Nama : F.X. Airell Valerio Satrio Wibowo

Kode Peserta : JVSB001ONL013

# Sesi 12 – Requirements 1

## Langkah – 1 : Membangun Framework

Company Name : PT Melodi Indah  
Tujuan : Database Informasi Pemusik yang rekaman di perusahaan.

Tabel Musisi:

* ID Musisi
* Nama
* SSN (NIK atau Social Security Number)
* Alamat
* Nomor Telepon

\* Beberapa punya alamat yang sama  
\*\* Beberapa tidak mencantumkan alamat

Tabel Instrumen:

* ID Instrumen
* Nama Instrumen (Gitar, Flute, dll)
* Kunci Musik (C, B#, dll)

Tabel Album:

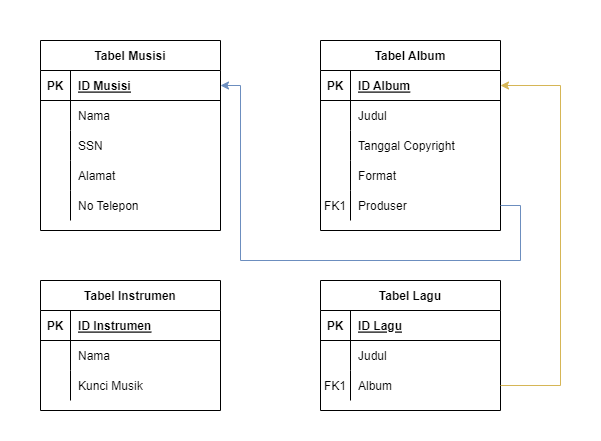
* ID Album
* Judul Album
* Tanggal Copyright
* Format (CD, DA, DS)
* Produser

Tabel Lagu:

* ID Lagu
* Judul Lagu
* Album

\* Musisi mungkin memainkan beberapa instrumen  
\*\* Instrumen mungkin dimainkan beberapa musisi  
\*\*\* Album mempunyai banyak lagu (lebih dari 1 lagu cukup)  
\*\*\*\* Tidak ada lagu yang ada di banyak album (lagu hanya tampil tempat di 1 album)  
\*\*\*\*\* Setiap lagu ditampilkan oleh satu atau lebih musisi  
\*\*\*\*\*\* Setiap musisi bisa menampilkan lebih dari satu lagu  
\*\*\*\*\*\*\* Setiap album mempunyai musisi sebagai produsernya  
\*\*\*\*\*\*\*\* Seorang musisi mungkin memproduksi lebih dari satu album

## Langkah – 2 : Membangun Skema Tabel dan Database



Gambar . Skema Database Requirements 1

## Langkah – 3 : Membuat Tabel

1. Membuat Tabel Musisi

DROP TABLE IF EXISTS `tabel\_musisi`;

CREATE TABLE `tabel\_musisi` (

`ID\_Musisi` int(11) NOT NULL AUTO\_INCREMENT,

`Nama` varchar(50) NOT NULL,

`NIK` int(20) NOT NULL,

`Alamat` text NOT NULL,

`No\_Telepon` int(15) NOT NULL,

PRIMARY KEY (`ID\_Musisi`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO tabel\_musisi (ID\_Musisi, Nama, NIK, Alamat, No\_Telepon) VALUES

('001', 'Olivia Hye', '12345', 'Eden', '6212345'),

('002', 'Chuu', '12346', 'Eden', '6212346'),

('003', 'Gowon', '12347', 'Eden', '6212347'),

('004', 'Yves', '12348', 'Eden', '6212348'),

('005', 'Choerry', '12349', '', '6212349'),

('006', 'Jinsoul', '12340', '', '6212340'),

('007', 'Kim Lip', '12341', '', '6212341'),

('008', 'Vivi', '12342', 'Korea', '6212342'),

('009', 'Yeojin', '12343', 'Korea', '6212343'),

('010', 'Haseul', '12344', 'Korea', '6212344'),

('011', 'Hyunjin', '123455', 'Korea', '62123455'),

('012', 'Heejin', '123456', 'Korea', '62123456');

