

IF2140 - Pemodelan Basis Data Milestone 3
Ketergantungan Fungsional dan Normalisasi



Kelompok 11

18221042 Ghaylan M. Fatih

18221058 Marvel Subekti

18221100 Vincent Franstyo

18221122 Reswara Trista Aulia C

Sistem dan Teknologi Informasi

Sekolah Teknik Elektro dan

Informatika

Institut Teknologi Bandung

2022

Deskripsi Sistem:

Perpustakaan merupakan tempat yang berfungsi untuk melakukan penyimpanan berbagai jenis media cetak yang diatur dan disusun sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pembaca (Sutarno NS, 2006:11). Perpustakaan juga sering digunakan oleh masyarakat luas untuk ‘berselanjar’ menambah ilmu pengetahuan melalui buku secara gratis. Perpustakaan itu sendiri memiliki sebuah layanan utama yaitu peminjaman buku. Oleh sebab itu, tidak heran apabila perpustakaan merupakan salah satu fasilitas umum yang sangat bermanfaat bagi setiap lapisan masyarakat.

Kami sebagai masyarakat yang sering menggunakan fasilitas umum ini, sangat tertarik untuk meninjau bagaimana sistem dan *workflow* dari perpustakaan khususnya perpustakaan nasional Indonesia dimana banyak sekali data buku serta peminjam yang harus disimpan dan tidak boleh *redundant* karena akan menyebabkan suatu permasalahan di kemudian hari. Selain itu, sistem peminjaman buku yang dibentuk haruslah memiliki data yang konsisten serta lengkap agar ketersediaan dan keberadaan buku bisa terus di *track* dan bisa dimanfaatkan kembali oleh peminjam-peminjam selanjutnya. Oleh karena itu data-data yang masif harus diintegrasikan kedalam suatu sistem basis data yang akan kami analisis dan tinjau serta akan direpresentasikan dalam bentuk *Entity Relationship Diagram*. Basis data yang akan kami analisis mencakup beberapa entitas seperti data akun, kartu perpustakaan, buku, peminjaman buku, kategori buku, reservasi buku, pertemanan, status akun, sinopsis buku, *review* buku, catatan pribadi, serta data riwayat peminjaman buku. Berikut penjelasannya lebih lanjut.

Untuk setiap pendataan suatu entitas peminjaman buku, akan disimpan satu data buku yang terdiri dari ID Buku yang unik untuk setiap unit buku, *file size* dari *soft copy* buku, jumlah buku yang merupakan jumlah total dari satu judul buku yang ada di dalam *database*, judul buku, penulis, penerbit, ISBN, e-ISBN, tahun rilis, penilaian, dan sinopsis dari buku tersebut.

Selain data buku, akan disimpan juga data dari akun *user* yang meminjam buku yang terdiri atas ID akun yang unik, *username* dari akun, *e-mail*, *password*, nomor telepon dimana satu akun dapat memiliki lebih dari satu nomor telepon, biodata dari setiap akun, serta status akun yang terdiri dari 2 jenis status yaitu *bookworm* dan *socializer*. Setiap akun pasti memiliki *bookworm* atau *socializer*.

Kedua status tersebut akan dipilih oleh pengguna saat pertama kali membuat akun Ipusnas, dimana perbedaannya terletak pada atribut yang dimiliki kedua status tersebut. Untuk status *bookworm* maka setiap pengguna bisa memilih avatar dengan tipe tumbuhan, data dari status *bookworm* ini akan dicatat pada entitas Status_BookWorm, sementara untuk status *socializer* maka avatar yang dapat dipilih adalah avatar yang bertipe hewan dan data-nya akan dicatat pada entitas Status_Socializer. Selain itu perbedaan dari keduanya adalah, untuk status *bookworm*, maka nanti akan ada fitur tambahan yaitu fitur rekomendasi buku. Sementara untuk status *socializer*, maka nanti akan ada fitur tambahan yaitu fitur rekomendasi teman. Perlu dicatat bahwa satu akun hanya mampu memiliki satu status saja.

Pada pendataan entitas peminjaman buku juga akan disimpan data peminjaman dengan isi entitas berupa ID peminjaman yang unik, tanggal data peminjaman dibuat, dan juga tanggal buku harus dikembalikan.

Dalam entitas peminjaman buku, tidak semua buku dipinjam dan satu buku hanya dapat dilakukan satu peminjaman oleh satu akun dalam satu waktu tertentu. Setiap peminjaman pasti memiliki buku yang dipinjam dan setiap buku dapat dilakukan banyak peminjam sesuai dengan ketersediaan buku.

