Ryerson University

LIBRARY DATABASE MANAGEMENT SYSTEM

Done by

Group - 97

Suchi Patel	**500922373**
Ryan Jean	**500950492**
Rehan Ahmed	**500896674**

Course Code: CPS 510
Course Name: Database Systems
Section - 05

Date of submission - December 03, 2021

Content

1. DBMS for Library Management System

Library is referred to as a collection or organized information and resources, made available to customers for the purpose of borrowing or just as reference. These information and resources are made available to the customers either in physical format (Handed over to the person in an actual building or room) or digitally (virtually) or even both. The aim of this database management system is to provide a user-friendly environment so that customer or library staff can easily perform the tasks/operations in an efficient manner.

As a brief overview, our database application allows users to store the book details and the customer details. Users can interact with the database in the following ways: they can contribute to the database (e.g. search the book/submit a new book information - this would require an appropriate ISBN, Title, Category, Author, Publisher, Status of the book) etc, they can issue the books and take the returns for the same (e.g. put in issue/return id, book name, customer id, issue/return date, & ISBN), and users can also fill in the customer information (e.g. Customer's name and id, books issued to customer, their address and the registration date).

Requirements

Entities

- BOOKS
- BRANCH
- EMPLOYEE
- CUSTOMER
- ISSUE STATUS
- RETURN STATUS

Attributes

- BOOKS
    - ISBN
    - Title

    - Category
    - Author
    - Publisher
    - Rental Price
    - Status

Books Entity: It has ISBN, Title, Category, Author, Publisher, Deposit Amount and Status. ISBN is the Primary Key for Books Entity

- BRANCH
    - Manager id
    - Library Branch id
    - Address
        - Branch No
        - Street
        - City
        - State
        - Postal Code
        - Contact No

Branch Entity: It has Manager id, Library Branch id and Address. Address is the composite attribute of Branch No, Street, City, State, Postal Code and Contact No.
Branch id is the Primary Key for Branch Entity

- CUSTOMER
    - Customer id
    - Books Issued
    - Full Name
    - Address
    - Registration Date

Customer Entity: It has Customer id, Books issued, Full Name, Address and Registration Date.
Customer id is the Primary Key for Customer Entity.

- ISSUE STATUS
    - Issue - Book Name
    - Issue - id
    - Issue - Date
    - ISBN
    - Customer id

Issue Status Entity: It has Issue - Book Name, Issue - id, Issue - Date, ISBN, and Customer id. Issue - id is the Primary Key for Issue Status Entity.

- RETURN STATUS
    - Return - Book Name
    - Return - id
    - Return - Date
    - ISBN
    - Customer id

Return Status Entity: It has Return - Book Name, Return - id, Return - Date, ISBN, and Customer id. Return - id is the Primary Key for Return Status Entity.

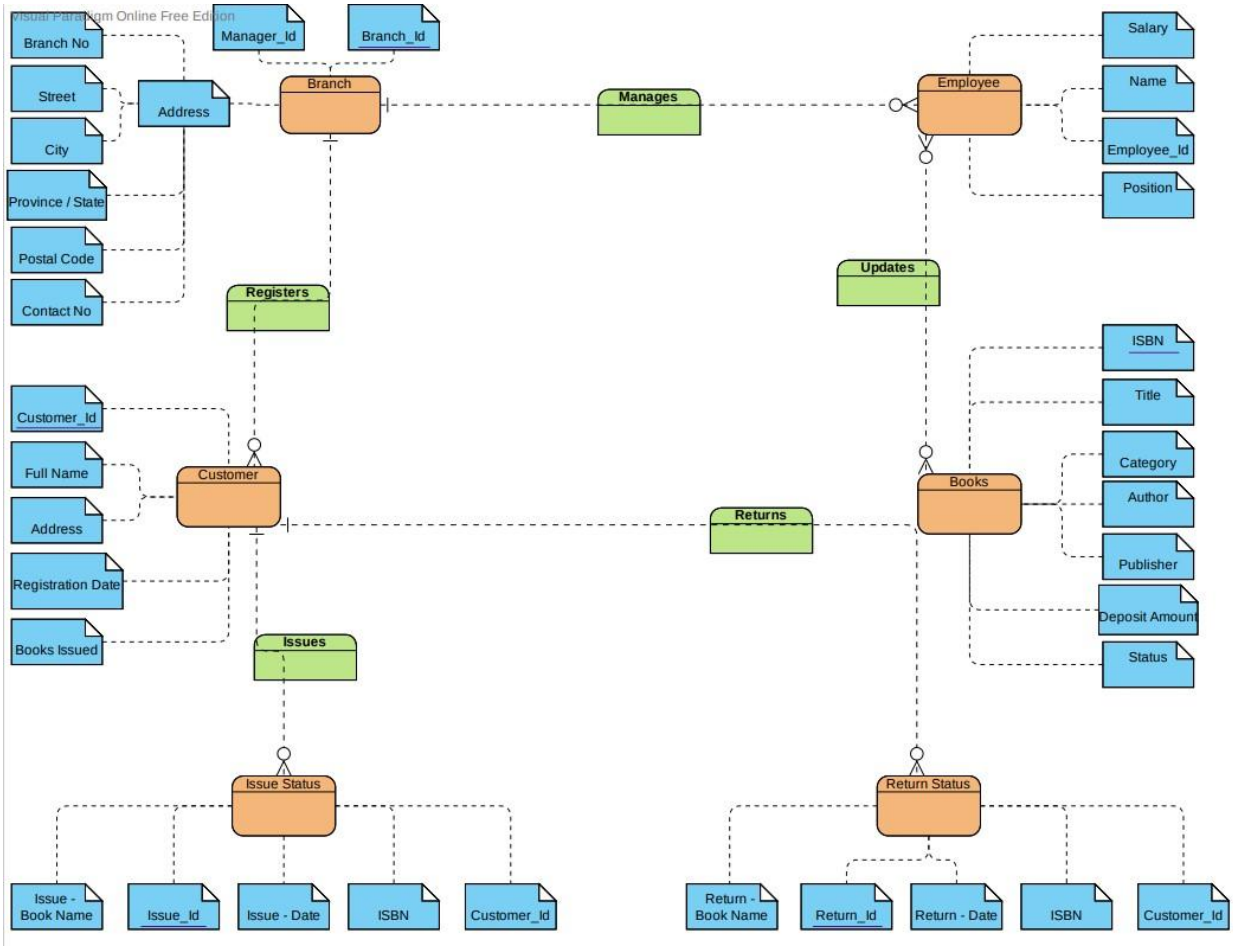
2. ER Model

Entities

- BOOKS
  Primary Key - ISBN
- BRANCH
  Primary Key - Branch id
- EMPLOYEE
  Primary Key - Employee Id
- CUSTOMER
  Primary Key - Customer id
- ISSUE STATUS
  Primary Key - Issue - id
- RETURN STATUS
  Primary Key - Return - id

Relationship between Entities

- Manager manages the Branch (1-N)
  - Multiple branches could be managed by one manager, but a single branch can only be managed by one manager.
- Customer registers in the respective Branch (N-1)
  - Multiple customers can register in one branch, but one branch can be registered by one customer.
- Customer issues Books (1-N)
  - Multiple books could be borrowed by one customer, but one book can only be borrowed by one customer.
- Customer returns Books (1-N)
  - Multiple books can be returned by one customer, but one book can only be returned by one customer who issued the book.
- Employee updates Books (N-N)
  - An employee can update many books simultaneously and each book can have many employees updating it. Hence, it's a many to many relationship.


