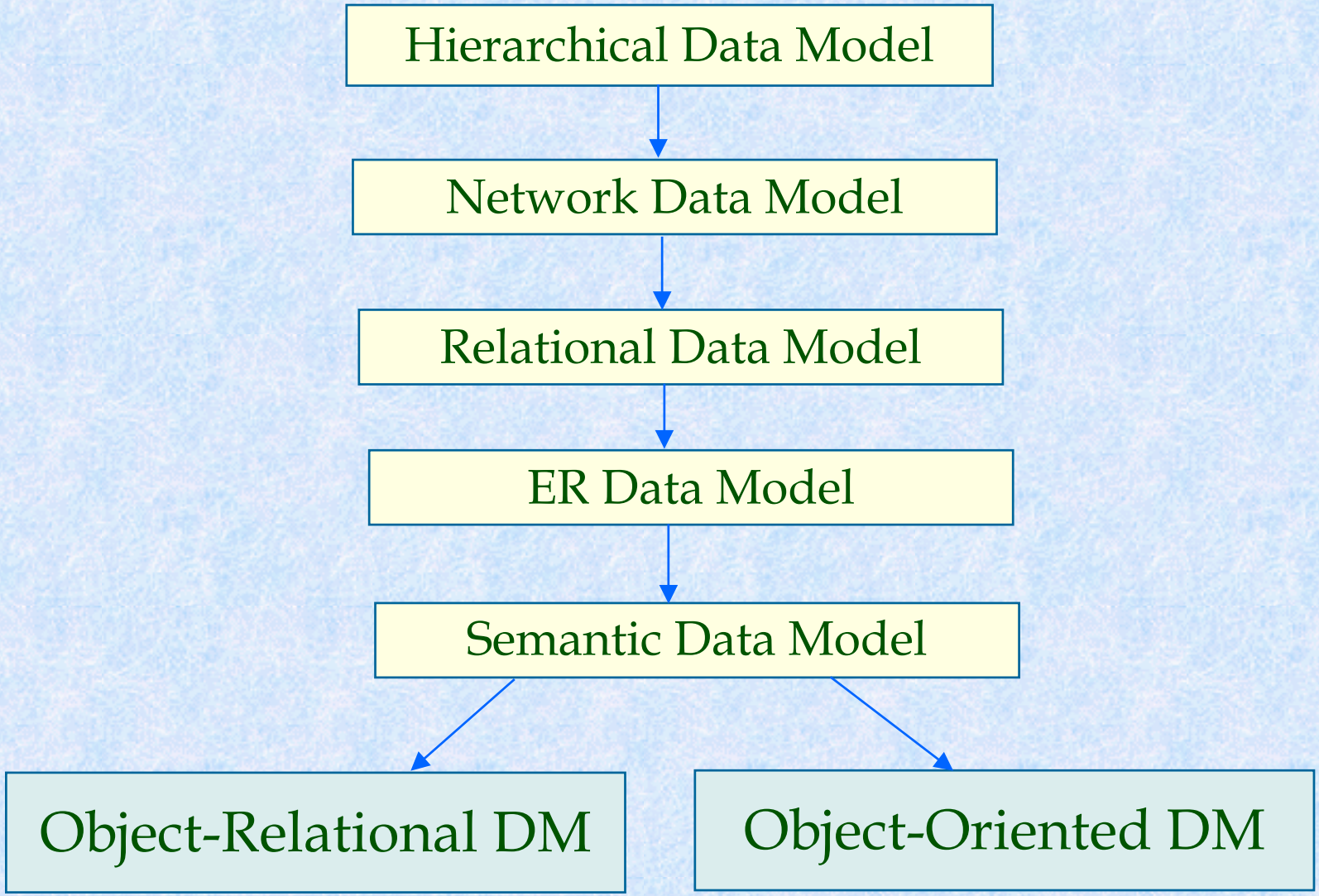
Overview RDBMS-ORDBMS-OODBMS

Database Models – Transition



Main Issues

- Relational DBMS Limitations
- What is an OODBMS?
- Advantages and Disadvantages of OODBMS
- What is an ORDBMS?
- What is SQL3?
- Comparison of OODBMS and ORDBMS
- When to use an OODBMS
- When to use an ORDBMS