1. Membuat Tabel Instrument

DROP TABLE IF EXISTS `tabel\_instrument`;

CREATE TABLE `tabel\_instrument` (

`ID\_Instrument` int(11) NOT NULL AUTO\_INCREMENT,

`Nama` varchar(20) NOT NULL,

`Kunci` char(5) NOT NULL,

PRIMARY KEY (`ID\_Instrument`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO tabel\_instrument (ID\_Instrument, Nama, Kunci) VALUES

('001', 'Guitar', 'A'), ('002', 'Guitar', 'C'),

('003', 'Bass', 'D'), ('004', 'Bass', 'F'),

('005', 'Piano', 'A'), ('006', 'Piano', 'B'),

('007', 'Piano', 'C'), ('008', 'Piano', 'D'),

('009', 'Piano', 'E'), ('010', 'Piano', 'F'),

('011', 'Piano', 'G'),

('012', 'Digital Sound', ''),

('013', 'Drum', ''),

('014', 'Saxophone', 'D'),

('015', 'Flute', 'E');

1. Membuat Tabel Album

DROP TABLE IF EXISTS `tabel\_album`;

CREATE TABLE `tabel\_album` (

`ID\_Album` int(11) NOT NULL AUTO\_INCREMENT,

`Judul` varchar(50) NOT NULL,

`Tanggal\_Copyright` date NOT NULL,

`Format` enum('CD','DA','DS') NOT NULL,

`Produser` int(11) NOT NULL,

PRIMARY KEY (`ID\_Album`),

KEY `Produser` (`Produser`),

CONSTRAINT `tabel\_album\_ibfk\_2` FOREIGN KEY (`Produser`) REFERENCES `tabel\_musisi` (`ID\_Musisi`) ON DELETE NO ACTION ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO tabel\_album (ID\_Album, Judul, Tanggal\_Copyright, Format, Produser) VALUES

('1', 'Love&Live', '2017-03-13', 'CD', '12'),

('2', 'Girl Front', '2017-03-13', 'CD', '7'),

('3', 'Love4Eva', '2017-09-21', 'CD', '4'),

('4', '+ +', '2018-08-20', 'CD', '1'),

('5', 'x x', '2019-02-19', 'CD', '2'),

('6', '#', '2020-02-05', 'CD', '3'),

('7', '12.00', '2020-10-19', 'CD', '4'),

('8', '&', '2021-06-28', 'CD', '5'),

('9', 'Flip That', '2022-06-20', 'CD', '6');

1. Membuat Tabel Lagu

DROP TABLE IF EXISTS `tabel\_lagu`;

CREATE TABLE `tabel\_lagu` (

`ID\_Lagu` int(11) NOT NULL AUTO\_INCREMENT,

`Judul` varchar(50) NOT NULL,

`Album` int(11) NOT NULL,

PRIMARY KEY (`ID\_Lagu`),

KEY `Album` (`Album`),

CONSTRAINT `tabel\_lagu\_ibfk\_4` FOREIGN KEY (`Album`) REFERENCES `tabel\_album` (`ID\_Album`) ON DELETE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO tabel\_lagu (ID\_Lagu, Judul, Album) VALUES

('1', 'Love&Live', '1'), ('2', 'Sonatine', '1'),

('3', 'Girl Front', '2'), ('4', 'Sweet Crazy Love', '2'),

('5', 'Love4Eva', '3'),

('6', 'Favourite', '4'), ('7', 'Hi High', '4'),

('8', 'Butterfly', '5'), ('9', 'Satelite', '5'),

('10', 'So What', '6'),

('11', 'Why Not', '7'),

('12', 'Paint The Town', '8'),

('13', 'Flip That', '9');

1. Membuat Tabel Hubungan Musisi dan Instrumen

DROP TABLE IF EXISTS `tabelc\_musinst`;

