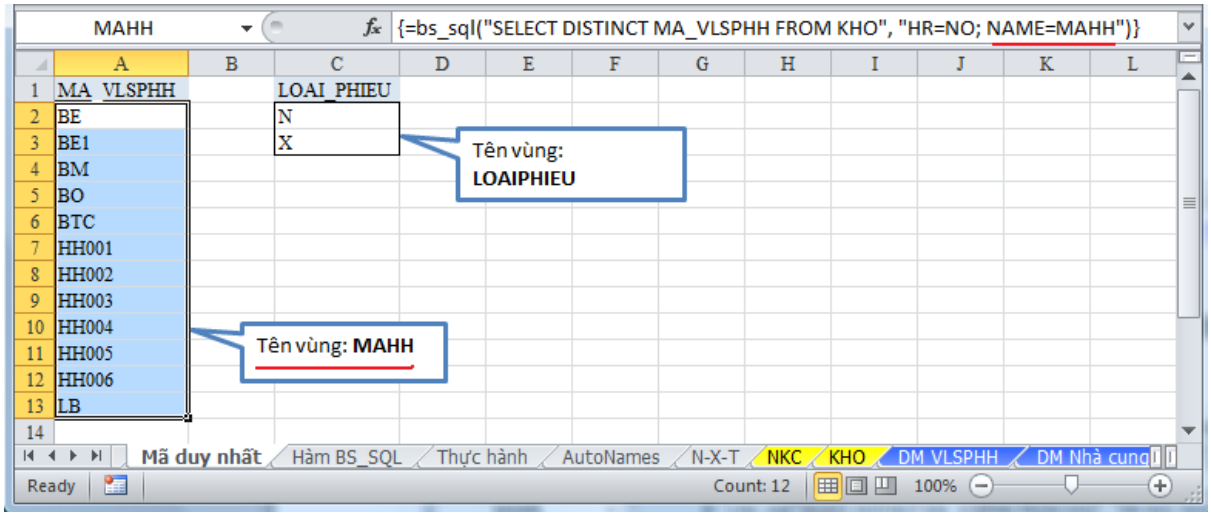
Mục chọn như trên chứa danh sách các mã hàng hóa, vật tư phát sinh trong cột MA_VLSPHH của bảng KHO.

Để nạp vào danh sách trên, đầu tiên cần tạo sheet mới (nếu chưa có) và đặt tên "Mã duy nhất" trong file "Examble.xls". Trong sheet này lập công thức BS_SQL để lấy ra danh sách mã hàng hóa, vật tư duy nhất, vùng kết quả được đặt tên là "MAHH"

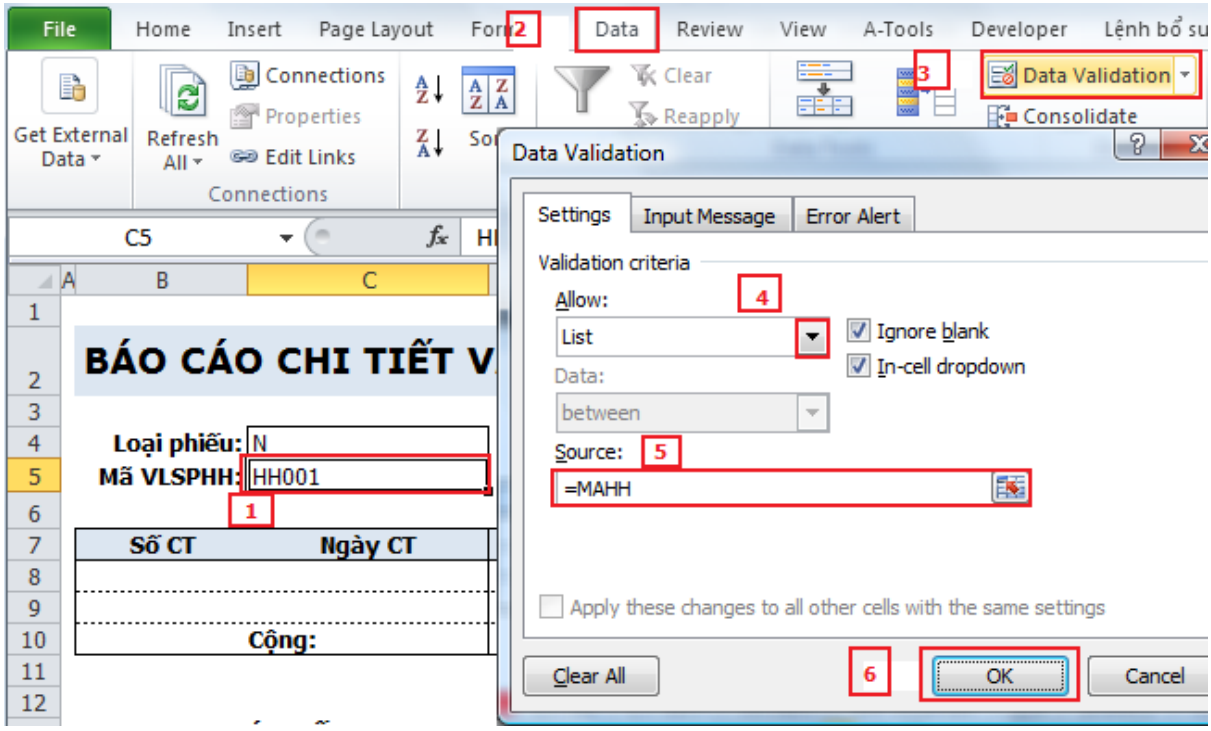
Công thức:

```
=bs_sql("SELECT DISTINCT MA_VLSPHH FROM KHO", "HR=NO; NAME=MAHH")
```

