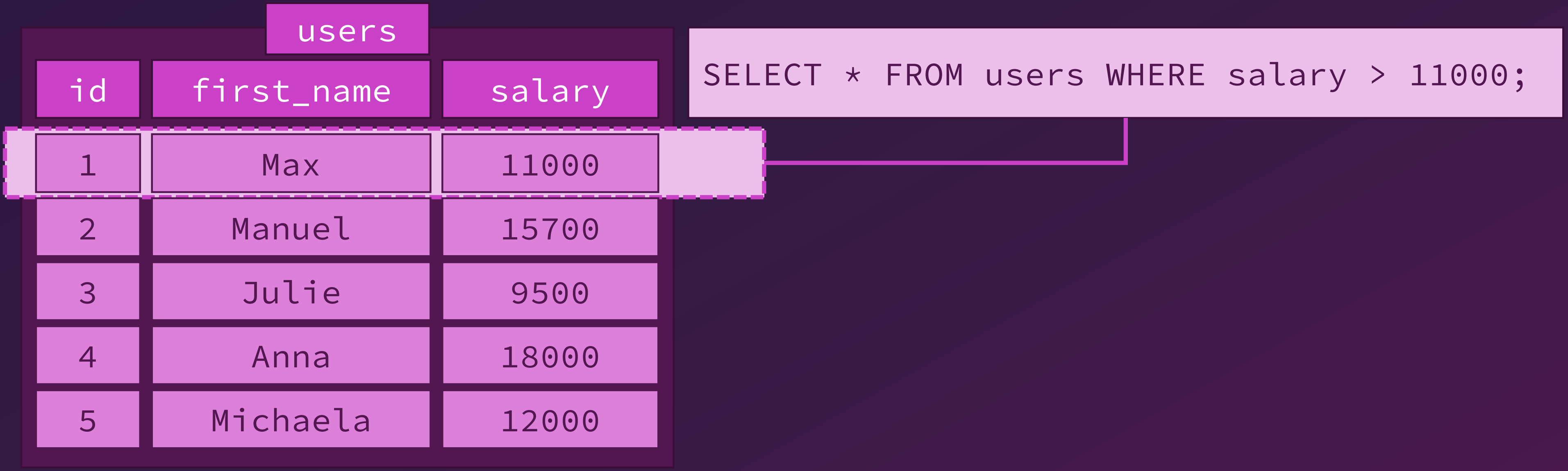
# What Are Indexes?

