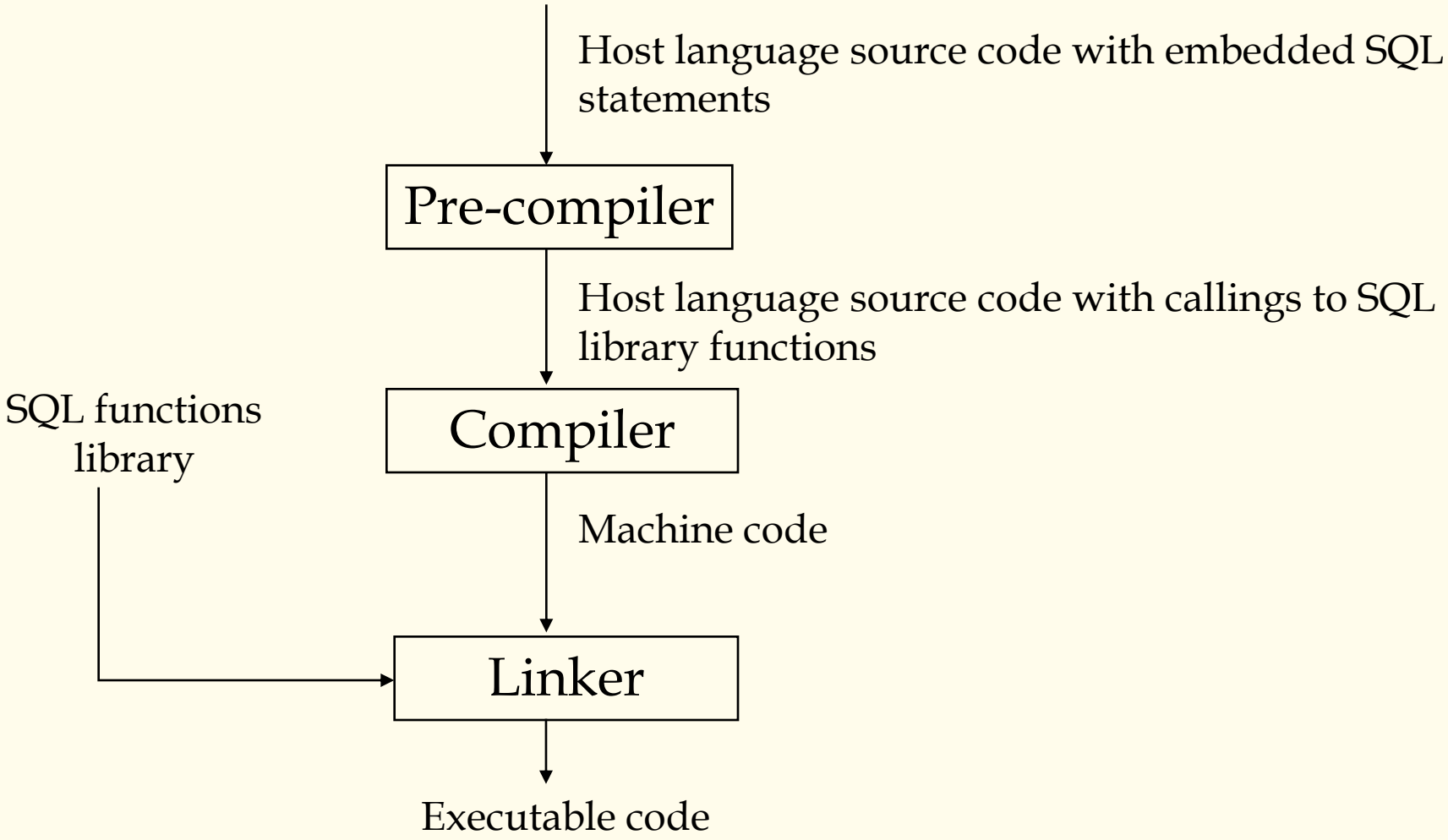
Cursor

1. Define a cursor
   - EXEC SQL DECLARE <*cursor name*> CURSOR FOR
     SELECT …
     FROM …
     WHERE …
2. EXEC SQL OPEN <*cursor name*>
   - Some like open a file
3. Fetch data from cursor
   - EXEC SQL FETCH <*cursor name*>
     INTO :hostvar1, :hostvar2, …;
4. SQLCA.SQLCODE will return 100 when arriving the end of cursor
5. CLOSE CURSOR <*cursor name*>

Example of Query with Cursor

```
        :
EXEC SQL DECLARE C1 CURSOR FOR
        SELECT SNO, GRADE
        FROM SC
        WHERE CNO = :GIVENCNO;
EXEC SQL OPEN C1;
if (SQLCA.SQLCODE<0) exit(1);          /* There is error in query*/
while (1) {
    EXEC SQL FETCH C1 INTO :SNO, :GRADE :GRADEI
    if (SQLCA.SQLCODE==100) break;
    /* treat data fetched from cursor, omitted*/
        :
}
EXEC SQL CLOSE C1;
        :
```

Conceptual Evaluation



Dynamic SQL

- In above embedded SQL, the SQL statements must be written before compiling. But in some applications, the SQL statement can't decided in ahead, they need to be built dynamically while the program running.
- Dynamic SQL is supported in SQL standard and most RDBMS pruducts
  - Dynamic SQL executed directly
  - Dynamic SQL with dynamic parameters
  - Dynamic SQL for query

Dynamic SQL executed directly

- Only used in the execution of non query SQL statements

```
        :
EXEC SQL BEGIN DECLARE SECTION;
char sqlstring[200];
EXEC SQL END DECLARE SECTION;
char cond[150];
strcpy( sqlstring, "DELETE FROM STUDENT WHERE ");
printf(" Enter search condition :");
scanf("%s", cond);
strcat( sqlstring, cond);
EXEC SQL EXECUTE IMMEDIATE :sqlstring;
        :
```

Dynamic SQL with dynamic parameters

- Only used in the execution of non query SQL statements. Use ***place holder*** to realize dynamic parameter in SQL statement. Some like the macro processing method in C.

```
        :
EXEC SQL BEGIN DECLARE SECTION;
char sqlstring[200];
int birth_year;
EXEC SQL END DECLARE SECTION;
strcpy( sqlstring, "DELETE FROM STUDENT WHERE
                        YEAR(BDATE) <= :y; ");
printf(" Enter birth year for delete :");
scanf("%d", &birth_year);
EXEC SQL PREPARE purge FROM :sqlstring;
EXEC SQL EXECUTE purge USING :birth_year;
        :
```