Datalog Educational System
Release Notes History

Fernando Sáenz Pérez
Grupo de Programación Declarativa (GPD)
Universidad Complutense de Madrid (UCM)

Contents

1. Release Notes History

This document lists release notes of all software releases previous to the current one in reverse chronological order. Release notes for the current version are listed in the User Manual.

1.1 Version 3.3 of DES (released on July, 12th, 2013)

- Enhancements:
  - ACIDE bundle includes a new version 0.11 featuring a database panel for managing both DES databases (default **`$des`** and external databases) and regular ODBC connections (tested with MySQL and DB2). In addition, it has been enhanced w.r.t. last bundle (including version 0.9) and fixes several bugs
  - Nulls can be disabled for trying pure Datalog. Also, disabling nulls enhances performance a bit (most likely, not noticeable)
  - Upgraded performance. Some tweaks for performance optimization
  - Extended support for negation in hypothetical rules
  - Help listings (commands and built-ins) restricted to a limited width
  - Added new ISO built-in infix operator **`mod`**
  - Redefinition of built-in comparison operators are avoided
  - New commands:
    - **`/display_nbr_of_tuples`** Display whether display of the number of computed tuples is enabled
    - **`/display_nbr_of_tuples`** ***`Switch`*** Enable or disable display of the number of computed tuples (**`on`** or **`off`**, resp.)
    - **`/nulls`** Display whether nulls are enabled
    - **`/nulls`** ***`Switch`*** Enable or disable nulls (**`on`** or **`off`**, resp.) If nulls are disabled, calls to outer join predicates included in already-loaded rules will fail, and attempts to use outer joins will not succeed. This, coupled with **`/duplicates off`** (as by default) allows to play with pure Datalog with negation and arithmetic built-ins
    - **`/optimize_cc`** Display whether complete computations optimization is enabled
    - **`/optimize_cc`** ***`Switch`*** Enable or disable complete computations optimization (**`on`** or **`off`**, resp. and enabled by default) Fixpoint iterations and/or extensional database retrievals might been saved
    - **`/optimize_ep`** Display whether extensional predicates optimization is enabled

- **`/optimize_ep`** ***`Switch`*** Enable or disable extensional predicates optimization (**`on`** or **`off`**, resp. and enabled by default). Fixpoint iterations and extensional database retrievals are saved for extensional predicates as a single linear fetching is performed for computing them
- **`/optimize_edb`** Display whether extensional database optimization is enabled
- **`/optimize_edb`** ***`Switch`*** Enable or disable extensional database optimization (**`on`** or **`off`**, resp. and enabled by default). Extensional database retrievals are saved for the extensional part of the deductive database
- **`/optimize_nrp`** Display whether non-recursive predicates optimization is enabled
- **`/optimize_nrp`** ***`Switch`*** Enable or disable non-recursive predicates optimization (**`on`** or **`off`**, resp. and enabled by default). Memoing is only performed for top-level goals
- **`/optimize_st`** Display whether stratum optimization is enabled
- **`/optimize_st`** ***`Switch`*** Enable or disable stratum optimization (**`on`** or **`off`**, resp. and enabled by default). Extensional table lookups are saved for non-recursive predicates calling to recursive ones, but more tuples might be computed if the non-recursive call is filtered, as in this case an open call is submitted instead (i.e., not filtered)
- **`/write`** ***`String`*** Write ***`String`*** to console. ***`String`*** can contain system variables as **`$stopwatch$`** (which holds the current stopwatch time) and **`$total_elapsed_time$`** (which holds the last total elapsed time) (See Subsection **¡Error! No se encuentra el origen de la referencia.** for system variables)
- **`/writeln`** ***`String`*** As **`/write`** but adding a new line at the end of the string
- **`/write_to_file`** ***`File String`*** Write ***`String`*** to ***`File`***. If ***`File`*** does not exist, it is created; otherwise, previous contents are not deleted and ***`String`*** is simply appended to ***`File`***. ***`String`*** can contain system variables as **`$stopwatch$`** (which holds the current stopwatch value) and **`$total_elapsed_time$`** (which holds the last total elapsed time)
- **`/writeln_to_file`** ***`File`*** As **`/write_to_file`** but adding a new line

- Changes:
  - Behaviour of **`/compact_listings`** ***`Switch`*** is immediate (neither trailing blank line when enabling compact listings nor absent blank line when disabling)
  - SQL-to-Datalog compilations for the division are not displayed unless development listings are enabled, analogously to outer join operations

- Operator `\/` in hypothetical literals has been changed to the more appropriate `/\`

- Fixed bugs:
  - During computing implications, some rules were not memorized correctly
  - Assumed rules in hypothetical queries and rules were not tested for safety
  - Parsing of **`/shell`** failed for arguments containing a comma. Now, the characters following the command are directly sent to the shell
  - Attributes in where conditions were not parsed in the command **`/des`**
  - External relations were warned as non-existing when processing a SQL statement with the command **`/des`**
  - Some commands did not accept upper case switches
  - Closing an ODBC connection broke external metadata retrieval for subsequent connections
  - The command **`/listing`** duplicated the rules for persistent predicates
  - The command **`/rm File`** (with synonym **`/del`**) did not find the file to remove for SICStus distros
  - Missing display of top-level query for exploded queries in normal listings
  - When **`show_sql`** was enabled, some tasks related to persistent predicates failed, as retracting rules
  - Hypothetical Datalog queries for persistent predicates did'nt retract assumed rules and facts

1.2 Version 3.2 of DES (released on February, 2nd, 2013)

- Enhancements:
  - Datalog hypothetical queries and rules. The implication operator `=>` is added to support this
  - Datalog division operator. This operator implements relational division but without resorting to schemas. Instead, variables are used
  - SQL **`CHECK`** clause as both column and table constraints, including any SQL check as occurring in a **`WHERE`** clause
  - Functional dependencies can be specified in **`CREATE TABLE`** SQL statements, *à la* DB2, with the **`CHECK`** constraint **`DETERMINED BY`**
  - New Datalog metapredicates **`order_by`**/2 and **`order_by`**/3, which allow multi-key sorting on answers
  - SQL **`ORDER BY`** clause is now working and with multi-key sorting
  - SQL **`INSERT INTO`** … **`VALUES`** allows a sequence of tuples to be inserted
  - New commands:
    - **`/order_answer`** Display whether displayed answers are ordered by default

- **`/order_answer`** ***`Switch`*** Enable or disable a default (ascending) ordering of displayed computed tuples (**`on`** or **`off`**, resp.)
- **`/statistics`** ***`Keyword`*** Display statistics for ***`Keyword`*** (**`runtime`** or **`total_runtime`**). For **`runtime`**, this command displays the CPU time used while executing, excluding time spent in memory management tasks or in system calls. For **`total_runtime`** , this command displays the total CPU time used while executing, including memory management tasks such as garbage collection but excluding system calls.

- Null identifiers in user queries are reused when possible
- Non existing columns are reported when asserting Datalog predefined constraints
- Added postfix arithmetic built-in +
- Enhanced parsing of SQL, RA and Datalog arithmetic expressions
- New port to SWI-Prolog 6.2.6

- Changes:
  - Body output of the command **`/status`** is ordered
  - For **`/dbschema`** output, a Datalog integrity constraint is displayed under a table if it only refers to this table and under the (database) integrity constraints otherwise. Therefore, Datalog constraints are not listed more than once as in previous versions, in which a constraint was listed for each table it referred to. If a constraint is created with a **`CREATE TABLE`** ***`Tablename`*** statement, it is listed under the table ***`Tablename`*** even when it refers to other tables or views
  - ISO Prolog **`xor`** bitwise exclusive or replaces # operator
- Fixed bugs:
  - Asserting a fact did not update the predicate dependency graph always
  - The metapredicates **`top`** and **`distinct`** raised unsafe rule warnings on safe rules in some cases
  - Only one not null column constraint was stored for a CREATE TABLE statement
  - Grouping might not create duplicates for aggregation results as in: **`group_by(t(_X,Y),[_X],S=sum(Y))`**

1.3 Version 3.1 of DES (released on December, 20th, 2012)

- Enhancements:
  - Deductive engine inference is now possible for an open ODBC connection, enabling to directly submit queries which are not supported by the external DBMS (cf. **`/des_sql_solving`** command)
  - Division operation is now supported for the relational algebra query language

