

---

{button a,JI('`\_a\_')}{button b,JI('`\_b\_')}{button c,JI('`\_c\_')}{button d,JI('`\_d\_')}{button e,JI('`\_e\_')}  
{button f,JI('`\_f\_')}{button g,JI('`\_g\_')}{button h,JI('`\_h\_')}{button i,JI('`\_i\_')}{button j,JI('`\_j\_')}  
{button k,JI('`\_k\_')}{button l,JI('`\_l\_')}{button m,JI('`\_m\_')}  
{button n,JI('`\_n\_')}{button o,JI('`\_o\_')}{button p,JI('`\_p\_')}{button q,JI('`\_q\_')}{button r,JI('`\_r\_')}  
{button s,JI('`\_s\_')}{button t,JI('`\_t\_')}{button u,JI('`\_u\_')}{button v,JI('`\_v\_')}{button w,JI('`\_w\_')}  
{button x,JI('`\_x\_')}{button y,JI('`\_y\_')}{button z,JI('`\_z\_')}

## **A**

[access method](#)  
[alias](#)  
[array](#)  
[attach](#)

## **B**

[BLOB data type](#)  
[BLOB filter](#)  
[buffer cache](#)

## **C**

[CHARACTER data type](#)  
[computed column](#)

## **D**

[database engine](#)  
[database handle](#)  
[DATE data type](#)  
[DECIMAL data type](#)  
[default transaction](#)  
[disk shadow](#)  
[domain](#)  
[DOUBLE PRECISION data type](#)

## **E**

[event](#)  
[event alerter](#)  
[event manager](#)  
[extended SQL descriptor area](#)  
[external function](#)  
[external table](#)

## **F**

[filter](#)  
[FLOAT data type](#)

## **G**

[garbage collection](#)  
[gbak](#)  
[generator](#)  
[gfix](#)  
[gpre](#)  
[gsec](#)

## **H**

[handle](#)  
[host language](#)

**I**

INTEGER data type

ISC\_LONG

ISC\_QUAD

isc4.gdb

isql

**J****K****L**

limbo transaction

local attachment

lock

locking

lock manager

**M**

metadata

multi-database access

multi-database application

multi-file database

multi-generational architecture

**N**

named transaction

NUMERIC data type

**O**

ODS

on disk structure (ODS)

**P**

page

procedure

procedure and trigger language

**Q****R**

record version

recovery

remote attachment

remote database

**S**

segment

shadow

shadow file

shadow set

slice

SMALLINT data type

SQLDA

SQL descriptor area

status vector

stored procedure

stored procedure and trigger language  
subtype  
sweep  
synchronous events

**T**

transaction name  
trigger  
two-phase commit

**U**

UDF  
user-defined function (UDF)

**V**

VARCHAR data type

**W**

**X**

XSQLDA

**Y**

**Z**

**alias**

An SQL correlation name.

**array**

A data structure consisting of multiple dimensions and elements of a single InterBase data type and stored in a single column. Similar to an array in C.

**attach**

To connect to and open a database for use. An attachment can be local or remote.

**BLOB data type**

A binary data type, variable in size, used to store unformatted data such as text, graphics, or any other data.

**BLOB filter**

A user-written routine for processing BLOB data, for example, to convert the data from one format to another.



**buffer cache**

An area of memory allocated to a database attachment on the server, used to speed data access by keeping frequently-used data and index pages in the server's core memory.

**CHARACTER data type**

A fixed-length string data type, 1 to 32,767 characters in size, used to store text.

**checkpoint**

A point in time in which accumulated log file data reaches a predetermined size and is written to disk. It is part of the deprecated WAL protocol.

**computed column**

A column whose value is derived from a formula or an arithmetic expression.

**database engine**

The InterBase server component that handles all database access to a database file.

**database handle**

A name assigned to a database by an application that must be used in subsequent references to that database.

**DATE data type**

A numeric data type, ISC\_QUAD in size, used to store day and time values from January 1, 100 a.d. to February 29, 32768 a.d.. The size of ISC\_QUAD is platform-specific.

**DECIMAL data type**

A numeric data type, variable in size, used to store real numbers with a fixed decimal point. The user-specified range of precision is at least one to fifteen digits.



**default transaction**

The transaction under which SQL statements are executed unless a named transaction is explicitly used instead.

**disk shadow**  
See shadow.

**domain**

A column definition template, global to the database, upon which actual column definitions in tables may be based.

**DOUBLE PRECISION data type**

A numeric data type, 64 bits in size, used to store real numbers up to fifteen decimal points of precision.

**event**

A named string, up to 31 characters in length, usually about a change, posted to the InterBase event manager by a trigger or a stored procedure.

**event alerter**

A trigger or stored procedure that posts an event message, usually about a change, to the InterBase event manager.

**event manager**

An InterBase process that receives event messages and notifies applications about them.

**extended SQL descriptor area**

A user-declared data structure, used to hold DSQL for input and output.



**external function**

See [user-defined function](#).

**external table**

A flat data file used by a database as if it were an internal table.

**filter**

See [BLOB filter](#).

**FLOAT data type**

A numeric data type, 32 bits in size, used to store real numbers up to seven decimal points of precision.

**garbage collection**

Automatic deletion of record versions superseded by more current data.

**gbak**

An InterBase utility for backing up and restoring databases.