CREATE TABLE `tabelc\_musinst` (

`ID\_Musisi` int(11) NOT NULL,

`ID\_Instrument` int(11) NOT NULL,

KEY `ID\_Musisi` (`ID\_Musisi`),

KEY `ID\_Instrument` (`ID\_Instrument`),

CONSTRAINT `tabelc\_musinst\_ibfk\_1` FOREIGN KEY (`ID\_Musisi`) REFERENCES `tabel\_musisi` (`ID\_Musisi`) ON DELETE NO ACTION,

CONSTRAINT `tabelc\_musinst\_ibfk\_2` FOREIGN KEY (`ID\_Instrument`) REFERENCES `tabel\_instrument` (`ID\_Instrument`) ON DELETE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO tabelc\_musinst (ID\_Musisi, ID\_Instrument) VALUES

('1', '1'), ('1', '2'),

('4', '5'), ('4', '6'), ('4', '7'), ('4', '8'),

('4', '9'), ('4', '10'), ('4', '11'), ('4', '5'),

('5', '2'), ('5', '7'), ('5', '12'), ('12', '14');

1. Membuat Tabel Hubungan Musisi dan Lagu

DROP TABLE IF EXISTS `tabelc\_lagumus`;

CREATE TABLE `tabelc\_lagumus` (

`ID\_Lagu` int(11) NOT NULL,

`ID\_Musisi` int(11) NOT NULL,

KEY `ID\_Lagu` (`ID\_Lagu`),

KEY `ID\_Musisi` (`ID\_Musisi`),

CONSTRAINT `tabelc\_lagumus\_ibfk\_4` FOREIGN KEY (`ID\_Lagu`) REFERENCES `tabel\_lagu` (`ID\_Lagu`) ON DELETE NO ACTION ON UPDATE CASCADE,

CONSTRAINT `tabelc\_lagumus\_ibfk\_5` FOREIGN KEY (`ID\_Musisi`) REFERENCES `tabel\_musisi` (`ID\_Musisi`) ON DELETE NO ACTION ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `tabelc\_lagumus` (`ID\_Lagu`, `ID\_Musisi`) VALUES

('1', '12'), ('1', '11'), ('1', '10'), ('1', '8'),

('2', '12'), ('2', '11'), ('2', '10'), ('2', '8'),

('3', '7'), ('3', '6'), ('3', '5'),

('4', '7'), ('4', '6'), ('4', '5'),

('5', '1'), ('5', '2'), ('5', '3'), ('5', '4'),

('6', '1'), ('6', '2'), ('6', '3'), ('6', '4'),

('7', '1'), ('7', '2'), ('7', '3'), ('7', '4'), ('7', '5'), ('7', '6'),

('7', '7'), ('7', '8'), ('7', '9'), ('7', '10'), ('7', '11'), ('7', '12'),

('8', '1'), ('8', '2'), ('8', '3'), ('8', '4'), ('8', '5'), ('8', '6'),

('8', '7'), ('8', '8'), ('8', '9'), ('8', '10'), ('8', '11'), ('8', '12'),

('9', '1'), ('9', '2'), ('9', '3'), ('9', '4'), ('9', '5'), ('9', '6'),

('9', '7'), ('9', '8'), ('9', '9'), ('9', '10'), ('9', '11'), ('9', '12'),

('10', '1'), ('10', '2'), ('10', '3'), ('10', '4'), ('10', '5'), ('10', '6'),

('10', '7'), ('10', '8'), ('10', '9'), ('10', '10'), ('10', '11'),

('10', '12'),

('11', '1'), ('11', '2'), ('11', '3'), ('11', '4'), ('11', '5'), ('11', '6'),

('11', '7'), ('11', '8'), ('11', '9'), ('11', '10'), ('11', '11'),

('11', '12'),

('12', '1'), ('12', '2'), ('12', '3'), ('12', '4'), ('12', '5'), ('12', '6'),

('12', '7'), ('12', '8'), ('12', '9'), ('12', '10'), ('12', '11'),

('12', '12'),

('13', '1'), ('13', '2'), ('13', '3'), ('13', '4'), ('13', '5'), ('13', '6'),

('13', '7'), ('13', '8'), ('13', '9'), ('13', '10'), ('13', '11'),

('13', '12');