- SQL has been extended beyond standard with the relational division operation as an operator in the **FROM** clause of a **SELECT** statement
- New commands:
  - **/db_schema** Synonym for all variants of **/dbschema**
  - **/des *Input*** Force DES to solve ***Input***. If ***Input*** is a SQL query, DES solves it instead of relying on external DBMS solving. This allows to try the more expressive queries which are available in DES (as, e.g., hypothetical and non-linear recursive queries)
  - **/des_sql_solving** Display whether DES is forced to solve SQL queries for external DBs
  - **/des_sql_solving *Switch*** Enable or disable DES SQL solving for external DBs
  - **/edit *Filename*** Edit ***Filename*** by calling the predefined external text editor. This editor is set with the command **/set_editor**
  - **/format_timing** Display whether formatted timing is enabled
  - **/format_timing *Switch*** Enable or disable formatted timing (on or off, resp.)
  - **/prompt** Display the prompt format
  - **/prompt *Switch*** Set the format of the prompt. The value 'des' sets the prompt to 'DES>'. The value 'des_db' adds the current database name DB as 'DES:DB>'. Finally, 'plain' sets the prompt to '>'. Note that, in any case, if a language other than Datalog is selected, the language name is also displayed before ">"
  - **/set_editor** Display the current external text editor
  - **/set_editor *Editor*** Set the current external text editor to ***Editor***
- Stopwatch precision of 1 millisecond for SICStus distros
- Better performance for persistent predicates updates by identifying base cases (facts)
- Better SQL and RA compilations to Datalog
- SQL to Datalog compilations are also displayed for DML statements
- Added verbose info on solving SQL and RA queries
- Better identification with relation names of non-existing columns
- Added the SQL type **NUMBER(POS,DEC)**, although not checked as domain restrictor
- Support for persistency tested in and adapted for more external RDBMS's (IBM DB2 10.1, Sybase ASE 15.7, Oracle 11g in addition to the already tested MS Access 2003, MySQL 5.1, PostgreSQL 9.1, and MS SQL Server 2008)
- New port to SICStus Prolog 4.2.3
- New port to SWI-Prolog 6.2.4

  - More examples (e.g., puzzle **`bridge.pl`** and **`even_odd.sql`**)

- Changes:
  - As the ODBC connection to Oracle RDBMS is really slow in some systems, for opening a connection, only user tables and views are read to build the PDG
  - When an opened ODBC connection is the current one, queries are no longer parsed by DES
  - Output display disabling also affects to logging
  - Tracing a view also shows its definition
  - Variables as shown in type errors become named
  - Visibility rules of RA are as original proposal
  - Last stage of source-to-source transformations is now displayed with **`/show_compilations on`**
- Fixed bugs:
  - The command **`/abolish`** with arguments and referring to a non-persisted predicate raised an input processing error
  - Mutually recursive definitions in a single SQL statement led to compilation rejection
  - Query schema was missing in the query outcome (bug introduced in version 3.0)
  - Some running information were displayed when output was off
  - Some non-ground arithmetical expressions generated an exception during parsing
  - Datalog rules as compiled from a SQL query were not removed upon an exception
  - In SICStus distros, SQL answers with repeated column names were not computed correctly for an ODBC connection
  - **`count(*)`** was simply displayed as **`count`** in SQL statements for command **`/dbschema`**
  - The full join operation returned more tuples than expected
  - Some SQL keywords (e.g., **`IN`**) required a leading or a preceding blank (an opening and closing parenthesis, resp., is also allowed now)
  - **`IS NULL`** and **`IS NOT NULL`** SQL conditions were not parsed (bug introduced in version 2.7)
  - Rules and SQL statements including null checking were not tested for safety
  - 'Group by' statements and rules including predicates/relations as data sources in the having condition were not correctly computed because the predicate dependency graph didn't include such data sources
  - Some safe rules including the predicate **`group_by`** were detected as unsafe

- Some code simplifications involving the metapredicate **`distinct`** were incomplete
- After an exception during solving a SQL query, compilations were not removed
- Some translations of SQL statements involving **`NOT EXIST`** clauses were not correct
- The condition in the predicate **`group_by`** was not tested for safety so that no program transformations were applied to possibly develop safe rules
- Inserting a foreign key assertion was not checked correctly in all cases
- Inserting a tuple into a table with a compound primary key was not checked correctly in all cases
- Some Datalog simplifications forced by SQL compilations were wrong
- A nested **`UNION`** (removing duplicates: **`DISTINCT`**) SQL statement lost the connecting references
- Some expression and column references in SQL statements have been fixed
- In some cases, a type error relating matching numeric expressions was raised

1.4 Version 3.0 of DES (released on May, 10th, 2012)

- Enhancements:
  - New commands:
    - **`/close_db`** ***`Name`*** Close the given ODBC connection. *TAPI enabled*
    - **`/drop_assertion`** Drop an assertion
    - **`/start_stopwatch`** Start stopwatch. Precision depends on host Prolog system (1 second or milliseconds)
    - **`/stop_stopwatch`** Stop stopwatch
    - **`/reset_stopwatch`** Reset stopwatch
    - **`/display_stopwatch`** Display stopwatch
    - **`/list_persisted`** Display the persisted predicates. *TAPI enabled*
    - **`/show_dbs`** Display the open database connections. *TAPI enabled*
    - **`/show_sql`** Display whether SQL statements which are sent to an external database are to be displayed
    - **`/show_sql`** ***`Switch`*** Enable or disable display of SQL statements which are sent to an external database (on or off, resp.)
    - **`/use_db`** ***`Name`*** Make ***`Name`*** the current ODBC connection. *TAPI enabled*
    - **`/dbschema`** ***`Connection`*****`:`*****`Name`** Display the database schema for the given view or table name in the given connection

- **`/license`** Display GPL and LGPL licenses. If not found, please visit http://www.gnu.org/licenses

o New assertions:

- **`:-persistent(`*`PredSpec`*`[,`*`Connection`*`]))`** Make a predicate to persist on an external RDBMS via an ODBC connection. ***`PredSpec`*** can be either the pattern ***`PredName/Arity`*** or ***`PredName`*`(`*`Schema`*`)`**, where ***`Schema`*** can be either **`ArgName1, …, ArgNameN`** or **`ArgName1:Type1, …, ArgNameN:TypeN`**. If a connection name is not provided, the current open database is used

o Binary flags in commands are no longer case-sensitive

o New port to SICStus Prolog 4.2.1. This release fixes in particular some issues with ODBC connections (exceptions about misencoded string in non-ASCII ODBC messages, and incorrect handling of SQL_BIGINT and related types)

o New port to SWI-Prolog 6.0.2

- Changes:

o License has been relaxed to GNU Lesser General Public License

o New versions of command **`/debug_sql`** does not admit a traversing order yet (order option removed)

o Release notes of older DES versions are moved to the new document: **`releasenotesDES.pdf`**

- Fixed bugs:

o Some spanned inputs without leading blanks in multi-line mode were not recognised

o Duplicated object rules were retrieved several times

o Some commands were not recognized in mixed or uppercase

o Some listings in development mode did not display all rules

o Some hypothetical queries led to exceptions

o Existency of table and attributes in an **`INSERT`** SQL statement with a SQL data source was not checked

o Parsing of a SQL relation separated by a leading space before the comma lead to syntax error

o Predefined strong constraints relating a tuple of column names were rejected if its lexicographic order did not match the order in which they occur in table definition

o Running info were logged

o Some rules with conjunctions and disjunctions were not parsed correctly from consulted files

o GNU Prolog source distribution stopped processing of batch files while encountering a **`/shell`** command

- Predicate dependency graph and strata were not computed after issuing DML SQL statements **INSERT**, **DELETE** and DQL SQL statement **WITH**

1.5 Version 2.7 of DES (released on January, 3rd, 2012)

- Enhancements:
  - Extended relational algebra processor including all the original operators but division, and extended operators for dealing with outer joins, duplicate elimination, recursion, and grouping with aggregates
  - Multi-line input is also allowed in addition to the current single-line input. Long inputs as typical SQL statements can be spanned over several lines. When multi-line is enabled with the command **/multiline on**, Datalog inputs must end with a dot (.), and SQL and RA inputs with a semicolon (;). When disabled, each line is considered as a single (Datalog, SQL or RA) input and ending characters are optional
  - When multi-line input is enabled, remarks enclosed between **/*** and ***/** can span over several lines and can be nested as well
  - Single-line (**--**) and multi-line (**/**/**/**) remarks can be included in SQL statements at any place a separating blank can occur
  - SQL statement **CREATE TABLE** can include **LIKE** for creating a table with the same schema as an existing one
  - SQL statement **DROP TABLE** can include **IF EXISTS** clause and can apply to a list of tables
  - New (non-standard) SQL metadata statements (catalogued under ISL, Information Schema Language):
    - **SHOW TABLES;** List table names. *TAPI enabled*
    - **SHOW VIEWS;** List view names. *TAPI enabled*
    - **SHOW DATABASES;** List database names. *TAPI enabled*
    - **DESCRIBE *Relation*;** Display schema for ***Relation***, as **/dbschema** command does. *TAPI enabled*
  - New commands:
    - **/list_tables** List table names. *TAPI enabled*
    - **/list_views** List view names. *TAPI enabled*
    - **/multiline** Display whether multi-line input is enabled
    - **/multiline *Switch*** Enable or disable multi-line input (**on** or **off** resp.)
    - **/ra** Switch to RA interpreter
    - **/ra *Query*** Execute an RA query
    - **/referenced_relations *Name*** Display relations directly referenced by a foreign key in ***Name***

