**JAWABAN SISTEM BASIS DATA**

**Pengenalan Sistem Basis Data**

* Perintah SQL

1. SHOW DATABASES; → untuk melihat database yang tersedia
2. CREATE DATABASE coba; → membuat database dengan nama coba
3. DROP DATABASE coba; → menghapus database yang bernama coba
4. QUIT; atau \q; → keluar dari console MySQL

**KUNCI JAWABAN TIPE A**

1. **PT1**

* CREATE DATABASE PT1\_NIM;
* CREATE TABLE film (judul VARCHAR(30), tokohUtama VARCHAR(20), sutradara VARCHAR(25));
* ALTER TABLE film ADD episode VARCHAR(5);
* ALTER TABLE film MODIFY COLUMN episode INT(3);
* ALTER TABLE film DROP COLUMN tokohUtama;
* DESC film;

1. **PT2**

* CREATE DATABASE PT2\_NIM;
* CREATE TABLE hotel (

id\_hotel INT(11) PRIMARY KEY AUTO\_INCREMENT,

nama\_hotel VARCHAR(100) NOT NULL,

alamat VARCHAR(255) NOT NULL,

bintang ENUM('1', '2', '3', '4', '5') NOT NULL,);

CREATE TABLE pegawai (

id\_pegawai INT(11) PRIMARY KEY AUTO\_INCREMENT,

nama\_pegawai VARCHAR(100) NOT NULL,

usia INT(11) NOT NULL,

id\_hotel INT(11),

CONSTRAINT fk\_hotel FOREIGN KEY (id\_hotel) REFERENCES hotel(id\_hotel) ON UPDATE SET NULL ON DELETE SET NULL

);

* ALTER TABLE hotel

DROP FOREIGN KEY fk\_hotel;

ALTER TABLE pegawai

ADD CONSTRAINT fk\_hotel

FOREIGN KEY (id\_hotel)

REFERENCES hotel(id\_hotel)

ON UPDATE RESTRICT

ON DELETE RESTRICT;

* ALTER TABLE pegawai

ADD CONSTRAINT cek\_usia CHECK (usia < 20);

* desc hotel;

desc pegawai;

1. **PT3**

* Create database Pt3\_NIM;
* CREATE TABLE makanan (

idMakanan int auto\_increment not null primary key,

namaMakanan varchar(255) not null,

jenis enum("Vegetarian","Daging") not null,

harga decimal(7,2) not null,

bintang int not null);

* INSERT INTO makanan VALUES

("","Burger Ayam","Daging", 13000, "3"),

("","Takoyaki","Daging",15000,"4"),

("","Nasi Goreng","Vegetarian",15000,"1"),

("","Lotek","Vegetarian",9000,"2"),

("","Ramen","Daging",80000,"5");

* A. delete from makanan where harga < 13000;

B. delete from makanan where harga <= 20000 and bintang = "4";

* A. update makanan set namaMakanan = "Burger" where namaMakanan = "Burger Ayam";

B. update makanan set jenis = "Daging", bintang = "5" where namaMakanan = "Nasi Goreng";

1. **PT4**

* select concat(judul, ' disutradarai oleh ', sutradara) as 'Info Sutradara' from film;
* select \* from film where rating > 8.2 and kategori = "Comedy";
* select judul as "Film Terbaru", tanggalRilis from film where tanggalRilis > 20100101 order by tanggalRilis desc;
* select judul as 'Nama Film', rating as 'Rating' from film where rating between 8 and 9;
* select \* from film where kategori like "act%";

1. **PT5**

* SELECT idProduct AS "ID Produk", namaProduct AS "Nama Produk", price as "Harga" FROM `product` WHERE price <= 90000
* SELECT product.idProduct, product.namaProduct, idTransaksi, jumlah\_produk FROM product JOIN transaksi ON product.idProduct = transaksi.idProduct WHERE jumlah\_produk > 1;
* SELECT karyawan.idKaryawan, namaKaryawan, transaksi.idTransaksi, transaksi.jumlah\_produk, IF(jumlah\_produk>=3, "Dapat Gaji Bonus", "-") AS Bonus FROM `karyawan` JOIN transaksi ON karyawan.idKaryawan = transaksi.idKaryawan;
* SELECT karyawan.namaKaryawan, transaksi.idTransaksi, product.namaProduct as "Nama Produk", product.price AS Harga FROM transaksi JOIN karyawan on karyawan.idKaryawan = transaksi.idKaryawan JOIN product on product.idProduct = transaksi.idProduct;

