

MISA Database Convention

[MariaDB]

Hướng dẫn tối ưu thiết kế Database

1. Tối ưu các thông tin đặc thù của dự án:

- Rà soát các thông tin đặc thù của dự án (thường hay truy xuất, tính toán, khả năng mở rộng của thông tin...)
- Thiết lập các ràng buộc (Constraints, index, unique...) để tăng tốc độ thực hiện query, lưu trữ dữ liệu
- Thực hiện tối ưu Length, DataType của các trường trong bảng
- Bổ sung các thông tin cho phép lưu trữ và tính toán trung gian

(VD: Thành tiền = Số lượng * Đơn giá, một số trường hợp Thành tiền không bằng Số lượng * Đơn giá do khách hàng tự nhập hoặc do số lẻ...)

2. Tối ưu các bảng, các mối quan hệ giữa các bảng từ các ràng buộc về nghiệp vụ đặc thù

- Rà soát các yêu cầu đặc thù, các bảng, mối quan hệ giữa các bảng, khối lượng lưu trữ, số lượng truy xuất, tần suất đọc/ghi...
- Tạo ra các bảng/thông tin phụ lưu trữ dữ liệu để tối ưu việc truy xuất, tính toán (có thể chấp nhận dư thừa dữ liệu)
- Thiết lập các ràng buộc giữa các bảng (Thiết lập Cascade, Trigger...)

3. Tối ưu các bảng, dữ liệu của hệ thống, cách thức lưu trữ trên các bảng chứa dữ liệu chung hoặc dữ liệu cần mang đi triển khai

- Rà soát các thông tin/dữ liệu cần mang đi triển khai cho khách hàng, cần lưu trữ phục vụ tính tiện ích của sản phẩm

(VD: Hệ thống tài khoản trong hệ thống kế toán, các dữ liệu mặc định/demo...)

- Rà soát kiến trúc hệ thống, cách thực hiện lưu trữ các thông tin hệ thống

(VD: hệ thống loại chứng từ, các Enum hệ thống, các tùy chọn...)

- Tối ưu các dữ liệu trong các bảng, kiểu dữ liệu, các giá trị mang đi, các giá trị mặc định trên các bảng có tính chất hệ thống hoặc các bảng dữ liệu mang đi triển khai

(VD: Các bảng chứa danh sách report, vai trò admin trên hệ thống mặc định mang đi...)

- Thiết lập các thông tin/dữ liệu cần lưu trữ phục vụ việc sử dụng: các giá trị ngầm định, các giá trị tham số lưu lại để dùng cho lần sau,...

