StorHouse®

StorHouse/RM Glossary

StorHouse/RM Release 3.4

Publication Number
900112 Rev. K

September 17, 2008

FileTek

All rights reserved. No part of this publication may be reproduced, translated, stored in any electronic retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of FileTek, Inc.

Copyright © 1996-2008 FileTek, Inc. As an Unpublished Licensed Work.
Publication Number: 900112 Rev. K

NOTICE: U.S. GOVERNMENT USERS

This notice applies to all acquisitions of this work by or for the U.S. Government ("Government"), or by any prime contractor or subcontractor (at any tier) under any contract, cooperative agreement or other activity with the Government. By accepting delivery of this work, the Government agrees that this work and the Licensed Program(s) described herein qualify as "commercial" computer software within the meaning of the acquisition regulation(s) applicable to this procurement. The terms of conditions of the license for the Licensed Program(s) shall pertain to the Government's use and disclosure of this work and the Licensed Program(s), and shall supersede any conflicting contractual terms or conditions. If the license for this work and the Licensed Program(s) fails to meet the Government's need or is inconsistent in any respect with Federal law, the Government agrees to return this work and the Licensed Program(s), unused, to FileTek, Inc. The following additional statement applies only to acquisitions governed by DFARS Subpart 227.4 (October 1988) "Restricted Rights - Use, duplication and disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 (OCT. 1988)." Unpublished licensed work property of FileTek, Inc. Unauthorized use, duplication or distribution prohibited. All rights reserved. A copyright notice on this work and/or on the Licensed Program(s) by itself does not constitute publication or public disclosure of this work or the Licensed Program(s). The contractor/manufacturer is:

FileTek, Inc.
9400 Key West Avenue
Rockville, Maryland 20850

Information in this document is subject to change without notice and does not represent a commitment on the part of FileTek, Inc. Further, FileTek, Inc. reserves the right to supplement the document with information not available at the time of creation of the document. FILETEK, INC. PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, AND CANNOT WARRANT THE RESULTS YOU MAY OBTAIN USING THE DOCUMENT. IN NO EVENT SHALL FILETEK, INC. BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OR DATA, INTERRUPTION OF BUSINESS, OR FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND, EVEN IF FILETEK, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES ARISING FROM ANY DEFECT OR ERROR IN THIS PUBLICATION. Some states or jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

FileTek and StorHouse are registered U.S. trademarks of FileTek, Inc. VRAM is a U.S. trademark of FileTek, Inc. All other brand or product names are trademarks or registered trademarks of their respective owners.

Documentation for FileTek's StorHouse product. Protected by the following U.S. Patents: 4,864,572; 5,247,660; 5,727,197; 6,049,804.

FileTek

Welcome

The *StorHouse/RM Glossary* defines the terms and acronyms used in the StorHouse/RM User Document Set. This document is intended as a stand-alone reference manual as well as a companion to other StorHouse/RM documents.

StorHouse family of products

StorHouse® is the FileTek enterprise-wide solution for managing the capture, storage, movement, and access of gigabytes to petabytes of relational and non-relational detail data. StorHouse technology combines industry-leading, scalable storage devices and Open System processors with specialized storage management and relational database management system (RDBMS) software components.

StorHouse/SM

StorHouse/SM, the storage management component, controls a hierarchy of storage devices composed of cache, redundant array of independent disks (RAID), Serial Advanced Technology Attached (SATA) disks, massive array of idle disks (MAID), erasable and write-once-read-many (WORM) optical disk jukeboxes, and erasable and WORM automated tape libraries. StorHouse/SM is also responsible for automating critical system management tasks, like data migration, backup, and recovery.

StorHouse/RM

StorHouse/RM, the RDBMS component, works in conjunction with StorHouse/SM to specifically store, access, and move relational data. StorHouse/RM provides row-level SQL access to high volumes of detail data on any layer in the StorHouse storage hierarchy, including tape. SQL access is available from different platforms through a variety of industry-standard protocols. StorHouse/RM runs on Sun™ Solaris™, Hewlett-Packard HP-UX, and IBM® AIX platforms.

StorHouse/Control Center

StorHouse/Control Center (CC) is the FileTek Windows®-based network computing system for providing administrative control of the StorHouse family of products. StorHouse/Control Center works with StorHouse/SM release 4.2 and higher and consists of one or more StorHouse/Control Center servers that communicate with StorHouse/Control Center clients over an IP network.

The *StorHouse/Control Center server*, which runs on Windows NT, XP Pro, and 2000 platforms, provides network connectivity to StorHouse. The *StorHouse/Control Center clients*, which run on Windows 95, 98, 2000, XP Pro, and NT platforms, consist of one or more graphical user interface (GUI) modules for performing StorHouse system and database administration tasks, configuring and managing StorHouse/Control Center servers, and analyzing and monitoring StorHouse activity and performance.

Cross-reference terms

This document uses the following cross-reference terms:

- *See* – refer to this term for the definition.
- *See also* – refer to this term for additional information.

- *Compare* – refer to this term for a similar meaning.
- *Contrast* – refer to this term for an opposite meaning.

Related documentation

This section lists the documents in the StorHouse/RM and StorHouse/SM documentation sets as well as miscellaneous publications that may help you understand and use StorHouse.

StorHouse/RM publications

For more information about StorHouse/RM, refer to these publications:

- The *StorHouse/RM Concepts*, publication number 900132, defines the structures that store relational data on StorHouse and the facilities that access and manage that data.

- The *StorHouse Database Administration Guide*, publication number 900108, explains how to perform StorHouse database administration tasks such as creating user tables and indexes, managing accounts and privileges, and setting up user tablespaces.

- The *FileTek MVS Data Loader Utility Manual*, publication number 900109, describes how to load data into StorHouse user tables from an MVS host.

- The *FileTek FTP Data Loader Manual*, publication number 900115, explains how to load data into StorHouse user tables with File Transfer Protocol (FTP).

- The *FileTek FTP Data Unloader Manual*, publication number 900137, explains how to unload data from StorHouse databases using FTP. It describes the UNLOAD control statement you prepare to format result data, the SELECT statement you prepare to select the data to unload, and the subset of

FTP commands you use to transfer control information and to receive result data.

- The *StorHouse SQL Reference Manual,* publication number 900111, describes the SQL statements, predicates, and functions supported by StorHouse. It also lists SQL status codes.

- The *StorHouse/RM SQL and Utility Quick Reference*, publication number 900122, contains formats and examples of SQL statements, predicates, functions, data loader statements, data unloader statement, and utilities.

- The *StorHouse ESQL Manual,* publication number 900121, explains how to use StorHouse SQL in application programs.

- The *StorHouse/RM Metadata Conversion Manual*, publication number 900142, explains how to convert metadata from one StorHouse/RM release to another.

StorHouse/SM publications

Some of the documents in the StorHouse/SM documentation set that may help you understand StorHouse are as follows:

- The *StorHouse Glossary,* publication number 900027, defines the terminology used in the StorHouse/SM documentation set.

- The *StorHouse Concepts and Facilities Manual*, publication number 900026, defines the basic concepts, structures, and functions of StorHouse.

- The *Command Language Reference Manual*, publication number 900005, describes StorHouse Command Language commands, modifiers, privileges, and system parameters.

- The *System Administrator's Guide*, publication number 900007, describes system recovery, account administration, and storage management procedures and concepts for StorHouse.
- The *Messages and Codes Manual,* publication number 900032, lists all StorHouse system and host software messages.

Miscellaneous publication

The following publication contains information for those customers with DB2 Universal Database (UDB). The *StorHouse/UDB Link Installation and Configuration Manual*, publication number 900163, explains how to install the StorHouse/UDB Link and how to configure DB2 Universal Database (DB2 UDB) to work with StorHouse. It also discusses security management, data modeling, and DB2 components that support federation.