➔ Từ khóa DISTINCT để lấy danh sách duy nhất
➔ NAME=MAHH để A-Tools tạo tên (Define Name) vùng dữ liệu kết quả là MAHH.



Quay về sheet "CTHH", chọn ô C5 để tạo Validation – List. Hãy thực hiện đúng 6 bước dưới đây:



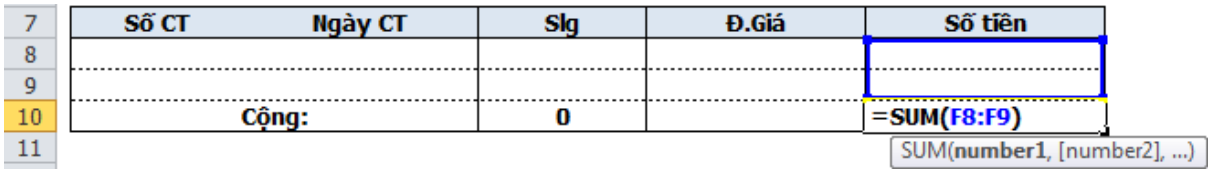
Chọn ô C5->Chọn menu "Data"->Chọn "Data Validation"->Allow chọn "List"->Source chọn "=MAHH"->Hoàn thành nhất [Ok].

Yêu cầu với khu vực chứa dữ liệu của báo cáo

+ Dòng tiêu đề nên nhập tay để thể hiện nội dung cụ thể.

+ Từ dòng thứ 2 của khu vực này (theo ví dụ này trong bảng tính Excel từ dòng 8) sẽ đặt công thức BS_SQL. Khu vực này phải để thừa ra ít nhất 2 dòng trống, nếu có định dạng dòng dữ liệu thì nên để 3 dòng trống.

+ Dòng cuối cùng của vùng dữ liệu, sau 2 hoặc 3 dòng trống đặt công thức Excel để tính tổng các cột dữ liệu. Theo ví dụ trên là dòng 10.

