## MYSQL JOINS

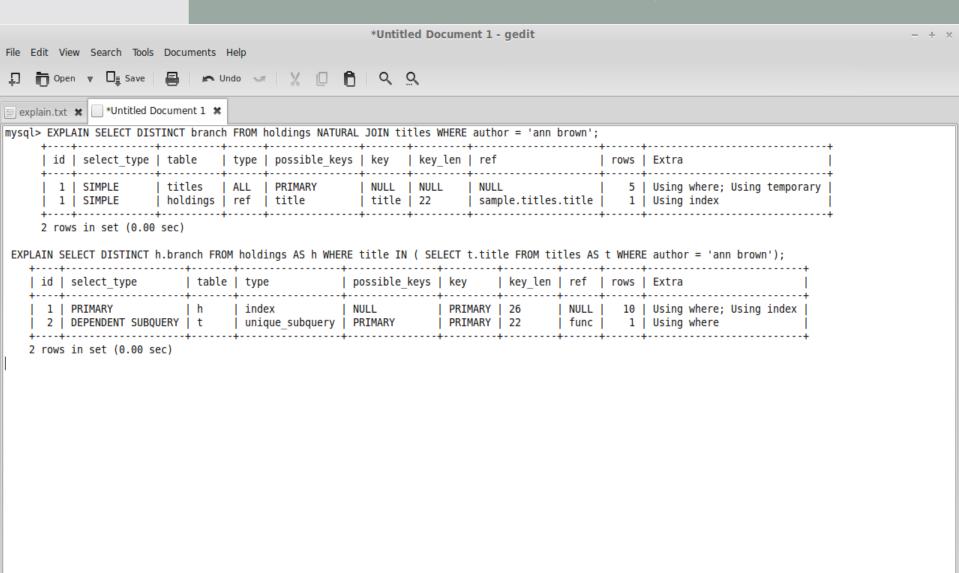
## by:ANKIT BANSAL

AUG 7, 2013.

- JOINS ARE USED TO COMBINE ROWS OF TWO OR MORE DIFFERENT TABLES IN A TEMPORARY TABLES.
- IT IS MORE OPTIMISED THAN THE SUBQUERIES METHOD.
- BASIC SYNTAX OF JOIN.

SELECT<column name>
FROM table1
join\_type table2
[ON(join\_condition)]

## PERFORMANE COMPARISION OF JOIN AND SUQUERIES USING EXPLAIN





























## **TYPES OF JOINS**

- INNER JOIN
- NATURAL JOIN
- OUTER JOIN
  - LEFT JOIN
  - o RIGHT JOIN
- SELF JOIN
- MULTIPLE TABLE JOIN

SYNTAX

```
SELECT<column name> FROM table1 JOIN/INNERJOIN table2
[ON(join_condition)] /
[WHERE(CONDITION)] /
[USING(column_name)];
```

- WHEN using USING TYPE THE COLUMN NAME SHOULD BE SAME IN BOTH THE TABLES.
- IF CONDITION IS NOT DEFINED IT WORKS LIKE A CARTESIAN PRODUCT OF THE TWO TABLE.

SYNTAX

SELECT<column name> FROM table1 NATURAL JOIN table2;

• ONE COLUMN NAME SHOULD BE SAME IN BOTH THE TABLES UPON WHICH JOIN WILL BE PERFORMED.

- LEFT JOIN IS USED TO PRESERVE ALL THE ROWS MATCHING
   OR NON-MATCHING OF THE TABLE WRITTEN LEFT OF THE JOIN.
- SYNTAX

```
SELECT<column name> FROM table1 LEFT JOIN table2 [ON(join_condition)] / [USING(column_name)];
```

• WHEN using USING TYPE THE COLUMN NAME SHOULD BE SAME IN BOTH THE TABLES.

- RIGHT JOIN IS USED TO PRESERVE ALL THE ROWS MATCHING OR NON-MATCHING OF THE TABLE WRITTEN RIGHT OF THE JOIN.
- SYNTAX

```
SELECT<column name> FROM table1 RIGHT JOIN table2 [ON(join_condition)] / [USING(column_name)];
```

 WHEN using USING TYPE THE COLUMN NAME SHOULD BE SAME IN BOTH THE TABLES.

- SELF JOIN IS JOINING TABLE TO ITSELF MAKING COPY OF IT TEMPORARY.
- SYNTAX

```
SELECT<column name> FROM table1 AS a JOIN table1 AS b [ON(join_condition)] / [WHERE(CONDITION)] / [USING(column_name)];
```

 WHEN using USING TYPE THE COLUMN NAME SHOULD BE SAME IN BOTH THE TABLES. WE CAN ALSO JOIN MORE THAN TWO TABLES.

LIKE: SELECT \* FROM tab1,tab2,tab3;

- THIS STATEMENT WILL PRODUCE THE CARTESIAN PRODUCT OF ALL THE 3 TABLES.
- EXAMPLE:

SELECT pp.emp\_id FROM proj\_tech AS pt JOIN technologies AS t

ON pt.tech\_id = t.id AND t.name = 'android' JOIN past\_proj pp

ON pp.p\_id = pt.p\_id GROUP BY pp.emp\_id;