1. Quy tắc đặt tên

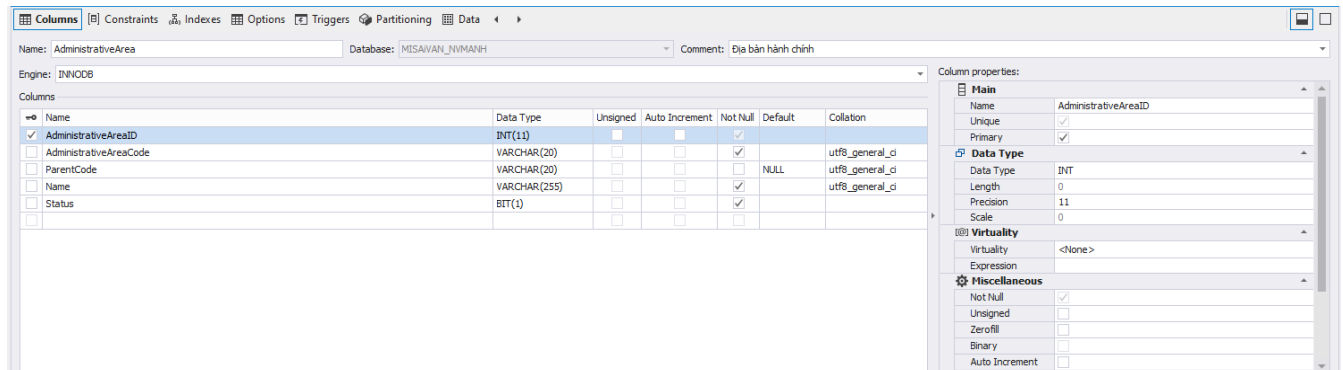
Mục	Nội dung
Quy tắc chung	<ul style="list-style-type: none"> Sử dụng tiếng Anh số ít để đặt tên kiểu Pascal case. → VD: InventoryItem Viết liền, không sử dụng dấu cách " " để đặt tên. → VD: không đặt tên bảng là "Order Detail" mà phải đặt là "OrderDetail". Không dùng số để đặt tên (trừ số hiệu báo cáo). → VD: không đặt tên bảng là Cash1, Cash2. Phải đặt là Cash và CashDetail. Tuyệt đối tránh không sử dụng các từ khóa đặc biệt (reserved words) để đặt tên đối tượng. → VD: phải đặt tên bảng là DBOption thay cho Option
Tên cơ sở dữ liệu	<ul style="list-style-type: none"> Đặt tên CSDL là tên sản phẩm. → VD: MISACRM2008 Thông thường, trong quá trình phát triển sản phẩm ta cần hai CSDL. Một dùng cho mục đích phát triển, một dùng cho mục đích đóng gói chương trình. Quy tắc đặt tên hai CSDL này như sau: <ul style="list-style-type: none"> Tên CSDL phát triển = Tên sản phẩm + "_Development" ▪ VD: MISAMimosa2009_Development Tên CSDL đóng gói = Tên sản phẩm + "_Distribution" ▪ VD: MISAMimosa2009_Distribution
Tên bảng	<ul style="list-style-type: none"> Tên bảng viết hoàn toàn bằng tiếng Anh Các bảng thuộc phân hệ thì có prefix là tên quy ước của phân hệ đó. → VD: <u>FixedAsset</u>Depreciation; <u>Inventory</u>InwardOutward Các bảng có quan hệ mật thiết đến nhau (quan hệ 1 – n, master – detail) phải có prefix giống nhau. → VD: InventoryItem – InventoryItemCategory; SaleInvoice – SaleInvoiceDetail Tên bảng không được chứa ký tự số nhiều: "s" Tên trường hoàn toàn bằng tiếng Anh.
Tên trường	<ul style="list-style-type: none"> Tên trường của một bảng nên có prefix là tên bảng (trừ khóa ngoại). → VD: bảng Contact có các trường ContactID, ContactName, ... Khóa chính phải có hậu tố (suffix) là "ID". → VD: AccountingObjectID Tên trường khóa ngoại phải đặt trùng với tên khóa chính ở bảng gốc. → VD: bảng Employee có khóa ngoại là DepartmentID trùng với tên khóa chính ở bảng Department. Không đưa kiểu dữ liệu vào tên trường vì điều này sẽ ngăn cản việc có thể phải đổi kiểu dữ liệu sau này.

