

8. Application Development

- Embedded
- ODBC, JDBC
- Application Architecture

database application

- static approaches: SQL in application (임베딩 하는 형식)
- dynamic approaches: API호출 후 function call

8.1 Embedded SQL

c, c++, java, cobol.. (host language) 안에 SQL 임베딩 -> preprocessor가 임베디드 sql을 api호출로 바꿈

SQL 표현은 언어마다 다양함

Cursor

2개 이상의 언어가 manipulate 하는 방식이 달라서

sql 수행 결과 - multi set host language - multi set 자료구조 없음

multi set 있어도 sql표현에 제약 -> 왜? sql은 너무 많은 칼럼 출력시 몇개 그냥 출력 안하니까.. --> cursor로 데이터 하나씩 받아오기 (tuple by tuple)

```
EXEC SQL declare myCursor cursor for
select SID, name from student where totalCredit > :creditAmount;
EXEC SQL open myCursor;
EXEC SQL fetch myCursor into :si,:sn;
//fetch 할 때 마다 tuple 하나씩 넘어옴 -> error code 02000이면 더이상 넘어올 튜플이 없는 것..
//si, sn은 declare된 지역 변수이며 :은 규약
close

fetch!!!
declare -> open -> fetch -> close
```

Dynamic SQL

Dynamic SQL Example

- `stcopy("select name from professor where salary > ?", myText); /* can have placeholders */`
`EXEC SQL prepare my1 from :myText;`
`mySalary = 1000;`
`EXEC SQL execute my1 using :mySalary;`
- `stcopy("select name from professor where pID='234', myText2);`
`EXEC SQL execute immediate from myText2;`
`/* cannot have any placeholders */`

런타임때 sql문장이 결정되면 prepare하고 execute함.. -> 한번에 수행하면 execute immediate

8.2 ODBC, JDBC

프로그램과 db 서버를 연결하는 API -> 명령 처리 등등 다 가능

ODBC 2.0

allocate SQL environment, database connection handle and open database connection using SQLConnect()

버전마다 handle call에 차이가...

ODBC Data Access

SQLExecDirect() -> 데이터 접근 SQLFetch() -> 결과 받을 때 SQLBindCol() -> 프로그램(c언어...) 변수와 바인딩 할때

ODBC Dynamic SQL

기본적으로 동적 sql지원 prepare필요 SQLPrepare() -> prepare하기 등등 다양한 함수존재....

ODBC Metadata

SQLTable(), SQLColumns().... sql object가 모여 하나의 스키마, 스키마가 모여 카탈로그

ODBC Transaction

당연히 지원되어야함....이러는데

등등...

JDBC - 자바에서 쓰는 ODBC

host language가 자바인 ODBC..

SQLJ -> 자바에서 sql 임베딩하는 것

요새는 잘 안씀

ADO.NET

이것도그냥 ODBC콜하는거임

Static vs Dynamic Approaches

임베디드, SQLJ - Static Database API call - Dynamic

8.3 Application Architecture

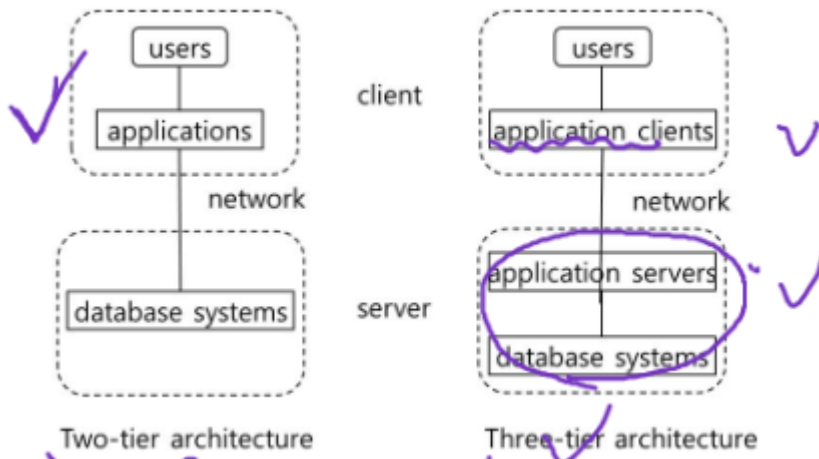
Web.. 관련 얘기들 정보시스템의 흐름

two-tier

Application이 Client 쪽에 존재

Three-tier

Application이 client - server로 나뉘서 server에는 db시스템도 함께..



Three-Layer Web Architecture

웹서버 - app서버 - db서버

Two-Layer Web Server

웹&App서버 - db서버

Cookies

Stateless Connection라는 Web(HTTP)의 특징을 보완하기 위해 사용 cookie로 이전에 사용자가 한 행동 저장

마지막 3페이지 skip