Appendix A
Benchmarking Scripts

A.1 Data Warehouse Schema Creation DDLs

A.1.1 Relational

These are the Data Definition Language (DDL) statements used to create the TPC-H database into the Relational Model instantiation (Oracle).

```sql
-- Sccsid:     @(#)dss.ddl 2.1.8.1
CREATE TABLE NATION   (N_NATIONKEY    INTEGER        NOT NULL,
                         N_NAME         CHAR(25)       NOT NULL,
                         N_REGIONKEY    INTEGER        NOT NULL,
                         N_COMMENT      VARCHAR(152));
CREATE TABLE REGION   (R_REGIONKEY    INTEGER        NOT NULL,
                        R_NAME         CHAR(25)       NOT NULL,
                        R_COMMENT      VARCHAR(152));
CREATE TABLE PART     (P_PARTKEY      INTEGER        NOT NULL,
                         P_NAME         VARCHAR(55)    NOT NULL,
                         P_MFGR         CHAR(25)       NOT NULL,
                         P_BRAND        CHAR(10)       NOT NULL,
                         P_TYPE         VARCHAR(25)    NOT NULL,
                         P_SIZE         INTEGER        NOT NULL,
                         P_CONTAINER    CHAR(10)       NOT NULL,
                         P_RETAILPRICE  DECIMAL(15,2)  NOT NULL,
                         P_COMMENT      VARCHAR(23)    NOT NULL);
CREATE TABLE SUPPLIER (S_SUPPKEY      INTEGER        NOT NULL,
                         S_NAME         CHAR(25)       NOT NULL,
                         S_ADDRESS      VARCHAR(40)    NOT NULL,
                         S_NATIONKEY    INTEGER        NOT NULL,
                         S_PHONE        CHAR(15)       NOT NULL,
                         S_ACCTBAL      DECIMAL(15,2)  NOT NULL,
                         S_COMMENT      VARCHAR(101)   NOT NULL);
CREATE TABLE PARTSUPP (PS_PARTKEY     INTEGER        NOT NULL,
                       PS_SUPPKEY     INTEGER        NOT NULL,
                       PS_AVAILQTY    INTEGER        NOT NULL,
                       PS_SUPPLYCOST  DECIMAL(15,2)  NOT NULL,
                       PS_COMMENT     VARCHAR(199)   NOT NULL);
CREATE TABLE CUSTOMER (C_CUSTKEY      INTEGER        NOT NULL,
                        C_NAME         VARCHAR(25)    NOT NULL,
                        C_ADDRESS      VARCHAR(40)    NOT NULL,
                        C_NATIONKEY    INTEGER        NOT NULL,
                        C_PHONE        CHAR(15)       NOT NULL,
                        C_ACCTBAL      DECIMAL(15,2)  NOT NULL,
                        C_MKTSEGMENT   CHAR(10)       NOT NULL,
                        C_COMMENT      VARCHAR(117)   NOT NULL);
CREATE TABLE ORDERS  ( O_ORDERKEY       INTEGER        NOT NULL,
                        O_CUSTKEY        INTEGER        NOT NULL,
                        O_ORDERSTATUS    CHAR(1)        NOT NULL,
                        O_TOTALPRICE     DECIMAL(15,2)  NOT NULL,
                        O_ORDERDATE      DATE           NOT NULL,
                        O_ORDERPRIORITY  CHAR(15)       NOT NULL,
                        O_CLERK          CHAR(15)       NOT NULL,
                        O_SHIPPRIORITY   INTEGER        NOT NULL,
                        O_COMMENT        VARCHAR(79)    NOT NULL);
CREATE TABLE LINEITEM ( L_ORDERKEY       INTEGER       NOT NULL,
                        LLINEITEMKEY      INTEGER        NOT NULL,
                        L_QUANTITY        INTEGER        NOT NULL,
                        L_COMMITTED       DECIMAL(15,2)  NOT NULL,
                        L_SHIPDATE       DATE           NOT NULL,
                        L_SHIPPRIORITY   INTEGER        NOT NULL,
                        L_COMMENT        VARCHAR(23)    NOT NULL);
```

