



Projekt	Objekt-Relationale Datenbanken
Autoren	Rituraj Singh - 19539 AmirHossein Roshanzadeh -19660 Saeide Dana - 19XXX
Kurs	Datenbanken und Informationssysteme (INFM1200)
Semester	Sommersemester 2021
Firma	Hochschule Stralsund
Telefon	+4917643627091
Eingereicht bei	Prof. Dr. Kay-Michael Otto
Eingereicht am	22.04.2021

Zusammenfassung

In diesem Dokument haben wir versucht, alle wichtigen Details zu objekt relationale Datenbankmanagementsysteme zu erklären, die auch für einen Einsteiger leicht verständlich sind. Das Dokument verdeutlicht die Unterschiede zwischen rein relationalen und objektrelationalen Datenbankmanagementsystemen und wie man letztere in der Praxis einsetzen kann. Um die Dinge richtig zu verstehen, muss man sich die Hände schmutzig machen, indem man tatsächlich einen Code implementiert. Daher haben wir versucht, eine kleine Beispielanwendung mit Java und PostgreSQL zu erstellen, um Datenbankzugriff zu bekommen. Letztendlich haben wir versucht, einige Fragen angemessen zu beantworten die beim Lernen über objekt relationale Datenbankmanagementsysteme auftreten können.

1 INHALTSVERZEICHNIS

2 LERNZIELE DER DOKUMENTATION

2.1 Übersicht über ORDBMS und RDBMS.....✓	3
2.2 Erweiterungen des objektrelationalen Modells.....✓	3
2.3 Funktionen von ORDBMS.....✓	3
2.4 Unterscheidung zwischen ORDBMS und RDBMS.....✓	3
2.5 Relevanz des objektrelationalen Modells in der Praxis.....	3
2.6 objektrelationalen Erweiterungen der SQL.....✓	3
2.7 Erfahrung im Umgang mit einer ORDBMS.....	3
2.7.1 Implementierung mit rein relationaler Datenbank.....	3
2.7.2 Aussehen der ORDBMS -Abfragen.....	3
2.7.3 Implementierung mit ORDBMS.....	3
2.7.4 Kleine Beispielanwendung mit Java und Datenbankzugriff.....	3
3 Interessante Fragen zu ORDBMS.....	3
3.1 Anwendungsbereiche.....	3
3.2 Stärken und Schwächen.....	3
3.3 Optimierungsmöglichkeiten von Abfragen.....	3
3.4 angebotene Schnittstellen zu Programmiersprachen.....	3
3.5 Schwerpunkt bei der Betrachtung des CAP-Theorems.....	3
4 Fazit.....	3
5 Anhang.....	3
6 Literaturverzeichnis.....	3

2.1 Übersicht über RDBMS und ORDBMS.....✓

RDBMS : Das relationale Datenbankverwaltungssystem ist ein Datenbankverwaltungssystem, das auf einem relationalen Datenmodell basiert. Dieses Modell verwendet das Konzept der mathematischen Beziehung und sein Baustein sind Tabellen und Werte.

- Eine relationale Datenbank enthält eine Beziehung oder eine Tabelle mit der Bezeichnung STUDENT mit Tupeln oder Datensätzen, die Informationen zu jedem Schüler enthalten.
- Pitfall : Komplexe Objekte können nicht beschrieben werden.
- Die Datensätze verwenden die in SQL-92 enthaltenen Standarddatentypen (wie varchar, float, int). Daher klassifizieren wir diese Art von Daten als "einfach".
- MS SQL Server, MySQL, SQLite, MariaDB sind einige Beispiele für RDBMS

ORDBMS : Das objektrelationale Datenbankverwaltungssystem ist ein Datenbankverwaltungssystem, das auf dem relationalen Modell und dem objektorientierten Datenbankmodell basiert.