List of terms

access privilege

A privilege that enables a StorHouse account to bypass various security checks and to perform specific StorHouse functions. For example, the StorHouse SQLADMIN access privilege gives an account DBA access to all StorHouse databases. *See also* privilege and SQLADMIN privilege.

Access Time Factor (ATF)

An attribute that works with the StorHouse migrate function to keep data that is most likely to be accessed in the StorHouse performance buffer while maintaining a supply of free space. StorHouse migrates files with higher ATF values from the performance buffer to their destination file sets before files with lower ATF values. You set this attribute when creating or altering a subspace in a user tablespace.

account, StorHouse

A collection of administrative data used to monitor and control the use of StorHouse. Each account has its own set of StorHouse privileges. One account can be dedicated to one user or shared by multiple users.

account default user tablespace

The user tablespace to be used as the default when an account omits a TABLE SPACE clause on a CREATE TABLE statement. Each StorHouse account can have an account default user tablespace. *See also* user tablespace and database default user tablespace.

account ID, StorHouse

An identifier or name for a StorHouse account. This 1- to 12-character account ID can consist of letters, numbers, a dollar sign ($), or an underscore (_). You can delimit account IDs that do not follow these conventions.

active database

A database that has one or more sessions currently connected.

active load

A load that is in progress (currently running), successfully completed but not confirmed, or incomplete (failed) with errors.

active release The StorHouse/RM software release that is current when the StorHouse software is started.

active set *See* result set.

aggregate function An SQL operation that derives its result from a collection of values across one or more rows of a database table. For example, the aggregate function AVG computes the average of a set of numbers. *See also* function and scalar function.

alias An alternate name that refers to a table or a view in an SQL statement. An alias is valid only for the life of a query.

answer set *See* result set.

archive To create an archive copy of a primary file. The primary copy remains intact.

archive copy A duplicate copy of a primary file. You can recover a primary file from an archive copy.

argument A component of an SQL statement that completes or modifies the meaning of the keyword it follows. *See also* keyword.

arithmetic operator An operator that performs arithmetic operations on column values. StorHouse SQL supports five arithmetic operators: + (addition), - (subtraction), * (multiplication), / (division), and unary + and -. *See also* unary operator.

array A group of data items, or elements, assigned to one variable name. Arrays enable you to process a collection of elements with one SQL statement.

ASCII transfer type A mode of transfer in which data is passed as a stream of ASCII text records that end with ASCII carriage return/line feed (CRLF) pairs. You can specify the transfer type when loading data with the FileTek FTP Data Loader or unloading data with the FileTek FTP Data Unloader. *Contrast* BINARY transfer type. *See also* transfer type.

ATF *See* Access Time Factor.

atomic A state in which all operations in a transaction are either committed or rolled back. *See also* commit and roll back.

authorization manager The StorHouse/RM process that checks granted privileges on database components.

auto-join A type of join that combines a table with itself. Also called *self-join*.

automatic metadata recovery process The StorHouse/RM software that restores metadata when a StorHouse engine abnormally terminates during COMMIT or metadata DELETE, INSERT, or UPDATE processing as well as when there's a power loss or an operating system crash.

back end communications manager The StorHouse/RM server software that unmarshals SQL statements and marshals (arranges) result set data.

back up To create a backup copy of a primary file.

backup copy A duplicate copy of a primary file. You can use a backup copy to recover a primary file that has been corrupted or destroyed. Backup copies are stored on one or more backup volume sets.

basic predicate A type of predicate that compares two values with a relational operator. If any expression in the predicate evaluates to null, the result is unknown.

binary large object *See* BLOB.

BINARY transfer type A mode of transfer in which data is passed intact from a data file. You can specify the transfer type when loading data with the FileTek FTP Data Loader or unloading data with the FileTek FTP Data Unloader. *Contrast* ASCII transfer type. *See also* transfer type.

bind descriptor *See* input SQLDA.

blank *See* whitespace.

BLOB A variable-length array of bytes up to 2 gigabytes (GB) in size. Examples of BLOBs are audio and video clips, photos, and graphics. *See also* large object.

boolean operator An operator that relates one or more true/false values and produces a single true/false value. StorHouse supports three boolean operators: AND, NOT, and OR. Also called *logical operator*.

byte order The sequence in which a host stores binary values in memory. Some hosts store them with the most significant byte (msb) first and the least significant byte (lsb) last. Others store them in reverse order (lsb to msb). When loading data with the FileTek FTP Data Loader, you can specify the native values key to identify the byte order of your client host. The FileTek MVS Data Loader utility automatically uses the appropriate byte order for MVS hosts (msb to lsb). *See also* native values key.

C-style comment Comment text bracketed by the characters /* and */. You can use C-style comments in C language statements, static SQL, or SQL that is dynamically prepared. *Contrast* SQL-style comment.

Callable Interface The StorHouse application program interface (API) that enables user application programs to access StorHouse. Through the Callable Interface, applications can perform file functions, transfer data, use selected Command Language commands, and receive StorHouse responses.

Cartesian product A result set from a join operation that omits a restrictive clause. In the Cartesian product of a set of tables, each row is the row in the first table, concatenated with a row in the second table, concatenated with a row in the third table, and so on.

catalog *See* system table.

CCAdmin The StorHouse/Control Center GUI client module you use to monitor and manage StorHouse/Control Center servers. *See also* StorHouse/Control Center.

CCSID Coded Character Set Identifier. An identifier for describing input data and control statements used by the FileTek MVS Data Loader utility and the FileTek FTP Data Loader. StorHouse supports the following CCSIDs: 500 (EBCDIC code page), 819 (ISO 8859-1 code page), and 850 (PC code page). The default CCSID is 500 for MVS and 819 for FTP. *See also* character set.

cell The intersection of a column and a row in a table. Also called *data value*.

character large object *See* CLOB.

character literal A type of constant that specifies a character or character string consisting of letters, numbers, blanks, or special characters enclosed in quotes. *See also* literal.

character set A finite group of characters defined for a specific purpose. When you load data, you specify the character set of the input data. StorHouse supports EBCDIC (CCSID 500), ISO 8859-1 (CCSID 819), and PC (CCSID 850) character sets as well as the following ORACLE® character set names: WE8EBCDIC500, WE8PC850, and WE8ISO8859P1. *See also* CCSID.

checkpoint A restart point. When loading from MVS, the client data loader takes a checkpoint after writing a certain number of megabytes to the temporary VRAM™ file. When loading from MVS or with FTP, the server data loader takes a checkpoint for each data extent when it reaches the maximum size.

checkpoint dataset or file The dataset or file containing the checkpoint value that's used to recover from an aborted or failed run.

clause A component of an SQL statement that enables you to select and manipulate a subset of information in your database. Clauses begin with a keyword followed by arguments. *See also* keyword and argument.

client data loader, MVS The FileTek MVS Data Loader utility component that resides on your host. The client data loader prepares your data for loading and writes the data stream to a temporary file on StorHouse. *See also* FileTek MVS Data Loader utility. *Contrast* server data loader, MVS.

client FTP tool The FTP program on your host computer that sends server-level FTP commands and transfers files to the StorHouse FTP server during data loading and unloading.

CLOB A type of character data up to 2 GB in size. *See also* large object.

closed cursor A cursor that is no longer associated with the active set of rows returned by a query. A closed cursor, however, remains associated with the SELECT statement. *Contrast* open cursor.

coalesce operation *See* merge operation.

column An attribute or characteristic of a table. A data field in an input data record that you load into a column of a user table. *Contrast* field. *See also* data field.

column definition The column name, data type, and any default value or LOB storage options assigned to a column on a CREATE TABLE statement. *See also* table definition.

