

Francis Hunger

## Transaktionsverarbeitung in relationalen Datenbanken

Zur Materialität von Daten

aus Perspektive der Transaktion

[...]

die Aktionen eines Prozesses sind in Sequenzen gruppiert,  
welche als Transaktionen bezeichnet werden  
und welche Einheiten von Konsistenz sind.<sup>1</sup>

Die Verarbeitung von Transaktionen stellte spätestens seit den 1890er Jahren für Dienstleistungsunternehmen, Produktionsbetriebe, Banken, Verwaltungen, Versicherungen und Bibliotheken ein dauerhaftes Problem dar. Fluggesellschaften koordinieren Kapazitäten und Leistungen,<sup>2</sup> Produktionsbetriebe wollen Maschinen auslasten, Materialien und Arbeitskräfte zum richtigen Zeitpunkt aktivieren und einen aktuellen Überblick über den Stand der Bestellungen leisten,<sup>3</sup> Versicherungen binden die Produktbedingungen verschiedenster Versicherungsprodukte an die Kunden,<sup>4</sup> Bibliotheken wollen Wissen verwalten und Ausleihen nachverfolgen<sup>5</sup> und Banken müssen sicher stellen, dass aus dem Haben des einen Kunden das Soll eines anderen Kunden resultiert und die Transaktion vollständig ist.<sup>6</sup> All diese Informationsbedürfnisse zeichnen sich dadurch aus, dass verschiedene Ordnungen miteinander zu vereinen sind: topologische, quantitative und zeitliche Ordnungen. Transaktionen zeichnen sich somit durch eine koordinierende Funktion aus, in deren Zuge Daten erhoben, in Formation gebracht und in Relation zueinander gesetzt werden, um daraus in einem nachfolgenden Schritt Analysedaten zu gewinnen. Während Stammdaten und Transaktionsdaten der unmittelbaren Abwicklung der Vorgänge dienen, erzeugen Analysedaten neue epistemische Ordnungen, die der Simulation zuarbeiten.

(1) Kapili P. Eswaran/Jim N. Gray/Raymond A. Lorie/Irving L. Traiger, The Notions of Consistency and Predicate Locks in a Database System, in: Communications of the ACM 19/11, 1976, S. 624–633, hier S. 624. (Soweit nicht anders angegeben, stammen die Übersetzungen englischsprachiger Zitate vom Verf.)

(2) Duncan G. Copeland/Richard O. Mason/James L. McKenney, Sabre. The Development of Information-Based Competence and Execution of Information-Based Competition, in: IEEE Annals of the History of Computing 17/3, Fall 1995, S. 30–57.

(3) John H. Van Deventer, Planning Production for Profit, New York (NY) 1921; Lisa Conrad, Organisation im soziotechnischen Gemenge. Mediale Umschichtungen durch die Einführung von SAP, Bielefeld 2017.

(4) JoAnne Yates, Structuring the Information Age. Life Insurance and Technology in the Twentieth Century, Baltimore (MD) 2005.

(5) Markus Krajewski, Zettelwirtschaft. Die Geburt der Kartei aus dem Geiste der Bibliothek, Berlin 2002.

(6) Bernardo Batiz-Lazo, Emergence and Evolution of ATM Networks in the UK, 1967–2000, in: SSRN Electronic Journal (2010) ([https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=1534173](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1534173) [Zugriff 13.03.2018]); Sebastian Gießmann, Ein weiteres gemeinsames Medium zur Banken-Kooperation. Der Fall der Eurocard, in: ders./Tobias Röhl/Ronja Trischler (Hg.), Materialität der Kooperation, Wiesbaden 2019, S. 169–198.

Wenn der vorliegende Beitrag sich mit der Transaktion einem Medium der Kooperation widmet, so geschieht dies in Anschluss an Erhard Schüttpelz' Hinweis darauf, dass nach wie vor Medientheorien des Computing ausstehen, welche die *kooperative* Verfasstheit der Medien in den Vordergrund stellen. »Der vernetzte Rechner ist keine Turingmaschine und das Internet ohnehin nicht; und eine Medientheorie des interaktiven und kollaborativen Computing liegt bis heute nur in Ansätzen vor.«<sup>7</sup>

Der vorliegende Beitrag analysiert daher eine spezifische Praxis der Kooperation, die sich als Operationsketten zuerst in den Betriebsbüros der industrialisierten Welt als formatierend-digitalisierte Karteikarten-, Tabellen- und Planaufstellungspraktiken herausbildeten, bevor sie mechanisiert, elektrifiziert, elektronifiziert und schließlich computerisiert zu Datensätzen wurden. Außer Frage steht, dass sich dieser Prozess non-linear, ungeplant, emergent und zeitversetzt parallel vollzog. Doch das Besondere an Transaktionen ist, dass sich an ihnen aufzeigen lässt, wie eine techno-soziale Informationspraxis zu Technologie geronnen ist: als Technik der *Transaktion* in Datenbanken. Dies wird im folgenden Text anhand dreier Datenbankmanagementsysteme aufgezeigt: erstens dem netzwerkorientierten Integrated Datastore (IDS); zweitens dem hierarchisch organisierten Information Management System (IMS) von IBM und drittens der relationalen Algebra, die IBM als System R und DB2 mit der Sprache SQL entwickelte, und die bis heute das weltweit führende Modell der Datenorganisation darstellt.<sup>8</sup>