## Langkah – 4 : Menampilkan Tabel

1. Menampilkan Tabel Musisi

Tabel menampilkan beberapa musisi dengan alamat yang sama ✓

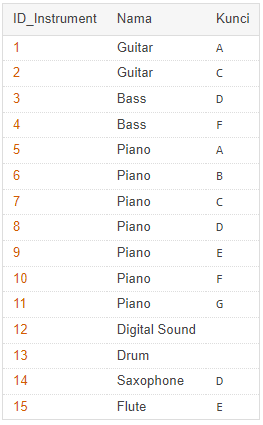
Tabel juga menampilkan beberapa musisi yang tidak mencantumkan alamat ✓



Gambar . Tabel Musisi

1. Menampilkan Tabel Instrument

Beberapa instrument dibedakan berdasarkan kunci dan model.

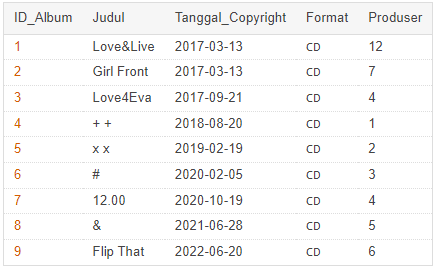


Gambar . Tabel Instrument

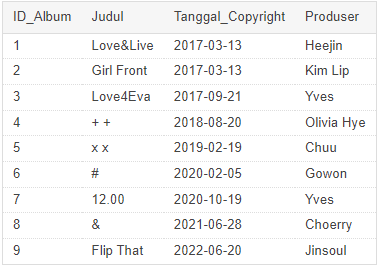
1. Menampilkan Tabel Album

Tabel menampilkan tepat 1 musisi yang memproduksi sebuah album ✓

Tabel juga menampilkan album yang diproduksi oleh 1 musisi yang sama ✓



Gambar . Tabel Album



Gambar . Tabel Album dengan Pengarang

QUERY:

CREATE VIEW album\_produser

AS SELECT ID\_Album, Judul, Tanggal\_Copyright, Nama AS Produser

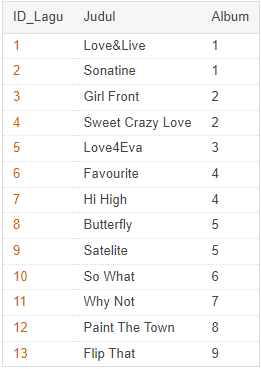
FROM tabel\_album a, tabel\_musisi b

WHERE a.Produser = b.ID\_Musisi;

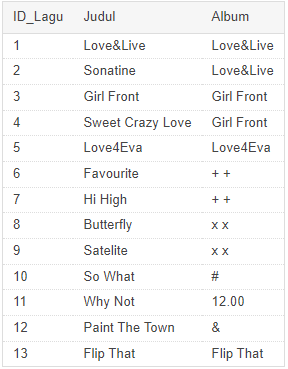
1. Menampilkan Tabel Lagu

Tabel menampilkan minimal 1 lagu terdapat pada suatu album ✓

Tabel menampilkan lagu berada tepat di 1 album ✓



Gambar . Tabel Lagu



Gambar . Tabel Lagu dengan Album

QUERY:

CREATE VIEW album\_lagu

AS SELECT ID\_Lagu, a.Judul, b.Judul AS Album

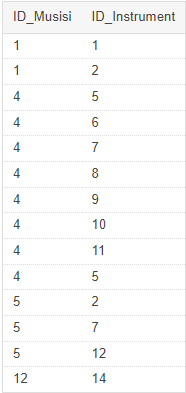
FROM tabel\_lagu a, tabel\_album b

WHERE a.Album = b.ID\_Album;

