LIBRARY MANAGEMENT SYSTEM

Introduction to Databases

Understanding of database design and implementation including Conceptual data modelling, logical & physical data modelling, Data Definition Language, and Structured Query language.

- Semester 1, 2022/23 –

Submitted by

DONGHYEOK LEE (21234175)

Table of Contents

1. Part 1: Conceptual Design - 40%.

Suggest a situation where you can use a database to manage and record daily transactions. (200-400 words.

(10 marks)

1.1. Situation of library management system database.

I have been asked to design a database for the library management system used in Dublin, Ireland. The requirements are as follows.

The main members of the library management are library administrators, employees, and members. An administrator is given a unique id and has other information of its name and contact number. An employee is given a unique id and has other information of its name, contact number, and designation. A member is given a unique id and has other information of its name, registration date, contact number, address, member type (normal member or blacklist member), and birthday. Other components within the management system include library, book, author, publisher, and vendor (book sales company). A library has a unique name and other information of its address and contact number information. A book has a unique id and other information of the book name, book price, book state (available to lent or not), and book received date. An author has a unique id, and other information of the name and the author subject. A publisher has unique a id and other information of its name and its country. A vendor (Book sales company) has a unique id and contact number.

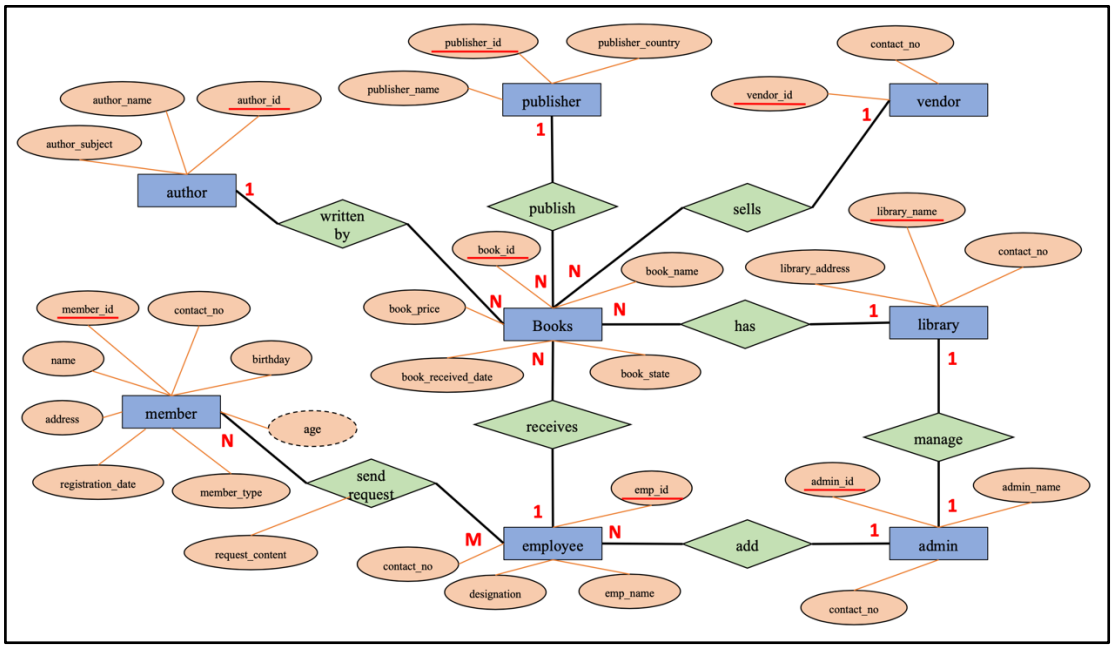
One library is managed by a one administrator. One library has many books. One administrator adds multiple employees for library management. One employee receives many books. Library members send requests to many of employees and employees also send reply to many of library members. One author writes many books. One publisher company publishes many books. One vendor (book sales company) sells many books.

Draw a suitable ER diagram for your database; consider the following in your conceptual design:

- *Include at least five entities.*
- *The relationship between entities should include one-to-one, one-to-many and many-to-many.*
- *Provide an example of a derived attribute in your design*

(15 marks)

1.2. ER Diagram of Library management system.



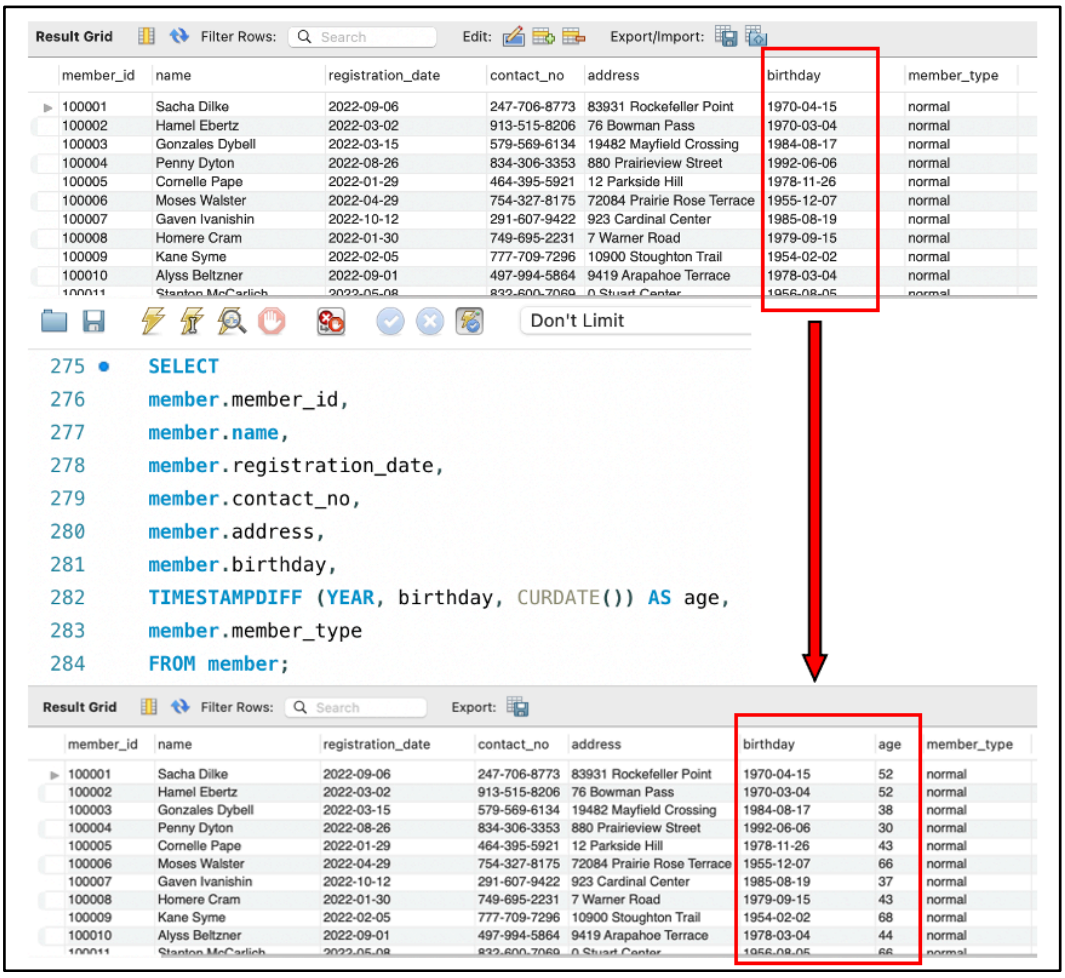
1) Include at least five entities.
   - In above ER diagram, there are 8 entities (library, admin, employee, vendor, author, publisher, member, and book).

2) The relationship between entities should include one-to-one, one-to-many and many-to-many.
   - In above ER diagram, the relationships between entities are like following.
     a. one-to-one.
        - library – admin.

     b. one-to-many.
        - admin – employee.
        - library – book.
        - employee – book.
        - publisher – book.
        - vendor – book.
        - author – book.

     c. many-to-many.
        - member – employee.

3) Provide an example of a derived attribute in your design.
   - In above ER diagram, there is a derived attribute. It is “age” attribute of member entity. It’s derived from birthday attribute of member entity.
   - Derived attribute is calculated like following.



Result Grid — Filter Rows: Search — Edit: — Export/Import:

member_id	name	registration_date	contact_no	address	birthday	member_type
100001	Sacha Dilke	2022-09-06	247-706-8773	83931 Rockefeller Point	1970-04-15	normal
100002	Hamel Ebertz	2022-03-02	913-515-8206	76 Bowman Pass	1970-03-04	normal
100003	Gonzales Dybell	2022-03-15	579-569-6134	19482 Mayfield Crossing	1984-08-17	normal
100004	Penny Dyton	2022-08-26	834-306-3353	880 Prairieview Street	1992-06-06	normal
100005	Cornelle Pape	2022-01-29	464-395-5921	12 Parkside Hill	1978-11-26	normal
100006	Moses Walster	2022-04-29	754-327-8175	72084 Prairie Rose Terrace	1955-12-07	normal
100007	Gaven Ivanishin	2022-10-12	291-607-9422	923 Cardinal Center	1985-08-19	normal
100008	Homere Cram	2022-01-30	749-695-2231	7 Warner Road	1979-09-15	normal
100009	Kane Syme	2022-02-05	777-709-7296	10900 Stoughton Trail	1954-02-02	normal
100010	Alyss Beltzner	2022-09-01	497-994-5864	9419 Arapahoe Terrace	1978-03-04	normal

Don't Limit

```
275 •  SELECT
276    member.member_id,
277    member.name,
278    member.registration_date,
279    member.contact_no,
280    member.address,
281    member.birthday,
282    TIMESTAMPDIFF (YEAR, birthday, CURDATE()) AS age,
283    member.member_type
284    FROM member;
```

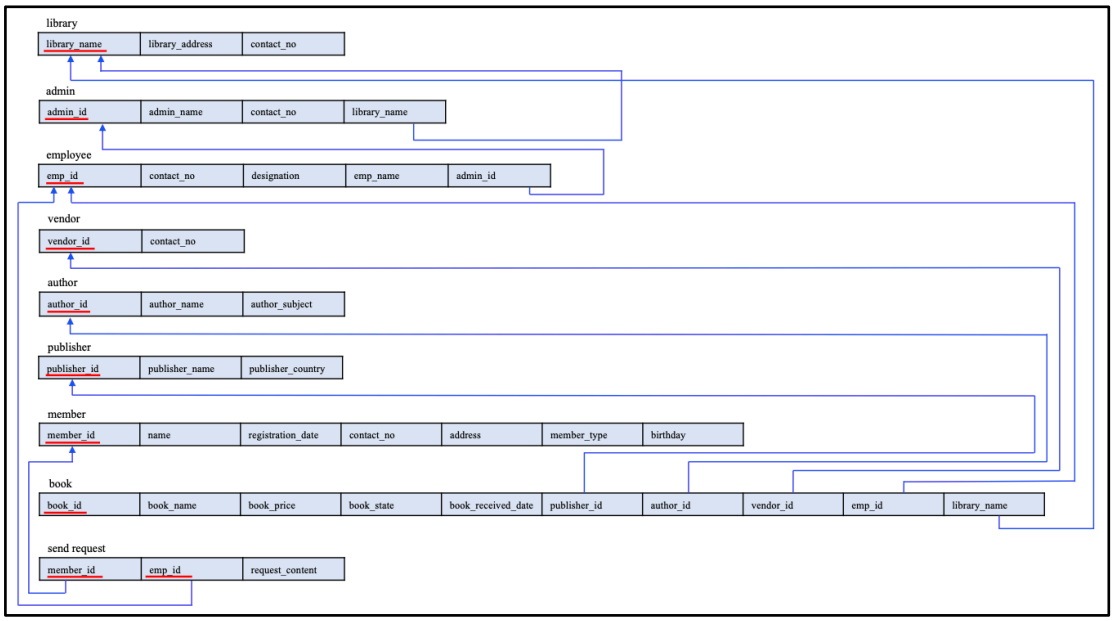
Result Grid — Filter Rows: Search — Export:

member_id	name	registration_date	contact_no	address	birthday	age	member_type
100001	Sacha Dilke	2022-09-06	247-706-8773	83931 Rockefeller Point	1970-04-15	52	normal
100002	Hamel Ebertz	2022-03-02	913-515-8206	76 Bowman Pass	1970-03-04	52	normal
100003	Gonzales Dybell	2022-03-15	579-569-6134	19482 Mayfield Crossing	1984-08-17	38	normal
100004	Penny Dyton	2022-08-26	834-306-3353	880 Prairieview Street	1992-06-06	30	normal
100005	Cornelle Pape	2022-01-29	464-395-5921	12 Parkside Hill	1978-11-26	43	normal
100006	Moses Walster	2022-04-29	754-327-8175	72084 Prairie Rose Terrace	1955-12-07	66	normal
100007	Gaven Ivanishin	2022-10-12	291-607-9422	923 Cardinal Center	1985-08-19	37	normal
100008	Homere Cram	2022-01-30	749-695-2231	7 Warner Road	1979-09-15	43	normal
100009	Kane Syme	2022-02-05	777-709-7296	10900 Stoughton Trail	1954-02-02	68	normal
100010	Alyss Beltzner	2022-09-01	497-994-5864	9419 Arapahoe Terrace	1978-03-04	44	normal

Convert the conceptual design into a relational model. Make sure that the tables are in a 3rd normal form.

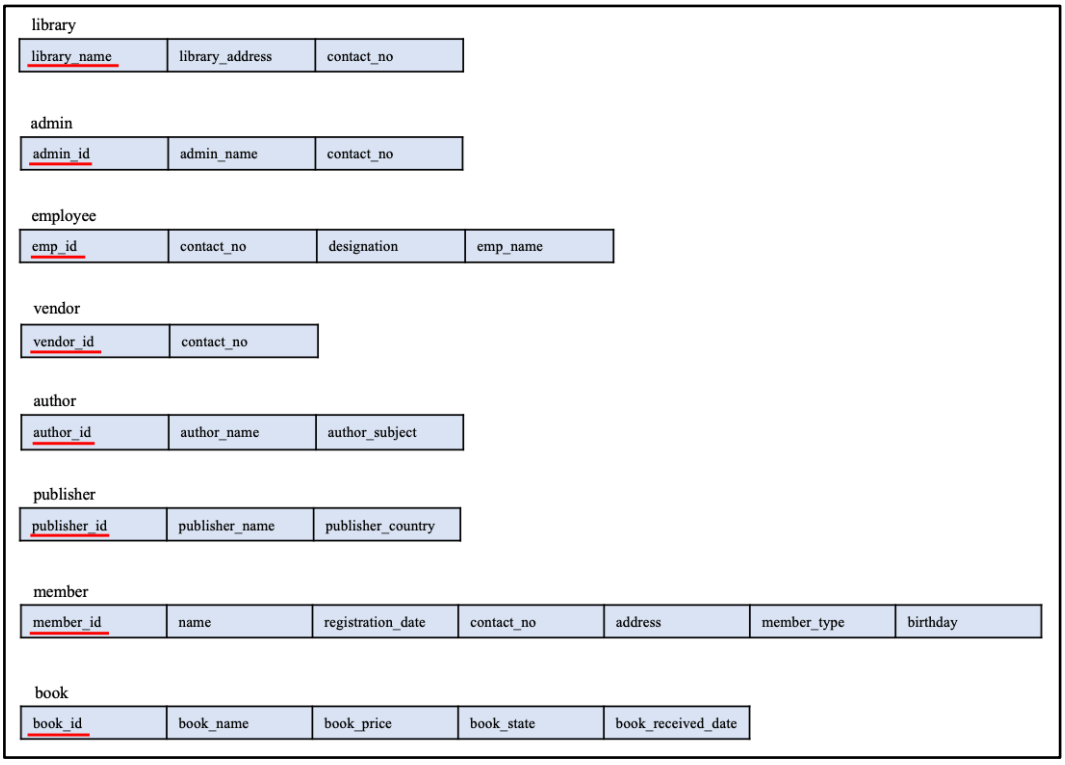
(15 marks)

1.3. Relational model of library management system.



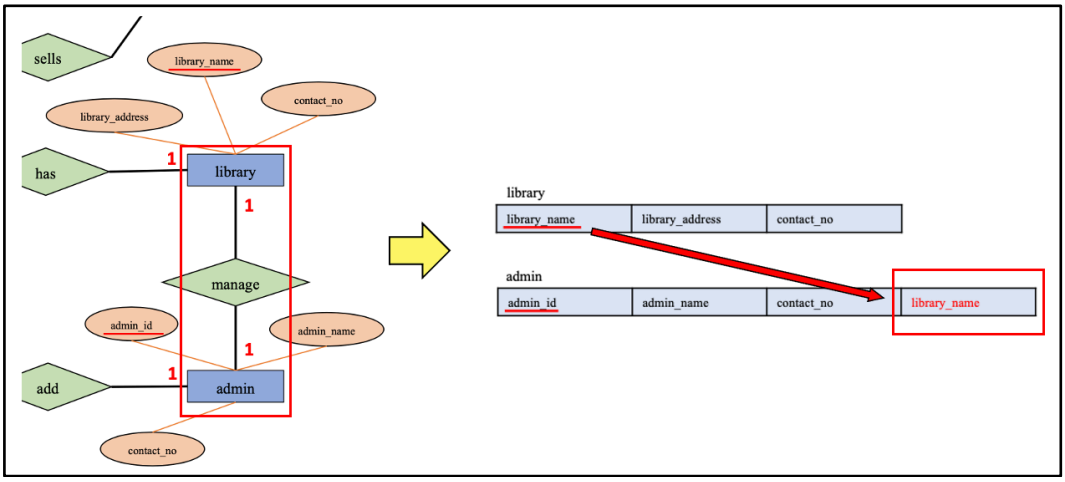
Above relational model was converted with below steps.