## Laufzettel und Transaktionsübersichten

Tabellen und Formulare hatten sich seit Jahrhunderten als »Denkwerkzeuge«<sup>9</sup> in die menschlichen Praktiken eingeschrieben. Sie zeichnen sich dadurch aus, dass sie eine feste Struktur für die Eintragung von Daten zur Verfügung stellten.<sup>10</sup> Die räumliche Verteilung von Informationen erlaubte das Aufspalten von Realität in Einzelteile, also die Digitalisierung von Realität. Dabei stellen Tabellen und Formulare ein Modell der Realität zur Verfügung: das Informationsmodell.<sup>11</sup> In dem die Felder entsprechend benannt wurden, war geklärt, welche Art von Daten an welcher Stelle einzutragen war, in welche Formation sie zu bringen waren. Daraus folgte, welche Art von Daten aus der Realität erhoben werden mussten, um eingetragen werden zu können.

(7) Erhard Schüttpelz, *Infrastrukturelle Medien und öffentliche Medien, Media in Action*, (Pre-Print) 2016, hier S. 2 ([https://www.mediacoop.uni-siegen.de/wp-content/uploads/2016/06/schuettpelz\\_infrastrukturelle\\_medien.pdf](https://www.mediacoop.uni-siegen.de/wp-content/uploads/2016/06/schuettpelz_infrastrukturelle_medien.pdf); [Zugriff 02.03.2021]).

(8) Die Auswahl erfolgt, in der gegebenen Kürze, im Sinne einer stringenter Darstellung. Dabei steht nicht die chronologische Abfolge im Zentrum der Darstellung, sondern eine Reihe von spezifischen Entwicklungsmomenten, die genealogisch Sinn ergeben. Die Darstellung ist hier US-zentriert, während ein genauerer Blick eine stärkere geographische ›Verstreutheit‹ von Datenbankentwicklung enthüllt.

(9) Sybille Krämer, *Notationen, Schemata und Diagramme: Über »Räumlichkeit« als Darstellungsprinzip*. Sechs kommentierte Thesen, in: Gabriele Brandstetter/Franck Hofmann/Kirsten Maar (Hg.), *Notationen und choreographisches Denken*, Freiburg–Berlin–Wien 2010, S. 29–45.

(10) Markus Krajewski, *In Formation. Aufstieg und Fall der Tabelle als Paradigma der Datenverarbeitung*, in: *Nach Feierabend. Zürcher Jahrbuch für Wissensgeschichte*, Zürich 2007, S. 37–55; Krajewski, *Zettelwirtschaft*, wie Anm. 5.

(11) Verf., *Die Form der Datenbank. Genealogien, Operationalitäten und Praxeologien relationaler Datenbanken in Ost und West*, Bauhaus Universität Weimar 2021 (in Vorbereitung).

Machine shop .....							
Order for .....							Tires.....
Do work on Tire No .....							
As follows and per blue print .....							
	Tem- plet	Size to be cut to	Depth of cut	Driving belt	Feed	Rate	Time this operation should take
Surface to be ma- chined .....	.....	.....	.....	.....	.....	.....	.....
Set tire on machine ready to turn....	.....	.....	.....	.....	.....	.....	.....
Rough face front edge	.....	.....	.....	.....	.....	.....	.....
Finish face front edge	.....	.....	.....	.....	.....	.....	.....
Rough bore front...	.....	.....	.....	.....	.....	.....	.....
Finish bore front...	.....	.....	.....	.....	.....	.....	.....
Rough face front I. S.C .....	.....	.....	.....	.....	.....	.....	.....
Cut out filled .....	.....	.....	.....	.....	.....	.....	.....
Rough bore front I. S.F .....	.....	.....	.....	.....	.....	.....	.....
Rough face back edge	.....	.....	.....	.....	.....	.....	.....
Finish face back edge	.....	.....	.....	.....	.....	.....	.....
Finish bore back ...	.....	.....	.....	.....	.....	.....	.....
Rough bore back ...	.....	.....	.....	.....	.....	.....	.....
Rough face back I. S.F .....	.....	.....	.....	.....	.....	.....	.....
Cut out filled .....	.....	.....	.....	.....	.....	.....	.....
Cut recess .....	.....	.....	.....	.....	.....	.....	.....
Rough turn thread..	.....	.....	.....	.....	.....	.....	.....
Finish turn thread..	.....	.....	.....	.....	.....	.....	.....
Rough turn flange..	.....	.....	.....	.....	.....	.....	.....
Finish turn edge....	.....	.....	.....	.....	.....	.....	.....
Clean fillet of flange.	.....	.....	.....	.....	.....	.....	.....
Remove tire from machine and clean face plate .....	.....	.....	.....	.....	.....	.....	.....