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Nullability</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_PARTKEY</td>
<td>INTEGER</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_SUPPKEY</td>
<td>INTEGER</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_LINENUMBER</td>
<td>INTEGER</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_QUANTITY</td>
<td>DECIMAL(15,2)</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_EXTENDEDPRICE</td>
<td>DECIMAL(15,2)</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_DISCOUNT</td>
<td>DECIMAL(15,2)</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_TAX</td>
<td>DECIMAL(15,2)</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_RETURNFLAG</td>
<td>CHAR(1)</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_LINESTATUS</td>
<td>CHAR(1)</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_SHIPDATE</td>
<td>DATE</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_COMMITDATE</td>
<td>DATE</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_RECEIPTDATE</td>
<td>DATE</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_SHIPINSTRUCT</td>
<td>CHAR(25)</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_SHIPMODE</td>
<td>CHAR(10)</td>
<td>NOT NULL,</td>
</tr>
<tr>
<td>L_COMMENT</td>
<td>VARCHAR(44)</td>
<td>NOT NULL,</td>
</tr>
</tbody>
</table>

*Figure 60 Relational DDLs*
A.1.2 Associative

These are equivalent to the DDL statements used to create the TPC-H database into the Associative/Triple Store Model instantiation (SentencesDB). As this data models is based on relations, it does not use the SQL language statements to create the structures to be used. The logical model of the TPC-H database schema was modelled first and then the actual header files were created in order to give meaning to the flat files loaded into the previously defined logical model. The logical model is presented in Appendix C.

```sql
/* Region */
REGIONKEY, R_NAME:is designated as, R_COMMENT:has a region comment

/* Nation */
NATIONKEY, REGIONKEY:is located in region, N_NAME:has a nation name, N_COMMENT:has a nation comment

/* Part */
PARTKEY, P_NAME:is known as, P_MFGR:is manufactured by, P_BRAND:is branded, P_TYPE:is classified in type, MONEY_AMOUNT:has a retail price, P_SIZE:has size, P_CONTAINER:is packed in, P_COMMENT:has a part comment

/* Supplier */
SUPPKEY, S_NAME:has a supplier name, S_ADDRESS:has a supplier address on, NATIONKEY:has a supplier location, S_PHONE:has a supplier phone on, MONEY_AMOUNT:has a supplier balance of, S_COMMENT:has a supplier comment

/* Part supplier */
SUPPKEY:provides, PARTKEY, MONEY_AMOUNT:has a supply cost of, PS_AVAILQTY:has an availability of, PS_COMMENT:has a part supplier comment

/* Customer */
CUSTKEY, C_NAME:has a customer name, C_ADDRESS:has a customer address on, NATIONKEY:has is nation in, C_PHONE:has a customer phone on, MONEY_AMOUNT:has a customer balance of, C_MKTSEGMENT:is classified in the market segment, C_COMMENT:has a customer comment

/* Orders */
ORDERKEY, O_CLERK:was processed by, CUSTKEY:is put by, DATES:is created in order date of, O_ORDERPRIORITY:has an order priority, O_ORDERSTATUS:has an order status, O_SHIPPPRIORITY:should be shipped in priority, MONEY_AMOUNT:has total price, O_COMMENT:has an order comment

/* Lineitem */
```
LINENUMBER, ORDERKEY: is part of order, DATES: has a commit date on, PERCENT_AMOUNT: applies a percentage discount of, MONEY_AMOUNT: has an extended price, L_LINESTATUS: has a line status of, SUPPKEY: is a selection supply, PARTKEY, L_QUANTITY: supplied in a quantity of, DATES: has a receipt date on, L_RETURNFLAG: has a return flag, DATES: has a ship date on, L_SHIPINSTRUCT: has the shipping instructions, L_SHIPMODE: needs to be shipped in the following mode, PERCENT_AMOUNT: has a tax rate of, L_COMMENT: has a line comment