- **Objektorientiertes Datenbankmodell** ermöglicht die Verwendung objektorientierter Programmiersprachentechniken in der Datenverwaltung. Ein objektorientiertes Datenbankmodell enthält eine Klasse, z. B. STUDENT, mit einem Objekt, das Informationen zu jedem Schüler enthält.
- Da ORDBMS Funktionen sowohl des relationalen Modells als auch des objektorientierten Datenbankmodells aufweist. Es kann sowohl als RDBMS als auch als ORDBMS verwendet werden.
- Das objektrelationale Datenbankverwaltungssystem versucht, die Lücke zwischen relationalen Datenbanken und den objektorientierten Modellierungstechniken zu schließen, die in Programmiersprachen wie Java, C ++, Visual Basic .NET oder C # verwendet werden.
- Oracle Database, PostgreSQL und Microsoft SQL Server sind einige Beispiele für RDBMS

2.2 Erweiterungen des objektrelationalen Modells.....✓

Erweiterungen des objektrelationalen Modells gegenüber dem rein relationalen Modell :

- Das objektrelationale Modell erweitert die Funktionen des relationalen Modells, damit Objekte in den Spalten einer relationalen Datenbank gespeichert werden können. Ein objektrelationales DBMS wird manchmal als hybrides DBMS bezeichnet.
- Der hybride DBMS-Ansatz wurde 1990 von Michael Stonebraker vorgeschlagen
- Das objektrelationale Datenmodell wurde mit dem Wunsch nach Datenbanken vorgeschlagen, in denen komplexere Entitäten und Regeln gespeichert werden können, die jedoch alle Stärken einer reinen relationalen Datenbank beibehalten.
- Das objektrelationale Datenmodell ist eine Erweiterung des relationalen Modells mit folgenden Funktionen:
 - Ein Feld kann ein Objekt mit Attributen und Operationen enthalten.
 - Komplexe Objekte können in relationalen Tabellen gespeichert werden

Beispielsweise unterstützt PostgreSQL die Möglichkeit, einen neuen benutzerdefinierten Datentyp zu deklarieren, der im Grunde eine Klasse ist, und diesen neuen Typ als Datentyp für eine Tabellenspalte festzulegen.

```

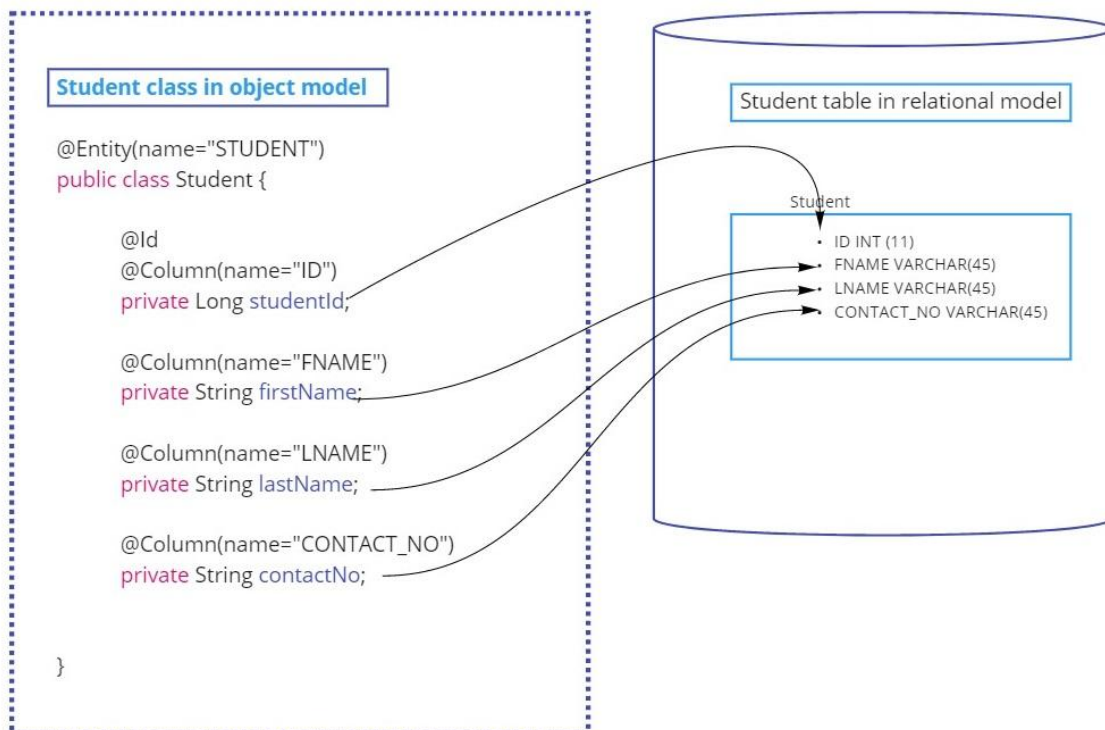
create table slides (
  id      int,
  date    date,
  caption document,
  picture photo_CD_image);

create table landmarks (
  name    varchar(30),
  location point);

