LanguageManual SubQueries

Subqueries in the FROM Clause

```
SELECT ... FROM (subquery) name ...
SELECT ... FROM (subquery) AS name ...   (Note: Only valid starting with Hive 0.13.0)
```

Hive supports subqueries only in the FROM clause (through Hive 0.12). The subquery has to be given a name because every table in a FROM clause must have a name. Columns in the subquery select list must have unique names. The columns in the subquery select list are available in the outer query just like columns of a table. The subquery can also be a query expression with UNION. Hive supports arbitrary levels of subqueries.

The optional keyword "AS" can be included before the subquery name in Hive 0.13.0 and later versions (HIVE-6519).

Example with simple subquery:

```
SELECT col
FROM (
  SELECT a+b AS col
  FROM t1
) t2
```

Example with subquery containing a UNION ALL:

```
SELECT t3.col
FROM (
  SELECT a+b AS col
  FROM t1
  UNION ALL
  SELECT c+d AS col
  FROM t2
) t3
```

Subqueries in the WHERE Clause

As of Hive 0.13 some types of subqueries are supported in the WHERE clause. Those are queries where the result of the query can be treated as a constant for IN and NOT IN statements (called *uncorrelated subqueries* because the subquery does not reference columns from the parent query):

```
SELECT *
FROM A
WHERE A.a IN (SELECT foo FROM B);
```

The other supported types are EXISTS and NOT EXISTS subqueries:

```
SELECT A
FROM T1
WHERE EXISTS (SELECT B FROM T2 WHERE T1.X = T2.Y)
```

There are a few limitations:

- These subqueries are only supported on the right-hand side of an expression.
- IN/NOT IN subqueries may only select a single column.
- EXISTS/NOT EXISTS must have one or more correlated predicates.
- References to the parent query are only supported in the WHERE clause of the subquery.