<div>Quy tắc đặt tên</div> <div><div>▪ Key</div><div>▪ Index</div><div>▪ Trigger</div><div>▪ View</div><div>▪ Stored</div><div>▪ Function</div></div>	<table><tr><th>Database Object</th><th>Name Syntax</th><th>Sample</th></tr><tr><td>Primary Key</td><td>PK_TableName</td><td>PK_AccountBalance</td></tr><tr><td>Index</td><td>IX_TableName_FieldName</td><td>IX_FixedAsset_FixedAssetID_FixedAsserYear</td></tr><tr><td>Trigger</td><td><div>✓ trgInsertTableName</div><div>✓ trgUpdateTableName</div><div>✓ trgDeleteTableName</div></td><td>trgDeleteAccount</td></tr><tr><td>View</td><td>View_</td><td>View_AccountList</td></tr><tr><td>Stored Procedure</td><td>Proc_</td><td>Proc_INVInwardStock</td></tr><tr><td>Function</td><td>Func_</td><td>Func_GetAccountClosingDebit</td></tr></table> <div>Riêng với View, Stored và Function, khi đặt tên tuân thủ nguyên tắc phân loại như sau:</div> <div><div>▪ Với View, Stored, Function dùng cho việc CRUD:</div><div><div><div>“View_”/“Proc_”/“Func_” + Mã phân hệ/Mã chức năng + “_” + Mục đích</div><div><div>Trường hợp phục vụ nhiều phân hệ nào thì bỏ qua mã phân hệ.</div><div>→ VD: Proc_PL_GetYearPlanExpenseAmount (PL: chức năng Lập dự toán) Func_GetAccountClosingDebit</div></div></div><div>▪ Với View, Stored, Function dùng cho việc in chứng từ (phần mềm kế toán):</div><div><div><div>“View_”/“Proc_”/“Func_” + Mã phân hệ + “V_” + Tên chứng từ + “_” + Số hiệu</div><div><div>→ VD: Proc_CAV_Receipt_01TT (In phiếu thu) ProcINV_Inward_01VT (In phiếu nhập)</div><div>▪ Với View, Stored, Function dùng cho việc in báo cáo:</div><div><div><div>“View_”/“Proc_”/“Func_” + Mã phân hệ + “R_” + Tên báo cáo + “_” + Số hiệu</div><div><div>→ VD: Proc_CAR_CashBook_S13X (Sổ quỹ) Proc_FAR_FixedAssetLedger_F10X (Sổ tài sản)</div></div></div></div></div><div>Lưu ý:</div><div><div>▪ Không bắt đầu tên Stored Procedure bằng dt_, sp_ hoặc xp_, vì nó được đặt cho các Stored Procedure của hệ thống.</div><div>▪ Không bắt đầu tên Function bằng fn_, vì nó được đặt cho các Function của hệ thống.</div></div><div>Quy tắc đặt tên tham số và biến trong Stored Procedure và Function: Kiểu pascal case, prefix là "@". VD: @FormDate, @AccountID</div></div></div></div></div>	Database Object	Name Syntax	Sample	Primary Key	PK_TableName	PK_AccountBalance	Index	IX_TableName_FieldName	IX_FixedAsset_FixedAssetID_FixedAsserYear	Trigger	<div>✓ trgInsertTableName</div> <div>✓ trgUpdateTableName</div> <div>✓ trgDeleteTableName</div>	trgDeleteAccount	View	View_	View_AccountList	Stored Procedure	Proc_	Proc_INVInwardStock	Function	Func_	Func_GetAccountClosingDebit
	Database Object	Name Syntax	Sample																			
	Primary Key	PK_TableName	PK_AccountBalance																			
	Index	IX_TableName_FieldName	IX_FixedAsset_FixedAssetID_FixedAsserYear																			
	Trigger	<div>✓ trgInsertTableName</div> <div>✓ trgUpdateTableName</div> <div>✓ trgDeleteTableName</div>	trgDeleteAccount																			
	View	View_	View_AccountList																			
	Stored Procedure	Proc_	Proc_INVInwardStock																			
	Function	Func_	Func_GetAccountClosingDebit																			
	<div>Quy tắc đặt tên Diagram</div>	Tên Diagram đánh theo tên phân hệ hoặc mã phân hệ.																				

→ Ví dụ: FixAsset (FA), LedgerBalance,...

2. Quy tắc thiết kế bảng

- Phải sử dụng Database Diagram để thiết kế bảng. Thiết lập custom view cho bảng như sau:
- Nên sử dụng Navicat hoặc dbForge để thiết kế



Các bước thiết kế một bảng (table) trong cơ sở dữ liệu SQLServer

1. Đặt tên bảng.
2. Viết description cho bảng.
3. Tạo cột (column) cho bảng:
 - a. Đặt tên cột.
 - b. Chọn kiểu dữ liệu (DataType).
 - c. Thiết lập AllowNull.
 - d. Thiết lập giá trị ngầm định (DefaultValue).
 - e. Viết description cho cột.
4. Thiết lập khóa chính cho bảng:
 - a. Nếu khóa chính là Integer: thiết lập identity tự tăng cho khóa chính
 - b. Nếu khóa chính là **UniqueIdentifier** tương ứng kiểu dữ liệu **GUID** trong .NET – đặt kiểu dữ liệu là **char(36)**: không thiết lập giá trị mặc định (do các phiên bản từ 5.7 MariaDB không hỗ trợ function tự sinh mã qua function UUID()). **Sử dụng Trigger** để thiết lập giá trị mặc định khi Insert dữ liệu.
5. Xác định khóa ngoại cho bảng.
6. Thiết lập quan hệ với các bảng khác (relation).
7. Thiết lập Index cho bảng.
8. Viết diễn giải bằng tiếng Việt vào thuộc tính Description cho bảng đó. VD: bảng AccountingObject có Description là "Danh mục đối tượng kế toán"