Command Language *See* StorHouse Command Language.

command privilege A privilege that determines which Command Language commands and parameters and Callable Interface functions a StorHouse account can use. For example, SQLEXECUTE is a command privilege that enables a StorHouse account to submit SQL statements. SQLCOMMAND is a command privilege that enables a StorHouse account to load data. *See also* SQLEXECUTE privilege and SQLCOMMAND privilege.

commit To accept and make permanent the changes that a transaction made to a database. *Contrast* roll back.

comparison operator A component of an SQL statement that enables you to perform comparisons on values. StorHouse SQL supports six comparison operators: = (equal), <> (not equal), < (less than), > (greater than), <= (less than or equal to), and >= (greater than or equal to).

complex predicate A type of predicate often containing boolean (logical) operators that relates one or more predicates and produces a single true, false, or unknown value. *See also* boolean operator.

compound index An index constructed on multiple columns of a table. Also called *composite index* or *concatenated index*.

concurrency The shared use of resources by more than one application process at the same time.

confirm To end a load. You must explicitly confirm a load when loading with FTP. The FileTek MVS Data Loader utility automatically confirms a successful load. Confirming a load updates the metadata and removes all restart information stored in checkpoint files.

connection string A fully qualified database name specified on a CONNECT statement to indicate a remote database. Also called *connect string*.

continuation field A string of characters, hexadecimal digits, or blanks you can identify to combine physical records into a logical record. When loading data, you can specify a continuation field with the CONTINUEIF clause of the LOAD DATA statement.

control file When loading data with the FileTek FTP Data Loader, the file on your client computer that contains control statements like LOAD DATA, LOAD INDEX, MERGE, or StorHouse SQL statements. A control file with a LOAD DATA statement can also contain data records.

control statements When loading data with the FileTek MVS Data Loader utility, the control statements (LOAD and SMDEF) you create to define load parameters and specify StorHouse information. For the FileTek FTP Data Loader, control statements are the LOAD DATA, LOAD INDEX, MERGE, and any SQL statements.

copy phase, load The transfer of a load data stream to a temporary VRAM file on StorHouse. *Compare* load phase.

copyin utility The metadata conversion program that updates metadata for a new StorHouse/RM release. *Compare* copyout utility.

copyout utility The metadata conversion program that makes a copy of the current metadata and performs the required conversions. *Compare* copyin utility.

creator The StorHouse account ID of the user who creates a database user component. The creator owns the component unless he or she assigns ownership to another account ID. *See also* owner.

current connection The database where you are currently executing SQL.

current row The most recently retrieved row of the active set and the row to which the cursor is pointing.

cursor A named control structure used by an application program to point to the current row in the active set and retrieve rows from that set.

data conversion The process that changes (if necessary) the data type from one representation to another, for instance, an input data type to a database data type when loading data, or a database data type to a host language data type when submitting a query with ESQL, or a database data type to an unloader data type when unloading data.

data definition language (DDL) A category of SQL. DDL statements create and drop database components, alter user tablespaces, and grant and revoke privileges.

data delimiter record A record in a load data stream that separates the data records from other record types. Data records begin immediately after the data delimiter record. *See also* data stream.

data dictionary *See* system table.

data extent A type of file extent that contains data records. *See also* extent.

data feeder A program on your host that supplies the input data for a load.

data field A column or field in an input data record. *See also* column and field.

data loader A program that loads data into StorHouse user tables and creates index entries for user table indexes. *See* FileTek MVS Data Loader utility or FileTek FTP Data Loader.

data manipulation language (DML) A category of SQL. DML statements query data and insert, update, and delete rows in system tables.

data source A server or a database in a federated system.

data stream Information packaged during a load. Also called *load stream*. A data stream consists of four record types: environment definition, LOAD DATA and other SQL statements, data delimiter record, and user data.

data type A classification that denotes the characteristics and length of data values in a column (for example, CHAR(5)). *See also* database data type, host language data type, loader data type, unloader data type, and native data type.

data value *See* cell.

database administrator, StorHouse The person responsible for security, operation, and tuning of a StorHouse database. *Compare* system administrator.

Database Administrator (DBA) privilege The highest database privilege that can be assigned to a StorHouse account. An account with DBA privilege can create database user components, access and maintain system tables, grant and revoke privileges, and select data from any table in a specific database.

database component privilege A StorHouse privilege that determines an account's access to specific database components. Database component privileges include ALL, DELETE, INDEX, INSERT, SELECT, and UPDATE.

database container directory The directory in the UNIX file system that contains all StorHouse database directories. *See also* system tablespace.

database data type A StorHouse data type used in a column definition in a CREATE TABLE statement.

database default user tablespace The user tablespace assigned to a user table when a StorHouse account does not specify a user tablespace on the CREATE TABLE statement and does not have an account default user tablespace. The database default user tablespace is the one assigned to PUBLIC in the SYSSMUSERS system table. *See also* user tablespace and account default user tablespace.

database directory *See* system tablespace.

database gateway A type of software that translates an application program's database requests from one relational database system protocol to another. Each gateway has a server component and a client component. StorHouse supports the ODBC gateway.

database name The identifier that names a StorHouse database. A database name must start with a letter, cannot exceed 32 characters, is case sensitive, and can consist of any combination of letters (a–z or A–Z), numbers (0–9), and underscore (_).

database privilege A privilege that controls the functions a StorHouse account can perform in a specific StorHouse database. Database privileges include DBA, RESOURCE, and SCAN. *See also* Database Administrator (DBA) privilege, RESOURCE privilege, and SCAN privilege.

database system components The structures that StorHouse/RM uses to manage a database. These components include system tables, system table indexes, system table logs, system table index logs, and the system tablespace. *Contrast* database user components.

database user components The structures that a StorHouse account works with to store, retrieve, and manage data. These components include user tables, columns, views, synonyms, indexes, and user tablespaces. *Contrast* database system components.

database system files The UNIX files that contain system tables, system table indexes, system table logs, and system table index logs. *Contrast* database user files.

database user files The StorHouse files that contain table data, index entries, and LOB data. *Contrast* database system files.

date literal A constant that specifies a month (MM), day (DD), and year (YYYY) string in a particular format.

DDL *See* data definition language.

decimal literal A constant that specifies a decimal number as a signed or unsigned number with a maximum of 31 digits that may include a decimal point.

declarative statement An SQL statement that does not generate any calls to StorHouse/RM routines.

Declare Section A required ESQL program component in which you declare host language and indicator variables.

default definition A default value to be loaded into a column if no value is supplied. You specify a default definition on the CREATE TABLE statement.

default subspace The lowest-numbered subspace that allows a component type. If you do not explicitly select subspaces or request rotation during a load, the FileTek data loader selects the default subspace.

default user tablespace *See* account default user tablespace or database default user tablespace.

deferred index An index created after a table is loaded. You use a FileTek data loader to create index entries for existing segments. *See also* index load operation.

delimited data Data records with delimiters. You can load delimited data. When unloading data, you can delimit the result data. *See also* enclosed data and terminated data.

delimited SQL identifier An alias, cursor, or database component name enclosed in double quotes (“) because it contains a space or special character, is case sensitive, starts with a

character other than a letter, or is an SQL reserved word. *See also* SQL identifier.

delimiter A marker for separating data fields in an input record or a result record. A termination delimiter follows a data field, and an enclosure delimiter surrounds a data field.

detail data The most granular level of information that an enterprise can collect at a single point of service. Examples of detail data are call detail records, clickstream data, e-mail message, and point-of-sale information.

DF extent A type of file extent that contains information necessary to retrieve data. For segment files, there is one DF extent for each data extent. *See also* extent.

discard file A StorHouse VRAM file that contains records discarded during a load operation. *See also* discarded record.

discarded record A logical record that did not meet any condition (WHEN clause) specified for a data load operation. StorHouse/RM collects discarded records in a discard file on StorHouse. *See also* discard file.