**Figure 61** Associative DDLs
A.1.3 Binary-Relational

These are the DDL statements used to create the TPC-H database into the Binary-Relational Model instantiation (MonetDB).

```sql
START TRANSACTION;
CREATE TABLE NATION   ( N_NATIONKEY      INTEGER      NOT NULL,
             N_NAME           CHAR(25)     NOT NULL,
             N_REGIONKEY      INTEGER      NOT NULL,
             N_COMMENT        VARCHAR(152));
CREATE TABLE REGION   ( R_REGIONKEY      INTEGER      NOT NULL,
             R_NAME           CHAR(25)     NOT NULL,
             R_COMMENT        VARCHAR(152));
CREATE TABLE PART     ( P_PARTKEY        INTEGER      NOT NULL,
             P_NAME           VARCHAR(55)  NOT NULL,
             P_MFGR           CHAR(25)     NOT NULL,
             P_BRAND          CHAR(10)     NOT NULL,
             P_TYPE           VARCHAR(25)  NOT NULL,
             P_SIZE           INTEGER      NOT NULL,
             P_CONTAINER      CHAR(10)     NOT NULL,
             P_RETAILPRICE    DECIMAL(15,2)NOT NULL,
             P_COMMENT        VARCHAR(23)  NOT NULL);
CREATE TABLE SUPPLIER ( S_SUPPKEY        INTEGER      NOT NULL,
             S_NAME           CHAR(25)     NOT NULL,
             S_ADDRESS        VARCHAR(40)  NOT NULL,
             S_NATIONKEY      INTEGER      NOT NULL,
             S_PHONE          CHAR(15)     NOT NULL,
             S_ACCTBAL        DECIMAL(15,2)NOT NULL,
             S_COMMENT        VARCHAR(101) NOT NULL);
CREATE TABLE PARTSUPP ( PS_PARTKEY       INTEGER      NOT NULL,
             PS_SUPPKEY       INTEGER      NOT NULL,
             PS_AVAILQTY      INTEGER      NOT NULL,
             PS_SUPPLYCOST    DECIMAL(15,2)NOT NULL,
             PS_COMMENT       VARCHAR(199) NOT NULL);
CREATE TABLE CUSTOMER ( C_CUSTKEY        INTEGER      NOT NULL,
             C_NAME           VARCHAR(25)  NOT NULL,
             C_ADDRESS        VARCHAR(40)  NOT NULL,
             C_NATIONKEY      INTEGER      NOT NULL,
             C_PHONE          CHAR(15)     NOT NULL,
             C_ACCTBAL        DECIMAL(15,2)NOT NULL,
             C_MKTSEGMENT     CHAR(10)     NOT NULL,
             C_COMMENT        VARCHAR(117) NOT NULL);
CREATE TABLE ORDERS   ( O_ORDERKEY       INTEGER      NOT NULL,
             O_CUSTKEY        INTEGER      NOT NULL,
             O_ORDERSTATUS    CHAR(1)      NOT NULL,
             O_TOTALPRICE     DECIMAL(15,2)NOT NULL,
             O_ORDERDATE      DATE         NOT NULL,
             O_ORDERPRIORITY  CHAR(15)     NOT NULL,
             O_CLERK          CHAR(15)     NOT NULL,
             O_SHIPPRIORITY   INTEGER      NOT NULL,
             O_COMMENT        VARCHAR(79)  NOT NULL);
CREATE TABLE LINEITEM ( L_ORDERKEY      INTEGER      NOT NULL,
             L_PARTKEY       INTEGER      NOT NULL,
             L_SUPPKEY       INTEGER      NOT NULL,
             L_LINENUMBER    INTEGER      NOT NULL,
             L_QUANTITY      DECIMAL(15,2)NOT NULL,
             L_EXTENDEDPRICE DECIMAL(15,2)NOT NULL,
             L_DISCOUNT      DECIMAL(15,2)NOT NULL,
             L_TAX           DECIMAL(15,2)NOT NULL,
             L_RETURNFLAG    CHAR(1)      NOT NULL,
             L_LINESTATUS    CHAR(1)      NOT NULL,
             L_SHIPDATE      DATE         NOT NULL,
             L_COMMITDATE    DATE         NOT NULL,
             L_RECEIPTDATE   DATE         NOT NULL,
             L_SHIPINSTRUCT  CHAR(25)     NOT NULL,
             L_SHIPMODE      CHAR(10)     NOT NULL,
             L_COMMENT       VARCHAR(44)  NOT NULL);
COMMIT;
```