Pada setiap entitas kategori buku, akan disimpan data dari buku dan jenis kategori buku yang ada di *database*, dengan isi entitas berupa ID kategori yang unik dan nama kategorinya. Dimana seluruh buku yang ada didalam *database* pasti memiliki kategori, dan bisa memiliki lebih dari satu kategori. Selain itu, setiap kategori buku pasti memiliki setidaknya satu buku yang termasuk dalam kategori tersebut.

Pada entitas *reservation* akan dicatat data mengenai ID reservasi, tanggal reservasi dilakukan, dan status reservasi. Dalam *database* ini 1 reservasi akan digunakan untuk 1 *copy* buku dengan setiap reservasi harus memiliki 1 buku dan buku tidak harus memiliki reservasi tetapi satu buku bisa memiliki banyak reservasi. Kemudian Tidak semua akun harus melakukan reservasi buku atau bisa melakukan banyak reservasi tetapi setiap reservasi harus tepat berelasi dengan 1 akun.

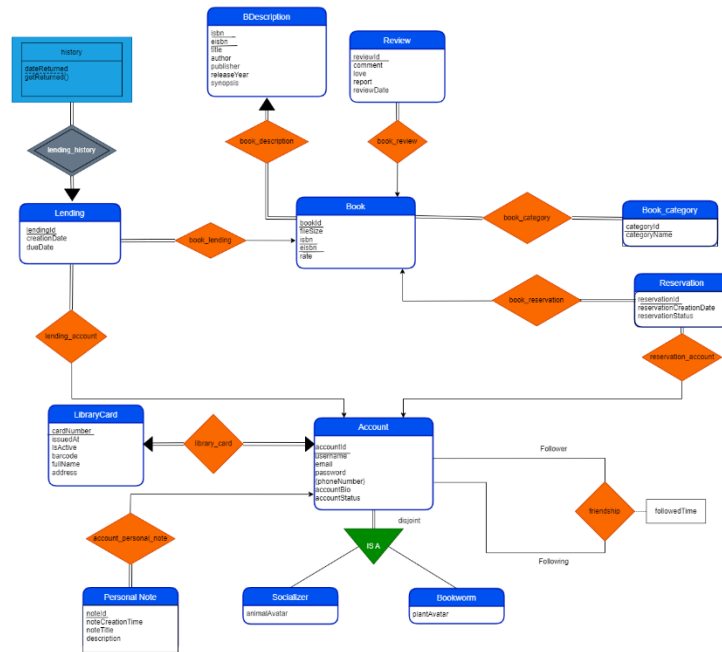
Setiap peminjaman akan tercatat dalam entitas *history*. Entitas *history* sendiri merupakan *weak entity* dari entitas peminjaman. Setiap peminjaman yang sudah berlalu akan dicatatkan status pengembalian dan tanggal pengembaliannya serta tanggal dipinjamnya buku tersebut yang mengacu kepada atribut *creationDate* pada entitas *lending* serta berfungsi sebagai acuan untuk atribut *getReturned*. Pencatatan peminjaman ini melibatkan entitas *lending* karena berisi buku yang sedang dipinjam serta berelasi secara tidak langsung dengan entitas *book* yang berisi informasi buku yang dipinjam dalam peminjaman tersebut serta *account* yang melakukan peminjaman buku tersebut melalui entitas peminjaman.

Setiap buku memiliki kesempatan untuk di-*review* oleh pengguna, namun tidak semua buku memiliki *review*. Satu buku juga memiliki kesempatan untuk memiliki banyak *review*. Setiap *review* ini memiliki ID *review*, jumlah komentar, jumlah *love*, jumlah *report*, dan waktu *review* tersebut diberikan.

Kemudian fitur lain yang bisa dipakai oleh setiap pengguna akun iPusnas adalah *personal note*. *Personal note* ini nantinya akan berfungsi untuk memberikan kemudahan bagi pengguna agar mampu mencatat hal-hal penting yang dirasa perlu dicatat oleh mereka. Oleh sebab itu, data yang akan disimpan dalam hal ini adalah *noteID*, judul *note*, deskripsi / teks *note*, dan waktu dibuatnya *note* tersebut. Perlu diperhatikan bahwa *personal note* bersifat opsional sehingga tidak semua akun perlu memiliki *personal note* tetapi setiap *note* yang dibuat oleh pengguna pasti dimiliki oleh suatu akun tertentu.