DML *See* data manipulation language.

durable A state in which all operations in a transaction are permanent (in other words, committed). *See also* commit.

dynamic SQL statement An SQL statement that is prepared and executed at runtime. The SQL statement can change several times during the execution of the application program. *Contrast* static SQL statement.

E-notation A floating-point literal expressed as two numbers separated by E, for example, 10E132.

EDC *See* error detection code.

embedded SQL statement *See* static SQL statement.

enclosed data Data fields preceded and followed by an enclosure delimiter. *See also* delimiter.

engine *See* StorHouse engine.

environment definition The load data stream record that contains nvk (affects interpretation of binary multibyte numerics), ccsid (character data), record type, and db_ref (affects interpretation of loader control information). The StorHouse FTP server and the FileTek MVS Data Loader utility generate this record based on parameters you provide for a load as well as internally supplied data.

environment variable A UNIX parameter that defines an aspect of your working environment, such as your home directory or frequently used directory path.

equi-join A type of join that uses predicates specifying equalities to join a column from one table with a column from another table.

error detection code (EDC) A parameter that determines whether to use the StorHouse error detection feature. For StorHouse/RM, you can set an EDC parameter in a subspace. StorHouse generates EDCs during data loads and uses EDCs to detect errors during segment file movement in StorHouse.

ESQL Embedded Structured Query Language. An interface that enables you to embed SQL in a C or C++ program.

ESQL construct An SQL statement prefixed by EXEC SQL and terminated by a semi-colon (;). The prefix, statement, and delimiter constitute the construct.

ESQL precompiler The precompiler used to build database applications in the StorHouse SQL development environment.

exclusive lock A lock that reserves a table for updating only. Also called *write lock*. This lock prohibits a table from being shared. One engine can have an exclusive lock on a table. All other lock requests (shared and exclusive) for the table are queued. *Contrast* shared lock.

executable statement

An SQL statement that generates StorHouse calls and return codes. Executable statements execute instructions on a specified database at runtime.

EXECUTE STH_LOAD command

The StorHouse Command Language command that starts the loading process in StorHouse. The FileTek MVS Data Loader utility and the FileTek FTP Data Loader generate and submit this command for you.

execution manager

The StorHouse/RM process that runs the constructed execution tree built by the optimizer.

execution tree

An internal representation of a query in a form that an engine can process. The StorHouse/RM explain facility enables you to create a representation of the execution tree in the form of relational tables. You can query these tables to view the execution strategies selected by the optimizer. *See also* explain facility and node.

explain facility

The StorHouse/RM feature that enables you to examine the strategy, or execution plan, implemented by the StorHouse/RM optimizer for a given query. The explain facility displays the execution plan in a set of explain tables. *See also* explain tables.

explain tables

A set of relational tables, located in a system tablespace, containing the execution plan for a query. You use StorHouse SQL statements—CREATE EXPLAIN TABLES, EXPLAIN PLAN, DROP EXPLAIN TABLES—to manage the explain tables. *See also* explain facility.

expression

An operand or a collection of operands and operators that yields a single value.

extension

An extra option or feature supported by StorHouse SQL in addition to those features specified in the ANSI SQL standard. *See also* Structured Query Language (SQL).

extent

Contiguous data or control information that StorHouse treats as a unit. StorHouse files consist of extents. *See also* data extent, DF extent, and map extent.

extraction A type of query that returns all rows for one or more columns in a user table. This type of query results in a full table scan or a full segment select. *See also* full table scan and full segment select and *contrast* select.

extractor A StorHouse feature that provides fast-path processing for some queries that result in full table scans and full segment selects.

federated system A database management system (DBMS) that supports applications and users who submit SQL statements that reference two or more DBMSs or databases in a single statement (for example, a join between tables in different databases). The IBM DB2 UDB product with StorHouse as a data source is an example of a federated system.

federator The software component in a federated system that coordinates SQL processing. For instance, a federator accepts and parses a query from a client, breaks the query into smaller queries, submits the queries to the appropriate data sources, receives the results from the data sources, assembles the results into a single result set, and returns the result set to the client. DB2 UDB is both a federator and a database system.

field A named portion of a record to be used in a condition, for instance, to indicate which records to load. Fields are not loaded into user tables. *Contrast* column. *See also* data field.

field terminator A delimiter that follows a data field in an input record or a result record.

file access group A set of files that are grouped for access control.

file extent *See* extent.

file migration The movement of StorHouse files from the performance buffer to their resident file sets. StorHouse automatically migrates files off the performance buffer based on various factors. When creating or altering a user tablespace, you specify file migration attributes in subspaces. You can also use the StorHouse defaults for these attributes.

file reference variable

A host variable used for transferring a LOB value from StorHouse to a client file.

file set (FSET)

A collection of files within a volume set. Table data files, index files, and LOB subsegment files are stored in file sets. When creating a user tablespace, you specify a file set for each subspace. You can change file sets by altering a user tablespace.

file set name

The name of a file set. A file set name can contain up to 8 characters consisting of letters (a–z or A–Z), numbers (0–9), underscore (_), and dollar sign ($). A file set name must be unique within the volume set. *See also* volume set name.

file system, StorHouse

The StorHouse/SM software that manages StorHouse files.

file system, UNIX

The UNIX operating system software that manages StorHouse/RM metadata and range indexes.

FileTek FTP Data Loader

The FileTek software that bulk loads data from a UNIX, VAX, or other FTP-enabled host computer into user tables on StorHouse.

FileTek FTP Data Unloader

The FileTek software that unloads data from StorHouse to a sequential file on your client, or to a VRAM file on StorHouse, or to another program. You can also unload LOB data to separate LOB files on your client computer or on a remote host. You use your standard client FTP software to unload data.

FileTek MVS Data Loader utility

The FileTek software that bulk loads data from an MVS host computer into user tables on StorHouse.

fixed-length record

A data record that has the same length as other data records in a data file. The last byte in the data file must exactly finish a fixed-length record. *Contrast* variable-length record and text record.

floating-point literal

A constant that specifies a floating-point number as two numbers separated by an E (for example, 15E1 or 2.3E5). Both numbers can include a sign, but only the first number can include a decimal point. The maximum number of

digits allowed in the first number is 17. The maximum number of digits allowed in the second number is 2.

format string The mask values that indicate the format StorHouse/RM should use to return date and time values.

front end communications manager The StorHouse/RM client software that arranges (marshals) prepared SQL in communication packets and unmarshals result set data.

FTP File Transfer Protocol. An established standard for transferring files between two computers connected by a network. FileTek provides the FileTek FTP Data Loader for bulk data loading through FTP.

FTP command An instruction you provide to run the FileTek FTP Data Loader and Unloader. You supply user-level FTP commands to your client FTP tool. Your client FTP tool sends server-level FTP commands to the StorHouse FTP server. For instance, *put* is a user-level FTP command and *STOR* is the corresponding server-level FTP command.