Figure 62 Binary-Relational DDLs
The Use of Alternative Data Models in Data Warehousing Environments

### A.1.4 Transrelational™

These are the DDL statements used to create the TPC-H database into the Transrelational™ Model instantiation that has been developed. These DDLs are in the syntax required by the SHQL database [SHQL, 2005] used to define the database tables structure.

```sql
create table nation   ( n_rownumber    8,
                      n_nationkey    8,
                      n_name        25,
                      n_regionkey    8,
                      n_comment    152)

create table region   ( r_rownumber    8,
                      r_regionkey    8,
                      r_name        25,
                      r_comment    152)

create table part     ( p_rownumber    8,
                      p_partkey      8,
                      p_name        55,
                      p_mfgr        25,
                      p_brand       10,
                      p_type        25,
                      p_size         8,
                      p_container   10,
                      p_retailprice 15,
                      p_comment     23)

create table supplier ( s_rownumber    8,
                      s_suppkey      8,
                      s_name        25,
                      s_address     40,
                      s_nationkey    8,
                      s_phone       15,
                      s_acctbal     15,
                      s_comment    101)

create table partsupp ( ps_rownumber   8,
                      ps_partkey     8,
                      ps_suppkey     8,
                      ps_availqty    8,
                      ps_supplycost 15,
                      ps_comment   199)

create table customer ( c_rownumber    8,
                      c_custkey      8,
                      c_name        25,
                      c_address     40,
                      c_nationkey    8,
                      c_phone       15,
                      c_acctbal     15,
                      c_mktsegment  10,
                      c_comment    117)
```
create table orders   ( o_rownumber     8,
o_orderkey        8,
o_custkey        8,
o_orderstatus    1,
o_totalprice    15,
o_orderdate     11,
o_orderpriority 15,
o_clerk         15,
o_shippriority   8,
o_comment       79)
/create/g
create table lineitem ( l_rownumber     8,
l_orderkey        8,
l_partkey        8,
l_suppkey        8,
l_linenumber     8,
l_quantity      15,
l_extendedprice 15,
l_discount      15,
l_tax           15,
l_returnflag     1,
l_linenumber     8,
l_shipdate      11,
l_commitdate    11,
l_receiptdate   11,
l_shipinstruct 25,
l_shipmode      10,
l_comment       44)
/create/g

Figure 63 Transrelational DDLs
A.2 Data Extraction and Transformation Scripts

A.2.1 Relational

This program extracts the flat files required by the Relational model instantiation that will be loaded.

```
time sqlplus victor/tpch @extract_region_data_file_relational.sql;
time sqlplus victor/tpch @extract_nation_data_file_relational.sql;
time sqlplus victor/tpch @extract_part_data_file_relational.sql;
time sqlplus victor/tpch @extract_supplier_data_file_relational.sql;
time sqlplus victor/tpch @extract_partsupp_data_file_relational.sql;
time sqlplus victor/tpch @extract_customer_data_file_relational.sql;
time sqlplus victor/tpch @extract_orders_data_file_relational.sql;
time sqlplus victor/tpch @extract_lineitem_data_file_relational.sql;
```

