Pada sesi ini, diberikan tugas membuat query untuk mendefiniskan database (DDL) dan memanipulasi database (DML). Tugas dibagi menjadi 2 yakni tugas Requirement 1 dan tugas Requirement 2.

1. **Requirement 1**

Pertama dilakukan pembuatan database “db\_melodiindah” menggunakan query berikut:

-- Adminer 4.8.1 MySQL 5.5.5-10.4.24-MariaDB dump

SET NAMES utf8;

SET time\_zone = '+00:00';

SET foreign\_key\_checks = 0;

SET sql\_mode = 'NO\_AUTO\_VALUE\_ON\_ZERO';

SET NAMES utf8mb4;

DROP DATABASE IF EXISTS `db\_melodiindah`;

CREATE DATABASE `db\_melodiindah` /\*!40100 DEFAULT CHARACTER SET utf8mb4 \*/;

USE `db\_melodiindah`;

DROP VIEW IF EXISTS `instrumen\_musisi`;

CREATE TABLE `instrumen\_musisi` (`nama` varchar(64), `alamat` varchar(64), `nama\_instrumen` varchar(32), `kunci\_musik` varchar(8));

DROP VIEW IF EXISTS `pengarang\_album`;

CREATE TABLE `pengarang\_album` (`judul` varchar(64), `pengarang` varchar(64), `nama\_album` varchar(32), `tgl\_copyright` date, `format` enum('cd','mc'), `id\_album` int(11));

DROP TABLE IF EXISTS `tb\_album`;

CREATE TABLE `tb\_album` (

  `id\_album` int(11) NOT NULL AUTO\_INCREMENT,

  `nama\_album` varchar(32) NOT NULL,

  `tgl\_copyright` date NOT NULL,

  `format` enum('cd','mc') NOT NULL,

  `ssn` int(11) NOT NULL,

  PRIMARY KEY (`id\_album`),

  KEY `ssn` (`ssn`),

  CONSTRAINT `tb\_album\_ibfk\_1` FOREIGN KEY (`ssn`) REFERENCES `tb\_musisi` (`ssn`) ON DELETE NO ACTION

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8mb4;

TRUNCATE `tb\_album`;

INSERT INTO `tb\_album` (`id\_album`, `nama\_album`, `tgl\_copyright`, `format`, `ssn`) VALUES

(1, 'Humbug',   '2020-05-01',   'mc',   2),

(2, 'TBHC', '2000-03-05',   'cd',   2),

(3, 'Gravity',  '2007-08-29',   'cd',   4),

(4, 'ebook',    '2005-02-15',   'cd',   1),

(5, 'sample a', '2012-11-11',   'mc',   2);

DROP TABLE IF EXISTS `tb\_combine\_musik`;

CREATE TABLE `tb\_combine\_musik` (

  `id\_combine` int(11) NOT NULL AUTO\_INCREMENT,

  `ssn` int(11) NOT NULL,

  `id\_instrumen` int(11) NOT NULL,

  PRIMARY KEY (`id\_combine`),

  KEY `ssn` (`ssn`),

  KEY `id\_instrumen` (`id\_instrumen`),

  CONSTRAINT `tb\_combine\_musik\_ibfk\_1` FOREIGN KEY (`ssn`) REFERENCES `tb\_musisi` (`ssn`) ON DELETE NO ACTION,

  CONSTRAINT `tb\_combine\_musik\_ibfk\_2` FOREIGN KEY (`id\_instrumen`) REFERENCES `tb\_instrumen` (`id\_instrumen`) ON DELETE NO ACTION

) ENGINE=InnoDB AUTO\_INCREMENT=12 DEFAULT CHARSET=utf8mb4;

TRUNCATE `tb\_combine\_musik`;

INSERT INTO `tb\_combine\_musik` (`id\_combine`, `ssn`, `id\_instrumen`) VALUES

(1, 1,  1),

(2, 1,  2),

(3, 1,  3),

(4, 2,  1),

(5, 2,  4),

(6, 2,  7),

(7, 3,  8),

(8, 4,  1),

(9, 4,  2),

(10,    4,  3),

(11,    4,  5);

DROP TABLE IF EXISTS `tb\_instrumen`;

CREATE TABLE `tb\_instrumen` (

  `id\_instrumen` int(11) NOT NULL AUTO\_INCREMENT,

  `nama` varchar(32) NOT NULL,

  `kunci\_musik` varchar(8) NOT NULL,

  PRIMARY KEY (`id\_instrumen`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4;

TRUNCATE `tb\_instrumen`;

INSERT INTO `tb\_instrumen` (`id\_instrumen`, `nama`, `kunci\_musik`) VALUES

(1, 'gitar',    'c#'),

(2, 'gitar',    'f'),

(3, 'gitar',    'c'),

(4, 'gitar',    'g'),

(5, 'gitar',    'e'),

(6, 'keyboard', 'a'),

(7, 'keyboard', 'am'),

(8, 'keyboard', 'g'),

(9, 'keyboard', 'gm'),

(10,    'keyboard', 'ab');

DROP TABLE IF EXISTS `tb\_lagu`;

CREATE TABLE `tb\_lagu` (

  `id` int(11) NOT NULL AUTO\_INCREMENT,

  `judul` varchar(64) NOT NULL,

  `pengarang` varchar(64) NOT NULL,

  `id\_album` int(11) NOT NULL,

  PRIMARY KEY (`id`),

  KEY `id\_album` (`id\_album`),

  CONSTRAINT `tb\_lagu\_ibfk\_1` FOREIGN KEY (`id\_album`) REFERENCES `tb\_album` (`id\_album`) ON DELETE NO ACTION

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8mb4;

TRUNCATE `tb\_lagu`;

INSERT INTO `tb\_lagu` (`id`, `judul`, `pengarang`, `id\_album`) VALUES

(1, 'ultracheese',  'alex turner',  2),

(2, 'gravity',  'john mayer',   3),

(3, 'thinking fast and slow',   'd. kahneman',  4),

(4, 'cornerstone',  'alex turner',  1),

(5, 'secret door',  'alex turner',  1);

DROP TABLE IF EXISTS `tb\_musisi`;

CREATE TABLE `tb\_musisi` (

  `ssn` int(11) NOT NULL AUTO\_INCREMENT,

  `nama` varchar(64) NOT NULL,

  `alamat` varchar(64) NOT NULL,

  `nomor\_telepon` varchar(64) NOT NULL,

  PRIMARY KEY (`ssn`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8mb4;

TRUNCATE `tb\_musisi`;

INSERT INTO `tb\_musisi` (`ssn`, `nama`, `alamat`, `nomor\_telepon`) VALUES

(1, 'd. kahneman',  'USA',  '165728495051'),

(2, 'alex turner',  'USA',  '168290683231'),

(3, 'w. r. supratman',  'ID',   '6285294018827'),

(4, 'john mayer',   'USA',  '167286018266');

DROP TABLE IF EXISTS `instrumen\_musisi`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `instrumen\_musisi` AS select `tm`.`nama` AS `nama`,`tm`.`alamat` AS `alamat`,`ti`.`nama` AS `nama\_instrumen`,`ti`.`kunci\_musik` AS `kunci\_musik` from ((`tb\_combine\_musik` `tcm` join `tb\_instrumen` `ti` on(`ti`.`id\_instrumen` = `tcm`.`id\_instrumen`)) join `tb\_musisi` `tm` on(`tm`.`ssn` = `tcm`.`ssn`));

DROP TABLE IF EXISTS `pengarang\_album`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY DEFINER VIEW `pengarang\_album` AS select `tl`.`judul` AS `judul`,`tl`.`pengarang` AS `pengarang`,`ta`.`nama\_album` AS `nama\_album`,`ta`.`tgl\_copyright` AS `tgl\_copyright`,`ta`.`format` AS `format`,`ta`.`id\_album` AS `id\_album` from (`tb\_lagu` `tl` join `tb\_album` `ta` on(`ta`.`id\_album` = `tl`.`id\_album`));

-- 2022-06-21 05:52:37

Query tersebut akan membuat sebuah database bernama “db\_melodiindah”. Di dalamnya terdapat beberapa table dengan kolom sbb:

1. tb\_album:

* id\_album: int, primary key, sebagai ID untuk album sekaligus primary key dari table
* nama\_album: varchar(32), berisi nama dari album yang terdapat dalam table
* tgl\_copyright: date, berisi tanggal album resmi terdaftar hak ciptanya.
* format: enum, berisi format dari album yakni antara ‘cd’ atau ‘mc’
* ssn: int, foreign key, berisi nomor identifikasi produser dari album dan berelasi foreign key dengan table tb\_musisi.

1. tb\_instrumen:

* id\_instrumen: int, primary key, berisi nomor id dari tiap instrument yang terdaftar sekaligus sebagai primary key dari table.
* nama: varchar(32), berisi nama instrument yang terdaftar.
* kunci\_musik: varchar(32), berisi kunci musik yang terdaftar untuk tiap instrument.

1. tb\_lagu:

* id: int, primary key, sebagai primary key table dan identifier tiap lagu.
* judul: varchar(64), berisi nama/judul lagu.
* pengarang: varhar(64), berisi nama pengarang lagu.
* id\_album: int(11), foreign key, berisi id album yang berelasi dengan table tb\_album.

1. tb\_musisi:

* ssn: int, primary key, berisi identifier tiap musisi yang terdaftar sekaligus sebagai primary key table.
* nama: varchar(64), berisi nama tiap musisi.
* alamat: varchar(64), berisi alamat tiap musisi.
* nomor\_telepon: varchar(64), berisi nomor telepon tiap musisi.

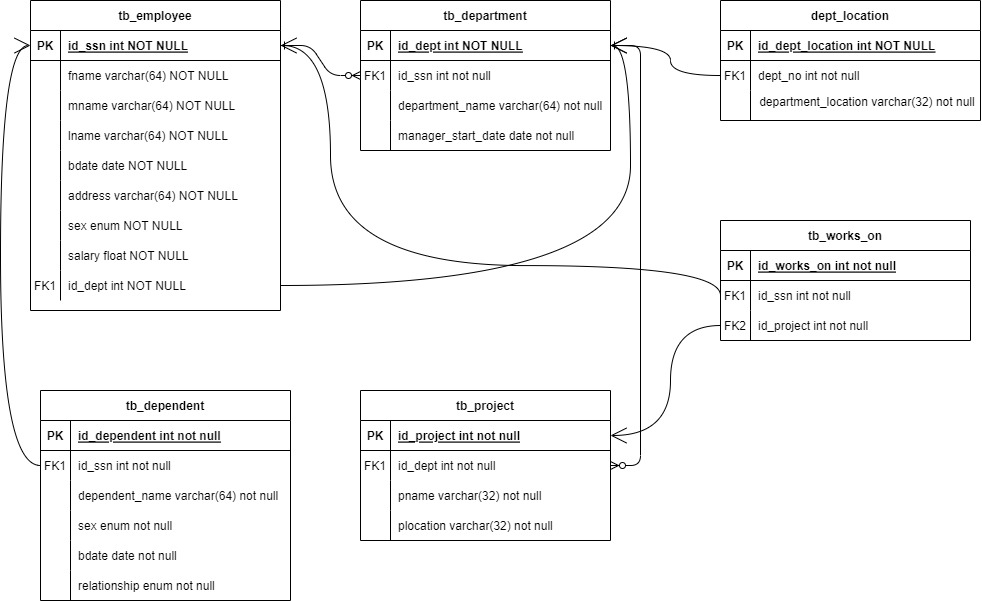
1. tb\_combine\_musik:

* ssn: int, foreign\_key, berisi ssn/identifier musisi yang berelasi dengan table tb\_musisi.
* id\_instrumen: int, foreign\_key, berisi id\_instrumen yang berelasi dengan table tb\_intrumen.

Selain itu, dibuat pula 2 view:

1. instrument\_musisi: berisi detail nama musisi, alamat musisi, serta instrumend dan kunci music dari tiap musisi untuk tiap instrument.
2. pengarang\_album: berisi detail judul lagu, pengarangnya, nama album, serta tanggal copyright dan format album untuk tiap lagu yang terdaftar.
3. **Requirement 2**

Pada tugas ini, dibuat sebuah database bernama “db\_company”. Berikut Entity Relationship Diagram (ERD) yang menunjukkan hubungan antar table dari database:



Database ini memiliki 6 tabel dengan detail sebagai berikut:

1. tb\_employee: table yang berisi detail employee

* id\_ssn: primary key dari table, sebagai identifier tiap employee.
* fname: nama depan dari employee.
* mname: nama tengah dari employee.
* lname: nama belakang dari employee.
* bdate: tanggal lahir dari employee.
* address: alamat dari employee.
* sex: jenis kelamin dari employee (male/female).
* salary: gaji per bulan dari employee.
* id\_dept: sebuah foreign key yang berelasi dengan table department. Menunjukkan departemen dari employee.

1. tb\_department: table yang berisi detail dari tiap department dalam company.

* id\_dept: primary key dan identifier dari tiap departemen dalam company.
* id\_ssn: foreign key yang berelasi dengan table employee. Menunjukkan identifier employee yang menjadi manajer departemen.
* manager\_start\_date: tanggal manajer departemen mulai bekerja sebagai manajer.

1. dept\_location: table yang berisi detil lokasi dari tiap departemen.

* id\_dept\_location: primary key dari table.
* dept\_no: foreign key yang berelasi id\_dept pada table departemen, menunjukkan identifier departemen.
* dept\_location: menunjukkan lokasi dari departemen.

1. tb\_project: berisi project-project yang sedang berjalan di dalam company.

* id\_project: primary key sekaligus identifier dari project di dalam company.
* id\_dept: foreign key yang berelasi dengan table departemen.
* project\_name: nama dari project.
* project\_location: lokasi dari project.
* hours: jam yang dikerjakan perminggu untuk tiap project.

1. tb\_dependent: berisi daftar pihak-pihak diluar employee yang berelasi dengan tiap employee.

* id\_dependent: primary key dari table, dan identifier dari tiap pihak dependent.
* id\_ssn: foreign key yang berelasi dengan table employee, menunjukkan dengan employee mana dependent berelasi.
* dependent\_name: nama pihak dependent.
* sex: jenis kelamin pihak dependent (male/female).
* bdate: tanggal lahir pihak dependent.
* relationship: hubungan pihak dependent dengan employee yang berelasi.

1. tb\_works\_on: berisi relasi antara project dengan employee.

* id\_works\_on: primary key dari table.
* id\_ssn: foreign key yang berelasi dengan table employee.
* id\_project: foreign key yang berelasi dengan table project.

Untuk mendefinisikan dan mengisi database, jalankan query berikut:

-- Adminer 4.8.1 MySQL 5.5.5-10.4.24-MariaDB dump

SET NAMES utf8;

SET time\_zone = '+00:00';

SET foreign\_key\_checks = 0;

SET sql\_mode = 'NO\_AUTO\_VALUE\_ON\_ZERO';

SET NAMES utf8mb4;

DROP DATABASE IF EXISTS `db\_company`;

CREATE DATABASE `db\_company` /\*!40100 DEFAULT CHARACTER SET utf8mb4 \*/;

USE `db\_company`;

DROP TABLE IF EXISTS `dept\_location`;

CREATE TABLE `dept\_location` (

  `id\_dept\_location` int(11) NOT NULL AUTO\_INCREMENT,

  `dept\_no` int(11) NOT NULL,

  `department\_location` varchar(32) NOT NULL,

  PRIMARY KEY (`id\_dept\_location`),

  KEY `dept\_no` (`dept\_no`),

  CONSTRAINT `dept\_location\_ibfk\_1` FOREIGN KEY (`dept\_no`) REFERENCES `tb\_department` (`id\_dept`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8mb4;

TRUNCATE `dept\_location`;

INSERT INTO `dept\_location` (`id\_dept\_location`, `dept\_no`, `department\_location`) VALUES

(1, 1,  '1st floor'),

(2, 2,  '2nd floor'),

(3, 3,  '2nd floor'),

(4, 4,  '2nd floor'),

(5, 5,  '11th floor');

DROP TABLE IF EXISTS `tb\_department`;

CREATE TABLE `tb\_department` (

  `id\_dept` int(11) NOT NULL AUTO\_INCREMENT,

  `id\_ssn` int(11) NOT NULL,

  `department\_name` varchar(64) NOT NULL,

  `manager\_start\_date` date NOT NULL,

  PRIMARY KEY (`id\_dept`),

  KEY `id\_ssn` (`id\_ssn`),

  CONSTRAINT `tb\_department\_ibfk\_2` FOREIGN KEY (`id\_ssn`) REFERENCES `tb\_employee` (`id\_ssn`) ON DELETE NO ACTION ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8mb4;

TRUNCATE `tb\_department`;

INSERT INTO `tb\_department` (`id\_dept`, `id\_ssn`, `department\_name`, `manager\_start\_date`) VALUES

(1, 5,  'ceo office',   '2014-09-16'),

(2, 2,  'board of commissioner',    '1956-08-11'),

(3, 4,  'treasury', '1513-09-17'),

(4, 3,  'legal',    '2021-05-24'),

(5, 1,  'engineering',  '1996-07-24');

DROP TABLE IF EXISTS `tb\_dependent`;

CREATE TABLE `tb\_dependent` (

  `id\_dependent` int(11) NOT NULL AUTO\_INCREMENT,

  `id\_ssn` int(11) NOT NULL,

  `dependent\_name` varchar(64) NOT NULL,

  `sex` enum('male','female') NOT NULL,

  `bdate` date NOT NULL,

  `relationship` enum('staff','lead','head') NOT NULL,

  PRIMARY KEY (`id\_dependent`),

  KEY `id\_ssn` (`id\_ssn`),

  CONSTRAINT `tb\_dependent\_ibfk\_1` FOREIGN KEY (`id\_ssn`) REFERENCES `tb\_employee` (`id\_ssn`) ON DELETE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=utf8mb4;

TRUNCATE `tb\_dependent`;

INSERT INTO `tb\_dependent` (`id\_dependent`, `id\_ssn`, `dependent\_name`, `sex`, `bdate`, `relationship`) VALUES

(1, 2,  'gatotkaca',    'male', '1509-11-11',   'staff'),

(2, 2,  'gundala',  'male', '1989-08-09',   'staff'),

(3, 1,  'guava',    'male', '1980-02-09',   'lead'),

(4, 3,  'big boss', 'male', '1903-07-01',   'lead'),

(5, 4,  'beluga whale', 'female',   '1967-12-01',   'head'),

(6, 1,  'rockstar', 'male', '1943-02-02',   'staff'),

(7, 2,  'rei',  'female',   '1999-09-09',   'lead');

DROP TABLE IF EXISTS `tb\_employee`;

CREATE TABLE `tb\_employee` (

  `id\_ssn` int(11) NOT NULL AUTO\_INCREMENT,

  `fname` varchar(64) NOT NULL,

  `mname` varchar(64) DEFAULT NULL,

  `lname` varchar(64) NOT NULL,

  `bdate` date NOT NULL,

  `address` varchar(32) NOT NULL,

  `sex` enum('male','female') NOT NULL,

  `salary` float NOT NULL,

  `id\_dept` int(11) NOT NULL,

  PRIMARY KEY (`id\_ssn`),

  KEY `id\_dept` (`id\_dept`),

  CONSTRAINT `tb\_employee\_ibfk\_1` FOREIGN KEY (`id\_dept`) REFERENCES `tb\_department` (`id\_dept`)

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8mb4;

TRUNCATE `tb\_employee`;

INSERT INTO `tb\_employee` (`id\_ssn`, `fname`, `mname`, `lname`, `bdate`, `address`, `sex`, `salary`, `id\_dept`) VALUES

(1, 'hafidz',   'firmansyah',   'ghufara',  '1997-11-29',   'solo', 'male', 10000000,   5),

(2, 'nadiem',   'a.',   'makarim',  '1984-07-04',   'jakarta',  'male', 12000000,   2),

(3, 'mahatma',  '', 'gandhi',   '1869-10-02',   'mumbai',   'male', 1000,   4),

(4, 'edward',   '', 'kenway',   '1693-03-10',   'swansea',  'male', 20000000,   3),

(5, 'cirilla',  'fionna',   'riannon',  '1251-12-25',   'nilfgaard',    'female',   25000000,   1),

(6, 'liquid',   'c.',   'snake',    '1967-05-11',   'europe',   'male', 7000000,    5),

(7, 'solid',    'c.',   'snake',    '1911-04-12',   'eur',  'male', 32000000,   1),

(8, 'solidus',  '', 'snake',    '1976-05-22',   'unknown',  'male', 12000000,   4);

DROP TABLE IF EXISTS `tb\_project`;

CREATE TABLE `tb\_project` (

  `id\_project` int(11) NOT NULL AUTO\_INCREMENT,

  `id\_dept` int(11) NOT NULL,

  `project\_name` varchar(64) NOT NULL,

  `project\_location` varchar(32) NOT NULL,

  PRIMARY KEY (`id\_project`),

  KEY `id\_dept` (`id\_dept`),

  CONSTRAINT `tb\_project\_ibfk\_1` FOREIGN KEY (`id\_dept`) REFERENCES `tb\_department` (`id\_dept`) ON DELETE NO ACTION

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8mb4;

TRUNCATE `tb\_project`;

INSERT INTO `tb\_project` (`id\_project`, `id\_dept`, `project\_name`, `project\_location`) VALUES

(1, 1,  'series-b funding', 'jakarta'),

(2, 2,  'digital transformation',   'jakarta'),

(3, 4,  'internal audit',   'solo'),

(4, 5,  'create login form',    'tangerang'),

(5, 5,  'create auth',  'jakarta'),

(6, 2,  'transparency', 'jakarta');

DROP TABLE IF EXISTS `tb\_works\_on`;

CREATE TABLE `tb\_works\_on` (

  `id\_works` int(11) NOT NULL AUTO\_INCREMENT,

  `id\_ssn` int(11) NOT NULL,

  `id\_project` int(11) NOT NULL,

  `hours` int(16) NOT NULL,

  PRIMARY KEY (`id\_works`),

  KEY `id\_ssn` (`id\_ssn`),

  KEY `id\_project` (`id\_project`),

  CONSTRAINT `tb\_works\_on\_ibfk\_1` FOREIGN KEY (`id\_ssn`) REFERENCES `tb\_employee` (`id\_ssn`),

  CONSTRAINT `tb\_works\_on\_ibfk\_2` FOREIGN KEY (`id\_project`) REFERENCES `tb\_project` (`id\_project`)

) ENGINE=InnoDB AUTO\_INCREMENT=15 DEFAULT CHARSET=utf8mb4;

TRUNCATE `tb\_works\_on`;

INSERT INTO `tb\_works\_on` (`id\_works`, `id\_ssn`, `id\_project`, `hours`) VALUES

(1, 1,  4,  40),

(2, 2,  2,  30),

(3, 3,  3,  15),

(4, 4,  3,  50),

(5, 5,  1,  100),

(6, 6,  5,  50),

(7, 6,  4,  100),

(8, 7,  3,  10),

(9, 7,  4,  170),

(10,    8,  1,  100),

(11,    1,  2,  35),

(12,    1,  3,  35),

(13,    1,  4,  50),

(14,    1,  5,  60);

-- 2022-06-21 10:32:21

1. **Requirement 2: Soal**

Kemudian, diberikan beberapa soal sebagai berikut:

**-- a) Tampilkan dependent\_name dan relationship dengan employee yang namanya diawali huruf R ?**

**select** \* **from** tb\_dependent td **where** dependent\_name **like** 'r%';

**-- b) Banyaknya employee yang mengerjakan project PNum = 1 ?**

**select** **count**(\*) **from** tb\_works\_on two **where** id\_project = 1;

**-- c) Banyaknya employee yang memiliki salary lebih dari 3500000 ?**

**select** \* **from** tb\_employee te **where** salary > 3500000;

**-- d) Banyaknya project yang dikerjakan DNum =2 ?**

**select** **count**(\*) **from** tb\_project tp **where** id\_dept = 2;

**-- e) Hitung total dan rata-rata salary dari setiap departemen ?**

**select** te.id\_dept, td.department\_name, **sum**(salary) total\_salary, **avg**(salary) average\_salary **from** tb\_employee te **join** tb\_department td **on** td.id\_dept = te.id\_dept **group** **by** te.id\_dept ;

**-- f) Banyaknya employee dari setiap department dan urutkan berdasarkan employee terbanyak ?**

**select** td.department\_name, te.id\_dept, **count**(\*) jumlah\_karyawan **from** tb\_employee te **join** tb\_department td **on** td.id\_dept = te.id\_dept **group** **by** id\_dept;

**-- g) Total hours perweek dari semua employee untuk setiap project ?**

**select** two.id\_project, **sum**(two.hours) hours\_total **from** tb\_works\_on two **join** tb\_employee te **on** te.id\_ssn = two.id\_ssn **group** **by** two.id\_project;

**-- h) Employee yang memiliki total hours perweek lebih besar dari 140 hours dan urutkan berdasarkan jumlah jam kerja terbanyak?**

**select** \* **from** (**select** te.id\_ssn, te.fname, te.lname, te.salary, **sum**(two.hours) hours\_total **from** tb\_works\_on two **join** tb\_employee te **on** te.id\_ssn = two.id\_ssn **group** **by** two.id\_ssn) **as** a **where** hours\_total > 140;

**-- i) Kelompokkan bonus employee berdasarkan jumlah jam kerjanya ?**

**-- (Jika >= 200 hours, maka bonus = 50%; Jika >= 150 hours, maka bonus = 25%, Selainnya bonus = 10%)**

**select** \*, **case**

**when** hours\_total >= 200 **then** 'Bonus 50%'

**when** hours\_total >= 150 **then** 'Bonus 25%'

**else** 'Bonus 10%'

**end** **as** bonus

**from** (

**select** te.id\_ssn, te.fname, te.lname, te.salary, **sum**(two.hours) hours\_total

**from** tb\_works\_on two

**join** tb\_employee te **on** te.id\_ssn = two.id\_ssn

**group** **by** two.id\_ssn) **as** a;

**-- j) Banyaknya project yang dikerjakan tiap employee dan urutkan dari yang terbanyak ?**

**select** id\_ssn, **count**(\*) **from** tb\_works\_on two **group** **by** id\_ssn **order** **by** 2 **desc**;

**-- k) Employee yang bekerja pada 4 project ?**

**select** \* **from** (**select** id\_ssn, **count**(\*) jumlah\_project **from** tb\_works\_on two **group** **by** id\_ssn **order** **by** 2 **desc**) **as** a **where** jumlah\_project > 4;

**-- l) Employee yang memiliki rata-rata hours perweek = 70 jam dan bekerja pada 2 project ?**

-- berdasarkan data tidak ada yang bekerja 70 jam perminggu,

-- namun berikut query untuk mencari yang bekerja 75 jam perminggu

**select** \*

**from** (

**select** te.id\_ssn,

te.fname, te.lname,

te.salary, **avg**(two.hours)

avg\_work\_hours, **count**(\*)

jumlah\_project

**from** tb\_works\_on two

**join** tb\_employee te **on** te.id\_ssn = two.id\_ssn

**group** **by** two.id\_ssn) **as** tbl1

**where** avg\_work\_hours = 75 **and** jumlah\_project = 2;

**-- m) Banyaknya Dependent berdasarkan relationship dengan employee ?**

**select** td.id\_ssn, te.fname, **count**(id\_dependent) jumlah\_dependent, td.relationship **from** tb\_dependent td **join** tb\_employee te **on** te.id\_ssn = td.id\_ssn **group** **by** relationship;

**-- n) Berapa lama Manager tiap Department sudah menjabat ?**

**select** \*, **datediff**(**curdate**(), manager\_start\_date) hari\_menjabat **from** tb\_department td;

**-- o) Lokasi project yang menjadi tempat lebih dari satu department?**

**select** \* **from** (**select** project\_location, **count**(\*) jml\_project **from** tb\_project tp **group** **by** project\_location) **as** tbl1 **where** jml\_project > 1;