```

User - defined datatypes

Abbildung 1 : Erläuterung : document , photo_CD_image , point sind Benutzerdefinierte abstrakte Datentypen . Jede Folie hat eine Kennung, das Aufnahmedatum, die oben genannte Beschriftung und die digitalisierten Bits im Kodak Photo-CD-Format. Punkt gibt die geografische Position jeder Folie an



mi

Abbildung 2 : Wie komplexe Objekte in relationalen Tabellen gespeichert werden.

2.3 Funktionen von ORDBMS.....✓

ORDBMS verfügen über vier Funktionen, die sie von RDBMS unterscheiden:

- **Benutzerdefinierte abstrakte Datentypen (ADTs)** - ADTs ermöglichen neue Datentypen mit Strukturen, die für bestimmte zu definierende Anwendungen geeignet sind.
- **Benutzerdefinierte Routinen (UDRs)** - UDRs bieten die Möglichkeit, benutzerdefinierte Serverfunktionen zu schreiben, die einen Großteil der in C zum Ausdruck gebrachten Leistung und Funktionalität aufweisen.
- **"SmartBLOBs"** - Hierbei handelt es sich um festplattenbasierte Objekte mit der Funktionalität von Dateien mit wahlfreiem Zugriff. ADTs verwenden sie, um Daten zu speichern, die nicht in eine Tabellenzeile passen.
- **Erweiterbare und flexible Indizierung**- Die R-Tree- oder GiST-Indizierung (Generalized Search Tree) für mehrdimensionale Daten ermöglicht die schnelle Suche nach bestimmten ADTs in einer Tabelle. Die herkömmliche B-Baum-Indizierung eignet sich für Daten, die in einer Dimension geordnet werden können (skalare Zahlen oder Textzeichenfolgen), für mehrdimensionale Daten (z. B. Polygone) ist jedoch ein R-Baum- oder GiST-Index besser geeignet.

Ein Beispiel für die Verwendung einiger der oben genannten Funktionen ist:

Sum the "area" of all "lakes" "contained in" "British Columbia"

- "area" = UDR
- "lakes" = ADT
- "contained in" = R-Tree
- "British Columbia" = Instance of ADT

2.4 Unterscheidung zwischen ORDBMS und RDBMS.....✓

	RDBMS	ORDBMS
Standard Definition	SQL2	SQL3/4 (in process)
Unterstützung für OOP	schlecht	Beschränkt hauptsächlich auf neue Datentypen
Idee	Basiert auf der Idee der Normalisierung, bei der redundante Daten aus Ihren Tabellen entfernt werden, um die Speichereffizienz, Datenintegrität und Skalierbarkeit zu verbessern.	Basiert auf der Idee, RDBMS objektorientierte Unterstützung anzubieten, wie: <ul style="list-style-type: none">• Vom Benutzer erweiterbare Typen• Verkapselung• Erbe• Polymorphismus
Sprache	SQL-92 als Datendefinitionssprache (DDL)	ORDBMS erfordert eine Abfragesprache, die

