

VERSION 2.0  
FEBRUARI, 2020



# DESAIN DATABASE

## MODUL 3: AGGREGATE FUNCTION

TIM PENYUSUN:

- AGUS EKO MINARNO
- BYAS AGENG HAQIQIE

PRESENTED BY: LABORATORIUM INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH MALANG

## DESAIN DATABASE

---

### CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa memahami konsep join, agregat function, grouping function, dan sorting !
2. Mahasiswa mampu mengimplementasikan join, agregat function, grouping function, dan sorting !

---

### SUB CAPAIAN PEMBELAJARAN MATA KULIAH

1. Memahami konsep join, agregat function, grouping function, dan sorting.
2. Mampu mengimplementasikan join, agregat function, grouping function, dan sorting

---

### KEBUTUHAN HARDWARE & SOFTWARE

1. Oracle XE

---

### MATERI POKOK

1. Join  
Join Adalah salah satu konstruksi dasar dari SQL dan basis data. Join dapat didefinisikan sebagai kombinasi record dari dua tabel atau lebih didalam basis data relasional dan menghasilkan sebuah tabel (tempory) baru yang disebut Joined Tabel.
  - a. Inner Join  
Digunakan untuk menghasilkan baris data dengan menggabungkan 2 buah table atau lebih.
  - b. Left Join  
Left Join merupakan penggabungan tabel dimana data akan ditampilkan secara keseluruhan pada tabel pertama (kiri) namun record pada tabel kedua (kanan) yang kosong akan ditampilkan dengan isi NULL.
  - c. Right Join  
Right Join memiliki fungsi yang bertolak belakang dengan left join, dimana right join akan menampilkan data secara keseluruhan pada tabel kedua (kanan), namun NULL pada tabel pertama (kiri)
  - d. Union Join  
Digunakan untuk menghasilkan baris data jika ada data yang sama pada salah satu tabel.
2. Aggregate Functions  
Fungsi yang bekerja terhadap suatu group.
  - a. Count  
Untuk menghitung baris dari suatu tabel.
  - b. Sum  
Untuk menghitung jumlah dari sekumpulan kolom dari suatu tabel.
  - c. Avg  
Untuk mendapatkan nilai rata-rata nilai kolom tertentu pada suatu tabel.
  - d. Max  
Untuk mendapatkan nilai terbesar dari sekumpulan nilai kolom dari suatu tabel.
  - e. Min

Untuk mendapatkan nilai terkecil dari sekumpulan nilai kolom dari suatu tabel

### 3. Grouping Functions

#### a. Group by

Untuk menentukan bagaimana baris-baris akan dikelompokkan

#### b. Having

Untuk menentukan pengelompokkan-pengelompokkan yang akan ditampilkan, lebih jauh lagi membatasi pengelompokkan berdasarkan informasi agregat

### 4. Sorting

Suatu proses menyusun kembali himpunan obyek menggunakan aturan tertentu. Macam-macam sorting :

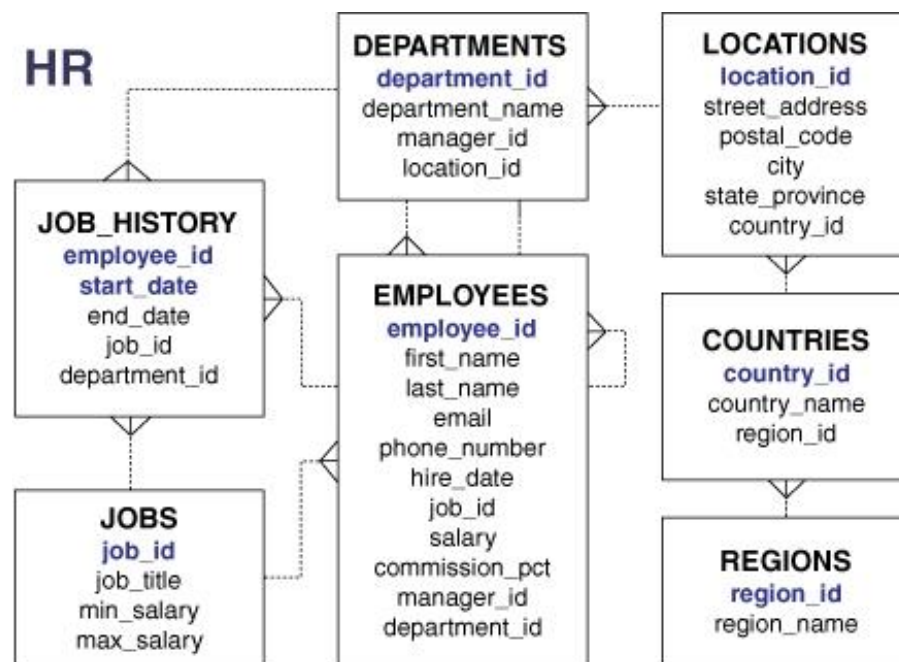
#### a. Urut naik (ascending)

Mengurutkan dari data yang mempunyai nilai paling kecil sampai paling besar.