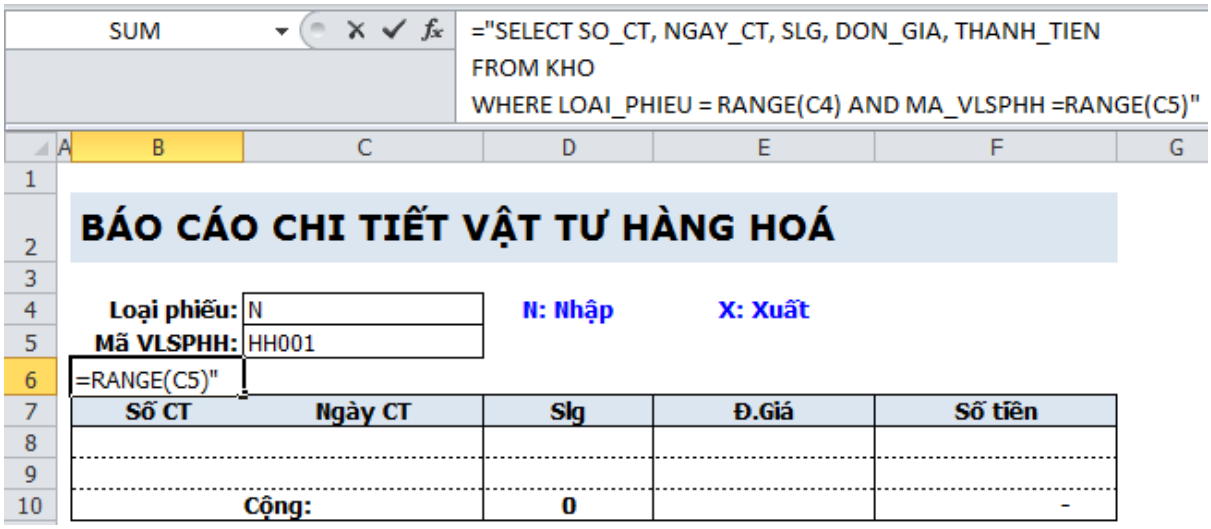


Bước 2: Lập công thức với hàm BS_SQL để lấy dữ liệu vào báo cáo

Cấu trúc của hàm BS_SQL(SQL [,OPTIONS]). Nếu chuỗi khai báo SQL mà ngắn thì ta có thể đặt trực tiếp trong hàm. Nếu nó dài thì nên để nó nằm ở một ô khác ví dụ B6. Tại công thức BS_SQL thì đặt công thức và sử dụng B6 làm tham số. =BS_SQL(B6). Làm như vậy Excel sẽ không báo lỗi khi chuỗi tham số SQL quá dài (>128 ký tự).

+ Đặt công thức có chuỗi khai báo SQL tại ô B6

```
="SELECT SO_CT, NGAY_CT, SLG, DON_GIA, THANH_TIEN
FROM KHO
WHERE LOAI_PHIEU = RANGE(C4) AND MA_VLSPHH =RANGE(C5)"
```



➔ Dùng hàm RANGE(C4), RANGE(C5) để khi thay đổi giá trị tại ô C4, C5 thì công thức được cập nhật lại.

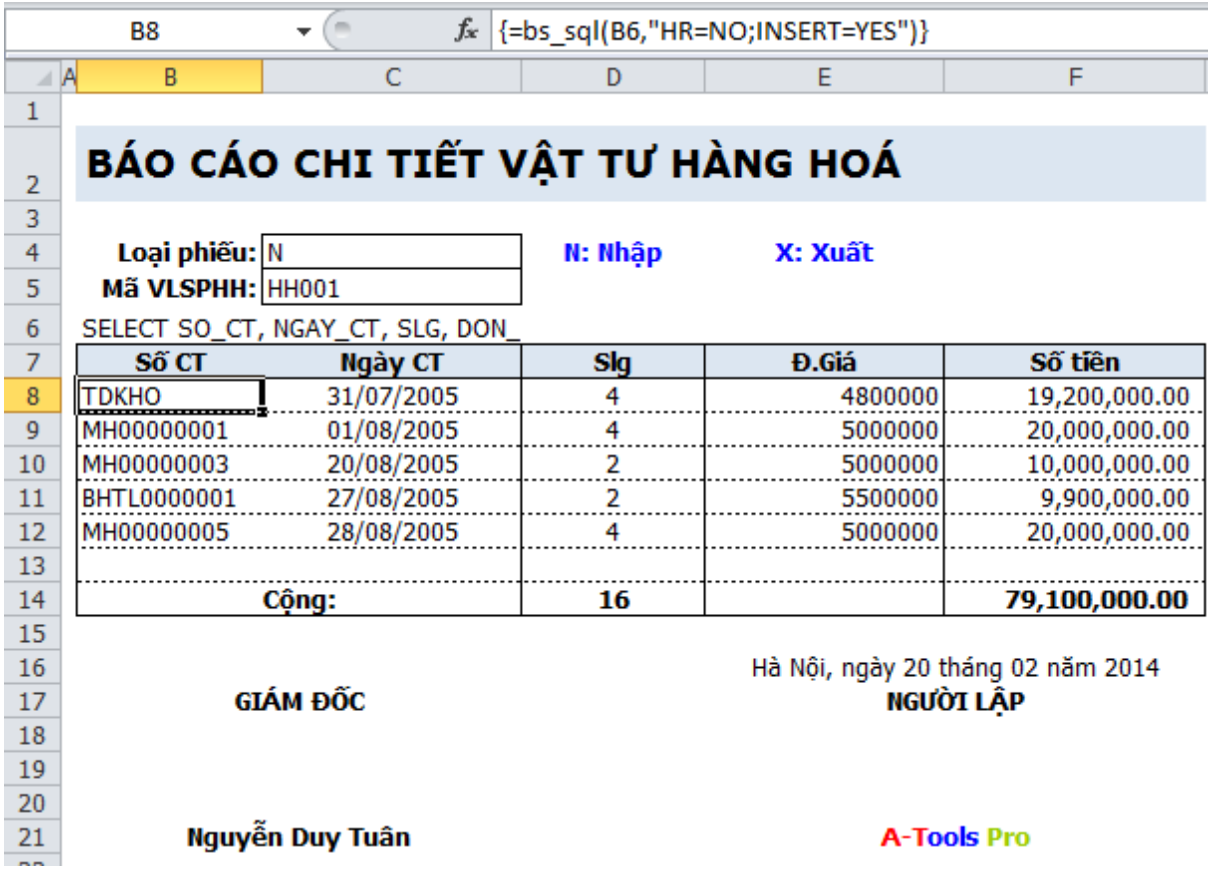
+ Đặt công thức BS_SQL tại ô B8

```
=bs_sql(B6,"HR=NO;INSERT=YES")
```

