CS245: Databases

SQL

Vijaya saradhi

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati

Cursor - I

Impedance Model Mis-match

- SQL always returns relations
- Other programming languages has data types that are not relations
- These languages cannot hold relations returned by SQL
- C language has pointers; where as SQL do not have any such construct
- As a result, passing data between SQL and other languages is not straightforward
- Mechanisms must be devised to allow the development of programs that use both SQL and other languages

Cursor - I

Impedance Model Mis-match

- Versatile way to connect SQL queries to a host language is with a cursor
- Cursor runs through the tuples of a relation
- This relation can be stored table, or it can be something that is generated by a query

Cursor - I

Details

- SELECT will return a relation
- Returned relation will not be stored
- Often the need to process one row at a time of returned relation arise
- Cursor helps examining one row at a time

Cursor - II

Details

- Assume the returned relation to be a file in itself
- Operations required for reading a file are
  - Declare file pointer
  - Open the file
  - Read one line at a time repeatedly
  - close the file
- Similar tasks are associated with cursor

Cursor - III

Declare cursor

```
DECLARE cursor_name CURSOR FOR SELECT statement;

OPEN  cursor_name;

FETCH cursor_name INTO variable_list;

CLOSE cursor_name
```

Cursor - Example

Example

```
DELIMITER //
CREATE PROCEDURE f11()
BEGIN
-- Declare variables
        DECLARE i INT DEFAULT 1;
        DECLARE sno INT;
        DECLARE sname char(50);
        DECLARE rating INT DEFAULT 10;
        DECLARE age INT DEFAULT 16;

-- Declare cursors
        DECLARE my_first_cursor CURSOR FOR
        SELECT  *
        FROM    Sailors
        WHERE   age > 20 AND rating BETWEEN 5 AND 7;

-- Declare cursor handler
        DECLARE CONTINUE HANDLER FOR NOT FOUND SET NO_records = 1;
```

Cursor - Example

Example

```
  OPEN my_first_cursor;

-- loop through all the rows
loop_1: REPEAT
-- Get one roll number from list of registered students into variable rn
    FETCH my_first_cursor INTO (sno, sname, rating, age);
-- Check number of records in the cursor
    IF NO_records = 1 THEN
      LEAVE loop_1;
    END IF;

    UNTIL NO_records
  END REPEAT loop_1;
  CLOSE my_first_cursor;
END; //
DELIMITER ;
```

Cursor - IV

Scrolling

- Cursor gives us flexibility as how to move through the tuples of the relation
- The default choice is to start at the beginning of the relation and fetch the tuples in order
- Fetch all tuples until end of the relation
- Other orders in which tuples may be fetched
- These options are not available in `MySQL` yet we will discuss these

Cursor - V

Scrolling

- Instruct the cursor to open in `SCROLL` model before the keyword `CURSOR`
- `EXEC SQL DECLARE name SCROLL CURSOR FOR MovieExec;`
- This will tell SQL that cursor may be used in a manner other than moving in forward direction alone
- The FETCH is responsible for specifying the direction from which the next tuple be obtained
  - `FETCH NEXT` retrieve next tuple
  - `FETCH PRIOR` retrieve previous tuple
  - `FETCH FIRST` retrieve first tuple
  - `FETCH LAST` retrieve last tuple
  - `FETCH ABSOLUTE i` specifies the position of the tuple to be fetched from the top of the relation