Set operations

Jaroslav Porubän, Miroslav Biňas,
Milan Nosáľ (c) 2011 - 2016

Introduction

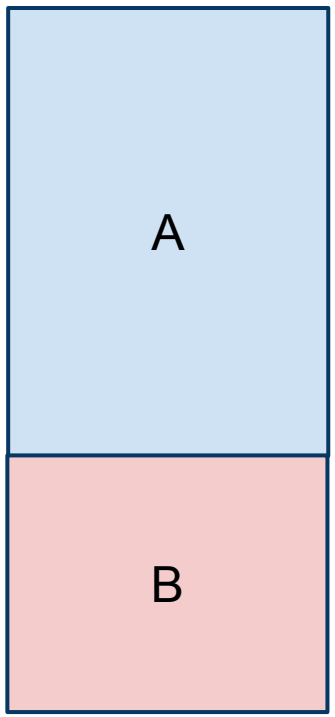
- Operations over mathematical sets
- Set operations in ORACLE SQL:
  - Set union- `UNION`, `UNION ALL`
  - Set intersection - `INTERSECT`
  - Set difference - `MINUS`

Union of tables - `UNION`

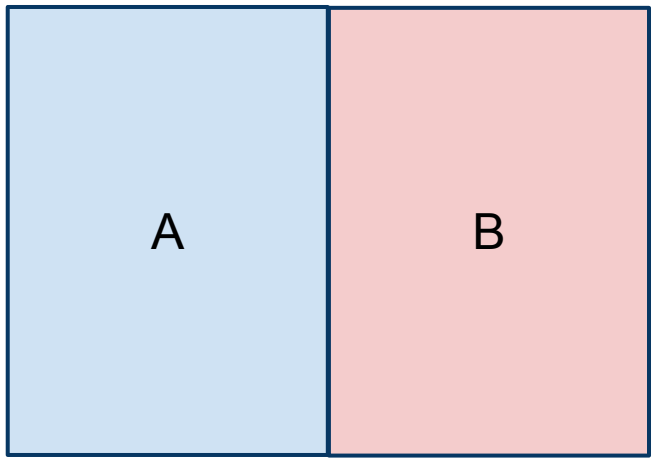
- Creates union of results of two (or more) `SELECT` queries into a single one
- Queries in the union have to have the same relational schema (columns in results):
  - Same data types (or compatible), and
  - Same number of columns
- The result does not contain duplicates
- syntax:

```
SELECT column(s) FROM table1
UNION
SELECT column(s) FROM table2
```

Difference between `UNION` and `JOIN`



`UNION`



`JOIN`

Example of `UNION`

- Usernames of nezbednik's followers together with his friends:

```
SELECT followee
FROM follows
WHERE follower = 'nezbednik'
UNION
SELECT friendWith
FROM friendsWith
WHERE friend = 'nezbednik';
```

`UNION` and `UNION ALL`

- Similar functionality as `UNION`
- Differences with `UNION`:
  - Result containes all tuples, duplicates are not eliminated
  - Result is not sorted (internal property)
  - Faster than `UNION`
- syntax:

```
SELECT column(s) FROM table1
UNION ALL
SELECT column(s) FROM table2
```

Example of `UNION ALL`

```
SELECT followee
FROM follows
WHERE follower = 'nezbednik'
UNION ALL
SELECT friendWith
FROM friendsWith
WHERE friend = 'nezbednik';
```

Properties of `UNION` and `UNION ALL`

- Alias can be used only in the first `SELECT`
- `ORDER BY` can be used only once at the end of the query
- If the columns' data types are not same, but compatible, automatic conversion is performed
- If data types are different, column values must be converted (e.g., using `TO_CHAR()`)
- Union of tables with different number of columns
  - Add a constant to simulate a column

When to use `UNION ALL`

- When the table contains duplicate tuples and we need to keep them in the result
- When the result cannot contain duplicates (no sense to eliminate them)
- If we do not care whether there are duplicates or not

Intersection - `INTERSECT`

- Intersection of two tables contains all tuples that are in both tables
- Can be replaced using `INNER JOIN`
- syntax:

```
SELECT column(s) FROM table1
INTERSECT
SELECT column(s) FROM table2
```

Usage example - INTERSECT

- Usernames of nezbednik's followers that are also his friends:

```
SELECT followee
FROM follows
WHERE follower = 'nezbednik'
INTERSECT
SELECT friendWith
FROM friendsWith
WHERE friend = 'nezbednik';
```

Difference - `MINUS (EXCEPT)`

- Allows us to find out which rows in the first table are not in the second
- Not commutative operation:
  - `A` minus `B` ≠ `B` minus `A`
- Can be replaced using `OUTER JOIN`
- Not a part of standard SQL
- syntax:

```
SELECT column(s) FROM table1
MINUS
SELECT column(s) FROM table2
```

Usage example - MINUS

- Usernames of nezbednik's followers that are not his friends:

```
SELECT followee
FROM follows
WHERE follower = 'nezbednik'
MINUS
SELECT friendWith
FROM friendsWith
WHERE friend = 'nezbednik';
```

Questions?