4 Query Processing

> Aufgabe der Anfrageverarbeitung

Realisierung eines mengenorientieren Zugriffs

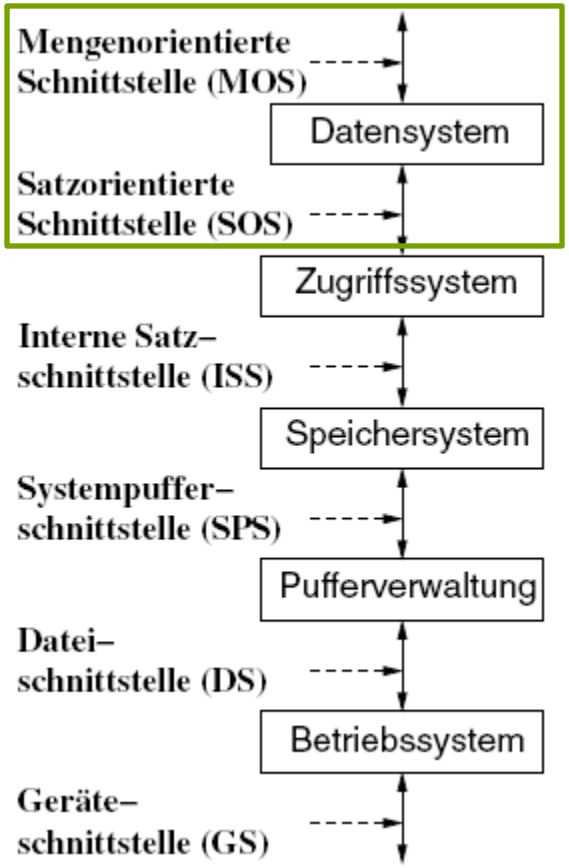
- Abbildung auf die inhaltliche Adressierung von Mengen von Sätzen

Aufgaben

- Überprüfung der syntaktischen Korrektheit von Anfragen
- Überprüfung? von Zugriffsberechtigungen und Integritätsbedingungen
  - Referenzielle Integrität, Eindeutigkeits- und Wertebereichs-zusicherungen, …
- Erzeugung einer optimalen ausführbargen Folge interner DBS-Operationen
  - Anfrageoptimierer ist (im Wesentlichen) für die effiziente Abarbeitung verantwortlich

Was heißt optimal?

Warum ist die Optimierung so aufwändig?



Prozedurale Sprachen

- Direkte Adressierung und satzorientierter Zugriff auf die Datensätze, keine mengenorientierte Verarbeitung
- Verantwortung für die Zugriffspfadwahl, d.h. Art und Reihenfolge der Zugriffe liegt beim Programmierer
- Erlauben leichte Abbildung der DML-Befehle auf interne Satzoperationen (1:1)

Deskriptive Sprachen

- Inhaltliche Adressierung und mengenorientierter Zugriff auf die Datensätze, kein Rückgriff auf einzelne Sätze
- Verantwortung für die Zugriffspfadauswahl, d.h. Art und Reihenfolge der Zugriffe liegt beim System, transparent für den Anwender
- Hohe Auswahlmächtigkeit
  - an der Prädikatenlogik erster Stufe orientiert ist
  - unabhängige oder korrelierte Teilanfragen zur Bestimmung von Suchargumenten in beliebiger Schachtelungstiefe zulässig
  - zusätzlich den Einsatz von Built-in- und Sortier-Funktionen auf Partitionen der Satzmenge gestattet
- Komplexe Abbildung der deskriptiven Anfrage auf interne Satzoperationen

Anfrage auf eine einzelne Tabelle

```
SELECT PNR, PNAME, GEHALT
  FROM PERS
 WHERE BERUF = 'Programmierer'
   AND PROVISION > GEHALT
```

Anfrage mit

```
SELECT P.PNR, P.NAME, A.NAME
  FROM PERS P, ABT A
 WHERE P.ANR = A.ANR
   AND P.GEHALT < (SELECT MAX(PROVISION)
                     FROM PERS)
   AND P.GEHALT > (SELECT AVG(PROVISION)
                     FROM PERS
                    WHERE A.ANR = P.ANR)
```