- Quy tắc thiết kế các trường (Column) cho bảng:
 - Xác định rõ ràng Kiểu dữ liệu, Kích thước, Allow NULL, Giá trị ngầm định cho trường đó.
 - Bắt buộc phải viết diễn giải vào thuộc tính Description của tất cả các trường. Diễn giải cho trường gồm:
 - Tên/ý nghĩa của trường bằng tiếng Việt.
 - Miền giá trị (nếu có). VD: trường CashWithdrawTypeID có Description là: "Nghịệp vụ: 0: Không chọn (Mặc định); 1: Tạm ứng; 2: Thực chi; 3: Chi từ tạm ứng; 4: Thanh toán tạm ứng; 5: Khôi phục"
 - Việc tạo trường cho bảng tuân theo nguyên tắc dưới đây:

Trường	Kiểu	Độ dài	Allow NULL	Ngầm định	Ví dụ
ID toàn cục	char(36)			(sử dụng Trigger để tạo mặc định qua UUID())	LeadID
Alias/Code	varchar	20			AccountingObjectAlias AccountNumber
Tên người	varchar	100	Tùy	''	ContactName
Tên các loại (trừ tên người). VD: Tên công ty, phòng ban, tên vật tư, tên nhóm khách hàng.	varchar	255	Tùy	''	AccountingObjectName
Chức danh	varchar	100		''	ContactTitle
Địa chỉ	varchar	255		''	ContactAddress
Diễn giải	varchar	255		''	Description
Số ĐT/Fax/Mobile	varchar	50		''	ContactMobile
Email	varchar	100		''	ContactEmail
Website	varchar	255		''	Website
Ngày	date		Tùy		PostedDate
Ngày giờ	datetime		Tùy		StartTime (Appointment)
Số CMT Thẻ tín dụng Tài khoản ngân hàng	varchar	25		''	BankAccount
Tài liệu đính kèm	varchar	255		''	DocumentIncluded
Số tiền/Đơn giá	decimal(18,4)			0	DebitAmount
Số lượng/Tỉ lệ/Hệ số	decimail (18,4)			0	DepreciationRate

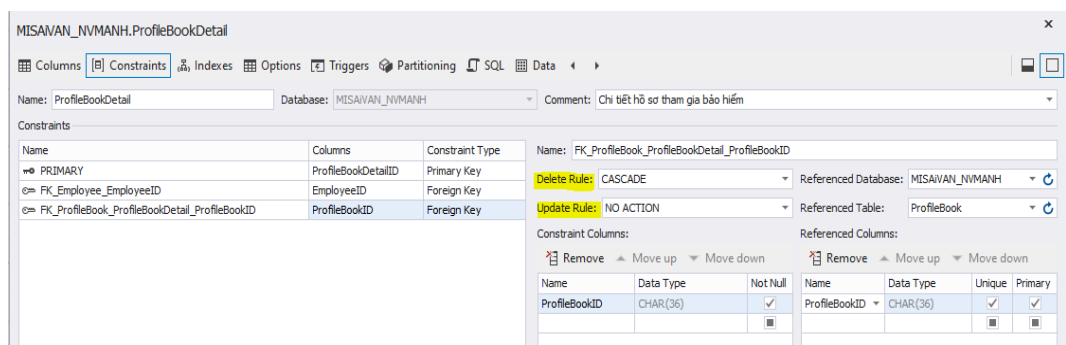
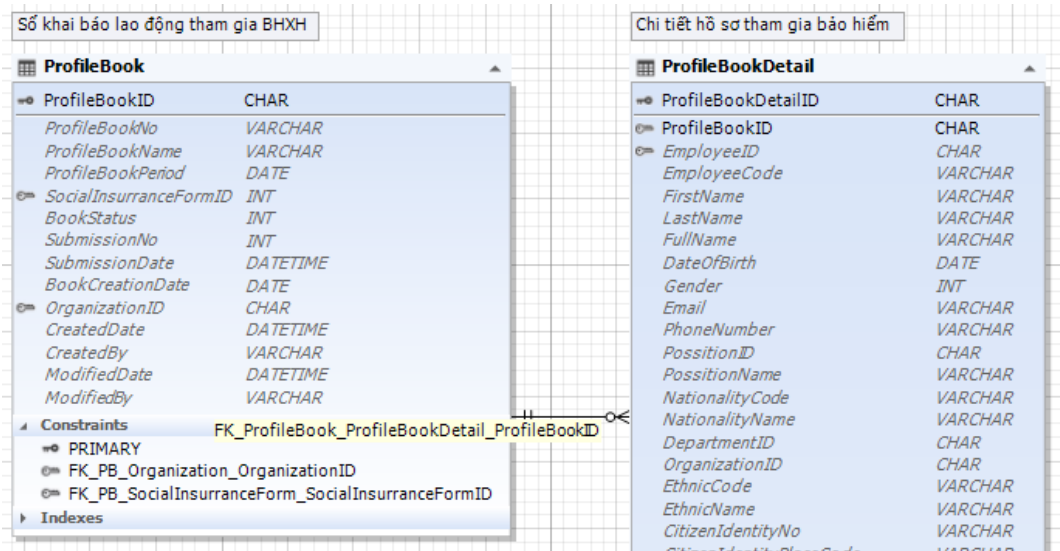
Chú ý: Với các trường đặc biệt độ dài có thể tùy theo yêu cầu nghiệp vụ/yêu cầu người dùng để thiết lập cho phù hợp