Selanjutnya, ada *library card* yang berfungsi sebagai kartu identitas pengguna fasilitas perpustakaan. *Library card* ini nanti akan dipakai setiap kali pengguna ingin meminjam buku fisik di perpustakaan. Oleh karena, itu di dalam *library card* ini dibutuhkan data pribadi pengguna, diantaranya adalah nama lengkap, alamat, nomor kartu, nomor *barcode*, keterangan aktif atau tidaknya kartu perpustakaan, serta tanggal terbentuknya kartu tersebut (*issuedAt*). Perlu dicatat bahwa pembuatan kartu perpustakaan dilakukan pada tahap awal registrasi akun sehingga setiap pengguna pasti memiliki sebuah kartu perpustakaan untuk dirinya sendiri dan setiap kartu perpustakaan pasti dimiliki oleh seorang pengguna.

Kemudian yang terakhir adalah relasi pertemanan memungkinkan setiap pengguna mampu melakukan pertemanan dengan teman-temannya. Untuk setiap pertemanan akan dicatat setiap waktu pertemanan (*follow*) dilakukan. Tetapi, hal ini bersifat opsional, artinya seorang pengguna bisa saja tidak melakukan pertemanan dengan pengguna lainnya tetapi untuk satu pengguna bisa melakukan banyak pertemanan dengan *user* lain.



gambar 1.1. Entity Relationship Diagram iPusnas

Daftar *functional Dependencies*, *Candidate Key*, dan bentuk normal setiap relasi:

Relasi Lending

<u>lendingId</u>	dueDate	bookId	accountId
1	2020-10-17	ID001	acc001

Lending = (lendingId, creationDate, dueDate, bookId, accountId)

Functional Dependencies

{lendingId → creationDate, dueDate, bookId, accountId}

Candidate Key: lendingId

Normal Form: BCNF

Relasi *Lending* berada pada bentuk normal BCNF karena seluruh atribut pada LHS (*Left Hand Side*) merupakan *superkey*. Dalam hal ini, *Candidate Key* terdefinisi, *lendingId*, merupakan *superkey* yang paling minimal yang dapat digunakan pada relasi *lending* untuk ketergantungan fungsionalnya.

Relasi Reservation

reservationId	reservationCreatedDate	reservationStatus	<u>bookId</u>	<u>accountId</u>
1	2020-10-3	Ready	ID001	Acc051

Reservation = (reservationId, reservationCreationDate, reservationStatus, bookId, accountId)

Functional Dependencies: {reservationId → reservationCreatedDate, reservationStatus, bookId, accountId}

Candidate Key: reservationId

Normal Form:

Relasi Reservation merupakan relasi yang berada pada bentuk normal BCNF karena pada ketergantungan fungsional yang ada, seluruh LHS (*Left Hand Side*)-nya telah merupakan *superkey*. Pada relasi ini, *Candidate Key* yang digunakan adalah reservationId karena itu merupakan *superkey* paling minimal yang diperlukan untuk membuat ketergantungan fungsional pada relasi ini.

Relasi Review

reviewId	Comment	Love	report	bookId
acn5222	98	108	0	ID228

Review = (reviewId, comment, love, report, bookId)

Functional Dependencies:

{reviewId → comment, love, report, bookId
reviewId, bookId → comment, love, report}

Candidate Key: reviewId

Normal Form: BCNF

Relasi Review telah mencapai bentuk normal BCNF karena pada ketergantungan fungsional yang ada, seluruh LHS (*Left Hand Side*)-nya telah merupakan *superkey*. Pada relasi ini, *Candidate Key* yang digunakan adalah reviewId karena itu merupakan *superkey* paling minimal yang diperlukan untuk membuat ketergantungan fungsional pada relasi ini.

Relasi History

History = (lendingId, dateReturned, getReturned)

lendingId	dateReturned	getReturned
1	2020-10-13	TRUE

Functional Dependencies:

{lendingId → dateReturned, getReturned}

Candidate Key: lendingId

Normal Form: BCNF

Relasi History telah mencapai bentuk normal BCNF karena pada ketergantungan fungsional yang ada, seluruh LHS (*Left Hand Side*)-nya telah merupakan *superkey*. Pada relasi ini, *candidate key* yang digunakan adalah lendingId yang merupakan *superkey* paling minimal yang diperlukan untuk menghasilkan ketergantungan fungsional yang ada.

Relasi Book

bookId	fileSize	title	author	publisher	isbn	eisbn	releaseYear	ra	synopsis
--------	----------	-------	--------	-----------	------	-------	-------------	----	----------

								t e	
ID001	49	Monk and the Fish, The (Le moine et le poisson)	Chas tity Rose nshi ne	Stehr Inc	3412 3470 3-5	00530 04 15-9	2004	3	Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet. Maecenas leo odio, condimentum id, luctus nec, molestie sed, justo. Pellentesque viverra pede ac diam.