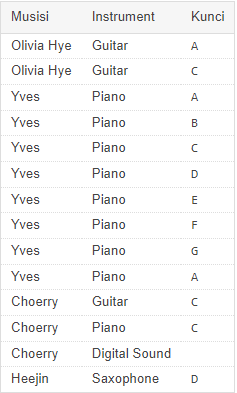
1. Menampilkan Tabel Hubungan Musisi dan Instrument

Tabel menampilkan musisi dapat memainkan beberapa instrument ✓

Tabel juga menampilkan instrument dapat dimainkan beberapa musisi ✓



Gambar . Tabel Musisi dengan Instrument



Gambar . Tabel Musisi dengan Instrument dan Kuncinya

QUERY:

CREATE VIEW musisi\_instrument

AS SELECT a.Nama AS Musisi, b.Nama AS Instrument, Kunci

FROM tabel\_musisi a, tabel\_instrument b, tabelc\_musinst c

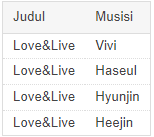
WHERE c.ID\_Musisi = a.ID\_Musisi AND c.ID\_Instrument = b.ID\_Instrument;

1. Menampilkan Tabel Hubungan Musisi dan Lagu

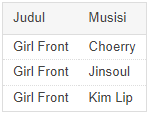
Tabel menampilkan musisi dapat mengisi lebih dari 1 lagu ✓

Tabel juga menampilkan lagu dapat dimainkan beberapa musisi ✓

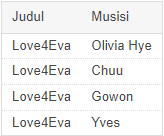
Tabel akan menampilkan tampilan view untuk mempersingkat penampilan tabel. Tabel akan menampilkan beberapa lagu yang dimainkan beberapa musisi. Tabel tidak akan menampilkan keseluruhan hubungan musisi dengan lagu, karena terlalu panjang.



Gambar . Tabel Musisi dengan Lagu Love&Live



Gambar . Tabel Musisi dengan Lagu Girl Front



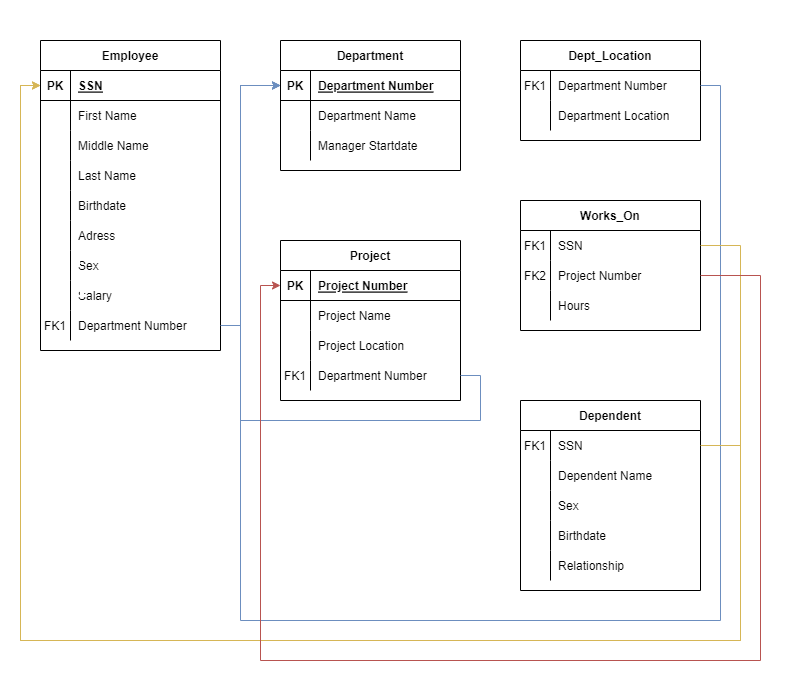
Gambar . Tabel Musisi dengan Lagu Love4Eva