❖ Các trường sau bắt buộc phải tạo Index

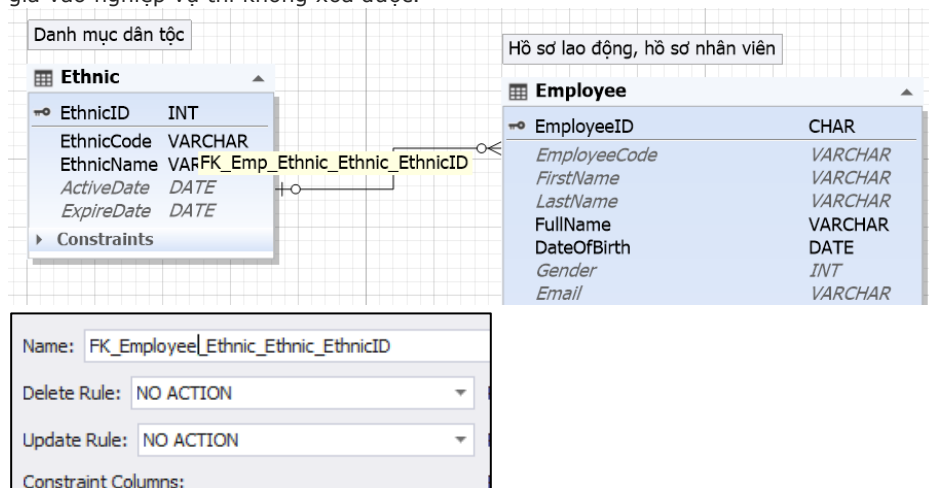
- Foreign Key
- Alias/Code (thiết lập: Is Unique = TRUE)

▪ Relation:

- Các bảng Master – Detail bắt buộc phải đặt relation 1 – n và Cascade Delete. Không cần đặt Cascade Update vì MISA thiết kế khóa chính của Master là bất biến.



- Giữa bảng Danh mục (chứa khóa chính) và bảng sử dụng Danh mục (chứa khóa ngoại) phải thiết lập relation 1 – n nhưng không đặt Cascade (No Action) để đảm bảo danh mục đã tham gia vào nghiệp vụ thì không xóa được.



3. Quy tắc thiết kế đối tượng truy vấn (Procedure, Function)

Quy tắc viết comment

- Mọi đối tượng truy vấn bắt buộc phải có comment, với các function/ procedure khi khai báo các tham số đầu vào phải chỉ rõ là **@input** hay **@output**:

General

Name: Proc_GetLongTermDiseaseByID Database: MISAIVAN_NVMANH Comment: Lấy đối tượng qua ID - NDVU

Parameters

Name	Type	Direction
LongTermDiseaseID	INT(11)	@ Input
Status	BIT	@ Input

Options

Invoker: ☐ Invoker ☒ Definer

Definer: root@'%'

SQL Access: CONTAINS SQL

☐ Deterministic

```
CREATE DEFINER = 'root'@'%'  
PROCEDURE MISAIVAN_NVMANH.Proc_GetLongTermDiseaseByID(  
    IN LongTermDiseaseID INT(11),  
    IN Status bit -- Trạng thái (1- đang sử dụng; 0- đã ngưng sử dụng)  
)  
COMMENT 'Lấy đối tượng qua ID - NDVU'  
BEGIN  
    SELECT ltd.LongTermDiseaseID,  
           ltd.LongTermDiseaseName,  
           ltd.LongTermDiseaseICD10,  
           ltd.LongTermDiseaseTT33,  
           ltd.LongTermDiseaseTT34,  
           ltd.LongTermDiseaseTT46,  
           ltd.Sort FROM LongTermDisease ltd WHERE ltd.LongTermDiseaseID = LongTermDiseaseID;  
END
```

