

?
 ?
 ?
 ?
 ?
 applications," sec : enterprise_applications", this is not sufficient for modern applications that handle large amounts of data
 ?
 ?
 ?
 ?
 ?
 ?
 TAR-
 DISP
 TAR-
 DISP
 Back-
 m-
 time
 SQL
 Very
 low-
 over-
 head
 Point
 of
 Sales
 Ex-
 plorer?
 s qlodb Data model of the Stored Procedure debugger.
 o db show the relational data model we use to store the required information. On top of the minimal required data, we record
 s stored procedures.
 Application-
 level
 Log-
 ging

null
 until IS NULL
 on <:
 dbgtime AND (valid_until >=:
 dbgtime OR valid_until IS NULL)
 BIT products(:
 dbgtime) WHERE id =
 17 AND valid_until IS NULL
 BIT products(IN dbgtime TIMESTAMP) RETURN STABLE LANGUAGES SQLSCRIPT AS BEGIN RETURN SEL
 dbgtime THEN valid_until ELSE NULL) AS valid_until FROM products WHERE created_on <:
 dbgtime END;
 Debug-
 level
 Log-
 ging
 ACID
 Trans-
 ac-
 tions
 t ransactions, allows developer to query intermediate points in time.
 t ransactions Nested transactions allow reversing data manipulation queries.
 Insert-
 only
 Databases
 open_orders(IN department VARCHAR(10), OUT orders T_ORDERS) LANGUAGES SQLSCRIPT AS BEGIN DECLAR
 SELECT id FROM departments WHERE name =:
 department;
 id FROM purchase_orders po WHERE po.department_id =:
 department_id AND po.status =
 "open";
 orders(:
 orders); END;
 tracing, firstnumber =
 1, stepnumber =
 5] CREATE PROCEDURE T_payopen_orders(IN t_id INT, IN ce_id INT, IN OUT s_id INT, IN department VARCHAR(10), OUT orders T_ORDERS) AS BEGIN
 id INTEGER :=
 SELECT id FROM departments WHERE name =:
 department; /*
 1*
 / INSERT INTO Tardisp.ControlFlows(trace_id, step, entry_step, type, value, line) VALUES(:t_id, :s_id+1, :ce_id, 'ENTER', 'payopen

p rojects variable, which contains ids, names, and budgets of all projects to be processed. To better understand the data, they also
 p rojects spr JOIN PurchaseOrders po ON po.project_id =
 pr id WHERE po.status = '