Verbundoperation

Unabhängige Unteranfrage

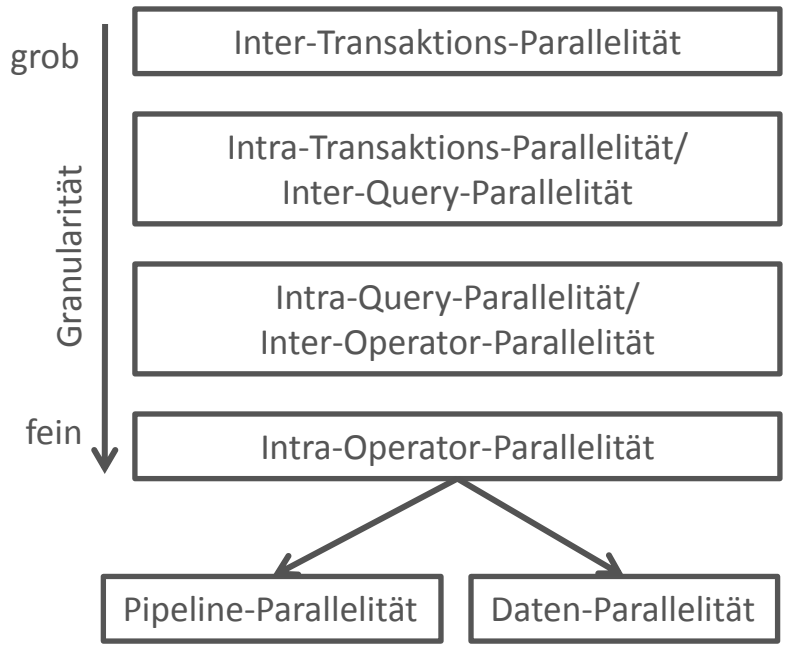
Korrelierte Unteranfrage

Anforderungen bei verteilten/parallelen Datenbankarchitekturen

- Berücksichtigung neuer Faktoren
  - physische Verteilung, Grad der Parallelisierung
  - Kommunikationskosten
  - Einsatz von Replikation
- Erhebliche Steigerung der Komplexität gegenüber Mono-Systemen

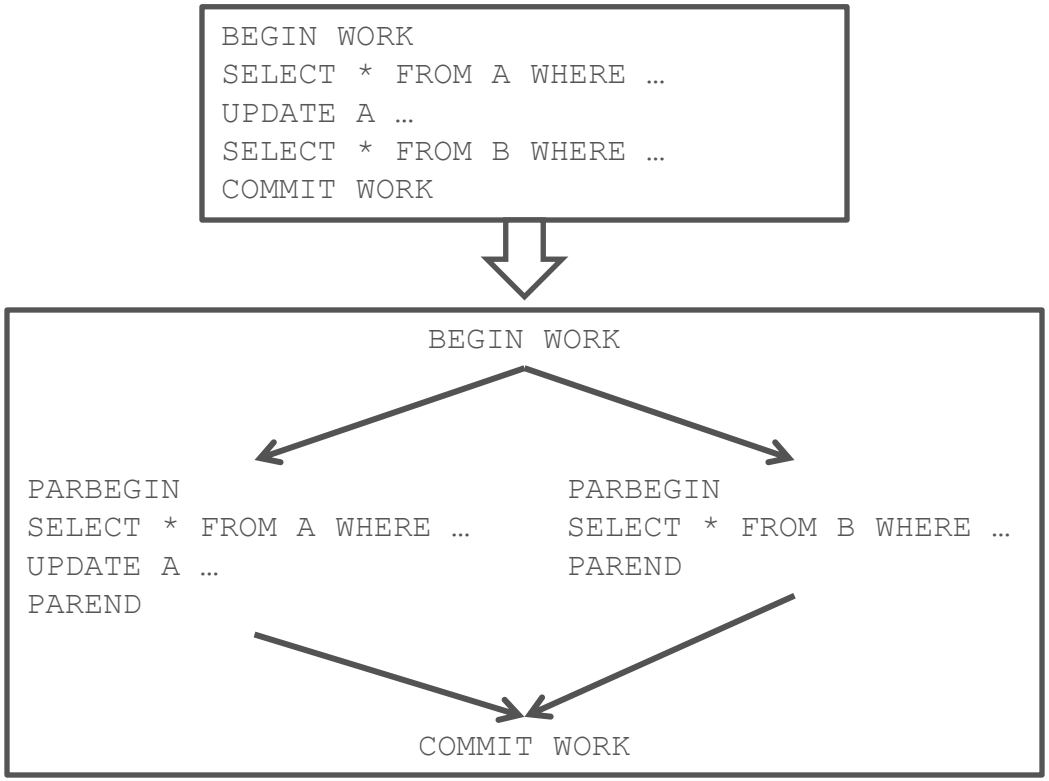
Unterschiedliche Arten von Parallelität

- Granularität der parallelisierten Verarbeitungsschritte
  - Transaktion, Anfrage, Planoperator
- Datenparallelität versus Funktionsparallelität (Pipeline-Parallelität)
- Verarbeitungs- versus E/A-Parallelität