  - **`/referenced_relations`** ***`Name/Arity`*** Display relations directly referenced by a foreign key in ***`Name/Arity`***
- Last line in a processed file must not end with a carriage return for its processing
- Faster abolish command and drop database SQL statement
- Display of the number of consulted constraints, if any
- Exceptions during constraint checking when consulting files are caught
- Faster parsing of Datalog rules and SQL statements
- A pivot variable that does not occur in the aggregate relation raises a syntax error
- Views are not required to be created with given column names
- Submitting a query or creating a view with duplicated columns is rejected
- Language command error messages instead of just "Input processing error"
- Improved compilation of **`EXISTS`** SQL clauses, using Datalog built-in **`top`**/2, which allows to prune the number of computed tuples

- Changes:
  - The system prompt for Datalog language changes to the old prompt **`DES>`**, as almost any input can be handled from this setting. The only inputs that must explicitly submitted to a language processor are those that can be handled by several language processors
  - Null identifiers are not wasted as eagerly as in previous versions
  - Negation algorithm **`et_not`** do not longer rely on computations by strata
  - New organization of system files:
    - **`des_sql_debug.pl`** (debugger extracted from former **`des_sql.pl`**)
    - **`des_dl_debug.pl`** (replaces **`des_debug.pl`**), and
    - **`des_ra.pl`** (includes RA processor)
- Fixed bugs:
  - Listings of SQL statements including Top-N queries failed
  - After submitting an incorrect SQL view, all of its temporary schema was not cleaned up
  - Variable names in consulted Datalog constraints were lost
  - New schema names defined in the list of local definitions inside a **`WITH`** or **`ASSUME`** SQL statement were not handled appropriately. Bug introduced in version 2.6
  - Only one blank was allowed after a **`SELECT`** statement. Bug introduced in version 2.6

- Operator precedence in SQL conditions and Datalog bodies was not correctly handled (parentheses were needed to ensure correct operator applications)
- Renamed relations could not be enclosed between parentheses
- A renamed argument in a nested query was not visible for the **`WHERE`** condition of its outer query
- Expressions in nested SQL queries could not be referenced from outermost queries
- Type inference failed in some situations for equivalent internal string types (cf. russell.sql). Bug introduced in version 2.6
- Underscored variables in a head rule made rule assertion fail
- The Prolog interpreter did not handle conjunctive and disjunctive queries

1.6 Version 2.6 of DES (released on October, 26th, 2011

- Enhancements:
  - A novel proposal for hypothetical SQL queries which allow to assume extra tuples in existing relations (either tables or views)
  - New SQL Top-N queries following ISO 2008 (another common form **`TOP N`** is also supported)
  - New Datalog built-in **`top`**/2 for computing Top-N queries, i.e., those with the number of tuples in the answer limited to N
  - SQL statements are allowed in the projection list, even as components of arithmetic expressions
  - Anonymous variables are discarded from the answer schema should they occur in queries, views and autoviews (even in heads)
  - New sub-graph algorithm for finding predicate dependency graphs restricted to queries. It replaces an older one with exponential complexity, which did consulting and/or querying some small programs to raise memory exhaust exceptions
  - Display of predicate dependency graph is ordered
  - Display of strata is first ordered by strata and then by predicate
  - Running info display about number of inferred tuples, working with console and windows applications
  - Datalog built-ins **`distinct`**/1 and **`distinct`**/2 also work for arbitrary queries, not only for atoms
  - Enhanced solving performance by hash-indexing of extension table
  - Enhanced time displays. Time is formatted as either milliseconds (MS) (MS ms.) for less than a second; or seconds (SS) and milliseconds (MS) (SS.MS s) for less than a minute; or minutes (MM), seconds and milliseconds (MM:SS.MS) for less than an hour; or hours, minutes, seconds and milliseconds (H:MM:SS) for greater than or equal to an hour

- Case-insensitive interactive user answers (debugging, test cases, ...)
- Keyboard interrupt caught for SWI-Prolog distributions. This allows interrupt the current computation without exiting DES
- Syntax error report on incompatible relation schemas in set operations
- Syntax error report about mismatch type for condition in metapredicate **`group_by`**
- Added hints on misspelled commands
- Help on commands and built-ins
- Hints on misspelled entries for:
  - Existing commands **`/debug_datalog`** and **`/trace_datalog`**
  - New commands **`/pdg`**
- Enabling TAPI for the next existing commands and synonyms:
  - **`/consult`**
  - **`/reconsult`**
  - **`/cd`**
  - **`/pwd`**
- New commands:
  - **`/check_db`** Check database consistency w.r.t. declared integrity constraints (types, existency, primary key, candidate key, foreign key, functional dependency, and user-defined). Display a report with the outcome
  - **`/display_answer`** Display whether display of computed tuples is enabled
  - **`/display_answer`** ***`Switch`*** Enable or disable display of computed tuples (**`on`** or **`off`**, resp.) The number of tuples is still displayed
  - **`/hypothetical`** Display whether hypothetical SQL queries are allowed
  - **`/hypothetical`** ***`Switch`*** Enable or disable hypothetical SQL queries (**`on`** or **`off`**, resp.)
  - **`/indexing`** Display whether hash indexing on extension table is enabled
  - **`/indexing`** ***`Switch`*** Enable or disable hash indexing on extension table (**`on`** or **`off`**, resp.) Default is enabled, which shows a noticeable speed-up gain in some cases
  - **`/pdg`** ***`PredName`*** Display the current predicate dependency graph restricted to the first predicate found with name ***`PredName`***

- **`/pdg` *`PredName/Arity`*** Display the current predicate dependency graph restricted to the predicate with name ***`PredName`*** and ***`Arity`***

- Changes:
  - Constraints assertions are not checked when disabling constraint checking
- Fixed bugs:
  - Integrity constraint checking could not be changed. Bug introduced in version 2.4
  - Syntax error exceptions when consulting files exited DES
  - A **`FROM`**-less SQL statement in a series of local view definitions of a **`WITH`** statement was not parsed
  - SQL parsing involving non-existent column names tweaks for discarding incorrect statements and accept correct statements
  - The command **`/abolish`** deleted rules of view definitions but not the schema
  - Datalog rule listings with added parentheses enclosing disjunctions when needed
  - Some nested SQL statements containing expressions were not parsed
  - SWI-Prolog distributions included incorrect computation time displays when detailed timing was enabled
  - Some unsafe rules involving set variables were not transformed when safety transformation was enabled
  - Use of set variables in equalities and is/2 always yielded to error messages, although the use were correct

1.7 Version 2.5 of DES (released on September, 13th, 2011)

- Enhancements:
  - A textual API for connecting DES with external systems. Several commands and queries can be read and answered using standard streams. Currently, TAPI-enabled queries and commands are those listed in Section 5.12, which are needed to interface database schemas and data to ACIDE [Sae07]. Queries include SQL DDL, DML, and DQL statements. Datalog constraint assertions and deletions are also supported
  - New Datalog (strong) integrity constraints: candidate key (uniqueness) and existency (forbid nulls) integrity constraints. Commands **`/save_ddb`** and **`/restore_ddb`** apply for such new constraints
  - Support for **`UNIQUE`** and **`NOT NULL`** column and table constraints in SQL **`CREATE TABLE`** statements
  - Added support to specify column names in SQL **`INSERT INTO`** statements
  - Nulls are no longer allowed in primary key columns

- Type inferencing added to SQL DQL queries, in addition to the already supported DDL queries
- Added type mismatch report error for ill-typed SQL statements
- Answers from SQL queries are annotated with their inferred types
- Limited-length types are also inferred for views and queries
- Types returned by ODBC connections are labelled with their lengths
- Tables and views are sorted in the result of command **`/dbschema`**
- Column names are ordered in predefined integrity constraint displays
- Enhanced SQL syntax error reporting for built-ins used as table and column identifiers
- SQL syntax error reporting for unknown columns includes hints about similar column names, in addition to the already hints about table and view names
- Commands involving table, view or relation names which are not defined provide hints
- Hints on alternative names also include names with swapped characters
- Trying to use a built-in symbol as a user identifier in a SQL statement is warned as a syntax error
- Simplified error messages
- Extension table is not cleared when enabling duplicates. Instead, complete flag is reset, avoiding much recomputation
- New non-standard SQL statements **`RENAME TABLE`** and **`RENAME VIEW`**
- New commands:
  - **`/tapi`** ***`Input`*** Process ***`Input`*** and format its output for TAPI communication.
  - **`/test_tapi`** Test the current TAPI connection. *TAPI enabled*
  - **`/drop_ic`** ***`Constraint`*** Drop the specified integrity constraint, which starts with ":-"
  - **`/dependent_relations`** ***`Relation`*** Display the names of relations that depend on relation ***`Relation`***. *TAPI enabled*
  - **`/dependent_relations`** ***`Relation/Arity`*** Display in format Name/Arity those relations that depend on relation ***`Relation/Arity`***. *TAPI enabled*
  - **`/is_empty`** ***`relation_name`*** Display **`$true`** if the given relation is empty, and **`$false`** otherwise *TAPI enabled*
  - **`/list_table_constraints`** ***`table_name`*** List table constraints for ***`table_name`***. *TAPI enabled*
  - **`/list_table_schemas`** List table schemas. *TAPI enabled*
  - **`/list_view_schemas`** List view schemas. *TAPI enabled*