Visual Paradigm Online Free Edition
Branch No
Manager_Id
Branch_Id
Street
Address
Branch
Manages
Employee
Salary
Name
City
Employee_Id
Province / State
Position
Postal Code
Updates
Contact No
Registers
ISBN
Customer_Id
Title
Full Name
Customer
Books
Category
Author
Address
Returns
Publisher
Registration Date
Deposit Amount
Books Issued
Issues
Status
Issue Status
Return Status
Issue - Book Name
Issue_Id
Issue - Date
ISBN
Customer_Id
Return - Book Name
Return_Id
Return - Date
ISBN
Customer_Id

3. Schema Design

```
CREATE TABLE books (
        ISBN int not null,
        book_title varchar(50) not null,
        category varchar(50) not null,
        rental_price int not null,
        status varchar(50),
        author varchar(50) not null,
        publisher varchar(50) not null,
        primary key(ISBN)
);
```

```
SQL> describe books;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ISBN                                      NOT NULL NUMBER(38)
 BOOK_TITLE                                NOT NULL VARCHAR2(50)
 CATEGORY                                  NOT NULL VARCHAR2(50)
 RENTAL_PRICE                              NOT NULL NUMBER(38)
 STATUS                                             VARCHAR2(50)
 AUTHOR                                    NOT NULL VARCHAR2(50)
 PUBLISHER                                 NOT NULL VARCHAR2(50)
```

```
CREATE TABLE employee (
        employ_id int not null,
        employ_name varchar(50) not null,
        position varchar(30) not null,
        salary int not null,
        primary key(employ_id)
);
```

```
SQL> describe employee
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPLOY_ID                                 NOT NULL NUMBER(38)
 EMPLOY_NAME                               NOT NULL VARCHAR2(50)
 POSITION                                  NOT NULL VARCHAR2(30)
 SALARY                                    NOT NULL NUMBER(38)
```

```
CREATE TABLE customer (
        customer_id int not null,
        customer_name varchar(50),
        customer_address varchar(100) not null,
        registration_date date not null,
        primary key(customer_id)
);
```

```
SQL> describe customer
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL NUMBER(38)
 CUSTOMER_NAME                                      VARCHAR2(50)
 CUSTOMER_ADDRESS                          NOT NULL VARCHAR2(100)
 REGISTRATION_DATE                         NOT NULL DATE
```

```
CREATE TABLE branch (
        branch_no int not null,
        manager_id int not null,
        branch_address varchar(100) not null,
        contact_no int not null,
        primary key(branch_no)
);
```

```
SQL> describe branch
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BRANCH_NO                                 NOT NULL NUMBER(38)
 MANAGER_ID                                NOT NULL NUMBER(38)
 BRANCH_ADDRESS                            NOT NULL VARCHAR2(100)
 CONTACT_NO                                NOT NULL NUMBER(38)
```

```
CREATE TABLE issue_status (
        issue_id int not null,
        issued_cust int not null,
        issued_book_name varchar(50) not null,
         issue_date date  not null, isbn_book int not null,
         primary key(issue_id),
        foreign key(isbn_book) references BOOKS(ISBN),
```

```
        foreign key(issued_cust) references customer(customer_id)
);
```

```
SQL> describe issue_status
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ISSUE_ID                                  NOT NULL NUMBER(38)
 ISSUED_CUST                               NOT NULL NUMBER(38)
 ISSUED_BOOK_NAME                          NOT NULL VARCHAR2(50)
 ISSUE_DATE                                NOT NULL DATE
 ISBN_BOOK                                 NOT NULL NUMBER(38)
```

```
CREATE TABLE return_status (
        return_id int not null,
        return_cust int not null,
        returned_book_name varchar(50) not null,
        return_date date not null, isbn_book2 int not null,
        primary key(return_id),
        foreign key(isbn_book2) references BOOKS(ISBN),
        foreign key(return_cust) references customer(customer_id)
);
```

```
SQL> describe return_status
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RETURN_ID                                 NOT NULL NUMBER(38)
 RETURN_CUST                               NOT NULL NUMBER(38)
 RETURNED_BOOK_NAME                        NOT NULL VARCHAR2(50)
 RETURN_DATE                               NOT NULL DATE
 ISBN_BOOK2                                NOT NULL NUMBER(38)
```

4. Part 1: Demo of Designing Views/Simple Queries

/* View Tables */

Select *from books;

	ISBN	book_title	category	rental_price	status	author	publisher
1	-822300	The_Hobbit	Fantasy_fiction	10	Available	JRR_Tolkien	George_Allen_and_Unwin
2	-451599	Don_Quixote	Adventure_fiction	8	Available	Miguel_de_Cervantes	Penguin_Books
3	-179886	A_Tale_of_Two_Cities	Historical_fiction	6	Available	Charles_Dickens	Penguin_Books
4	-147451	Real	Adventure_fiction	8	Available	Ryan_Jean	Penguin_Books
5	-109425	The_lord_of_the_Rings	Fantasy_fiction	1	Not_Available	JRR_Tolkien	Allen_and_Unwin

Select *from employee;

	employ_id	employ_name	position	salary
1	1	Hani Asim	Branch Manager	50000
2	7	John Proctor	Library Staff	20000
3	13	M.D. Luffy	Library Staff	20000
4	34	Hakeem Olajuwon	Library Security	35000
5	897985	Joe Goldberg	Library Staff	20000

Select *from customer;

	customer_id	customer_name	customer_address	registration_date
1	100	John Smith	17 Zane Lane	2021-07-22
2	150	Lola Tran	30 Resident Street	2021-07-23
3	200	Raul Garcia	100 Auberg Court	2021-07-24
4	500	Alex Ien	230 Victoria Place	2021-07-25

Select *from branch;

	branch_no	manager_id	branch_address	contact_no
1	250	450	29 Divon Court	-9349
2	499	150	39 Private Drive	-4022
3	750	100	103 Zane Lane	-2552
4	1000	50	123 Chester Street	-3785

Select *from issue_status;

	issue_id	issued_cust	issued_book_name	issue_date	isbn_book
1	51	100	Don_Quixote	2021-08-25	-451599
2	52	150	A_Tale_of_Two_Cities	2021-08-26	-179886
3	53	200	The_lord_of_the_Rings	2021-08-27	-109425
4	55	500	Real	2021-08-28	-147451

Select *from return_status;

	return_id	return_cust	returned_book_name	return_date	isbn_book2
1	61	100	Don_Quixote	2021-09-25	-451599
2	62	150	A_Tale_of_Two_Cities	2021-09-26	-179886
3	63	200	The_lord_of_the_Rings	2021-09-27	-109425
4	64	500	Real	2021-09-28	-147451

/* Test Queries */

/* Books */

```
/* Finds all books from a given publisher  */
SELECT
        isbn,
        book_title,
        category,
        rental_price,
        status,
        author,
        publisher
FROM
        books
WHERE
        publisher = 'Penguin_Books';
```

	isbn	book_title	category	rental_price	status	author	publisher
1	-451599	Don_Quixote	Adventure_fiction	8	Available	Miguel_de_Cervantes	Penguin_Books
2	-179886	A_Tale_of_Two_Cities	Historical_fiction	6	Available	Charles_Dickens	Penguin_Books
3	-147451	Real	Adventure_fiction	8	Available	Ryan_Jean	Penguin_Books

```
/* Finds all available books for rental */
SELECT
        isbn,
        book_title,
        category,
        rental_price,
        status,
        author,
        publisher
FROM
        books
WHERE
        status = 'Available';
```

	isbn	book_title	category	rental_price	status	author	publisher
1	-822300	The_Hobbit	Fantasy_fiction	10	Available	JRR_Tolkien	George_Allen_and_Unwin
2	-451599	Don_Quixote	Adventure_fiction	8	Available	Miguel_de_Cervantes	Penguin_Books
3	-179886	A_Tale_of_Two_Cities	Historical_fiction	6	Available	Charles_Dickens	Penguin_Books
4	-147451	Real	Adventure_fiction	8	Available	Ryan_Jean	Penguin_Books

---

/* Employee */

```
/* Displays list of all employees with employee numbers listed from smallest to largest */
SELECT
        employ_name,
        employ_id,
        position,
        salary

FROM
         Employee
ORDER BY employ_id  ASC;
```

	Employ_name	Employ_id	Position	salary
1	Hani Asim	1	Branch Manager	50000
2	John Proctor	7	Library Staff	20000
3	M.D. Luffy	13	Library Staff	20000
4	Hakeem Olajuwon	34	Library Security	35000
5	Joe Goldberg	897985	Library Staff	20000

/* Customer */

```
/* Selects customer characteristics where customer_id is greater than 100 indicating
newer customers*/
SELECT
        customer_id,
        customer_name,
        customer_address,
        registration_date
FROM
        CUSTOMER
Where
        customer_id>100;
```

	customer_id	customer_name	customer_address	registration_date
1	150	Lola Tran	30 Resident Street	2021-07-23
2	200	Raul Garcia	100 Auberg Court	2021-07-24
3	500	Alex Ien	230 Victoria Place	2021-07-25

```
/* Selects distinct results from the customer data table where the customer_id=150 */
SELECT Distinct
        customer_id,
        customer_name,
        customer_address,
        registration_date
FROM
        customer
Where
        customer_id=150;
```

	customer_id	customer_name	customer_address	registration_date
1	150	Lola Tran	30 Resident Street	2021-07-23

```
/* Selects the customer characteristics and uses keyword ORDER BY to sort by
customer_id in descending order */
SELECT Distinct
        customer_id,
        customer_name,
        customer_address,
        registration_date
FROM
        customer
ORDER BY Customer_id DESC;
```

	customer_id	customer_name	customer_address	registration_date
1	500	Alex Ien	230 Victoria Place	2021-07-25
2	200	Raul Garcia	100 Auberg Court	2021-07-24
3	150	Lola Tran	30 Resident Street	2021-07-23
4	100	John Smith	17 Zane Lane	2021-07-22