- Không sử dụng SELECT * mà cần chỉ rõ các cột dữ liệu nào cần sử dụng.
- Comment cho đoạn SQL:
 - Comment ý nghĩa, giá trị đặc biệt của từng tham số (bên phải).
 - Ví dụ: @Status bit > như hình ảnh mô tả
 - Những đoạn SQL phức tạp cần có comment gắn liền bên trên để chú giải.
 - Những đoạn SQL được sửa đổi (modified), bổ sung (added) hoặc rem (removed) bởi người không phải tác giả cần có comment rõ ngay tại nơi sửa đổi, bổ sung: người sửa, ngày sửa, mục đích

4. Từ khóa tránh sử dụng để đặt tên đối tượng CSDL

- (Những từ bị bôi xanh trong giao diện soạn query của SQL Server thì cũng phải tránh)

A	END	MOUNT	SERIALIZABLE
ABORT	ENDDATA	MOVE	SERVICE
ABS	ENDDISPLAY	NAMED	SESSION
ABSOLUTE	ENDEXEC	NAMES	SESSION_USER
ACCESS	END-EXEC	NATIONAL	SET
ACOS	ENDFORMS	NATURAL	SETS
ACQUIRE	ENDIF	NCHAR	SETUSER
ACTION	ENDLOOP	NCLOB	SIN
ACTIVATE	EQUALS	NEW	SIMPLE
ADA	ENDSELECT	NEXT	SIGN
ADD	ENDWHILE	NHEADER	SHUTDOWN
ADDFORM	ERASE	NO	SHORT
ADMIN	ERRLVL	NOARCHIVELOG	SHARE
AFTER	ERREXIT	NOAUDIT	SHARED
AGGREGATE	ESCAPE	NOCACHE	SETUSER
ALIAS	EVENTS	NOCHECK	SIZE
ALL	EVERY	NOCOMPRESS	SLEEP
ALLOCATE	EXCEPT	NOCYCLE	SMALLINT
ALTER	EXCEPTION	NOECHO	SNAPSHOT
AN	EXCEPTIONS	NOMAXVALUE	SOME
ANALYZE	EXCLUDE	NOMINVALUE	SORT
AND	EXCLUDING	NONCLUSTERED	SOUNDEX
ANY	EXCLUSIVE	NONE	SPACE
APPEND	EXEC	NOORDER	SPECIFIC
ARCHIVE	EXECUTE	NORESETLOGS	SPECIFICTYPE
ARCHIVELOG	EXISTS	NORMAL	SQL
ARE	EXIT	NOSORT	SQLEXCEPTION
ARRAY	EXP	NOT	SQLBUF
ARRAYLEN	EXPLAIN	NOTFOUND	SQLCA
AS	EXPLICIT	NOTRIM	SQLCODE
ASC	EXTENT	NOWAIT	SQLERROR
ASCII	EXTERNAL	NULL	SQLSTATE
ASIN	EXTERNALLY	NULLIF	SQLWARNING
ASSERTION	EXTRACT	NULLVALUE	SQRT
AT	FALSE	NUMBER	START
ATAN	FETCH	NUMERIC	STATE
AUDIT	FIELD	OBJECT	STATEMENT
AUTHORIZATION	FIELDPROC	NUMPARTS	STATIC
AVG	FILE	NVL	STRUCTURE
AVGU	FILLFACTOR	OBID	STATISTICS
BACKUP	FINALIZE	ODBCINFO	STOGROUP
BECOME	FINALIZE	OF	STOP
BEFORE	FIRST	OFF	STORAGE
BEGIN	FLOAT	OFFLINE	STORPOOL
BETWEEN	FLOOR	OFFSETS	SUBMENU
BIGINT	FLOPPY	OLD	SUBPAGES