Book = (bookId, fileSize, title, author, publisher, isbn, eisbn, releaseYear, rate, synopsis)

Functional Dependencies:

bookId → fileSize, isbn, eisbn, rate

isbn, eisbn → author, publisher, title, releaseYear, synopsis

Candidate Key: bookId

Normal Form: 2NF

Relasi Book merupakan relasi yang berada pada bentuk normal 2NF karena seluruh atribut *non primary key* bergantung total pada atribut *primary key*. Lalu, FD yang menyebabkan relasi *Book* tidak BCNF adalah FD2. Maka dari itu, kita perlu membuat relasi baru agar mencapai *Normal Form* yang lebih tinggi.

Maka dari itu, dapat dilakukan dekomposisi menjadi dua relasi baru, yaitu:

Relasi Book

Book = (bookId, fileSize, isbn, eisbn, rate)

Functional Dependency:

{bookId → fileSize, isbn, eisbn, rate}

bookId	fileSize	isbn	eisbn	rate
ID001	49	341234703-5	0053004 15-9	3

Relasi BDescription

(isbn, eisbn, title, author, publisher, releaseYear, synopsis)

Functional Dependency:

{isbn, eisbn → author, publisher, title, releaseYear, synopsis}

isbn	eisbn	author	title	publisher	ReleaseYear	synopsis
------	-------	--------	-------	-----------	-------------	----------

341234703-5	005300415-9	Stephen Piwell	Monk and the Fish, The (Le moine et le poisson)	Stehr Inc	2004	Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet. Maecenas leo odio, condimentum id, luctus nec, molestie sed, justo.
-------------	-------------	----------------	---	-----------	------	--

Kedua relasi baru tersebut sudah termasuk dalam bentuk BCNF karena LHS (*Left Hand Side*) dari setiap relasi merupakan *superkey*.

Relasi Category

<u>categoryId</u>	categoryName
CAT001	Fiction
CAT002	Non-Fiction

Category = (categoryId, categoryName)

Functional Dependency:
{categoryId → categoryName}

Candidate Key: categoryId

Normal Form: BCNF

Relasi *Category* sudah berada pada bentuk BCNF karena setiap LHS (*Left Hand Side*) merupakan *superkey*. Dalam hal ini, *Candidate Key* yang terdefinisi adalah *categoryId* karena *categoryId* mendefinisikan atribut *categoryName*.

Relasi Book Category

<u>bookId</u>	<u>categoryId</u>
ID023	CAT001

Functional Dependency:
{categoryId, bookId → bookId, categoryId}

Candidate Key: categoryId, bookId

Normal Form: BCNF

Relasi *BookCategory* sudah berada pada bentuk BCNF karena atribut di LHS (*Left Hand Side*) merupakan *superkey*. Dalam hal ini, *categoryId* dan *bookId* bersifat trivial.

Relasi Account

accountId	username	email	password	accountBio	cardNumber
acc001	lattenborough0	bdecleen0@va.gov	Q4HXIX	Duis bibendum, felis sed interdum venenatis, turpi...	89931080721

Account = (accountId, username, email, password, accountBio, cardNumber)

Functional Dependencies:

{accountId → accountId, username, email, password, accountBio, cardNumber}

{email → accountId, username, email, password, accountBio, cardNumber}

{cardNumber → accountId, username, email, password, accountBio, cardNumber}

Candidate Key: accountId, email

Normal Form: BCNF

Relasi *Account* berada pada bentuk BCNF karena setiap *LHS (Left Hand Side)* pada *Functional Dependency* yang terdefinisi merupakan sebuah *superkey*. Dalam hal ini *Candidate Key* yang terdefinisi ada dua yaitu *accountId* dan *email*. Hal ini dikarenakan kedua atribut bisa mendefinisikan semua atribut lainnya didalam relasi *Account* akibat nilai yang unik didalam atribut ini.

