

Die Neuausrichtung der IT als Partner des Business am Beispiel technischer Primärschlüssel

Der IT-Berater ist gefragt

Autor(en): Gerhild Aselmeyer, Dipl. Math. – Analysen, Konzepte & Anwendungsentwicklung für EDV

Was hat ein Detail der Datenbankarchitektur mit der Ausrichtung der IT zu schaffen? Diese oder eine ähnliche Frage wird sich wohl jeder beim Lesen der Überschrift stellen. Aber am Beispiel des technischen Primärschlüssels lässt sich anschaulich darstellen, welche Auswirkungen eine rein technische Ausrichtung in der IT auf verarbeitete Daten und damit auf Geschäftsprozesse haben kann. Führt man sich dabei vor Augen, dass seit dem Aufkommen von SOA immer wieder die Forderung laut wird, die IT stärker ins Geschäft zu integrieren oder sogar, die IT müsse sich als Vordenker und Lenker für Geschäftsprozesse positionieren, dann muss meiner Ansicht nach noch umgedacht werden. Allerdings bleibt die Frage, woher kommt die Technikfixierung: sind es die wissenschaftlichen Grundlagen, die Vorgehensmethoden, die eingesetzten Tools und Entwicklungsumgebungen oder mangelnde Kommunikation.

Daher geht dieser Artikel zunächst auf wissenschaftliche Grundlagen des Datenbankdesigns ein, beschäftigt sich danach mit Primärschlüsseln in der praktischen Umsetzung auch und gerade beim Einsatz von Modellierungs- und Entwicklungswerkzeugen (Frameworks wie z.B. APEX) und beschreibt einige Beispiele. Abschließend wird darauf eingegangen, was sich daraus für eine Neuausrichtung der IT ergibt.

Die Idee zu diesem Artikel kam mir, nachdem ich verschiedene kritische Artikel bzgl. der derzeitigen und zukünftigen IT, deren Ausrichtung sowie deren Verhältnis zum Business gelesen und mich gleichzeitig über Restriktionen in APEX bzgl. Primärschlüsseln geärgert hatte.

Ich fragte mich, woher dieses Diktum technischer Primärschlüssel eigentlich herrührt: Habe ich neuere Forschungsergebnisse übersehen oder handelt es sich eher um praktische Gründe? Allerdings können diese künstlichen Schlüssel in der Praxis zu erheblichen Schwierigkeiten und Performanzverlusten führen; mir fielen aus meiner langjährigen Erfahrung gleich mehrere Beispiele ein. Aber viele Tools und Frameworks (Entwicklungswerkzeuge) – wie z.B. APEX - setzen die Verwendung von technischen Primär- oder wenigstens Alternativ-Schlüsseln voraus, da sonst kaum erklärbar ist, warum (Primär-)Schlüssel von Tabellen auf maximal zwei Attribute beschränkt werden.

Wie will die IT als Partner in Geschäft und Produktion fungieren, wenn sie sich durch technische Prämissen geleitet inhaltlich davon abkoppelt?

Datenbankdesign in Forschung und Lehre

Literatur- und Internetrecherche ergaben, dass die wissenschaftlichen Grundlagen für Datenbankdesign immer noch auf der Codd'schen Normalformlehre aufbauen. Zwar gibt es inzwischen Erweiterungen durch die *Boyce-Codd-Normalform* sowie die 4. und 5. Normalform, doch keine von diesen fordert statt des natürlichen einen fachlich unabhängigen – also technischen – Primärschlüssel.

Im Gegenteil habe ich in einem *Lese- und Übungsbuch* ([1], S. 28) sogar folgendes gefunden:

“Aus dem Beispiel der Studentendaten kann der Schluss gezogen werden, dass durchaus ein künstlich erzeugter Schlüsselkandidat sinnvoll sein kann (11), jedoch muss auch warnend erwähnt werden, dass dadurch die Gefahr eines Verlustes an Semantik besteht: Wählt

man (*Name, Vorname, Geburtsdatum, Geburtsort, Anschrift*) zum Primärschlüssel, ist jede Kombination der Attributwerte einzig in der Datenbank. Ist die Matrikelnummer der Primärschlüssel, muss die Einzigkeit der Wertekombination der genannten Attribute zur zusätzliche(n) Bedingung(...) ausdrücklich gefordert werden. Deshalb sollte ein künstlicher Primärschlüssel nur ein letzter Ausweg sein und deshalb (möglichst) nicht im E/R-Modell angewendet werden. Er kann in einem späteren Stadium der Modellierung immer noch in ein Modell eingebracht werden.“