	und Datenmanipulationssprache (DML).	mindestens benutzerdefinierte Funktionen und Operatoren zulässt. Die erste Standardversion von SQL mit diesen Funktionen ist SQL-3
Komposition	Zweidimensionale Tabellen mit Zeilen und Spalten, die verwandte Tupel oder Datensätze enthalten. Dies ist eine schlechte Darstellung von Entitäten der „realen Welt“	Versucht, diesen Nachteil zu überwinden, indem ein Feld oder eine Spalte Objekte als Werte enthält.
Umfang der verarbeiteten Daten	"Einfache" Daten werden gemäß den SQL-92-Standards behandelt	Bietet Unterstützung für komplexe Datenverarbeitungen (wie Bilder, Videos usw.), die in RDBMS nicht möglich waren
Eindeutige Kennung	Primärschlüssel	Objektkennung (OID)
Visualisierung	ER-Diagramm wird bevorzugt	Klassendiagramm wird bevorzugt
Produktreife	Sehr mündig	Unreif; Erweiterungen sind neu und relativ unbewiesen
Kosten	Nicht so kostenintensiv	Kostenintensiv
Indizierung - Techniken	B-Tree für den schnellen Zugriff auf skalare Daten	Generics B-Tree oder R-Tree für den schnellen Zugriff auf 2/3-dimensionale Daten

2.5 Relevanz des objektrelationalen Modells in der Praxis

Das folgende Diagramm (aus S. 2, Objektrelationale DBMS - Verfolgung der nächsten großen Welle, M. Stonebraker und P. Brown, Morgan Kaufmann, 1999) klassifiziert Datenbanken nach Verwendung.

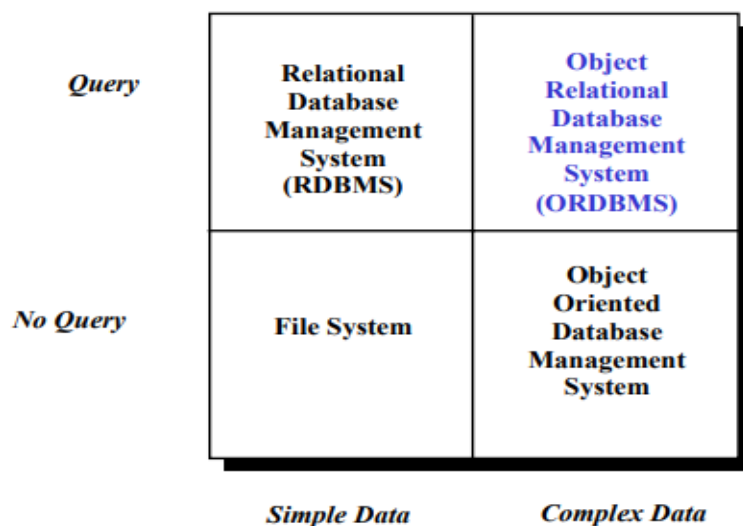


Abbildung 4 : einfache Datenbank Klassifizierungsmatrix.

Dies zeigt an, dass die Verwendung eines ORDBMS bestimmte Vorteile bietet, wenn Abfragen in einer Datenbank mit komplexen Daten durchgeführt werden sollen.

Von benutzerdefinierten Objekten können erstellt werden. Und dann gibt es neben benutzerdefinierten Datentypen auch eingebaute Datentypen, die von PostgreSQL unterstützt werden und heutzutage für den Umgang mit Daten sehr nützlich sind. Erklären Sie beispielsweise, wie monetäre Joins-Datentypen in der realen Welt hilfreich sein können Szenario.

2.6 objektrelationalen Erweiterungen der SQL.....✓

Vor SQL: 1999 gab es keine Unterstützung für benutzerdefinierten Typ (UDT), Sammlung, typisierte Tabellen und andere Funktionen, die heute unter objektrelationalen Bezeichnungen enthalten sind. Dann führte SQL: 1999 das objektrelationale SQL-Modell ein, das zwei unterschiedliche Komponenten aufweist:

benutzerdefinierte Typen und typisierte Tabellen (Tabellen, die mit einem benutzerdefinierten strukturierten Typ definiert sind).