Indexes can enhance query performance



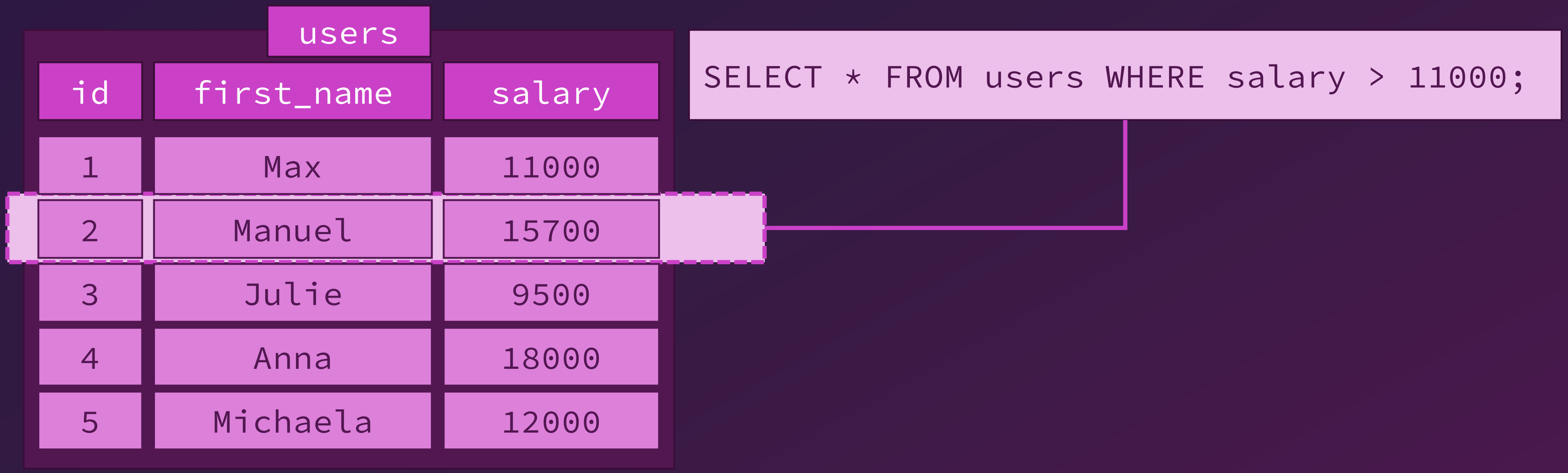
users

| id | first_name | salary |
|---|---|---|
| 1 | Max | 11000 |
| 2 | Manuel | 15700 |
| 3 | Julie | 9500 |
| 4 | Anna | 18000 |
| 5 | Michaela | 12000 |

```sql
SELECT * FROM users WHERE salary > 11000;
```

# What Are Indexes?

Indexes can enhance query performance



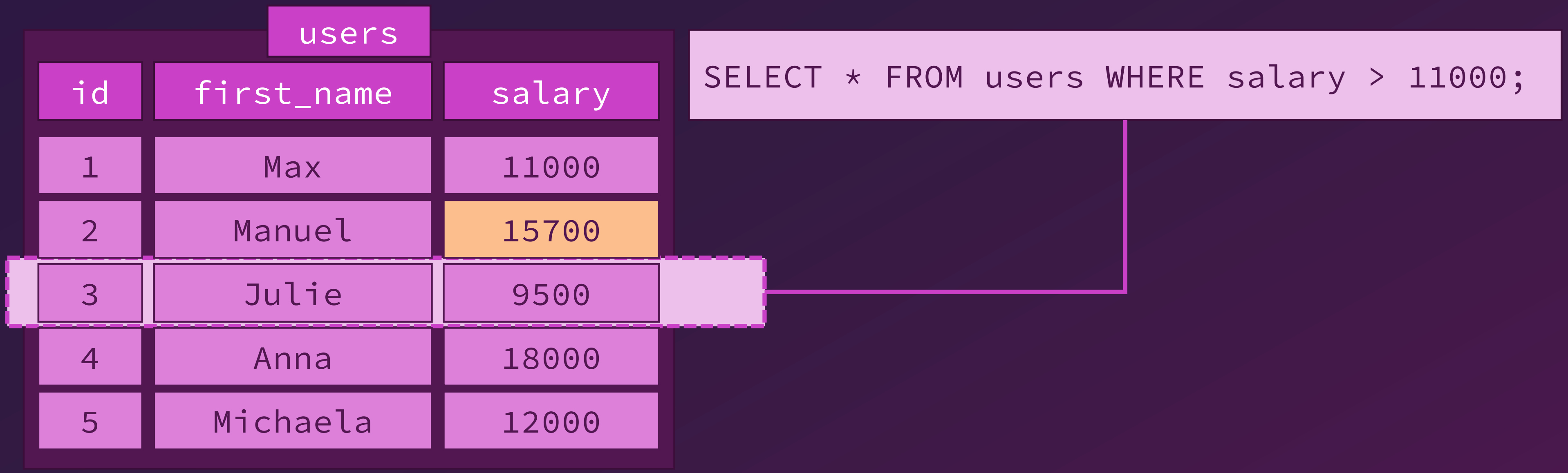
users

| id | first_name | salary |
|---|---|---|
| 1 | Max | 11000 |
| 2 | Manuel | 15700 |
| 3 | Julie | 9500 |
| 4 | Anna | 18000 |
| 5 | Michaela | 12000 |

```
SELECT * FROM users WHERE salary > 11000;
```