FTP session When loading or unloading, the period of time or interaction that begins when you log into the StorHouse FTP server and ends when you log off. During an FTP session, you can run one or multiple loads and unloads.

full segment select A type of query that returns one or more entire segments with the use of a range index. A full segment select may qualify for extractor processing.

full table scan To read all rows in a table without the use of an index. A query that results in a full table scan may qualify for extractor processing. A StorHouse account requires the SCAN database privilege to perform a full table scan.

function A named operation in an SQL statement, followed by one or more expressions. StorHouse supports two types of functions: aggregate and scalar. *See also* expression, aggregate function, and scalar function.

gateway *See* database gateway.

grantee The account ID that is granted a database or database component privilege. *Contrast* grantor.

grantor The account ID that grants a database or database component privilege to another account ID. *Contrast* grantee.

group *See* file access group.

hash index An index that uses a proprietary FileTek algorithm to locate individual table rows based on individual index values. StorHouse typically uses hash indexes to search for a single value or for a small range of specific values.

hash index file A STORHOUSE-type file that contains entries for a hash index. *See also* STORHOUSE file type.

hexadecimal literal A string of one or more hexits, enclosed in quotation marks and prefaced by an uppercase or lowercase X. Hexadecimal literals represent binary data. You can use hexadecimal literals (for example, X '01020ABC') in StorHouse SQL WHERE clauses and SELECT lists. *See also* hexit.

hexit A member of the character set A-F, 0-9, and a-f. This character set defines the valid character set for hexadecimal numbers.

host data file *See* data file or input dataset.

host language data type A data type that is native to the host language. For example, long is a C language data type.

host language variable An application variable that can be referenced by host language statements and embedded SQL statements. *See also* input host variable and output host variable.

hybrid IN join method A type of join method that the optimizer may use when a query is an equi-join and the inner table has a value index on the join column or for inner-join and outer-join operations. *Compare* nested loop join method. *See also* join method.

identifier *See* SQL identifier.

in-line LOB A LOB data value stored in a row with the other table data. You can store a LOB data value in-line when the value and the row do not exceed 32 KB. *Contrast* out-of-line LOB.

inconsistent database The state of a database when a single atomic operation that updates one or more system tables fails before completing and committing all updates.

index A database component used to locate data stored in tables. StorHouse supports three types of indexes for user tables: hash, range, and value. *See also* hash index, range index, and value index.

index file A StorHouse file that contains index entries for user tables. There's one index file for each hash index and each value index for each segment in a user table. *See also* hash index file, value index file, and STORHOUSE file type.

index ID The numeric identifier assigned to each index in a database. Index IDs are unique within a database.

index load operation A type of load that creates index entries of deferred indexes for existing segments. You run an index load with a FileTek data loader. *See also* deferred index.

indicator variable An optional variable used to detect null or truncated values in an application program.

inner-join A type of join that combines matching rows from specified tables. The unmatched rows are omitted from the result table. *Contrast* outer-join.

input data file A file that contains the data records you're loading into a StorHouse user table. This file can reside on your client computer or on StorHouse. LOB values can also reside in separate LOB data files. Also called *input dataset*. *Contrast* result file. *See also* LOB data file.

input data record A record in an input dataset or data file. An input data record typically corresponds to a row in the user table. *Contrast* result data record.

input dataset The file that contains the input data records you are loading into StorHouse user tables. This file can reside on your host or on StorHouse. Also called *data file* or *input data file*.

input host variable A host variable used to pass data to your program. Input host variables are generally specified in the WHERE clause of a SELECT statement. *Contrast* output host variable.

input SQLDA A storage area that holds information about input host variables in a dynamic SQL statement. Also known as *bind descriptor*. *Contrast* output SQLDA.

integer literal A constant that specifies a binary integer as a signed or unsigned number with a maximum of 10 significant digits and no decimal point. The range for integer literals is +2147483647 to -2147483648.

interior node A node, in an execution tree, that has both a parent and one or more children. *See also* node, leaf node, root node, and execution tree.

invalidated segment A segment that has been replaced or made obsolete through a FileTek data loader merge, or coalesce, operation.

ISQL Interactive SQL. The StorHouse/Admin module of StorHouse/Control Center provides an ISQL tool for submitting SQL statements to StorHouse.

join A database operation (SELECT) that combines rows from one or more tables or views. StorHouse supports outer-joins, inner-joins, equi-joins and self-joins. *See also* outer-join, inner-join, equi-join, and self-join.

join method The series of steps that the optimizer performs to join tables. StorHouse/RM supports two join methods: nested loop and hybrid IN. *See also* nested loop join method and hybrid IN join method.

journal chain Multiple journal files—current, cycled, and archived—that are linked together and replayed when a database is restored. The journal replay utility can replay an entire journal chain or a partial chain. *See also* redo journal.

journal file *See* redo journal.

keyword An SQL statement component with a predefined meaning that is reserved for special tasks (for example, WHERE). Arguments complete or modify the meaning of the keyword. *See also* argument.

large object (LOB) A sequence of bytes with a length up to 2 GB. A LOB may be a binary large object (BLOB) or a character large object (CLOB). *See also* BLOB and CLOB.

leading blanks One or more blank characters at the beginning of a data field. A FileTek data loader does not trim blanks that are part of a data field that's enclosed by enclosure delimiters. It may trim leading blanks from a character-type data field when optional enclosure delimiters are not present.

leaf node A node, in an execution tree, that represents operations on base tables. A leaf node has a parent, but no children. *See also* node, interior node, root node, and execution tree.

left outer-join A type of join that returns all of the matched rows of the tables, and for unmatched rows, returns all rows from the outer or left table with null values for the inner or right table. *See also* outer-join. *Contrast* inner-join.

library device (LD) A StorHouse hardware component that contains multiple volumes and uses robotics to transfer media between drives, slots, and an exchange station. A single StorHouse system can support multiple library devices.

literal A string or numeric constant that denotes a specific value. Literals can be classified as string, numeric (exact and approximate), and binary. *See also* character literal, date literal, decimal literal, floating-point literal, hexadecimal literal, integer literal, time literal, and timestamp literal.

load ID An identifier that uniquely names each load. A load ID is case sensitive and can consist of up to 40 alphanumeric characters.

load phase The portion of a load when the server data loader builds any indexes and writes segments on StorHouse. *Compare* copy phase, load.

LOAD DATA statement An SQL-like statement that describes the data to be loaded into one or more user tables in a StorHouse database. *Contrast* UNLOAD statement.

load stream *See* data stream.

loader data type A data type that describes the data fields (columns) being loaded into a database. You provide this data type in a LOAD DATA statement.

LOB *See* large object.

LOB data file A file on a client computer or a remote system that contains one LOB data value to be loaded into a user table on StorHouse.

LOB locator *See* locator variable.

LOB record A BLOB or CLOB data field that consists of one or more physical records in the input data file. A LOB record starts with a 64-bit length field followed by the data. LOB records are automatically considered part of the logical record, that is, no CONCATENATE or CONTINUEIF clause is needed and, if present, does not apply to these records.

LOB result file A file on a client computer or a remote system that contains one LOB data value unloaded from a StorHouse user table.

LOB subsegment file A StorHouse file that contains out-of-line LOB values. Multiple LOB columns in a user table can share a LOB subsegment file. *See also* out-of-line LOB and STORHOUSE file type.

LOB subsegment ID A numeric identifier for each LOB subsegment file in a segment.

local database A database that resides on your host system. *Contrast* remote database.

locator variable A host variable that represents a single LOB value and enables an application to manipulate that value on StorHouse with various SQL functions.

logical operator An operator that relates one or more true/false values and produces a single true/false value. StorHouse supports three logical operators: AND, NOT, and OR. Also called *boolean operator*.

logical record in input data A record that is assembled from one or more physical records. Logical records can contain both fields and columns. *Contrast* physical record in input data.

map extent A type of file extent that contains the high-level index that StorHouse reads first when doing index lookups. Hash index files and value index files consist of data, DF, and map extents. *See also* extent.

merge operation A type of load that consolidates existing segments (based on selection criteria) into one segment or a smaller number of segments. Also called *coalesce operation*. You use a FileTek data loader to merge segments.

metadata The data that StorHouse/RM uses to manage a database. *See also* system table and system table index.

metadata backup file A file created by the metadata backup utility. A metadata backup file contains a copy of system tables, system table indexes, system table logs, and system table index logs. If necessary, you can use metadata backup files to restore metadata.

metadata backup utility The software that backs up the system components in a database directory. This utility creates metadata backup files.

metadata conversion utility The software that copies and reformats existing metadata so that a new StorHouse/RM software release can use it. *See also* copyout utility and copyin utility.

metadata entry A row in a system table.

metadata recovery *See* automatic metadata recovery process or metadata restore utility.

metadata restore utility

The software that recovers one or more database directories with metadata backup files. This utility locks the metadata while copying the metadata backup files to the database directories.

middleware

A class of software used to provide database gateways with network communications support.