BINARY	FLUSH	ON	SUBSTR
BIND	FOR	ONCE	SUBSTRING
BINDING	FORCE	ONLINE	SUCCESSFUL
BIT	FORMDATA	ONLY	SUFFIX
BLOB	FORMINIT	OPEN	SUM
BLOCK	FORMS	OPERATION	SYSTEM_USER
BODY	FORTRAN	OPENDATASOURCE	SUMU
BOOLEAN	FOREIGN	OPENQUERY	SWITCH
BOTH	FOUND	OPENROWSET	SYNONYM
BREADTH	FREELIST	OPTIMAL	SYSCAT
BREAK	FREELISTS	OPTIMIZE	SYSDATE
BREAKDISPLAY	FREETEXT	OPTION	SYSFUN
BROWSE	FREETEXTTABLE	OR	SYSIBM
BUFFERPOOL	FROM	ORDER	SYSSTAT
BULK	FREE	ORDINALITY	SYSTEM
BY	FULL	OUT	SYSTIME
BYREF	FUNCTION	OUTER	SYSTIMESTAMP
CACHE	GENERAL	OUTPUT	TABLE
CALL	GET	OVER	TABLEDATA
CALLPROC	GETCURRENTCONNECTION	OVERLAPS	TABLES
CANCEL	GETFORM	OWN	TABLESPACE
CAPTURE	GETOPER	PACKAGE	TAN
CASCADE	GETROW	PAD	TAPE
CASCADEED	GLOBAL	PARAMETER	TEMP
CASE	GO	PARAMETERS	TEMPORARY
CAST	GOTO	PAGE	TERMINATE
CATALOG	GRANT	PAGES	THAN
CCSID	GRANTED	PARALLEL	TEXTSIZE
CEILING	GRAPHIC	PART	THEN
CHANGE	GREATEST	PARTIAL	THREAD
CHAR	GROUP	PATH	TIME
CHARACTER	GROUPING	POSTFIX	TIMEOUT
CHARTOROWID	GROUPS	PASCAL	TIMESTAMP
CHECK	HASH	PCTFREE	TIMEZONE_HOUR
CLASS	HAVING	PCTINCREASE	TIMEZONE_MINUTE
CLOB	HOST	PCTINDEX	TINYINT
CHECKPOINT	HELP	PCTUSED	TO
CHR	HELPERFILE	PERCENT	TOP
CLEANUP	HOLDLOCK	PERM	TPE
CLEAR	HOURL	PERMANENT	TRACING
CLEARROW	HOURS	PERMIT	TRAILING
CLOSE	IDENTIFIED	PI	TRAN
CLUSTER	IDENTITY	PIPE	TRANSACTION
CLUSTERED	IGNORE	PLAN	TRANSLATE
COALESCE	IDENTITYCOL	PLI	TRANSLATION
COBOL	IF	POSITION	TREAT
COLGROUP	IFNULL	POWER	TRIGGER
COLLATE	IIMESSAGE	PRECISION	TRIGGERS
COLLATION	IIPRINTF	PREFIX	TRIM
COLLECTION	IMMEDIATE	PREORDER	TRUE
COLUMN	IMPORT	PREPARE	TRUNCATE
COMMAND	IN	PRESERVE	TSEQUAL