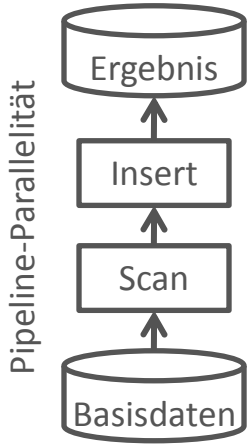
parallele Bearbeitung unabhängiger DB-Operationen (Queries) eines Transaktionsprogramms

- Programmierer muss Parallelisierung spezifizieren
- nur begrenzte Parallelisierung möglich



Pipeline-Parallelität

- Datenfluss-Prinzip zum Datenaustausch zwischen Operatoren/Teiloperatoren
- frühzeitige Weitergabe von Tupeln bei Zwischenergebnissen
- Einsatz vor Allem bei Inter-Operator-Parallelität
- Pipeline-Unterbrechung bei Operatoren, die vollständige Eingabe zur Ergebnisberechnung verlangen:
  - Sortierung, Duplikateleminierung, Gruppierung (GROUP BY), Aggregationsfunktionen etc.
- Pipelines oft sehr kurz (< 10 Operatoren)
- Achtung: Halloween-Effekt



Halloween-Problem

- ist ein Phänomen, das bei einem Datenbankzugriff auftreten kann, bei dem eine Aktualisierung durch eine angeknüpfte Bedingung von dem, was es ändert, abhängt

Beispiel

- UPDATE EMP_TABLE SET SALARY=SALARY*1.1 WHERE SALARY<25000

Problem

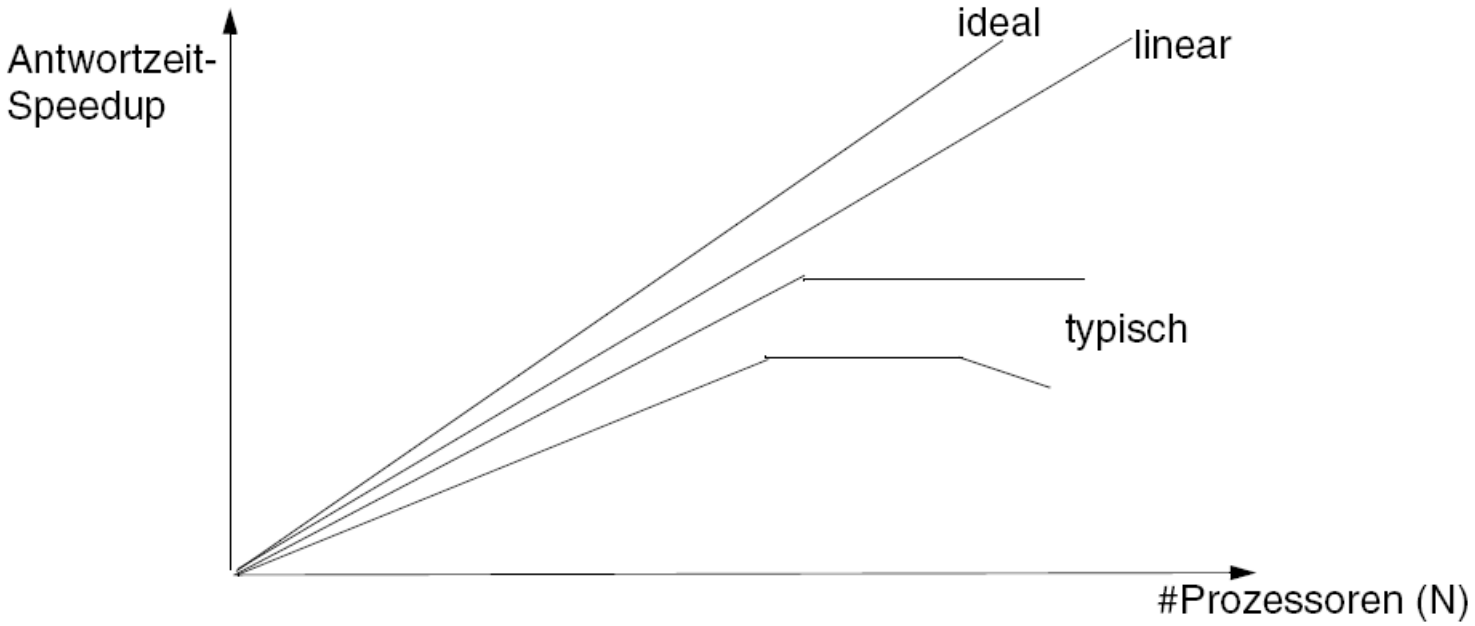
- Gehalt (SALARY) wird so lange erhöhte, bis die Bedingung des Update-Statements nicht mehr greift
- Jeder der unter 25000 verdient, erhält dann 25000