.netrc file

A file (on the home directory of your client computer) that contains macros for automating FTP command entry.

native data type

A binary data type. StorHouse supports the following native data types: BINARY, BYTE, DECIMAL, DOUBLE, FLOAT, INTEGER, NUMERIC, RAW, SMALLINT, VARBINARY, VARBYTE, and VARCHAR. When loading data with the FileTek FTP Data Loader, different hosts interpret binary data differently, so you may have to identify a native values key on the FTP put command. *See also* native values key.

native values key

A value that identifies the client hardware type used to interpret byte order, floating point representation, and INT length of binary data. You can specify a native values key when loading data with the FileTek FTP Data Loader.

nested loop join method

A join method that the optimizer may use when the number of qualifying rows in the outer table is small, when the predicate is not an equals-type, or when the join column of the inner table does not have an index. *Compare* hybrid IN join method. *See also* join method.

nickname

In a federated system, the representation in a DB2 catalog for a remote table or view controlled by a particular server. You can create nicknames for StorHouse tables and views through the DB2 CREATE NICKNAME statement.

node

A discreet step in the execution of the query. You can display the nodes in an execution plan by using the StorHouse/RM explain facility. A node may be a root node, an interior node, or a leaf node. *See also* explain facility, root node, interior node, and leaf node.

NOT NULL A column constraint that indicates you must provide a value for the column during a load operation for a user table or during an insert or update operation for a system table.

null value A value that indicates unknown, not applicable, or missing information.

object identifier (OID) The entry in a table row that identifies an out-of-line LOB. The OID contains the LOB length (in bytes), the LOB subsegment ID, and the starting frame number and offset within the LOB subsegment file. StorHouse/RM uses OIDs to access LOB values in LOB subsegment files. *See also* out-of-line LOB.

ODBC Microsoft Open Database Connectivity.

OID *See* object identifier.

open cursor A cursor that is associated with the result set returned by a query. *Contrast* closed cursor.

operator (SQL) A component of an SQL statement that enables you to manipulate table data by performing such tasks as comparisons and arithmetic operations on column values. There are four main types of operators: arithmetic, comparison, logical, and set. *See also* arithmetic operator, comparison operator, logical operator, and set operator.

optimizer A component of the StorHouse engine that receives analyzed SQL statements from the parser and develops the most efficient plans to execute the statements. *See also* parser.

out-of-line LOB A large binary or character object that's stored in a LOB subsegment file. *Contrast* in-line LOB.

outer-join A type of join that returns all of the matched rows of the tables, and for unmatched rows, returns all rows from one table with null values for the other table. StorHouse supports left outer-joins. *See also* left outer-join. *Contrast* inner-join.

output host variable A host variable that passes data and status information to a program. You specify output host variables on the INTO clause of a SELECT or FETCH statement and on the VALUES INTO statement. *Contrast* input host variable.

output SQLDA A storage area that holds information about the output host variables in a dynamic SQL statement. Also known as *select descriptor*. *Contrast* input SQLDA.

owner A StorHouse account ID that creates or is assigned ownership of a database component. *See also* creator.

package A set of stored SQL statements that have been bound statically and are available for processing.

page manager The StorHouse process that translates retrieval requests into calls to StorHouse/SM data retrieval services.

parameter *See* system parameter.

parser A component of the StorHouse engine that analyzes an SQL request for syntax and semantics, and then breaks it into basic elements for the optimizer. *See also* optimizer.

passthru In a federated system, a mechanism by which DB2 allows a client to issue queries and other SQL statements directly to a data source. With the StorHouse/UDB Link, StorHouse is the data source

performance buffer The portion of the StorHouse magnetic disk layer that contains performance copies of StorHouse files waiting to be backed up or migrated to their resident file sets.

performance copy The active file extents stored on high-speed magnetic disk for optimal access.

physical record in input data One input data record (or line) in an input dataset or data file. *Contrast* logical record in input data.

precision The total number of digits in a number defined as a DECIMAL or NUMERIC data type. *Contrast* scale.

predicate An element of a search condition that implies or expresses a comparison operation. Predicates reduce the number of rows returned by a query. StorHouse supports three types of predicates: basic, complex, and quantified. *See also* basic predicate, complex predicate, and quantified predicate.

prepared statement An SQL statement that has been parsed for syntax errors and assigned an identifier. You can execute a prepared statement as often as necessary within the same transaction.

primary copy A file copy available for normal access.

privilege The authority to perform a certain type of function or to bypass a certain type of security in the system. Accounts can have access, command, database, and database component privileges. *See also* access privilege, command privilege, database privilege, and database component privilege.

project A table operation that selects all rows for one or more columns in a table. *Contrast* select.

PUBLIC All accounts that can access StorHouse. Granting database component privileges to PUBLIC is the same as granting them to all accounts that can access StorHouse.

pushdown In a federated system, the act of moving processing into a particular data source.

quantified predicate A type of predicate that compares a value to a collection of values preceded by the keyword ALL, ANY, or SOME.

RAID (Redundant Array of Independent Disks) An array of StorHouse magnetic disks where part of the storage capacity is used to store redundant information about data on the other disks.

range index A type of StorHouse index that contains the highest and lowest values contained in each segment. Range indexes are stored in a set of system tables.

read lock *See* shared locked.

record The storage representation of a row or other data. Also called *row*.

record terminator A delimiter that marks the end of an input record or a result record.

recovery *See* automatic metadata recovery process or metadata restore utility.

recovery log file A file that lists any system table logs and system table index logs containing undo records. A recovery process checks this log during metadata recovery.

redo journal A file that contains data used to re-create metadata. The following committed transactions are captured in a redo journal: INSERT ROW, DELETE ROW, UPDATE ROW, DROP INDEX, DROP TABLE, CREATE INDEX, and CREATE TABLE. You use redo journaling utilities to manage the redo journal. Also called *journal file*. *See also* redo journaling utilities and journal chain.

redo journaling utilities A set of database recovery utilities for capturing, storing, and restoring transactions that affect metadata. The utilities enable you to cycle a redo journal (close the current one and start a new one), archive a redo journal to StorHouse, audit a redo journal, purge a redo journal from disk, and replay a redo journal to restore changes. *See also* redo journal.

remote database A database that resides on a system other than your host system. For example, a StorHouse database is considered a remote database in a DB2 application running on an MVS host. *Contrast* local database.

replaced segment A segment that has been invalidated. Users cannot access files in replaced segments.

replay checkpoint A point (or journal file) in a journal chain where the journal replay utility stops restoring transactions. With replay checkpointing, you perform a

partial replay, restoring only those transactions up to a checkpoint. *See also* journal chain and redo journaling utilities.

reserved word *See* SQL reserved word.

RESOURCE privilege A StorHouse database privilege that enables an account to create tables, synonyms, views, and indexes and to select information from those tables, synonyms, and views.

restrictive clause The clause that limits the number of rows returned by a query. For example, WHERE and HAVING are restrictive clauses on a SELECT statement.

result data record A record in a result file. *Contrast* input data record.

result file A file on a client computer that contains data unloaded from a user table on StorHouse. *Contrast* input data file.

result set The set of rows returned from a SELECT statement. Also called *answer set* and *result table*.

result table *See* result set.

roll back To reverse changes made by a transaction to a database. *Contrast* commit.

root node The entry point into an execution tree. *See also* node, interior node, leaf node, and execution tree.

row A collection of columns. Also called *record*.

scalar function An SQL operation that produces a single value from another value. For example, the scalar function ABS computes the absolute value of a specified expression. *See also* function and aggregate function.

scale The number of digits to the right of the decimal point in a number defined as a DECIMAL or NUMERIC data type. *Contrast* precision.

scan *See* full table scan.

