

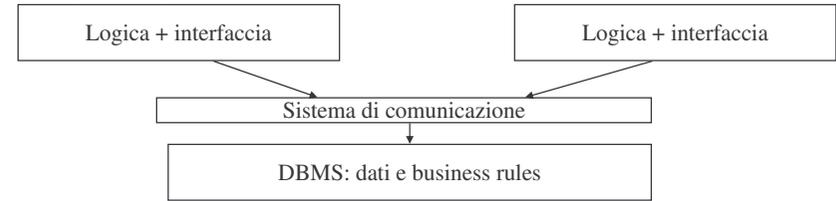
LA REALIZZAZIONE DI APPLICAZIONI

- Quattro parti:
 - Gestione dati
 - Business rules
 - Logica applicativa
 - Interfaccia utente
- Molte possibili architetture
- L'approccio tradizionale: uso di un linguaggio

PLSQL 1.1

ALCUNE ARCHITETTURE

Client-Server

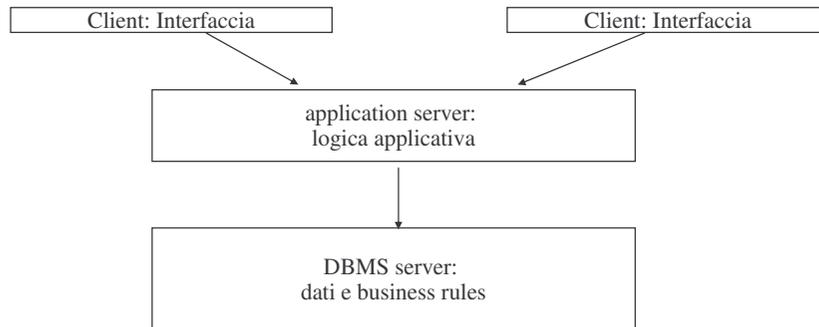


Thin Client



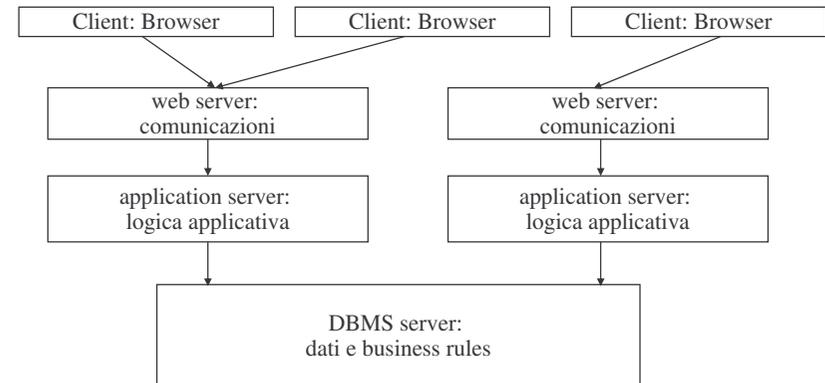
PLSQL 1.2

THREE TIERS



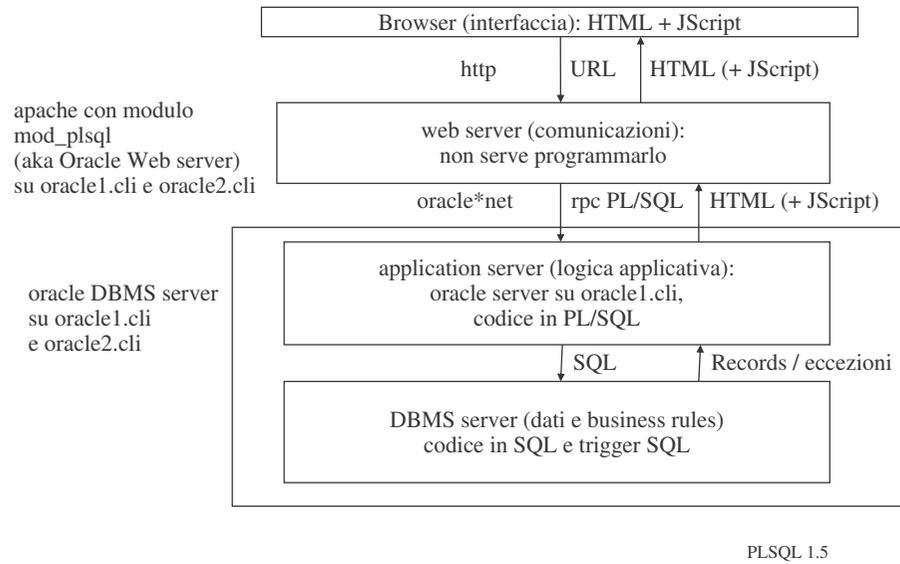
PLSQL 1.3

WEB SERVER



PLSQL 1.4

LA NOSTRA ARCHITETTURA



IL LINGUAGGIO

- Scrivere un'applicazione che visualizzi una schermata, raccolga dei dati, segnali eventuali inconsistenze con i dati nella BD oppure inserisca i nuovi dati
- Serve un linguaggio che possa:
 - Effettuare I/O
 - Effettuare interrogazioni
 - Controllare il flusso in un modo che dipende dal risultato dell'interrogazione
 - Effettuare aggiornamenti