Problem erkannt

- 31.Oktober 1976 (Halloween) im IBM System R Prototypen

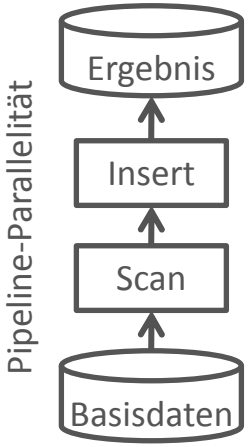
Maximal inhärente Parallelisierung begrenzt

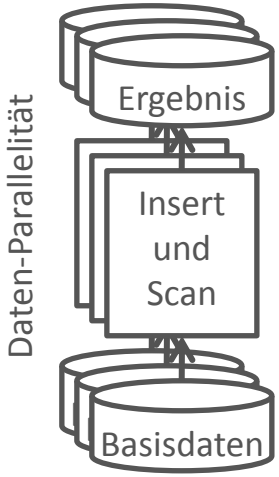
- Amdahls-Law
- Startup- und Terminierungs-Overhead
- Interferenzen bei physischen und logischen Ressourcen
- Varianz (Skew) in den Ausführungszeiten der Teilaufgaben

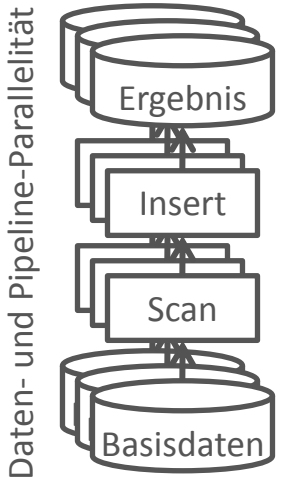


Daten-Parallelität

- basiert auf breiter (horizontaler) Datenverteilung (Declustering)
- parallele Bearbeitung von Teiloperationen auf disjunkten Datenmengen
- Parallelitätsgrad kann mit Datenumfang gesteigert werden







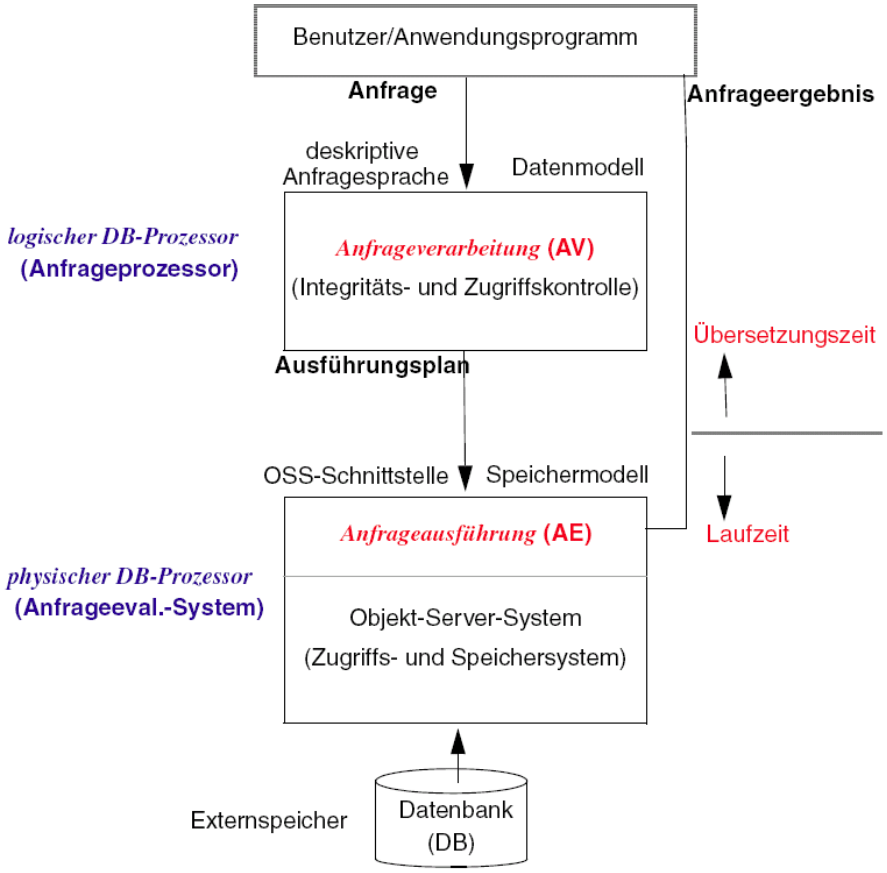
Überblick Anfrageverarbeitung

Anfrageverarbeitung (AV)
(im engeren Sinne)