Gambar . Tabel Musisi dengan Lagu Butterfly

# Sesi 12 – Requirements 2

## Langkah – 1 : Membangun Skema Tabel dan Database

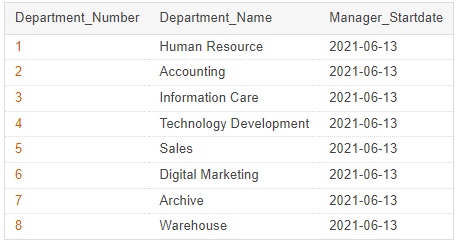


Gambar . Skema Database Requirements 2

## Langkah – 2 : Asumsi dan Pengisian Tabel

Seluruh tabel diisi dengan data acak. Data dimasukkan tanpa melihat kebutuhan kemunculan jawaban sebagai tabel. Apabila terdapat pertanyaan yang tidak memiliki jawaban, berarti tidak disesuaikan agar menampilkan jawaban dan jawaban sudah berdasarkan data tabel yang dimasukkan.

1. Menampilkan Tabel Departemen



Gambar . Tabel Departemen

1. Menampilkan Tabel Employee



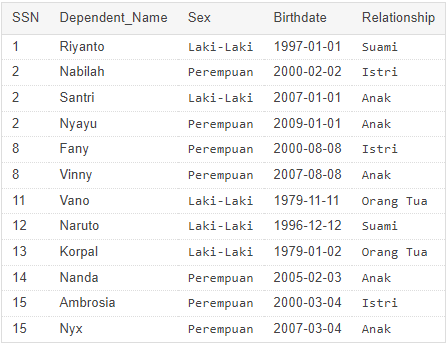
Gambar . Tabel Employee

1. Menampilkan Tabel Project



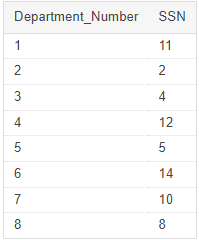
Gambar . Tabel Project

1. Menampilkan Tabel Dependent



Gambar . Tabel Dependent

1. Menampilkan Tabel Manager



Gambar . Tabel Manager

\* Tabel Works On dan Tabel Department Location tidak ditampilkan, karena memuat cukup banyak data, sehingga lebih baik tidak ditampilkan.

\*\* QUERY data tidak ditampilkan karena buka merupakan pokok dari requirements ini. QUERY data ditampilkan pada file berbeda.

## Langkah – 3 : Menjawab Pertanyaan

1. Tampilkan dependent\_name dan relationship dengan employee yang namanya diawali huruf R ?



Gambar . Hasil Nomor 1

QUERY:

CREATE VIEW no\_1

AS SELECT First\_Name, Middle\_Name, Last\_Name, Dependent\_Name, Relationship

FROM employee, dependent

WHERE First\_Name LIKE 'R%';

1. Banyaknya employee yang mengerjakan project PNum = 1 ?



Gambar . Hasil Nomor 2

QUERY:

CREATE VIEW no\_2

AS SELECT COUNT(Project\_Number) AS Banyak\_Employee

FROM works\_on

WHERE Project\_Number = 1;

1. Banyaknya employee yang memiliki salary lebih dari 3500000 ?



Gambar . Hasil Nomor 3

QUERY:

CREATE VIEW no\_3

AS SELECT COUNT(Salary) AS Banyak\_Employee

FROM employee

WHERE Salary > 3500000;

1. Banyaknya project yang dikerjakan DNum =2 ?



Gambar . Hasil Nomor 4

QUERY:

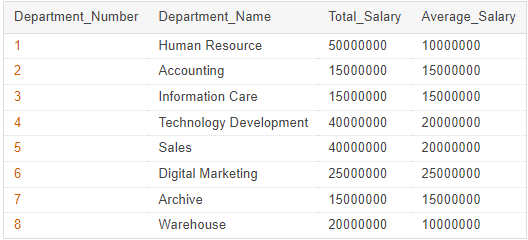
CREATE VIEW no\_4

AS SELECT COUNT(Department\_Number) AS Banyak\_Project

FROM project

WHERE Department\_Number = 2;

