MySQL Arithmetic Operators

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide
%	Modulo

MySQL Bitwise Operators

Operator	Description
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR

1. Bitwise AND (&)

Query:

SELECT user_id, permissions, (permissions & 1) AS has_read_permission

```
FROM user_permissions
WHERE (permissions & 1) = 1;
```

Explanation:

- The bitwise AND operation compares each bit of `permissions` with the bit of `1` (which is `0001` in binary).
- If both corresponding bits are `1`, the result bit is set to `1`. Otherwise, the result bit is `0`.

For example:

- For `permissions = 1` (binary `0001`):
  - `0001 & 0001 = 0001` (result is `1`, indicating the Read permission is present)
- For `permissions = 3` (binary `0011`):
  - `0011 & 0001 = 0001` (result is `1`, indicating the Read permission is present)
- For `permissions = 4` (binary `0100`):
  - `0100 & 0001 = 0000` (result is `0`, indicating the Read permission is not present)
- For `permissions = 7` (binary `0111`):
  - `0111 & 0001 = 0001` (result is `1`, indicating the Read permission is present)

2. Bitwise OR (|)

Query:

```
UPDATE user_permissions
SET permissions = permissions | 8
WHERE (permissions & 2) = 2;
```

Explanation:

- The bitwise OR operation compares each bit of `permissions` with the bit of `8` (which is `1000` in binary).
- If either bit is `1`, the result bit is set to `1`. Otherwise, the result bit is `0`.

For example:

- For `permissions = 2` (binary `0010`):
  - `0010 | 1000 = 1010` (result is `10` in decimal)
- For `permissions = 3` (binary `0011`):
  - `0011 | 1000 = 1011` (result is `11` in decimal)

- For `permissions = 5` (binary `0101`):
    - `0101 | 1000 = 1101` (result is `13` in decimal)

3. Bitwise XOR (^)

Query:

```
UPDATE user_permissions
SET permissions = permissions ^ 4
WHERE user_id = 5;
```

Explanation:

- The bitwise XOR operation compares each bit of `permissions` with the bit of `4` (which is `0100` in binary).
- If the corresponding bits are different, the result bit is set to `1`. Otherwise, the result bit is `0`.

For example:

- For `permissions = 3` (binary `0011`) initially:
    - `0011 ^ 0100 = 0111` (result is `7` in decimal, toggling the Execute permission)
- If `permissions = 7` (binary `0111`):
    - `0111 ^ 0100 = 0011` (result is `3` in decimal, toggling the Execute permission back)

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

MySQL Comparison Operators

Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to

MySQL Compound Operators

Operator	Description
+=	Add equals
-=	Subtract equals
*=	Multiply equals
/=	Divide equals
%=	Modulo equals
&=	Bitwise AND equals
^-=	Bitwise exclusive equals
|*=	Bitwise OR equals

MySQL Logical Operators

Operator	Description
ALL	TRUE if all of the subquery values meet the condition
AND	TRUE if all the conditions separated by AND is TRUE
ANY	TRUE if any of the subquery values meet the condition
BETWEEN	TRUE if the operand is within the range of comparisons
EXISTS	TRUE if the subquery returns one or more records
IN	TRUE if the operand is equal to one of a list of expressions
LIKE	TRUE if the operand matches a pattern
NOT	Displays a record if the condition(s) is NOT TRUE
OR	TRUE if any of the conditions separated by OR is TRUE
SOME	TRUE if any of the subquery values meet the condition