➔ Giá trị chuỗi SQL ở ô B6 được đưa vào tham số đầu tiên hàm BS_SQL
➔ HR=NO để bảng kết quả không có dòng tiêu đề. Vì dòng tiêu đề ta tự nhập tay.
➔ INSERT=YES để các dòng kết quả tự động co giãn khi thay đổi loại phiếu, mã hàng ở các ô C4, C5.

Sau khi nhập xong công thức nhấn ENTER, dữ liệu lập tức trả về và điền vào cả bảng báo cáo. Dòng cuối tự tính tổng các cột SQL, THANH_TIEN.

➔ Nếu muốn sửa công thức phải đảm bảo các việc: Chọn đúng một ô trong vùng công -> sửa công thức-> Nhấn CTRL+SHIFT+ENTER để chạy vì đây là công thức mảng. Nếu không muốn thực hiện nhấn phím ESC để hủy việc sửa.



Video hướng dẫn làm báo cáo nhanh với SQL Builder trong Add-in A-Tools

http://www.youtube.com/watch?v=NWZM6kJAPZE

Các hàm CSDL: Tìm kiếm và thống kê có nhiều điều kiện

Add-in A-Tools cung cấp bộ hàm tìm kiếm và thống kê có nhiều điều kiện, nguồn dữ liệu có thể lấy từ trong chính tập tin Excel hoặc từ các nguồn CSDL bên ngoài như hàm BS_SQL. Các hàm là: BS_DVLOOKUP, BS_DSUM, BS_DCOUNT, BS_DMIN, BS_DMAX, BS_DAVG, BS_DVAR.

Để học tốt các hàm này bạn hãy mở tập tin sau để học và làm theo

C:\A-Tools\HELP & DEMOS\Bai 3 Hàm CSDL (Database Functions).xls

Hàm BS_DVLOOKUP: Tìm kiếm nhiều điều kiện

Cấu trúc: BS_DLOOKUP(expr, table_source, search_condition[, options])

Hàm BS_DLOOKUP làm một hàm mảng. Hàm tìm kiếm theo nhiều điều kiện theo search_condition, dữ liệu được tìm có thể trong một hay nhiều bảng. *Giá trị trả về có thể 1 hoặc cả mảng giá trị.*

expr: Biểu thức trả về (thường là một cột/field trong table_source) nếu việc tìm kiếm thoả mãn điều kiện search_condition

table_source: Một hay nhiều bảng dữ liệu nguồn (có quan hệ) chứa dữ liệu cần tìm và trả về, nó có thể là một Name trỏ tới một vùng dữ liệu. Nếu dữ liệu cần

tìm hay trả về có từ nhiều bảng thì table_source là nguồn dữ liệu được kết hợp từ nhiều table, cách kết hợp chúng giống như trong cú cú pháp sau từ khoá FROM của ngôn ngữ T-SQL.