Abb. 1: Reifenproduktion Anweisungskarte (aus: Taylor, Shop Management, wie Anm. 12, S. 86)

Anfang des 20. Jahrhunderts schlug Frederick Winslow Taylor vor, sich mit der Formalisierung von Arbeitsabläufen im Betrieb auseinanderzusetzen. Ziel dieser Informatisierung der Betriebsorganisation war eine Optimierung der Arbeitsschritte aus ökonomischer Sicht. Tabellenpraktiken und Formularpraktiken halfen laut Taylor einerseits die betriebliche Wirklichkeit zu erfassen und andererseits Optimierungsstrategien zu entwickeln. Die Ergebnisse der Überlegungen sollten wiederum in Form gebracht und als Information in den Betriebsablauf eingeschleust werden. So entstanden Listen, die abuarbeiten waren, Anweisungszettel für die Arbeiter und Laufzettel zum Beispiel Materialscheine und Lieferscheine, die den Stand einzelner Transaktionen belegen (Abb. 1).<sup>12</sup>

Viele dieser durch Taylor als Ideal beschriebenen Informationspraktiken mündeten in Infrastrukturen, sodass in Betrieben sowohl Plantafeln und Laufzettel als auch deren Übersetzungen in Software als Enterprise-Ressource-Planning (ERP)-Systeme wie SAP oder Salesforce

(12) Frederik Winslow Taylor, Shop Management [1903], London–New York (NY) 1911.

REGULAR CONTINUATION REOPERATION				DISPATCH ORDER MEMORANDUM		L.V.E. FORM 104 PART NO. 1155	
DELIVERIES				DEFECTIVES		ORDER NO	LOT NO.
DATE	QUANTITY	DATE	QUANTITY	DATE	QUANTITY	22386	1
						TO START	STARTED
						12-20-'18	8 AM
						EST. DIE TIME	ACTUAL TIME
						EST. SET-UP TIME	ACTUAL OPN TIME
						3.5 hrs.	
OPN. NO.	DEPT. NO.	MACH. NO.	CLOCK NO.	OPERATION			
2	6	D-26		Milling			
PART NAME							SET-UP TIME EFF
Pinion Bolt							%
PIECES THIS LOT		PIECES MADE THIS OPERATION		BAL. TO BE MADE THIS OPERATION		OPN TIME EFF	
1200						%	
							TOTAL EFF C-DT
							%

Abb. 2: Materiallauffkarte. Tabellare Felder und Formfelder sind hier gemischt. Jedes Feld ist benannt und erhält damit seine semantische Bedeutung. Die Teilenummer 1155 liefert eine Speicheradresse für die Informationen des Transaktionsvorganges, die Bestellnummer 22386 verknüpft den Vorgang mit der Speicheradresse der Stammdaten der Bestellung (aus: Van Deventer, Planning Production for Profit, wie Anm. 3, S. 234)

erfahrbar sind.<sup>13</sup> Die Einführung von ERP Software ist nicht ohne die Übersetzung transaktionaler Praktiken in Datenbanktechnologie zu denken. Es lässt sich mit dieser Beobachtung argumentieren, dass für die Emergenz von Datenbankmanagementsysteme die Speicherfunktion nicht allein maßgeblich war, sondern dass Transaktionen dafür eine ganz wesentliche Rolle spielten. Vor diesem Hintergrund können die heutigen ERP, Customer-Relationship-Management und Supply-Chain-Management Softwaresysteme als encodierte Praktiken des »Algotaylorismus«<sup>14</sup> bezeichnet werden (Abb.2). Diese komplexen Softwarefigurationen setzten allerdings die Formalisierung von Transaktionen in Datenbanken voraus, ein historischer Schritt der in den 1960er bis 1990er Jahren erfolgte und im Folgenden genauer beschrieben werden soll.

### Automatisierter Zugriff auf Informationsressourcen

In einem Interview bezeichnete der Ingenieur Charles Bachman die Einführung von Festplattentechnologie ab Ende der 1950er Jahre und die damit einhergehenden softwaretechnischen Neuerungen als »grundlegende Verschiebung«.<sup>15</sup> Damit sprach Bachmann ein Problem an, welches die Informationsverarbeitung in Büros seit Anbeginn plagte. Der gleichzeitige Zugriff auf ein und dieselbe Informationsressource sorgte beständig für Probleme: Sei es, dass eine Vorgangskarteikarte nicht im Karteikasten auffindbar war, da sie sich auf dem Schreibtisch einer Mitarbeiterin befand, oder dass im Falle eines Computersystems immer nur ein Nutzer gleichzeitig eine Datei öffnen und darin schreiben konnte.

(13) Conrad, Organisation im soziotechnischen Gemenge, wie Anm. 3.

(14) Aude Launey, Algotaylorism, in: <http://www.launayau.de/index.php/2019/12/21/algotaylorism/> [Zugriff 29. 12. 2020].

(15) Charles W. Bachman, Charles W. Bachman Interview by Thomas Haigh, Minneapolis (MN), 26.09.2004, hier S. 53 (<https://dl.acm.org/doi/pdf/10.1145/1141880.1141882> [Zugriff 03.03.2021]).

