DATABASE DESIGN I - 1DL300

Spring 2013

An Introductory Course on Database Systems

http://www.it.uu.se/edu/course/homepage/dbastekn/vt13/

Silvia Stefanova
Uppsala Database Laboratory
Department of Information Technology, Uppsala University,
Uppsala, Sweden
Database APIs

Elmasri/Navathe ch 12
Padron-McCarthy/Risch ch 20

Silvia Stefanova

Department of Information Technology
Uppsala University, Uppsala, Sweden
Outline

1. How to query by SQL
2. Database Applications (SQL queries inside programs)
3. Database APIs
4. Examples of using different database APIs
5. Prepared SQL query
1. Textual interface
   The user types SQL into a monitor

2. Graphical query interface
   The user constructs SQL by Query by Example (QBE)

3. SQL application programming interfaces
   • SQL is included in an application program (in C, Java, PHP, etc. …)
   • Host language
   • Data sublanguage
SQL queries inside a program

**Native APIs (Application Programming Interface) in C, PHP, Python, etc.**
- Libraries of database functions and procedures
- Dependent of the DBMS

**ODBC (Open Database Connectivity)**
- API in C
- Independent of the DBMS and the operating systems

**JDBC (Java Database Connectivity)**
- API in Java
- Independent of DBMS and operating systems

**ESQL (Embedded SQL)**
- Embedded SQL in host language
Requirements for the APIs?

1. Connect the application program with the DBMS, i.e. introduce *sessions*

2. Send SQL statements to the DBMS for execution

3. Fetch and manage the result of the SQL statements

4. Avoid unnecessary optimization of the SQL queries which can contribute to time delays
Database API – in General

• Connection: initialize connection with the database

```java
connection=db_connect
("jdbc:microsoft:sqlserver://localhost;DatabaseName=person",
 "com.microsoft.jdbc.sqlserver.SQLServerDriver",
 "person",
 "person",
 "silvia",
 "123456” );
```

• Query:

```java
result = db_query(connection, “SELECT name,tel from personalinfo”);
```
Database API – in General

• Fetch the result: fetches one row at a time

```c
while( row = db_fetch(result)) != NULL
{
    print(“Name, Telephone “,
           row->fields[0], row->fields[1] );
}
```

• Close connection

```c
db_close(connection)
```
The ODBC architecture

ODBC API is independent of any one programming language, database system or operating system; it is a standard.
Database API – Examples

Example 1: **Native API** – connect to a DB stored in MySQL by PHP

Example 2: **ODBC** – connect to a DB stored in MySQL by PHP through ODBC
Advantages of ODBC

• Standard API

• API is independent of DBMS
  ➢ By the same program code a user can connect to different types of DBMS without changing it

• Easy to connect to different data sources and DBMS
  ➢ It is possible to communicate with several data sources of different type
JDBC API

• The JDBC API: set of Java interfaces that allow database applications to:
  ➢ open connections to a database
  ➢ execute SQL statements
  ➢ process the results

• Main methods:
  ➢ java.sql.DriverManager: loads the specific drivers and supports creating new database connections
  ➢ java.sql.Connection: represents a connection to a specific database
  ➢ java.sql.Statement: allows the application to execute a SQL statement
  ➢ java.sql.PreparedStatement: represents a pre-compiled SQL statement
  ➢ java.sql.ResultSet: controls access to rows resulting from executing a statement
The JDBC architecture

JDBC API is independent of (relational) DBMS and operating system
Database API – Examples

Example 3: **JDBC** – connect to a DB stored in MySQL by a Java application
Prepared SQL query

• Prepare an SQL query
  ➢ The process to compile and optimize SQL query
  ➢ Takes longer time (much longer sometimes) compared to the time for executing the SQL query

• Execute the same SQL query several times
  ➢ Prepare the query (*parameterized query*) when it is executed for the first time.
  ➢ Use prepared query next time to skip compilation and optimization
Example 4: **JDBC** – connect to a DB stored in MySQL by a Java application; use prepared SQL
Summary

- Database APIs
- ODBC
- JDBC
- Prepared query