- **`/referenced_relations`** ***`Relation`*** Display the name of relations that are directly referenced by a foreign key in relation ***`Relation`***. *TAPI enabled*
- **`/referenced_relations`** ***`Relation/Arity`*** Display in format Name/Arity those relations that are directly referenced by a foreign key in relation ***`Relation/Arity`***. *TAPI enabled*
- **`/relation_exists`** ***`relation_name`*** Display **`$true`** if the given relation exists, and **`$false`** otherwise. *TAPI enabled*
- **`/relation_schema`** ***`relation_name`*** Display relation schema of ***`relation_name`***. *TAPI enabled*
- **`/sql_left_delimiter`** Display the SQL left delimiter as defined by the current database manager (either DES or the external DBMS via ODBC) . *TAPI enabled*
- **`/sql_right_delimiter`** Display the SQL right delimiter as defined by the current database manager (either DES or the external DBMS via ODBC) . *TAPI enabled*

- New port to SWI-Prolog 5.10.5
- New port to Ciao Prolog 1.14.2

- Changes:
  - Identifier delimiters in output messages have been changed from **`[`** and **`]`** to **`"`**
  - Either by consulting a file, or by dropping the database, or by abolishing the complete database imply to completely reset the database (Datalog rules, tables, views and constraint definitions are removed)
- Fixed bugs:
  - When enabling duplicates without clearing extension table, some duplicates were lost. Bug introduced in version 2.4
  - Datalog queries to ODBC connections failed when involving ground or aliased arguments
  - Some foreign key constraint were not properly checked against database before posting
  - Comma-separated arguments in commands were not always correctly parsed. In particular, this affected to consulting more than one file at a time
  - Command **`/dbschema`** did not always show all integrity constraints
  - Command **`/save_ddb`** incorrectly quoted ending dots in constraints
  - User-defined integrity constraints are syntactically identified (including variable names), therefore avoiding ambiguity for unifiable constraints
  - Several SQL delimited columns in the same statement were not correctly parsed
  - Creating an incorrect SQL view dealt to a table with the same name. Bug introduced in version 2.4

- Most user-defined integrity constraints were not correctly parsed from files
- When restoring the database, not all strong integrity constraints were removed
- Write option in command **`log`** was not parsed

1.8 Version 2.4 of DES (released on July, 6th, 2011)

- Enhancements:
  - Safety and computability revisited for aggregate metapredicates. Most checks are moved to compile-time, covering also the new metapredicate **`distinct`**/2 and equality over evaluable expresssions
  - Added the Datalog tabled metapredicate **`distinct`**/2, which computes distinct outcomes for different values of given arguments and for a given relation. It takes effect when duplicates are enabled via the command **`/duplicates on`**
  - Comparison of expressions including null values are now supported. Two expressions are considered equivalent if they are *syntactically* equal. For instance, **`X=null,X+1=X+1`** succeeds, whereas **`X=null,Y=null,X+1=Y+1`** and **`X=null,X+1=1+X`** do not
  - Syntax error reporting about unbalanced parentheses in Datalog and SQL
  - Syntax error reporting for metapredicate **`group_by`** involving incorrect use of variables in Datalog
  - Simplified error reporting when syntax errors are detected
  - Compilation of Datalog rules keep variable program names for exploded rules (way cool in development mode)
  - Successive applications of **`not/1`** are simplified instead of rewritten
  - Negated calls to primitives are simplified by their complemented counterparts (e.g., **`not(1<0)`** is translated to **`1>=0`**). This in turn avoids the following null-related flaw: **`not(null\=null)`**, which should be semantically equivalent to **`null=null`**
  - New commands:
    - **`/running_info`** Display whether running information (as the incremental number of consulted rules as they are read) is to be displayed
    - **`/running_info`** ***`Switch`*** Enable or disable display of running information (**`on`** or **`off`**, resp.)
    - **`/rm`** ***`FileName`*** Delete ***`FileName`*** from the file system
    - **`/del`** ***`FileName`*** Synonym for **`/rm`**
    - **`/system`** ***`Goal`*** Submit ***`Goal`*** to the underlying Prolog system (implementor's command)
  - Internal null identifiers are reset whenever the database is cleared, and they otherwise start from 0 instead of 1

- Enabling (disabling) flags with commands **/compact_listings**, **/check**, **/development**, **/duplicates**, **/pretty_print**, **/safe**, **/simplification**, and **/verbose** warns should they are already enabled (disabled, resp.)
- New port to GNU-Prolog 1.4.0. Tested successfully for Ubuntu 10.04 and Windows 7
- New version of Windows GUI: ACIDE 0.8 with many improvements

- Changes:
  - Most errors regarding incorrect use of set variables are moved from run-time to compile-time
  - Unknown columns, tables and views are enclosed between double quotes
  - Datalog prompt is restored upon exception when processing a SQL statement
  - Internal representation of Datalog rules. Compiled rules are referenced by its rule identifiers in compilation roots, instead of storing full copies, therefore reclaiming less memory
  - Each rule has attached its textual variable names if they come from user inputs or instead they are automatically generated
  - Showing Datalog compilations on the fly is also controlled by the command **/show_compilations**. Listings of compilations with the command /listing is still controlled by the command **/development**
  - Showing running information is enabled by default. Such information display is not sent to the log, if enabled

- Fixed bugs:
  - Negated, compound calls involving either conjunction or disjunction were not correctly translated. Bug introduced in version 2.3
  - A Datalog 'having' condition with a variable to the right was incorrectly translated
  - Compound expressions including aggregate function **count/0** were rejected
  - Parentheses in arithmetic expressions involving infix operators were not displayed when required
  - The listing command in development mode with pattern **Name/Arity** did not filter by **Arity**
  - Evaluation of an expression containing a null returned a non ground null representation. This, for instance, made **X=null,Y=null,X+1=Y+1** true

1.9 Version 2.3 of DES (released on May, 24th, 2011)

- Enhancements:
  - SQL declarative debugger: Users can debug SQL views from a declarative debugging point-of-view. The system interactively asks questions to the user about relations involved in the computation of the debugged view.

Trusted specifications add semantic references in order to cut the number of questions down

- Added the Datalog tabled metapredicate **`distinct/1`**, which computes distinct outcomes for its argument, discarding duplicates. It takes effect when duplicates are enabled via the command **`/duplicates on`**
- New Datalog functions and predicates for discarding duplicates along aggregation:
  - Aggregate functions: **`count_distinct/1`**, **`count_distinct/2`**, **`avg_distinct/2`**, **`sum_distinct/2`**, and **`times_distinct/2`**
  - Aggregate predicates: **`count_distinct/2`**, **`count_distinct/3`**, **`avg_distinct/3`**, **`sum_distinct/3`**, and **`times_distinct/3`**
- Working **`DISTINCT`** and **`ALL`** keywords in SQL **`SELECT`** and **`UNION`** statements following SQL2 standard
- Working **`DISTINCT`** and **`ALL`** keywords in SQL aggregate functions following SQL2 standard
- Display of SQL statements compilations to Datalog, selectable with the new command **`/show_compilations`**
- Output from shell commands in windows applications are logged
- Compact listings can be enabled so that blank lines in the console output are removed
- New commands:
  - **`/debug_sql`** ***`View`*** **`[`*****`Options`***`]`** Debug a SQL view, optionally specifying whether trusting tables or not, selecting a trust file and selecting a traverse order
  - **`/show_compilations`** Display whether compilations from SQL DQL statements to Datalog rules are to be displayed
  - **`/show_compilations`** ***`Switch`*** Enable or disable display of extended information about compilation of SQL DQL statements to Datalog clauses (**`on`** or **`off`**, resp.)
  - **`/compact_listings`** Display whether compact listings are enabled
  - **`/compact_listings`** ***`Switch`*** Enable or disable compact listings (**`on`** or **`off`**, resp.)
  - **`/nospy`** ***`SPred`*****`[`*****`/Arity`***`]`** Removes the spy point on the given predicate in the host Prolog interpreter (implementor's command)
  - **`/debug`** Set debug mode in the host Prolog interpreter (implementor's command)
- Saving the deductive database to a file also includes constraints (type, existency, primary key, candidate key, functional dependency, foreign key, and user-defined). Restoring a database from file also recovers these constraints

- Added option **`force`** to command **`/save_ddb`**, which avoids asking the user should the file exists already
- Added help on **`output`** without argument
- Datalog predicate symbols and alphanumeric constants containing extended characters do not longer need to be enclosed between quotes
- SQL correlated queries are also allowed in comparison operators
- Escaped single quotes allowed in SQL strings
- SQL syntax error reporting for unknown tables, views and columns. Similar table and view names are provided for the user to choose
- Some run-time errors regarding incorrect use of built-ins (e.g., aggregates) are detected earlier during compilation
- Improved clpfd library for Ciao port, which allows to support type checking (thanks to Rémy Haemmerlé)
- New port to SWI-Prolog 5.10.4