Mit Einführung der neuen Software erfolgte der Buchungsvorgang in Tacoma (WA) zentralisiert über einen GE 235-Computer mit den Programmen WEYCOS 1, Integrated Datastore und Problem Controller. Ermöglicht wurde dies durch einen vorgeschalteten General Electric Datamet 30-Computer, welcher mit bis zu 100 Fernschreibern verbunden war, die zur Eingabe und Ausgabe dienten. Damit zählte WEYCOS 1 neben SABRE von American Airlines zu den frühesten elektronischen Online-Transaction-Processing-Figurationen weltweit, die nicht mehr im Modus der Stapelverarbeitung sondern des direkten Festplattenzugriffes auf Rechenressourcen operierten. Für die transaktionale Verarbeitung war das Programm Problem Controller zuständig:

»Die Fernschreiber waren über die gesamten Vereinigten Staaten verteilt, in allen Weyerhaeuser Verkaufsbüros, Sägewerken und Lagerhäusern. Sozusagen war diese Version des Problem Controller ein tatsächlich Nachrichten-orientiertes Betriebssystem. Einkommende Nachrichten konnten an Dritte gesendet werden, oder sie kamen herein und wurden an den GE 235-Computer übermittelt, wo sie per Problem Controller verteilt wurden.«<sup>16</sup>

Ein Beispiel für eine höhere Prioritätszuordnung von Transaktionen war das Anlegen neuer Kunden in der Datenbank. Erst wenn die Stammdatensätze angelegt waren, konnten die Bestellungen überhaupt verarbeitet werden.

Information nicht-sequentiell zu lesen und die Ebenen in den Informationshierarchien wechseln zu können, machte laut Bachman das wichtigste Novum des Integrated Data Store aus. Wichtige neue Operationen standen damit zur Verfügung: *Concurrency Control* (der gleichzeitige Zugriff verschiedener Nutzerinnen), Zugriffskontrolle, und die Möglichkeit, die Datenbank in einen vorherigen Zustand zu versetzen, der jenen Datenstand abbildet, der vor dem Scheitern einer Transaktion herrschte.

Transaktion, Programm und Wartemechanismus setzte Bachman in ein Verhältnis zueinander: »In der Tat waren es die Transaktionen, die warteten, nicht die Programme. Programme sind nur Verarbeitungs-Regeln, es sind in Wirklichkeit die Transaktionen, die aktiv sind oder warten.«<sup>17</sup> Damit betont Bachmann nicht die Hardware oder Software, sondern die Operationsketten, welche sie abbilden: die Transaktionen. Folgt man Bachmans Zentrierung auf die Transaktion, so lässt sich in Bezug auf die elektronische Datenbank zumindest für die 1950er bis 1970er Jahre eine Zentralstellung der Transaktion als wichtige Triebfeder für technologische Neuerungen konstatieren. War bis dahin die Gleichzeitigkeit des Zugriffes auf Informationsressourcen ein wiederkehrendes informationsorganisatorisches Problem, so erlaubte der Einsatz von Festplatten und Transaktionssoftware eine Geschwindigkeit der Zugriffe, die den Eindruck von Gleichzeitigkeit erweckte.

(16) Ebenda, S. 83.

(17) Ebenda, S. 90.

Bachmans technologische Lösung folgte dem sogenannten Netzwerkmodell. Doch erwies sich dieses auf Dauer als zu komplex und wartungsintensiv.<sup>18</sup> Unter Auslassung einiger Entwicklungen soll nun verfolgt werden, wie in heutigen Datenbankmanagementsystemen, die zu über 70% auf dem relationalen Modell basieren,<sup>19</sup> Transaktionen realisiert werden.

## Kontrollsphären – Formalisierung von Transaktionen

IBMs Datenbankprodukt Information Management System (ab 1969) kannte im Unterschied zur Netzwerkstruktur des Integrated Data Store allein hierarchische Eltern-Kind-Beziehungen.<sup>20</sup> IMS ermöglichte den gleichzeitigen Zugriff auf Daten in der Datenbank, indem Programmen erlaubt wurde, auf abgegrenzte Segmente zuzugreifen. War ein Segment belegt, mussten nachfolgende Programme warten, bis dieses freigegeben war. Was einerseits wichtig war, um überhaupt ›gleichzeitigen‹ Zugriff zu ermöglichen, bereitete in der Anwendungspraxis zahlreiche Probleme, gerade bei Datensätzen mit mehreren Verknüpfungen. Erst ab 1974 war es möglich, dass Programme nicht mehr auf Ebene der Segmente, sondern einzelner Datensätze die Daten editierten, während diese für andere Programme gesperrt blieben. Realisiert wurde dieses neue Verhalten über Synchronisierungspunkte, die es erlaubten, die Softwarefiguration gegebenenfalls zurückzusetzen.

Alle Schritte bis zu einem Synchronisationspunkt wurden in einer Log-Datei aufgezeichnet, um gegebenenfalls Fehler zu finden, bzw. Transaktionen zu wiederholen. Damit konnten Transaktionen aus mehreren Einzelschritten abgebildet werden, und bei Fehlern im Rahmen einer Transaktion brach das laufende Programm ab. Ein Systembetreuer wurde informiert, um die Software auf den letzten konsistenten Zustand anhand eines Synchronisationspunktes zurückzusetzen und neu zu starten. Dafür musste die in der Log-Datei aufgezeichnete Historie von Synchronisationspunkt zu Synchronisationspunkt rückwärts abgeschritten werden. Dies bedeutete in der Praxis, dass manuell eingegriffen werden musste und überprüft werden musste, an welchem Punkt die Datenverarbeitung stehen geblieben war. Die betreffenden Daten waren erneut in die Datenbank zu laden.<sup>21</sup>