/* Branch */

```
/* Selects branch characteristics where the branch_no identifier is greater than or equal to
250 and the manager_id is greater than 100  */
SELECT
        branch_no,
```

```
        manager_id,
        branch_address,
        contact_no
FROM
        branch
WHERE
        branch_no>=250
        and manager_id>100;
```

	branch_no	manager_id	branch_address	contact_no
1	250	450	29 Divon Court	-9349
2	499	150	39 Private Drive	-4022

```
/* Selects characteristics of branches and uses ORDER BY keyword to sort by
manager_id in ascending order*/
SELECT
        branch_no,
        manager_id,
        branch_address,
        contact_no
FROM
        branch
        ORDER BY manager_id ASC;
```

	branch_no	manager_id	branch_address	contact_no
1	1000	50	123 Chester Street	-3785
2	750	100	103 Zane Lane	-2552
3	499	150	39 Private Drive	-4022
4	250	450	29 Divon Court	-9349

```
/* Selects branches with branch number>500 (newer branches) with distinct keyword to
list different results */
SELECT DISTINCT
        branch_no,
        manager_id,
        branch_address,
        contact_no
FROM
        branch
Where
        branch_no>500;
```

	branch_no	manager_id	branch_address	contact_no
1	750	100	103 Zane Lane	-2552
2	1000	50	123 Chester Street	-3785

/* Issue Status */

```
/* Identify all the distinct book name and date issued */
SELECT DISTINCT  issued_book_name, issue_date
FROM  issue_status;
```

	issued_book_name	issue_date
1	A_Tale_of_Two_Cities	2021-08-26
2	Don_Quixote	2021-08-25
3	Real	2021-08-28
4	The_lord_of_the_Rings	2021-08-27

/* Return Status */

```
/* List the return dates and number of books that were returned on that date */
SELECT return_date,count(*) as returned_book_name
FROM return_status
GROUP BY return_date
```

	return_date	returned_book_name
1	2021-09-25	1
2	2021-09-26	1
3	2021-09-27	1
4	2021-09-28	1

4.1. Part 2: Complex Queries

**** VIEW 1 FOR CUSTOMER TABLE ****

```
CREATE VIEW library_members AS
SELECT CUSTOMER_NAME, CUSTOMER_ID
FROM CUSTOMER
ORDER BY CUSTOMER_ID ASC;
SELECT * FROM library_members;
```

CUSTOMER_NAME	CUSTOMER_ID
John Smith	100
Lola Tran	150
Raul Garcia	200
Alex Ien	500

**** VIEW 2 FOR issue_status TABLE ****

```
CREATE VIEW ISSUED_BOOKS AS
SELECT ISSUED_CUST,ISSUED_BOOK_NAME
FROM issue_status
ORDER BY ISSUED_CUST ASC;
SELECT ISSUED_CUST ,
ISSUED_BOOK_NAME   FROM issued_books;
```

ISSUED_CUST	ISSUED_BOOK_NAME
100	Don_Quixote
150	A_Tale_of_Two_Cities
200	The_lord_of_the_Rings
500	Real

**** JOIN 1 FOR issue_status and customer TABLE ****

```
SELECT CUSTOMER.CUSTOMER_ID, CUSTOMER_NAME
FROM  issue_status, CUSTOMER
WHERE issue_status.ISSUED_CUST = CUSTOMER.CUSTOMER_ID,
ORDER BY issue_status.ISSUED_CUST ASC;
```

CUSTOMER_ID	CUSTOMER_NAME
100	John Smith
150	Lola Tran
200	Raul Garcia
500	Alex Ien

**** VIEW 3 FOR BOOKS TABLE****

```
CREATE VIEW adventure_category AS
SELECT BOOK_TITLE, RENTAL_PRICE, AUTHOR
FROM BOOKS
WHERE CATEGORY = 'Adveture_fiction' AND STATUS = 'Available';

SELECT * FROM adventure_category;
```

BOOK_TITLE	RENTAL_PRICE	AUTHOR
Real	8	Ryan_Jean
Don_Quixote	8	Miguel_de_Cervantes

**** VIEW 4 FOR EMPLOYEE TABLE ****

```
CREATE VIEW staff_members AS
SELECT EMPLOY_NAME, EMPLOY_ID
FROM EMPLOYEE
WHERE POSITION = 'Library Staff'
ORDER BY EMPLOY_ID ASC;

SELECT * FROM staff_members;
```

EMPLOY_NAME	EMPLOY_ID
John Proctor	7
M.D. Luffy	13
Joe Goldberg	897985

**** QUERY 1 FOR EMPLOYEE TABLE ****

SELECT CAST(AVG(EMPLOYEE.SALARY) AS INT) AS avg_salary FROM EMPLOYEE;

```
AVG_SALARY
----------
     29000
```

**** QUERY 2 FOR BOOKS TABLE**

SELECT COUNT(ISBN), AUTHOR
FROM BOOKS
GROUP BY AUTHOR;

```
COUNT(ISBN) AUTHOR
----------- --------------------------------------------------
          1 Charles_Dickens
          1 Ryan_Jean
          2 JRR_Tolkien
          1 Miguel_de_Cervantes
```

5. Demonstration of advanced queries By Unix shell Implementation

Note: Below listed source code already submitted under Assignment 5 Submission folder.
A5Shell_Menue_Template.sh
create_tables.sh
drop_tables.sh
populate_tables.sh
queries.sh

We created a general bash script for this assignment as below to create, drop, populate & query the tables.

~ % vi A5Shell_Menue_Template.sh

```
#!/bin/sh

MainMenu()
{
   while [ "$CHOICE" != "START" ]
   do
        clear
        echo "================================================================="
        echo "|                    Oracle All Inclusive Tool                  |"
        echo "|           Main Menu - Select Desired Operation(s):            |"
        echo "|       <CTRL-Z Anytime to Enter Interactive CMD Prompt>        |"
        echo "================================================================="
        echo " $IS_SELECTEDM M)  View Manual"
        echo " "
        echo " $IS_SELECTED1 1)  Drop Tables"
        echo " $IS_SELECTED2 2)  Create Tables"
        echo " $IS_SELECTED3 3)  Populate Tables"
        echo " $IS_SELECTED4 4)  Query Tables"
        echo " "
        echo " $IS_SELECTEDX X)  Force/Stop/Kill Oracle DB"
        echo " "
        echo " $IS_SELECTEDE E)  End/Exit"
        echo "Choose: "

        read CHOICE
echo
        if [ "$CHOICE" == "0" ]
        then
                echo "Nothing Here"

        elif [ "$CHOICE" == "1" ]
        then
                bash drop_tables.sh
                Pause

        elif [ "$CHOICE" == "2" ]
        then
                bash create_tables.sh
                Pause

        elif [ "$CHOICE" == "3" ]
        then
                bash populate_tables.sh
                Pause

        elif [ "$CHOICE" == "4" ]
        then
                bash queries.sh
                Pause

        elif [ "$CHOICE" == "E" ]
        then
                exit
        fi

    done
}
```

Dropping Tables

```
============================================================
|                   Oracle All Inclusive Tool                |
|           Main Menu - Select Desired Operation(s):         |
|       <CTRL-Z Anytime to Enter Interactive CMD Prompt>     |
============================================================
  M)  View Manual

  1)  Drop Tables
  2)  Create Tables
  3)  Populate Tables
  4)  Query Tables

  X)  Force/Stop/Kill Oracle DB

  E)  End/Exit
Choose:
1


SQL*Plus: Release 12.1.0.2.0 Production on Wed Oct 27 23:48:35 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.
```

Creating Tables

```
Choose:
2


SQL*Plus: Release 12.1.0.2.0 Production on Wed Oct 27 23:25:29 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>   2    3    4    5    6    7    8    9   10   11
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10   11
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10   11
Table created.
```

Populating Tables

```
============================================================
|                  Oracle All Inclusive Tool                |
|          Main Menu - Select Desired Operation(s):         |
|       <CTRL-Z Anytime to Enter Interactive CMD Prompt>    |
============================================================
  M)  View Manual

  1)  Drop Tables
  2)  Create Tables
  3)  Populate Tables
  4)  Query Tables

  X)  Force/Stop/Kill Oracle DB

  E)  End/Exit
Choose:
3


SQL*Plus: Release 12.1.0.2.0 Production on Fri Oct 29 22:00:42 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL> SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.
```

Querying Table

```
================================================================
|                    Oracle All Inclusive Tool                 |
|           Main Menu - Select Desired Operation(s):           |
|       <CTRL-Z Anytime to Enter Interactive CMD Prompt>       |
================================================================
  M)  View Manual

  1)  Drop Tables
  2)  Create Tables
  3)  Populate Tables
  4)  Query Tables

  X)  Force/Stop/Kill Oracle DB

  E)  End/Exit
Choose:
4


SQL*Plus: Release 12.1.0.2.0 Production on Fri Oct 29 22:20:34 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>
CUSTOMER_NAME                                      CUSTOMER_ID
-------------------------------------------------- -----------
John Smith                                                 100
Lola Tran                                                  150
Raul Garcia                                                200
Alex Ien                                                   500

SQL>   2
ISSUED_CUST ??
----------- --------------------------------------------------
        100 Don_Quixote
        150 A_Tale_of_Two_Cities
        200 The_lord_of_the_Rings
        500 Real
```

```
SQL>
EMPLOY_NAME                                        EMPLOY_ID
-------------------------------------------------- ----------
John Proctor                                                7
M.D. Luffy                                                 13
Joe Goldberg                                           897985

SQL> SQL> SQL>
AVG_SALARY
----------
     29000

SQL> SQL> SQL>   2    3
COUNT(ISBN) AUTHOR
----------- --------------------------------------------------
          1 Charles_Dickens
          1 Ryan_Jean
          2 JRR_Tolkien
          1 Miguel_de_Cervantes
```

6. Normalization of the Database/Functional Dependencies

Books Table

Column names
ISBN, book_title, category, rental_price, status, author, publisher

Table data
978-0-74-452502-1, 'Don_Quixote', 'Adventure_fiction', 8, 'Available', 'Miguel_de_Cervantes', 'Penguin_Books'
978-1-42-180819-2, 'A_Tale_of_Two_Cities', 'Historical_fiction', 6, 'Available', 'Charles_Dickens', 'Penguin_Books'
978-0-26-110368-9, 'The_lord_of_the_Rings', 'Fantasy_fiction', 1, 'Not_Available', 'JRR_Tolkien', 'Allen_and_Unwin'
978-3-16-148410-0, 'Real','Adventure_fiction', 8, 'Available', 'Ryan_Jean', 'Penguin_Books'
978-0-04-823273-1, 'The_Hobbit', 'Fantasy_fiction', 10, 'Available', 'JRR_Tolkien', 'George_Allen_and_Unwin'

Functional Dependencies:
ISBN -> book_title, category, rental_price, status, author, publisher
book_title -> ISBN, category, rental_price, status, author, publisher
category, status -> rental_price, publisher
rental_price -> category, status, publisher
author -> category
publisher -> status

Few partial Dependencies:

category, rental_price, status -> publisher
category, rental_price, publisher -> status
category, status, publisher -> rental_price
rental_price, status, publisher -> category
rental_price, publisher -> category, status

Employee Table

Column names
employ_id, employ_name, position, salary

Table data
000001,'Hani Asim', 'Branch Manager', 50000
000034,'Hakeem Olajuwon', 'Library Security', 35000
000007,'John Proctor', 'Library Staff', 20000
897985,'Joe Goldberg', 'Library Staff', 20000
000013,'M.D. Luffy', 'Library Staff', 20000

Functional Dependencies:
employ_id -> employ_name, position, salary
employ_name -> employ_id, position, salary
position -> salary
salary -> position

Few partial Dependencies:
employ_id, employ_name -> position, salary
employ_id, employ_name, position -> salary
employ_id, employ_name, salary -> position
employ_id, position, salary -> employ_name
employ_id, salary -> employ_name, position

Customer Table

Column names
customer_id customer_name customer_address registration_date

Table data
100,'John Smith','17 Zane Lane', '2021-07-22'
150,'Lola Tran','30 Resident Street','2021-07-23'
200,'Raul Garcia','100 Auberg Court','2021-07-24'
500,'Alex Ien','230 Victoria Place', '2021-07-25'

Functional Dependencies:
customer_id -> customer_name, customer_address, registration_date
customer_name -> customer_id, customer_address, registration_date
customer_address -> customer_id, customer_name, registration_date
registration_date -> customer_id, customer_name, customer_address

Few partial Dependencies:
customer_id, customer_name, customer_address -> registration_date
customer_id, customer_name, registration_date -> customer_address
customer_id, registration_date -> customer_name, customer_addres
customer_name, customer_address -> customer_id, registration_date
customer_address, registration_date -> customer_id, customer_name

Branch Table

Column names

branch_no,manager_id,branch_address,contact_no

Table data

1000,50,'123 Chester Street',905-123-4567
750,100,'103 Zane Lane',416-034-2934
499,150,'39 Private Drive',416-046-4392
250,450,'29 Divon Court',904-930-9323

Functional Dependencies:

branch_no -> manager_id, branch_address, contact_no
manager_id -> branch_no, branch_address, contact_no
branch_address -> branch_no, manager_id, contact_no
contact_no -> branch_no, manager_id, branch_address

Few partial Dependencies:

branch_no, manager_id, branch_address -> contact_no
branch_no, manager_id, contact_no -> branch_address
branch_no, branch_address -> manager_id, contact_no
branch_no, branch_address, contact_no -> manager_id
manager_id, branch_address, contact_no -> branch_no

Issue Status Table

Column names

issue_id, issued_cust, issued_book_name, issue_date, isbn_book

Table data

51, 100, 'Don_Quixote','2021-08-25',978-0-74-452502-1
52, 150, 'A_Tale_of_Two_Cities', '2021-08-26',978-1-42-180819-2
53, 200, 'The_lord_of_the_Rings', '2021-08-27',978-0-26-110368-9
55, 500, 'Real', '2021-08-28', 978-3-16-148410-0

Functional Dependencies:

issue_id -> issued_cust, issued_book_name, issue_date, isbn_book
issued_cust -> issue_id, issued_book_name, issue_date, isbn_book
issued_book_name -> issue_id, issued_cust, issue_date, isbn_book
issue_date -> issue_id, issued_cust, issued_book_name, isbn_book
isbn_book -> issue_id, issued_cust, issued_book_name, issue_date

Few partial Dependencies:
issue_id, issued_cust, issued_book_name, issue_date -> isbn_book
issue_id, issued_cust, issued_book_name, isbn_book -> issue_date
issue_id, issued_cust, issue_date, isbn_book -> issued_book_name
issue_id, issued_book_name, issue_date, isbn_book -> issued_cust
issued_cust, issued_book_name, issue_date, isbn_book -> issue_id

Return Status Table

Column names
return_id, return_cust, returned_book_name, return_date, isbn_book2

Table data
61, 100, 'Don_Quixote', '2021-09-25',978-0-74-452502-1
62, 150, 'A_Tale_of_Two_Cities', '2021-09-26',978-1-42-180819-2
63, 200, 'The_lord_of_the_Rings', '2021-09-27',978-0-26-110368-9
64, 500, 'Real', '2021-09-28',978-3-16-148410-0

Functional Dependencies:
return_id -> return_cust, returned_book_name, return_date, isbn_book2
returned_book_name -> return_id, return_cust, return_date, isbn_book2
return_date -> return_id, return_cust, returned_book_name, isbn_book2
isbn_book2 -> return_id, return_cust, returned_book_name, return_date
return_cust -> return_id, returned_book_name, return_date, isbn_book2

Few partial Dependencies:
return_id, return_cust, returned_book_name, return_date -> isbn_book2
return_id, return_cust, returned_book_name, isbn_book2 -> return_date
return_id, return_cust, return_date, isbn_book2 -> returned_book_name
return_id, returned_book_name, return_date, isbn_book2 -> return_cust
return_cust, returned_book_name, return_date, isbn_book2 -> return_id

7. Normalization / 3rd NF

Books Table

ISBN	Book Title	Category	Rental Price	Status	Author	Publisher
0000000000001	Real	Adventure Fiction	8	Available	Ryan Jean	Penguin Books
9780744525021	Don Quixote	Adventure Fiction	8	Not Available	Miguel de Cervantes	Penguin Books
9781421808192	A Tale of Two Cities	Historical Fiction	6	Available	Charles Dickens	Penguin Books
9780261103689	The Lord of the Rings	Fantasy Fiction	1	Not Available	JRR Tolkien	Allen and Unwin
9780048232731	The Hobbit	Fantasy Fiction	10	Available	JRR Tolkien	George Allen and Unwin
0000000000002	Realer Than Real	Science Fiction	7	Available	Ryan Jean	George Allen and Unwin

1st Normal Form :

- Cells are single valued ✓
- Entries in a column are of the same type ✓
- Rows are uniquely identified ✓

2nd Normal Form:

- All attributes dependant on primary key ✓

3rd Normal Form:

- All fields can be determined only by the key in the table and no other column ✓

Employee Table

Employ ID	Employ Name	Position	Salary
000001	Hani Asim	Branch Manager	50000
000034	Hakeem Olajuwon	Library Security	35000
000007	John Proctor	Library Staff	20000
897985	Joe Goldberg	Library Staff	20000
000013	M.D. Luffy	Library Staff	20000

Employee Table Decomposition

Employ ID	Employ Name
000001	Hani Asim
000034	Hakeem Olajuwon
000007	John Proctor
897985	Joe Goldberg
000013	M.D. Luffy

Employ ID	Position
000001	Branch Manager
000034	Library Security
000007	Library Staff
897985	Library Staff
000013	Library Staff

Position	Salary
Branch Manager	50000
Library Security	35000
Library Staff	20000

Since an employee's salary is fixed based on their position, we separate the tables as shown so that all attributes depend only on the primary key. With this change made the table meets the requirements for 1st, 2nd, and 3rd Normal Forms.

1st Normal Form :

- Cells are single valued ✓
- Entries in a column are of the same type ✓
- Rows are uniquely identified ✓

2nd Normal Form:

- All attributes dependant on primary key ✓

3rd Normal Form:

- All fields can be determined only by the key in the table and no other column ✓

Customer Table

Customer ID	Customer Name	Customer Address	Registration Date
100	John Smith	17 Zane Lane	21-07-22
150	Lola Tran	30 Resident Street	21-07-23
200	Raul Garcia	100 Auberg Court	21-07-24
500	Alex Ien	230 Victoria Place	21-07-25

Customer Table Decomposition

Customer ID	**Customer Name**
100	John Smith
150	Lola Tran
200	Raul Garcia
500	Alex Ien

Customer ID	**Customer Address**
100	17 Zane Lane
150	30 Resident Street
200	100 Auberg Court
500	230 Victoria Place

Customer ID	**Registration Date**
100	21-07-22
150	21-07-23
200	21-07-24
500	21-07-25

The Customer table is already in 1NF, 2NF and 3NF as all fields can be determined only by the key (customer id) in the table and no other column and all attributes are dependant on primary key (customer id)

Branch Table

BRANCH_NO	MANAGER_ID	BRANCH_ADDRESS	CONTACT_NO
250	450	29 Divon Court	904-930-9323
499	150	39 Private Drive	416-046-4392
750	100	103 Zane Lane	416-034-2934
1000	50	123 Chester Street	905-123-4567

BRANCH_NO	MANAGER_ID
250	450
499	150
750	100
1000	50

BRANCH_NO	BRANCH_ADDRESS
250	29 Divon Court
499	39 Private Drive
750	103 Zane Lane
1000	123 Chester Street

BRANCH_NO	CONTACT_NO
250	904-930-9323
499	416-046-4392
750	416-034-2934
1000	905-123-4567

The Branch table is already in 1NF, 2NF and 3NF as all fields can be determined only by the key (branch no.) in the table and no other column and all attributes are dependent on primary key (branch no.) For example, the contact number depends only on the branch number(Primary Key).

Issue Status

Issue ID	Issue Cust	Issued Book Name	Issue Date	ISBN Book
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....

Composite Primary Key	Composite Primary Key			
Issue ID	ISBN Book	Issue Cust	Issued Book Name	Return Date
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....

1st Normal Form :

- Cells are single valued ✓
- Entries in a column are of the same type ✓
- Rows are uniquely identified ✓

2nd Normal Form:

- All attributes dependant on primary key ✓

3rd Normal Form:

- All fields can be determined only by the key in the table and no other column ✓

Return Status

Return ID	Return Cust	Returned Book Name	Return Date	ISBN Book 2
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....

Composite Primary Key	Composite Primary Key			
Return ID	ISBN Book 2	Return Cust	Returned Book Name	Return Date
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....
....	....	....	....	....

1st Normal Form :

- Cells are single valued ✓
- Entries in a column are of the same type ✓
- Rows are uniquely identified ✓

2nd Normal Form:

- All attributes dependant on primary key ✓

3rd Normal Form:

- All fields can be determined only by the key in the table and no other column ✓

8. Normalization / BCNF

Books Table

ISBN (I), Book Title (B), Category (C), Rental price (R), Author (A), Publisher(P).

Bernstein's Algorithm:

R = (I, B, C, R, A, P)

FD = (I → B ; I → C ; I → R ; I → A ; I → P)

Find redundancies

I → B: I+ = (I, B, C, R, A, P) we do not get B, so not redundant

I → C: I+ = (I, B, C, R, A, P) we do not get C, so not redundant

I → R: I+ = (I, B, C, R, A, P) we do not get R, so not redundant

I → A: I+ = (I, B, C, R, A, P) we do not get A, so not redundant

I → P: I+ = (I, B, C, R, A, P) we do not get P, so not redundant

∴ R1(I, B, C, R, A, P) with FD = I → B, C, R, A, P
R2(I) with no FD

Employee Table

Employ ID (I), Employ Name (N), Position (P), Salary (S).

Bernstein's Algorithm:

R = (I, N, P, S)
FD = (I → N, I → P, I → S, P → S)

Find redundancies

I → N: I+ = (I, N, P, S) we do not get N, so not redundant

I → P I+ = (I, N, P, S) we get P, so it is redundant

I → S: I+ = (I, N, P, S) we do not get S, so not redundant

P→ S: I+ = (P , S) we get S so it is redundant

∴ R1(I, N, P) with FD = I → N, P
R2(P, S) with FD = P → S

Transitivity: I → P → S

Customer Table

customer ID (I), Customer Name (N), Customer address (A), registration date (D).

Bernstein's Algorithm:

R = (I, N, A, D)
FD = (I → N, I → A, I → D)

Checking for redundancies

I → N: I+ = (I, N, A, D) we don't get N, so it isn't redundant

I → A I+ = (I, N, A, D) we don't get A, so it isn't redundant

I → D: I+ = (I, N, A, D) we don't get D, so it isn't redundant

∴ R(I, N, A, D) with FD = I → N, A, D since there are no redundancies

Branch Table

Branch number (N), Manager ID (M), Branch address (A), contact number (C).

Bernstein's Algorithm:

R = (N, M, A, C)
FD = (N → M, N → A, N → C)

Find redundancies

N → M: N+ = (N, M, A, C) We don't get M, so it isn't redundant

N → A N+ = (N, M, A, C) we don't get A, so it isn't redundant

N → C: N+ = (N, M, A, C) we don't get C, so it isn't redundant

∴ R(N, M, A, C) with FD = N→ M, A, C since there are no redundancies

Issue Status Table

Algorithm: Lossless decomposition into BCNF

Table data:
51 100 Don_Quixote 2021-08-25 978-0-74-452502-1
52 150 A_Tale_of_Two_Cities 2021-08-26 978-1-42-180819-2
53 200 The_lord_of_the_Rings 2021-08-27 978-0-26-110368-9
55 500 Real 2021-08-28 978-3-16-148410-0

R = (issue_id (A), issued_cust (B), issued_book_name (C), issue_date (D), isbn_book(E))
FD = (A->B, C, D, E; B->A, C, D, E; C->A, B, D, E; D->A, B, C, E; E->A, B, C, D)

Evaluating R(A,B,C,D,E)
    A->B, C, D, E is not BCNF
Decomposing into R1(A,B) and R2(A,C,D,E)
    Projecting FD's onto A,B
        Evaluating fd's on basis of A
        A->B
        Evaluating fd's on basis of B
        B->A
    Projecting FD's onto A,C,D,E
        Evaluating fd's on basis of A
        Evaluating fd's on basis of C
        C->A
        Evaluating fd's on basis of A,C
        Evaluating fd's on basis of D
        D->A
        Evaluating fd's on basis of A,D
        Evaluating fd's on basis of C,D
        Evaluating fd's on basis of A,C,D
        Evaluating fd's on basis of E
        E->A
        Evaluating fd's on basis of A,E
        Evaluating fd's on basis of C,E
        Evaluating fd's on basis of A,C,E
        Evaluating fd's on basis of D,E
        Evaluating fd's on basis of A,D,E
        Evaluating fd's on basis of C,D,E
Decomposed result:
R1(A,B):
    A->B

```
        B->A
R2(A,C,D,E):
        C->A
        D->A
        E->A
-- end of iteration  --
Evaluating R(A,B)
        A->B is BCNF
        B->A is BCNF
Evaluating R(A,C,D,E)
        C->A is not BCNF
Decomposing into R4(A,C) and R5(C,D,E)
        Projecting FD's onto A,C
                Evaluating fd's on basis of A
                Evaluating fd's on basis of C
                C->A
        Projecting FD's onto C,D,E
                Evaluating fd's on basis of C
                Evaluating fd's on basis of D
                Evaluating fd's on basis of C,D
                Evaluating fd's on basis of E
                Evaluating fd's on basis of C,E
                Evaluating fd's on basis of D,E
Decomposed result:
R4(A,C):
        C->A
R5(C,D,E):
        None
-- end of iteration  --
Evaluating R(A,C)
        C->A is BCNF
Evaluating R(C,D,E)
        no FD's. Relation is BCNF
```

Solution:
R(A, B)
R(A, C)
R(C, D, E)

Return Status Table

Partial Dependency = A, B, C, E->D

Algorithm: Lossless decomposition into BCNF

Table data:
61 100 Don_Quixote 2021-09-25 978-0-74-452502-1
62 150 A_Tale_of_Two_Cities 2021-09-26 978-1-42-180819-2
63 200 The_lord_of_the_Rings 2021-09-27 978-0-26-110368-9
64 500 Real 2021-09-28 978-3-16-148410-0

R = (return_id (A), return_cust (B), returned_book_name (C), return_date (D), isbn_book2(E))
FD = (A->B, C, D, E; B->A, C, D, E; C->A, B, D, E; D->A, B, C, E; E->A, B, C, D; A, B, C, E->D)

Evaluating R(A,B,C,D,E)
A, B, C, E->D is not BCNF
Decomposing into R1(B, C, E,A,B,D) and R2(B, C, E,A,C,E)
Projecting FD's onto B, C, E,A,B,D
Evaluating fd's on basis of B
Evaluating fd's on basis of C
Evaluating fd's on basis of B, C
Evaluating fd's on basis of E
Evaluating fd's on basis of B, E
Evaluating fd's on basis of C, E
Evaluating fd's on basis of B, C, E
Evaluating fd's on basis of A
A-> C, E,B
Evaluating fd's on basis of B,A
B,A-> C, E,B,D
Evaluating fd's on basis of C,A
C,A-> E,B
Evaluating fd's on basis of B, C,A
Evaluating fd's on basis of E,A
E,A-> C,B
Evaluating fd's on basis of B, E,A
Evaluating fd's on basis of C, E,A
Evaluating fd's on basis of B, C, E,A
Evaluating fd's on basis of B
B-> C, E,A
Evaluating fd's on basis of B,B
Evaluating fd's on basis of C,B

Evaluating fd's on basis of B, C,B
Evaluating fd's on basis of E,B
Evaluating fd's on basis of B, E,B
Evaluating fd's on basis of C, E,B
Evaluating fd's on basis of B, C, E,B
Evaluating fd's on basis of A,B
Evaluating fd's on basis of B,A,B
Evaluating fd's on basis of C,A,B
Evaluating fd's on basis of B, C,A,B
Evaluating fd's on basis of E,A,B
Evaluating fd's on basis of B, E,A,B
Evaluating fd's on basis of C, E,A,B
Evaluating fd's on basis of B, C, E,A,B
Evaluating fd's on basis of D
D-> B, C, E,A,B
Evaluating fd's on basis of B,D
Evaluating fd's on basis of C,D
Evaluating fd's on basis of B, C,D
Evaluating fd's on basis of E,D
Evaluating fd's on basis of B, E,D
Evaluating fd's on basis of C, E,D
Evaluating fd's on basis of B, C, E,D
Evaluating fd's on basis of A,D
Evaluating fd's on basis of B,A,D
Evaluating fd's on basis of C,A,D
Evaluating fd's on basis of B, C,A,D
Evaluating fd's on basis of E,A,D
Evaluating fd's on basis of B, E,A,D
Evaluating fd's on basis of C, E,A,D
Evaluating fd's on basis of B, C, E,A,D
Evaluating fd's on basis of B,D
Evaluating fd's on basis of B,B,D
Evaluating fd's on basis of C,B,D
Evaluating fd's on basis of B, C,B,D
Evaluating fd's on basis of E,B,D
Evaluating fd's on basis of B, E,B,D
Evaluating fd's on basis of C, E,B,D
Evaluating fd's on basis of B, C, E,B,D
Evaluating fd's on basis of A,B,D
Evaluating fd's on basis of B,A,B,D
Evaluating fd's on basis of C,A,B,D
Evaluating fd's on basis of B, C,A,B,D
Evaluating fd's on basis of E,A,B,D

Evaluating fd's on basis of B, E,A,B,D
Evaluating fd's on basis of C, E,A,B,D
Projecting FD's onto B, C, E,A,C,E
Evaluating fd's on basis of B
Evaluating fd's on basis of C
Evaluating fd's on basis of B, C
Evaluating fd's on basis of E
Evaluating fd's on basis of B, E
Evaluating fd's on basis of C, E
Evaluating fd's on basis of B, C, E
Evaluating fd's on basis of A
A-> C, E
Evaluating fd's on basis of B,A
B,A-> C, E
Evaluating fd's on basis of C,A
C,A-> E
Evaluating fd's on basis of B, C,A
Evaluating fd's on basis of E,A
E,A-> C
Evaluating fd's on basis of B, E,A
Evaluating fd's on basis of C, E,A
Evaluating fd's on basis of B, C, E,A
Evaluating fd's on basis of C
C-> B, C, E,A
Evaluating fd's on basis of B,C
Evaluating fd's on basis of C,C
Evaluating fd's on basis of B, C,C
Evaluating fd's on basis of E,C
Evaluating fd's on basis of B, E,C
Evaluating fd's on basis of C, E,C
Evaluating fd's on basis of B, C, E,C
Evaluating fd's on basis of A,C
Evaluating fd's on basis of B,A,C
Evaluating fd's on basis of C,A,C
Evaluating fd's on basis of B, C,A,C
Evaluating fd's on basis of E,A,C
Evaluating fd's on basis of B, E,A,C
Evaluating fd's on basis of C, E,A,C
Evaluating fd's on basis of B, C, E,A,C
Evaluating fd's on basis of E
E-> B, C, E,A
Evaluating fd's on basis of B,E
Evaluating fd's on basis of C,E

Evaluating fd's on basis of B, C,E
Evaluating fd's on basis of E,E
Evaluating fd's on basis of B, E,E
Evaluating fd's on basis of C, E,E
Evaluating fd's on basis of B, C, E,E
Evaluating fd's on basis of A,E
Evaluating fd's on basis of B,A,E
Evaluating fd's on basis of C,A,E
Evaluating fd's on basis of B, C,A,E
Evaluating fd's on basis of E,A,E
Evaluating fd's on basis of B, E,A,E
Evaluating fd's on basis of C, E,A,E
Evaluating fd's on basis of B, C, E,A,E
Evaluating fd's on basis of C,E
Evaluating fd's on basis of B,C,E
Evaluating fd's on basis of C,C,E
Evaluating fd's on basis of B, C,C,E
Evaluating fd's on basis of E,C,E
Evaluating fd's on basis of B, E,C,E
Evaluating fd's on basis of C, E,C,E
Evaluating fd's on basis of B, C, E,C,E
Evaluating fd's on basis of A,C,E
Evaluating fd's on basis of B,A,C,E
Evaluating fd's on basis of C,A,C,E
Evaluating fd's on basis of B, C,A,C,E
Evaluating fd's on basis of E,A,C,E
Evaluating fd's on basis of B, E,A,C,E
Evaluating fd's on basis of C, E,A,C,E

Decomposed result:

R1(B, C, E,A,B,D):

A-> C, E,B
B,A-> C, E,B,D
C,A-> E,B
E,A-> C,B
B-> C, E,A
D-> B, C, E,A,B

R2(B, C, E,A,C,E):

A-> C, E
B,A-> C, E
C,A-> E
E,A-> C
C-> B, C, E,A
E-> B, C, E,A

```
-- end of iteration  --
Evaluating R( B, C, E,A,B,D)
        A-> C, E,B is not BCNF
Decomposing into R3( C, E,A,B) and R4( B,A,D)
        Projecting FD's onto  C, E,A,B
                Evaluating fd's on basis of  C
                Evaluating fd's on basis of  E
                Evaluating fd's on basis of  C, E
                Evaluating fd's on basis of A
                A-> C, E,B
                Evaluating fd's on basis of  C,A
                 C,A-> E,B
                Evaluating fd's on basis of  E,A
                 E,A-> C,B
                Evaluating fd's on basis of  C, E,A
                Evaluating fd's on basis of B
                B-> C, E,A
                Evaluating fd's on basis of  C,B
                Evaluating fd's on basis of  E,B
                Evaluating fd's on basis of  C, E,B
                Evaluating fd's on basis of A,B
                Evaluating fd's on basis of  C,A,B
                Evaluating fd's on basis of  E,A,B
        Projecting FD's onto  B,A,D
                Evaluating fd's on basis of  B
                Evaluating fd's on basis of A
                Evaluating fd's on basis of  B,A
                 B,A->D
                Evaluating fd's on basis of D
                D-> B,A
                Evaluating fd's on basis of  B,D
                Evaluating fd's on basis of A,D
Decomposed result:
R3( C, E,A,B):
        A-> C, E,B
         C,A-> E,B
         E,A-> C,B
        B-> C, E,A
R4( B,A,D):
         B,A->D
        D-> B,A
-- end of iteration  --
Evaluating R( C, E,A,B)
```

A-> C, E,B is BCNF
C,A-> E,B is BCNF
E,A-> C,B is BCNF
B-> C, E,A is BCNF
Evaluating R(B,A,D)
B,A->D is BCNF
D-> B,A is BCNF
Evaluating R(B, C, E,A,C,E)
A-> C, E is not BCNF
Decomposing into R7(C, E,A) and R8(B,A,C,E)
Projecting FD's onto C, E,A
Evaluating fd's on basis of C
Evaluating fd's on basis of E
Evaluating fd's on basis of C, E
Evaluating fd's on basis of A
A-> C, E
Evaluating fd's on basis of C,A
C,A-> E
Evaluating fd's on basis of E,A
E,A-> C
Projecting FD's onto B,A,C,E
Evaluating fd's on basis of B
Evaluating fd's on basis of A
Evaluating fd's on basis of B,A
Evaluating fd's on basis of C
C-> B,A
Evaluating fd's on basis of B,C
B,C->A
Evaluating fd's on basis of A,C
A,C-> B
Evaluating fd's on basis of B,A,C
Evaluating fd's on basis of E
E-> B,A
Evaluating fd's on basis of B,E
Evaluating fd's on basis of A,E
Evaluating fd's on basis of B,A,E
Evaluating fd's on basis of C,E
Evaluating fd's on basis of B,C,E
Evaluating fd's on basis of A,C,E
Decomposed result:
R7(C, E,A):
A-> C, E
C,A-> E

```
        E,A-> C
R8( B,A,C,E):
        C-> B,A
        B,C->A
        A,C-> B
        E-> B,A
-- end of iteration  --
Evaluating R( C, E,A)
        A-> C, E is BCNF
        C,A-> E is BCNF
        E,A-> C is BCNF
Evaluating R( B,A,C,E)
        C-> B,A is not BCNF
Decomposing into R10( B,A,C) and R11(C,E)
        Projecting FD's onto  B,A,C
                Evaluating fd's on basis of  B
                Evaluating fd's on basis of A
                Evaluating fd's on basis of  B,A
                Evaluating fd's on basis of C
                C-> B,A
                Evaluating fd's on basis of  B,C
                 B,C->A
                Evaluating fd's on basis of A,C
                A,C-> B
        Projecting FD's onto C,E
                Evaluating fd's on basis of C
                Evaluating fd's on basis of E
Decomposed result:
R10( B,A,C):
        C-> B,A
        B,C->A
        A,C-> B
R11(C,E):
        None
-- end of iteration  --
Evaluating R( B,A,C)
        C-> B,A is BCNF
        B,C->A is BCNF
        A,C-> B is BCNF
Evaluating R(C,E)
        no FD's. Relation is BCNF

Solution:
```

R(C,E,A,B)
R(B,A,D)
R(C,E,A)
R(B,A,C)
R(C,E)

9. Demonstration of User Interface

Source Code Submitted in A9 folder as well all source codes are included in final submission

View of Interface Design:

```
[s352pate@thebe:~/Documents$ ls
create_tables.sh  Library_DBMS.sh     queries.sh
drop_tables.sh    populate_tables.sh
[s352pate@thebe:~/Documents$ bash Library_DBMS.sh
Library_DBMS.sh: line 59: StartMessage: command not found

================================================================
|                    Oracle All Inclusive Tool                 |
|           Main Menu - Select Desired Operation(s):           |
|       <CTRL-Z Anytime to Enter Interactive CMD Prompt>       |
================================================================

  1)  Drop Tables
  2)  Create Tables
  3)  Populate Tables
  4)  Run Existing Queries

  X)  Force/Stop/Kill Oracle DB

  E)  End/Exit
Choose:
```

1) Drop Tables

```
Choose:
1

SQL*Plus: Release 12.1.0.2.0 Production on Fri Dec 3 16:28:41 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL> SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Press ENTER to continue...
```

2) Create Tables

```
Choose:
2

SQL*Plus: Release 12.1.0.2.0 Production on Fri Dec 3 16:23:59 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>   2    3    4    5    6    7    8    9   10   11
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10   11
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10   11
Table created.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Press ENTER to continue...
```

3) Populate Tables

```
Choose:
3

SQL*Plus: Release 12.1.0.2.0 Production on Fri Dec 3 16:24:25 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL> SQL>   2
1 row created.
```

```
SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.
```

```
SQL> SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL>   2
1 row created.

SQL> SQL> SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Press ENTER to continue...
```

4) Run Queries (Posting screenshots of some queries)

```
Choose:
4

SQL*Plus: Release 12.1.0.2.0 Production on Fri Dec 3 16:26:13 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12
      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
   -451599 Don_Quixote
Adventure_fiction                                             8
Available

      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
Miguel_de_Cervantes
Penguin_Books
```

```
      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
   -179886 A_Tale_of_Two_Cities
Historical_fiction                                            6
Available

      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
Charles_Dickens
Penguin_Books
```

```
      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
   -147451 Real
Adventure_fiction                                             8
Available
```

```
      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
Ryan_Jean
Penguin_Books
```

```
SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12
      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
   -451599 Don_Quixote
Adventure_fiction                                             8
Available

      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
Miguel_de_Cervantes
Penguin_Books
```

```
      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
   -147451 Real
Adventure_fiction                                             8
Available

      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
Ryan_Jean
Penguin_Books


      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
   -822300 The_Hobbit
Fantasy_fiction                                              10
Available

      ISBN BOOK_TITLE
---------- --------------------------------------------------
CATEGORY                                           RENTAL_PRICE
-------------------------------------------------- ------------
STATUS
--------------------------------------------------
AUTHOR
--------------------------------------------------
PUBLISHER
--------------------------------------------------
JRR_Tolkien
George_Allen_and_Unwin
```

```
SQL> SQL>   2
ISSUED_BOOK_NAME                                   ISSUE_DAT
-------------------------------------------------- ---------
A_Tale_of_Two_Cities                               26-AUG-21
Don_Quixote                                        25-AUG-21
The_lord_of_the_Rings                              27-AUG-21
Real                                               28-AUG-21
```

```
SQL> SQL>   2    3    4    5    6    7    8    9
CUSTOMER_ID CUSTOMER_NAME
----------- --------------------------------------------------
CUSTOMER_ADDRESS
--------------------------------------------------------------------------------
REGISTRAT
---------
        150 Lola Tran
30 Resident Street
23-JUL-21

        200 Raul Garcia
100 Auberg Court
24-JUL-21

CUSTOMER_ID CUSTOMER_NAME
----------- --------------------------------------------------
CUSTOMER_ADDRESS
--------------------------------------------------------------------------------
REGISTRAT
---------

        500 Alex Ien
230 Victoria Place
25-JUL-21
```

```
SQL>
CUSTOMER_NAME                                      CUSTOMER_ID
-------------------------------------------------- -----------
John Smith                                             100
Lola Tran                                              150
Raul Garcia                                            200
Alex Ien                                               500
```

```
SQL> SQL>   2    3
COUNT(ISBN) AUTHOR
----------- ------------------------------------------------
          1 Charles_Dickens
          1 Ryan_Jean
          2 JRR_Tolkien
          1 Miguel_de_Cervantes

SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Press ENTER to continue...
```