- Changes:
  - SQL natural joins keep the order of columns in the projection list, as usual in RDBMS implementations
- Fixed bugs:
  - Queries involving equality between nulls at the system prompt were incorrect. Bug introduced in version 2.1
  - When requesting help on a keyword which is both a command and a built-in (e.g., **`log`**), only help on command was displayed
  - Some synonyms in the help were not displayed
  - Type error raised when creating SQL views involving **`TIMES`** aggregate function
  - Single quotes inside quoted atoms were allowed
  - Predicate/Arity specifications in commands did not apply for arities greater than 9
  - Removing the compilation of nested applications of outer joins in SQL left some unremoved rules (when submitting SQL queries and dropping views)
  - Some safety warnings during compilation of some aggregate predicates were raised, even avoiding to solve some safe queries
  - After executing an INSERT, DELETE and UPDATE involving other (nested) SQL statements, the extension table was not cleared (so, an answer entry might occur in the table after finishing the modification statement)
  - Some incorrect SQL statements involving unexistent columns were not rejected

1.10 Version 2.2 of DES (released on March, 24th, 2011)

- Enhancements:
  - Type constraints can be imposed and checked on intensional predicates, not only on extensional ones
  - Improved type inference precision
  - Propositional relations can also be typed
  - Datalog type **`char`** added
  - Added alternative syntax for Datalog type constraints:

    `:- type(pred(col`*`1`*`:type`*`1`*`,...,col`*`n`*`:type`*`n`*`))` and

    `:- type(pred(type`*`1`*`,...,col`*`n`*`))`
  - Added Oracle predefined **`'dual'`** table
  - Added **`FROM`**-less SQL **`SELECT`** statements
  - Help system refactoring
  - New commands:
    - **`/help`** ***`Keyword`*** Display detailed help about command ***`Keyword`***
    - **`/apropos`** ***`Keyword`*** Synonym for **`/help`** ***`Keyword`***
    - **`/prolog_system`** Display the underlying Prolog engine version
  - Formatted ODBC error messages
  - From the Datalog input, **`ALTER`**, **`USE`** and **`CREATE TABLE`** SQL statements are also automatically sent to the opened ODBC datatabase, if a connection is already opened
  - Added warning for undeclared predicates occurring in basic queries (i.e., those predicates which have not been provided with either a defining rule or a type declaration)
  - When executing a query in development mode, its compilation is displayed
  - Multi-line remarks are allowed at the system prompt
  - Developer commands are now available in Ciao source distribution
  - New port to Ciao Prolog 1.13.0. Includes unreleased all-new **`clpfd`** library. This port replaces the old port to version 1.10p5
  - New port to SICStus Prolog 4.2.0, which includes enhanced and fixed ODBC library. Former limitations of DES w.r.t. this port are removed
  - In the context of an opened ODBC connection, predicate dependency graph and stratification are computed from the relational database schema, instead of querying each table and view. Computing the deductive database part does not change
  - More robust handling of ODBC exceptions
  - **`SIGINT`** interrupt signal is caught in SWI-Prolog version so that users can now interrupt DES (Ctrl-C usual keyboard interrupt)

- SQL Server ODBC connections tested on spatial databases

- Changes:
  - Built-in Prolog DCG expansion replaced with an explicit translation, which can now be found in **`des_dcg.pl`**. This file is an adaptation of Ciao's **`dcg_expansion.pl`**. It works with all supported Prolog systems but GNU Prolog 1.3.1, which does not provide term expansion
  - A singleton anonymous variable denoted by an underscore in listings is displayed with an underscore. Up to version 2.1, it was given a name with letters, starting with **`A`**
  - Instantiation error exceptions coming from code implementing DES are now displayed (only useful for DES implementors)
  - Some program simplifications related to equalities have been omitted for the sake of type inferencing. This involves different source-to-source translations during query evaluation
  - Built-in is/2 is translated into =/2 at compilation-time when its right argument is already evaluated
  - Removed input error message from attempting to add types in the context of an opened ODBC connection

- Fixed bugs:
  - Unsafety was not reported for anonymous variables (as, e.g., asserting the rule **`p(_)`**)
  - Negation involving an aggregate or outer join predicate with atom syntactic form was not transformed (e.g., **`not(count(p,0))`**)
  - Changing some system flags in no verbose mode displayed an info message
  - Datalog types **`char(N)`** and **`varchar(N)`** were not parsed
  - Some Datalog queries with duplicates enabled and an opened ODBC connection were incomplete in some special cases where rule identifiers collided
  - Log recording upon exceptions repeated previous lines in some keyboard inputs
  - Predicate dependency graph and stratification was not computed when closing an ODBC connection
  - Some calls to predicates resulting from translating SQL queries involving disjunctions were incorrectly built, as in **`select * from t where a=1 or b=1`**

1.11 Version 2.1 of DES (released on November, 30th, 2010)

- Enhancements:
  - Access to Datalog relations from SQL statements. To this end, type declarations are provided to allow both give types and names to relation columns

- Datalog (strong) integrity constraints: type, primary key, foreign key, functional dependencies and user-defined integrity constraints
- SQL statements can be directly submitted from the Datalog prompt
- Revised compilation of SQL views to Datalog rules, avoiding unnecesary intermediate relations
- Enhanced performance: Built-in operators (**`is`**, **`<`**, **`>`**, ...) do not longer rely on the extension table mechanism, speeding computations up to ten times (cf. **`fib(1000,X)`** in **`fib.dl`**)
- Negation can be applied to compound goals
- Displaying of the number of consulted rules
- Formatted Datalog syntax errors (error text, file name, line and column)
- Updated manual
- Output from the command **`/builtins`** rearranged in a way similar to **`/help`**
- User inputs with trailing blanks after the ending optional dot are now accepted
- Now, string type constraints limit the length of strings as specified in their declarations (e.g., **`char(1)`** does not permit strings of length more than 1). Working but in Ciao Prolog source distribution
- Consulted Datalog files can contain multi-line remarks enclosed between **`/*`** and **`*/`**
- Reworked shell command. Outtput and error streams are redirected to the window application in MS Windows distros (this applies to GNU Prolog, SICStus Prolog and SWI-Prolog)
- New commands:
  - **`/cat`** ***`Filename`*** Type the contents of Filename. Also, the synonym **`/type`** ***`Filename`*** is provided
  - **`/check`** Display whether integrity constraint checking is enabled
  - **`/check [`*`Switch`*`]`** Enable or disable integrity constraint checking (**`on`** or **`off`**, resp.)
  - **`/spy`** ***`Pred`*****`[`**`/`*****`Arity`*****`]`** Set a spy point on the given predicate in the host Prolog interpreter (command intended for implementors, not users)
  - **`/nospyall`** Remove all Prolog spy points (command intended for implementors, not users)
  - **`/t`** Terminate the current DES session without halting the host Prolog system (command intended for implementors, not users)

- Changes:
  - For ODBC connections, the table **`db_schema`**, which is automatically created to have access to table and view names, was hiden in SICStus Prolog source version

  - Consulting an incorrect line in a Datalog program does not halt from reading subsequent files, if any

- Fixed bugs:
  - Empty strings were not displayed between single quotes
  - The empty constant (`''`) was rejected in Datalog rules
  - SICStus Prolog source version failed in retrieving tuples with Datalog queries in ODBC connections
  - Exiting without closing an ODBC connection raised an exception
  - Duplicated answers containing null values were not removed with duplicates disabled. Bug introduced in version 2.0
  - Character inputs were not displayed during batch processing as, e.g., answering single-character inputs ('**y**', '**n**', ...)
  - Ciao Prolog source distribution displayed incorrect paths when listing contents of directories containing '`..`'
  - When duplicates were enabled, some recursive rules did not provide answers, as **`fib(3)`** in **`fib.dl`**
  - Windows application entered a loop when closing the window with the white-crossed red button
  - DES could not be interrupted via Ctrl-C in the Ciao source distribution
  - Bitwise disjunction and conjunction were not correctly parsed

1.12 Version 2.0.1 of DES (released on September, 13th, 22nd, and October 7th, 2010)

- Enhancements:
  - DES 2.0.1 executable for Mac OS X Leopard (32bit, Intelx86).
  - DES 2.0.1 patches to SWI source distributions in order to support SWI-Prolog 5.10.1
- Fixed bugs:
  - DES 2.0.1 SICStus source version patched Datalog queries against ODBC connections

1.13 Version 2.0 of DES (released on August, 31st, 2010)

- Enhancements:
  - Connection to RDBs via ODBC connections (DSN providers as MySQL, MS Access, Oracle, ...) RDB tables and views can be queried both from SQL and Datalog
  - Duplicates are allowed in answers, both for Datalog and SQL
  - Datalog and SQL tracers
  - New commands:

- **`/open_db` *`Name`* `[`*`Options`*`]`** Open and set the current ODBC connection to *`Name`*, where *`Options`*=[**`user(`***`Username`***`)`**] [**`password(`***`Password`***`)`**]. This connection must be already defined at the OS layer
- **`/close_db`** Close the current ODBC connection
- **`/current_db`** Display the current ODBC connection name and DSN provider
- **`/duplicates`** Display whether duplicates are enabled
- **`/duplicates`** ***`Switch`*** Enable or disable duplicates (**`on`** or **`off`**, resp.)
- **`/trace_sql`** ***`View`*** **`[`*`Order`*`]`** Trace a SQL view in the given order (**`postorder`** or the default **`preorder`**)
- **`/trace_datalog`** ***`Goal`*** **`[`*`Order`*`]`** Trace a Datalog basic goal in the given order (**`postorder`** or the default **`preorder`**)
- **`/output`** ***`Switch`*** Enable or disable display output (**`on`** or **`off`**, resp.)
- **`/save_ddb`** ***`Filename`*** Save the current Datalog database to a file
- **`/restore_ddb`** ***`Filename`*** Restore the Datalog database in the given file (same as **`consult`**)

o Results from **`SELECT SQL`** statements (those sent to an ODBC connection) can contain duplicates

o Added **`UPDATE`** SQL statement

o Added **`varchar2`** Oracle SQL datatype

o Remarks can now start with '**`--`**', as in Oracle SQL

o Both **`EXCEPT`** and **`MINUS`** are allowed to express SQL set difference

o SQL user identifiers can be enclosed between quotation marks (either double quotes **`""`**, or square brackets **`[]`**, or backquotes ``` `` ```)

o Closing the opened log file, if any, on quitting

o Added timing information to SQL query processing, including listings which may include view processing from RDBs

o Some dead code removal

- Changes:
  - o New port to SICStus 4.*x*, which replaces the old port to SICStus 3.*x*
  - o The command **`debug`** is renamed as **`debug_datalog`**
  - o Executables have been built with SWI-Prolog, instead of SICStus Prolog
- Fixed bugs:
  - o Asserting rules with a number as atom/string changed the type to number, as in **`/assert t('1')`**, which asserted **`t(1)`** instead of **`t('1')`**
  - o Disjunctions in aggregate goals might lead to missing answers, as in **`group_by((p(X,Y),(Y=a;Y=b)),[X],C=count)`**

- Some infix builtins were accepted without delimiting blanks, as **`Xis1`**, posed as a goal, and interpreted as **`X is 1`**

- Caveats and limitations:
  - See Section 10 of the user manual
- Known bugs:
  - The projection list of a natural outer join is not correct in all cases
  - Disjunctions in having conditions in the **`group_by`** clause may display errors which are not
  - Operator precedence in SQL conditions are not correctly handled. Use parentheses to ensure correct operator applications

1.14 Version 1.8.1 of DES (released on March, 17th, 2010)

- Fixed bugs:
  - The Windows and Linux executable distributions lacked some libraries regarding test case generation, which have been added in the current distributions
- Caveats and limitations:
  - See Section 10 of the user manual
- Known bugs:
  - The projection list of a natural outer join is not correct in all cases
  - Disjunctions in having conditions in the **`group_by`** clause may display errors which are not

1.15 Version 1.8.0 of DES (released on December, 18th, 2009)

- Enhancements:
  - An advanced test case generator supporting positive-negative, positive and negative test cases for views, ranging over integer and string data types
  - New command:
    - **`/tc_size`** ***`Min Max`*** Sets the minimum and maximum number of tuples generated for a test case
  - New use for existing command:
    - **`/test_case`** ***`View [Options]`*** Generates test case classes for the view ***`View`***. ***`Options`*** may include a class and/or an action parameters. The test case class is indicated by the values **`all`** (positive-negative), **`positive`**, or **`negative`** in the class parameter. The action is indicated by the values **`display`** (only display tuples), **`replace`** (replace contents of the involved tables by the computed test case), or **`add`** (add the computed test case to the contents of the involved tables) in the action parameter. Default parameters are **`all`** and **`display`**

- More precise type inferring system
- Enhanced syntax error reporting when consulting Datalog programs. An offending rule which is a valid term but is not a valid Datalog rule is listed together with location information
- Enhanced pretty-print:
  - Rules: disjunctive bodies and quoted constants
  - SQL: indentation
  - **`/dbschema`**: bullets and expanded indentation
- Informing that a goal cannot be debugged when its predicate is not defined
- New switch for existing command:
  - **`/timing detailed`** Displays detailed elapsed time (parsing, computation, display and total elapsed times)
- Line number information of consulted files is available also for the source distributions of both Ciao and SWI Prolog

- Changes:
  - The displayed integer type for tables and views has changed from **`int`** to **`integer`**
  - Any sequence of characters enclosed between quotes are allowed as a constant, as **`'2*3'`**
  - A bit more precise verbose output messages
- Fixed bugs:
  - Select statements with empty relations and **`group_by`** gave incorrect results
  - Translations of disjunctions in **`group_by`** conditions involving shared variables were incorrect
  - Some output displays were not logged via the command **`/log`**
  - Rule retraction may behave incorrectly when compiled rules cannot be differentiated
  - When a set of tables were dropped, their foreign keys were not
  - A renaming in the projection list of a SQL statement with the same identifier as input relations was incorrectly translated
  - Dropping and recreating a view failed to delete the defining Datalog rules for the rule, raising a warning
  - Removed meaningless warning message when redefining a table
  - Consulting a datalog program with syntax errors when safety is enabled yielded a loop
  - When asserting a rule and simplification enabled, the correct variable names where not displayed in the translation in some cases
- Caveats and limitations:

- See Section 10 of the user manual

- Known bugs:
  - The projection list of a natural outer join is not correct in all cases
  - Disjunctions in having conditions in the **group_by** clause may display errors which are not

1.16 Version 1.7.0 of DES (released on October, 30th, 2009)

- Enhancements:
  - Extended SQL grammar and processor to cope with types as well as table and column constraints (primary key and foreign key)
  - Type system for SQL. Primitive types include: **char**, **char(n)**, **varchar(n)**, **varchar**, **string**, **int**, **integer**, and **real**
  - Basic type checking/inferring system for SQL views. Inferred types for views are displayed via **/dbschema** and, for autoviews, in the answer relation. Inferring precision is low (the types of expressions and numbers are not inferred)
  - Domain, primary key, and referential integrity constraints for tables created with SQL statements
  - Datalog aggregate predicates: **group_by/3**, **count/3**, **count/2**, **sum/3**, **times/3**, **avg/3**, **min/3**, and **max/3**
  - Datalog aggregate functions: **count/0**, **count/1**, **sum/1**, **times/1**, **avg/1**, **min/1**, and **max/1**
  - Datalog predicate builtins: **is_null/1** and **is_not_null/1** for determining whether their single argument is a null value or not, respectively
  - Test case generation for views
  - New commands:
    - **/test_case *View*** Generates all test case classes of for the given view
    - **/p *Filename*** Shorthand for **/process *Filename***
  - Upgraded commands:
    - **/listing *Head*** Lists all rules whose heads are subsumed by ***Head***
    - **/listing *Head:-Body*** Lists all rules that are subsumed by ***Head:-Body***
  - The command **process** looks for its input filename, allowing to omit the extensions **.sql** and **.ini**
  - Comparison operators can include arithmetic expressions, as in **A<2*B**. This also means that equality behaves more generally than **is/2**, as shown in the query **sqrt(2)=X**, which returns **{ sqrt(2) = 1.4142135623730951 }**
  - Some arithmetic expressions are precomputed when translating SQL statements to Datalog rules

- Displaying the number of tuples in rule listings, retracts, and abolishes
- Adding development flag status to the listing of **`/status`**

- Changes:
  - A table definition with a CREATE TABLE statement must include a type for each attribute. Former table definitions (up to version 1.6.2) are no longer valid
  - Evaluation of an arithmetic expression including a null value returns a null, instead of raising an exception
  - Operands of comparison operators are evaluated. Only arithmetic expressions are allowed, up to now. So, **`X=Y+2`** is allowed whenever Y is bound
  - The distribution files **`des1.pl`**, **`dessql.pl`**, and **`desdebug.pl`** have been renamed to **`des_glue.pl`**, **`des_sql.pl`**, and **`des_debug.pl`**, respectively
- Fixed bugs:
  - Development listings via **`/dbschema`** were not displaying compiled Datalog rules
  - String constants including only digits were incorrectly parsed as numbers
  - Failed to parse SQL set statements involving constants in the projection list
  - Nulls were not correctly read from files
  - **`IS NULL`** and **`IS NOT NULL`** in SQL statements were not behaving correctly
  - Safety checks involving disjunctions were not always properly performed, as in **`p(X) :- q(X);r(X)`**
  - The command **`/operators`** was never implemented but listed via **`/help`**. It has been removed
  - Listings of exploded rules were not displaying the correct source variable names in bodies
  - Some rules could not been asserted under simplification, as **`p(X):-X=1;X=2`**
  - Error when a multiply renamed table occurs in a SQL statement, as in **`select * from t t1,t t2 where t1.a=t2.a`**
- Caveat:
  - Batch processing cannot be nested
- Known bugs:
  - The projection list of a natural outer join is not correct in all cases
  - Disjunctions in having conditions in the **`group_by`** clause may yield to errors which are not

