Problem
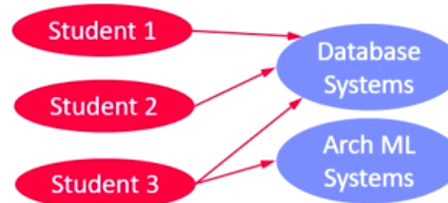
- application use object-oriented approach
- data stored in normalized flat tables
- application is responsible for bridging
- example

  - **SELECT** * **FROM** Students
  - **SELECT** C.Name, C.ECTS **FROM**
    Courses C, Attendance A
    WHERE C.CID = A.CID
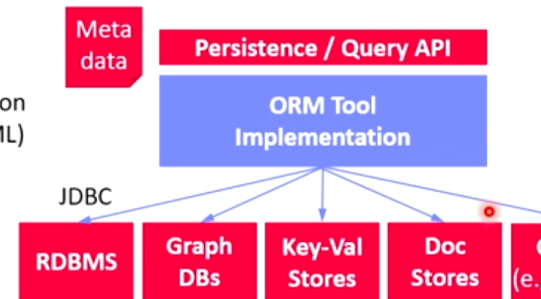   **AND** A.SID = 7;
  - … A.SID = 8;

Object-Relational Mapping

- ORM tools allow automatic
  - handling of object percistence lifecycle
  - querying of underlying data stores
- reduced development effort

Common High-Level Architecture
- **#1** Persistence definition (meta data → e.g., XML)
- **#2** Persistence API
- **#3** Query language / query API

- improved testing and independence of DBMS

Pros and Cons

- advantages
  - simple CRUD operations
  - simple queries
  - application centric development
- disadvantages
  - unnecessary indirections and complexity
    * mapping
    * meta data
  - performance harder to ensure
  - no application centric development
    * schema ownership
    * already existing data
  - dependent on framework APIs

1

[[Call Level Interfaces]]