Im weiteren wird für den Übergang vom logischen E/R-Modell zum - physikalisch umsetzbaren - Relationenmodell folgende Regel aufgestellt ([1], S. 47):

„... Der Primärschlüssel wird übernommen, sofern er (...) nur aus einfachen Attributen gebildet wurde. Andernfalls ist er aus den Attributen neu zu bestimmen, die nach Auflösung der zusammengesetzten und mehrfachen Attribute verbleiben. Dabei kann es durchaus vorkommen, dass sich kein Schlüsselkandidat findet und ein künstlicher Primärschlüssel nach Abschnitt 2.3.2 (Seite 27) eingeführt werden muss.“

Auch die Forderung nach Unveränderbarkeit und Einwertigkeit von Primärschlüsseln, die im allgemeinen für den Einsatz technischer Primärschlüssel angeführt wird, lässt sich wissenschaftlich kaum begründen. In zwei Vorlesungsskripten für Wirtschaftsinformatik aus 2005 und 2009 finden sich mit Einschränkungen oder indirekt diese Forderungen, aber immer im Zusammenhang mit der Forderung, dass die Schlüssel auch ‚*sprechend*‘ sein - also wohl Bezug zur Fachlichkeit haben - sollten.

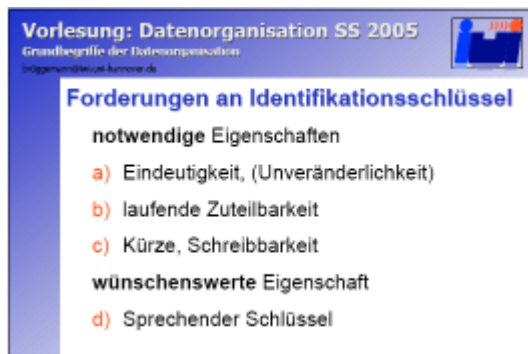


Abbildung 1: aus [2] Seite 18/ Folie 35

In der zweiten Vorlesung gibt es die folgende Definition: “Ein Identifikationsschlüssel ist ein Attribut oder eine minimale Attributkombination, anhand dessen bzw. deren jedes Tupel der Relation eindeutig identifiziert werden kann und dessen bzw. deren Wert zeitinvariant ist, d.h. sich während der Existenz des Tupels nicht ändert.” ([3], Folie 119). Auf der folgenden Folie werden dann Forderungen für ‘künstliche Identifikationsschlüssel’ aufgestellt:

Forderungen an künstliche Identifikationsschlüssel sind:

- Eindeutigkeit (Unveränderlichkeit)
- laufende Zerteilbarkeit (max. auftretende Entität erhält Schlüssel sofort)
- Kürze und Schreibbarkeit
- „sprechende“ Schlüssel
(aus den Schlüsselwerten sollen bestimmte Eigenschaften der beschriebenen Entität erkennbar werden)

Verbundschlüssel umfassen einen ? Identifikationsteil und einen ? Klassifikationsteil.

Parallelschlüssel sind Verbundschlüssel, bei denen es keine Beziehung zwischen klassifizierendem und identifizierendem Teil gibt.

Bsp.: Zugbezeichnung IC 100.
IC bezeichnet einen Zug der Klasse IC.
100 ist eine willkürliche Bezeichnung, die keine Aussage über die Art des Zuges liefert.

Abbildung 2: aus [3] Folie 120

Aber auch hier wird nirgendwo gefordert, dass jede Relation einen künstlichen Identifikationsschlüssel als (technischen) Primärschlüssel haben muss.