Semantiken der Wiederherstellung (*Recovery Semantics*) sollen die Konsistenz von Datensätzen aufrechterhalten. Die Fragestellung war, wie Datensätze, trotz einzelner fehlerhafter oder abgebrochener Eingaben, in ihrer Gesamtheit integer, unkorumpiert bleiben. Dazu wurde die Idee der Kontrollhüllen (*Spheres of Control*) eingeführt (Abb. 3). Diese stellten eine Begrenzung dar, innerhalb derer eine Reihe von Datenbankaktionen,

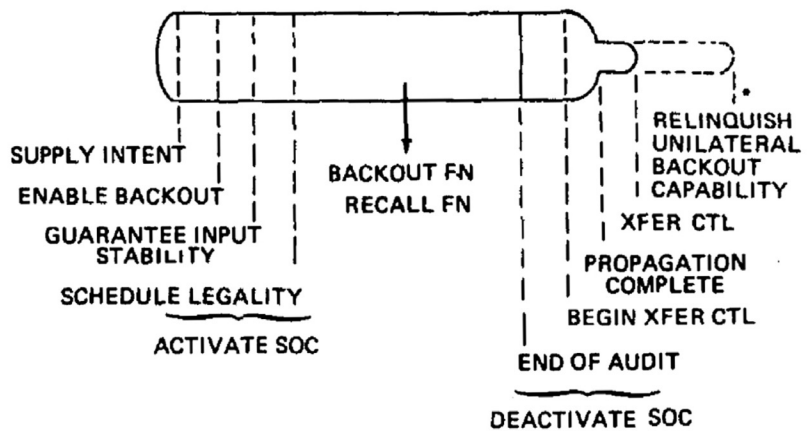
(18) Edgar F. Codd/Charles Bachman, Discussion. Panel and Audience, in: Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control, New York (NY) 1975.

(19) DB-Engines/Solid IT, Ranking pro Datenbankmodell Kategorie, März 2020, in: [https://db-engines.com/de/ranking\\_categories](https://db-engines.com/de/ranking_categories) [Zugriff 10.03.2020].

(20) Diese hierarchische Struktur verdankte Information Management System seiner Herkunft aus einem anderem Software-Produkt, dem Bills of Materials Processor (BOMP), welcher ab 1966 durch die Firmen Caterpillar und North American Aviation/Rockwell mit IBM für das Apollo-Raumfahrtprogramm entwickelt wurde und die Stücklisten des Raumfahrzeugbaus verwaltete.

(21) W. C. McGee, Data Base Technology, in: IBM Journal of Research and Development 25/5, 1981, S. 505–519, hier S. 515f.





\* First commit or release of backout guarantee.

Abb. 3: Verschiedene Abschnitte einer Sphere of Control. Sichtbar ist bereits die Konzeptualisierung mehrerer Phasen, die als Activate SOC und Deactivate SOC gekennzeichnet sind (aus: Davies, Recovery Semantics for a DB/DC System, wie Anm. 22, S. 137)

zum Beispiel das Ändern oder Anlegen von Datensätzen, von spezifischen Nutzern ausgeführt wurden. Entweder alle Aktionen, die innerhalb einer Kontrollhülle stattfanden, schlugen sich *vollständig und korrekt* in der Datenbank nieder oder der Vorgang galt als unabgeschlossen und wurde in seiner Gesamtheit zurück genommen. »Eine Sphere of Control (SOC) verfügt über zugehörige Operationen: aktiviere die SOC, verlasse die SOC, rufe die SOC erneut auf, beende die SOC, übernehme eine Ressource (Nutzungsabsicht), und gib die Ressource frei (tatsächliche Nutzung)«<sup>22</sup>

Davies konzeptueller Vorschlag bestand darin, parallel laufenden Kontrollsphären in einer übergeordneten Kontrollsphäre zu kapseln. Dieses Konzept wurde im Zuge der Arbeiten am relationalen Prototypen System R 1976 auf das Feld der relationalen Datenbanken zugespitzt und mit seinem heutigen Namen als ›Transaktionen‹ bezeichnet.

## Transaktionen in Datenbankmanagementsystemen

Auf der Suche nach neuen Methoden zur Datenspeicherung in Datenbanken entstand 1969 die so genannte relationale Algebra. Ein IBM Forscher schlug vor, Informationen als mathematische Mengen zu prozessieren. Im Unterschied zu vielen vorherigen Entwicklern, hatte Edgar F. Codd keinen Hintergrund als Ingenieur, sondern hatte Mathematik studiert.<sup>23</sup> Bereits von Beginn an war in der Datenbankpraxis ein Problem aufgefallen: die

(22) Charles T. Davies, Recovery Semantics for a DB/DC System, in: Proceedings of the ACM Annual Conference, New York (NY) 1973, S. 136–141, hier S. 140.

(23) Die Darstellung erfolgt hier in der gegebenen Kürze. Codd baute auf eine Reihe vorhergehender Überlegungen, die in meiner Dissertation Die Form der Datenbank. Genealogien, Operationalitäten und Praxeologien relationaler Datenbanken in Ost und West (voraussichtlich 2021) ausführlich diskutiert werden. So sind neben Codd weitere Personen, wie beispielsweise David Childs, der bereits ein Jahr früher eine relationale Algebra entwickelte, als auch Charles Davies zu nennen, der mit Codd Fragen der Maschinenunabhängigkeit von Datenbanken diskutierte.