- Logischer DB-Prozessor
  (System R: relational data system)
- liefert einen Ausführungsplan
  (query execution plan; QEP) zur Übersetzungszeit

Anfrageausführung (AE)

- Physischer DB-Prozessor
  (System R: relational storage system)
- tatsächliche Ausführung des Anfrageplanes zur Laufzeit

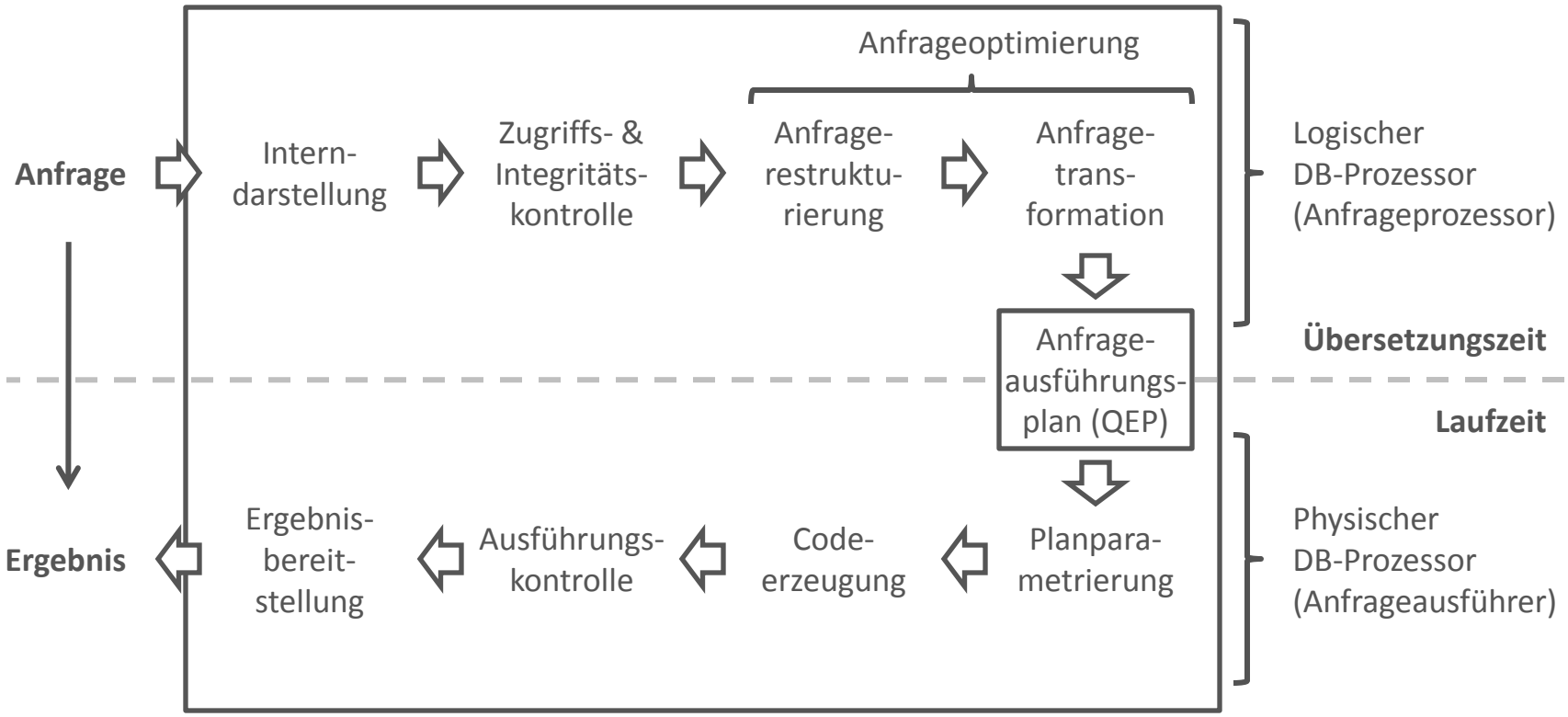


Aufgabe des logischen Datenbankprozessors

- Interndarstellung mit Zugriffs- und Integritätskontrolle
  - Überprüfung der referenzierten Datenbankobjekte wie Tabellen oder Attribute
  - Erweiterung der Anfrage um Operatoren, die bei der Ausführung der Anfrage die strukturelle Konsistenz der Datenbank sicherstellen bzw. deren Verletzung verhindern
- Anfragerestrukturierung (logische Optimierung)
  - Vereinfachung der Anfrage durch algebraische Eigenschaften der Relationenalgebra
  - Umformung auf Schemaebene; unabhängig von der konkreten Ausprägung der Datenbank (z.B. unabhängig von Werteverteilungen, der Existenz von Indexstrukturen etc.)
- Anfragetransformation
  - Zuordnung eines Planoperators zu jedem Operator der relationalen Algebra