1. Hitung total dan rata-rata salary dari setiap departemen ?



Gambar . Hasil Nomor 5

QUERY:

CREATE VIEW no\_5

AS SELECT a.Department\_Number, Department\_Name,

SUM(Salary) Total\_Salary,

AVG(Salary) Average\_Salary

FROM department a, employee b

WHERE a.Department\_Number = b.Department\_Number

GROUP BY a.Department\_Number;

1. Banyaknya employee dari setiap department dan urutkan berdasarkan employee terbanyak ?



Gambar . Hasil Nomor 6

QUERY:

CREATE VIEW no\_6

AS SELECT a.Department\_Number, Department\_Name,

COUNT(b.Department\_Number) AS Total\_Employee

FROM department a, employee b

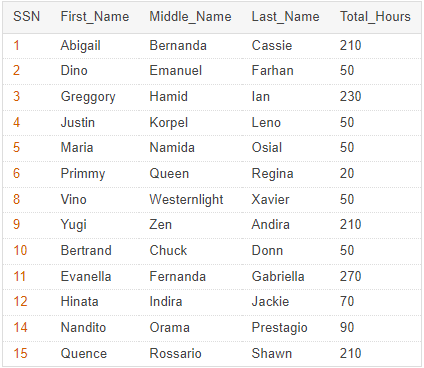
WHERE a.Department\_Number = b.Department\_Number

GROUP BY a.Department\_Number

ORDER BY Total\_Employee DESC;

1. Total hours perweek dari semua employee untuk setiap project ?

\* Asumsi : Total waktu adalah total hours per week



Gambar . Hasil Nomor 7

QUERY:

CREATE VIEW no\_7

AS SELECT a.SSN, First\_Name, Middle\_Name, Last\_Name,

SUM(Hours) AS Total\_Hours

FROM employee a, works\_on b

WHERE a.SSN = b.SSN

GROUP BY SSN;

1. Employee yang memiliki total hours perweek lebih besar dari 140 hours dan urutkan berdasarkan jumlah jam kerja terbanyak?



Gambar . Hasil Nomor 8

QUERY:

CREATE VIEW no\_8

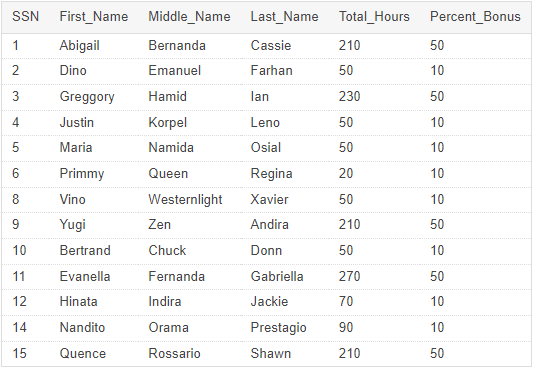
AS SELECT \*

FROM no\_7

WHERE Total\_Hours > 140

ORDER BY Total\_Hours DESC;

1. Kelompokkan bonus employee berdasarkan jumlah jam kerjanya ? (Jika >= 200 hours, maka bonus = 50%; Jika >= 150 hours, maka bonus = 25%, Selainnya bonus = 10%)



Gambar . Hasil Nomor 9

QUERY:

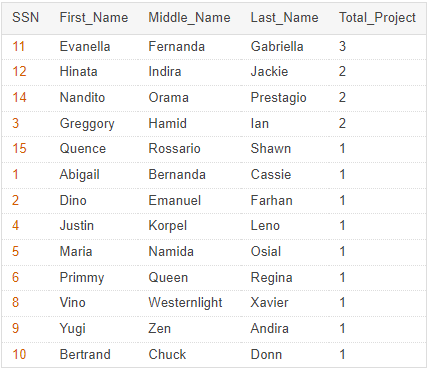
CREATE VIEW no\_9

AS SELECT SSN, First\_Name, Middle\_Name, Last\_Name, Total\_Hours,

IF(Total\_Hours >= 200, 50, IF(Total\_Hours >= 150, 25, 10)) AS Percent\_Bonus

FROM no\_7;