Relasi Friendship

accountId	followerId	followedTime
acc001	acc283	2022-08-29

Friendship = (accountId, followerId, followedTime)

Functional Dependency:

{accountId, followerId → accountId, followerId, followedTime}

Candidate Key: (accountId, followerId)

Normal Form: BCNF

Relasi *friendship* berada pada bentuk normal BCNF karena *LHS (Left Hand Side)* pada *Functional Dependency* yang terdefinisi merupakan sebuah *superkey*. Dalam relasi ini, *accountId* dan *followerId* merupakan sebuah *key* yang *composite* dimana kedua pasangan atribut ini akan mengidentifikasi atribut *followedTime* apabila mereka secara bersama-sama digabung membentuk suatu *superkey*. Hal ini dikarenakan kedua atribut tersebut bernilai tidak unik, sehingga perlu digabung supaya nilainya menjadi unik

Relasi Account_PhoneNumber

accountId	phoneNumber
acc001	6208077618936
acc001	6289772367350

Account_PhoneNumber: (accountId, phoneNumber)

Functional Dependency:
{phoneNumber → accountId}

Candidate Key: phoneNumber

Normal Form: BCNF

Relasi *Account_PhoneNumber* berada pada bentuk normal BCNF, hal dikarenakan *LHS (Left Hand Side)* pada *Functional Dependency* yang terdefinisi merupakan sebuah *superkey*. Hal ini terlihat dari tabel diatas bahwa tabel pada relasi *Account* menunjukan nilai *phoneNumber* yang *unique* dan mampu mendefinisikan *accountId* yang berperan sebagai *foreign key*.

Relasi PersonalNote

noteId	noteTitle	description	noteCreationTime	accountId
0J94	maecenas ut massa quis augue	Integer ac leo. Pellentesque ultrices mattis odio...	2021-08-25	acc128

PersonalNote = (noteId, noteTitle, description, noteCreationTime, accountId)

Functional Dependencies:
{noteId → noteTitle, description, noteCreationTime, accountId}

Candidate Key: noteId

Normal Form: BCNF

Relasi *PersonalNote* berada pada bentuk normal BCNF karena *LHS (Left Hand Side)* pada *Functional Dependency* yang terdefinisi merupakan sebuah *superkey*. Hal ini dikarenakan *noteId* merupakan satu satunya *superkey* yang mampu merepresentasikan keseluruhan atribut didalam relasi *PersonalNote*.

Relasi Status_BookWorm

accountId	plantAvatar
acc001	Strigula Lichen

Status_BookWorm = (accountId, plantAvatar)

Functional Dependencies:
{accountId → plantAvatar}

Candidate Key: accountId

Normal Form: BCNF

Relasi *Status_BookWorm* berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung kepada *superkey*, yaitu atribut *accountId*.

Relasi Status_Socializer

accountId	animalAvatar
acc151	Sloth, Hoffman's

Status_Socializer = (accountId, animalAvatar)

Functional Dependencies:

{accountId → animalAvatar}

Candidate Key: accountId

Normal Form: BCNF

Relasi *Status_Socializer* berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung kepada *superkey*, yaitu atribut *accountId*.

Relasi LibraryCard

cardNumber	issuedAt	barcode	fullName	address
10166086739	2021-11-22	59315087340	Chester Sonnenschein	0 Hoffman Hill

LibraryCard = (cardNumber, issuedAt, barcode, fullName, address)

Functional Dependency:

{cardNumber → issuedAt, barcode, fullName, address
barcode → issuedAt, cardNumber, fullName, address}

Candidate Key: cardNumber, Barcode

Normal Form: BCNF

Relasi *LibraryCard* berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung kepada *superkey*, yaitu atribut *cardNumber* atau atribut *barcode*.

Daftar informasi yang bisa didapatkan dari basis data yang telah dibuat

Informasi	Daftar nama peminjam buku yang belum mengembalikan bukunya sesuai tenggat waktu pengembalian buku yang dijanjikan
Query	select fullName from LibraryCard natural join account natural join lending where lendingId in ((select lendingId from History) intersect (select lendingId from History where getReturned = "FALSE"));
Hasil Query	<pre> MariaDB [perpusnas]> select fullName from LibraryCard natural join account natural join lending where lendingid in ((select lendingid from History) intersect (select lendingid from History where getReturned = "FALSE")); +-----+ fullName +-----+ Gal Mussard Korrie Quinney Mufinella Maxillius Darell O'Sherlin Selene Lorraine Rudiger Rickis Tommy Denness Wisse Akitt Joellen Senescall Sig Hawford Nev Manson Lisa Verheijden Stanly Huckster Ashlie Cardiff Juliana Pomfrett Chaddie Gronaver Stanford Messier Maurie Shewring Heinrich Dureden Vank Minty Christiana Boylin Dav Harvison Sandi Desquesnes Benni Goldbourn Ashley Suttie Sherilyn Stede Salome Poyres +-----+ 27 rows in set (0.016 sec) </pre>
Tabel Yang Terlibat	Account, Account_PhoneNumber, LibraryCard, History

Informasi	Daftar judul buku abad 20 yang disukai oleh lebih dari 100 pembaca
Query	(Select title from book where releaseYear between 1900 and 2000) intersect (select title from book where bookId in (select bookId from review where love > 100));
Hasil Query	<pre> MariaDB [perpusnas]> (select title from book where releaseYear between 1900 and 2000) intersect (select title from book where bookId in (select bookId from review where love > 100)); +-----+ title +-----+ Promise, The (Versprechen, Das) Tarzan's Greatest Adventure Malibu High Down in the Delta Rollercoaster Hawking Stone of Destiny David Cross: Let America Laugh À nous la liberté (Freedom for Us) Muriel's Wedding Enemy Simple Simon (I rymden finns inga känslor) Cat's Meow, The Vito Dr. Ehrlich's Magic Bullet Cocoon: The Return Våreitå Servant, The Claire of the Moon Get Well Soon Claymation Christmas Celebration, A Happy Family Into the Woods Perfect Stranger Funny Games U.S. It's a Great Feeling Crime on the Highway Lady and the Tramp II: Scamp's Adventure The Stone Council Paper Heart Objective, Burma! Cook the Thief His Wife & Her Lover, The Chalte Chalte Cinderella Messenger, The With a Friend Like Harry... (Harry, un ami qui vous veut du bien) Hell Ride Ferris Bueller's Day Off +-----+ </pre>

Tabel Yang Terlibat	Book, Review
---------------------	--------------

Informasi	Daftar Jenis buku dan jumlah bukunya
Query	Select categoryName, count(bookId) as "Jumlah_Buku" from Category natural join Book natural join BookCategory group by categoryName order by Jumlah_Buku asc;
Hasil Query	<pre> MariaDB [perpusnas]> Select categoryName , COUNT(bookId) as "Jumlah_Buku" from Category natural join Book natural join BookCategory group by categoryName order by Jumlah_Buku asc ; +-----+ categoryName Jumlah_Buku +-----+ Thriller 7 Drama 9 Legend 9 Fable 10 Reference Book 12 Horror 12 Biography 12 Comic 12 Fiction 12 Non fiction 13 Mythology 13 Mystery 14 Textbook 15 +-----+ 13 rows in set (0.002 sec) </pre>
Tabel Yang Terlibat	Book, Category, Book Category

Informasi	Avatar tumbuhan yang digunakan lebih dari 1 pengguna
Query	Select plantAvatar, count(accountId) as "Jumlah_Pemakai" from status_bookworm group by plantAvatar Having Jumlah_Pemakai > 1 order by Jumlah_Pemakai asc
Hasil Query	<pre> MariaDB [perpusnas]> select plantAvatar , COUNT(accountId) as "Jumlah_Pemakai" from status_bookworm group by plantAvatar Having Jumlah_Pemakai > 1 order by Jumlah_Pemakai asc; +-----+ plantAvatar Jumlah_Pemakai +-----+ Thickleaf Phlox 2 +-----+ 1 row in set (0.003 sec) </pre>
Tabel Yang Terlibat	Status_bookworm

Informasi	Avatar hewan yang digunakan oleh lebih dari 1 pengguna
Query	Select animalAvatar, count(accountId) as "Jumlah_Pemakai" from status_socializer group by animalAvatar Having Jumlah_Pemakai > 1 order by Jumlah_Pemakai asc
Hasil Query	<pre> MariaDB [perpusnas]> select animalAvatar , COUNT(accountId) as "Jumlah_Pemakai" from status_socializer group by animalAvatar Having Jumlah_Pemakai > 1 order by Jumlah_Pemakai asc; +-----+ animalAvatar Jumlah_Pemakai +-----+ Red-winged blackbird 2 Tortoise, desert 2 Black and white colobus 2 Yellow-necked spurfowl 2 Onager 2 Cockatoo, long-billed 2 Woodchuck 2 Coatimundi, ring-tailed 2 Cat, toddy 2 Pine siskin 2 +-----+ 10 rows in set (0.001 sec) </pre>

Tabel Yang Terlibat	Status_socializer
Informasi	Buku dengan jumlah peminjaman lebih banyak daripada jumlah reservasi
Query	<p>Create view jumlahReservasi as Select bookId, count(reservationId) as "TOTAL" from reservation group by bookId;</p> <p>Create view jumlahPeminjaman as Select bookId, count(LendingId) as "TOTAL_PINJAM" from lending group by bookId;</p> <p>Select title from book where bookId in (select bookId from jumlahReservasi natural join jumlahPeminjaman where TOTAL > TOTAL_PINJAM);</p>
Hasil Query	<pre> MariaDB [perpusnas]> create view jumlahReservasi as -> select bookId , Count(reservationId) as "TOTAL" from reservation group by bookId ; Query OK, 0 rows affected (0.006 sec) MariaDB [perpusnas]> create view jumlahPeminjaman as -> select bookId , Count(lendingId) as "TOTAL_PINJAM" from lending group by bookId ; Query OK, 0 rows affected (0.137 sec) MariaDB [perpusnas]> select title from book where bookId in (select bookId from jumlahReservasi natural join jumlahPeminjaman where TOTAL > TOTAL_PINJAM); Empty set (0.018 sec) </pre>
Tabel Yang Terlibat	Reservation, Lending, Book

Informasi	Daftar Nama Lengkap pengguna perpustakaan yang membuat note terbanyak pada tahun 2022
Query	<p>Create view hitungNote as Select accountId, count(nodeId) as "Jumlah" from personalNode where noteCreationTime like '2022%' group by accountId;</p> <p>select fullName, Jumlah as "Jumlah_Note" from libraryCard natural join hitungNote natural join account where accountId in (select accountId from hitungNote where jumlah = (select Max(Jumlah) from hitungNote)) ;</p>

Hasil Query

```
MariaDB [perpusnas]> create view hitungNote as  
-> select accountId , count(noteId) as "Jumlah" from personalNote where noteCreationTime like '2022%' group by accountId;  
Query OK, 0 rows affected (0.004 sec)
```

```
MariaDB [perpusnas]> _
```

```
MariaDB [perpusnas]> select fullName , Jumlah as "Jumlah_Note" from libraryCard natural join hitungNote natural join account where accountId in (select accountId from hitungNote where jumlah = (select Max(Jumlah) from hitungNote));
```

fullName	Jumlah_Note
Tobe Swaile	2
Tim Jordine	2
Carey Weber	2
Valerye Gisburne	2
Ruthi Lindenbaum	2
Matthaeus Coely	2
Codee Scarbarrow	2
Leighton Bedell	2
Vallie Ferrulli	2
Rourke Conechie	2
Halli Andrioletti	2
Aggie Wherrit	2
Mattha Lissaman	2
Malnwright Stephens	2
Maggie Laughtisse	2
Ed Cattonnet	2
Rancell Christer	2
Ames-corinne Farrans	2
Jarrid Furtado	2
Dyann Hemphall	2
Lucias Trulock	2
Parnell Matuszinski	2
Lira Moulding	2
Miguelita Menworth	2
Maris Ickovici	2
Sandie Eatttok	2
Aida Jeanneau	2
Tirrell De Francisci	2
Brice Dwane	2
Eldredge Senior	2
Selene Lorraine	2
Cherey Layborn	2
Tommy Denness	2
Hadlee Stubbs	2
Geordina Helton	2
Wendell Jursch	2
Giuseppe Janosevic	2
Becka Ripon	2

Activate Windows
Go to Settings to activate Windows.

Becka Ripon	2
Joellen Senescall	2
Weiden Rizzardini	2
Bobby Ibert	2
Lonnie Feehan	2
Murdoch Kennler	2
Roland Hawock	2
Ofelia Ogilvie	2
Neille Cattrall	2
Anatole Tichner	2
Mehetabel Bonnyson	2
Malcolm Lorinez	2
Flem Miell	2
Nev Hanson	2
Mill Grant	2
Gilbertina McClymont	2
Stanleigh Litel	2
Oily Yates	2
Ole Annaudot	2
Sapphira Cremer	2
Juliana Pomfrett	2
Sib Skellion	2
Fleur Goldie	2
Loraloe Marzella	2
Chaddie Gronaver	2
Sada Routhorn	2
Jenni Kitcherside	2
Conant Brason	2
Ezri Sedwick	2
Virginia Middleditch	2
Stefan Connelly	2
Colleen Donzey	2
Bryon Hurne	2
Maurie Shewing	2
Merrily Handscomb	2
Ned Reimard	2
Reinaldo Keld	2
Chris Beaves	2
Yulma Dulany	2
Paddle Vedall	2
Car Ashness	2
Benedetta Grooby	2
Lesli Spunner	2
Kelly Purviss	2

Activate Windows
Go to Settings to activate Windows.

```
SQL Command Prompt - mysql ~ root
```

