

Перезапуск Data Engineer

Решаем тест по предыдущей лекции
до 18.40

Курсоры

При обработке **SQL** - операторов **Oracle** выделяет область памяти называемую "контекстной областью" (context area). Она содержит информацию необходимую для начала и завершения обработки **SQL** оператора. В том, числе - число строк обрабатываемых оператором, указатель на представление этого оператора после синтаксического анализа (еще это называют парсинг) и активный набор (active set) - например набор строк возвращаемых запросом.

Итак: **КУРСОР** - это указатель (хотя как такового, понятия "указатель" в **PL/SQL** нет!) на контекстную область памяти, с помощью которого программа на языке **PL/SQL** может управлять контекстной областью и ее состоянием во время обработки оператора.

Объявление курсора определяет какое выражение языка **SQL** - будет передано программе **SQL Statement Executor** (системе исполнителю выражения **SQL**). Курсор может представлять собой любое допустимое предложение языка **SQL**! Так же, курсор является основным базовым "кирпичиком" для построения блоков **PL/SQL**. Курсоры обеспечивают циклический механизм оперирования наборами данных в БД. Курсор может возвращать одну или несколько строк данных или вообще ни одной.

Типичная последовательность, при операциях в данном случае с явными (определенными курсорами) будет такая:

- Объявление курсора и структуры данных, в которую, будут помещены найденные строки.
- Открытие курсора.
- Последовательная выборка данных.
- Закрытие курсора.

Курсоры

CURSOR -- имя (передаваемые параметры) --- IS
SELECT список полей FROM таблица выбора
WHERE условия выбора в курсор

Курсорные атрибуты

В **PL/SQL** имеется четыре основных курсорных атрибута

- **%FOUND (NULL, TRUE, FALSE)**
- **%NOTFOUND (NULL, TRUE, FALSE)**
- **%ISOPEN (TRUE, FALSE)**
- **%ROWCOUNT** (является числовым атрибутом и возвращает число строк считанных курсором на определенный момент времени)

%FOUND и %NOTFOUND принимают противоположные значения друг другу

Атрибуты курсора объявляются подобно операторам **%TYPE** и **%ROWTYPE**, справа от имени курсора, вот так:

имя курсора%атрибут

Курсоры. FOR UPDATE или WHERE CURRENT OF

Эти конструкции нужны чтобы обновить строки внутри курсора

```
DECLARE
  CURSOR upd_emp(dep_id employees.department_id%type) IS
    SELECT *
    FROM employees
    WHERE department_id = dep_id
    FOR UPDATE
    ;
BEGIN

  FOR v_gt IN upd_emp(230) LOOP

    UPDATE employees
    SET employees.salary = employees.salary + 1000
    WHERE CURRENT OF upd_emp;

  END LOOP;
  COMMIT;

END;
```

Подсказки оптимизатору

`/*+ HINT HINT ... */` в PL/SQL пробел между «+» и первой буквой подсказки имеет значение, в случае отсутствия пробела подсказка может быть игнорирована, т.е. `/*+ ALL_ROWS */` правильное использование, а `/*+ALL_ROWS */` — неправильное

`SELECT /*+ FULL (employees) */ empno FROM employees myalias WHERE empno > 10; -- НЕправильно`

`SELECT /*+ FULL (myalias) */ empno FROM employees myalias WHERE empno > 10; -- правильно`

Подсказки оптимизатору

— общие цели оптимизатора

[/*+ RULE */](#)

[/*+ ALL ROWS */](#)

[/*+ FIRST ROWS */](#)

[/*+ FIRST ROWS\(n\) */](#)

— порядок доступа

[/*+ LEADING */](#)

[/*+ ORDERED */](#)

— методы соединения

[/*+ USE HASH */](#)

[/*+ USE NL */](#)

[/*+ USE MERGE */](#)

[/*+ USE HASH AGGREGATION */](#)

[/*+ NATIVE FULL OUTER JOIN */](#)

[/*+ INDEX JOIN */](#)

[/*+ INDEX COMBINE */](#)

[/*+ NUM INDEX KEYS */](#)