Primärschlüssel in der Praxis

Die aus dem Lese- und Übungsbuch zitierten Aussagen bilden auch mein Verständnis von der Handhabung künstlicher oder technischer Primärschlüssel. Darüber hinaus können solche als Alternativ- oder Sekundärschlüssel bei der Denormalisierung im Übergang zum physikalischen Relationenmodell aus Performanzgründen oder zur Programmiererleichterung eingeführt werden. Allerdings sollte man dabei immer bedenken, dass auch der natürliche Primärschlüssel eindeutig und die Bijektion der beiden Schlüssel – d.h. einem technischen umkehrbar eindeutig ein natürlicher Schlüssel zugeordnet werden kann - über alle Anwendungsfälle erhalten bleiben muss. Daher sollten auch die Attribute natürlicher Primärschlüssel bis auf Ausnahmen unveränderbar sein, das entspricht meiner Erfahrung nach auch weitgehend der Realität. Aber gerade letzterem widersprechen die Verfechter der technischen Primärschlüssel: Jedes fachliche Attribut ist änderbar und dann benötigt man einen enormen Aufwand, um Änderungen an Primärschlüsseln durchzuführen - man denke nur z.B. an die Umstellung der Postleitzahlen oder wenn Firmen fusionieren an Personal-, Abteilungs-, Rechnungsnummern. Ja, der Aufwand ist hoch, doch diesen sollte man zur Gewährleistung der Datenqualität betreiben.

Denn gerade die Umstellung der Postleitzahlen in Datenbeständen mit technischen Primärschlüsseln dürfte wohl heute noch zum Teil der Grund für fehlerhafte Kundendaten sein. Jeder Kundensatz enthielt den technischen Schlüssel für die Postleitzahl seines Wohnortes. Die Postleitzahl zu diesem Schlüssel wurde auf den Startwert (z.B. 10001 für Berlin) geändert und die zusätzlichen mit neuen Schlüssel eingefügt – und alles war erledigt, nur die Adressen der Kunden leider nicht mehr ganz korrekt.

Noch gravierender wirkte sich die Umstellung auf technische Primärschlüssel in einer Datenbank mit Rechnungen aus, wo – nach Aussage eines neuen DBA gemäß dem Stand der Technik – auf den natürlichen Schlüssel ganz verzichtet wurde. Nur die Batchanwendung zur automatischen Verarbeitung der Rechnungen wurde nicht umgestellt; es hatte sich fachlich nichts geändert und der technische Schlüssel wird von der Datenbank generiert. Doch die Entscheidung, ob eine Position zu einer neuen oder bereits existierenden Rechnung gehört, lief über ein INSERT-Statement des Rechnungskopfes – die Rechnungsnummer muss schließlich auf Grund gesetzlicher Forderung eindeutig sein. Anwenden und Revision anschließend zu erklären, dass eine rein technische Umstellung gemäß dem Stand der Technik zu doppelten Rechnungsnummern mit unvollständigen Positionslisten, also einer fehlerhaften Abbildung der Real- in der Miniwelt führt, war nicht ganz unproblematisch. Die Korrektur der fehlerhaften Daten erforderte einen nicht unerheblichen Aufwand.

Die beiden Beispiele zeigen, dass auch technische Primärschlüssel erhebliche Probleme aufwerfen. Außerdem grenzt sich die IT damit eindeutig von der Fachbezogenheit der Daten und damit den Anwendern und dem Geschäft ab.

Jetzt interessierte mich aber doch, ob sich nicht doch ein Grund für diese Einstellung finden ließe. Eine weitere Internetrecherche ergab, dass die Forderung nach unveränderbaren technischen Primärschlüsseln außer in einschlägigen Foren, Büchern und Aufsätzen von IT-Beratern sowie bei Beschreibungen von Frameworks, in wissenschaftlichen Aufsätzen und Vorlesungen zu objektorientierter Programmierung (OOP) und Datenbanksystemen (OODBS) vorkommt. “Allerdings stützt sich ein derartiges Datenbanksystem auf ein eigenes Datenbankmodell (objektorientiert, OODM) ab und darin liegt die Problematik: Eine einheitliche, allgemeingültige Definition für ein OODM gibt es noch nicht.” ([4], S. 57), stellt Prof. Sauer in seinem Vorlesungsskript aus dem Jahr 2006 fest. Im folgenden wird dann die eindeutige fachlich unabhängige Identifizierung jedes Objektes in einem OODM postuliert und festgestellt, dass “der Anwender entlastet (ist) von der Festlegung eindeutiger, unveränderlicher Schlüssel für alle Objekte (, er darf dies auf Wunsch aber auch weiterhin tun).” ([4], S. 68). Letzteres wird der verständige Anwender wohl auch tun, da er auch in der Realität seine Rechnungen, Fahrzeuge und Kunden eindeutig identifizieren können muss. Nach dem Lesen des Kapitels “Identity Field” im Aufsatz “Muster für betriebliche Anwendungen - Objekt-Relationale Muster“ ([5], S. 43f) fiel mir ein, dass, nachdem sich die objektorientierte Programmierung mit C++ gefolgt von Java in der Praxis durchsetzte, in den späten 1990er Jahren die Hersteller von relationalen Datenbankmanagementsystemen (RDBMS) diese objektorientiert erweiterten. Erst im Anschluss daran kamen dann die Forderungen nach technischen Primärschlüsseln und Frameworks, die diese voraussetzen und einfordern.

Hat die IT-Welt und allen voran die Hersteller von entsprechenden Modellierungs- und Entwicklungswerkzeugen wissenschaftliche Ergebnisse zur Objektorientierung relationalen Datenbanken übergestülpt, ohne die Auswirkungen zu bedenken? Mir kommt es so vor; als IT-Berater gerät man damit aber in eine Zwickmühle –

jedenfalls wenn man den Anwender, sein Geschäft und damit seine Anliegen wirklich ernst nimmt. Denn der Einsatz von Werkzeugen ist mit Sicherheit sinnvoll, da sich hierdurch nicht nur Kosten sparen sondern auch einfacher graphische Darstellungen und Prototypen als Kommunikationsgrundlage erstellen lassen. Manche Hersteller solcher Entwicklungswerkzeuge träumen sogar davon, die IT im Bereich Anwendungsentwicklung überflüssig zu machen, da sich der Anwender framework-gestützt doch gleich selbst helfen kann; aber das ist weder seine Aufgabe noch bringt es auf Dauer den erwünschten Nutzen.

Beim Einsatz von Entwicklungswerkzeugen wie z.B. APEX von Oracle zur Erstellung von Datenbank-Web-Anwendungen steht man spätestens bei der Darstellung einer vom Anwender manipulierbaren n:m-Beziehung zwischen Objekten, von denen eines einen natürlichen Primärschlüssel aus mehr als einem Attribut besitzt, vor der Wahl, entweder auf die kostengünstige automatische Generierung zu verzichten und die Relation manuell aufzubauen oder zumindest technische Sekundärschlüssel einzuführen. Außerdem wird dem Anwender selbst die Eingabe eines Primärschlüsselattributes konsequent vom Tool verweigert. Damit muss man entweder wirklich auf natürliche Primärschlüssel verzichten oder mit höherem Entwicklungsaufwand das Tool überlisten. Jetzt ist mit Sicherheit dem Anwender egal, ob eine Applikation bzw. das darunterliegende Datenbankmodell mit technischen Schlüsseln arbeitet, solange auch die Eindeutigkeit seiner natürlichen Schlüssel gewährleistet wird und auf jeder Ebene die Information des natürlichen Schlüssels zur Verfügung steht. Doch gerade letzteres kann bei konsequentem Einsatz von technischen Schlüsseln zu Performanzproblemen führen.

Stellen wir uns einen Betrieb mit einer Abteilung vor, in welcher mehrere Gruppen Versuche durchführen und diese in einem elektronischen Journal protokollieren. Dann besteht für den Anwender am Bildschirm das Versuchsprotokoll zumindest aus der Angabe der Kennungen des Versuches, des Protokollführers und eventuell der Gruppe sowie der Beschreibung der einzelnen Versuchsschritte. Bei der Darstellung in der Miniwelt des Datenmodells ausschließlich mit technischen und dem vollständigen Verzicht auf natürliche Schlüssel in Kindtabellen benötigt man im Minimum eine Verknüpfung (Join) von drei Datenbanktabellen, um die Informationen anzuzeigen, wenn die Versuchskennung firmenweit eindeutig ist und der Anwender sich daher überzeugen lässt, auf die Anzeige der Gruppenkennung zu verzichten. Bedenkt man weiterhin, dass aus Gründen der Vergleichbarkeit und damit Datenqualität eine Reihe von Informationen in den Beschreibungen der Versuchsschritte auf vorgegebene Wertmengen aus Vorgabe-(Lookup-)Tabellen eingeschränkt sind, kommen schnell Verknüpfungen über 6 oder mehr Tabellen zusammen. Dass sich dieses verheerend auf die Zugriffsgeschwindigkeit auswirkt, lässt sich leicht vorstellen. Bei der Verwendung von natürlichen eindeutigen Schlüsseln in den Kindtabellen, deren Werte für die Kennungen der Anwender trotzdem selbst wählen darf, reduziert sich die zu verknüpfende Tabellenanzahl erheblich. Setzt man darüberhinaus auch Fremdschlüsselbeziehungen für die natürlichen Schlüssel ein, erhält der Anwender ein System, das ihn in seiner Arbeit wirklich unterstützt und nicht durch lange Wartezeiten ausbremst.

IT: Partner in Geschäft und Produktion

Auf Grund der bisherigen Schilderungen reicht es für eine Neuausrichtung der IT wohl kaum, die IT-Abteilungen zu zerschlagen wie in einer Debatte des COI-Netzwerks im Internet [6] gefordert wird oder einen Anwender zum IT-Chef zu machen ([6],

S.4). Denn im ersten Fall wird sich das technisch ausgerichtete Denken der IT-Mitarbeiter auch bei organisatorischer Eingliederung in Anwendungsabteilungen nicht automatisch ändern und einem Anwender als IT-Chef fehlt wohl doch der technische Hintergrund, um Risiken wirklich abzuschätzen und technische Innovationen hinreichend zu verstehen.

Betrachten wir hierzu das Thema SOA, das ja unter anderem den Anstoß gab, über neue Ausrichtungen oder Definitionen von IT nachzudenken. SOA soll schließlich nicht primär als IT-Thema gesehen werden, sondern geschäftsgetrieben sein; allerdings besitzt es einen nicht unerheblichen und nicht zu unterschätzenden IT-technischen Anteil. Kein Anwender wird Interesse haben und die IT-technischen Erfahrungen und Kenntnisse besitzen, um die entsprechenden technischen Komponenten für die Umsetzung zu bewerten. Um derartige Projekte wirklich nutzbringend durchzuführen, ist meiner Ansicht nach echte Partnerschaft gefragt. Partnerschaft bedeutet für mich, dass mehrere Personen(-gruppen) sich auf gleicher Augenhöhe zusammenfinden, um die jeweiligen Erfahrungen und Kenntnisse zum gemeinsamen Nutzen einzubringen. Diese Partnerschaft muss dann aber auch auf allen Hierarchieebenen und in allen Projektstufen gelebt werden. Es reicht somit nicht, wenn IT- und technischer Projekt-Leiter sich mit CEO und fachlichem Projektleiter mittels UML-Diagrammen über den Prozessablauf verständigen und Konzepte verabschieden, aber anschließend für lange Zeit alle ihren eigenen Wege gehen. Dies kann nämlich leicht zu dem Ergebnis führen, dass in einem manuellen Prozessschritt der Sachbearbeiter wieder viel zu lange oder vielleicht sogar vergeblich auf alle Informationen am Bildschirm warten muss, da diese über technische Primärschlüssel aus zu vielen einzelnen Tabellen zusammengesucht werden müssen oder müßten. Damit ist der Sachbearbeiter unzufrieden und sperrt sich gegen die Neuerungen; der IT-Mitarbeiter fühlt sich beleidigt und seine Arbeit zu wenig gewürdigt.

Eine Neuausrichtung der IT auf eine derartige Partnerschaft kann aber nur gelingen, wenn beide Seiten auf einander zu gehen. Jeder IT-Mitarbeiter muss sich als Berater fühlen und dafür das Verstehen und Unterstützen des Anliegens seines Gegenüber höher priorisieren als (vermeintliche) technische Wahrheiten. Vor allem sollten nicht die Vorgaben von Tools und Rahmenwerken für Modellierung und Entwicklung unhinterfragt als Maß genommen werden. Hier sollte auch jeder mit IT Beschäftigte sich selbst als Anwender fühlen und seine Bedürfnisse gegenüber Herstellern formulieren. Andererseits sollte die Geschäftsseite sich nicht als verkappte IT-Techniker geben und technische Komponenten oder Verfahren bestimmen wollen.

Auch wenn er am heimischen PC die Verwaltung seiner Termin- und Telefonliste mit ACCESS programmiert hat und er damit schon vertraut ist und vielleicht dann bei Fehlern oder zusätzlichen Wünschen doch gleich selber Hand anlegen könnte und und und Trotzdem sollte der Anwender sich auf die Beurteilung von Funktionalität, Fehlerfreiheit und ergonomische Handhabung von IT-Systemen beschränken und Entscheidungen über technische Komponenten, Modelle und Verfahren den Fachleuten überlassen. Schauen wir jetzt auf der Hierarchieebene noch einmal ganz nach oben. Auf der Leitungsebene bedeutet eine solche Partnerschaft auch, dass der IT-Leiter nicht aus allen wichtigen Entscheidungen herausgehalten und anschließend mit unrealistischen Terminforderungen konfrontiert werden darf. Andererseits ist in den meisten Firmen die IT nicht das Kerngeschäft, wenn auch aus eben diesem nicht mehr weg zu denken; daher muss die Technik an den Bedürfnissen von Geschäft und Produktion ausgerichtet werden und nicht umgekehrt.

■ Fazit

Eine derartige Partnerschaft wird wohl kaum gelingen, wenn "ITler ... Soft Skills als Anzeichen für technische Inkompetenz (betrachten)" ([7], S.2). Aber stellt die IT sich in der Mehrheit so dar? Stehen für die Mehrheit der ITler (vermeintliche) technische Wahrheiten vor dem Anwendernutzen oder hinterfragen sie diese nur einfach zu wenig? Oder hält sich die Mehrheit der Anwender einfach nur für die besseren ITler und geben sich daher beratungsresistent?

In meiner langjährigen Praxis als IT-Beraterin habe ich Vertreter für alle diese Einstellungen und Verhaltensweisen getroffen, aber ob diese die Mehrheit bilden, wage ich nicht zu beurteilen. Daher kann ich auch nicht sagen, ob – zu Recht oder Unrecht – die IT-Abteilungen in der Mehrzahl der Firmen als Partner des Kerngeschäfts gesehen werden bzw. sich selber sehen oder nicht. Ist somit ein Wandel der IT bzw. ein Umdenken der ITler für die Zukunft notwendig?

Ich für meinen Teil betrachte mich schon als Partner und Ratgeber meiner Kunden und bespreche Konzepte und Vorgehen am liebsten mit Vertretern der Anwendungs- und IT-Seite. In großen Projekten vor allem bei Konzernen lässt sich die Einbeziehung der Anwender meist nicht realisieren, doch habe ich dabei häufig das ungute Gefühl, dass nach dem Prinzip der 'Stillen Post' Informationen auf dem Weg von der Anforderung bis zur Umsetzung verloren gehen. Vielleicht scheitern auch gerade daran Großprojekte und sollten daher lieber zerlegt und in kleineren Gruppen erledigt werden – aber das wäre ein neues Thema.

Literaturverzeichnis:

1. Sosna, Dieter – Institut für Informatik/ Abt. Datenbanken der Universität Leipzig, Lese- und Übungsbuch Datenbanken: E/R- und Relationenmodell (2008).
2. Brüggemann, Tobias – Institut für Wirtschaftsinformatik/ Wirtschaftswiss. Fakultät der Universität Hannover, Vorlesung: Datenorganisation SS 2005.
3. Prof. Fritzsche – Fachbereich Informatik/ Mathematik der Universität Dresden, Wirtschaftsinformatik II: Datenorganisation – Datenbanken – Kommunikation (SS 2009)
4. Prof. Sauer, Jürgen – Fachhochschule Regensburg, Datenbanken: Skriptum zur Vorlesung im SS 2007.
5. Becker, Steffen u.a. – Fakultät Informatik der Universität Karlsruhe, Modellgetriebene Software-Entwicklung - Architekturen, Muster und Eclipse-basierte MDA, Interner Bericht 2006-18.
6. Wolf, Tanja – Debatte im COI-Netzwerk, Zerschlagt die IT-Abteilung (<http://www.cio.de/strategien/2224614/index.html>)
7. König, Andrea – CIO-Netzwerk, Zeichen für Inkompetenz: Itler halten nichts von Soft Skills (<http://www.cio.de/karriere/2225711/index2.html>)

Kontakt:

Gerhild Aselmeyer
g.aselmeyer@aka-edv.de