Kelly Purviss	2
Calv Early	2
Tera Ferrao	2
Nicoli Whittles	2
Christabella Readman	2
Birgit Cosbey	2
Celle Grubey	2
Veronica Swalowe	2
Yank Minty	2
Bartolomeo Ledgerton	2
Joni Brookfield	2
Rob Defew	2
Dotty Thomsson	2
Corliss Newhouse	2
Roderich Agass	2
Geoffrey Joyce	2
Sandi Desquesnes	2
Laurence Fike	2
Marlena Albutt	2
Nixie Worsam	2
Cyndy Pickston	2
Herrick Budibent	2
Maisey Gozard	2
Skipp Bocken	2
Krishnan Bree	2
Saundra Todd	2
Costa Tomaskunas	2
Pascale Wissbey	2
Mazella Blench	2
Stevy Causy	2
Jeannie Pauli	2
Camey Iredale	2
Yoko Boswell	2
Estelle Spencock	2
Barb Andryushin	2
Don Tubbles	2
Bastian Seale	2
Adelle Hallis	2
Heidi Devonside	2
Berkie Hammarberg	2
Errick Beesley	2

121 rows in set (0.009 sec)

Activate Windows
Go to Settings to activate Windows.

Tabel Yang Terlibat	personalNote, libraryCard, Account
---------------------	------------------------------------

Informasi	Daftar akun yang memiliki minimal 5 follower
Query	select accountId, count(followerId) as "Jumlah_Follower" from friendship group by accountId having Jumlah_Follower >= 5 order by accountId;
Hasil Query	<pre> MariaDB [perpusnas]> select accountId, count(followerId) as "Jumlah_Follower" from friendship group by accountId having Jumlah_Follower >= 5 order by accountId; +-----+ accountId Jumlah_Follower +-----+ acc015 5 acc042 6 acc049 5 acc073 7 acc093 6 acc098 5 acc105 5 acc139 5 acc157 6 acc161 5 acc215 5 acc216 5 acc234 5 acc254 5 acc256 6 acc287 5 +-----+ 16 rows in set (0.001 sec) </pre>
Tabel Yang Terlibat	Friendship

Daftar Pustaka

(Alternate Key in DBMS - Javatpoint, n.d.;

DBMS Functional Dependency - Javatpoint, n.d.;

Finding Attribute Closure and Candidate Keys Using Functional Dependencies - GeeksforGeeks, n.d.;

Functional Dependency and Attribute Closure - GeeksforGeeks, n.d.;

Identiy Functional Dependency (Normalization), n.d.;

Relational Database Design, n.d.;

What Is Foreign Key?(Database Column Reference), n.d.)

Lampiran

Pembagian Tugas

Nama	Deskripsi Tugas
Marvel Subekti	<ul style="list-style-type: none">- Melakukan identifikasi functional dependencies dan bentuk normal dari tabel relasi Account, Account_PhoneNumber, Friendship, dan Personal_Note- Menghubungi asisten dan menjadwalkan asistensi- Merapikan MoM- Melakukan Dekomposisi tabel pada sql- Melakukan- MeMerapikan laporan milestone 2 dan 3
Reswara Trista Aulia C	<ul style="list-style-type: none">- Melakukan identifikasi functional dependencies dan bentuk normal dari tabel relasi Book, Category, BookCategory, Reservation- Merapikan laporan- Merevisi skema relasional
Vincent Franstyo	<ul style="list-style-type: none">- Melakukan identifikasi functional dependencies dan bentuk normal dari table relasi Review, History, Lending.- Merapikan MoM- Merapikan daftar pustaka- Melakukan Dekomposisi tabel pada sql
Ghaylan M. Fatih	<ul style="list-style-type: none">- Melakukan identifikasi functional dependencies dan bentuk normal dari tabel relasi Status_BookWorm, Status_Socializer, libraryCard- Merapikan MoM

Moments of Meeting

Tugas Besar IF2140 Pemodelan Basis Data

Hari	Senin	Tanggal	21 November 2022
Kelas	02	Kelompok	11
Waktu	19.00-19.30	Tempat	Zoom
Anggota Kelompok	NIM	Nama	
	18221042	Ghaylan Muhammad Fatih	
	18221058	Marvel Subekti	
	18221100	Vincent Franstyo	
	18221122	Reswara Trista Aulia Candrakanti	
Milestone	Milestone 3		
Nama Asisten	Dewa Ayu Mutiara Kirana P D		

Hasil Asistensi

No.	Pertanyaan	Jawaban
1.	Untuk Week Entity pada relasi history apakah perlu di dekomposisi kan ?	Tidak , karena nilai atribut lendingId saja sudah cukup bisa untuk menjadi superkey
2	Untuk Foreign Key apakah perlu dimasukan ke functional dependency juga ?	Perlu karena ditabel relasi itu udah satu kesatuan

Dokumentasi