SCAN privilege A StorHouse database privilege that enables an account to read all rows in any user table (in other words, perform a full table scan) on which it has SELECT privilege. An account with SQLADMIN or DBA privilege can grant SCAN privilege to other accounts.

segment A set of StorHouse database user files. Each user table consists of one or more segments. Each segment consists of a table data file, one index file for each hash index and each value index, and one or more LOB subsegment files. Segments consist of extents. *See also* extent, table data file, index file, and LOB subsegment file.

segment delete utility The software that identifies invalidated segments, removes the segment files (table data file, index files, LOB subsegments), and removes the associated entries in system tables. Use this facility to mass-remove invalidated segments and their associated objects from a StorHouse database.

segment file A StorHouse file that contains table data, hash index entries, value index entries, or LOB data. Each segment consists of a set of segment files.

segment ID A numeric identifier for each segment in a user table.

segment tag A name assigned to a segment with the SEGMENT clause on a LOAD DATA or MERGE statement. The default segment tag is the load ID.

select A table operation that selects all columns from one or more rows in a table. *Contrast* project. Also known as a *selection*, which contrasts with *extraction*.

select descriptor *See* output SQLDA.

select list Column names and other expressions specified in the SELECT clause.

self-join A type of join that joins a table with itself. Also called *auto-join*.

serializable The ANSI/ISO transaction isolation level that guarantees the highest read consistency and data integrity in a database. StorHouse supports this isolation level.

server data loader, MVS The FileTek MVS Data Loader utility software component that loads data into StorHouse user tables. The server data loader runs on StorHouse. *See also* FileTek MVS Data Loader utility. *Contrast* client data loader, MVS.

set operator An operator you use to combine separate SQL queries in different ways. StorHouse supports the UNION and UNION ALL set operators.

shared lock A lock that reserves a table for reading only. Also called *read lock*. Multiple engines can have a shared lock on the same table. This lock prevents a table from being dropped. *Contrast* exclusive lock.

short record An input data record that is missing a data field described in a field specification.

shortword A 2-byte integer (SMALLINT) preceding each variable-length data record. This shortword contains the length of each data record. The shortword is not included in the length of the data record.

simple query A type of query processed by the extractor. A simple query must conform to a specific set of rules and results in a full table scan.

single-column index An index created on one column in a table.

SNA interface The client and server hardware and software necessary to communicate using LU6.2 with the DRDA gateway.

soft drop A feature that enables you to recover from an errant table drop. When you drop a user table, you can later undrop or purge it.

space *See* whitespace.

special register A storage area for specific information that can be referenced in SQL statements. StorHouse supports three special registers: USER, SYSDATE, and SYSTIME.

SQL-style comment Comment text that starts with two hyphens (--) and terminates by the end of the line. You use SQL-style comments in static SQL statements. *Contrast* C-style comment.

SQL identifier A basic syntactical unit that provides a name for a database component, alias, or cursor. An SQL identifier must start with a letter and can be followed by a–z, A–Z, 0–9, or an underscore (_). Identifiers in StorHouse SQL can have a maximum length of 32 characters. *See also* delimited SQL identifier.

SQL reserved word A keyword that has been predefined to SQL to initiate specific processing tasks within the SQL environment. You should not use reserved words as database component names, but if you do, you must delimit the names.

SQL statement A StorHouse SQL instruction used to manage and access a StorHouse database.

SQL statement manager The StorHouse/RM software that tracks SQL statements through all processing steps.

SQL status code A numeric code that StorHouse/RM generates in response to StorHouse SQL statements.

SQLADMIN privilege A StorHouse access privilege that gives an account DBA privilege in all StorHouse databases.

SQLCA SQL Communications Area. A storage area that holds information about the execution of an application's most recently executed SQL statement.

SQLCOMMAND privilege A StorHouse command privilege that enables an account to load data (invoke a FileTek data loader to submit the StorHouse Command Language EXECUTE STH_LOAD command).

SQLDA SQL Descriptor Area. A storage area that holds information about the input or output host variables in a dynamic SQL statement. *See also* input SQLDA and output SQLDA.

SQLEXECUTE privilege
A StorHouse command privilege that enables an account to submit SQL statements.

static SQL statement
An SQL statement that is embedded in an application program, prefixed by the keywords EXEC SQL, and terminated by a semi-colon. Once prepared, a static SQL statement does not change. *Contrast* dynamic SQL statement.

status code
See SQL status code.

STH
The default StorHouse file access group for segment files.

sthname
The name of the StorHouse system to which you connect when you load or unload StorHouse data.

storage level
A type of storage managed by StorHouse. The three storage levels are fixed, library, and shelf.

StorHouse®
The FileTek enterprise-wide solution for managing the capture, storage, movement, and access of gigabytes to petabytes of relational and non-relational detail data. StorHouse technology combines industry-leading, scalable storage devices and Open System processors with the FileTek specialized storage management and relational database management system (RDBMS) software components. *See also* StorHouse/RM and StorHouse/SM.

StorHouse/Admin
The StorHouse/Control Center GUI client module you can use to perform StorHouse system and database administration tasks such as creating databases and database components, creating volume sets and file sets, setting system parameters, creating accounts, assigning privileges, and so on. *See also* StorHouse/Control Center.

StorHouse/Control Center
The FileTek Windows®-based network computing system for providing administrative control of the StorHouse family of products. The system consists of one or more StorHouse/Control Center servers communicating with StorHouse/Control Center client modules over an IP network. The available client modules are CCAdmin, StorHouse/Admin, and StorHouse/Performance Monitor. *See also* CCAdmin, StorHouse/Admin, and StorHouse/Performance Monitor.

StorHouse/ Performance Monitor

The StorHouse/Control Center GUI client module you can use to monitor and analyze the performance and activities of StorHouse. *See also* StorHouse/Control Center.

StorHouse/RM

The FileTek RDBMS component that works in conjunction with StorHouse/SM to specifically administer the storage, access, and movement of relational data. StorHouse/RM provides row-level SQL access to high volumes of detail data on any layer in the StorHouse storage hierarchy, including tape. SQL access is available from different platforms through a variety of industry-standard protocols. *See also* StorHouse and StorHouse/SM.

StorHouse/RM API

The client software that passes prepared SQL statements to the front end communications manager.

StorHouse/SM

The FileTek storage management component that controls a hierarchy of storage devices comprised of cache, redundant array of independent disks (RAID), Advanced Technology Attached (ATA) disks, massive array of idle disks (MAID), erasable and write-once-read-many (WORM) optical disk jukeboxes, and erasable and WORM automated tape libraries. StorHouse/SM is also responsible for automating critical system management tasks, like data migration, backup, and recovery. *See also* StorHouse and StorHouse/RM.

StorHouse/UDB Link

The FileTek wrapper software that implements the connection between the DB2 UDB and StorHouse databases. This software gives DB2 users near-transparent access to data on StorHouse.

StorHouse API

The StorHouse software that enables user applications to access StorHouse, perform file functions, and transfer data.

StorHouse Command Language

The set of commands that make up the StorHouse Interactive Interface. You issue Command Language commands to communicate with, monitor, and control StorHouse.

StorHouse communications manager

The StorHouse software that manages socket-level communications between StorHouse client and server systems, including ESCON channel support.

StorHouse engine The main StorHouse/RM process. A StorHouse engine parses, optimizes, and executes SQL statements. During a load, a StorHouse engine executes non-LOAD statements in a control file, obtains necessary metadata for a load, verifies StorHouse privileges for loading, and updates metadata at the end of a load.