Es gibt drei Arten von **benutzerdefinierte Typen**:

1. **einzelner integrierter Datentyp** (z. B. INTEGER, VARCHAR), deren Werte in Operationen nicht direkt mit diesem integrierten Typ oder anderen unterschiedlichen Typen gemischt werden können; Sie können wie jeder integrierte SQL-Typ zum Definieren von Spalten verwendet werden
2. **strukturierte benutzerdefinierte Typen als Werte**, B. Adresstyp, bestehend aus Straße, Nummer, Stadt, Landkreis, Postleitzahl.
3. **strukturierte benutzerdefinierte Typen als Objekte**. In einer typisierten Tabelle ist jede Zeile eine Instanz eines strukturierten benutzerdefinierten Typs als Objekt. Die Tabelle enthält eine Spalte für jede Eigenschaft (Attribut) des benutzerdefinierten Typs, für den die Tabelle definiert ist, sowie eine selbstreferenzierende Spalte, die die eindeutige Objektkennung (OID) darstellt.

Folglich gibt es in neuen SQL-Standards mehrere Kategorien von Typen:

1. **grundlegende Datentypen**, als atomar oder intrinsisch betrachtet.
2. **abstrakte Datentypen** (mit ihren eigenen Komponenten oder Attributen, Identität, Vererbung, Polymorphismus)
3. **Sammlungsarten** (Zeilentyp, Listentyp, Satztyp, Multiset-Typ). Diese Artenkategorien werden hauptsächlich zum Definieren der Attributtypen aus den Überschriften der Tabellen verwendet.

Wir werden uns in den nächsten Kapiteln einige der oben genannten Ideen ansehen und

wir sind damit beschäftigt, den Code zu implementieren.

2.7 Erfahrung im Umgang mit einer ORDBMS

2.7.1 Implementierung mit rein relationaler Datenbank

2.7.2 Implementierung mit ORDBMS

2.7.3 Kleine Beispielanwendung mit Java und Datenbankzugriff

3.1 Anwendungsbereiche

- Bioinformatik.
- Genomics
- Ozeanobservatorien
- Wettermodellierung und Vorhersage
- Proteomics.
- Diagnose.
- Wirkstoffentdeckung.
- Biomarker-Erkennung.
- Wechselwirkungen mit anderen Medikamenten.
- Gentechnik.
- Data Mining zur Erkennung von Mustern und zur Bereitstellung von Erkenntnissen.

3.2 Stärken und Schwächen

VORTEIL DES OBJEKTORIENTIERTEN DATENBANKMODELLS:

- Wiederverwendung und Freigabe: Die Hauptvorteile der Erweiterung des relationalen Datenmodells liegen in der Wiederverwendung und Freigabe. Die Wiederverwendung ergibt sich aus der Möglichkeit, den DBMS-Server so zu erweitern, dass Standardfunktionen zentral ausgeführt werden, anstatt sie in jeder Anwendung codieren zu lassen.
- Erhöhte Produktivität: ORDBMS bietet sowohl für den Entwickler als auch für den Endbenutzer eine höhere Produktivität
- Der erweiterte relationale Ansatz bewahrt den bedeutenden Wissens- und Erfahrungsschatz, der für die Entwicklung relationaler Anwendungen aufgewendet wurde. Dies ist ein erheblicher Vorteil, da das Ändern für viele Unternehmen unerschwinglich teuer

wäre. Ein weiterer Hauptvorteil dieser objektrelationalen Datenbanken ist eine hohe Wartbarkeit, da die objektorientierte Struktur das Datenbankdesign anstelle des herkömmlichen relationalen Datenbankmodells problemlos beibehalten kann

NACHTEIL DES OBJEKTORIENTIERTEN DATENBANKMODELLS:

- Der ORDBMS-Ansatz hat den offensichtlichen Nachteil der Komplexität und der damit verbundenen erhöhten Kosten
- Es gibt Befürworter des relationalen Ansatzes, die glauben, dass die wesentliche Einfachheit und Reinheit der relationalen Modelle bei diesen Arten von Erweiterungen verloren geht.
- ORDBMS-Anbieter versuchen immer, Objektmodelle als Erweiterungen des relationalen Modells mit einigen zusätzlichen Komplexitäten darzustellen. Diese Möglichkeit verfehlt den Punkt der Objektorientierung und unterstreicht die große semantische Lücke zwischen diesen beiden Technologien.