Relational DBMS Limitations

- Semantic overloading.
- Poor representation of ‘real world’ entities
- Poor support for integrity & business constraints.
- Homogeneous data structure.
- Limited operations.
- Difficulty handling recursive queries.
- Difficulty with ‘Long Transactions’.

Relational DBMS Limitations

- Normalisation (Normal Forms and FDs) sometimes lead to relations which do not exist, or correspond, to entities in the real world. This compounds on the ‘join’ feature of query processing
- The many to many relationship is difficult to express.
- The RDBMS has domains, keys, multi-valued and join dependencies

OODBMS Architecture

OBJECT ORIENTED DATABASE MANAGEMENT SYSTEM

OOPL	**DBMS**
▪Complex Objects	Persistence
▪Object Identity	Disc Management
▪Methods & Messages	Data Sharing
▪Inheritance	Reliability
▪Polymorphism	Security
▪Extensibility	Ad Hoc Querying
▪Computational Completeness	

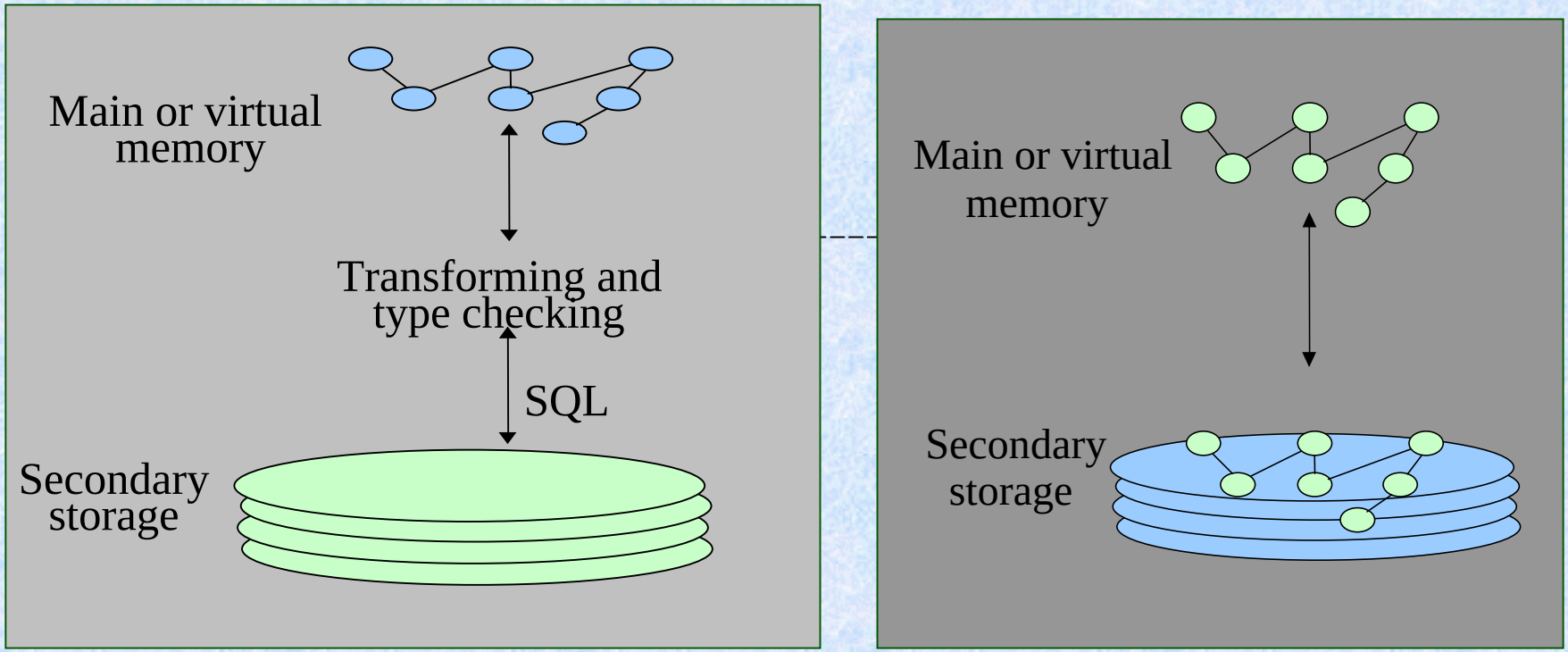
OODBMS Main Features

- Complex objects
- Object Identity
- Encapsulation
- Classes
- Inheritance
- Overriding and late-binding
- Extensibility
- Computational completeness
- Persistence
- Concurrency
- Recovery
- Ad-hoc querying

Storage-Levels

RDBMS Storage

ODBMS Storage



OODBMS Advantages

- Enriched modelling capabilities
- Extensibility
- Support for schema evolution.
- Applicable for advanced database applications
- Improved performance.

OODBMS Disadvantages

- Lack of a universal data model ?
- Ad-hoc querying compromises encapsulation.
- Locking at object-level impacts performance
- Complexity
- Lack of support for views
- Lack of support for security

Then What is an ORDBMS ?

- An Object-Relational database adds features associated with object-oriented systems to a RDBMS

OR

- Extend the relational data model by including object orientation and constructs to deal with added data types

Object-Relational DBMS Features

OODBS support noted by RDBMS vendors include

- User-extensible type system
- Encapsulation
- Inheritance
- Polymorphism
- Dynamic binding of methods
- Complex objects including first normal form objects
- Object Identity

ORDBMS Drawbacks

Disadvantages of ORDBMS

- Complexity
- Increased costs