Figure 64 Extract_all_relational.sh

This is a sample program to extract the lineitem data file as required by the relational model instantiation.

```
set lines 1500;set pagesize 0;
set feed off;set space 0;
set trimout on;set trimspool on;
set colsep '|';set TERMOUT off;
sPOOL LINEITEM_DATA;
SELECT TRIM(L_COMMENT)||'|'||
   TRIM(L_COMMITTDATE)||'|'||
   TRIM(L_DISCOUNT)||'|'||
   TRIM(L_EXTENDEDPRICE)||'|'||
   TRIM(L_LINENUMBER)||'|'||
   TRIM(L_LINESTATUS)||'|'||
   TRIM(L_ORDERKEY)||'|'||
   TRIM(L_PARTKEY)||'|'||
   TRIM(L_QUANTITY)||'|'||
   TRIM(L_RECEIPTDATE)||'|'||
   TRIM(L_RETURNFLAG)||'|'||
   TRIM(L_SHIPDATE)||'|'||
   TRIM(L_SHIPINSTRUCT)||'|'||
   TRIM(L_SHIPMODE)||'|'||
   TRIM(L_SUPPKEY)||'|'||
   TRIM(L_TAX)||'

from LINEITEM;
sPOOL off;
```

Figure 65 Extract Lineitem_relational.sql
A.2.2 Associative

This program extracts all the flat files required by the Associative model instantiation.

```
time sqlplus victor/tpch@ORCL @extract_region_data_file_associative.sql;
time sqlplus victor/tpch@ORCL @extract_nation_data_file_associative.sql;
time sqlplus victor/tpch@ORCL @extract_part_data_file_associative.sql;
time sqlplus victor/tpch@ORCL @extract_supplier_data_file_associative.sql;
time sqlplus victor/tpch@ORCL @extract_partsupp_data_file_associative.sql;
time sqlplus victor/tpch@ORCL @extract_customer_data_file_associative.sql;
time sqlplus victor/tpch@ORCL @extract_orders_data_file_associative.sql;
time sqlplus victor/tpch@ORCL @extract_lineitem_data_file_associative.sql;
```

Figure 66 Extract_all_associative.sh

Here is a sample program to extract the lineitem table from the relational instantiation and prepare the loading file as required by the Associative Model instantiation. The first query generates the header as required by the Associative model instantiation (SentencesDB). The second query extracts the data.

```
set pagesize 0;
set feed off;
set space 0;
set trimout on;
set trimspool on;
set colsep '|';
spool LINEITEM_DATA_ASSOCIATIVE.csv;
select
  'LINENUMBER,ORDERKEY:is part of order,DATES:has a commit date on,'\n  'PERCENT_AMOUNT:applies a percentage discount of,'\n  'MONEY_AMOUNT:has an extended price,'\n  'L_LINETYPE:has a line status of,'\n  'SUPPKEY:is a selection supply,'\n  'PARTKEY,L_QUANTITY:supplied in a quantity of,'\n  'DATES:has a receipt date on,'\n  'L_RETURNFLAG:has a return flag,'\n  'DATES:has a shipdate on,'\n  'L_SHIPINSTRUCT:has the shipping instructions,'\n  'L_SHIPMODE:needs to be shipped in the following mode,'\n  'PERCENT_AMOUNT:has a tax rate of,L_COMMENT:has a line comment'\nfrom dual;
select TRIM(L_LINENUMBER)||','||TRIM(L_ORDERKEY)||','||TRIM(L_COMMITDATE)||','||TRIM(L_DISCOUNT)||';'||TRIM(L_RETURNFLAG)||','||TRIM(L_SHIPINSTRUCT)||','||TRIM(L_SHIPMODE)\n```
The Use of Alternative Data Models in Data Warehousing Environments

```
TRIM(L_EXTENDEDPRICE) || ',' ||
TRIM(L_LINESTATUS) || ',' ||
TRIM(L_SUPPKEY) || ',' ||
TRIM(L_PARTKEY) || ',' ||
TRIM(L_QUANTITY) || ',' ||
TRIM(L_RECEIPTDATE) || ',' ||
TRIM(L_RETURNFLAG) || ',' ||
TRIM(L_SHIPDATE) || ',' ||
TRIM(L_SHIPINSTRUCT) || ',' ||
TRIM(L_SHIPMODE) || ',' ||
TRIM(L_TAX) || ',' ||
TRIM(L_COMMENT) || '*'
from LINEITEM;
spool off;
exit;
```

Figure 67 Extract_lineitem_associative.sql
A.2.3 Binary-Relational

This program extracts all the flat files required by the Binary-Relational data model.