# What Are Indexes?

Indexes can enhance query performance



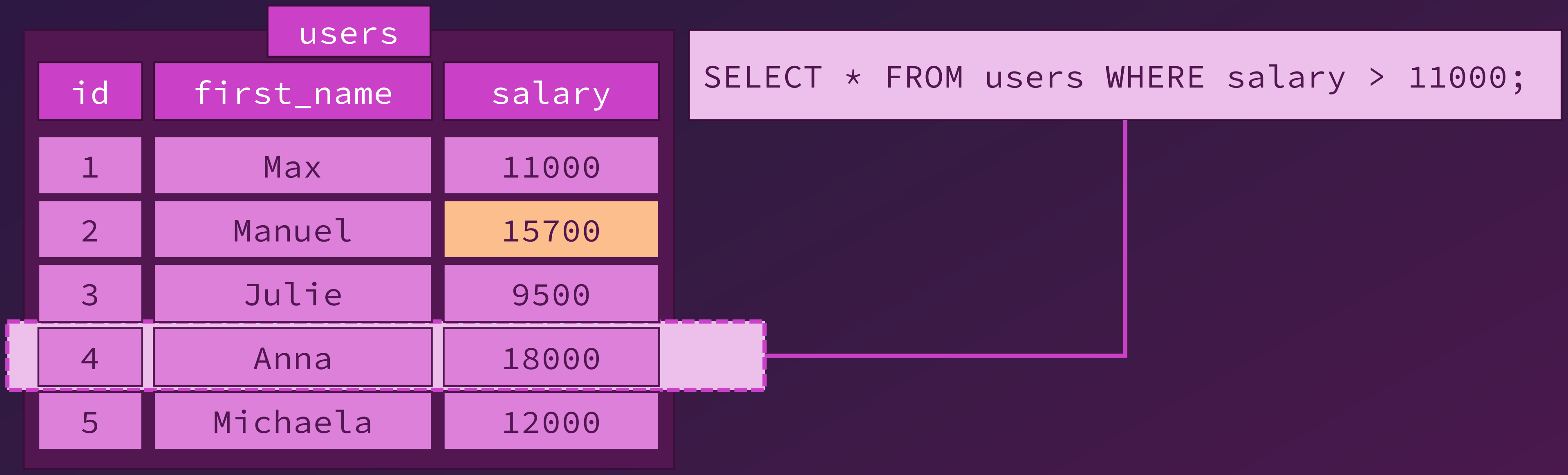
users

| id | first_name | salary |
|---|---|---|
| 1 | Max | 11000 |
| 2 | Manuel | 15700 |
| 3 | Julie | 9500 |
| 4 | Anna | 18000 |
| 5 | Michaela | 12000 |

```
SELECT * FROM users WHERE salary > 11000;
```