1) Step 1: Mapping of Regular Entity Types.

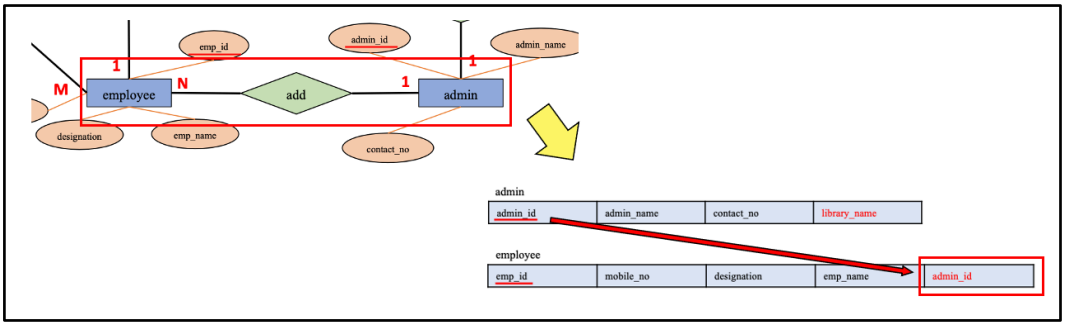


2) Step 2: Mapping of Weak Entity Types.
   - Admin is a weak entity type. So, use library_name column as a foreign key.
   - Admin_id and library_name are used for composite primary key.

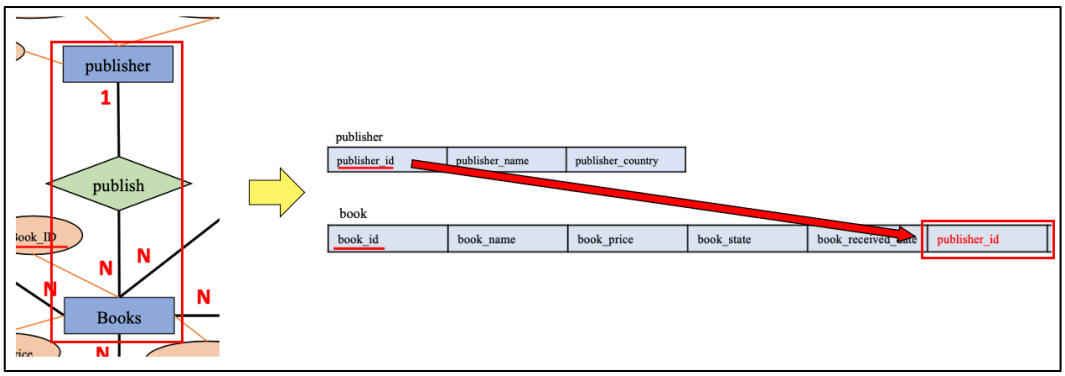
3) Step 3: Mapping of Binary 1:1 Relationship Types.
   - library – admin.



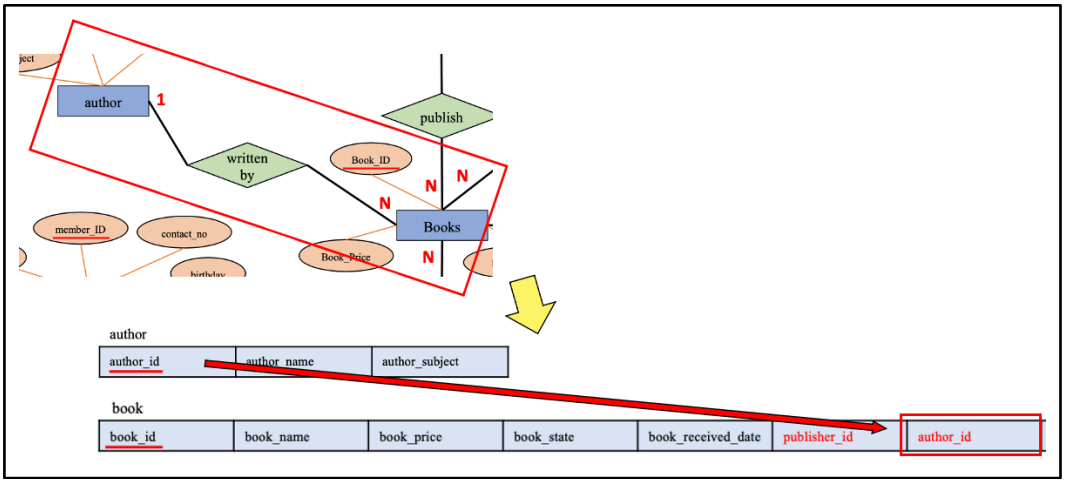
4) Step 4: Mapping of Binary 1:N Relationship Types.
   - admin – employee.



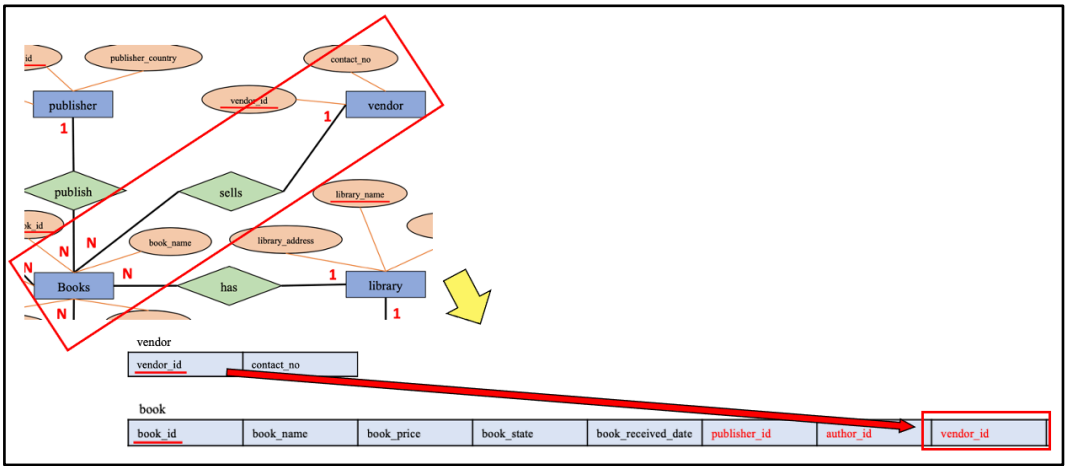
   - publisher – book.



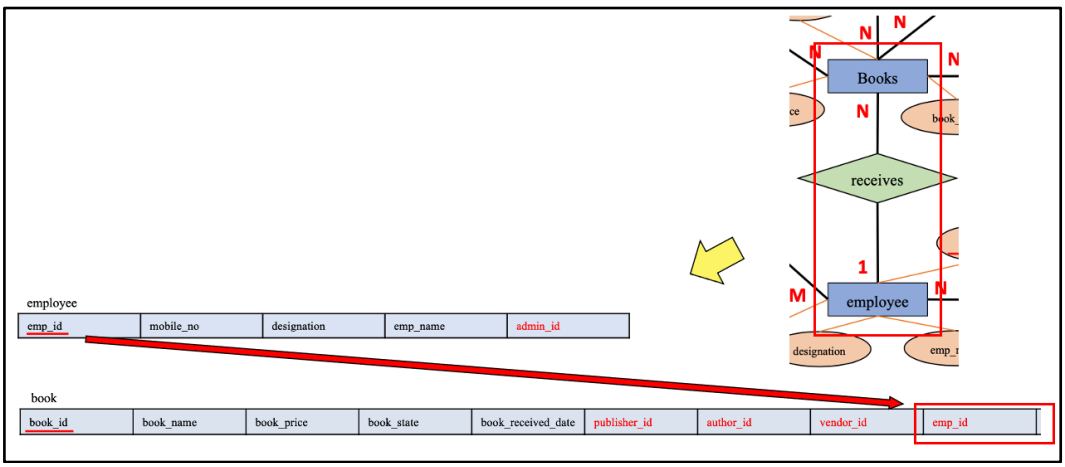
- author – book.



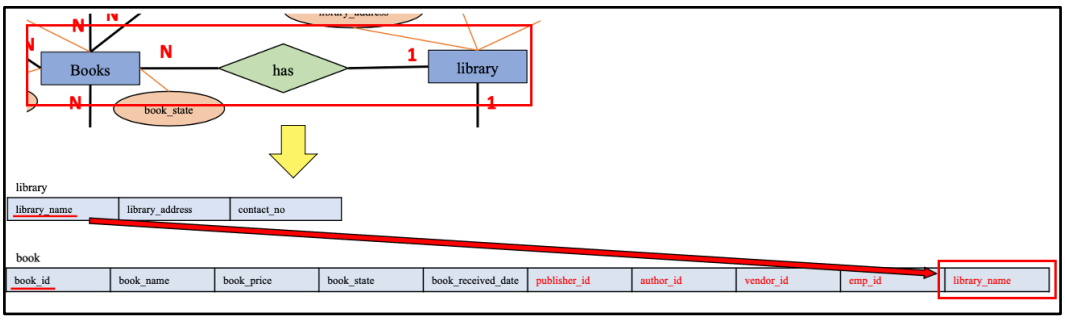
- vendor – book.



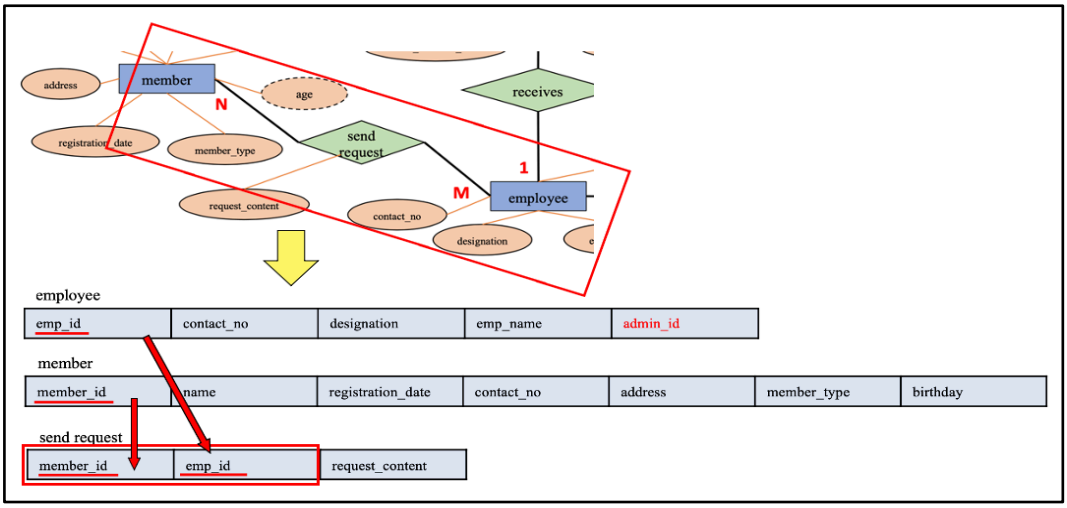
- employee – book.



- library – book.

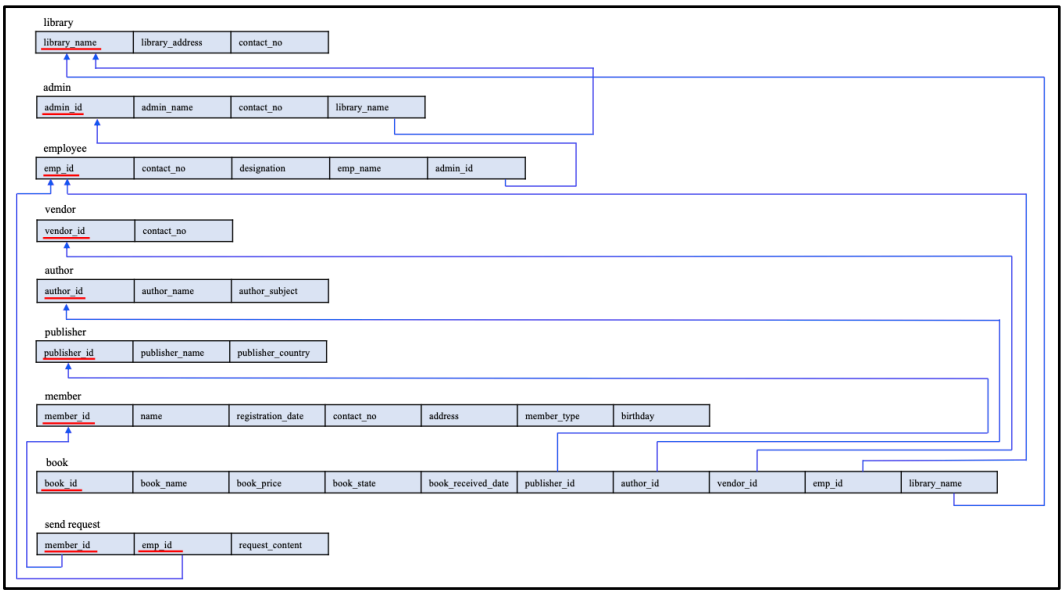


5) Step 5: Mapping of Binary M:N Relationship Types.
   - member – employee.



6) Step 6: Mapping of Multivalued Attributes .
   - There is no multivalued attribute in this database.

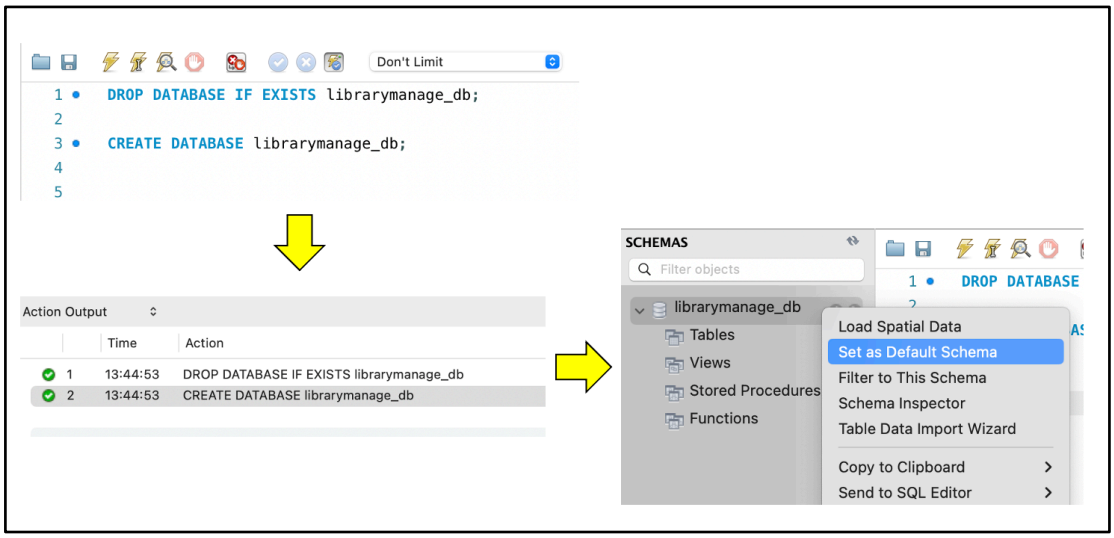
7) Step 7: Mapping of N-ary Relationship Types.



2. Part 2: Physical design - 20% (5 marks each).

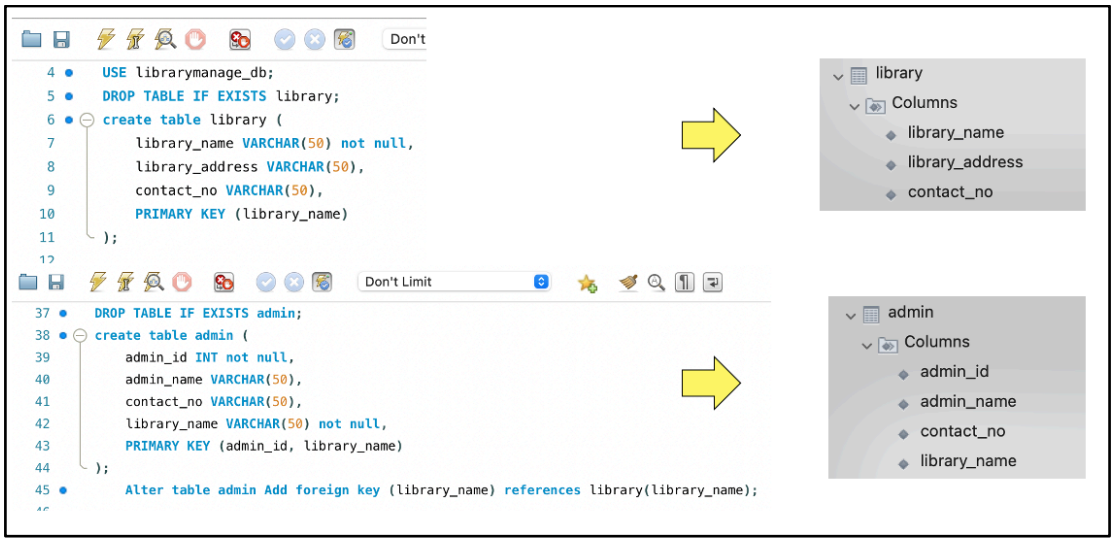
Create the corresponding database using DDL

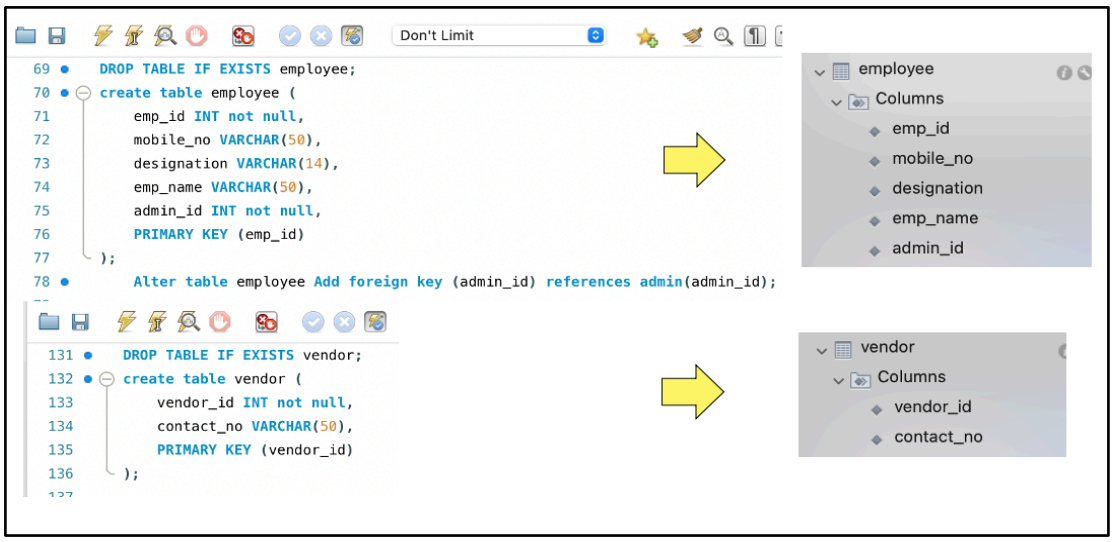
2.1. Create 'Librarymanage_db' database using DDL.

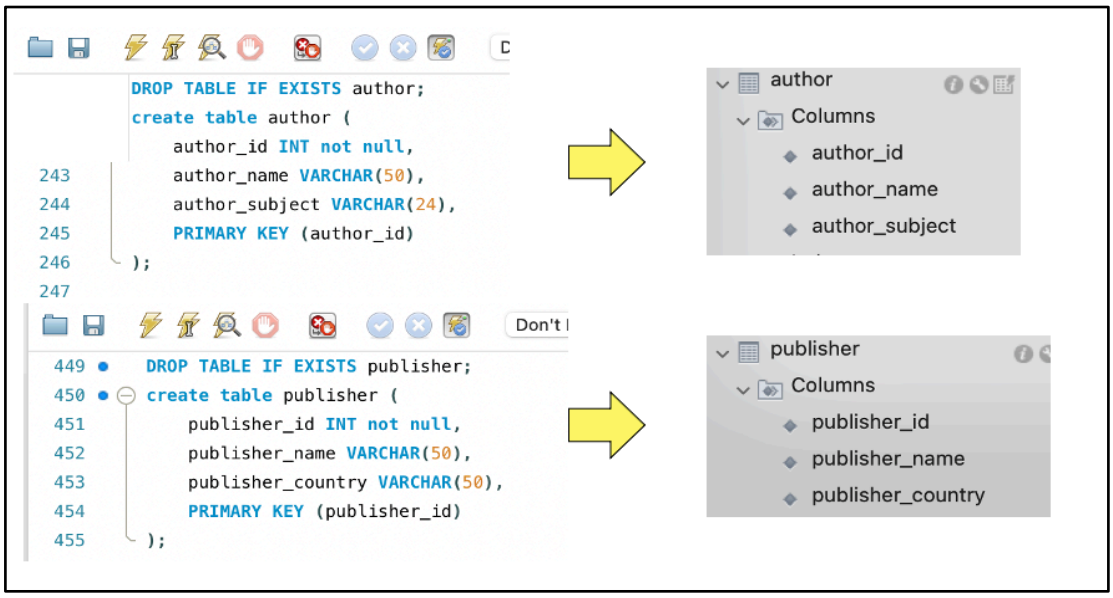


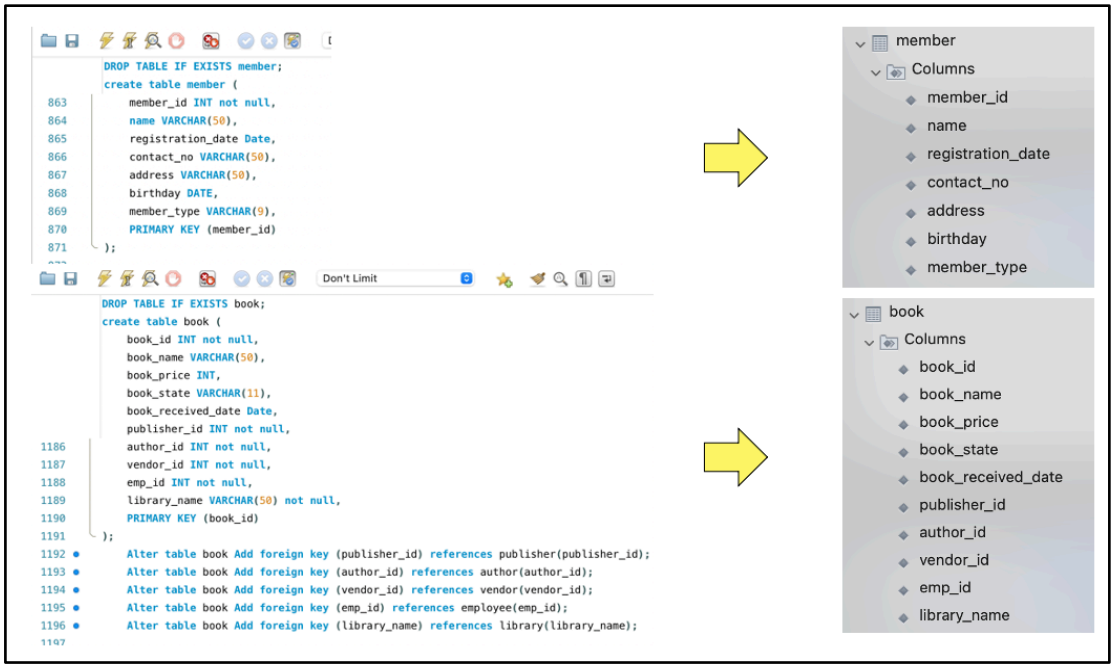
Create all the necessary tables identified above using DDL

2.2. Create all tables of 'Librarymanage_db' database using DDL.




Don't Limit
DROP TABLE IF EXISTS employee;
create table employee (
emp_id INT not null,
mobile_no VARCHAR(50),
designation VARCHAR(14),
emp_name VARCHAR(50),
admin_id INT not null,
PRIMARY KEY (emp_id)
);
Alter table employee Add foreign key (admin_id) references admin(admin_id);
employee
Columns
emp_id
mobile_no
designation
emp_name
admin_id
DROP TABLE IF EXISTS vendor;
create table vendor (
vendor_id INT not null,
contact_no VARCHAR(50),
PRIMARY KEY (vendor_id)
);
vendor
Columns
vendor_id
contact_no


DROP TABLE IF EXISTS author;
create table author (
author_id INT not null,
author_name VARCHAR(50),
author_subject VARCHAR(24),
PRIMARY KEY (author_id)
);
author
Columns
author_id
author_name
author_subject
DROP TABLE IF EXISTS publisher;
create table publisher (
publisher_id INT not null,
publisher_name VARCHAR(50),
publisher_country VARCHAR(50),
PRIMARY KEY (publisher_id)
);
publisher
Columns
publisher_id
publisher_name
publisher_country


DROP TABLE IF EXISTS member;
create table member (
member_id INT not null,
name VARCHAR(50),
registration_date Date,
contact_no VARCHAR(50),
address VARCHAR(50),
birthday DATE,
member_type VARCHAR(9),
PRIMARY KEY (member_id)
);
member
Columns
member_id
name
registration_date
contact_no
address
birthday
member_type
DROP TABLE IF EXISTS book;
create table book (
book_id INT not null,
book_name VARCHAR(50),
book_price INT,
book_state VARCHAR(11),
book_received_date Date,
publisher_id INT not null,
author_id INT not null,
vendor_id INT not null,
emp_id INT not null,
library_name VARCHAR(50) not null,
PRIMARY KEY (book_id)
);
Alter table book Add foreign key (publisher_id) references publisher(publisher_id);
Alter table book Add foreign key (author_id) references author(author_id);
Alter table book Add foreign key (vendor_id) references vendor(vendor_id);
Alter table book Add foreign key (emp_id) references employee(emp_id);
Alter table book Add foreign key (library_name) references library(library_name);
book
Columns
book_id
book_name
book_price
book_state
book_received_date
publisher_id
author_id
vendor_id
emp_id
library_name

```
2203   DROP TABLE IF EXISTS send_request;
2204   create table send_request (
2205       member_id INT not null,
2206       emp_id INT not null,
2207       request_content VARCHAR(50),
2208       primary key (member_id, emp_id)
2209   );
2210       Alter table send_request Add foreign key (member_ID) references member(member_ID);
2211       Alter table send_request Add foreign key (emp_ID) references employee(emp_ID);
```

Populate at least three of your tables with some data using DML (insert into statement)

2.3. Populate 3 tables of 'Librarymanage_db' database with some data using DML.

```
13   insert into library (library_name, library_address, contact_no) values ('Murrays Plum', '7 Kipling Lane', '991-870-9077');
14   insert into library (library_name, library_address, contact_no) values ('Trailing Arbutus', '5 Maywood Road', '543-488-0116');
15   insert into library (library_name, library_address, contact_no) values ('Azure Blue Sage', '6320 Lawn Plaza', '128-142-5586');
16   insert into library (library_name, library_address, contact_no) values ('Nehe Kuhiwa', '1 Menomonie Alley', '769-512-3820');
17   insert into library (library_name, library_address, contact_no) values ('Hoary Yellowcress', '2 Troy Parkway', '724-198-9156');
18   insert into library (library_name, library_address, contact_no) values ('Boise Milkvetch', '295 Lighthouse Bay Trail', '866-992-5832');
19   insert into library (library_name, library_address, contact_no) values ('Broadleaf Solomons Seal', '96354 Loomis Road', '229-826-0680');
20   insert into library (library_name, library_address, contact_no) values ('Dusty Beardtongue', '73 Ilene Way', '818-277-4636');
21   insert into library (library_name, library_address, contact_no) values ('Mediterranean Cabbage', '64710 Eagle Crest Park', '660-185-7702');
22   insert into library (library_name, library_address, contact_no) values ('Hybrid Willow', '0563 Express Road', '771-808-1861');
23   insert into library (library_name, library_address, contact_no) values ('Western Pearlwort', '70 Northridge Court', '875-554-7682');
24   insert into library (library_name, library_address, contact_no) values ('Disc Lichen', '9683 Acker Drive', '712-584-8021');
25   insert into library (library_name, library_address, contact_no) values ('Southern Crab Apple', '7 Oneill Park', '334-248-7910');
26   insert into library (library_name, library_address, contact_no) values ('Warnstorfia Moss', '2 Troy Way', '390-360-4549');
27   insert into library (library_name, library_address, contact_no) values ('Eatonella', '10 Melrose Way', '568-487-3261');
28   insert into library (library_name, library_address, contact_no) values ('Rim Lichen', '7 Anthes Place', '744-566-2481');
29   insert into library (library_name, library_address, contact_no) values ('Cade Juniper', '16 Cherokee Parkway', '791-244-1662');
30   insert into library (library_name, library_address, contact_no) values ('Desert Yellow Fleabane', '35 Red Cloud Center', '556-467-8941');
31   insert into library (library_name, library_address, contact_no) values ('Butte Desertparsley', '83 Parkside Junction', '434-969-3443');
32   insert into library (library_name, library_address, contact_no) values ('Lanceleaf Ticktrefoil', '9 Claremont Terrace', '710-339-6387');
```

Result Grid Filter Rows: Search Edit:

library_name	library_address	contact_no
Azure Blue Sage	6320 Lawn Plaza	128-142-5586
Boise Milkvetch	295 Lighthouse Bay Trail	866-992-5832
Broadleaf Solomons Seal	96354 Loomis Road	229-826-0680
Butte Desertparsley	83 Parkside Junction	434-969-3443
Cade Juniper	16 Cherokee Parkway	791-244-1662
Desert Yellow Fleabane	35 Red Cloud Center	556-467-8941
Disc Lichen	9683 Acker Drive	712-584-8021
Dusty Beardtongue	73 Ilene Way	818-277-4636
Eatonella	10 Melrose Way	568-487-3261
Hoary Yellowcress	2 Troy Parkway	724-198-9156

```
insert into admin (admin_id, admin_name, contact_no, library_name) values (1, 'Cammy Hinstock', '256-606-5975', 'Murrays Plum');
insert into admin (admin_id, admin_name, contact_no, library_name) values (2, 'Cos Cocking', '348-821-1178', 'Trailing Arbutus');
insert into admin (admin_id, admin_name, contact_no, library_name) values (3, 'Rachele Tamsett', '347-358-7143', 'Azure Blue Sage');
insert into admin (admin_id, admin_name, contact_no, library_name) values (4, 'Zared Ratley', '406-758-2897', 'Nehe Kuhiwa');
insert into admin (admin_id, admin_name, contact_no, library_name) values (5, 'Fonsie Jouhan', '462-109-5552', 'Hoary Yellowcress');
insert into admin (admin_id, admin_name, contact_no, library_name) values (6, 'Saidee Menichi', '693-659-5372', 'Boise Milkvetch');
insert into admin (admin_id, admin_name, contact_no, library_name) values (7, 'Melisent Bridewell', '512-737-8469', 'Broadleaf Solomons Seal');
insert into admin (admin_id, admin_name, contact_no, library_name) values (8, 'Barnard Hellmer', '757-674-2639', 'Dusty Beardtongue');
insert into admin (admin_id, admin_name, contact_no, library_name) values (9, 'Aldus Mallett', '874-568-6626', 'Mediterranean Cabbage');
insert into admin (admin_id, admin_name, contact_no, library_name) values (10, 'Gracie Strudwick', '607-151-5214', 'Hybrid Willow');
insert into admin (admin_id, admin_name, contact_no, library_name) values (11, 'Austin McCleod', '936-818-0121', 'Western Pearlwort');
insert into admin (admin_id, admin_name, contact_no, library_name) values (12, 'Kally Bulter', '998-943-3179', 'Disc Lichen');
insert into admin (admin_id, admin_name, contact_no, library_name) values (13, 'Liv McCadden', '740-142-0618', 'Southern Crab Apple');
insert into admin (admin_id, admin_name, contact_no, library_name) values (14, 'Van Devon', '688-177-2238', 'Warnstorfia Moss');
insert into admin (admin_id, admin_name, contact_no, library_name) values (15, 'Helaina Templman', '602-632-9117', 'Eatonella');
insert into admin (admin_id, admin_name, contact_no, library_name) values (16, 'Raffaello Martyns', '477-273-0035', 'Rim Lichen');
insert into admin (admin_id, admin_name, contact_no, library_name) values (17, 'Irina Pavinese', '834-659-0459', 'Cade Juniper');
insert into admin (admin_id, admin_name, contact_no, library_name) values (18, 'Rab Clendening', '508-545-3709', 'Desert Yellow Fleabane');
insert into admin (admin_id, admin_name, contact_no, library_name) values (19, 'Wylie Towner', '360-794-7328', 'Butte Desertparsley');
insert into admin (admin_id, admin_name, contact_no, library_name) values (20, 'Ingamar Kubiczek', '731-628-4914', 'Lanceleaf Ticktrefoil');
```

Result Grid Filter Rows: Search Edit:

admin_id	admin_name	contact_no	library_name
1	Cammy Hinstock	256-606-5975	Murrays Plum
2	Cos Cocking	348-821-1178	Trailing Arbutus
3	Rachele Tamsett	347-358-7143	Azure Blue Sage
4	Zared Ratley	406-758-2897	Nehe Kuhiwa
5	Fonsie Jouhan	462-109-5552	Hoary Yellowcress
6	Saidee Menichi	693-659-5372	Boise Milkvetch
7	Melisent Bridewell	512-737-8469	Broadleaf Solomons Seal
8	Barnard Hellmer	757-674-2639	Dusty Beardtongue
9	Aldus Mallett	874-568-6626	Mediterranean Cabbage
10	Gracie Strudwick	607-151-5214	Hybrid Willow
11	Austin McCleod	936-818-0121	Western Pearlwort

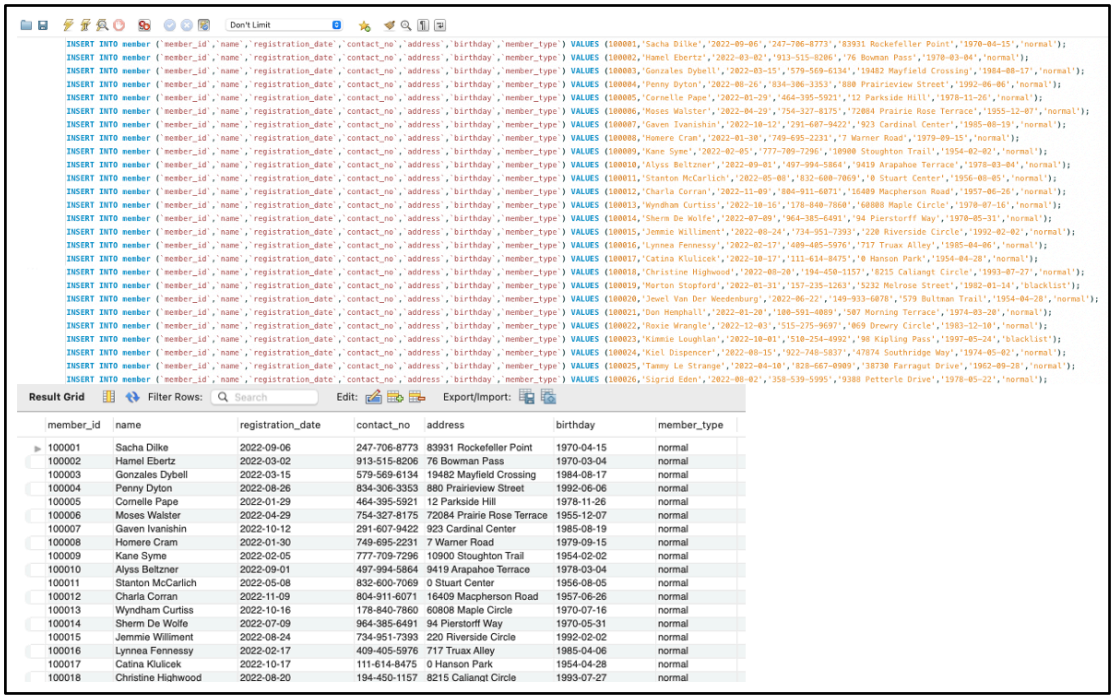
```
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (101, '354-343-6183', 'supervisor', 'Wynnie Marishenko', 1);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (102, '798-406-5703', 'senior manager', 'Natty Ismirnioglou', 2);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (103, '490-868-6350', 'manager', 'Janine Boundley', 3);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (104, '370-424-2222', 'senior manager', 'Mathilda Roddan', 4);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (105, '881-106-5778', 'associate', 'Hilly Cathee', 5);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (106, '509-134-2678', 'assistant', 'Alfonse Aitcheson', 6);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (107, '892-311-6995', 'assistant', 'Haroun Rabat', 7);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (108, '345-659-2454', 'supervisor', 'Cherin Runchman', 8);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (109, '640-942-4644', 'associate', 'Laetitia Benedicte', 9);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (110, '406-327-6593', 'assistant', 'Dulcie Bewfield', 10);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (111, '478-642-9077', 'assistant', 'Darrell Mora', 11);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (112, '167-964-3737', 'assistant', 'Olva Gockelen', 12);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (113, '680-894-9737', 'assistant', 'Tally Mordey', 13);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (114, '669-997-4696', 'manager', 'Page Colisbe', 14);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (115, '947-176-6910', 'manager', 'Nona McBeith', 15);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (116, '361-470-6771', 'associate', 'Cece OCalleran', 16);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (117, '214-996-6563', 'senior manager', 'Chaunce Unthank', 17);
insert into employee (emp_id, mobile_no, designation, emp_name, admin_id) values (118, '987-676-5914', 'associate', 'Babara Trett', 18);
```

Result Grid Filter Rows: Search Edit:

emp_id	mobile_no	designation	emp_name	admin_id
101	354-343-6183	supervisor	Wynnie Marishenko	1
102	798-406-5703	senior manager	Natty Ismirnioglou	2
103	490-868-6350	manager	Janine Boundley	3
104	370-424-2222	senior manager	Mathilda Roddan	4
105	881-106-5778	associate	Hilly Cathee	5
106	509-134-2678	assistant	Alfonse Aitcheson	6
107	892-311-6995	assistant	Haroun Rabat	7
108	345-659-2454	supervisor	Cherin Runchman	8
109	640-942-4644	associate	Laetitia Benedicte	9
110	406-327-6593	assistant	Dulcie Bewfield	10

Populate your database with a large data set representing a one-year transaction (01/01/2022 - 31/12/2022) on each table. (Use online data generators such as Mockaroo or generate data to generate synthetic data.)

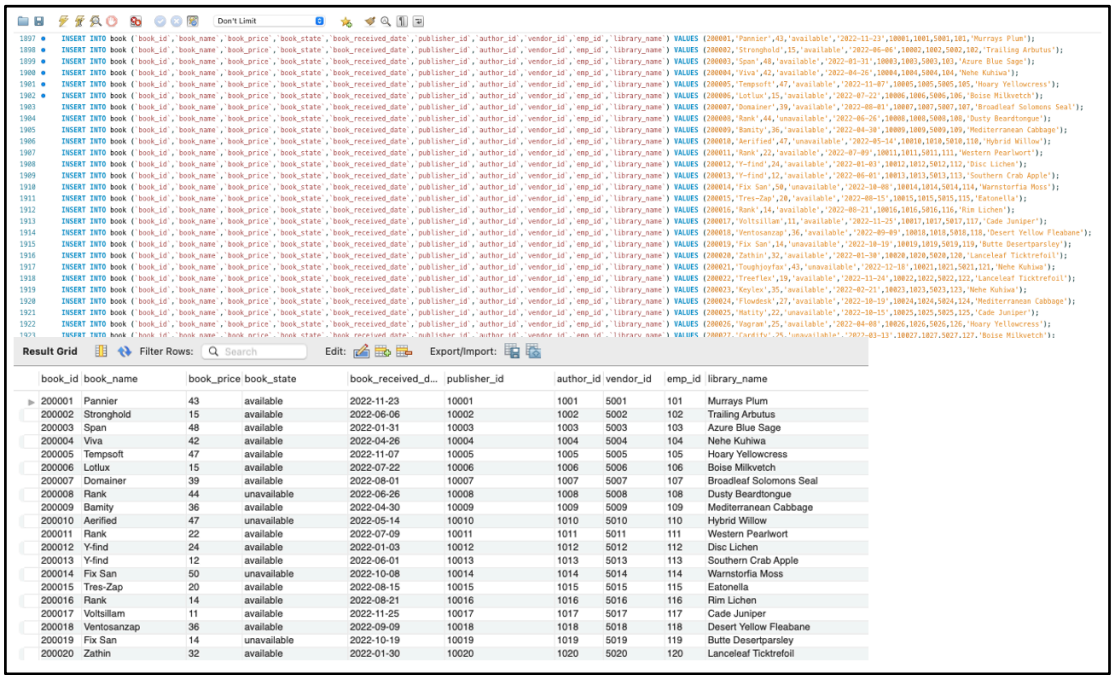
2.4. Populate tables of 'Librarymanage_db' database a large data set representing a one-year transaction (01/01/2022 - 31/12/2022) on each table.



```
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100001,'Sacha Dilke','2022-09-06','247-706-8773','83931 Rockefeller Point','1970-04-15','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100002,'Hamel Ebertz','2022-03-02','913-515-8206','76 Bowman Pass','1970-03-04','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100003,'Gonzales Dybell','2022-03-15','579-569-6134','19482 Mayfield Crossing','1984-08-17','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100004,'Penny Dyton','2022-08-26','834-306-3353','880 Prairieview Street','1992-06-06','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100005,'Cornelle Pape','2022-01-29','464-395-5921','12 Parkside Hill','1978-11-26','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100006,'Moses Walster','2022-04-29','754-327-8175','72084 Prairie Rose Terrace','1955-12-07','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100007,'Gaven Ivanishin','2022-10-12','291-607-9422','923 Cardinal Center','1985-08-19','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100008,'Homere Cram','2022-01-30','749-695-2231','7 Warner Road','1979-09-15','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100009,'Kane Syme','2022-02-05','777-709-7296','10900 Stoughton Trail','1954-02-02','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100010,'Alyss Beltzner','2022-09-01','497-994-5864','9419 Arapahoe Terrace','1978-03-04','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100011,'Stanton McCarlich','2022-05-08','832-600-7069','0 Stuart Center','1956-08-05','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100012,'Charla Corran','2022-11-09','804-911-6071','16409 Macpherson Road','1957-06-26','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100013,'Wyndham Curtiss','2022-10-16','178-840-7860','60808 Maple Circle','1970-07-16','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100014,'Sherm De Wolfe','2022-07-09','964-385-6491','94 Pierstorff Way','1970-05-31','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100015,'Jemmie Williment','2022-08-24','734-951-7393','220 Riverside Circle','1992-02-02','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100016,'Lynnea Fennessy','2022-02-17','409-405-5976','717 Truax Alley','1985-04-06','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100017,'Catina Klulicek','2022-10-17','111-614-8475','0 Hanson Park','1954-04-28','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100018,'Christine Highwood','2022-08-20','194-450-1157','8215 Caliangt Circle','1993-07-27','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100019,'Morton Stopford','2022-01-31','157-235-1263','5232 Melrose Street','1902-01-14','blacklist');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100020,'Jewel Van Der Weerdenburg','2022-06-22','149-933-6078','579 Bultman Trail','1954-04-28','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100021,'Don Hemphall','2022-01-20','180-591-4089','507 Morning Terrace','1974-03-20','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100022,'Roxie Wrangle','2022-12-03','515-275-9697','069 Drewry Circle','1983-12-10','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100023,'Kimmie Loughlan','2022-10-01','510-254-4992','98 Kipling Pass','1997-05-24','blacklist');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100024,'Kiel Dispencer','2022-08-15','922-740-5837','47874 Southridge Way','1974-05-02','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100025,'Tammy Le Strange','2022-04-10','828-667-0909','38730 Farragut Drive','1962-09-28','normal');
INSERT INTO member (`member_id`,`name`,`registration_date`,`contact_no`,`address`,`birthday`,`member_type`) VALUES (100026,'Sigrid Eden','2022-08-02','358-539-5995','9388 Petterle Drive','1970-05-22','normal');
```

Result Grid | Filter Rows: | Edit: | Export/Import:

member_id	name	registration_date	contact_no	address	birthday	member_type
100001	Sacha Dilke	2022-09-06	247-706-8773	83931 Rockefeller Point	1970-04-15	normal
100002	Hamel Ebertz	2022-03-02	913-515-8206	76 Bowman Pass	1970-03-04	normal
100003	Gonzales Dybell	2022-03-15	579-569-6134	19482 Mayfield Crossing	1984-08-17	normal
100004	Penny Dyton	2022-08-26	834-306-3353	880 Prairieview Street	1992-06-06	normal
100005	Cornelle Pape	2022-01-29	464-395-5921	12 Parkside Hill	1978-11-26	normal
100006	Moses Walster	2022-04-29	754-327-8175	72084 Prairie Rose Terrace	1955-12-07	normal
100007	Gaven Ivanishin	2022-10-12	291-607-9422	923 Cardinal Center	1985-08-19	normal
100008	Homere Cram	2022-01-30	749-695-2231	7 Warner Road	1979-09-15	normal
100009	Kane Syme	2022-02-05	777-709-7296	10900 Stoughton Trail	1954-02-02	normal
100010	Alyss Beltzner	2022-09-01	497-994-5864	9419 Arapahoe Terrace	1978-03-04	normal
100011	Stanton McCarlich	2022-05-08	832-600-7069	0 Stuart Center	1956-08-05	normal
100012	Charla Corran	2022-11-09	804-911-6071	16409 Macpherson Road	1957-06-26	normal
100013	Wyndham Curtiss	2022-10-16	178-840-7860	60808 Maple Circle	1970-07-16	normal
100014	Sherm De Wolfe	2022-07-09	964-385-6491	94 Pierstorff Way	1970-05-31	normal
100015	Jemmie Williment	2022-08-24	734-951-7393	220 Riverside Circle	1992-02-02	normal
100016	Lynnea Fennessy	2022-02-17	409-405-5976	717 Truax Alley	1985-04-06	normal
100017	Catina Klulicek	2022-10-17	111-614-8475	0 Hanson Park	1954-04-28	normal
100018	Christine Highwood	2022-08-20	194-450-1157	8215 Caliangt Circle	1993-07-27	normal



```
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200001,'Pannier',43,'available','2022-11-23',10001,1001,5001,101,'Murrays Plum');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200002,'Stronghold',15,'available','2022-06-06',10002,1002,5002,102,'Trailing Arbutus');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200003,'Span',48,'available','2022-01-31',10003,1003,5003,103,'Azure Blue Sage');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200004,'Viva',42,'available','2022-04-26',10004,1004,5004,104,'Nehe Kuhiwa');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200005,'Tempsoft',47,'available','2022-11-07',10005,1005,5005,105,'Hoary Yellowcress');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200006,'Lotlux',15,'available','2022-07-22',10006,1006,5006,106,'Boise Milkvetch');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200007,'Domainer',39,'available','2022-08-01',10007,1007,5007,107,'Broadleaf Solomons Seal');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200008,'Rank',44,'unavailable','2022-06-26',10008,1008,5008,108,'Dusty Beardtongue');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200009,'Bamity',36,'available','2022-04-30',10009,1009,5009,109,'Mediterranean Cabbage');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200010,'Aerified',47,'unavailable','2022-05-14',10010,1010,5010,110,'Hybrid Willow');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200011,'Rank',22,'available','2022-07-09',10011,1011,5011,111,'Western Pearlwort');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200012,'Y-find',24,'available','2022-01-03',10012,1012,5012,112,'Disc Lichen');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200013,'Y-find',12,'available','2022-06-01',10013,1013,5013,113,'Southern Crab Apple');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200014,'Fix San',50,'unavailable','2022-10-08',10014,1014,5014,114,'Warnstorfia Moss');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200015,'Tres-Zap',20,'available','2022-08-15',10015,1015,5015,115,'Eatonella');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200016,'Rank',14,'available','2022-08-21',10016,1016,5016,116,'Rim Lichen');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200017,'Voltsillam',11,'available','2022-11-25',10017,1017,5017,117,'Cade Juniper');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200018,'Ventosanzap',36,'available','2022-09-09',10018,1018,5018,118,'Desert Yellow Fleabane');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200019,'Fix San',14,'unavailable','2022-10-19',10019,1019,5019,119,'Butte Desertparsley');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200020,'Zathin',32,'available','2022-01-30',10020,1020,5020,120,'Lanceleaf Ticktrefoil');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200021,'Toughjoyfax',43,'unavailable','2022-12-18',10021,1021,5021,121,'Nehe Kuhiwa');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200022,'Treeflex',19,'available','2022-11-24',10022,1022,5022,122,'Lanceleaf Ticktrefoil');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200023,'Keylex',35,'available','2022-02-21',10023,1023,5023,123,'Nehe Kuhiwa');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200024,'Flowdesk',27,'available','2022-10-19',10024,1024,5024,124,'Mediterranean Cabbage');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200025,'Matity',22,'unavailable','2022-10-15',10025,1025,5025,125,'Cade Juniper');
INSERT INTO book (`book_id`,`book_name`,`book_price`,`book_state`,`book_received_date`,`publisher_id`,`author_id`,`vendor_id`,`emp_id`,`library_name`) VALUES (200026,'Yograv',25,'available','2022-04-08',10026,1026,5026,126,'Hoary Yellowcress');
```

Result Grid | Filter Rows: | Edit: | Export/Import:

book_id	book_name	book_price	book_state	book_received_d...	publisher_id	author_id	vendor_id	emp_id	library_name
200001	Pannier	43	available	2022-11-23	10001	1001	5001	101	Murrays Plum
200002	Stronghold	15	available	2022-06-06	10002	1002	5002	102	Trailing Arbutus
200003	Span	48	available	2022-01-31	10003	1003	5003	103	Azure Blue Sage
200004	Viva	42	available	2022-04-26	10004	1004	5004	104	Nehe Kuhiwa
200005	Tempsoft	47	available	2022-11-07	10005	1005	5005	105	Hoary Yellowcress
200006	Lotlux	15	available	2022-07-22	10006	1006	5006	106	Boise Milkvetch
200007	Domainer	39	available	2022-08-01	10007	1007	5007	107	Broadleaf Solomons Seal
200008	Rank	44	unavailable	2022-06-26	10008	1008	5008	108	Dusty Beardtongue
200009	Bamity	36	available	2022-04-30	10009	1009	5009	109	Mediterranean Cabbage
200010	Aerified	47	unavailable	2022-05-14	10010	1010	5010	110	Hybrid Willow
200011	Rank	22	available	2022-07-09	10011	1011	5011	111	Western Pearlwort
200012	Y-find	24	available	2022-01-03	10012	1012	5012	112	Disc Lichen
200013	Y-find	12	available	2022-06-01	10013	1013	5013	113	Southern Crab Apple
200014	Fix San	50	unavailable	2022-10-08	10014	1014	5014	114	Warnstorfia Moss
200015	Tres-Zap	20	available	2022-08-15	10015	1015	5015	115	Eatonella
200016	Rank	14	available	2022-08-21	10016	1016	5016	116	Rim Lichen
200017	Voltsillam	11	available	2022-11-25	10017	1017	5017	117	Cade Juniper
200018	Ventosanzap	36	available	2022-09-09	10018	1018	5018	118	Desert Yellow Fleabane
200019	Fix San	14	unavailable	2022-10-19	10019	1019	5019	119	Butte Desertparsley
200020	Zathin	32	available	2022-01-30	10020	1020	5020	120	Lanceleaf Ticktrefoil

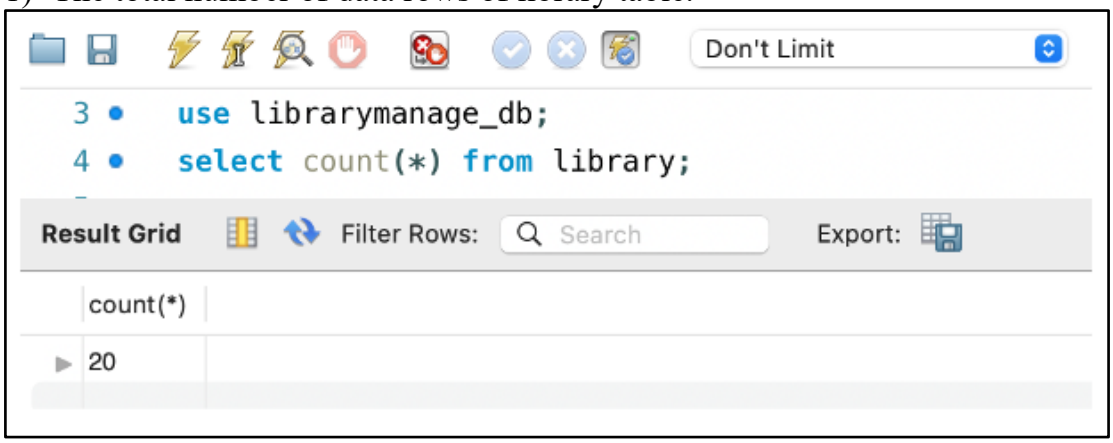
3. Part 3: Write SQL Statements to answer the following queries - 40%.

Show the total number of transactions your database is storing and, depending on your database, the most sold/listed item or customer with the highest number of purchases.

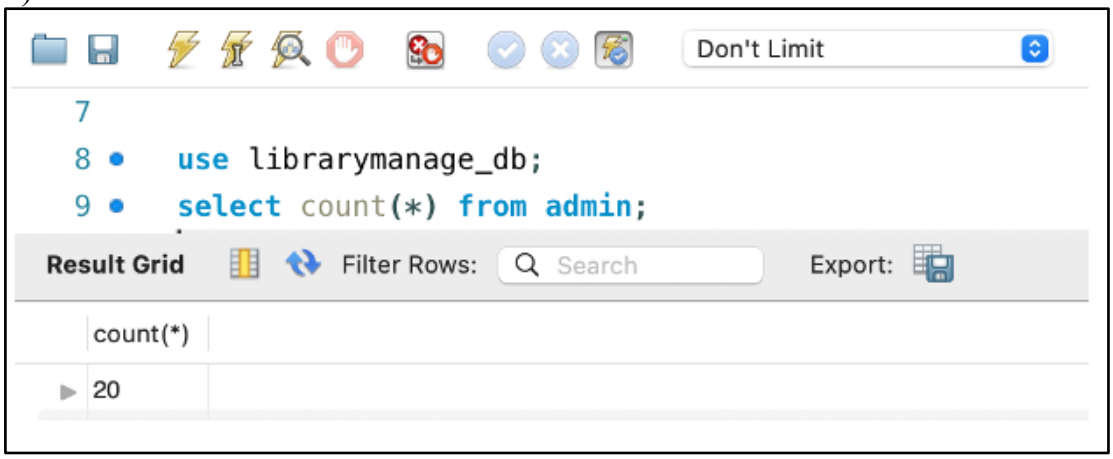
(4 marks)

3.1. The total number of transactions of 'Librarymanage_db' database.

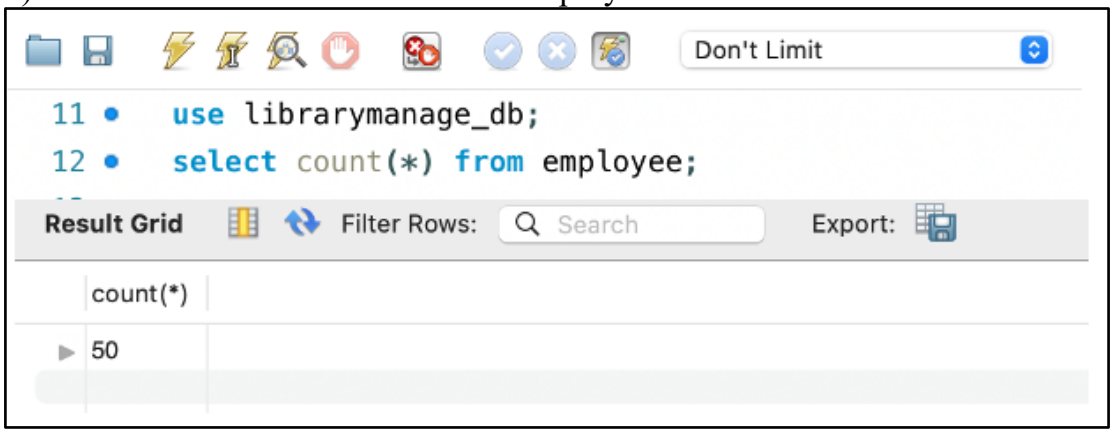
1) The total number of data rows of library table.



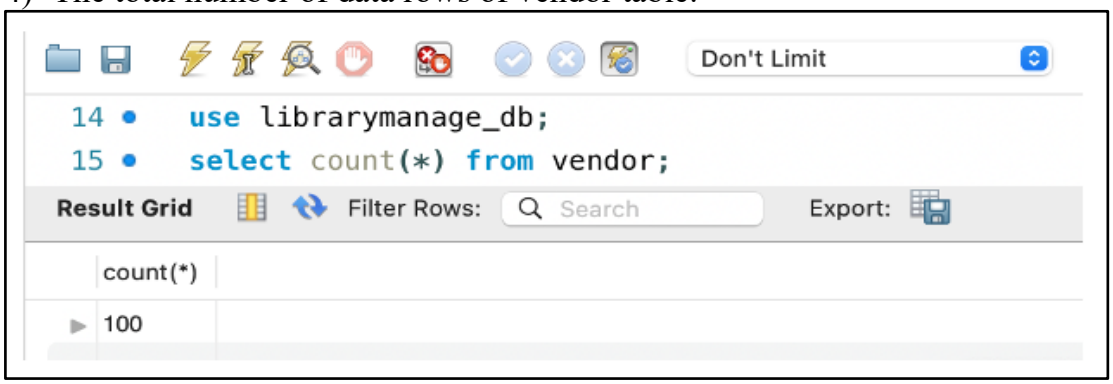
2) The total number of data rows of admin table.



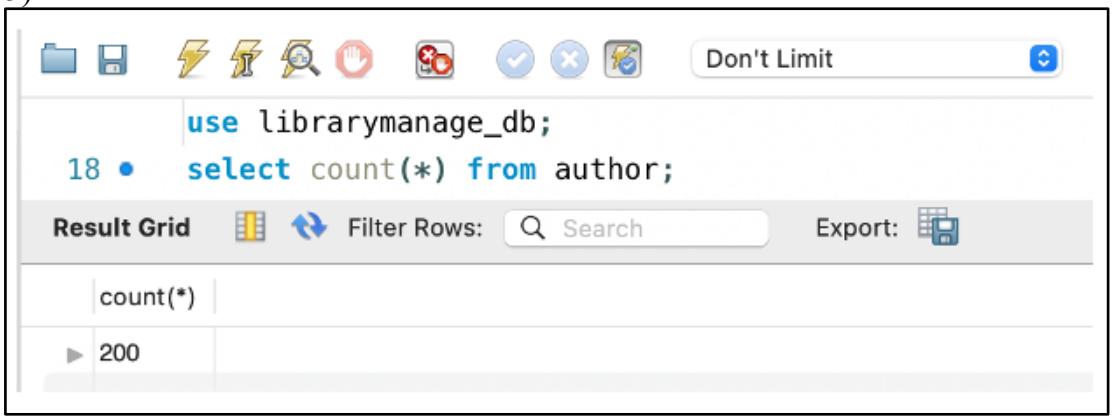
3) The total number of data rows of employee table.



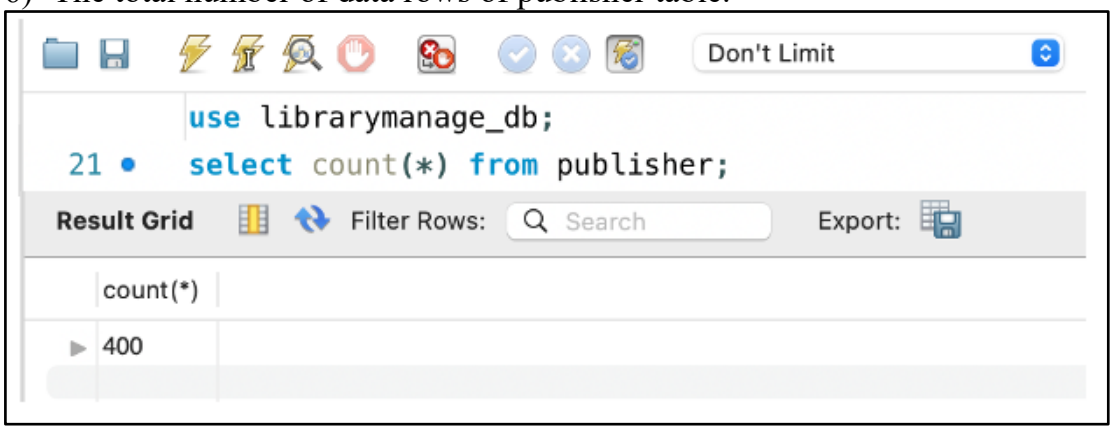
4) The total number of data rows of vendor table.



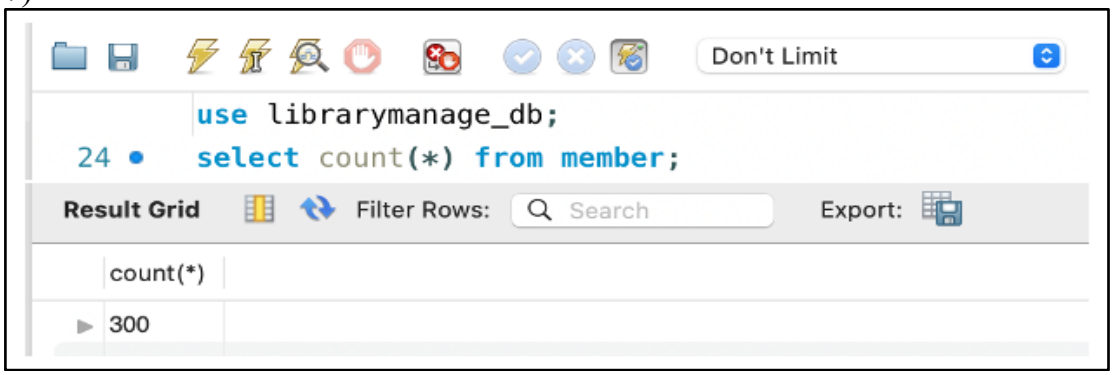
5) The total number of data rows of author table.



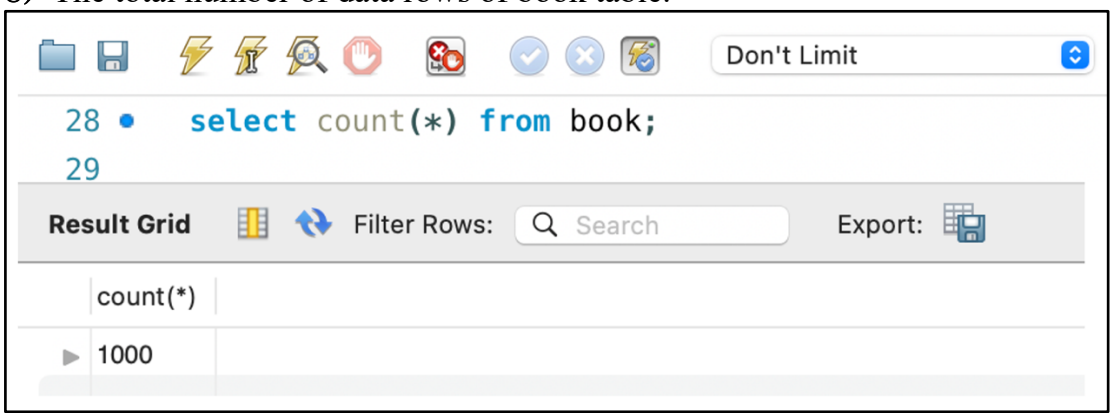
6) The total number of data rows of publisher table.



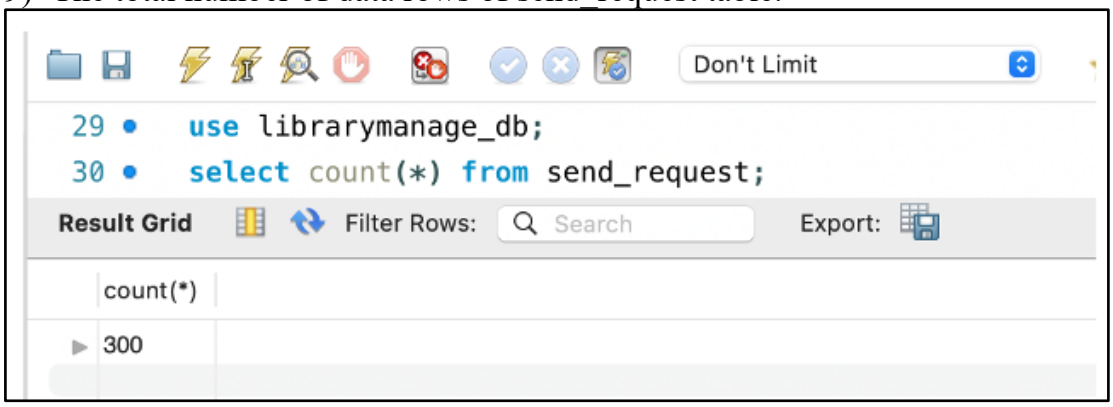
7) The total number of data rows of member table.



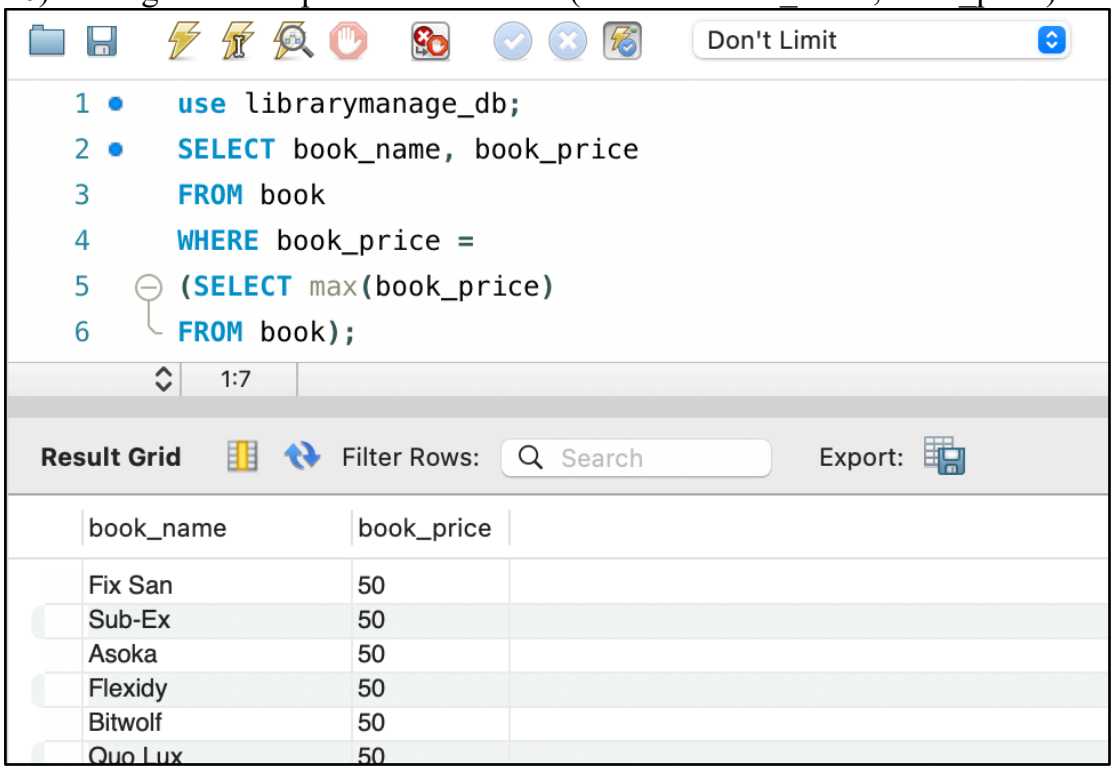
8) The total number of data rows of book table.



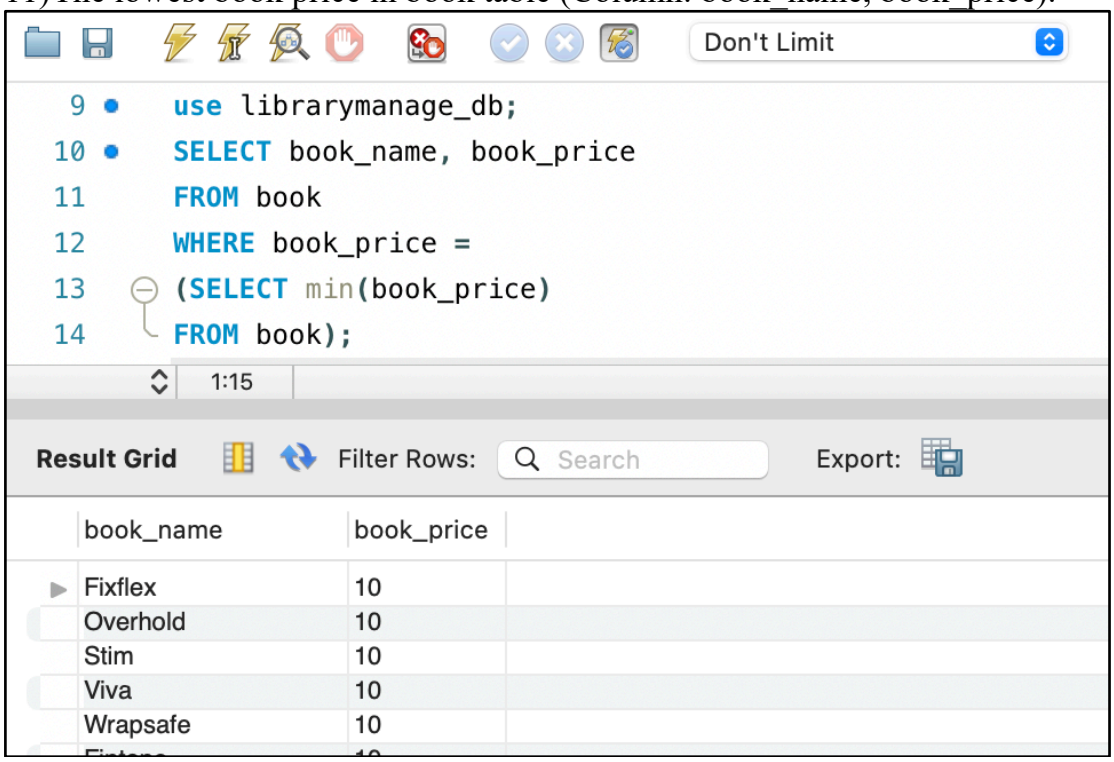
9) The total number of data rows of send_request table.



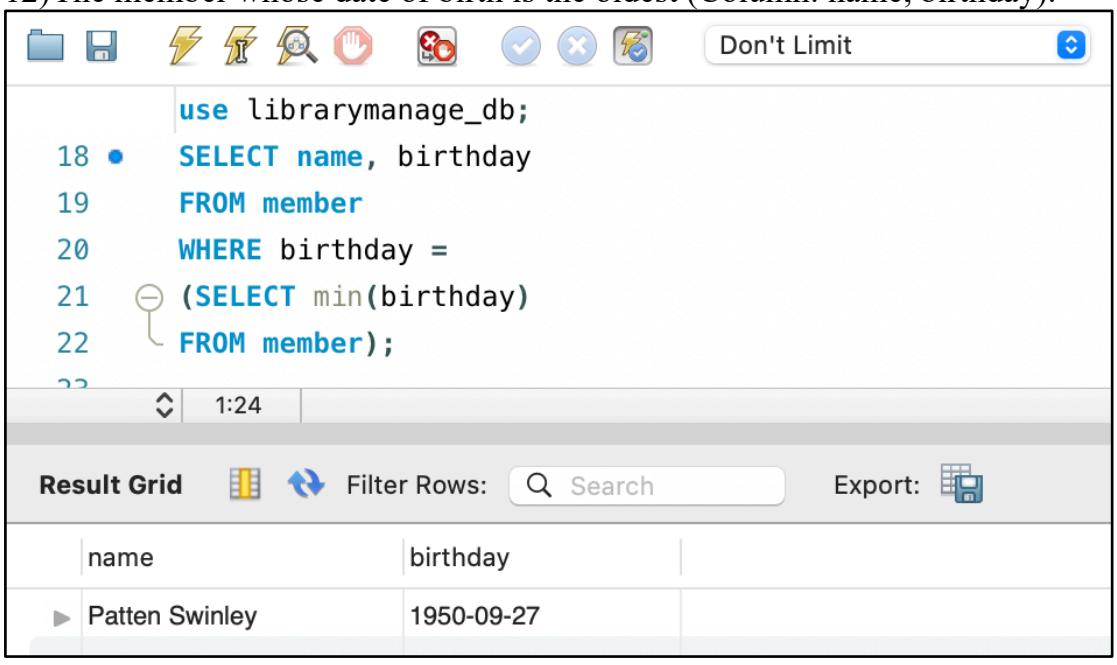
10)The highest book price in book table (Column: book_name, book_price).



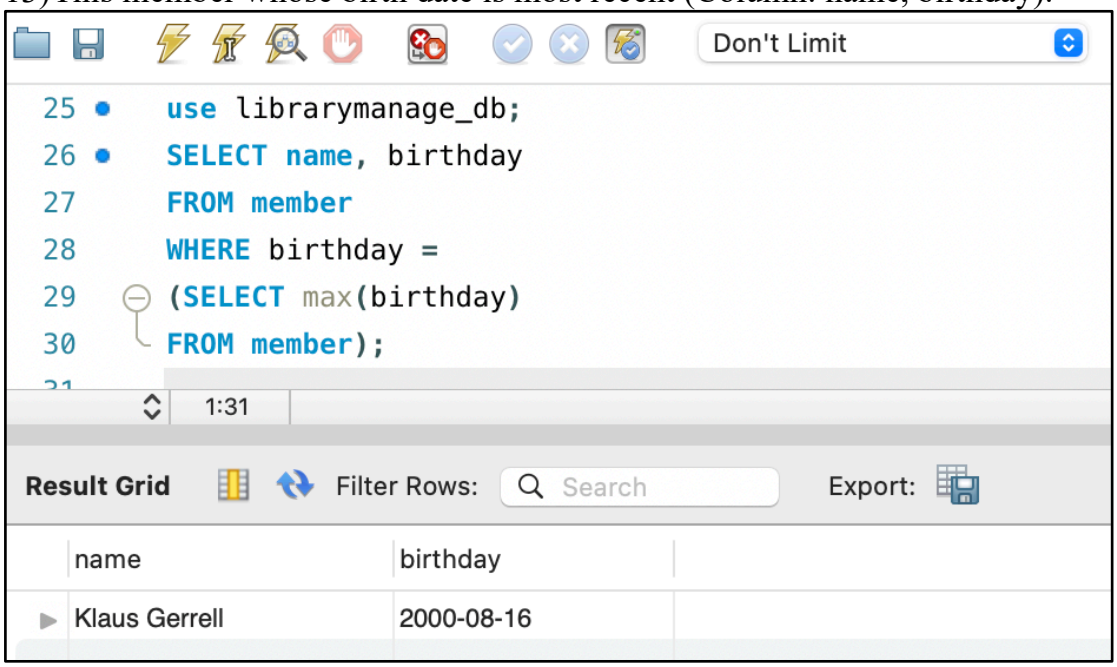
11)The lowest book price in book table (Column: book_name, book_price).



12)The member whose date of birth is the oldest (Column: name, birthday).



13)This member whose birth date is most recent (Column: name, birthday).

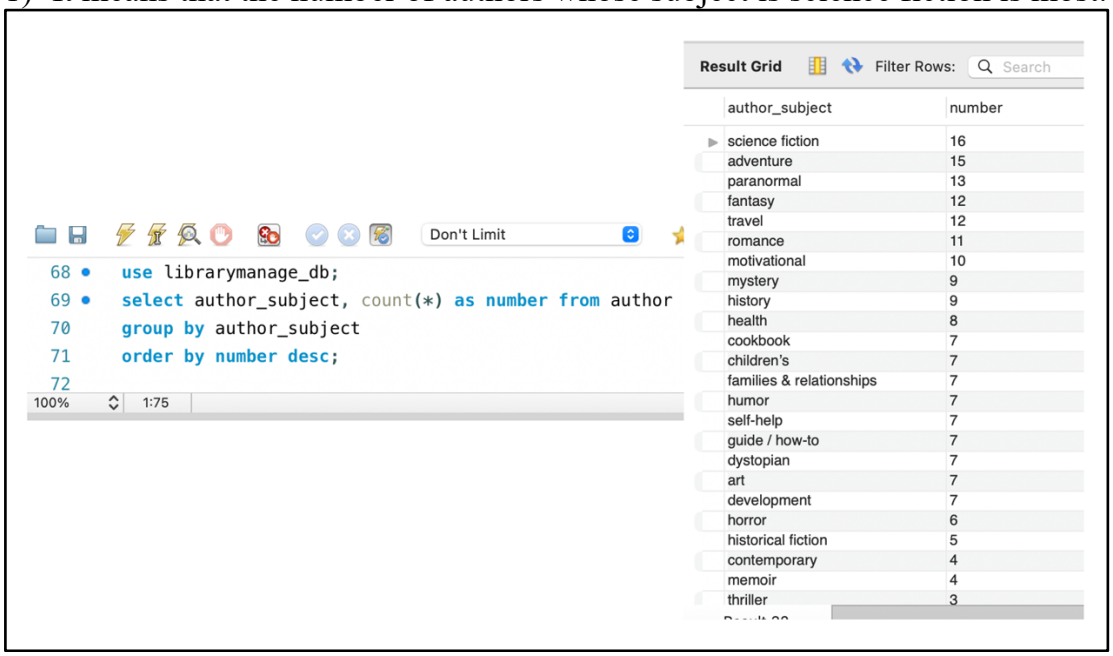


Write a query statement that includes "Order by" and "Group by".

(6 marks)

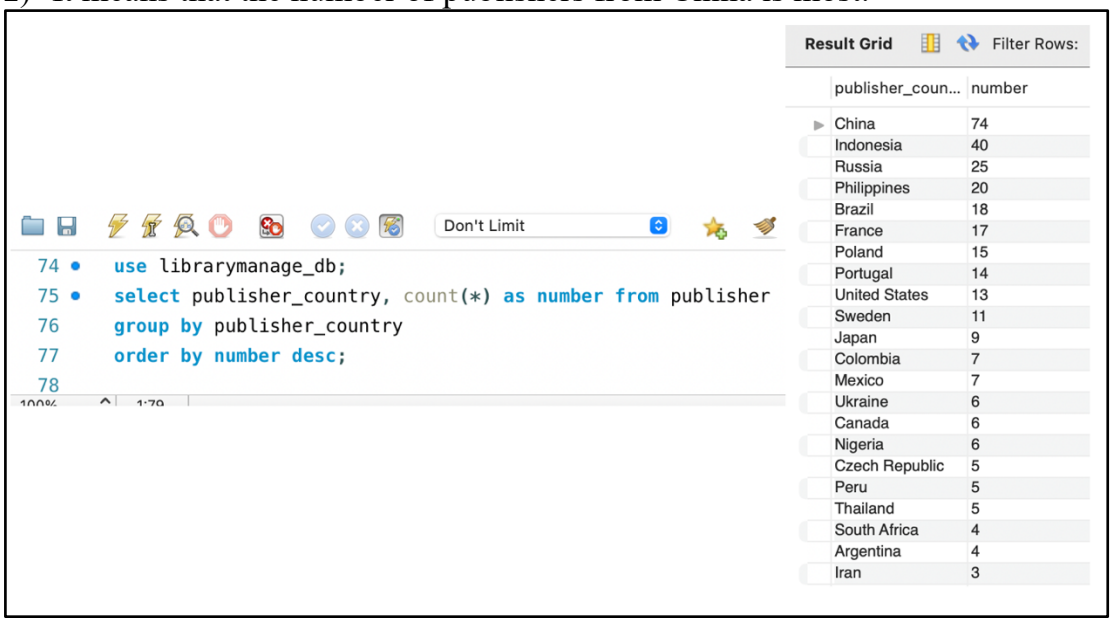
3.2. Query statement that includes "Order by" and "Group by".

1) It means that the number of authors whose subject is science fiction is most.



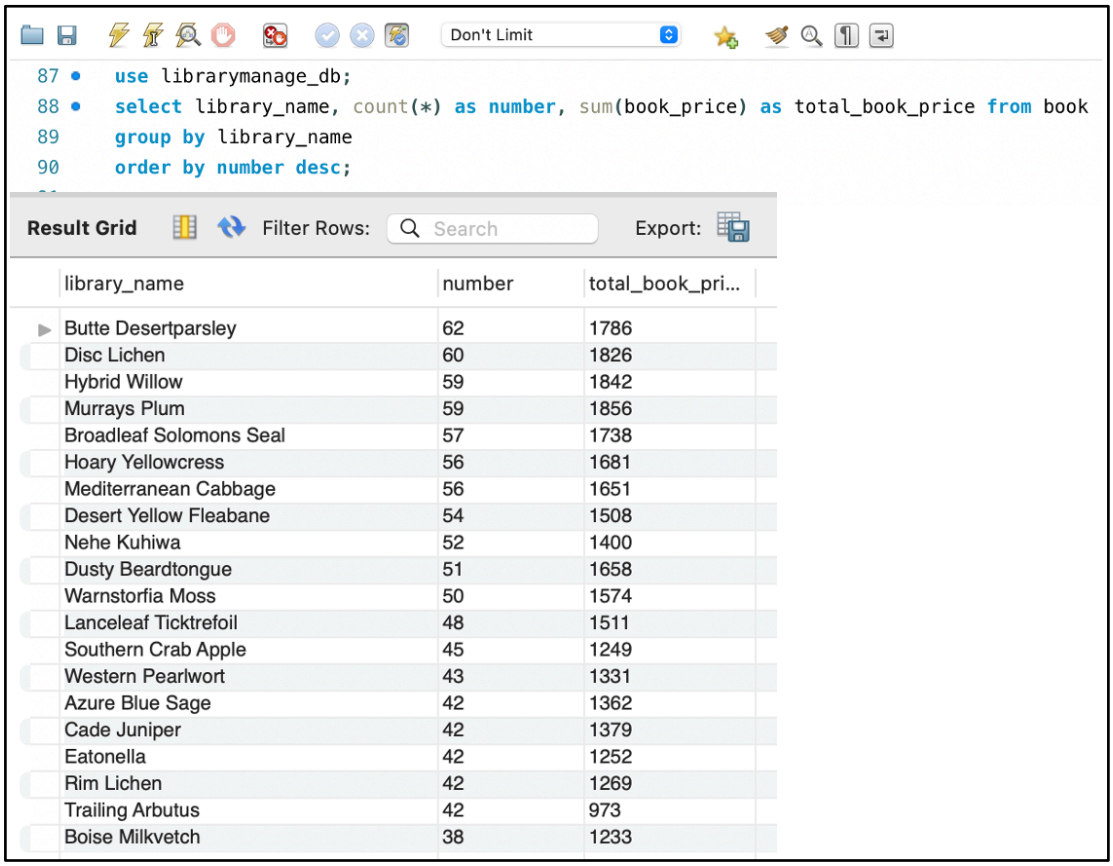
author_subject	number
science fiction	16
adventure	15
paranormal	13
fantasy	12
travel	12
romance	11
motivational	10
mystery	9
history	9
health	8
cookbook	7
children's	7
families & relationships	7
humor	7
self-help	7
guide / how-to	7
dystopian	7
art	7
development	7
horror	6
historical fiction	5
contemporary	4
memoir	4
thriller	3

2) It means that the number of publishers from China is most.



publisher_coun...	number
China	74
Indonesia	40
Russia	25
Philippines	20
Brazil	18
France	17
Poland	15
Portugal	14
United States	13
Sweden	11
Japan	9
Colombia	7
Mexico	7
Ukraine	6
Canada	6
Nigeria	6
Czech Republic	5
Peru	5
Thailand	5
South Africa	4
Argentina	4
Iran	3

3) It means that during 2022, ‘Butte Desertparsley’ library bought most books and total price of those books is 1,786 USD.

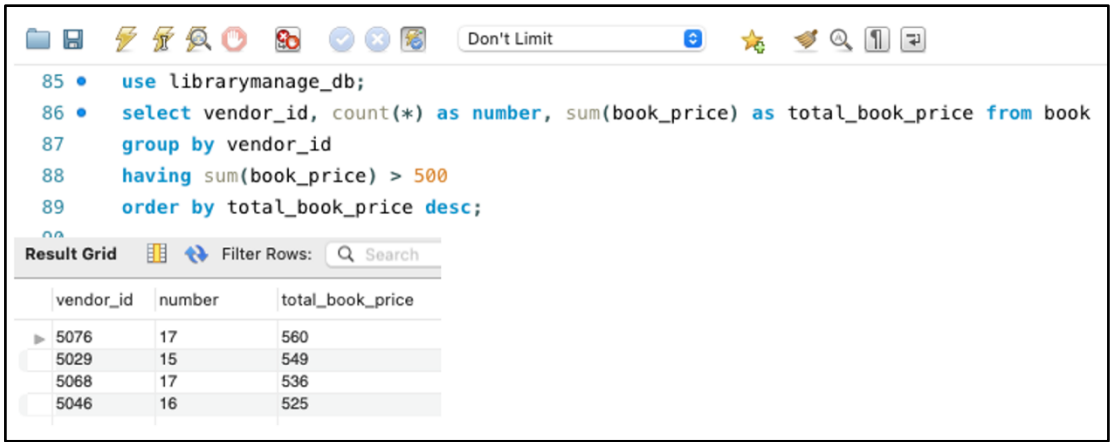


```
87 • use librarymanage_db;
88 • select library_name, count(*) as number, sum(book_price) as total_book_price from book
89   group by library_name
90   order by number desc;
```

Result Grid | Filter Rows: Search | Export:

library_name	number	total_book_pri...
Butte Desertparsley	62	1786
Disc Lichen	60	1826
Hybrid Willow	59	1842
Murrays Plum	59	1856
Broadleaf Solomons Seal	57	1738
Hoary Yellowcress	56	1681
Mediterranean Cabbage	56	1651
Desert Yellow Fleabane	54	1508
Nehe Kuhiwa	52	1400
Dusty Beardtongue	51	1658
Warnstorfia Moss	50	1574
Lanceleaf Ticktrefoil	48	1511
Southern Crab Apple	45	1249
Western Pearlwort	43	1331
Azure Blue Sage	42	1362
Cade Juniper	42	1379
Eatonella	42	1252
Rim Lichen	42	1269
Trailing Arbutus	42	973
Boise Milkvetch	38	1233

4) It means that during 2022, vendor_id(5076) sold most books (17 books) and total price of those books is 560 USD.
Datas that have total price of sold book more than 500 USD are appeared.



```
85 • use librarymanage_db;
86 • select vendor_id, count(*) as number, sum(book_price) as total_book_price from book
87   group by vendor_id
88   having sum(book_price) > 500
89   order by total_book_price desc;
```

Result Grid | Filter Rows: Search

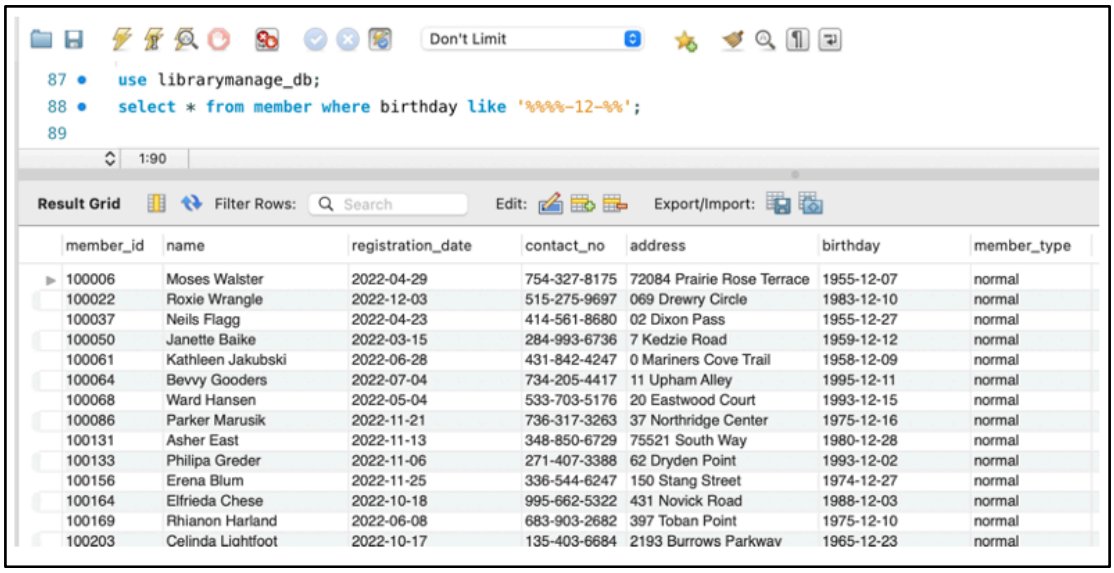
vendor_id	number	total_book_price
5076	17	560
5029	15	549
5068	17	536
5046	16	525

Write a query statement that uses pattern matching (example: customer living in a given street, number of Johns, people with today's birthday…).

(6 marks)

3.3. Query statement that uses pattern matching.

1) Library would like to send message to member whose birth month is December. Whose birth month is December?

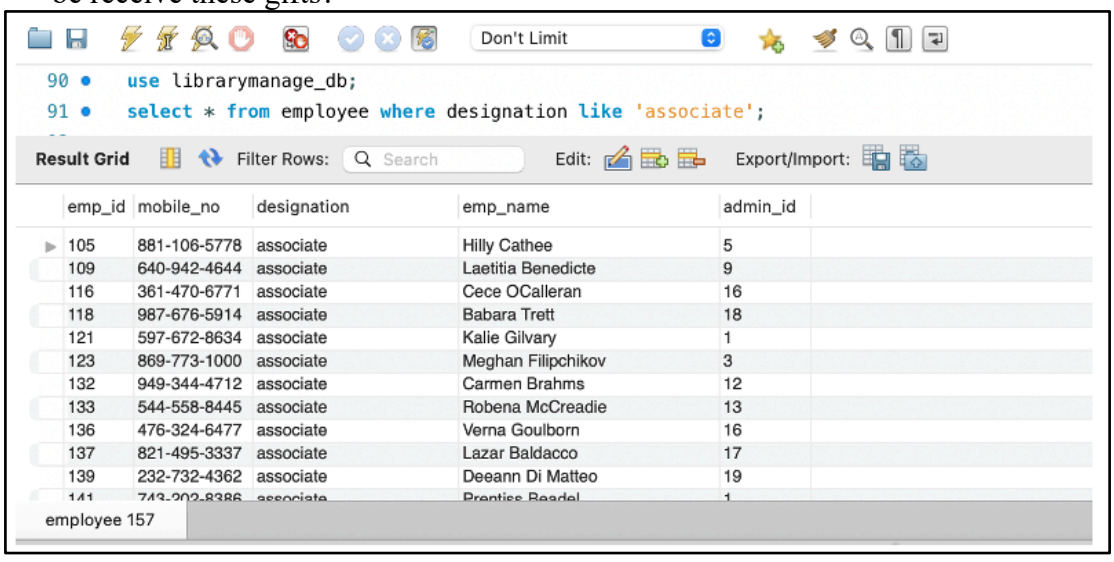


```
87 •  use librarymanage_db;
88 •  select * from member where birthday like '%%%%-12-%%';
89
```

Result Grid | Filter Rows: Search | Edit: | Export/Import:

member_id	name	registration_date	contact_no	address	birthday	member_type
100006	Moses Walster	2022-04-29	754-327-8175	72084 Prairie Rose Terrace	1955-12-07	normal
100022	Roxie Wrangle	2022-12-03	515-275-9697	069 Drewry Circle	1983-12-10	normal
100037	Neils Flagg	2022-04-23	414-561-8680	02 Dixon Pass	1955-12-27	normal
100050	Janette Baike	2022-03-15	284-993-6736	7 Kedzie Road	1959-12-12	normal
100061	Kathleen Jakubski	2022-06-28	431-842-4247	0 Mariners Cove Trail	1958-12-09	normal
100064	Bevvy Gooders	2022-07-04	734-205-4417	11 Upham Alley	1995-12-11	normal
100068	Ward Hansen	2022-05-04	533-703-5176	20 Eastwood Court	1993-12-15	normal
100086	Parker Marusik	2022-11-21	736-317-3263	37 Northridge Center	1975-12-16	normal
100131	Asher East	2022-11-13	348-850-6729	75521 South Way	1980-12-28	normal
100133	Philipa Greder	2022-11-06	271-407-3388	62 Dryden Point	1993-12-02	normal
100156	Erena Blum	2022-11-25	336-544-6247	150 Stang Street	1974-12-27	normal
100164	Elfrieda Chese	2022-10-18	995-662-5322	431 Novick Road	1988-12-03	normal
100169	Rhianon Harland	2022-06-08	683-903-2682	397 Toban Point	1975-12-10	normal
100203	Celinda Lightfoot	2022-10-17	135-403-6684	2193 Burrows Parkway	1965-12-23	normal

2) Administrator would like to prepare some gifts for associate employees. Who will be receive these gifts?



```
90 •  use librarymanage_db;
91 •  select * from employee where designation like 'associate';
```

Result Grid | Filter Rows: Search | Edit: | Export/Import:

emp_id	mobile_no	designation	emp_name	admin_id
105	881-106-5778	associate	Hilly Cathee	5
109	640-942-4644	associate	Laetitia Benedicte	9
116	361-470-6771	associate	Cece OCalleran	16
118	987-676-5914	associate	Babara Trett	18
121	597-672-8634	associate	Kalie Gilvary	1
123	869-773-1000	associate	Meghan Filipchikov	3
132	949-344-4712	associate	Carmen Brahms	12
133	544-558-8445	associate	Robena McCreadie	13
136	476-324-6477	associate	Verna Goulborn	16
137	821-495-3337	associate	Lazar Baldacco	17
139	232-732-4362	associate	Deeann Di Matteo	19
141	743-202-8386	associate	Prentiss Beadel	1

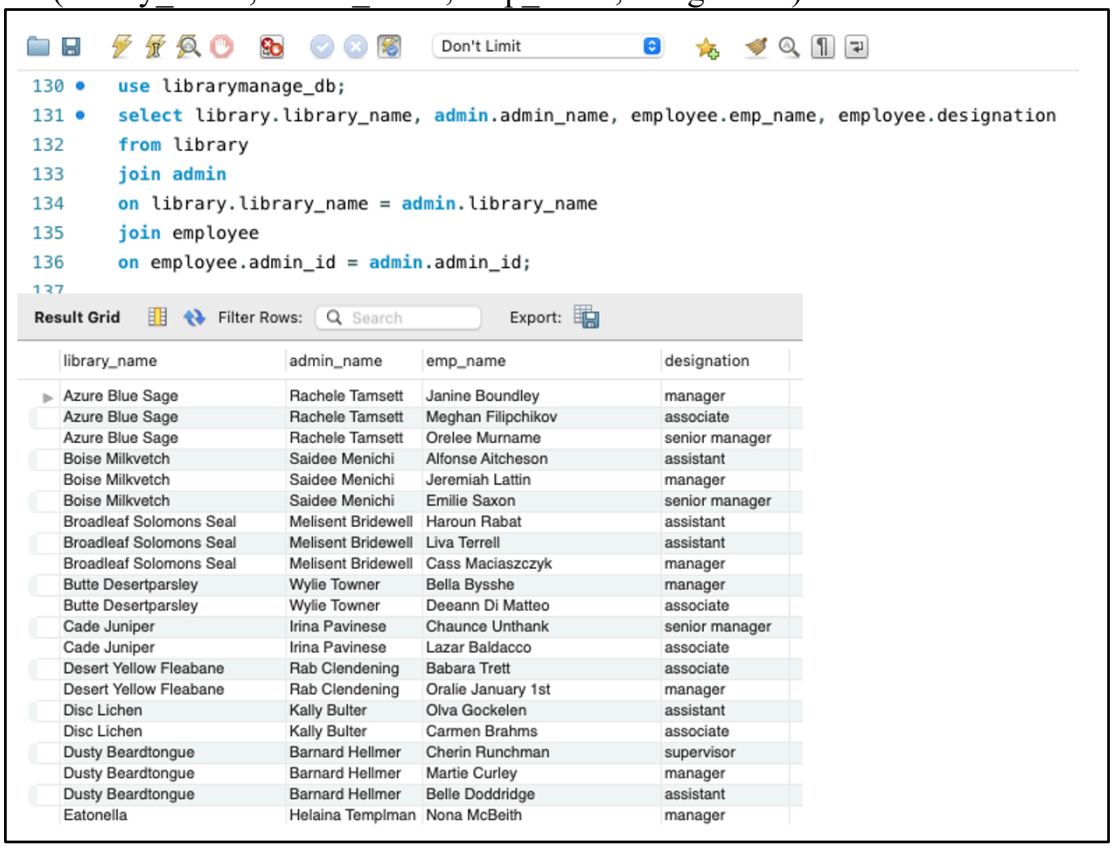
employee 157

Show information from three tables based on criteria of your choice (hint: join).

(6 marks)

3.4. Information from three tables by joins.

1) Join 3 tables (library, admin, and employee) to show column (library_name, admin_name, emp_name, designation).

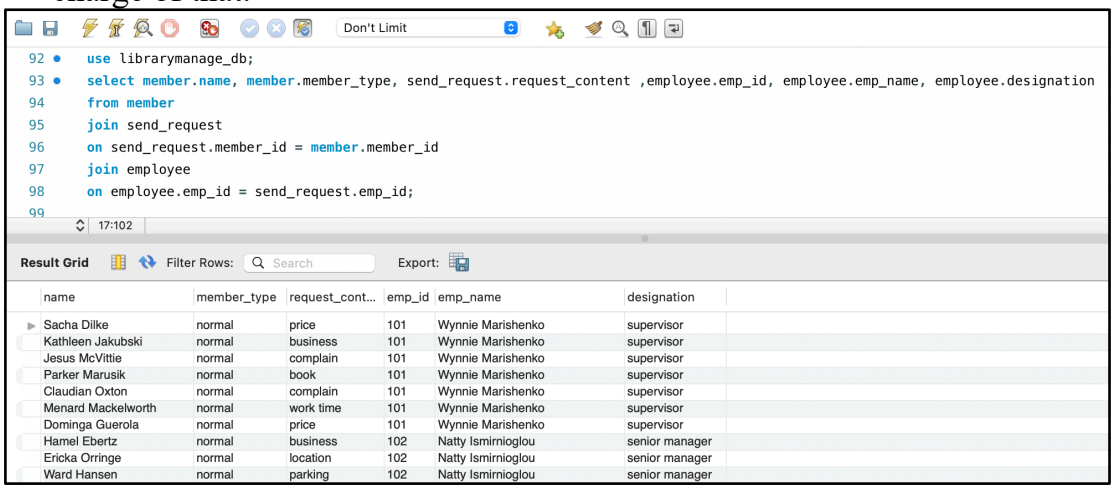


```
130 • use librarymanage_db;
131 • select library.library_name, admin.admin_name, employee.emp_name, employee.designation
132   from library
133   join admin
134   on library.library_name = admin.library_name
135   join employee
136   on employee.admin_id = admin.admin_id;
```

Result Grid — Filter Rows: Search — Export:

library_name	admin_name	emp_name	designation
Azure Blue Sage	Rachele Tamsett	Janine Boundley	manager
Azure Blue Sage	Rachele Tamsett	Meghan Filipchikov	associate
Azure Blue Sage	Rachele Tamsett	Orelee Murname	senior manager
Boise Milkvetch	Saidee Menichi	Alfonse Aitcheson	assistant
Boise Milkvetch	Saidee Menichi	Jeremiah Lattin	manager
Boise Milkvetch	Saidee Menichi	Emilie Saxon	senior manager
Broadleaf Solomons Seal	Melisent Bridewell	Haroun Rabat	assistant
Broadleaf Solomons Seal	Melisent Bridewell	Liva Terrell	assistant
Broadleaf Solomons Seal	Melisent Bridewell	Cass Maciaszczyk	manager
Butte Desertparsley	Wylie Towner	Bella Bysshe	manager
Butte Desertparsley	Wylie Towner	Deeann Di Matteo	associate
Cade Juniper	Irina Pavinese	Chaunce Unthank	senior manager
Cade Juniper	Irina Pavinese	Lazar Baldacco	associate
Desert Yellow Fleabane	Rab Clendening	Babara Trett	associate
Desert Yellow Fleabane	Rab Clendening	Oralie January 1st	manager
Disc Lichen	Kally Bulter	Olva Gockelen	assistant
Disc Lichen	Kally Bulter	Carmen Brahms	associate
Dusty Beardtongue	Barnard Hellmer	Cherin Runchman	supervisor
Dusty Beardtongue	Barnard Hellmer	Martie Curley	manager
Dusty Beardtongue	Barnard Hellmer	Belle Doddridge	assistant
Eatonella	Helaina Templman	Nona McBeith	manager

2) Join 3 tables (library, admin, and employee).
Library would like to check the member request histories including information of member name, request content, and employee's name in charge of that.



```
92 • use librarymanage_db;
93 • select member.name, member.member_type, send_request.request_content ,employee.emp_id, employee.emp_name, employee.designation
94   from member
95   join send_request
96   on send_request.member_id = member.member_id
97   join employee
98   on employee.emp_id = send_request.emp_id;
99
```

17:102

Result Grid — Filter Rows: Search — Export:

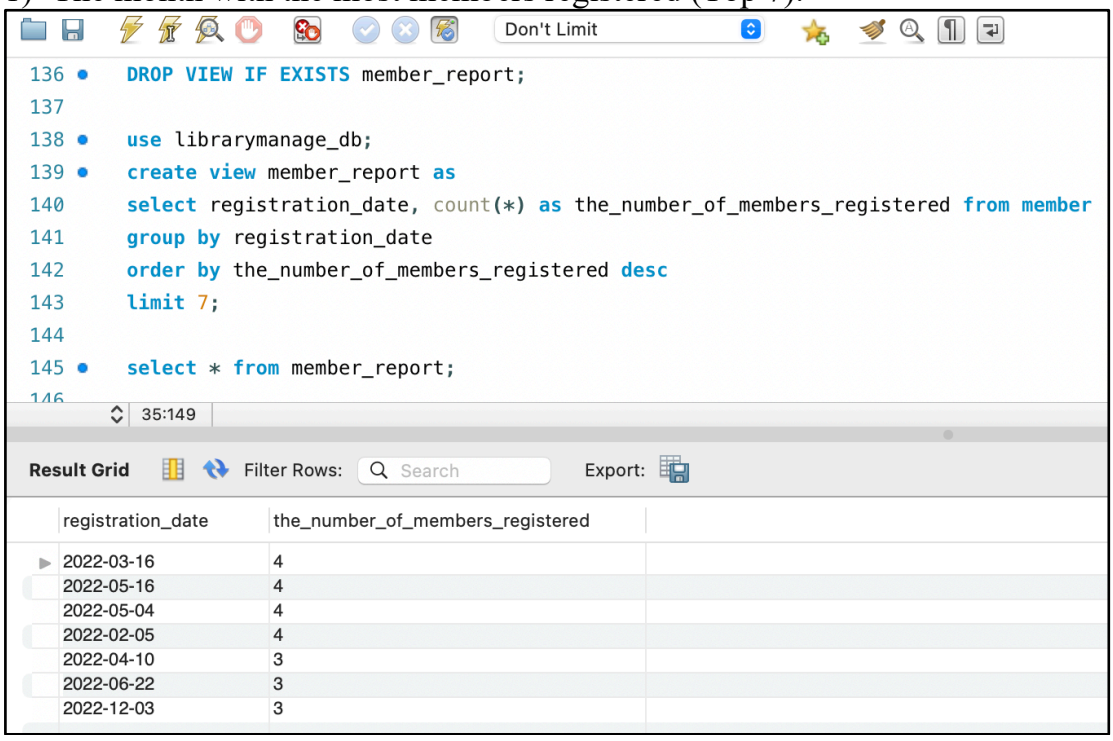
name	member_type	request_cont...	emp_id	emp_name	designation
Sacha Dilke	normal	price	101	Wynnie Marishenko	supervisor
Kathleen Jakubski	normal	business	101	Wynnie Marishenko	supervisor
Jesus McVittie	normal	complain	101	Wynnie Marishenko	supervisor
Parker Marusik	normal	book	101	Wynnie Marishenko	supervisor
Claudian Oxton	normal	complain	101	Wynnie Marishenko	supervisor
Menard Mackelworth	normal	work time	101	Wynnie Marishenko	supervisor
Dominga Guerola	normal	price	101	Wynnie Marishenko	supervisor
Hamel Ebertz	normal	business	102	Natty Ismirnioglou	senior manager
Ericka Orringe	normal	location	102	Natty Ismirnioglou	senior manager
Ward Hansen	normal	parking	102	Natty Ismirnioglou	senior manager

Create a view that includes information from the most frequent seven transactions (customer names or most sold items ...).

(6 marks)

3.5. The most frequent seven transactions.

1) The month with the most members registered (Top 7).

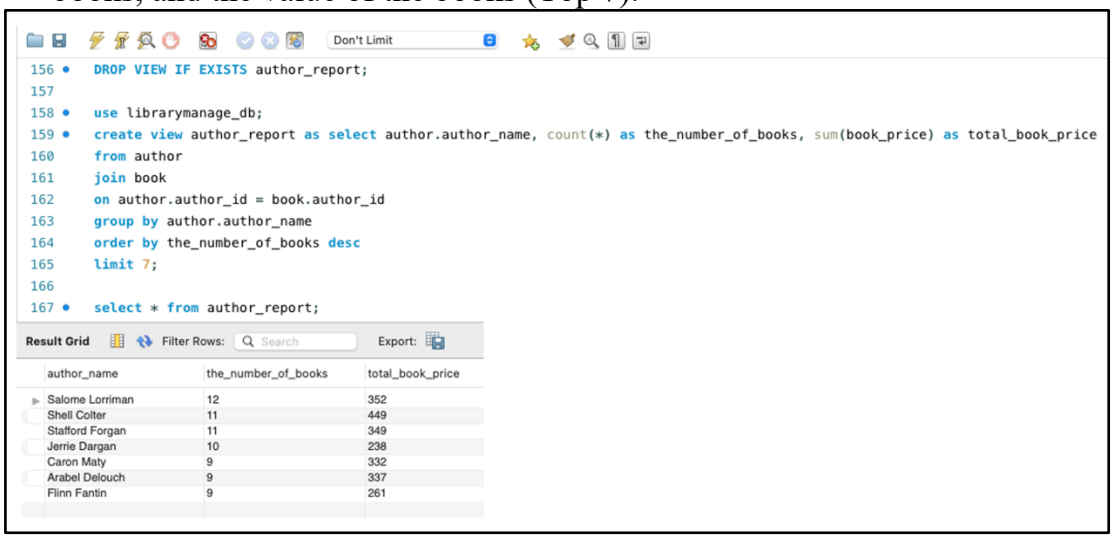


```
136  DROP VIEW IF EXISTS member_report;
137
138  use librarymanage_db;
139  create view member_report as
140  select registration_date, count(*) as the_number_of_members_registered from member
141  group by registration_date
142  order by the_number_of_members_registered desc
143  limit 7;
144
145  select * from member_report;
```

Result Grid — Filter Rows: Search — Export:

registration_date	the_number_of_members_registered
2022-03-16	4
2022-05-16	4
2022-05-04	4
2022-02-05	4
2022-04-10	3
2022-06-22	3
2022-12-03	3

2) The name of the author who wrote the most books, the number of books, and the value of the books (Top 7).

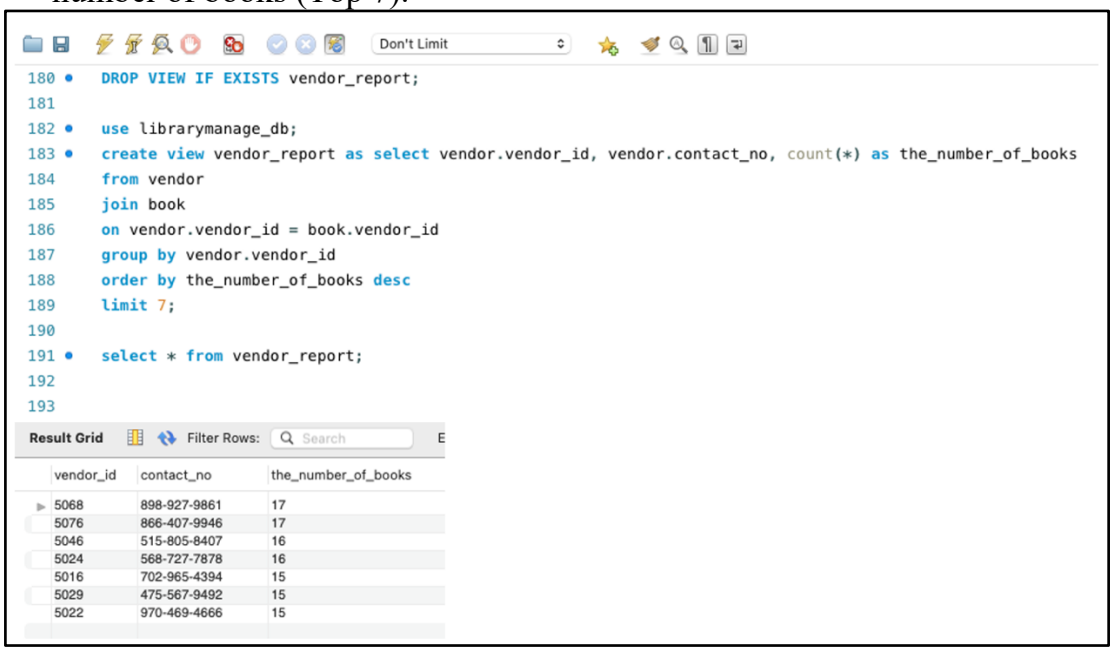


```
156  DROP VIEW IF EXISTS author_report;
157
158  use librarymanage_db;
159  create view author_report as select author.author_name, count(*) as the_number_of_books, sum(book_price) as total_book_price
160  from author
161  join book
162  on author.author_id = book.author_id
163  group by author.author_name
164  order by the_number_of_books desc
165  limit 7;
166
167  select * from author_report;
```

Result Grid — Filter Rows: Search — Export:

author_name	the_number_of_books	total_book_price
Salome Lorriman	12	352
Shell Colter	11	449
Stafford Forgan	11	349
Jerrie Dargan	10	238
Caron Maty	9	332
Arabel Delouch	9	337
Flinn Fantin	9	261

3) Vendor ID and contact number that sold the most books, and the number of books (Top 7).



```
180 •  DROP VIEW IF EXISTS vendor_report;
181
182 •  use librarymanage_db;
183 •  create view vendor_report as select vendor.vendor_id, vendor.contact_no, count(*) as the_number_of_books
184    from vendor
185    join book
186    on vendor.vendor_id = book.vendor_id
187    group by vendor.vendor_id
188    order by the_number_of_books desc
189    limit 7;
190
191 •  select * from vendor_report;
192
193
```

Result Grid — Filter Rows: Search

vendor_id	contact_no	the_number_of_books
5068	898-927-9861	17
5076	866-407-9946	17
5046	515-805-8407	16
5024	568-727-7878	16
5016	702-965-4394	15
5029	475-567-9492	15
5022	970-469-4666	15

Create a set of queries that summarises the annual transactions. For example, if your transaction table is about selling product, you can create queries that:

- *Shows the total number of transactions with corresponding details every month*
- *Shows customer purchase value per month*
- *Shows name of product and number sold each month*

(12marks)

3.6. Set of queries that summarizes the annual transactions.

1) Number of monthly books received (number of books purchased from vendors) and the total cost of monthly book purchases.

```
198 • alter table book add column month int;
199 • update book set month =
200
201   case when book_received_date between '2022-01-01' and '2022-01-31' then 1
202   when book_received_date between '2022-02-01' and '2022-02-28' then 2
203   when book_received_date between '2022-03-01' and '2022-03-31' then 3
204   when book_received_date between '2022-04-01' and '2022-04-30' then 4
205   when book_received_date between '2022-05-01' and '2022-05-31' then 5
206   when book_received_date between '2022-06-01' and '2022-06-30' then 6
207   when book_received_date between '2022-07-01' and '2022-07-31' then 7
208   when book_received_date between '2022-08-01' and '2022-08-31' then 8
209   when book_received_date between '2022-09-01' and '2022-09-30' then 9
210   when book_received_date between '2022-10-01' and '2022-10-31' then 10
211   when book_received_date between '2022-11-01' and '2022-11-30' then 11
212   when book_received_date between '2022-12-01' and '2022-12-31' then 12
213   else 13 end;
214
215 • select month, count(*) as totoal_number, sum(book_price) as total_book_price from book
216   group by month
217   order by month;
```

Result Grid

month	totoal_number	total_book_price
1	79	2406
2	88	2666
3	88	2788
4	61	1845
5	93	2750
6	90	2700
7	85	2628
8	93	2632
9	76	2213
10	80	2542
11	80	2438
12	87	2471

2) Number of members registered as library members per month.

```
224 • alter table member add column month int;
225 • update member set month =
226
227   case when registration_date between '2022-01-01' and '2022-01-31' then 1
228   when registration_date between '2022-02-01' and '2022-02-28' then 2
229   when registration_date between '2022-03-01' and '2022-03-31' then 3
230   when registration_date between '2022-04-01' and '2022-04-30' then 4
231   when registration_date between '2022-05-01' and '2022-05-31' then 5
232   when registration_date between '2022-06-01' and '2022-06-30' then 6
233   when registration_date between '2022-07-01' and '2022-07-31' then 7
234   when registration_date between '2022-08-01' and '2022-08-31' then 8
235   when registration_date between '2022-09-01' and '2022-09-30' then 9
236   when registration_date between '2022-10-01' and '2022-10-31' then 10
237   when registration_date between '2022-11-01' and '2022-11-30' then 11
238   when registration_date between '2022-12-01' and '2022-12-31' then 12
239   else 13 end;
240
241 • select month, count(*) as the_number_of_members_registered from member
242   group by month
243   order by month;
```

Result Grid

month	the_number_of_members_registered
1	28
2	25
3	36
4	22
5	24
6	23
7	24
8	23
9	27
10	25
11	19
12	24

3) Merge above 2 tables.

```
247 •  select book.month, count(*) as the_number_of_books_received, sum(book_price) as total_book_price,
248
249    (select count(*) as the_number_of_members_registered
250    from member where member.month = book.month) as the_number_of_members_registered
251    from book
252    group by month
253    order by month;
```

Result Grid | Filter Rows: Search | Export:

month	the_number_of_books_received	total_book_price	the_number_of_members_registered
1	79	2406	28
2	88	2666	25
3	88	2788	36
4	61	1845	22
5	93	2750	24
6	90	2700	23
7	85	2628	24
8	93	2632	23
9	76	2213	27
10	80	2542	25
11	80	2438	19
12	87	2471	24

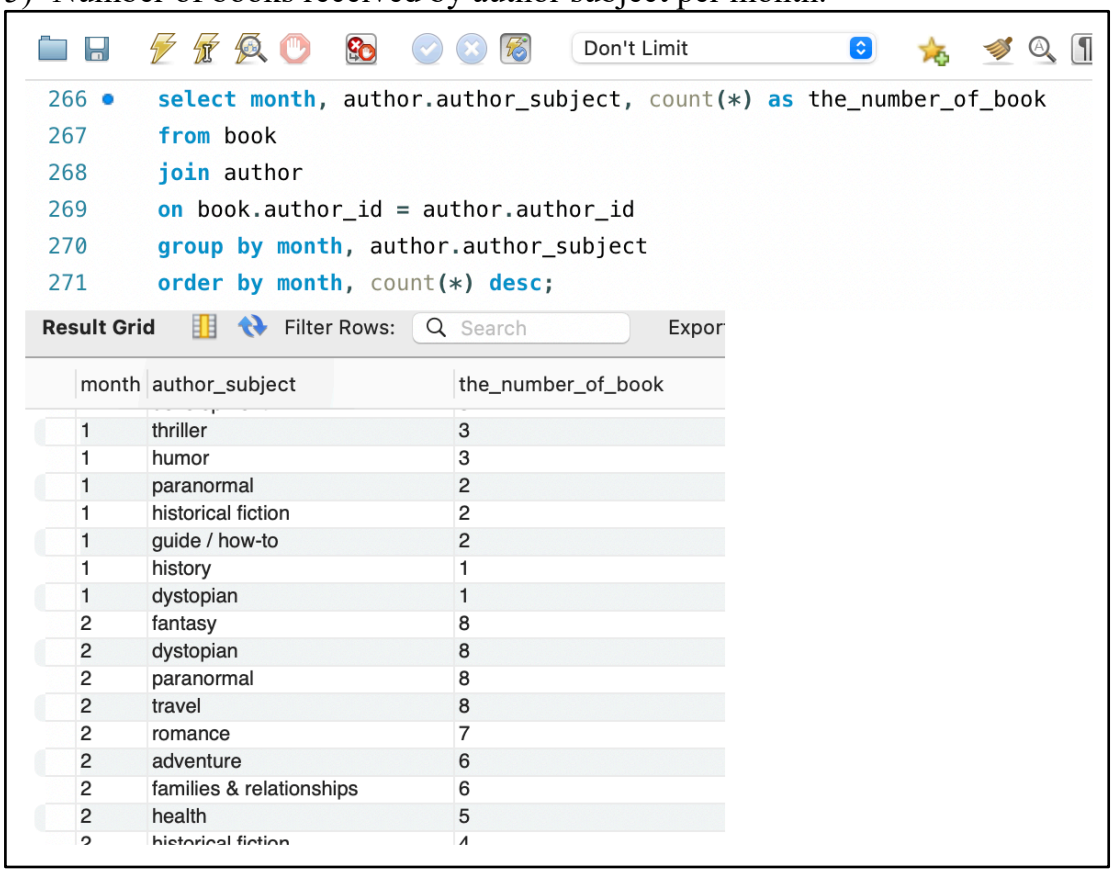
4) Number of books received by author per month.

```
257 •  select month, author.author_name, count(*) as the_number_of_book
258    from book
259    join author
260    on book.author_id = author.author_id
261    group by month, author.author_name
262    order by month, count(*) desc;
```

Result Grid | Filter Rows: Search | Export:

month	author_name	the_number_of_book
1	Candace Muckersie	1
1	Demetre Rafter	1
1	Kathleen Couves	1
1	Orelie Stritton	1
1	Jeffy Fielders	1
1	Barron Earengey	1
1	Kendell Charrier	1
2	Michele Sturmey	4
2	Cal Swatten	3
2	Flinn Fantin	3
2	Jules Harfleet	3
2	Muffin McAndie	2
2	Savina Osichev	2
2	Stormi Roussell	2
2	Sophia Boulsher	2
2	Kalina Baszkiewicz	2
2	Gertie Mulhill	2

5) Number of books received by author subject per month.



```
Don't Limit

266 •  select month, author.author_subject, count(*) as the_number_of_book
267    from book
268    join author
269    on book.author_id = author.author_id
270    group by month, author.author_subject
271    order by month, count(*) desc;
```

Result Grid Filter Rows: Search Expor

month	author_subject	the_number_of_book
1	thriller	3
1	humor	3
1	paranormal	2
1	historical fiction	2
1	guide / how-to	2
1	history	1
1	dystopian	1
2	fantasy	8
2	dystopian	8
2	paranormal	8
2	travel	8
2	romance	7
2	adventure	6
2	families & relationships	6
2	health	5
2	historical fiction	4