```
time sqlplus victor/tpch@ORCL @extract_region_data_file_binary.sql;
time sqlplus victor/tpch@ORCL @extract_nation_data_file_binary.sql;
time sqlplus victor/tpch@ORCL @extract_part_data_file_binary.sql;
time sqlplus victor/tpch@ORCL @extract_supplier_data_file_binary.sql;
time sqlplus victor/tpch@ORCL @extract_partsupp_data_file_binary.sql;
time sqlplus victor/tpch@ORCL @extract_customer_data_file_binary.sql;
time sqlplus victor/tpch@ORCL @extract_orders_data_file_binary.sql;
time sqlplus victor/tpch@ORCL @extract_lineitem_data_file_binary.sql;
```

**Figure 68 Extract_all_binary.sh**

Sample program to extract the lineitem table from the relational database and place the data according to the Binary Relational Model specifications.

```
set lines 1500; set pagesize 0;
set feed off; set space 0;
set trimout on;
set trimspool on;
set colsep '|';
set TERMOUT off;
spool LINEITEM_DATA;
select TRIM(L_COMMENT)||'|'||
    TRIM(L_COMMITDATE)||'|'||
    TRIM(L_DISCOUNT)||'|'||
    TRIM(L_EXTENDEDPRICE)||'|'||
    TRIM(L_LINENUMBER)||'|'||
    TRIM(L_LINESTATUS)||'|'||
    TRIM(L_ORDERKEY)||'|'||
    TRIM(L_PARTKEY)||'|'||
    TRIM(L_QUANTITY)||'|'||
    TRIM(L_RECEIPTDATE)||'|'||
    TRIM(L_RETURNFLAG)||'|'||
    TRIM(L_SHIPDATE)||'|'||
    TRIM(L_SHIPINSTRUCT)||'|'||
    TRIM(L_SHIPMODE)||'|'||
    TRIM(L_SUPPKEY)||'|'||
    TRIM(L_TAX)
from LINEITEM;
spool off;
exit;
```

**Figure 69 Extract_lineitem_binary.sh**
A.2.4 Transrelational™

This script extracts all the data from the Oracle Database and generates the flat files required by the Transrelational instantiation.

```sql
time sqlplus victor/tpch@ORCL @extract_region_data_file_transrel.sql;
time sqlplus victor/tpch@ORCL @extract_nation_data_file_transrel.sql;
time sqlplus victor/tpch@ORCL @extract_part_data_file_transrel.sql;
time sqlplus victor/tpch@ORCL @extract_supplier_data_file_transrel.sql;
time sqlplus victor/tpch@ORCL @extract_partsupp_data_file_transrel.sql;
time sqlplus victor/tpch@ORCL @extract_customer_data_file_transrel.sql;
time sqlplus victor/tpch@ORCL @extract_orders_data_file_transrel.sql;
time sqlplus victor/tpch@ORCL @extract_lineitem_data_file_transrel.sql;

cd $LOADING_DATA_FILES
rename *.lst '' *.lst

cd $TRANSREL_HOME/extraction_scripts_transrel
```

**Figure 70 Extract_all_transrelational.sh**

Here is a sample program to extract the lineitem table from the relational instantiation and generate the flat file required for the Transrelational Instantiation.