3.3 Optimierungsmöglichkeiten von Abfragen

4 Literaturverzeichnis

<https://www.youtube.com/watch?v=BqHILMEY9no>

<https://www.youtube.com/watch?v=zPR7Oy2yemU&t=124s>

https://www.youtube.com/watch?v=nyQ_MVmKcgE&list=PLk1kxccoEnNHIA_R2ggnzlkOc7jxql-w2

<https://www.youtube.com/watch?v=OLmAZmBSwMo>

df

[https://www.postgresql.org/files/documentation/pdf/13/postgresql-13-A4.p](https://www.postgresql.org/files/documentation/pdf/13/postgresql-13-A4.pdf)

https://docs.oracle.com/cd/B19306_01/appdev.102/b14260/adobjxmp.htm

<https://www.postgresql.org/docs/9.5/datatype.html>

https://link.springer.com/content/pdf/10.1007%2F978-1-4302-0176-2_3.pdf

<https://pediaa.com/what-is-the-difference-between-rdbms-and-ordbms/#:~:text=The%20main%20difference%20between%20RDBMS.store%20and%20manage%20data%20efficiently.&text=ORDBMS%20has%20features%20of%20both%20RDBMS%20and%20OODBMS.>

Anwendungsbereich :

<https://www-356.ibm.com/partnerworld/gsd/showimage.do?id=25870>

<https://www.databasejournal.com/features/db2/how-do-oodbms-and-ordbms-differ-from-rdbms.html>

<https://www.geeksforgeeks.org/difference-between-rdbms-and-oodbms/>

<https://www.tutorialspoint.com/Object-relational-Data-Model#:~:text=An%20Object%20relational%20model%20is,data%20types%2C%20tabular%20structures%20etc.>

<https://www.tutorialspoint.com/difference-between-rdbms-and-oodbms>

<https://www.yumpu.com/en/document/read/4032215/comparison-of-rdbms-oodbms-and-ordbms>

<https://www.slideshare.net/BaabtraMentoringPartner/difference-between-rdbms-oodbms-and-ordbms>

https://miro.com/app/board/o9J_lIZPVCg=/

https://en.wikipedia.org/wiki/Object%E2%80%93relational_database

<https://www.slideshare.net/BaabtraMentoringPartner/difference-between-rdbms-oodbms-and-ordbms>

<https://dsf.berkeley.edu/papers/Informix/www.informix.com/informix/corpinfo/zines/whitpprs/illuswp/wave.htm>

<https://pediaa.com/what-is-the-difference-between-rdbms-and-ordbms/>

<https://www.geeksforgeeks.org/difference-between-rdbms-and-ordbms/>

file:///D:/HOST-INFM%20Degree/Hausaufgaben/Semester-2/DBMS/Object-Relational_Databases_An_Area_with_Some_Theo.pdf

<https://www.learndb.com/oracle/main-features-and-advantages-object-relational-database-systems-ordbms>

<https://www.postgresql.org/docs/7.4/jdbc-binary-data.html>

<https://www-356.ibm.com/partnerworld/gsd/showimage.do?id=25870>

file:///D:/HOST-INFM%20Degree/Hausaufgaben/Semester-2/DBMS/Object-Relational_Databases_An_Area_with_Some_Theo.pdf

EXTRA

Du kannst hier deine Zusammenfassung hinzufügen , später werde ich prüfen und in der Dokumentation einordnen.

(vilen Dank) Bist du da ?

Datenmodelle

Erste Generation

Hierarchisches Datenmodell

Die bekannteste hierarchische Datenbank ist das Information Management System (IMS) von IBM, das es seit den 1960er Jahren gibt und das noch immer von IBM unterstützt wird.

Netzwerkdatenmodell

Es wurde eine wichtige Funktion für die Datenmodellierung hinzugefügt. Mehrfache Abstammung bedeutet, dass sich eine einzelne Entität an den „vielen“ Enden mehrerer Beziehungen befinden kann.

Zweite Generation

Beziehungsmodell

Zu Beginn werden Datenentitäten durch einfache tabellarische Strukturen dargestellt, die als Beziehungen bezeichnet werden.