# What Are Indexes?

Indexes can enhance query performance



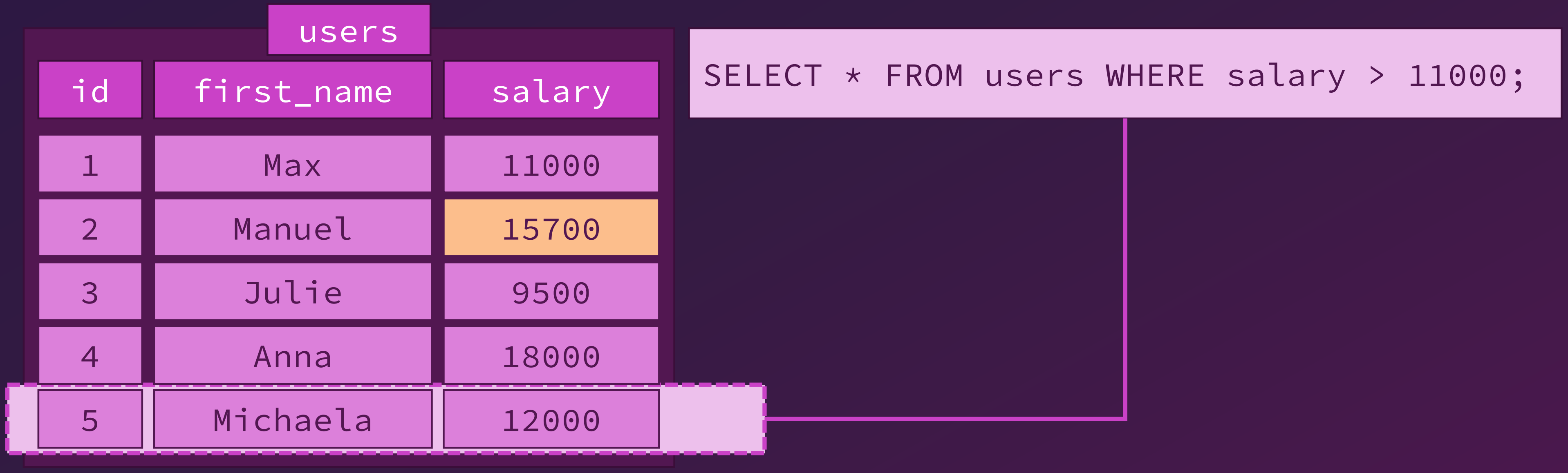
users

| id | first_name | salary |
|---|---|---|
| 1 | Max | 11000 |
| 2 | Manuel | 15700 |
| 3 | Julie | 9500 |
| 4 | Anna | 18000 |
| 5 | Michaela | 12000 |

```
SELECT * FROM users WHERE salary > 11000;
```

# What Are Indexes?

Indexes can enhance query performance



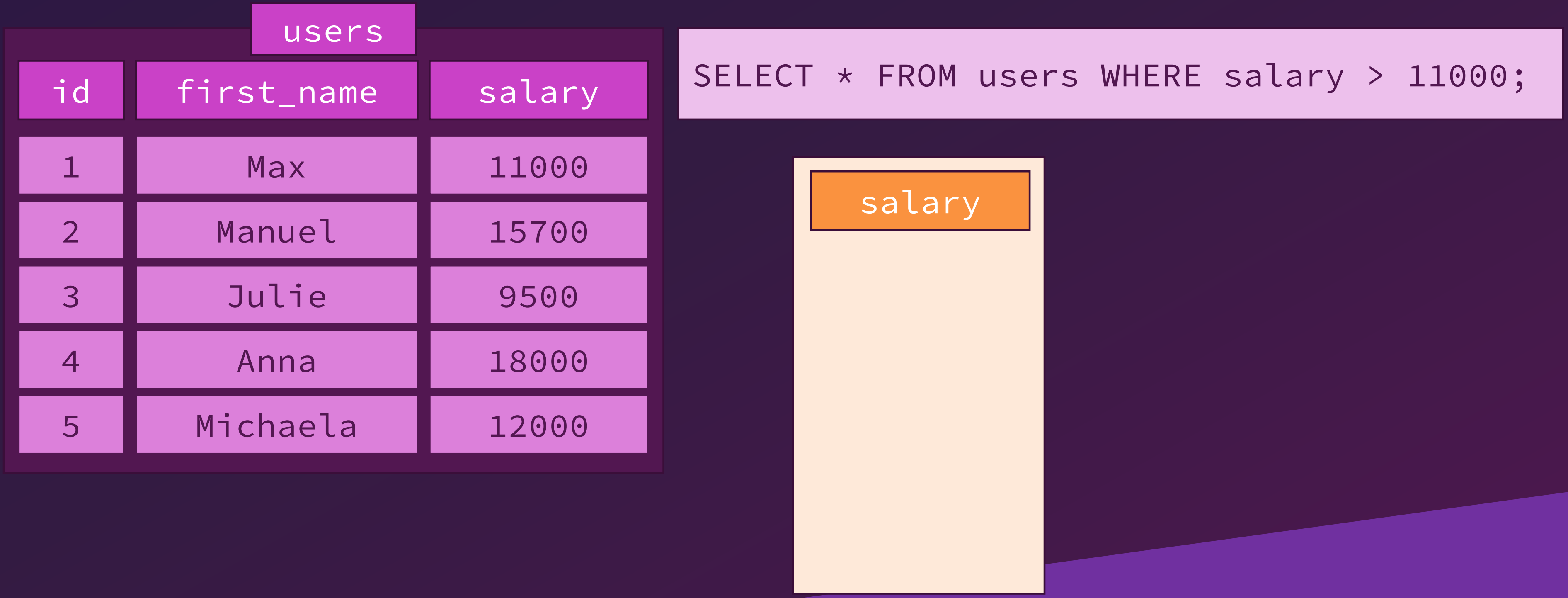
users

| id | first_name | salary |
|---|---|---|
| 1 | Max | 11000 |
| 2 | Manuel | 15700 |
| 3 | Julie | 9500 |
| 4 | Anna | 18000 |
| 5 | Michaela | 12000 |

```
SELECT * FROM users WHERE salary > 11000;
```

# What Are Indexes?

Indexes can enhance query performance



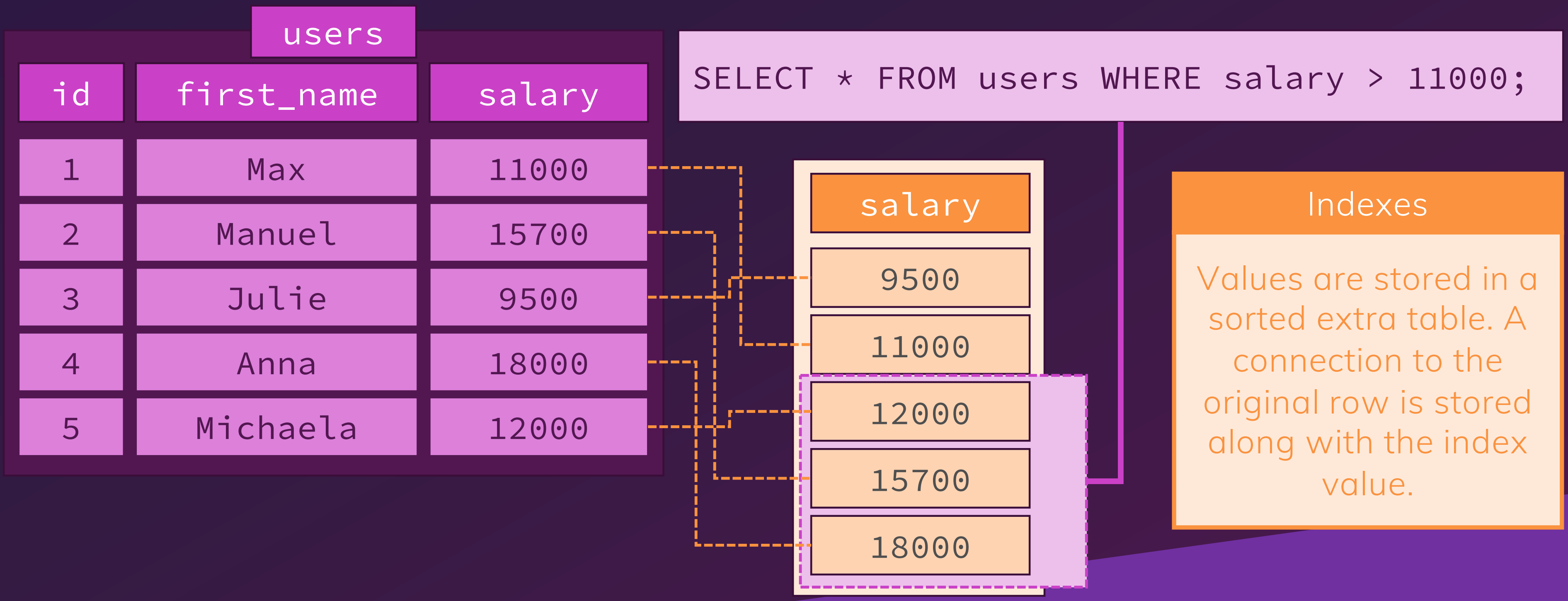
users

| id | first_name | salary |
|---|---|---|
| 1 | Max | 11000 |
| 2 | Manuel | 15700 |
| 3 | Julie | 9500 |
| 4 | Anna | 18000 |
| 5 | Michaela | 12000 |

```sql
SELECT * FROM users WHERE salary > 11000;
```

# What Are Indexes?

Indexes can enhance query performance



users

| id | first_name | salary |
|---|---|---|
| 1 | Max | 11000 |
| 2 | Manuel | 15700 |
| 3 | Julie | 9500 |
| 4 | Anna | 18000 |
| 5 | Michaela | 12000 |

```
SELECT * FROM users WHERE salary > 11000;
```

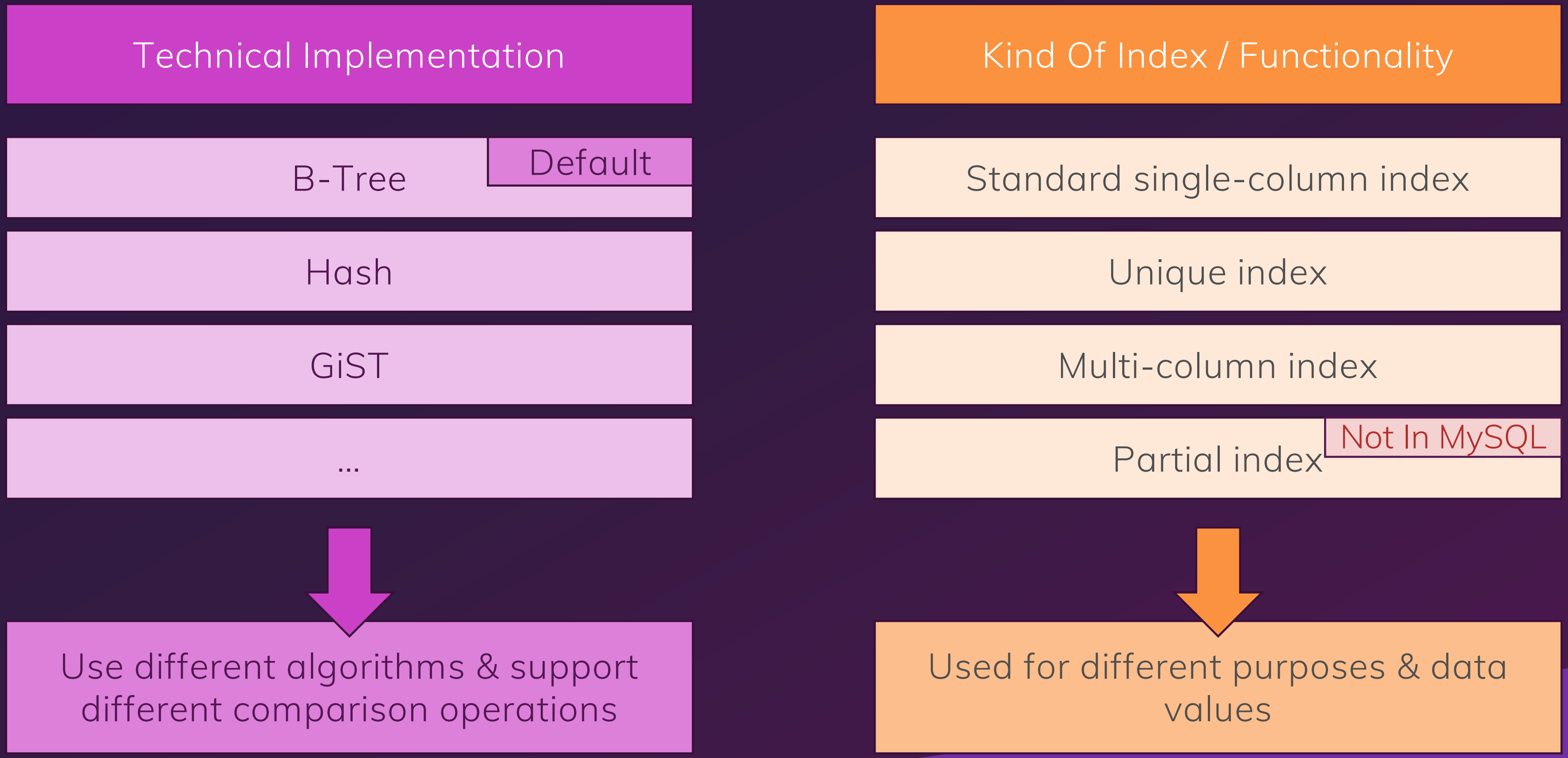
| salary |
|---|
| 9500 |
| 11000 |
| 12000 |
| 15700 |
| 18000 |

Indexes

Values are stored in a sorted extra table. A connection to the original row is stored along with the index value.

# By Default, An Index Is Created For The Primary Key

# Different Types Of Indexes



| Technical Implementation | Kind Of Index / Functionality |
|---|---|
| B-Tree — Default | Standard single-column index |
| Hash | Unique index |
| GiST | Multi-column index |
| … | Partial index — Not In MySQL |
| Use different algorithms & support different comparison operations | Used for different purposes & data values |

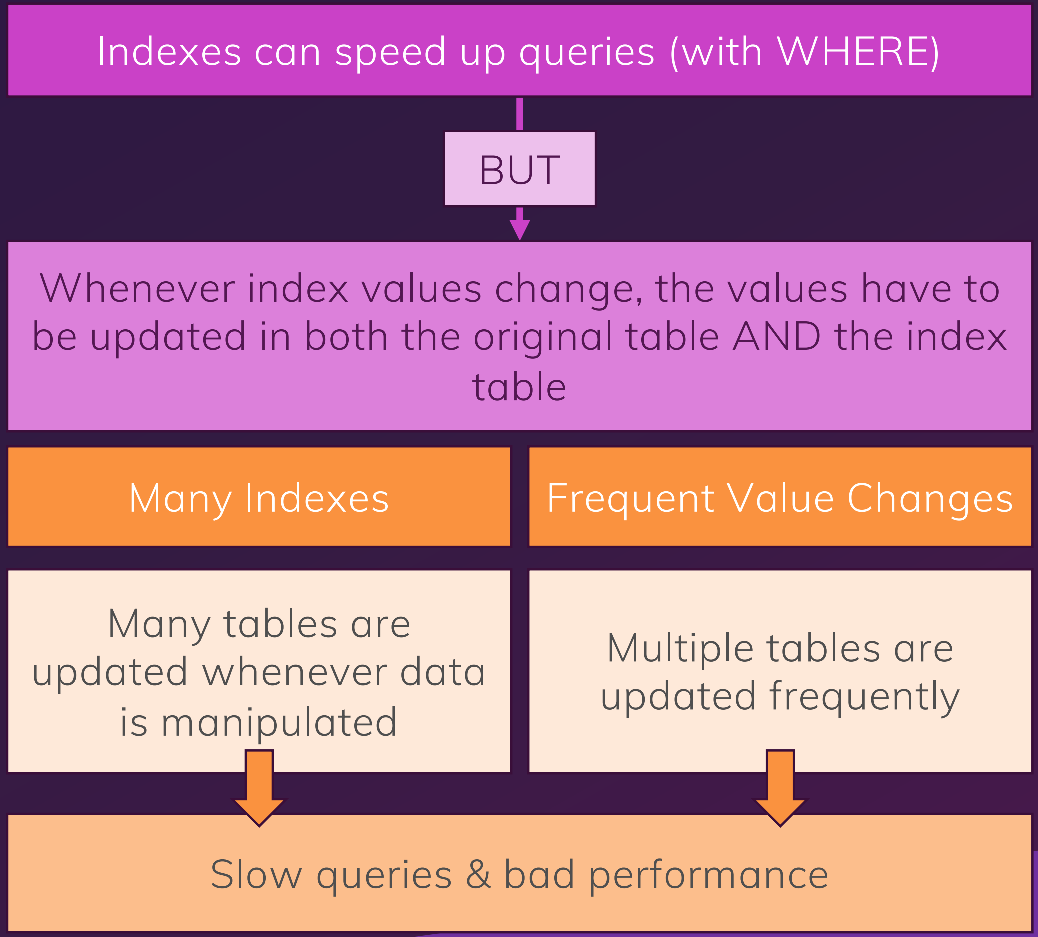
# When To Use Which Type Of Index

| Standard single-column index | Unique index | Multi-column index | Partial index |
|---|---|---|---|
| Use if you frequently use a column in the WHERE clause of your queries | | Use if you frequently use multiple columns in the WHERE clause of your queries | Use if only a subset of column values should be optimized for querying |
| | In addition, duplicate values are blocked | | |

ACADEMIND

# Don’t Use Too Many Indexes!

# The Problem With Too Many Indexes



Indexes can speed up queries (with WHERE)

BUT

Whenever index values change, the values have to be updated in both the original table AND the index table

Many Indexes

Frequent Value Changes

Many tables are updated whenever data is manipulated

Multiple tables are updated frequently

Slow queries & bad performance