1.17 Version 1.6.2 (released on March, 10th, 2009)

- Enhancements:

- Null values has been included both for Datalog programs and SQL statements
- Novel outer join Datalog functions: **`lj/3`**, **`rj/3`**, and **`fj/3`**
- Outer join SQL clauses added: **`LEFT [OUTER] JOIN`**, **`RIGHT [OUTER] JOIN`**, and **`FULL [OUTER] JOIN`**
- Solving algorithm enhanced for stratified queries. Partial recomputations of lower-stratum predicates are avoided
- Compilation of SQL **`WHERE`** conditions to Datalog rules now provides shorter and more efficient programs
- Disjunctions in Datalog rule bodies
- New commands:
  - **`/development`** ***`Switch`*** Enables/Disables development listings. These listings show the source-to-source translations needed to handle null values, Datalog outer join built-ins, and disjunctive literals
  - **`/development`** Displays whether development listings are enabled
  - **`/simplification`** ***`Switch`*** Enables/Disables simplification of Datalog rules. Rules with equalities, **`true`**, and **`not(`*`BooleanValue`*`)`** are simplified
  - **`/simplification`** Displays whether rule simplification is enabled
- **`WHERE`** conditions accept arithmetic expressions (e.g., **`1+t.a>3`**)
- Display of the number of undefined computed tuples, and the number of tuples in the extension table answer and call sets
- Parentheses in Datalog rule bodies, not only in arithmetic expressions, are allowed
- Parentheresed listings of Datalog rule bodies, making more readable bodies with conjunctions and disjunctions
- Simplification of rules containing equalities

- Changes:
  - Rule listings are grouped by predicate name and arity. For a given predicate name and arity, facts come first, followed by rules with right hand sides. The order of facts and rules follows Prolog standard order between terms
  - Datalog rules resulting from translating views change the naming convention to (the more readable) *ViewName_Arity_Number* in lieu of *ViewName***`$p`***Number*
  - Results from Datalog autoviews are given the relation name **`answer`** instead of **`autoview`**
  - Pretty-print is applied to all Datalog rule listings
  - Safety warnings are not hidden by computability warnings
- Fixed bugs:

- Unformatted SQL statement display for certain conditions and joins
- Parsing error for **EXISTS** clause (no blanks between **EXISTS** and opening parenthesis were allowed)
- SQL arithmetic functions could only be written in lowercase
- Some **WHERE** conditions incorrectly translated into Datalog conditions (bug introduced in version 1.6.1)
- Some **WHERE** conditions involving parentheses incorrectly parsed
- Correlated SQL queries with non-basic conditions were incorrectly translated into Datalog rules
- **DELETE** SQL statements failed to be parsed (copy-paste bug introduced in version 1.6.1)
- Some unsafe Datalog queries were not rejected for computation (as **X=Y**)
- During startup batch processing of **des.ini**, some tasks upon exceptions were not performed
- Typing **des.** in a Prolog interpreter after abnormally quitting the system did not result in exception catching anymore
- A class of unsafe rules was not be able to be preprocessed for reordering body goals, yielding non-termination
- Incomplete error message

- Known bugs:
  - The projection list of a natural outer join is not correct in all cases
- Caveat:
  - Computable SQL statements follow the grammar in the manual. The current grammar parses extra clauses which cannot be computed yet (e.g., **ORDER BY**, ...)

1.18 Version 1.6.1 (released on November, 10th, 2008)

- Enhancements:
  - Arithmetic expressions are allowed in the projection list of **SELECT** statements
  - Subqueries in comparisons (**=**, **<**, **>**, ...), in either side or even in both sides of the comparison operator (read as **ANY**, not **ALL**, which is unsupported up to now)
  - Display of the number of computed, inserted and deleted tuples
  - Commands are case-insensitive
  - Some tweaks on the SQL parsing code for making it hopefully more understandable and efficient
  - The answer to a SQL query is a relation with name **'answer'**, and its schema is displayed when solving it

- A new use for the **`/dbschema`** command: Now, it accepts an optional argument (a database object, which can be a view or a table name) for restricting the displayed schema
- The **`/dbschema`** command informs about local view definitions for each view
- A new SQL DDL statement: **`drop database`**, which drops the database (including tables, views, and rules)
- Stratifications are not computed during building a view that involves local views. As a consequence, several messages are suppressed (as 'undefined' and 'non stratifiable')

- Changes:
  - Inserted and deleted tuples are not shown
- Fixed bugs:
  - Complex left-hand-side relations in joins failed to be parsed
  - Conjunctive Prolog goals failed to be parsed (bug introduced in version 1.6.0)
  - Natural joins now return common attributes only once
  - Datalog rules involving expressions with (prefix) unary operators were incorrectly displayed as infix
  - Parsing of Datalog bodies failed is some situations were arithmetic operators were involved (as in **`/assert p(X) :- X is -(1)`**)
  - Parsing of projection lists failed in some situations where ***`table`*`.*`** was intermixed with references to single table attributes
  - Program transformation for obtaining safe rules yielded incorrect results in some cases
  - When dropping a view, its local view definitions (if any) were not dropped as well
  - Different views could define the same local view name
  - **`/listing`** ***`Name`*** failed to list rules of different arities (bug introduced in version 1.6.0)

1.19 Version 1.6.0 (released on July, 28th, 2008)

- Enhancements:
  - SQL query language added to the system: DDL (Data Definition Language), DML (Data Manipulation Language), and DQL (Data Query Language)
  - Common database for different query languages. Relations defined via SQL or via Datalog can be interchangeably accessed by queries in any language
  - Pretty-print listings for Datalog programs and SQL statements
  - Processing of batch files via the new command **`/process`** ***`File`***

- Display of 'File not found' errors
- Lexicographically ordered listings
- New commands:
  - **`/datalog`** Switches to Datalog interpreter
  - **`/datalog`** ***`Query`*** Executes a Datalog query
  - **`/prolog`** Switches to Prolog interpreter
  - **`/sql`** Switches to SQL interpreter
  - **`/sql`** ***`Query`*** Executes a SQL query
  - **`/dbschema`** Displays the database schema
  - **`/pretty_print`** Displays whether pretty print for listings is enabled
  - **`/pretty_print`** ***`Switch`*** Enables/Disables pretty print
  - **`/process`** ***`File`*** Processes the contents of File as if they were typed at the system prompt

- Changes:
  - Changed some output formatting for the debugger
  - Some tweaks on system messages, mainly referring to safety/ computability
  - Initial status: Program transformation and time display are disabled by default
  - System status is listed at start-up
  - Listings of Datalog rules are ordered
- Fixed bugs:
  - The debugger in SICStus-based releases yielded incorrect results
  - Asserting/Consulting some unsafe clauses without program transformation yielded failure, raising an input error /failing to consult

1.20 Version 1.5.0 (released on December, 30th, 2007)

- Enhancements:
  - A more fine-grained debugging as long as individual clauses can be inspected
  - Warning and error messages provided for:
    - Undefined predicates which are called by rules each time the database is changed
    - Unsafe rules
    - Execution exceptions known at compile-time
  - Exception messages provided for:

        - Execution exceptions unknown at compile-time
    - Rule transformation for allowing computation of safe rules which may raise run-time exceptions due to built-ins
    - Rejection of unsafe or uncomputable queries, views and autoviews
    - Catching of instantiation errors
    - Rule source annotated for debugging and informative errors, i.e., file and lines in the program (if consulted) or assertion time (if manually asserted)
    - Elapsed time display
    - New basic, simpler (although less efficient than the already implemented) algorithm for computing stratified negation, following [SD91]
    - Fresh variables are given new variable names instead of numbers
    - New commands:
        - **`/negation`** Displays the selected algorithm for solving negation
        - **`/negation`** ***`Algorithm`*** Sets the required Algorithm for solving negation (**`strata`** or **`et_not`**)
        - **`/timing`** Displays whether elapsed time display is enabled
        - **`/timing`** ***`Switch`*** Enables or disables elapsed time display (**`on`** or **`off`**, resp.)
        - **`/safe`** Displays whether program transformation is enabled
        - **`/safe`** ***`Switch`*** Enables or disables program transformation (**`on`** or **`off`**, resp.)
    - Changed commands:
        - **`/verbose`** Displays whether verbose output is enabled
        - **`/verbose`** ***`Switch`*** Enables or disables verbose output messages (**`on`** or **`off`**, resp.)
    - Deprecated commands:
        - **`/noverbose`**
    - Slight modifications on existing commands:
        - **`/debug`** ***`Goal Level`*** The inspection level can be set with the second optional argument with **`p`** for predicate level and **`c`** for clause level
        - **`/status`** Now, it also displays the selected algorithm for negation and whether program transformation is enabled
        - **`/version`** For matching the 'standard' display
    - New examples added to the directory **`examples`**
    - The Prolog database corresponding to the Datalog loaded programs has been discarded, therefore using only one representation for them
    - Revised and upgraded user's manual
- Changes:

- Inequality built-ins cause an error and stops execution whenever they are computed with any non-ground argument (formerly, they silently failed)

- Fixed bugs:
  - The Linux version did not work. Now, it has been fixed and tested on Ubuntu 6.10, Kubuntu 7.04 (Feisty), and Mandriva Linux 2007 Spring
  - The parser did not detect that the argument of **`not`** could be a variable
  - Name clashes when loading programs and asserting rules are avoided

1.21 Version 1.4.0 (released on September, 2nd, 2007)

- Enhancements:
  - Arithmetic has been added. The infix builtin 'is' allows the evaluation of arithmetic expressions
  - Arithmetic operators:
    - `\` Bitwise negation
    - `-` Negative value of its single argument
    - `**` Power
    - `^` Synonym for power
    - `*` Multiplication
    - `/` Real division
    - `+` Addition
    - `-` Subtraction
    - `//` Integer quotient
    - **`rem`** Integer remainder
    - `\/` Bitwise disjunction between integers
    - `#` Bitwise exclusive or between integers
    - `/\` Bitwise conjunction between integers
    - `<<` Shift left the first argument the number of places indicated by the second one
    - `>>` Shift right the first argument the number of places indicated by the second one
  - Arithmetic functions:
    - **`sqrt`** Square root
    - **`log`** Natural logarithm of its single argument
    - **`ln`** Synonym for **`log/1`**
    - **`log`** Logarithm of the second argument in the base of the first one
    - **`sin`** Sine

- **cos** Cosine
- **tan** Tangent
- **cot** Cotangent
- **asin** Arc sine
- **acos** Arc cosine
- **atan** Arc tangent
- **acot** Arc cotangent
- **abs** Absolute value
- **float** Float value of its argument
- **integer** Closest integer between 0 and its argument
- **sign** Returns -1 if its argument is negative, 0 otherwise
- **gcd** Greatest common divisor
- **min** Least of two numbers
- **max** Greatest of two numbers
- **truncate** Integer part as a float
- **float_integer_part(*X*)** Integer part as a float
- **float_fractional_part(*X*)** Fractional part as a float
- **round** Closest integer
- **floor** Greatest integer less or equal to its argument
- **ceiling** Least integer greater or equal to its argument

o Arithmetic constants:

- **pi** Archimedes' constant
- **e** Euler's number

o Scientific notation supported

o Autoviews (automatic temporary views) for conjunctive queries on the fly

o Parsing of programs, queries, and asserted rules

o New command:

- **/status** Displays the current status of the system

o Output from the command **/builtins** has been rearranged

o Upgraded input error message

o Prolog goals (submitted via the command **/prolog**) can be conjunctive goals

o Revised and upgraded user's manual

o Revised and homogeneized input processing

- Line comments (starting with `%`) are allowed as prompt inputs (useful for commenting lines in batch files)
- File and path names enclosed between single quotes for error reporting in OS commands (therefore clarifying misusing of blanks)

- Fixed bugs:
  - Underscores in variables were incorrectly parsed
  - Asserted rules had missing program variable names
  - The output stream was not flushed when prompting user input in the debugger and when prompting new Prolog solutions using **`/prolog`**
  - File and directory names as numbers threw an exception in OS commands
  - Incorrect goal when abolishing no rules
  - Some commands did not admit blanks between arguments
  - Fixed some disarranged displays
  - Batch processing tried to open both **`.ini`** and **`.pl`** files
  - Dangling choice points in several places
  - Anonymous variables were incorrectly parsed
  - Debugging was not possible during batch processing

1.22 Version 1.3.0 (released on May, 2nd, 2007)

- Enhancements:
  - Declarative debugger
- Fixed bugs:
  - The output stream was not flushed before waiting the user input. This presented a connection problem with the configurable IDE ACIDE (See Contributions)
- Contributions:
  - ACIDE (A Configurable Development Environment). Authors: Diego Cardiel Freire, Juan José Ortiz Sánchez, and Delfín Rupérez Cañas, leaded by Fernando Sáenz. 3/2007. Description: This project is aimed to provide a multiplatform configurable integrated development environment which can be configured in order to be used with any development system such as interpreters, compilers and database systems. Features of this system include: project management, multifile editing, syntax colouring, and parsing on-the-fly (which informs of syntax errors when editing programs prior to the compilation). Status: alpha.
  - Emacs development environment. Author: Markus Triska. 22/2/2007. Description: Provides an integration of DES into Emacs. Once a Datalog file has been opened, you can consult it by pressing F1 and submit queries and commands from Emacs.

1.23 Version 1.2.0 (released on February, 9th, 2007)

- Enhancements:
  - Solving-by-stratum algorithm
  - Temporary views, which allow to write a temporary rule whose head is solved as a query
  - Program variable names are kept to allow more readable program listings
  - Syntax error reports when loading programs in standalone applications and SICStus source distribution
  - Handling and reporting of Prolog exceptions in standalone applications, SWI and SICStus source distribution
  - New commands:
    - **`/verbose`** (default option) for verbose output
    - **`/noverbose`** for abbreviated messages
    - **`/strata`** displays the current stratification
    - **`/pdg`** displays the current predicate dependency graph
    - **`/dir`** synonym of **`/ls`**
    - **`/log`** ***`FileName`*** sets the current log to ***`FileName`***
    - **`/log`** displays the current log file, if any
    - **`/nolog`** disables logging
    - **`/version`** for displaying the current system version
  - New uses for existing command: **`/abolish`** ***`Name`***, and **`/abolish`** ***`Name/Arity`***
  - Batch processing
  - Rearranged and revised help
  - Reworked command and query-related messages
  - Consulting/Reconsulting files avoids duplicates
  - Added examples
- Fixed bugs:
  - Loading an incorrect Datalog program exited standalone applications (**`des.exe`** and **`deswin.exe`** applications)
  - Evaluating Prolog goals via /prolog failed for programs containing negation
  - For several commands, blanks between a command and its arguments were not consumed but the first one
  - Non existent directory errors were not caught in command **`/ls`**

1.24 Version 1.1.2 (released on December, 20th, 2006)

- Enhancements:
  - New uses for existing commands: **`/list_et`** ***`Name`***, **`/listing`** ***`Name`***
- Fixed bugs:
  - The commands **`/list_et`** and **`/clear_et`** were not properly parsed
  - Infix operators allowed a variable argument

1.25 Version 1.1.1 (released on February, 21st, 2005)

- A new executable version for Linux
- Enhancements:
  - Atoms can contain blanks
- Fixed bugs:
  - When using /prolog, DES1.1 did not find predicates defined without facts.

1.26 Version 1.1 (released on March, 4th, 2004)

- Full recursion
- Memoization techniques
- Gathering of undefined facts under non stratified programs (incomplete algorithm)
- Several new commands:
  - **`/listing`** ***`Name/Arity`***. Lists Datalog rules matching the pattern
  - **`/retractall`** ***`Head`***. Deletes all Datalog rules matching head
  - **`/list_et`**. Lists contents of the extension table
  - **`/list_et`** ***`Name/Arity`***. Lists contents of the extension table matching the pattern
  - **`/clear_et`**. Clears the extension table
  - **`/builtins`**. Lists built-in operators
  - **`/cd`** ***`Path`***. Sets the current directory
  - **`/cd`**. Sets the current directory to the directory where DES was started from
  - **`/pwd`**. Displays the current directory
  - **`/ls`**. Displays the contents of the current directory
  - **`/ls`** ***`Path`***. Displays the contents of the given absolute or relative directory
  - **`[`*`FileNames`*`]`**. Consults a list of Datalog files abolishing previous rules
  - **`[+`*`FileNames`*`]`**. Consults a list of Datalog files keeping previous rules
  - **`/shell`** ***`Command`***. Submits a command to the operating system shell
- Cosmetic changes:

- Commands start with a slash
- Command arguments are no longer enclosed in brackets
- Both commands and queries may end with a dot

- Fixed bugs:
  - Primitives fail adequately when they should do it, instead of exiting from the interpreter

1.27 Version 1.0 (released on December, 2003)

Version 1.0 of DES, the first public release of the system, featured:

- Naïve Datalog system intended to be complete w.r.t. relational algebra
- Limited support for recursion: Termination is not guaranteed for some recursive programs
- Basic Negation
- Built-in Operators
  - **=** Syntactic equality
  - **\=** Syntactic disequality
  - **>** Greater than
  - **>=** Greater than or equal to
  - **<** Less than
  - **=<** Less than or equal to
  - **not(*Goal*)** Negation
- Commands
  - **consult(*File*)** Loads a Datalog program abolishing current rules
  - **reconsult(*File*)** Loads a Datalog program keeping current rules
  - **assert((*Head*:-*Body*))** Asserts a new rule
  - **retract((*Head*:-*Body*))** Retracs a rule
  - **abolish** Abolishes the loaded program
  - **listing** Lists the loaded rules.
  - **prolog(*Goal*)** SLD execution of Goal.
  - **halt** Quits the system
  - **help** Displays the help

Appendix A. Documentation License

GNU Free Documentation License

Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be

listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their

titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with … Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being
LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.