Entitätsbeziehungen und Datenintegrität werden durch Primärschlüssel und Fremdschlüssel definiert.

Der Entwurf einer relationalen Datenbank basiert auf der Idee der Normalisierung, dem Entfernen redundanter Daten aus Ihren Tabellen, um die Speichereffizienz, Datenintegrität und Skalierbarkeit zu verbessern.

Datenzugriff im relationalen Modell

Der Datenzugriff verwendet eine nicht prozedurale Hochsprache (SQL). Dies macht relationale Datenbanken ideal für Ad-hoc-Abfragen.

Verwenden des relationalen Modells mit objektorientierten Programmen

versuchen objektorientierte Systeme, die Problemdomäne in Bezug auf Objekte zu modellieren. Die Entitäten im System werden im Klassendiagramm und nicht im Entitätsbeziehungsdiagramm (ERD) des herkömmlichen Systems beschrieben. Plötzlich ist die Datenbank nicht mehr der Treiber für die Anwendung. Stattdessen besteht ihre Hauptfunktion darin, der Anwendung einen Dienst bereitzustellen, nämlich die Fähigkeit, Objekte beizubehalten.

Beispiele für Beziehungen, die in letzteren schwer zu modellieren sind. Beispielsweise wird die Vererbung im relationalen Modell nicht direkt unterstützt.

Das Problem der eingebetteten SQL-Zeichenfolgen gilt auch für objektorientierte Programme, möglicherweise mehr, da Tools, die ein sicheres und einfaches Refactoring unterstützen, diese nicht verstehen oder ändern können.

Dritte Generation

„Postrelationale“ Modelle

Die Modelle der dritten Generation, die erstmals in den 1980er Jahren vorgeschlagen wurden, sind eine Antwort auf die Probleme, die häufig auftreten, wenn ein objektorientiertes System mit einer relationalen Datenbank verbunden wird.

Die objektorientierte dritte Generation hat sich in zwei verschiedene Richtungen entwickelt:

1- Das Objekt Datenmodell

Das Objektdatenmodell entspricht weitgehend dem objektorientierten Paradigma, außer dass die Objekte persistent sind. Das heißt, sie bleiben auch nach Abschluss des Programmlaufs bestehen.

Das Objektdatenmodell ist eine Navigation. Der Objektzugriff folgt genau definierten Beziehungen, wie im Entwurfsmodell angegeben.

Für den Datenzugriff werden weniger Codezeilen benötigt, und die Datenbankkonfiguration wird reduziert oder eliminiert. Das Ergebnis ist eine höhere Entwicklerproduktivität. Die Leistung kann auch für Abfragen, die den definierten Beziehungen folgen, erheblich verbessert werden.

2- Das objektrelationale Modell

Das objektrelationale Modell erweitert die Funktionen des relationalen Modells, damit Objekte in den Spalten einer relationalen Datenbank gespeichert werden können. Ein objektrelationales DBMS wird manchmal als hybrides DBMS bezeichnet.

Das Ziel von Erstellung Object-Relational Model

Die Motivation war der Wunsch nach Datenbanken, in denen komplexere Entitäten und Regeln gespeichert werden konnten, die jedoch alle Stärken der Datenbanken der zweiten Generation beibehielten.

Das objektrelationale Datenmodell ist eine Auswertung relationaler Modelle mit folgenden Funktionen:

- Ein Feld kann ein Objekt mit Attributen und Operationen enthalten.
- Komplexe Objekte können in relationalen Tabellen angegeben werden

Der objektrelationale Datenansatz ist so ziemlich das Gegenteil von den Objektdatenbanken. Anstatt einem objektorientierten System Persistenzfähigkeiten zu verleihen, bietet es objektorientierte Funktionen innerhalb der Datenbank.

Arbeitsaufteilung

Rituraj Singh (Mat.Nr.: 19539):

- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXXXXXXX

Amirhossein Roshanzadeh (Mat.Nr.: 19660):

- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXXXXXXX

Saeide Dana(...)

- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXXXXXXX