S03-L03 Composite index

```
SELECT
        department_employee_rel.*
FROM `department_employee_rel`
WHERE 1=1
        AND `department_employee_rel`.`department_id` = 3
        AND `department_employee_rel`.`employee_id` IN (10005, 10006, 10007)
        AND `department_employee_rel`.`from_date` = '1989-09-12'
        AND `department_employee_rel`.`to_date` IS NULL
;
```

Partially used index

Check `salary` table - it doesn't have a single index on `employee_id`.

But there's a composite index on multiple attributes:

- `employee_id`
- `from_date`
- `to_date`

```
SELECT * /* Ignore index - 2-3 seconds */
FROM `salary` IGNORE INDEX (`ak_salary`)
WHERE 1=1
        AND `salary`.`employee_id` = 499998
;

SELECT * /* Use index - very fast */
FROM `salary` USE INDEX (`ak_salary`)
WHERE 1=1
        AND `salary`.`employee_id` = 499998
;
```

You can also create a new index `idx_employee_id`, but not needed thanks to the composite `ak_salary`.

3 column index

If you have a three-column index on `(col1, col2, col3)`, you have indexed search capabilities on

- `(col1)`
- `(col1, col2)`
- and `(col1, col2, col3)`

= the order of the columns in an index matters if you'd like to query partially.