search_condition: Điều kiện để tìm kiếm, cách viết điều kiện giống như với cú pháp SQL sau từ khoá WHERE của ngôn ngữ T-SQL. Nếu table_source được kết hợp từ nhiều bảng/table thì search_condition phải chỉ ra biểu thức quan hệ.

options: Tuỳ chọn cho hàm, cách dùng tham số này giống như trong hàm BS_SQL, các nhóm hàm này (BS_D*) thì chỉ sử dụng các khai báo tham số sau:

+ DBKEY = Mã DBKEY
  Mã DB KEY được tạo để kết nối với một CSDL bên ngoài. Xem hướng dẫn tạo DBKEY.
+ HAVING = search_condition
  search_condition là điều kiện có sử dụng các hàm thống kê như SUM, COUNT, AVG, MIN, MAX
+ GROUPBY = group_by_expression
  group_by_expression là danh sách các cột (field) cần nhóm dữ liệu.
+ ORDERBY = order_expression
  order_expression là danh sách các cột (field) cần được sắp xếp. Cần dùng với hàm BS_SQL, BS_TABLE, BS_DVLOOKUP
+ SERVERSOURCE = YES/NO

Nếu SERVERSOURCE = YES thì dữ liệu được lấy từ máy chủ, NO hoặc không khai báo thì dữ liệu lấy tại máy đang chạy.

- ➔ Cách khai báo HAVING, GROUPBY, ORDERBY tương tự như trong cấu trúc câu lệnh SQL sau các từ khoá HAVING, GROUP BY, ORDER BY
  Các tham số trong ngoặc vuông ([ ]) có thể bỏ qua.
- ➔ Các hàm BS_DSUM, BS_DCOUNT, BS_DMIN, BS_DMAX, BS_DAVG cũng có cấu trúc và cách dùng tương tự hàm BS_DLOOKUP

Ví dụ 49: Tìm một mã hàng mà trong bảng KHO có số lượng nhập kho là 10.

Hướng giải quyết: chỉ cần tìm trên một sổ KHO vì tại đây có các thông tin để tìm và dữ liệu trả về.

Công thức:

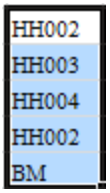
```
=BS_DLOOKUP("TOP 1 MA_VLSPHH","KHO","LOAI_PHIEU = 'N' AND SLG = 10")
```

- ➔ TOP 1: để lấy một giá trị đầu tiên

Ví dụ 50: Lấy ra các mã hàng có số lượng nhập > 5

```
=BS_DLOOKUP("MA_VLSPHH","KHO","LOAI_PHIEU = 'N' AND SLG > 5")
```

Kết quả trả về một mảng giá trị.



- ➔ Muốn sửa lại công thức nhấn CTRL+SHIFT+ENTER

Vẫn yêu cầu ví dụ trên bây giờ cần lấy ra tên hàng. Công thức là

```
=BS_DLOOKUP("DMVLSPHH.TEN",
"KHO INNER JOIN DMVLSPHH ON KHO.MA_VLSPHH = DMVLSPHH.MA_VLSPHH",
"LOAI_PHIEU = 'N' AND SLG >5")
```

➔ Nguồn dữ liệu table_source là hai bảng KHO và DMVLSPHH có quan hệ "KHO INNER JOIN DMVLSPHH ON KHO.MA_VLSPHH = DMVLSPHH.MA_VLSPHH"

Vẫn theo ví dụ trên nhưng dữ liệu tìm và lấy ở tập tin Access "Examble.mdb" và đã kết nối tới DBKEY là "MDB. Công thức như sau.

```
=BS_DLOOKUP("DMVLSPHH.TEN",
"KHO INNER JOIN DMVLSPHH ON KHO.MA_VLSPHH = DMVLSPHH.MA_VLSPHH",
"LOAI_PHIEU = 'N' AND SLG >5",
"DBKEY=MDB")
```

Hàm BS_DSUM: Tính tổng với nhiều điều kiện

Cấu trúc: BS_DSUM(expr, table_source , [search_condition] , [options])

Hàm BS_DSUM tính tổng theo nhiều điều kiện, dữ liệu làm điều kiện có thể từ một hoặc nhiều bảngtrong CSDL. Cách thức dùng giống như hàm BS_DVLOOKUP.