- Unclear if the ORDBMS will actually combine relationships and encapsulated objects to correctly and completely mirror the ‘real world
- Provision of a language(s) which will front end to SQL and will provide a migration path for existing SQL users

ORDBMS-SQL3

SQL3 is a superset of SQL/92, in that it supports all of the constructs supported by that standard, as well as adding new ones of its own.

New Types

- Extended Base Types.
- Row Types.
- User-Defined Types.
- User-Defined Routines.
- Sub-Types and Super-Types.
- Sub-Tables and Super-Tables.
- Reference Types and Object Identity.
- Collection Types.

OODBMS or ORDBMS ?

- OODBMS put more emphasis on the role of the client side This can improve long, process intensive, transactions.

- ORDBMS SQL is still the language for data definition, manipulation and query

- OODBMS have been optimised to directly support object-oriented applications and specific OO languages

- ORDBMS are supported by most of the ‘database vendors’ in the DBMS market place

OODBMS or ORDBMS ?

- ORDBMS Most third-party database tools are written for the relational model and will therefore be compatible with SQL3

- ORDBMS search, access and manipulate complex data types in the database with standard (SQL3 ?), without breaking the rules of the relational data model

- OODBMS The ODMG standard group's OQL is now the standard query language amongst OODBMS vendors

OODBMS or ORDBMS ?

When to use an ODBMS?

- In applications that generally retrieve relatively few (generally physically large) highly complex objects and work on them for long periods of time.

When to use an ORDBMS?

- In applications that process a large number of short-lived (generally ad-hoc query) transactions on data items that can be complex in structure

Concluding Remarks

- OODBMS:

  Abandon SQL (use an OO language instead)

- ORDBMS:

  Extend SQL (with OO features)

Concluding Remarks

- Object-oriented capabilities have in effect turned the relational model and the process of normalization on its head.

- Various nested objects and/ or references can be queried without doing a join.

- A shift from the relational to the object-oriented model is taking place ?