1. Buatlah queryyang menampilkan nama lengkap (*full\_name*), tanggal lahir, jenis kelamin, dan nama departemen dari manajer perusahaan yang mulai bekerja pada tahun 1985!

**Query**

|  |
| --- |
| select concat(first\_name, ‘ ‘, last\_name) as full\_name, birth\_date, gender, D.dept\_name  from employees as E, dept\_manager as DM, departments as D  where E.emp\_no = DM.emp\_no and DM.dept\_no = D.dept\_no and E.hire\_date like "1985-%-%"; |

**Hasil Query (screenshot)**

|  |
| --- |
|  |

1. Buatlah query untuk menampilkan nama lengkap, title, dan tanggal bergabung pegawai yang bekerja di departemen Quality Management sejak bulan Juli 2002.

**Query**

|  |
| --- |
| select distinct concat(first\_name, ' ', last\_name) as full\_name, T.title, hire\_date  -> from employees as E, titles as T, dept\_emp as DE, departments as D  -> where E.emp\_no = DE.emp\_no and DE.dept\_no = D.dept\_no and dept\_name = "Quality Management"  -> and DE.from\_date like "2002-07-%"; |

**Hasil Query (screenshot)**

|  |
| --- |
|  |

1. Buatlah query untuk menampilkan 10 nama pegawai tertua dan tanggal lahirnya yang jenis kelaminnya tidak bernilai null.

**Query**

|  |
| --- |
| select concat(first\_name, ' ', last\_name), birth\_date from employees  -> where gender is not null order by birth\_date limit 10; |

**Hasil Query (screenshot)**

|  |
| --- |
|  |

1. Buatlah query yang menampilkan gender dan rata-rata gaji untuk setiap gender

**Query**

|  |
| --- |
| select distinct gender, sum(salary) / count(E.emp\_no) from employees E, salaries S  -> where E.emp\_no = S.emp\_no  -> group by gender; |

**Hasil Query (screenshot)**

|  |
| --- |
|  |

1. Buatlah query untuk memperlihatkan rata-rata gaji saat ini dari pegawai di departemen d009 berdasarkan nama belakangnya! (Hint: gunakan CURDATE() untuk mendapatkan tanggal hari ini)

**Query**

|  |
| --- |
| select last\_name, avg(salary) from employees E, salaries S  -> where to\_date in (select curdate()) and E.emp\_no in (select emp\_no from dept\_emp where dept\_no = 'd009')  -> group by last\_name; |

**Hasil Query (screenshot)**

|  |
| --- |
|  |

1. Buatlah query yang menampilkan nama departemen dan jumlah pegawai pada masing-masing departemen yang mulai menjabat di tahun 2000.

**Query**

|  |
| --- |
| select dept\_name, count(DE.emp\_no) from departments D, dept\_emp DE  -> where from\_date like '2000-%-%' and DE.dept\_no = D.dept\_no group by dept\_name; |

**Hasil Query (screenshot)**

|  |
| --- |
|  |

1. Buatlah query untuk mencari departemen-departemen yang rata-rata gaji manajernya di atas 50.000. Tampilkan nama departemen dan rata-rata gaji manajer pada departemen tersebut!

**Query**

|  |
| --- |
| select dept\_name from departments D, salaries S, employees E, dept\_manager DM  -> where D.dept\_no = DM.dept\_no and DM.emp\_no = E.emp\_no and E.emp\_no = S.emp\_no  -> group by dept\_name  -> having avg(salary) > 50000; |

**Hasil Query (screenshot)**

|  |
| --- |
|  |

1. Buatlah sebuah query yang menampilkan 10 baris pertama dari nomor karyawan, nama pertama, tanggal ulang tahun, gaji terbaru, dan diurutkan menurun berdasarkan gaji saat ini, kemudian menaik berdasarkan tanggal lahir!

**Query**

|  |
| --- |
| select E.emp\_no, E.first\_name, E.birth\_date, S.salary  -> from employees E, salaries S where E.emp\_no = S.emp\_no  -> order by salary desc, birth\_date limit 10; |

**Hasil Query (screenshot)**

|  |
| --- |
|  |

1. Buatlah query untuk menampilkan nama belakang pegawai non manajer yang memiliki nama belakang yang sama dengan manajer yang memiliki gaji di atas 90.000. CATATAN: Gunakan *set operation.*

**Query**

|  |
| --- |
| select last\_name from employees E, dept\_manager DM  -> where E.emp\_no not in (select emp\_no from dept\_manager DM)  -> intersect  -> select last\_name from employees E, dept\_manager DM  -> where E.emp\_no = DM.emp\_no and E.emp\_no in (select emp\_no from salaries where salary > 90000); |

**Hasil Query (screenshot)**

|  |
| --- |
|  |

1. Buatlah query untuk menampilkan 10 baris pertama dari semua data karyawan yang saat ini memiliki posisi sebagai engineer dan memiliki gaji kurang dari 50000

**Query**

|  |
| --- |
| select first\_name, last\_name from employees E, titles T, salaries S  -> where E.emp\_no = T.emp\_no and T.title = 'Engineer' and E.emp\_no = S.emp\_no and  -> S.salary < 50000 limit 10; |

**Hasil Query (screenshot)**

|  |
| --- |
|  |

1. Buatlah query untuk menampilkan semua employee dengan title manager yang membawahi departemen dengan nomor d009

**Query**

|  |
| --- |
| select \* from employees E where  -> E.emp\_no in (select emp\_no from dept\_manager DM where dept\_no like 'd009'); |

**Hasil Query (screenshot)**

|  |
| --- |
|  |

1. Buatlah query untuk memasukkan data baru ke tabel departemen dengan nama departemen Engineer dan nomor departemen d010

**Query**

|  |
| --- |
| insert into departments(dept\_name, dept\_no) values ('Engineer', 'd010'); |

**Hasil Query (screenshot)**

|  |
| --- |
|  |