```sql
set lines 1500;set pagesize 0;
set feed off;set space 0;
set trimout on;set trimspool on;
set colsep '|';set TERMOUT off;
spool $LOADING_DATA_FILES/lineitem;
select TRIM(ROWNUM) || '' ||
    TRIM(L_ORDERKEY) || '' ||
    TRIM(L_PARTKEY) || '' ||
    TRIM(L_SUPPKEY) || '' ||
    TRIM(L_LINENUMBER) || '' ||
    TRIM(L_QUANTITY) || '' ||
    TRIM(L_EXTENDEDPRICE) || '' ||
    TRIM(L_DISCOUNT) || '' ||
    TRIM(L_TAX) || '' ||
    TRIM(L_RETURNFLAG) || '' ||
    TRIM(L_LINESTATUS) || '' ||
    TRIM(L_SHIPDATE) || '' ||
    TRIM(L_COMMITDATE) || '' ||
    TRIM(L_RECEIPTDATE) || '' ||
    TRIM(L_SHIPINSTRUCT) || '' ||
    TRIM(L_SHIPMODE) || '' ||
    TRIM(L_COMMENT)
from LINEITEM;
spool off;
```

**Figure 71 Extract_lineitem_transrel.sql**
A.3 DATABASE LOAD SCRIPTS

A.3.1 Relational

This program loads the TPC-H database into the Relational model instantiation (Oracle).

```sql
time sqlldr victor/tpch@ORCL control=region.ctl;
time sqlldr victor/tpch@ORCL control=nation.ctl;
time sqlldr victor/tpch@ORCL control=part.ctl;
time sqlldr victor/tpch@ORCL control=supplier.ctl;
time sqlldr victor/tpch@ORCL control=partsupp.ctl;
time sqlldr victor/tpch@ORCL control=customer.ctl;
time sqlldr victor/tpch@ORCL control=orders.ctl;
time sqlldr victor/tpch@ORCL control=lineitem.ctl;
```

**Figure 72 load_all_relational.sh**

This is a sample program to load the lineitem table into Oracle. It is written according to the specifications of the Oracle loader.

```sql
LOAD DATA infile '/oracle/tpch/data10GB/lineitem.tbl.1'
  Badfile '/oracle/tpch/data10GB/lineitem.bad.1'
  discardfile '/oracle/tpch/data10GB/lineitem.disc.1'
REPLACE
  INTO TABLE lineitem FIELDS TERMINATED BY '|' 
    (L_ORDERKEY, 
    L_PARTKEY, 
    L_SUPPKEY, 
    L_LINENUMBER, 
    L_QUANTITY, 
    L_EXTENDEDPRICE, 
    L_DISCOUNT, 
    L_TAX, 
    L_RETURNFLAG, 
    L_LINESTATUS, 
    L_SHIPDATE DATE "YYYY-MM-DD", 
    L_COMMITDATE DATE "YYYY-MM-DD", 
    L_RECEIPTDATE DATE "YYYY-MM-DD", 
    L_SHIPINSTRUCT, 
    L_SHIPMODE, 
    L_COMMENT)
```

**Figure 73 load_lineitem_relational.ctl**
A.3.2 Associative

This is the script to load the TPC-H database into the Associative model instantiation (SentencesDB).

```
date
./load_region_nation.sh
./load_supplier.sh
./load_part.sh
./load_customer.sh
./load_partsupp.sh
./load_orders.sh
./load_lineitem.sh
date
```

**Figure 74 load_all_associative.sh**

This is an example loading program for the Lineitem table; as this table is the biggest table in the database, it was necessary to split the input file into many smaller files.