1. Banyaknya project yang dikerjakan tiap employee dan urutkan dari yang terbanyak ?



Gambar . Hasil Nomor 10

QUERY:

CREATE VIEW no\_10

AS SELECT a.SSN, First\_Name, Middle\_Name, Last\_Name,

COUNT(b.SSN) AS Total\_Project

FROM employee a, works\_on b

WHERE a.SSN = b.SSN

GROUP BY a.SSN

ORDER BY Total\_Project DESC;

1. Employee yang bekerja pada 4 project ?



Gambar . Hasil Nomor 11

QUERY:

CREATE VIEW no\_11

AS SELECT \*

FROM no\_10

WHERE Total\_Project = 4;

1. Employee yang memiliki rata-rata hours perweek = 70 jam dan bekerja pada 2 project ?



Gambar . Hasil Nomor 12

QUERY:

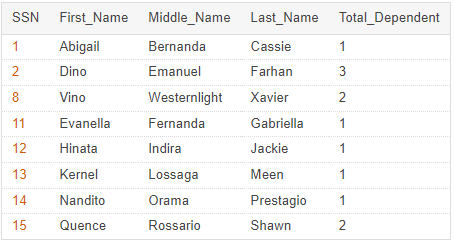
CREATE VIEW no\_12

AS SELECT a.SSN, a.First\_Name, a.Middle\_Name, a.Last\_Name, a.Total\_Hours, b.Total\_Project

FROM no\_7 a, no\_10 b

WHERE a.SSN = b.SSN AND a.Total\_Hours = 70 AND b.Total\_Project = 2;

1. Banyaknya Dependent berdasarkan relationship dengan employee ?



Gambar . Hasil Nomor 13

QUERY:

CREATE VIEW no\_13

AS SELECT a.SSN, First\_Name, Middle\_Name, Last\_Name,

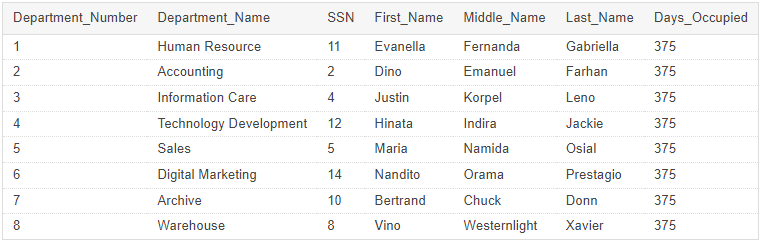
COUNT(b.SSN) AS Total\_Dependent

FROM employee a, dependent b

WHERE a.SSN = b.SSN

GROUP BY a.SSN;

1. Berapa lama Manager tiap Department sudah menjabat ?



Gambar . Hasil Nomor 14

QUERY:

CREATE VIEW no\_14

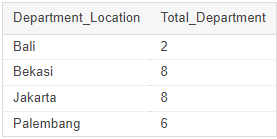
AS SELECT a.Department\_Number, Department\_Name, b.SSN, First\_Name, Middle\_Name, Last\_Name,

DATEDIFF(CURDATE(), Manager\_Startdate) AS Days\_Occupied

FROM department a, manager b, employee c

WHERE a.Department\_Number = b.Department\_Number AND b.SSN = c.SSN;

1. Lokasi project yang menjadi tempat lebih dari satu department?



Gambar . Hasil Nomor 15

QUERY:

CREATE VIEW no\_15a

AS SELECT Department\_Location,

COUNT(Department\_Location) AS Total\_Department

FROM dept\_location

GROUP BY Department\_Location;

CREATE VIEW no\_15b

AS SELECT \*

FROM no\_15a

WHERE Total\_Department > 1;