Aufgabe des physischen Datenbankprozessors

- Planparametrierung und Codeerzeugung
- Ausführungskontrolle und Ergebnisbereitstellung


Anfrageoptimierung
Anfrage
Intern-darstellung
Zugriffs- & Integritäts-kontrolle
Anfrage-restruktu-rierung
Anfrage-trans-formation
Logischer DB-Prozessor (Anfrageprozessor)
Anfrage-ausführungs-plan (QEP)
Übersetzungszeit
Laufzeit
Ergebnis
Ergebnis-bereit-stellung
Ausführungs-kontrolle
Code-erzeugung
Planpara-metrierung
Physischer DB-Prozessor (Anfrageausführer)

Überführung in Interndarstellung

- Wie ist eine Anfrage intern repräsentiert? -> Operatorengraph
- Lexikalische und syntaktische Analyse
  - Überprüfung auf korrekte Syntax (Parsing)
  - Erstellung eines Anfragegraphen für die nachfolgenden Übersetzungsschritte (Überführung in Interndarstellung)
- Semantische Analyse
  - Feststellung der Existenz und Gültigkeit der referenzierten Relationen und Attribute
  - Ersetzen der externen durch interne Namen (Namensauflösung)
  - Konversion vom externen Format in interne Darstellung

Zugriffs- und Integritätskontrolle

- Durchführung einfacher Integritätskontrollen (Kontrolle von Formaten und Konversion von Datentypen)
- Generierung von Laufzeitaktionen für werteabhängige Kontrollen

Anfragerestrukturierung: algebraische bzw. logische Optimierung

- Anwendung von heuristischen Regeln
- zielt auf globale Verbesserung des Anfragegraphen ab

Anfragetransformation: nicht-algebraische bzw. physischer Optimierung

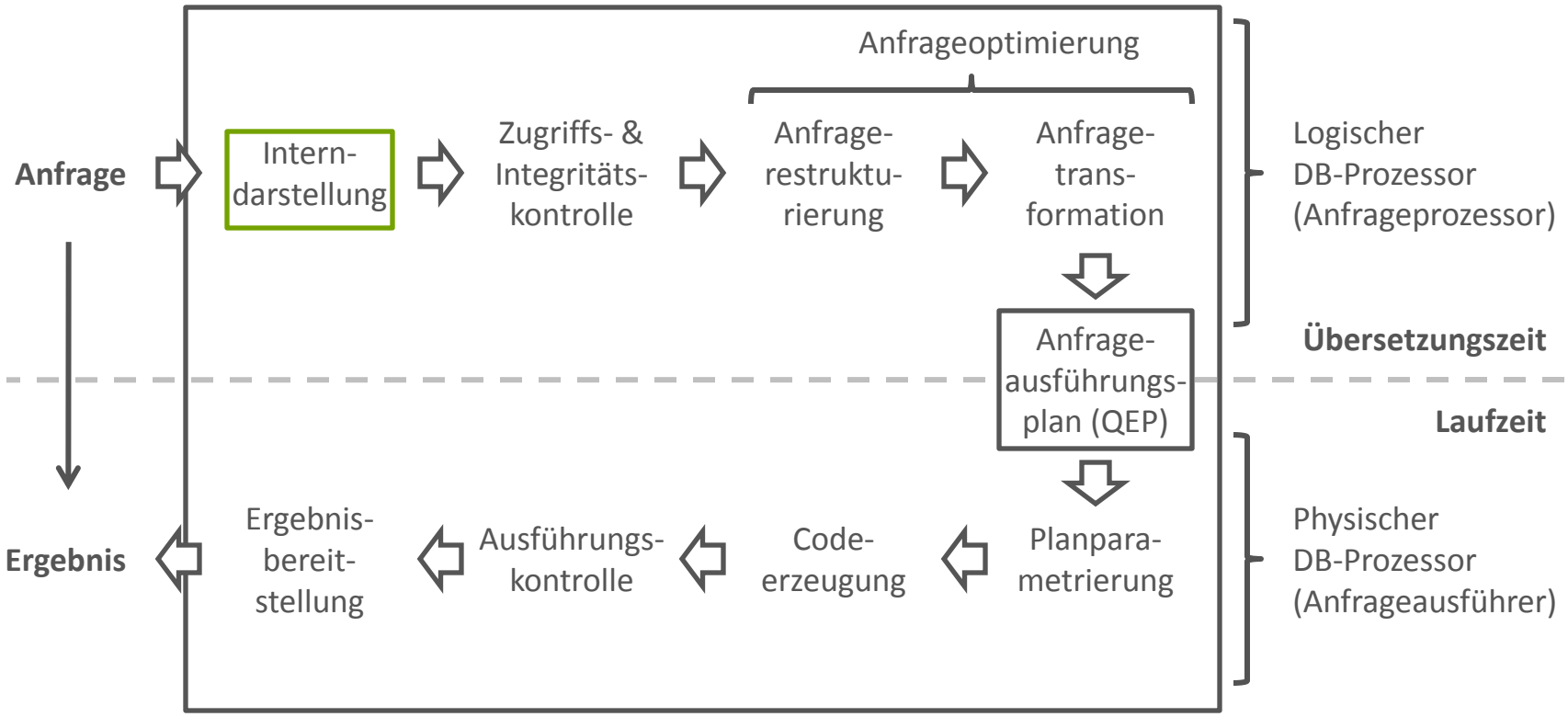
- Berücksichtigung ausführbarer Operationen
- Ersetzung und ggf. Zusammenfassen der logischen Operatoren durch Planoperatoren
- Auswahl der günstigsten Planalternative
  - meist sind mehrere Planoperatoren als Implementierung eines logischen Operators verfügbar
  - meist sind viele Ausführungsreihenfolgen oder Zugriffspfade auswählbar
  - Bewertung der Kosten und Auswahl des günstigsten Ausführungsplanes