Reihenfolge der Daten im Speicher orientierte sich an den maschinellen Gegebenheiten der Speicher. Dies traf sowohl auf Bachmans Integrated Datastore zu, der nach dem Netzwerkmodell organisiert war, als auch auf IBMs Datenbankprodukt Information Management System, welches durch hierarchische Beziehungen strukturiert wurde. Beide Ansätze verknüpften die Datensätze über die Speicheradressen, also über unmittelbar auf Maschinenebene wirksame Logiken. Bei Änderungen auf Maschinenebene mussten die Datenbanken in aufwändiger Arbeit restrukturiert und an die neuen Verhältnisse angepasst werden.

Codd schlug stattdessen vor, eine zusätzliche logische Ebene einzuziehen, welche die Repräsentation der Daten von der Art und Weise wie sie gespeichert wurden, abtrennte. Das ›relationale Modell‹ fasste Daten als mathematische Mengen auf, die miteinander durch Relationen in Beziehung gesetzt worden. Auf die Mengen konnten verschiedene Operationen ausgeübt werden. Die relationale Algebra erlaubte Rechenoperationen wie zum Beispiel den Vergleich durch ›größer als‹, ›gleich‹ oder ›kleiner als‹. Boolesche Operatoren wie ›UND‹, ›NICHT‹, ›ODER‹ ermöglichten logische Verknüpfungen dieser Rechenoperationen. Die Auffassung als mathematische Mengen erlaubte außerdem, mathematische Mengenoperatoren zu verwenden wie ›Einschluss‹, ›Ausschluss‹, ›Überschneidung‹ und ›Vereinigung‹.

In diesem an der relationalen Logik orientierten Datenmodell waren die Daten als Array, eine Art Matrix, angeordnet. Der wirkliche Durchbruch erfolgte jedoch, als erkannt wurde dass dieser Array auch als Tabelle repräsentiert werden konnte. Damit war eine auch für Nichtmathematiker verständliche visuelle Metapher gefunden, die als User Interface dienen konnte.

Als bewusste Vereinfachung von Codds mathematisch orientiertem relationalen Calculus DSL / Alpha System, welches in San Jose zwar publiziert, aber nicht implementiert wurde,<sup>24</sup> entstand bei IBM San Jose ab 1974 das System R.<sup>25</sup>

Als wichtigen Teilaspekt bearbeitete im Anschluss an Charles Davies ein Teil des Forschungsteams jene Fragen, welche die Transaktion betrafen – gleichzeitiger Zugriff (*Concurrency*) und Konsistenz (bei Davies: *Recovery Semantics*). Diese Vorgehensweise unterschied sich insofern von vorherigen, als dass die gesuchten Eigenschaften nicht in der Alltagspraxis einer bereits existierenden Software hinzugefügt wurden, sondern von vornherein als integraler Bestandteil von System R, ausgehend von den theoretischen Reflexionen Davies', systematisch entwickelt wurden. Nicht alle dabei eingeschlagenen Wege erwiesen sich als erfolgreich. So verfolgte man

(24) Edgar F. Codd, A Data Base Sublanguage Founded on the Relational Calculus, ACM Press, 1971; Edgar F. Codd/Kenneth L. Deckert/Irving L. Traiger/Dines Bjørner, The GAMMA-0 n-ary Relational Data Base Interface: Specification of Objects and Operations, in: IBM Research Report RJ1200, San Jose (CA) 1973.

(25) Hauptentwickler der Structured English Query Language (SEQUEL), später umbenannt in Structured Query Language (SQL), für System R waren Donald D. Chamberlin und Raymond F. Boyce. Innerhalb von IBM wurde ihr Entwurf an mehreren Standorten aufgegriffen und mit verschiedenen Geschwindigkeiten in funktionierende Systeme implementiert, als SQL/DS von IBM Endicott 1981 und als DB2 von IBM Santa Teresa für Mainframes ein Jahr später (Paul McJones, The 1995 SQL Reunion. People, Projects, and Politics, 20.08.1997; Donald D. Chamberlin, An Interview with Donald D. Chamberlin, Minneapolis (MN), 10.03.2001 (<https://conservancy.umn.edu/bitstream/handle/11299/107215/oh329dc.pdf?sequence=1&isAllowed=y> [Zugriff 03.03.2021]); Bradford W. Wade/Donald D. Chamberlin, IBM Relational Database System. The Early Years, in: IEEE Annals of the History of Computing 34/4, Oct 2012, S. 38–48).



für einige Zeit die Möglichkeit, im Sinne von Codd's Prädikatenlogik, die Frage des gleichzeitigen Zugriffs und der Konsistenz deklarativ zu lösen,<sup>26</sup> musste schließlich diesen Ansatz aus Gründen der Rechenleistung und Komplexität verwerfen und entwickelte eine prozedurale Vorgehensweise.<sup>27</sup> Dem Aufsatz *The Notions of Consistency and Predicate Locks in a Database System* entstammt jedoch das grundlegende Konzept der Atomizität, welches wie folgt hergeleitet wird: »Der Systemzustand ist nicht statisch. Er unterliegt fortwährenden Änderungen aufgrund der Aktionen, welche von Prozessen an Entitäten [Informationsobjekten – F.H.] durchgeführt werden. Lesen und Schreiben sind Beispiele für solche Aktionen.«<sup>28</sup> Eine Transaktion wurde darin als eine Abfolge von Aktionen beschrieben, welche Änderungen verursachen.

Auffallend ist, dass von vornherein das Datenbankmanagementsystem nicht als statisch, sondern als dynamisch gedacht wurde. Die beständigen Änderungen erzeugten eine Situation, in der nicht allein die Informationsobjekte dynamisch sind, sondern auch die Figuration selbst, welche die Informationsobjekte verwaltet. Permanent passt sich das Datenbankmanagementsystem dem Wandel im Datenkorpus an – zum Beispiel indem Indexe aktualisiert werden, indem abhängig von der Komplexität der Abfragen (*Queries*) unterschiedlicher Maschinencode der in den Tiefen der Maschine agiert, generiert wird, oder indem die Zugriffe verschiedenster Nutzer, die sich teils gleichen können, und teils sehr unterschiedlich sein können,<sup>29</sup> entsprechend abgebildet werden.

Weiterhin verfolgte das Team intensiv die Frage, wie ein konsistenter Zustand der Datenbank hergestellt werden könnte, wie also ein Zustand erreicht würde, indem alle Transaktionen abgeschlossen sind. Dabei war spätestens seit Davies klar, dass Konsistenz immer nur zu bestimmten Zeitpunkten erreicht werden würde, an einem »Checkpoint«. Der Dauerzustand relationaler Datenbanken ist Inkonsistenz, die sich aus den ständig ablaufenden, sich überschneidenden Transaktionen und deren atomischen Aktionen ergibt. System R war dazu ausgelegt, immer dann, wenn gerade alle laufenden Transaktionen der verschiedensten Nutzer und Prozesse abgeschlossen waren, den für jenen kurzen Moment konsistenten Zustand festzuhalten und einen Checkpoint zu setzen. Dieser konnte zur Wiederherstellung bei Abstürzen, Lese-/Schreibfehlern von Datenträgern oder Abbrüchen innerhalb von Transaktionen aufgesucht werden.

Neu war auch, dass die Nutzer bzw. Programme Datenbank-Transaktionen in der neu entwickelten Sprache SEQUEL unmittelbar starten und beenden konnten, durch die Befehle BEGIN TRANS und END-TRANS. Inmitten einer Transaktion wurden mit SAVE spezifische Rückkehrpunkte festgelegt, die mit Hilfe von RESTORE wiederhergestellt werden konnten.<sup>30</sup> Diese Rückgängig-Funktion erlaubte das Zurückspulen von Zeit und ein erneutes Anstoßen von Transaktionen.

(26) Eswaran/Gray/Lorie/Traiger, *The Notions of Consistency*, wie Anm. 1.

(27) Donald D. Chamberlin/Morton M. Astrahan/Michael W. Blasgen/James N. Gray/W. Frank King III/Bruce G. Lindsay, *A History and Evaluation of System R*, in: *Communications of the ACM* 24/10, 1981, S. 632–646, hier S. 639.

(28) Eswaran/Gray/Lorie/Traiger, *The Notions of Consistency*, wie Anm. 1, S. 624.

(29) Chamberlin/Astrahan/Blasgen/Gray/King/Lindsay, *A History and Evaluation of System R*, wie Anm. 27.

(30) Morton M. Astrahan/Mike W. Blasgen/Donald D. Chamberlin/Kapali P. Eswaran/Jim Gray/Patricia P. Griffiths/W. Frank King III/Raymond A. Lorie/Paul R. McJones/James W. Mehl/Gianfranco R. Putzolu/Irving L. Traiger/Bradford W. Wade, *System R. Relational Approach to Database Management*, in: *ACM Transactions on Database Systems* 1/2, 1976, S. 97–137, hier S. 108.

Um wiederholende Transaktionen, wie beispielsweise Banküberweisungen, zu optimieren, kamen wiederverwendbare ›Views‹ zum Einsatz. Üblicherweise wurde für jede Abfrage der ausführbare Maschinencode neu optimiert und kompiliert und die optimalen Zugriffspfade auf die Daten ermittelt. Patricia Selinger entwickelte diese Komponente – den Optimizer – maßgeblich.<sup>31</sup> Für sich wiederholende Transaktionen, wie Banküberweisungen, gab es jedoch einen effizienteren Weg: Mit Hilfe von Views konnten Abfragen einmal definiert und durch den Optimizer vorbereitet und zwischengespeichert werden. In der Folge konnten sich wiederholende Transaktionen durch die gespeicherten Views aufgerufen werden, wobei der Umweg über den Optimizer eingespart wurde. Dies ermöglichte es, sowohl Ad-Hoc-Abfragen, als auch regelmäßig wiederkehrende Transaktionen gleichermaßen zufriedenstellend zu verarbeiten – ein wichtiger Erfolgsfaktor für die Durchsetzung relationaler Datenbanksysteme.<sup>32</sup>

Diese ausführliche Darstellung wurde unternommen, um aufzuzeigen, mit welcher Intensität im Zuge von System R Methoden der Transaktionsverarbeitung erforscht und implementiert wurden. In der Kombination all dieser Methoden vollzieht sich der Übergang zu einem neuen Begriff der Transaktionen. Waren mit ›Transaktion‹ bisher die von Menschen in Verbindung mit mechanischen Systemen durchgeführten Aktionen gemeint, welche in der Rechenmaschine modelliert und nachvollzogen wurden, übersiedelte der Begriff ›Transaktion‹ nun in das Feld der Informatik.

