

# Combining Queries

```
query1 UNION [ALL] query2
query1 INTERSECT [ALL] query2
query1 EXCEPT [ALL] query2
```

Here we use the following tables as example

```
create table T1
(
    num    int primary key,
    name   varchar(10)
);
insert into T1(num, name)
values (1, 'qqq'),
       (2, 'yyy'),
       (4, 'zzz'),
       (7, 'qqq'),
       (8, 'zzz');

create table T2
(
    num    int primary key,
    name   varchar(10)
);
insert into T2(num, name)
values (2, 'yyy'),
       (4, 'qqq'),
       (6, 'qqq');
```

## UNION and UNION ALL

`UNION` effectively appends the result of `query2` to the result of `query1` (although there is no guarantee that this is the order in which the rows are actually returned). Furthermore, it eliminates duplicate rows from its result, in the same way as `DISTINCT`, unless `UNION ALL` is used.

### Experiment 1: Union

```
-- same attribution name
select num from T1
UNION
select num from T2;

-- attribution names are same with T1
select * from T1
UNION
select * from T2;
```

**Tips:** First combine all, then do the distinct.

## Experiment 2: Union ALL

```
-- Union all
select num from T1
UNION all
select num from T2;

-- union all + order by
select num from T1
UNION all
select num from T2
order by num;

-- union all + order by II
select * from T1
UNION all
select * from T2
order by num;

-- union all + distinct
select distinct(U.*) from(
select * from T1
UNION all
select * from T2
) U;
```

## INTERSECT and INTERSECT ALL

`INTERSECT` returns all rows that are both in the result of `query1` and in the result of `query2`. Duplicate rows are eliminated unless `INTERSECT ALL` is used.

## Experiment 3: INTERSECT

```
-- intersect
select name from T1
intersect
select name from T2;
```

**Tips:** First intersect all, then do the distinct.

## Experiment 4: INTERSECT ALL

```
-- intersect all
select name from T1
intersect all
select name from T2;
```

```
-- intersect all + distinct
select distinct (I.name)
from (
    select name
    from T1
    intersect all
    select name
    from T2
)I;
```

## EXCEPT

**EXCEPT** returns all rows that are in the result of *query1* but not in the result of *query2*. (This is sometimes called the *difference* between two queries.) Again, duplicates are eliminated unless **EXCEPT ALL** is used.

### Experiment 5: EXCEPT

```
-- except I
select name
from T1
    except
select name
from T2;

-- except II
select name
from T2
    except
select name
from T1;
```

**Tips:** Notice the order of tables.

### Experiment 6: EXCEPT ALL

```
-- EXCEPT ALL
select name
from T1
    except all
select name
from T2;

-- EXCEPT ALL + distinct
select distinct(E.name)
from (
    select name
    from T1
    except all
```

```
select name  
from T2  
) E;
```