Code-Generierung

- Generierung eines zugeschnittenen Programms für die vorgegebene (SQL-) Anfrage
- Erzeugung eines ausführbaren Zugriffsmoduls
- Verwaltung der Zugriffsmodule in einer DBVS-Bibliothek

... zusätzlich bei verteilten Datenbanksystemen ...

- Anfragetransformation, Datenlokalisierung, globale Optimierung


Anfrageoptimierung
Anfrage
Intern-darstellung
Zugriffs- & Integritäts-kontrolle
Anfrage-restruktu-rierung
Anfrage-trans-formation
Logischer DB-Prozessor (Anfrageprozessor)
Anfrage-ausführungs-plan (QEP)
Übersetzungszeit
Laufzeit
Ergebnis
Ergebnis-bereit-stellung
Ausführungs-kontrolle
Code-erzeugung
Planpara-metrierung
Physischer DB-Prozessor (Anfrageausführer)

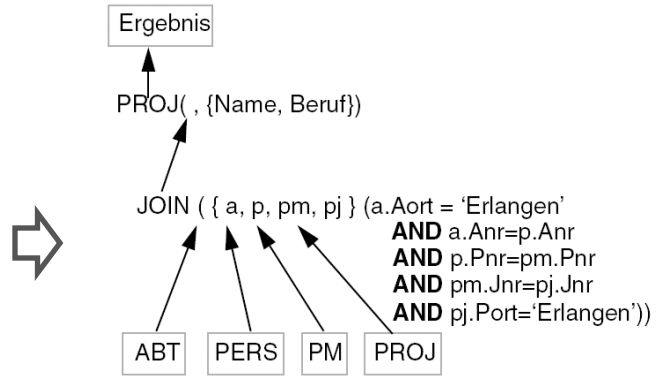
Wie ist eine Anfrage intern repräsentiert?

- Strukturelle Betrachtung: Relationen, visualisiert als Tabellen
- Operationelle Betrachtung: Operatoren der Relationalen Algebra
  - Selektion - Auswahl von ‚Zeilen': $_{pred()}(R)$
  - Projektion - Auswahl von ‚Spalten': $_{\{A1,\ldots,Ak\}}(R)$
  - Gruppierung - Auswahl von ‚Spalten' und Aggregatbildung auf Duplikaten: $_{\{G1,\ldots,Gn:A1,\ldots,Ak\}}(R)$
  - Verbund - Verbinden von Relationen R und S: $R_{P(Ai,Bj)} \bowtie S$
    auf logischer Ebene: n-äre Verbundoperationen
  - Beispiel SQL-Server: ca. 200 logische DB-Operatoren
- Umsetzung in relationalen Operatoren
  - effiziente Datenstruktur mit geeigneten Zugriffsfunktionen
  - prozedurale Darstellung einer deskriptiven, mengenorientierten Anfrage
  - Knoten sind Operatoren der Relationalen Algebra
  - Blattknoten sind (üblicherweise) Relationen
  - gerichtete Kanten repräsentieren den Datenfluss