— способы выполнения [под]запроса

[/*+ DRIVING SITE */](#)

[/*+ MATERIALIZE */](#)

[/*+ INLINE */](#)

[/*+ PRECOMPUTE SUBQUERY */](#)

— статистика объектов

[/*+ DYNAMIC SAMPLING */](#)

[/*+ DYNAMIC SAMPLING EST CDN */](#)

[/*+ CARDINALITY */](#)

[/*+ OPT_ESTIMATE */](#)

[/*+ TABLE_STATS | INDEX_STATS | COLUMN_STATS */](#)

Подсказки оптимизатору

— трансформации [под]запросов

/*+ NO_QUERY_TRANSFORMATION */

/*+ PUSH_SUBQ */

/*+ NO_UNNEST */

/*+ UNNEST */

/*+ NO_ELIMINATE_OBY */

/*+ MERGE */ /*+ NO_MERGE */

/*+ USE_CONCAT */

/*+ NO_EXPAND */

/*+ PUSH_PRED */ /*+ NO_PUSH_PRED */

/*+ FACTORIZE_JOIN */ /*+ NO_FACTORIZE_JOIN */

/*+ OR_EXPAND */

— использование курсоров

/*+ CURSOR_SHARING_EXACT */

/*+ BIND_AWARE */ /*+ NO_BIND_AWARE */

— параллельное выполнение

/*+ PARALLEL */

/*+ SHARED */

/*+ STATEMENT_QUEUING */ /*+ NO_STATEMENT_QUEUING */

— прочие

/*+ QB_NAME */

/*+ GATHER_PLAN_STATISTICS */

/*+ OPT_PARAM */

/*+ OPTIMIZER_FEATURES_ENABLE */

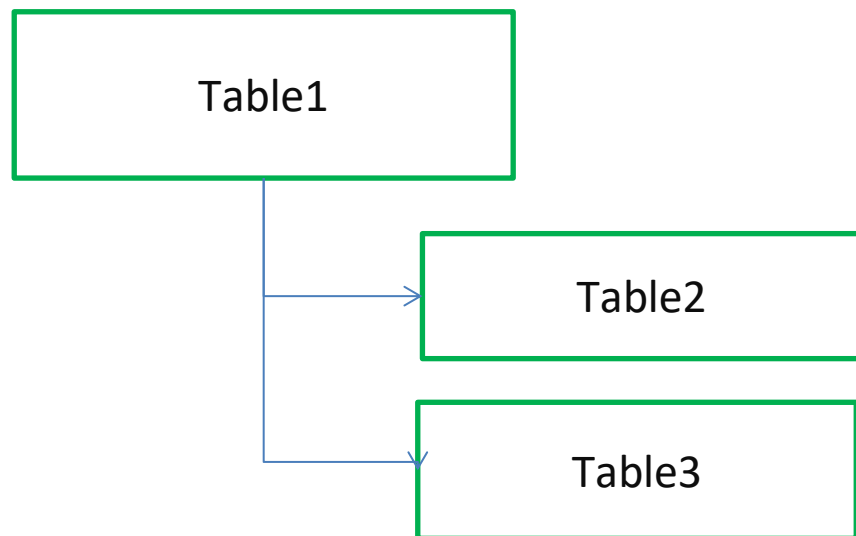
/*+ APPEND */ /*+ NOAPPEND */

/*+ APPEND_VALUES */

/*+ RESULT_CACHE */ /*+ NO_RESULT_CACHE */

/*+ IGNORE_ROW_ON_DUPKEY_INDEX */

Пример как делать не нужно



На одну строку из table1 приходится несколько строк из table2

На одну строку из table1 приходится несколько строк из table3

```
Select *  
From Table1  
Join Table2 on.....  
Join Table3 on.....
```

EXPLAIN PLAN

EXPLAIN PLAN FOR

```
select /*+ SOME_HINT*/ *  
from table.....;
```

SELECT * FROM table(DBMS_XPLAN.DISPLAY);

Виды физических соединений

- Соединение хешированием (hash join)
- Соединение вложенными циклами (nested loops join)
- Соединение слиянием сортированных списков (merge join, sort merge join, sort-merge join)