```
echo "Load LineItem"
lst /oracle/home/sentencesdb/usr/local/Lazy/SentencesData35/Chapters/tpch2.chap
time ./CSVImport tpch2 "LINEITEM_DATA_ASSOCIATIVEaa.csv"
echo "completed ... LINEITEM_DATA_ASSOCIATIVEaa.csv"
time ./CSVImport tpch2 "LINEITEM_DATA_ASSOCIATIVEab.csv"
echo "completed ... LINEITEM_DATA_ASSOCIATIVEab.csv"
time ./CSVImport tpch2 "LINEITEM_DATA_ASSOCIATIVEac.csv"
echo "completed ... LINEITEM_DATA_ASSOCIATIVEac.csv"
time ./CSVImport tpch2 "LINEITEM_DATA_ASSOCIATIVEad.csv"
echo "completed ... LINEITEM_DATA_ASSOCIATIVEad.csv"
time ./CSVImport tpch2 "LINEITEM_DATA_ASSOCIATIVEae.csv"
echo "completed ... LINEITEM_DATA_ASSOCIATIVEae.csv"
time ./CSVImport tpch2 "LINEITEM_DATA_ASSOCIATIVEaf.csv"
echo "completed ... LINEITEM_DATA_ASSOCIATIVEaf.csv"
time ./CSVImport tpch2 "LINEITEM_DATA_ASSOCIATIVEag.csv"
echo "completed ... LINEITEM_DATA_ASSOCIATIVEag.csv"
time ./CSVImport tpch2 "LINEITEM_DATA_ASSOCIATIVEah.csv"
echo "completed ... LINEITEM_DATA_ASSOCIATIVEah.csv"
time ./CSVImport tpch2 "LINEITEM_DATA_ASSOCIATIVEai.csv"
echo "completed ... LINEITEM_DATA_ASSOCIATIVEai.csv"
time ./CSVImport tpch2 "LINEITEM_DATA_ASSOCIATIVEaj.csv"
echo "completed ... LINEITEM_DATA_ASSOCIATIVEaj.csv"
lst /oracle/home/sentencesdb/usr/local/Lazy/SentencesData35/Chapters/tpch2.chap
echo "End Load LineItem"
```

**Figure 75 load_lineitem_associative.sh**
A.3.3 Binary-Relational

The following program was used to load the TPC-H database to the Binary Relational Instantiation (MonetDB).

```plaintext
time Mserver --config=/home/monetdb/MonetDB.conf < load_region_tpch.mil

time Mserver --config=/home/monetdb/MonetDB.conf < load_nation_tpch.mil

time Mserver --config=/home/monetdb/MonetDB.conf < load_part_tpch.mil

time Mserver --config=/home/monetdb/MonetDB.conf < load_supplier_tpch.mil

time Mserver --config=/home/monetdb/MonetDB.conf < load_partsupp_tpch.mil

time Mserver --config=/home/monetdb/MonetDB.conf < load_customer_tpch.mil

time Mserver --config=/home/monetdb/MonetDB.conf < load_orders_tpch.mil

time Mserver --config=/home/monetdb/MonetDB.conf < load_lineitem_tpch.mil
```

Figure 76 load_all_binary.sh

This is a sample program used to load the lineitem table according to the specifications of MonetDB’s bulk loader. It is written in Monet’s native language called “mil”

```plaintext
module(ascii_io, unix);
make_persistent(load_data(load_format("/oracle/home/monetdb/tpch/data10GB/LINEITEM"), "/oracle/home/monetdb/tpch/data10GB/LINEITEM_DATA", -1));
commit(); quit();
```

Figure 77 load_lineitem_binary.sh

This is a sample of a header file as required by the MonetDB bulk loader; it has the column name, the field separator and the data type used for that column.

```plaintext
LINEITEM_L_COMMENT,"|",str
LINEITEM_L_COMMITDATE,"|",str
LINEITEM_L_DISCOUNT,"|",int
LINEITEM_L_EXTENDEDPRICE,"|",int
LINEITEM_L_LINENUMBER,"|",str
LINEITEM_L_ORDERKEY,"|",int
LINEITEM_L_PARTKEY,"|",int
LINEITEM_L_QUANTITY,"|",int
LINEITEM_L_RECEIPTDATE,"|",str
LINEITEM_L_RETURNFLAG,"|",str
LINEITEM_L_SHIPPEDDATE,"|",str
LINEITEM_L_SHIPINSTRUCT,"|",str
LINEITEM_L_SHIPMODE,"|",str
LINEITEM_L_SUPPKEY,"|",int
LINEITEM_L_TAX,"\n",int
```

Figure 78 LINE_ITEM_ Binary-Relational header file
A.3.4 Transrelational™

This is the program to load the TPC-H database into the Transrelational instantiation.

```
Figure 79 load_all_transrelational.sh

In order to load each table, the program called crea_TR_Tables_V23.sh has been developed. It is the program which implements the Algorithms described by C.J. Date in [Date, 2004]. This program is listed in Appendix B.
```