Beispiel

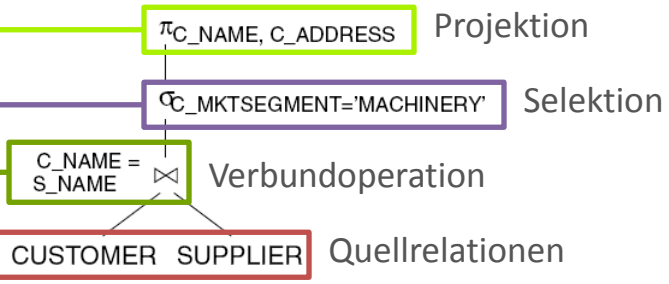
```
SELECT Name, Beruf
  FROM ABT a, PERS p, PM pm, PROJ pj
 WHERE a.Anr = p.Anr AND a.Aort = 'Erlangen'
   AND p.Pnr = pm.Pnr AND pm.Jnr = pj.Jnr
   AND pj.Port = 'Erlangen'
```

Operatorengraph



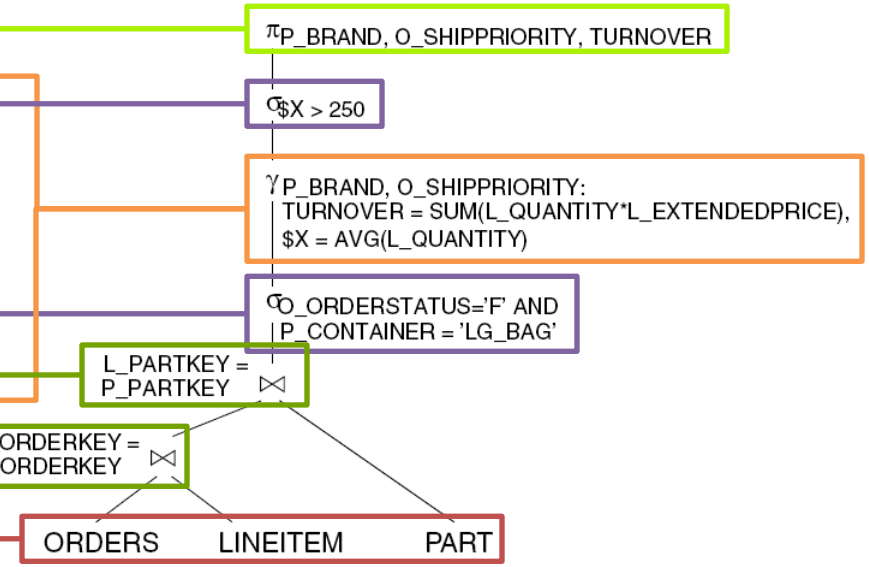
Mono-Block-Anfrage

```
SELECT C_NAME, C_ADDRESS
  FROM TPCD.CUSTOMER, TPCD.SUPPLIER
 WHERE C_NAME = S_NAME
   AND C_MKTSEGMENT = 'MACHINERY';
```



Star-Query

```
SELECT
  P_BRAND, O_SHIPPRIORITY,
  SUM(L_QUANTITY*L_EXTENDEDPRICE)
     AS TURNOVER
FROM
  TPCD.LINEITEM, TPCD.ORDERS, TPCD.PART
WHERE L_ORDERKEY = O_ORDERKEY
  AND L_PARTKEY = P_PARTKEY
  AND O_ORDERSTATUS = 'F'
  AND P_CONTAINER = 'LG BAG'
GROUP BY
  P_BRAND, O_SHIPPRIORITY
HAVING
  AVG(L_QUANTITY) > 250;
```



Query Processing

- Query compilation
- Query execution

Phases of query processing

- Internal representation
- Access and integrity control
- Query restructuring
- Query transformation
- Code generation

Literature

- Härder, T. & Rahm, E. Datenbanksysteme: Konzepte und Techniken der Implementierung. Springer-Verlag, 1999
- Saake, G.; Heuer, A. & Sattler, K.-U. Datenbanken: Implementierungstechniken. MITP-Verlag, 2005
- Hellerstein, J. M.; Stonebraker, M. & Hamilton, J. R. Architecture of a Database System. Foundations and Trends in Databases, 2007, 1, 141-259
- Graefe, G. Query Evaluation Techniques for Large Databases. ACM Computing Surveys, 1993, 25, 73-170
- Jarke, M. & Koch, J. Query Optimization in Database Systems. ACM Computing Surveys, 1984, 16, 111-152