Exiting the Interface

```
================================================================================
|                      Oracle All Inclusive Tool                               |
|            Main Menu - Select Desired Operation(s):                          |
|        <CTRL-Z Anytime to Enter Interactive CMD Prompt>                      |
================================================================================

  1)  Drop Tables
  2)  Create Tables
  3)  Populate Tables
  4)  Run Existing Queries

  X)  Force/Stop/Kill Oracle DB

  E)  End/Exit
Choose:
E
s352pate@thebe:~/Documents$
```

10. Relational Algebra and Final Documents

Relational Algebra - 7 Queries

Query 1

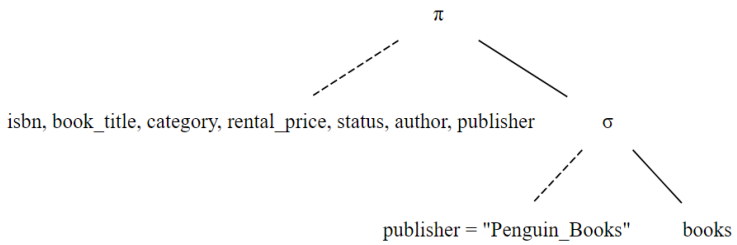
SQL:

```
SELECT
        isbn,
        book_title,
        category,
        rental_price,
        status,
        author,
        publisher
FROM
        books
WHERE
        publisher = 'Penguin_Books';
```

Relational Algebra:

$\pi_{\text{isbn, book\_title, category, rental\_price, status, author, publisher}}$

$\sigma_{\text{publisher = "Penguin\_Books"}}$ books



Query 2:

SQL:

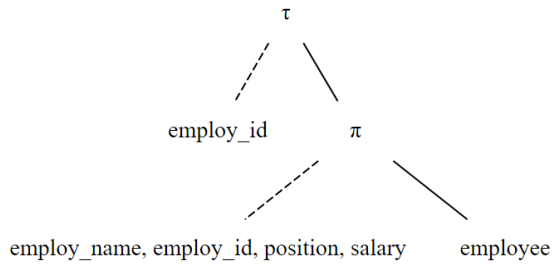
```
SELECT
        employ_name,
        employ_id,
        position,
        salary

FROM
         Employee
ORDER BY employ_id  ASC;
```

Relational Algebra:

τ_{employ_id}

$\pi_{employ\_name,\ employ\_id,\ position,\ salary}$ employee



Query 3

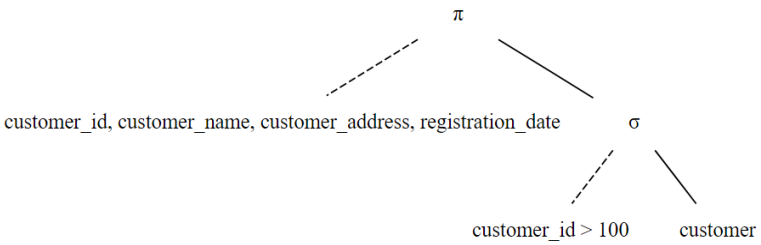
SQL:

```
SELECT
        customer_id,
        customer_name,
        customer_address,
        registration_date
FROM
        CUSTOMER
Where
        customer_id>100;
```

Relational Algebra:

$\pi_{customer\_id,\ customer\_name,\ customer\_address,\ registration\_date}$

$\sigma_{customer\_id\ >\ 100}$ customer



Query 4

SQL:

```
SELECT DISTINCT
        branch_no,
        manager_id,
        branch_address,
        contact_no
FROM
        branch
Where
```

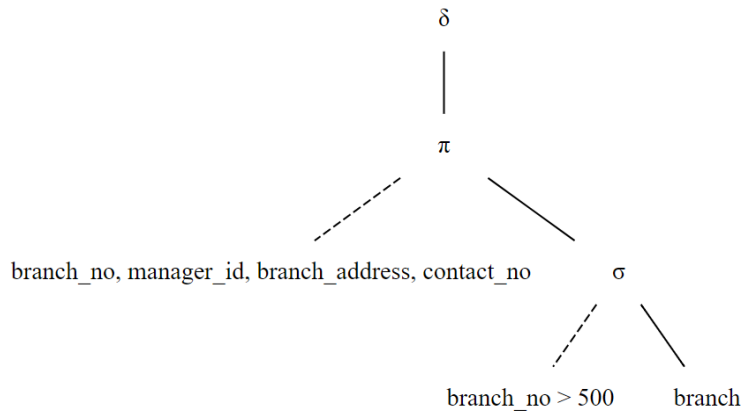
branch_no>500;

Relational Algebra:

δ

$\pi_{branch\_no,\ manager\_id,\ branch\_address,\ contact\_no}$

$\sigma_{branch\_no\ >\ 500}$ branch



Query 5

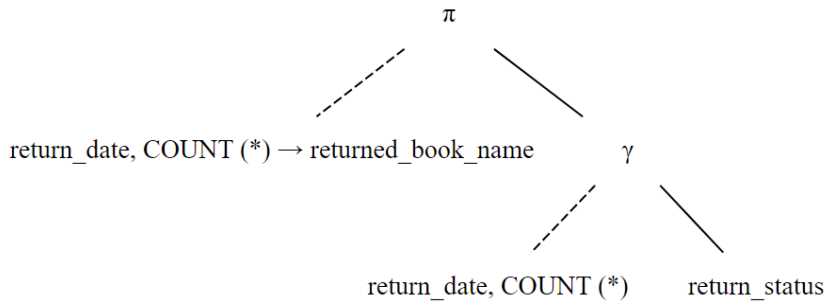
SQL:

```
SELECT return_date,count(*) as returned_book_name
FROM return_status
GROUP BY return_date
```

Relational Algebra:

$\pi_{return\_date,\ COUNT\ (*)\ \rightarrow\ returned\_book\_name}$

$\gamma_{return\_date,\ COUNT\ (*)}$ return_status



Query 6

SQL:

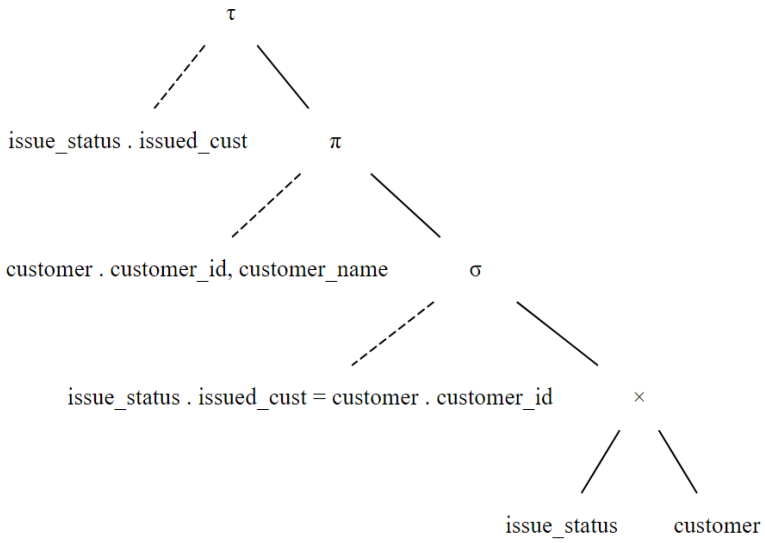
```
SELECT CUSTOMER.CUSTOMER_ID, CUSTOMER_NAME
FROM  issue_status, CUSTOMER
WHERE issue_status.ISSUED_CUST = CUSTOMER.CUSTOMER_ID,
ORDER BY issue_status.ISSUED_CUST ASC;
```

Relational Algebra:

$\tau_{\text{issue\_status . issued\_cust}}$

$\pi_{\text{customer . customer\_id, customer\_name}}$

$\sigma_{\text{issue\_status . issued\_cust = customer . customer\_id}}(\text{issue\_status} \times \text{customer})$



Query 7

SQL:

```
SELECT COUNT(ISBN), AUTHOR
FROM BOOKS
GROUP BY AUTHOR;
```

Relational Algebra:

$\gamma_{\text{author, COUNT (isbn)}}\ \text{books}$