Ví dụ 51: Tính tổng số lượng nhập kho mà mã hàng là HH001

Công thức:

```
=BS_DSUM("SLG","KHO","LOAI_PHIEU='N' AND MA_VLSPHH='HH001'")
```

Hàm BS_DCOUNT: Đếm giá trị với nhiều điều kiện

Hàm này có cấu trúc và cách dùng tương tự hàm BS_DSUM

Hàm BS_DMIN: Tính giá trị nhỏ nhất với nhiều điều kiện

Hàm này có cấu trúc và cách dùng tương tự hàm BS_DSUM

Hàm BS_DMAX: Tính giá trị lớn nhất với nhiều điều kiện

Hàm này có cấu trúc và cách dùng tương tự hàm BS_DSUM

Hàm BS_DAVG: Tính trung bình với nhiều điều kiện

Hàm này có cấu trúc và cách dùng tương tự hàm BS_DSUM

Hàm BS_DVAR: Tính độ lệch chuẩn với nhiều điều kiện

Hàm này có cấu trúc và cách dùng tương tự hàm BS_DSUM

Hàm BS_DFUNC: Tính nhiều điều kiện theo tham số là tên hàm

Cấu trúc: BS_DFUNC(FuncName, expr, table_source, [search_condition], [options])

FuncName: Là một chuỗi chỉ ra tên hàm dùng để tính, nhận một trong các tên hàm SUM, COUNT, AVG, MIN, MAX, VAR

Ví dụ 52: Tính tổng số lượng nhập kho mà mã hàng là HH001

Công thức:

```
=BS_DSUM("SLG","KHO","LOAI_PHIEU='N' AND MA_VLSPHH='HH001'")
```

Hoặc:

```
=BS_DFUNC("SUM","SLG","KHO" , "LOAI_PHIEU='N' AND MA_VLSPHH='HH001' ")
```

Hàm BS_TABLE: Trích lọc dữ liệu - Cách làm khác của BS_SQL

Cấu trúc: BS_TABLE(select_list, table_source , [search_condition] , [options])

Hàm BS_TABLE là hàm mảng, ý nghĩa tương tự như hàm BS_SQL, các tham số của hàm BS_TABLE tương ứng với các thành phần của khai báo SQL như: SELECT, FROM, WHERE,...

Các tham số trong ngoặc vuông ([ ]) có thể bỏ qua.

select_list: là danh sách các cột được trả về trong bảng kết quả. select_list chính là khai báo sau từ khoá SELECT của câu lệnh SQL.

table_source: Bảng dữ liệu nguồn chứa các thông tin cần trích lọc, nó có thể là một Name trỏ tới một vùng dữ liệu. Tham số này chính là khai báo sau từ khoá FROM của câu lệnh SQL.

search_condition: điều kiện để trích lọc. Tham số này chính là khai báo sau từ khoá WHERE của câu lệnh SQL.

options: Cách dùng tương tự như hàm BS_SQL.

➔ Sử dụng hàm BS_SQL hay hàm BS_TABLE đều cho ra một kết quả. Hàm BS_TABLE phù hợp cho bài toán cũng như cách dùng đơn giản.

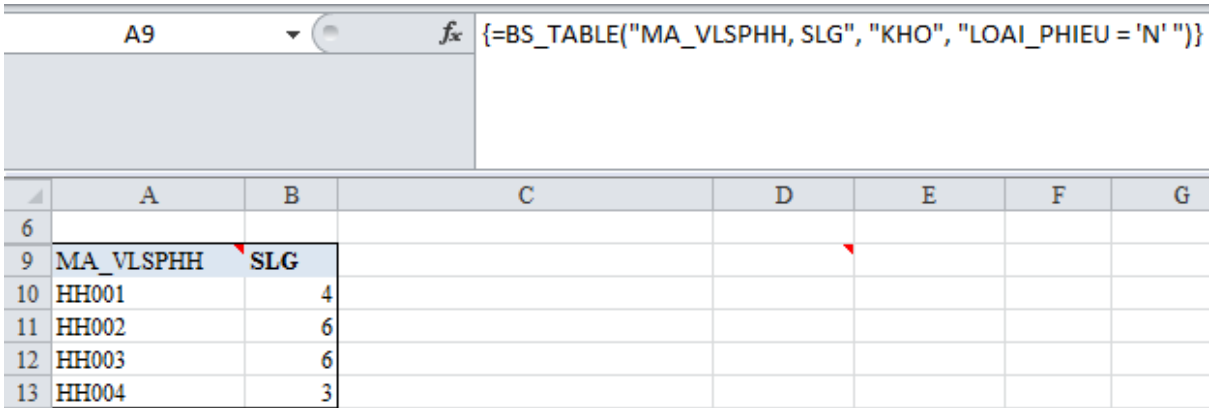
Ví dụ 53: Lập một danh sách gồm các cột: MA_VLSPHH, SLG từ sổ KHO mà loại phiếu là loại nhập (LOAI_PHIEU='N').