PLSQL 1.6

IL LINGUAGGIO

- Tre soluzioni:
 - Linguaggio di programmazione + API
 - Linguaggio immerso
 - Linguaggio integrato

PLSQL 1.7

LINGUAGGIO INTEGRATO

- Integrazione dei tipi di dato
- Integrazione dello scoping
- Integrazione del DML

PLSQL 1.8

UNA SOLUZIONE

```
procedure prenotaIf(  
  ilLogin      in prenota.login%TYPE,  
  laData       in date,  
  IOra         in prenota.ora%TYPE) is  
  unaPrenotazione prenota%ROWTYPE;  
  cursor c is select *  
    from prenota  
    where data = laData and ora = IOra;  
begin  
  open c;  
  fetch c into unaPrenotazione;  
  if c%NOTFOUND  
  then insert into prenota  
    values (codiceSeq.nextval, ilLogin, laData,  
           IOra, ilTerm, laDataP, ilTermP);  
  else ...;  
  end if;  
end prenotaIf;
```

PLSQL 1.9

L'ASPETTO DEL CODICE

```
procedure creaElaborato(  
  CodTemp      AllocationsTemp.CodTemp%type,  
  Resp         number,  
) is  
  i binary_integer;  
  cursor callocazioni is select CodEl, Matricola,  
  from Allocations  
  where Allocations.CodEl = CreaElaborato.CodTemp;  
begin  
  insert into Elaborati (CodEl, PassEl, CodCo)  
  values (CodEl, PassEl, '1');  
  open callocazioni;  
  loop  
    fetch callocazioni into ICodTemp, LaMatricola;  
    exit when callocazioni%notfound;  
    insert into Allocations (Matricola, CodEl, Responsabile)  
    values (LaMatricola, CodEl, 'N');  
  end loop;  
  close callocazioni;  
end creaElaborato;
```

PLSQL 1.10

PL/SQL

- Un linguaggio per manipolare basi di dati che integra DML (SQL) con il linguaggio ospite
- Un linguaggio a blocchi con una struttura del controllo completa che contiene l'SQL come sottolinguaggio
- Permette:
 - Di definire variabili di tipo scalare, record (annidato), insieme di scalari, insieme di record piatti, cursore
 - Di definire i tipi delle variabili a partire da quelli della base di dati
 - Di eseguire interrogazioni SQL ed esplorarne il risultato
 - Di modificare la base di dati
 - Di definire procedure e moduli
 - Di gestire il flusso del controllo, le transazioni, le eccezioni

PLSQL 1.11

STRUTTURA

- Il blocco:
 - DECLARE <dichiarazioni>
 - BEGIN <comandi>
 - EXCEPTION <gestori>
 - END;
- La procedura:
 - PROCEDURE <parametri> IS
 - <dichiarazioni>
 - BEGIN <comandi>
 - EXCEPTION <gestori>
 - END;

PLSQL 1.12

Il modulo: interfaccia e implementazione

- CREATE PACKAGE <nome> AS ... END <nome>;
- CREATE PACKAGE BODY <nome> AS ...
END <nome>;

PLSQL 1.13

LE DICHIARAZIONI DI VARIABILI

- <nome> <tipo>;
- <nome> CONSTANT <tipo> := <expr>;
- CURSOR <nome> IS <query>;
 - Operazioni: OPEN, FETCH, CLOSE
- Attributi di variabili e cursori:
 - Var%TYPE, dove var è una variabile o una colonna
 - Cur%ROWTYPE dove cur è un cursore o una tabella

PLSQL 1.14

IL FLUSSO DEL CONTROLLO

- IF - THEN - (ELSIF THEN) - ELSE - END IF
- LOOP <comandi> END LOOP (EXIT WHEN <cond>)
- FOR var IN seq LOOP <comandi> END LOOP
- GOTO label ... <<label>> comandi

PLSQL 1.15

TABELLE E RECORD

- Una tabella è un'array associativo dinamico, di tipo
 - TABLE OF <tipo> INDEX BY <tipo>
- I tipi record sono annidabili:
 - RECORD (nome tipo, ..., nome tipo)

PLSQL 1.16

ECCEZIONI

- DECLARE aaa EXCEPTION;
- RAISE aaa
- EXCEPTION WHEN aaa THEN ... END

PLSQL 1.17

ARCHITETTURA

- Tre possibilità:
 - Un programma PL\SQL può essere eseguito da uno strumento, che invia i comandi SQL al server;
 - Un programma PL\SQL può essere inviato da uno strumento al server, il quale lo esegue;
 - Un programma PL\SQL può risiedere nel server, che lo esegue.

PLSQL 1.18