STORHOUSE file type A StorHouse file type created and used exclusively for StorHouse database data. Segment files—table data file, hash index file, value index file, and LOB subsegment file—are STORHOUSE-type files.

StorHouse FTP server The FileTek software component that acts as a valid FTP server. It accepts FTP commands from your client FTP tool, generates and transfers data streams to StorHouse, and replies to your FTP commands.

Structured Query Language (SQL) A standardized language for defining and manipulating data in a relational database. StorHouse SQL is compatible with the ANSI SQL-99 standard and includes extensions to support additional capabilities.

subsegment ID *See* LOB subsegment ID.

subquery A SELECT statement within the WHERE or HAVING clause of another SQL statement.

subspace A part of a user tablespace that defines storage parameters for a specific component type—table data, hash index, value index, LOB data—or for all component types. *See also* default subspace.

subspace number A number you assign to each subspace when you create a user tablespace. Subspace numbers range from 0 to 2,147,483,647.

symbolic name An alphabetic code in a StorHouse message. Symbolic names are listed and explained in the StorHouse *Messages and Codes Manual*.

synonym An alternate name you can assign to a user table, view, or synonym for easy identification.

SYSADM account A StorHouse account with the privileges to administer all StorHouse databases. SYSADM has all StorHouse/SM command privileges (ALLPRIVILEGE) and all StorHouse/RM database privileges—DBA, RESOURCE, and SCAN—in all databases.

syscreate utility The program that creates StorHouse databases.

system administrator, StorHouse The person responsible for StorHouse security, operation, and tuning. *Contrast* database administrator.

system components *See* database system components.

system files *See* database system files.

system parameter The information that influences how StorHouse manages its resources. Your system administrator uses system parameters to set specified system-wide defaults.

system table A database table that contains definition and tracking information about the database. StorHouse/RM automatically creates system tables for each new database and updates system tables when you create and drop database user components and when you load data. Range indexes are stored in system tables. *See also* metadata.

system table index A type of index that StorHouse/RM automatically creates for some system tables when a database is created.

system table index log A system component that is used to recover changes to system table indexes. Before StorHouse/RM updates a system table index, it first copies a before image (or undo record) to the system table index log.

system table log A system component that is used to recover changes to system tables. Before StorHouse/RM updates a system table, it first copies a before image (or undo record) to the system table log.

system tablespace Logically, a database component that stores metadata in a database. Physically, the database directory under which all database system component files reside. Each StorHouse database has a system tablespace.

table ID A numeric identifier for each table in a database. StorHouse/RM assigns a table ID to a user table when you issue a CREATE TABLE statement. StorHouse/RM assigns table IDs to system tables when a database is created.

table data file A StorHouse file that contains user table data. Each segment contains one table data file.

table definition The table name, column definitions, and user tablespace assignment supplied on a CREATE TABLE statement. *See also* column definition.

tablespace *See* system tablespace, user tablespace, or temporary tablespace.

temporary tablespace The area on RAID where StorHouse/RM temporarily stores result sets and builds and sorts indexes.

terminated data Data fields followed by a termination delimiter. *See also* delimiter.

text record A record that contains text only and ends with the appropriate end-of-line character, such as a UNIX newline (LF) or an ASCII carriage return/line feed (CRLF) pair. *Contrast* fixed-length record and variable-length record.

time literal A constant that specifies hours (HH), minutes (MM), seconds (SS), and optionally milliseconds (CCC).

timestamp literal A constant that specifies a date and time combination.

transaction One or more SQL statements that are treated as a single unit of work. If successful and if all its changes are accepted, a transaction can be committed (made permanent). If unsuccessful, a transaction can be rolled back (canceled). *See also* atomic, commit, durable, and roll back.

trailing blanks One or more blank characters at the end of a data field. A FileTek data loader does not trim trailing blanks from character-type data fields, VARCHAR data fields, or enclosed data fields (within enclosure delimiters).

transfer type A mode of transfer between your client FTP tool and the StorHouse FTP server. A transfer type defines how data and end-of-line characters are interpreted. Two transfer types are ASCII and BINARY. *See also* ASCII transfer type and BINARY transfer type.

tuple A row in a table.

tuple ID A row ID.

unary operator An arithmetic operator that reverses the sign of a positive numeric value operand (-, unary minus) or does not change the operand (+, unary plus). *See also* arithmetic operator.

undo record A "before image" of a system table or system table index, copied to a system table log or system table index log. The automatic metadata recovery process applies undo records during metadata recovery to roll back incomplete transactions.

unload To select data from a StorHouse user table(s) and copy it to a sequential file on a host or pipe it to a program. You can also unload LOB data to separate LOB files on a client or remote system.

UNLOAD statement The SQL-like control statement that specifies the character set of result data, defines the format of result data, specifies a character to append to the end of result records, describes each data field in result records, and contains the query that selects the StorHouse data to unload. *Contrast* LOAD DATA statement.

unloader data type A data type that describes the result data fields being unloaded from a database. You provide this data type in the UNLOAD statement.

unprotected database The state of a database when access and further updates to inconsistent metadata is allowed.

user components *See* database user components.

user mapping In a federated system, the correspondence between a user's DB2 identity, or authorization ID, and the identify used when communicating with a data source, for instance, a StorHouse account ID.

user table A database component that contains columns and rows of user data.

user table name An SQL identifier that you provide on the CREATE TABLE statement to name a user table.

user tablespace A logical database component that defines how to store segment files on StorHouse. A user tablespace consists of one or more subspaces. *See also* subspace.

user tablespace ID A numeric identifier that StorHouse/RM assigns to a user tablespace when you issue a CREATE TABLE SPACE statement. Tablespace ID 0 is reserved for the system tablespace and tablespace ID 1 is reserved for the temporary tablespace.

user tablespace name An SQL identifier that you provide on the CREATE TABLE SPACE statement to name a user tablespace.

value index A list of all values in one or more columns of a given user table. StorHouse/RM typically uses a value index to search for a range of values or for a specific value.

value index file The value index entries that are stored in STORHOUSE-type files on StorHouse.

variable-length record A record with a length that varies from other records in the file. Each record is preceded by a shortword (2-byte integer) that indicates the length of the data record. *Contrast* fixed-length record, text record, and LOB record.

view An alternative representation of data from one or more tables. A view can include some or all of the columns contained in tables on which the view is based.

volume A unit of removable media, such as an optical disk cartridge or tape cartridge.

volume set (VSET) A collection of one or more physical volumes that StorHouse treats as a unit. Every removable volume in StorHouse is a member of a volume set.

volume set name A name of a volume set which can contain from one to eight characters and can consist of any of the following characters: a–z, A–Z, 0–9, _ (underscore), and $ (dollar sign). A volume set name must be unique within StorHouse. *See also* file set name.

VRAM file A StorHouse file that is managed by the StorHouse Virtual Record Access Manager (VRAM) software component. You can load input data from a VRAM file, and you can collect discarded records in a discard (VRAM) file.

VTF *See* Vulnerability Time Factor.

Vulnerability Time Factor (VTF) An attribute that determines whether and when to create performance copies and/or backup copies of segment files. You set this attribute in each subspace of a user tablespace.

whitespace A blank, space, or tab in character data. *See also* leading blanks and trailing blanks.

WITH GRANT OPTION The SQL GRANT statement component that enables the specified accounts to grant their privileges or a subset of their privileges to other accounts.

wrapper The software that implements a connection in a federated system. The StorHouse/UDB Link is the FileTek-supplied wrapper software to implement the connection between DB2 UDB and StorHouse.

write-back To copy new file extents from the performance buffer to their primary file sets. The VTF attribute in a user tablespace determines when write-back occurs for segment files.

write lock *See* exclusive lock.