#### b. Urut turun (descending)

Mengurutkan dari data yang mempunyai nilai paling besar sampai paling kecil.

## MATERI PRAKTIKUM



### 1. Join

#### a. Inner Join

```

SELECT d.department_name,
       e.employee_name
FROM   departments d
       JOIN employees e ON d.department_id = e.department_id
WHERE  d.department_id >= 30
ORDER BY d.department_name;
  
```

#### b. Left Join

```

SELECT d.department_name,
  
```

```

        e.employee_name
FROM   departments d
      LEFT OUTER JOIN employees e ON d.department_id = e.department_id
WHERE  d.department_id >= 30
ORDER BY d.department_name, e.employee_name;

```

c. Right Join

```

SELECT d.department_name,
       e.employee_name
FROM   employees e
      RIGHT OUTER JOIN departments d ON e.department_id = d.department_id
WHERE  d.department_id >= 30
ORDER BY d.department_name, e.employee_name;

```

d. Union Join

```

SELECT d.department_name,
       e.employee_name
FROM   employees e, departments d
WHERE  e.department_id = d.department_id (+)
UNION ALL
SELECT d.department_name,
       e.employee_name
FROM   departments d, employees e
WHERE  d.department_id = e.department_id (+)
AND    e.employee_name IS NULL
ORDER BY 1, 2;

```

2. Aggregate Functions

a. Count

```

Select count (*) from employees;

```

b. Sum

```

SELECT job_id, SUM(salary) PAYROLL FROM employees WHERE job_id NOT LIKE '%REP%'
GROUP BY job_id HAVING SUM(salary) > 13000 ORDER BY SUM(salary);

```

c. Avg

```

SELECT department_id, AVG(salary) FROM employees WHERE AVG(salary) > 8000 GROUP BY
department_id;

```

d. Max

```

SELECT MAX(AVG(salary)) FROM employees GROUP BY department_id;

```

e. Min

```

SELECT MIN(AVG(salary)) FROM employees GROUP BY department_id;

```

3. Grouping functions

a. Group by

```

SELECT department_id, AVG(salary) FROM employees GROUP BY department_id;

```

b. Having

```

SELECT department_id, MAX(salary) FROM employees GROUP BY department_id HAVING
MAX(salary) > 10000;

```

4. Sorting

a. Ascending

- Select \* from employees order by first\_name asc;
- b. Descending  
Select \* from employees order by first\_name asc;

## LEMBAR KERJA

### KEGIATAN 1

Buatlah masing-masing 1 query untuk masing-masing fungsi dengan live query :

- Inner Join
- Left join
- Right Join
- Union Join
- Count
- Sum
- Avg
- Min
- Max
- Group by
- Having
- Ascending
- Descending

### KEGIATAN 2

Buatlah query untuk masing masing soal dibawah ini , dengan output berikut:

- a. Tampilkan ID negara dan jumlah kota yang dimiliki di negara tersebut

COUNTRY_ID	COUNT(*)
US	4
SG	1
CA	2
CH	2
IT	2
MX	1
CN	1
DE	1
JP	2
IN	1
More than 10 rows available. Increase rows selector to view more rows.	

(output)

- b. Tampilkan ID pekerjaan untuk pekerjaan dengan gaji rata-rata lebih dari 10.000

JOB_ID	AVG(SALARY)
AC_MGR	12008
PU_MAN	11000
AD_VP	17000
FI_MGR	12008
SA_MAN	12200
MK_MAN	13000
AD PRES	24000

(output)

---

## RUBRIK PENILAIAN

1. Kegiatan 1
  - a. Inner Join
  - b. Left join
  - c. Right Join
  - d. Union Join
  - e. Count
  - f. Sum
  - g. Avg
  - h. Min
  - i. Max
  - j. Group by
  - k. Having
  - l. Ascending
  - m. Descending

Masing-masing point bernilai 10.

2. Kegiatan 2
 

Masing-masing point bernilai 10.