Is Object Oriented model always better for DBMS?

CSCI 5708 Architecture and Implementation of Database Management Systems (Spring 2018)

G16 — Hao Fei Cheng, Limu Xiao

What is Object Relational Database?

- Object-Relational databases (ORDBMS) extend the Relational Database (RDBMS) to add object oriented features
- Commonality with relational Database: SQL queries, queries optimization
- Added to RDMBS:
  - Objects
  - Inheritance in database schemes and query languages
  - Support customized data types and methods

Bridge between RDBMS and the obsolete object-oriented database (OODBMS)
Object model comparison: Java vs ORDBMS

- **Common**
  - User defined types
  - Methods
  - Single inheritance (subtype inherits from supertype)
  - Encapsulation (public, private and protected attributes)
  - Polymorphism

- **Difference**
  - Stored objects in ORDBMS vs. transient objects in Java
  - Dilemma of encapsulation: Yes (for ORDBMS) vs, No (for Java)
Why encapsulation

- Simpler modifications.
  - Encapsulation minimizes a method’s dependencies on other classes.
- Less programming complexity.
  - It restricts the number of types the programmer must be aware of when writing a method.

Figure 1  UML class model for an airline flight reservation system
Tradeoff of encapsulation

- To achieve high performance in a RDMBS:
  - Generate optimized execution plan through optimizer
  - Optimizer would join multiple tables together in a single SQL statement
  - Broadly stated queries allow higher freedom for the optimizer

- Encapsulation encourages:
  - Minimize the combinations of tables in queries
  - Disperse the tables across multiple statements.

Contradictions!
In a RDBMS application, should a programmer encapsulate the code when
- Program is complex and performance doesn’t decrease severely after encapsulation
- Program is easy and broadly stated queries have good improve the performance
- Program is easy and broadly stated queries lead to poorer query performance
Quiz

CREATE TYPE PERSON_TYPE AS (
    NAME VARCHAR(35),
    SEX CHAR,
    BIRTH_DATE DATE,
    PHONES USA_PHONE_TYPE ARRAY [4],
    ADDR USA_ADDR_TYPE
) INSTANTIABLE
NOT FINAL
REF IS SYSTEM GENERATED
INSTANCE METHOD AGE() RETURNS INTEGER;
CREATE INSTANCE METHOD AGE() RETURNS INTEGER
FOR PERSON_TYPE
BEGIN
    RETURN /* CODE TO CALCULATE A PERSON'S AGE FROM TODAY'S DATE AND SELF.BIRTH_DATE */
END;

• Is there encapsulation in the code on the left?
KEY MESSAGE

- ORDBMS has objects similar to that in Java, but there are also differences, such as persistence of objects and encapsulation dilemma.
- There is tradeoff for encapsulation. One should choose encapsulation according to the situation.