## Materialität und Skalierung

Die ›kleine Form‹ des Laufzettels skalierte in relationalen Datenbanken und ermöglichte eine Flexibilisierung in Produktion (Neukombination von Produktelementen, Verdichtung des Arbeitstaktes) und Dienstleistung (Koordination von Passagierflügen, Absicherung von Geldtransfers). Damit gelingt es eine ›Materialität‹ von Stammdaten, Transaktionsdaten und schließlich Analysedaten herauszuarbeiten. Die Medienpraktiken der transaktionalen Kooperation zeichnen sich durch räumliche Verhältnisse verschiedenster Skalierungen aus:

Erstens produzieren die ›kleinen Formen‹ der Karteikarten mit Formularen, Plantafeln, Listen und Tabellen verteilten Daten räumliche Relationen, die gleichzeitig epistemische Relationen abbilden. Heute existieren diese als Datensatz, beziehungsweise in relationalen Datenbanken als Datenmenge. Diese tragen ihre eigenen semantischen Verhältnisse mit sich, da in der Benennung der Datenfelder und Relationen deren Bedeutungsgehalt beschrieben ist. Sie sind Teil eines Informationsmodells, welches die *computable* Realität der jeweiligen Datenbank abbildet. Transaktionsdaten zeichnen sich aus durch eine hohe zeitliche Dichte aus, denn sie beziehen sich auf einzelne Transaktionen, derer es viele gibt. Daher verfallen Transaktionsdaten im Vergleich zu Stammdaten schneller.

(31) Patricia G. Selinger/Morton M. Astrahan/Donald D. Chamberlin/Raymond A. Lorie/Thomas G. Price, Access Path Selection in a Relational Database Management System, in: Proceedings of the 1979 ACM SIGMOD International Conference on the Management of Data, New York (NY) 1979, S. 23–34.

(32) Wade/Chamberlin, IBM Relational Database Systems wie Anm. 25, S. 44.

Zweitens erlauben die in Transaktionen involvierten Übertragungsmedien die Überwindung und Synchronisierung geographischer Distanzen. Hier spannen Transaktionen Relationen zwischen lokal verteilten Informationshubs auf und synchronisieren deren Informationsbedürfnisse. Die kleinen Formen der Karteikarten als Datensatz werden zwecks Übertragung in serielle Logiken überführt, sei es als Tonfolge per Akustik-Modem oder in Datenpakete per Internet, um bei ihrer Ankunft wieder in räumliche Verhältnisse, zum Beispiel auf dem Festplattenspeicher übertragen zu werden, dort allerdings verdichtet.

Drittens ermöglichen die in Datenbankmanagementsysteme implementierten Logiken der Transaktion eine Verdichtung der Formationen, die zu Informationen werden. Diese Verdichtung basiert auf der Automatisierung jenes Anteils koordinatorischer Praktiken, der nunmehr maschinell vollzogen werden kann, während die nicht oder nur schwer formalisierbaren Anteile von Realität, die der menschlichen Aufmerksamkeit bedürfen, weiterhin als Praktiken durch Menschen vollzogen werden. Basierend auf diesen Verdichtungen in Datenbankmanagementsystemen können Transaktionsdaten und Stammdaten derart miteinander kombiniert werden, dass sie Analysedaten ergeben, welche einen neuen epistemischen Wert erzeugen und in Techniken der Simulation und Vorhersage münden. Ende der 1980er Jahre kristallisierte sich dies in der Technik des Online Analytical Processing (OLAP), welche Ende der 1990er zu Big Data weiterentwickelt wurde. Dies ebnete den Weg für eine Lockerung der Stringenz von Datenbanktransaktionen ab den 2000er Jahren, die mit dem Begriff des Post-SQL bezeichnet wurden, und neben den relationalen Datenbanken Grundlagen heutiger Medien der Kooperation und öffentlichen Medien sind.

Transaktionale Datenbank-Praktiken bilden das Rückgrat für Logistiken der globalen, vernetzten Warenproduktion und Kommunikation. Die semantische Verfasstheit der aus Transaktionspraktiken stammenden Datensätze in relationalen Datenbanken benötigt einen anderen Informationsbegriff als den am Signal und Algorithmus orientierten Informationsbegriff nach Shannon/Weaver,<sup>33</sup> der ebenso wie die gewichteten Netzwerke<sup>34</sup> der Künstlichen ›Intelligenz‹ auf das Signal abzielt und die Semantiken ignoriert. Als bester Kandidat für eine semantisch-räumliche Konzeption von Daten tritt hier ›Information als Formation‹ auf, welche durch Formierung der Daten in Relationen räumlich-semantische Episteme erzeugt.

Francis Hunger forscht an der Bauhaus-Universität Weimar.

(33) Claude E. Shannon, A Mathematical Theory of Communication, in: Bell System Technical Journal 27/4, Oct 1948, S. 623–656.

(34) Frank Rosenblatt, The Perceptron. A Perceiving and Recognizing Automation, Project Para Report No. 85-460-1, Cornell Aeronautical Laboratory (CAL), Buffalo (NY) Jan. 1957.