Các cách làm với BS_SQL:

```
=BS_SQL("SELECT MA_VLSPHH, SLG FROM KHO WHERE LOAI_PHIEU = 'N' ")
```

Các cách làm với BS_TABLE:

```
= BS_TABLE("MA_VLSPHH, SLG" , "KHO" , "LOAI_PHIEU = 'N' ")
```



Ví dụ 54: Lập một danh sách tổng hợp SLG nhập gồm các cột: MA_VLSPHH, SLG từ sổ KHO mà loại phiếu là loại nhập (LOAI_PHIEU='N'), sắp xếp cột SLG tăng dần.

Các cách làm với BS_SQL:

```
=BS_SQL("SELECT MA_VLSPHH, SUM(SLG)
        FROM KHO
        WHERE LOAI_PHIEU = 'N'
        GROUP BY MA_VLSPHH
        ORDER BY 2 ")
```

Các cách làm với BS_TABLE:

```
=BS_TABLE("MA_VLSPHH, SUM(SLG)" ,
          "KHO" ,
          "LOAI_PHIEU = 'N' ",
           "GROUPBY = MA_VLSPHH; ORDERBY = 2 ")
```

➔ GROUPBY **,** ORDERBY là viết liền (không có dấu cách), cách viết không giống từ khóa trong select GROUP BY, ORDER BY (có dấu cách).

Kết quả là:

A1 fx {=BS_TABLE("MA_VLSPHH, SUM(SLG)",
"KHO",
"LOAI_PHIEU = 'N' ",
"GROUPBY = MA_VLSPHH; ORDERBY = 2 ")}

	A	B	C	D	E	F	G	H
1	MA_VLSPHH	Expr1001						
2	HH005	2						
3	HH006	3						
4	BE1	5						
5	BM	10						
6	HH004	14						
7	HH001	16						
8	HH003	19						
9	HH002	33						

Chức năng trích lọc dữ liệu và làm báo cáo động trong Excel của Add-in A-Tools rất mạnh. Nó thực hiện trích lọc với tốc độ nhanh, chính xác, móc nối dữ liệu giữa các bảng với nhau, kết nối nhiều CSDL, cho phép liên kết từ máy tính này sang máy tính khác (mô hình client-server). Bạn học hết tất cả các hàm trên sẽ làm chủ được trong công việc xử lý dữ liệu Excel của mình. Trong bộ cài Add-in A-Tools cung cấp nhiều tập tin Excel có các ví dụ về lập bạn hãy mở và làm theo sẽ rõ hơn.
Thư mục lưu dữ liệu ví dụ: C:\A-Tools\HELP & DEMOS\

Toàn bộ dữ liệu và các ví dụ trong tài liệu này sử dụng bốn tập tin dưới đây:

Bai 1 - Huong dan su dung ham BS_SQL
Bai 2 - Cac ham ho tro nhom ham Database Functions.xls
Bai 3 Hàm CSDL (Database Functions).xls
Bai 4 - Huong dan tao bao cao.xls

Tệp "SoKeToanSQL.xls" chứa hệ thống các sổ kế toán được thiết kế bằng hàm BS_SQL của A-Tools chạy rất nhanh và linh hoạt đây là ví dụ tốt để bạn học và làm các sổ sách cho mình một cách chuyên nghiệp.

Thông tin chi tiết mới nhất về **Add-in A-Tools** mời các bạn xem tại đây:
http://www.bluesofts.net/Products/AddIns/A-Tools/Index.htm

Xem video hướng dẫn Add-in A-Tools

http://www.bluesofts.net/Products/AddIns/A-Tools/Help/DEMO/Index.htm
Hướng dẫn kết nối dữ liệu Excel qua mạng:
http://www.bluesofts.net/Products/AddIns/A-Tools/Help/DEMO/A-Tools_Network/A-Tools_Network.html

Xử lý lỗi khi Add-in A-Tools không khởi động cùng Microsoft Excel

http://www.bluesofts.net/Baiviet/Office/ExcelEnabledAddin/Index.html