1. **PT6**

* SELECT posts.Judul, COUNT(comments.UserID) AS "Total Comment" FROM posts LEFT JOIN comments ON posts.PostID = comments.PostID GROUP BY posts.Judul ORDER BY COUNT(comments.UserID);
* SELECT Users.Nama as "Sang Posting", GROUP\_CONCAT(Posts.Judul) as "Semua Postingan", posts.TanggalPosting as "Tanggal Postingan" FROM Users INNER JOIN Posts ON Users.UserID = Posts.UserID GROUP BY users.Nama ORDER BY posts.TanggalPosting DESC;
* SELECT users.nama, COUNT(posts.PostID) AS "Jumlah Postingan" FROM users LEFT JOIN posts ON users.UserID = posts.UserID WHERE users.UserID <> 3 GROUP BY users.Nama;
* SELECT users.Nama, COUNT(comments.CommentID) AS jumlah\_komentar FROM users JOIN comments ON users.UserID = comments.UserID GROUP BY users.Nama HAVING jumlah\_komentar < 3;

**KUNCI JAWABAN TIPE B**

1. **PT1**

* CREATE database PT1\_NIM;
* CREATE table buku (book\_id CHAR(4), title VARCHAR(30), pages int);
* ALTER TABLE buku ADD genre int(4);
* ALTER TABLE buku MODIFY COLUMN genre VARCHAR(20);
* ALTER TABLE buku DROP COLUMN pages;
* DESC buku;

1. **PT2**

* create database PT2\_NIM;
* create table supir (

id\_supir int(11) AUTO\_INCREMENT PRIMARY KEY,

jenis\_kelamin enum("L","P") NOT NULL,

usia int NOT NULL);

CREATE TABLE bus (

plat\_nomor varchar(8) PRIMARY KEY,

kursi int(2) NOT NULL,

warna varchar(50) NOT NULL,

id\_supir int(11) NULL,

CONSTRAINT fk\_supir

FOREIGN KEY (id\_supir)

REFERENCES supir(id\_supir)

ON UPDATE CASCADE

ON DELETE CASCADE

);

* alter table supir

add constraint check\_usia check (usia < 70);

* ALTER TABLE bus

DROP FOREIGN KEY fk\_supir;

alter table bus

add CONSTRAINT fk\_supir

FOREIGN KEY (id\_supir)

REFERENCES supir (id\_supir)

ON UPDATE SET NULL

ON DELETE SET NULL;

1. **PT3**

* create database PT3\_NIM
* create table film(

-> id\_film int auto\_increment not null primary key,

-> judul\_film varchar(100) not null,

-> genre varchar(50) not null,

-> rating float not null,

-> tanggal\_rilis date not null);

* insert into film

->values("","Oppenheimer","Romance",8,"2023-07-21"),

-> ("","Spiderman 2","Adventure",9,"2019-06-30"),

->("","I'm Thinking of Ending Things","Thriller",7.6,"2010-08-28"),

-> ("","As the Gd Will","Horror",7.4,"2014-11-15"),

->("","Nanti Kita Cerita Tentang Hari Ini","Romance",6.9,"2020-02-02");

* A. update film

-> set tanggal\_rilis = "2020-09-11", rating = 9.1 where genre = "Horror";

B. update film

-> set judul\_film = "Interstellar" where genre = "Adventure" and rating = 9;

* A. delete from film

-> where genre = "Romance" and rating = 8;

B. delete from film

-> where rating < 7;

1. **PT4**

* select concat(namaHotel, ' berada di Kota ', lokasi) as 'Info Hotel' from hotel;
* select \* from hotel where rating >= 4.5 and lokasi = "Surabaya";
* select namaHotel, concat("Rp. ", harga \* 3) as "Harga Tiga Malam" from hotel
* select namaHotel as 'Nama Hotel', kapasitasTamu as 'Kapasitas' from hotel order by kapasitasTamu asc
* select \* from hotel where lokasi like "sura%";

1. **PT5**

* SELECT pegawai.ID, pegawai.Nama AS 'Nama Pegawai', usia FROM pegawai WHERE usia>25;
* SELECT pegawai.id as 'ID Pegawai', pegawai.Nama as 'Nama Pegawai', logproyek.ID as 'ID LOG', logproyek.JamKerja FROM pegawai INNER JOIN logproyek ON pegawai.ID = logproyek.PegawaiID WHERE logproyek.JamKerja > 80;
* SELECT pegawai.id as 'ID Pegawai', pegawai.Nama, logproyek.ID as 'ID LOG', logproyek.JamKerja, IF(jamkerja >=100, 'Pelatihan Lanjutan', 'Pelatihan Dasar') AS 'Jenis Pelatihan' FROM pegawai INNER JOIN logproyek ON pegawai.ID = logproyek.PegawaiID;
* SELECT pegawai.Nama as 'Nama Pegawai', proyek.ID as 'ID Proyek', proyek.Nama as 'Nama Proyek', logproyek.JamKerja FROM logproyek INNER JOIN pegawai ON logproyek.PegawaiID = pegawai.ID INNER JOIN proyek ON logproyek.ProyekID = proyek.ID;