COMMENT	INCLUDE	PRIMARY	TYPE
COMMIT	INCLUDING	PRINT	UID
COMPLETION	INCREMENT	PRINTSCREEN	UNCOMMITTED
COMMITTED	INDEX	PRIOR	UNDER
COMPILE	INDEXPAGES	PRIQTY	UNION
COMPLEX	INDICATOR	PRIVATE	UNIQUE
COMPRESS	INITCAP	PRIVILEGES	UNKNOWN
COMPUTE	INITIAL	PROC	UNNEST
CONCAT	INITIALIZE	PROCEDURE	UNLIMITED
CONFIRM	INITIALLY	PROCESSEXIT	UNLOADTABLE
CONNECT	INTRANS	PROFILE	UNSIGNED
CONNECTION	INITTABLE	PROGRAM	UNTIL
CONSTRAINT	INNER	PROMPT	UP
CONSTRAINTS	INOUT	PUBLIC	UPDATE
CONSTRUCTOR	INPUT	PUTFORM	UPDATETEXT
CONTAINS	INSENSITIVE	PUTOPER	UPPER
CONTAINSTABLE	INSERT	PUTROW	USAGE
CONTENTS	INSERTROW	QUALIFICATION	USE
CONTINUE	INSTANCE	QUARTER	USER
CONTROLFILE	INSTR	QUOTA	USING
CONTROLROW	INT	RADIANS	UUID
CONVERT	INTEGER	RAISE	VALIDATE
COPY	INTEGRITY	RAISERROR	VALIDPROC
CORRESPONDING	INTERFACE	RAND	VALIDROW
COS	INTERSECT	RANGE	VALUE
COUNT	INTERVAL	RAW	VALUES
COUNTU	INTO	READ	VARBINARY
CREATE	IS	READS	VARCHAR
CROSS	ISOLATION	READTEXT	VARIABLE
CUBE	ITERATE	REAL	VARIABLES
CURRENT	JOIN	RECURSIVE	VARYING
CURRENT_DATE	KEY	REF	VCAT
CURRENT_PATH	KILL	RECONFIGURE	VERSION
CURRENT_ROLE	LABEL	RECORD	VIEW
CURRENT_TIME	LANGUAGE	RECOVER	VOLUMES
CURRENT_TIMESTAMP	LARGE	REDISPLAY	WAITFOR
CURRENT_USER	LAST	REFERENCES	WEEK
CURSOR	LATERAL	REFERENCING	WHEN
CVAR	LAYER	RELATIVE	WHENEVER
CYCLE	LEADING	REGISTER	WHERE
DATA	LEAST	RELEASE	WHILE
DATABASE	LEFT	RELOCATE	WITH
DATAFILE	LESS	REMOVE	WITHOUT
DATAHANDLER	LENGTH	RENAME	WORK
DATAPAGES	LEVEL	REPEAT	WRITE
DATE	LIKE	REPEATABLE	WRITETEXT
DAY	LIMIT	REPEATED	YEAR
DAYOFMONTH	LINENO	REPLACE	YEARS
DAYOFWEEK	LINK	REPLICATE	ZONE
DAYOFYEAR	LIST	REPLICATION	
DAYS	LISTS	RESET	
DBA	LOAD	RESETLOGS	

DBCC	LOADTABLE	RESOURCE
DBSPACE	LOCAL	RESTORE
DEALLOCATE	LOCALTIME	RESTRICT
DEC	LOCALTIMESTAMP	RESULT
DECIMAL	LOCATOR	RESTRICTED
DECLARATION	LOCATE	RESUME
DECLARE	LOCK	RETRIEVE
DECODE	LOCKSIZE	RETURN
DEFAULT	LOG	RETURNS
DEFERRABLE	LOGFILE	REUSE
DEFERRED	LONG	REVOKE
DEFINE	LONGINT	RIGHT
DEFINITION	LOWER	ROLE
DEGREES	LPAD	ROLES
DELETE	LTRIM	ROLLBACK
DEPTH	LVARBINARY	ROLLUP
DEREF	LVARCHAR	ROUTINE
DELETEROW	MAIN	ROW
DENY	MANAGE	ROWS
DESC	MANUAL	ROWCOUNT
DESCRIBE	MAP	ROWGUIDCOL
DESCRIPTOR	MATCH	ROWID
DESTROY	MAX	ROWIDTOCHAR
DHTYPE	MAXDATAFILES	ROWLABEL
DESTRUCTOR	MAXEXTENTS	ROWNUM
DETERMINISTIC	MAXINSTANCES	ROWS
DICTIONARY	MAXLOGFILES	RPAD
DIAGNOSTICS	MAXLOGHISTORY	RRN
DIRECT	MAXLOGMEMBERS	RTRIM
DISABLE	MAXTRANS	RULE
DISCONNECT	MAXVALUE	RUN
DISK	MENUITEM	RUNTIMESTATISTICS
DISMOUNT	MESSAGE	SAVE
DISPLAY	MICROSECOND	SAVEPOINT
DISTINCT	MICROSECONDS	SCHEDULE
DISTRIBUTE	MIN	SCHEMA
DISTRIBUTED	MINEXTENTS	SCN
DO	MINUS	SCREEN
DOMAIN	MINUTE	SCROLL
DOUBLE	MODIFIES	SCOPE
DOWN	MINUTES	SEARCH
DROP	MINVALUE	SCROLLDOWN
DUMMY	MIRROREXIT	SCROLLUP
DUMP	MOD	SECOND
DYNAMIC	MODE	SECONDS
EACH	MODIFY	SECQTY
EDITPROC	MODULE	SECTION
ELSE	MONEY	SEGMENT
ELSEIF	MONTH	SELECT
ENABLE	MONTHS	SEQUENCE