**generator**

A database object accessed by the built-in function, `GEN_ID()`, to generate sequential integer values, often used to assign unique values to a column.

**gfix**

An InterBase utility for performing database maintenance such as shutting down the database and recovering limbo transactions.



**gpre**

The InterBase preprocessor used to convert embedded SQL statements and variables into a format acceptable to a host-language compiler.

**gsec**

An InterBase utility for maintaining the security database, [isc4.gdb](#).

**handle**

See [database handle](#).

**host language**

A programming language, such as C, used for database application development.

**INTEGER data type**

A numeric data type, 32 bits in size, used to store integer values ranging from -2,147,483,648 to 2,147,486,647.

**ISC\_LONG**

Data type of correct size to contain an address.

**ISC\_QUAD**

64-byte data structure used to hold BLOB IDs and DATE values.

**isc4.gdb**

The InterBase security database, unique to each server, whose entries determine whether a remote client has permission to attach to a database.



**isql**

Interactive SQL, an InterBase utility for data definition, data manipulation, and viewing metadata.

**limbo transaction**

A transaction that is neither committed nor rolled back, usually because of a failure of a two-phase commit or system failure.

**lock**

A level of access granted by the lock manager to a database resource.

**locking**

A mechanism that controls access to database resources.

**lock manager**

A feature that coordinates the sharing of database resources among multiple processes.

**log buffer**

An area of memory set aside to store database changes. These changes are written to log files when the buffers become full or when the changes are committed.

**log file**

Disk files that permanently record changes flushed from log buffers. When a log file becomes full, the WAL protocol determines whether a new log file is used or an old log file is reused.

**metadata**

Database system tables that describe the database's tables, columns, indexes, triggers, privileges, views, domains, procedures, and integrity constraints.



**multi-database access**

Accessing more than one database at the same time.

**multi-database application**

An application that accesses more than one database at a time.

**multi-file database**

A database that stores its data in more than one file.

**multi-generational architecture**

InterBase's structure for maintaining multiple record versions of data. It enables readers and writers concurrent and consistent access to data.

**named transaction**

A transaction explicitly named and started by an application to control execution of SQL statements in place of the default transaction.

**NUMERIC data type**

A numeric data type, variable in size, used to store real numbers with a fixed decimal point. The user-specified range of precision can be exactly one to fifteen digits.

**ODS (on disk structure)**

The physical layout and binary image of database files stored on disk. The ODS version depends on the InterBase version being used.

**page**

An internal InterBase storage structure that determines how data is stored on disk. A page affects the levels of index buckets and storage, retrieval, and buffering of data. Allowable page sizes are: 1KB, 2KB, 4KB, and 8KB.



**record version**

An instance of data tied to a specific transaction. Transactions that update the same data create multiple record versions.

**recovery**

Reconstructing the database in the event of disk or network failure, or other disasters.

**remote database**

A database on a server that is accessed by a client across a network.

**segment**

A unit of BLOB data stored or retrieved in a single operation. Segments can be up to 64 kilobytes in size.

**shadow**

A physical copy of the database. Changes to the database are written simultaneously to the shadow.

**shadow file**

One or more files that make up a shadow.

**shadow set**

A group of shadow files that make up a single shadow.

**short-term recovery**

Bringing a database up to date by applying changes stored in log files. Log files contain committed changes not yet written to the database.



**slice**

A contiguous subset of array elements retrieved and manipulated with a single statement.

**SMALLINT data type**

A numeric data type, signed short in length, used to store integer values ranging from -32,768 to 32,767.

**SQLDA**

SQL descriptor area. See [extended SQL descriptor area](#).

**status vector**

An array of twenty ISC\_LONG elements used to report run-time errors by a database to an application.

**stored procedure**

A stored procedure is a self-contained program written in InterBase procedure and trigger language, and stored as part of a database's metadata. Stored procedures can be invoked directly from applications, or can be substituted for a table or view in a SELECT statement. Stored procedures can receive input parameters from and return values to applications and can execute on the server.

**stored procedure and trigger language**

A complete programming language for stored procedures and triggers that includes SQL data manipulation statements and extensions, including IF . . THEN . . ELSE, WHILE . . . DO, FOR SELECT . . . DO, exceptions, and error handling.

**subtype**

A number that specifies the type of data contained in a BLOB.

**sweep**

A process that checks a database record by record to garbage collect outdated record versions.



**synchronous events**

An event posted to a waiting application when it occurs. See [event](#).

**transaction name**

A unique name that must be used to distinguish a transaction from others in a multi-transaction program.

**trigger**

A self-contained routine associated with a table or view that automatically performs an action when a row in the table or view is inserted, updated, or deleted.

**two-phase commit**

A mechanism that guarantees transactions affecting multiple databases are committed in their entirety or not at all.

**UDF (user-defined function)**

A database function written entirely in a host language to perform data manipulation tasks not directly supported by InterBase. Executed on the server.

**VARCHAR data type**

A variable string data type, 1 to 32,767 characters bytes in size, used to store text.

**WAL (write-ahead log) protocol**

WAL was a feature of earlier versions of InterBase on NetWare only. **This feature is deprecated and not supported in current versions of InterBase.** WAL was a method for caching data input and output. The WAL protocol stores changes in log buffers and files that are used in short-term recovery.

**XSQLDA**

See extended SQL descriptor area.