1. **PT6**

* Select akun.username, count(pertandingan.idAkun) as "Total Pertandingan" from pertandingan inner join akun on akun.idAkun = pertandingan.idAkun group by akun.username;
* select akun.username, akun.TanggalBergabung sum(pertandingan.poin) as total\_poin from pertandingan inner join akun on akun.idAkun = pertandingan.idAkun group by akun.username order by sum(pertandingan.poin);
* Select akun.username as "Nama Akun", Max(pertandingan.poin) as "Poin terbanyak", Min(pertandingan.poin) as "Poin terdikit" from pertandingan inner join akun on akun.idakun = pertandingan.idakun group by akun.username;
* SELECT karakter.namaKarakter, count(pertandingan.HasilPertandingan) AS total\_main FROM pertandingan join karakter on pertandingan.IdKarakter = karakter.IdKarakter GROUP BY karakter.namaKarakter HAVING total\_main >= 2;

**Dari kating**

1. Buatlah tabel dengan ketentuan sebagai berikut.

CREATE TABLE keahlian\_fasilitator (

-> id\_keahlian INT NOT NULL,

-> id\_fasilitator INT NOT NULL,

-> nama\_keahlian VARCHAR(50) NOT NULL,

-> level\_keahlian ENUM('Pemula','Mahir') NOT NULL,

-> PRIMARY KEY (id\_keahlian,id\_fasilitator),

-> CONSTRAINT fasilit

-> FOREIGN KEY (id\_fasilitator)

-> REFERENCES fasilitator(id\_fasilitator));

1. Isilah tabel yang telah dibuat dengan record di bawah ini

INSERT INTO keahlian\_fasilitator values

(1,1,'Teknologi','Mahir'),

(1,3,'Teknologi','Pemula'),

(2,1,'Entertainment','Pemula'),

(2,2,'Entertainment','Mahir'),

(2,4,'Entertainment','Pemula');

1. Ubahlah durasi\_konten dari konten yang namanya diakhiri kata Skill (1 poin)

update konten -> set durasi\_konten = 24

where nama\_konten like '%Skill';

1. Hapus record yang masuk ke dalam kategori teknologi dan memiliki durasi konten 12 menit (1 poin)

delete from konten

where kategori\_konten = 'Teknologi' and durasi\_konten = 12;

1. Tampilkan nama\_konten, kategori\_konten, dan durasi konten sesuai dengan contoh berikut. Perhatikan nama kolom (1 poin)

select nama\_konten as 'Judul', durasi\_konten as 'Durasi', kategori\_konten as 'Kategori' from konten;

1. Tampilkan konten yang diupload sebelum tanggal 01-07-2022

select konten.nama\_konten, konten.tgl\_upload from konten where konten.tgl\_upload < '2022-07-01';

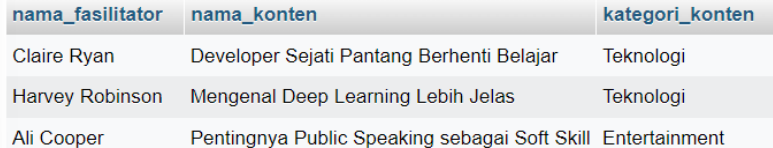


SELECT konten.nama\_konten, konten.durasi\_konten, konten.kategori\_konten, fasilitator.nama\_fasilitator from konten INNER JOIN fasilitator on konten.id\_fasilitator = fasilitator.id\_fasilitator;

1. SELECT DISTINCT fasilitator.nama\_fasilitator from fasilitator order by fasilitator.nama\_fasilitator desc;
2. A screenshot of a phone

   Description automatically generatedSELECT fasilitator.nama\_fasilitator as 'Nama Fasilitator', COUNT(konten.id\_konten) as 'Total Konten' from fasilitator -> INNER JOIN konten on fasilitator.id\_fasilitator = konten.id\_fasilitator -> group by fasilitator.nama\_fasilitator -> HAVING COUNT(konten.id\_konten) >=2;
3. Select konten.kategori\_konten, MAX(konten.tgl\_upload) as 'tgl\_upload terbaru' from konten group by konten.kategori\_konten;
4. Tampilkan nama fasilitator, nama konten, dan kategori konten untuk konten yang durasinya lebih dari 12 menit dan tanggal upload di antara tanggal 1 April 2022 dan 1 September 2022.

A screenshot of a computer

Description automatically generated

select fasilitator.nama\_fasilitator, konten.nama\_konten, konten.kategori\_konten from konten INNER JOIN fasilitator on konten.id\_fasilitator fasilitator.id\_fasilitator where konten.durasi\_konten > 12 and (konten.tgl\_upload between '2022-04-01' and '2022-09-01');