Supplementary Material: SQL queries for CFD^{p}s

Shuai Ma, Liang Duan, Wenfei Fan, Chunming Hu, and Wenguang Chen

Below we show how the SQL queries Q_i^c and Q_i^v are generated for validating CFD^{p}s in $\Sigma^i_{\mathsf{cfd^p}}$, which is an extension of the SQL techniques for CFDs and eCFDs discussed in [2] and [1], respectively.

The queries Q_i^c and Q_i^v for the violations of $\Sigma^i_{\mathsf{cfd^p}}$ are given as follows, which capitalize on the data table enc_L, enc_R and $\mathsf{enc}_{\neq}$ that encode CFD^{p}s in $\Sigma^i_{\mathsf{cfd^p}}$.

Q_i^c: **select** $R_i.*$ **from** R_i, enc_L L, enc_R R, $\mathsf{enc}_{\neq}$ N
where $L.\mathsf{cid} = R.\mathsf{cid}$ **and** $R_i.X \asymp L$ **and** $R_i.X \asymp N$ **and**
not ($R_i.Y \asymp R$ **and** $R_i.Y \asymp N$)

Q_j^v: **select distinct** X_L
from (**select** $L.\mathsf{cid}$ **as** cid, X_L, Y_R **from** R_i, enc_L L, enc_R R, $\mathsf{enc}_{\neq}$ N
where $L.\mathsf{cid} = R.\mathsf{cid}$ **and** $R_i.X \asymp L$ **and**
$R_i.X \asymp N$ **and** $R.Y =' \_'$) **as** M
group by cid, X_L **having count** (**distinct** Y_R)> 1

Here (1) $X = \{\mathsf{A}_1, \ldots, \mathsf{A}_{\mathsf{m1}}\}$ and $Y = \{\mathsf{B}_1, \ldots, \mathsf{B}_{\mathsf{m2}}\}$ are the sets of attributes in LHS and RHS of $\Sigma^i_{\mathsf{cfd^p}}$ respectively; (2) $R_i.X \asymp L$ is the conjunction of

$L.A_j$ **is null or** $R_i.A_j = L.A_j$ **or** ($L.A_j =$ '_'
and ($L.A_{j_>}$ **is null or** $R_i.A_j > L.A_{j_>}$)
and ($L.A_{j_\geq}$ **is null or** $R_i.A_j \geq L.A_{j_\geq}$)
and ($L.A_{j_<}$ **is null or** $R_i.A_j < L.A_{j_<}$)
and ($L.A_{j_\leq}$ **is null or** $R_i.A_j \leq L.A_{j_\leq}$))

for each $j \in [1, m_1]$; (3) $R_i.Y \asymp R$ is defined similarly for attributes in Y; (4) $R_i.X \asymp N$ is the conjunction of

not exists (**select** $*$ **from** N
where $L.\mathsf{cid}$ = $N.\mathsf{cid}$ **and** $N.\mathsf{pos}$ = 'LHS' **and**
$N.\mathsf{att}$ = 'A_j' **and** $R_i.A_j = N.\mathsf{val}$)

for each $j \in [1, m_1]$; (5) $R_i.Y \asymp N$ is defined similarly, but with $N.\mathsf{pos} =' \mathsf{RHS}'$; (6) X_L is the set of following attributes

(**case when** $L.A_j$ **is not null then** $R_i.A_j$ **end**) **as** A_{Lj}

for each $j \in [1, m_1]$; (7) Similarly, Y_R is the set of

(**case when** $R.B_k$ **is not null then** $R_i.B_k$ **end**) **as** B_{Rk}

for each $k \in [1, m_2]$; (8) $R.Y$ = '_' is the disjunction of $R.B_k$ = '_' for each $k \in [1, m_2]$.

Intuitively, detecting violations of CFD^{p}s is a two-step process. First, query Q_i^c detects single-tuple violations, *i.e.*, the tuples t in I_i that match the LHS of a CFD^p in $\Sigma^i_{\mathsf{cfd^p}}$, but do not match its RHS. Second, query Q_i^v finds multi-tuple violations, *i.e.*, the tuples t in I_i such that (a) there exists another tuple t' in I_i, t and t' match and agree on the LHS of a CFD^p in $\Sigma^i_{\mathsf{cfd^p}}$, but do not agree on the RHS of the CFD^p.

Example 1: Using the coding of Fig. 4, two SQL queries for checking CFD^{p}s φ_2, φ_3 and φ_4 of Fig. 2 are given as follows:

Q_1^c: **select** $R_1.*$ **from** item R_1, enc_L L, enc_R R, $\mathsf{enc}_{\neq}$ N
where $L.\mathsf{cid} = R.\mathsf{cid}$ **and**
($L.\mathsf{sale}$ **is null or** $R_1.\mathsf{sale} = L.\mathsf{sale}$ **or** $L.\mathsf{sale} =' \_'$) **and**
not exists (**select** * **from** N
where $N.\mathsf{cid} = L.\mathsf{cid}$ **and** $N.\mathsf{pos} =' \mathsf{LHS}'$ **and**
$N.\mathsf{att} =' sale'$ **and** $R_1.\mathsf{sale} = N.\mathsf{val}$) **and**
($L.\mathsf{price}$ **is null or** $R_1.\mathsf{price} = L.\mathsf{price}$ **or** ($L.\mathsf{price} =' \_'$ **and**
($L.\mathsf{price}_>$ **is null or** $R_1.\mathsf{price} > L.\mathsf{price}_>$) **and**
($L.\mathsf{price}_\leq$ **is null or** $R_1.\mathsf{price} \leq L.\mathsf{price}_\leq$))) **and**
not exists (**select** * **from** N
where $N.\mathsf{cid} = L.\mathsf{cid}$ **and** $N.\mathsf{pos} =' \mathsf{LHS}'$ **and**
$N.\mathsf{att} =' price'$ **and** $R_1.\mathsf{price} = N.\mathsf{val}$) **and**
not (($R.\mathsf{shipping}$ **is null or** $R_1.\mathsf{shipping} = R.\mathsf{shipping}$ **or**
$R.\mathsf{shipping} =' \_'$) **and**
not exists (**select** * **from** N
where $N.\mathsf{cid} = R.\mathsf{cid}$ **and** $N.\mathsf{pos} =' \mathsf{RHS}'$ **and**
$N.\mathsf{att} =' shipping'$ **and** $R_1.\mathsf{shipping} = N.\mathsf{val}$) **and**
($R.\mathsf{price}$ **is null or** $R_1.\mathsf{price} = R.\mathsf{price}$ **or** ($R.\mathsf{price} =' \_'$ **and**
($R.\mathsf{price}_\geq$ **is null or** $R_1.\mathsf{price} \geq R.\mathsf{price}_\geq$) **and**
($R.\mathsf{price}_<$ **is null or** $R_1.\mathsf{price} < R.\mathsf{price}_<$))) **and**
not exists (**select** * **from** N
where $N.\mathsf{cid} = R.\mathsf{cid}$ **and** $N.\mathsf{pos} =' \mathsf{RHS}'$ **and**
$N.\mathsf{att} =' price'$ **and** $R_1.\mathsf{price} = N.\mathsf{val}$))

Q_1^v: **select distinct** sale_L, $\mathsf{price}_\mathsf{L}$ **from** (
select $L.\mathsf{cid}$ **as** cid ,
(**case when** $L.\mathsf{sale}$ **is not null then** $R_1.\mathsf{sale}$ **end**) **as** sale_L,
(**case when** $L.\mathsf{price}$ **is not null then** $R_1.\mathsf{price}$ **end**) **as** $\mathsf{price}_\mathsf{L}$,
(**case when** $R.\mathsf{shipping}$ **is not null then** $R_1.\mathsf{shipping}$ **end**) **as** $\mathsf{shipping}_\mathsf{R}$,
(**case when** $R.\mathsf{price}$ **is not null then** $R_1.\mathsf{price}$ **end**) **as** $\mathsf{price}_\mathsf{R}$
from item R_1, enc_L L, enc_R R, $\mathsf{enc}_{\neq}$ N
where $L.\mathsf{cid} = R.\mathsf{cid}$ **and**
($L.\mathsf{sale}$ **is null or** $R_1.\mathsf{sale} = L.\mathsf{sale}$ **or** $L.\mathsf{sale} =' \_'$) **and**
not exists (**select** * **from** N
where $N.\mathsf{cid} = L.\mathsf{cid}$ **and** $N.\mathsf{pos} =' \mathsf{LHS}'$ **and**
$N.\mathsf{att} =' sale'$ **and** $R_1.\mathsf{sale} = N.\mathsf{val}$) **and**
($L.\mathsf{price}$ **is null or** $R_1.\mathsf{price} = L.\mathsf{price}$ **or** ($L.\mathsf{price} =' \_'$ **and**
($L.\mathsf{price}_>$ **is null or** $R_1.\mathsf{price} > L.\mathsf{price}_>$) **and**
($L.\mathsf{price}_\leq$ **is null or** $R_1.\mathsf{price} \leq L.\mathsf{price}_\leq$))) **and**
not exists (**select** * **from** N
where $N.\mathsf{cid} = L.\mathsf{cid}$ **and** $N.\mathsf{pos} =' \mathsf{LHS}'$ **and**
$N.\mathsf{att} =' price'$ **and** $R_1.\mathsf{price} = N.\mathsf{val}$) **and**
($R.\mathsf{shipping} =' \_'$**or** $R.\mathsf{price} =' \_'$)) **as** M
group by cid, sale_L, $\mathsf{price}_\mathsf{L}$
having count (**distinct** $\mathsf{shipping}_\mathsf{R}$, $\mathsf{price}_\mathsf{R}$)$> 1$

□

- *S. Ma, L. Duan and C. Hu are with the SKLSDE lab, School of Computer Science and Engineering, Beihang University, China. E-mail: {mashuai, duanl, hucm}@act.buaa.edu.cn.*
- *W. Fan is with the RCBD center, Beihang University, China and the School of Informatics, Edinburgh University, UK. E-mail: wenfei@inf.ed.ac.uk.*
- *W. Chen is with the Department of Information Management, Peking University, China. E-mail: chenwg@pku.edu.cn.*

Manuscript received XXX, 2014; revised XXX, 2014.

ACKNOWLEDGMENTS

This work is supported in part by 973 program (No. 2014CB340300) and NSFC (No. 61322207). Fan is supported in part by 973 Program (No. 2012CB316200), NSFC (No. 61133002), Guangdong Innovative Research Team Program (2011D005), Shenzhen Peacock Program (1105100030834361) of China, and EPSRC (EP/J015377/1) of UK.

REFERENCES

[1] L. Bravo, W. Fan, F. Geerts, and S. Ma. Increasing the expressivity of conditional functional dependencies without extra complexity. In *ICDE*, 2008.

[2] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for capturing data inconsistencies. *TODS*, 33(2), 2008.