|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6169 | ISYS6169001  Database Systems |
| **Information Systems** | **O222-ISYS6169-NT08-00** |
| ***Valid on*** *Odd Semester Year 2021/2022* | **Revision 00** |

1. Seluruh kelompok tidak diperkenankan untuk:

*The whole group is not allowed to:*

* + - Melihat sebagian atau seluruh proyek kelompok lain,

*Seeing a part or the whole project from another groups*

* + - Menyadur sebagian maupun seluruh proyek dari buku,

*Adapted a part or the whole project from the book*

* + - Mendownload sebagian maupun seluruh proyek dari internet,

*Downloading a part or the whole project from the internet,*

* + - Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,

*Working with another theme which is not in accordance with the existing theme in the matter of the project,*

* + - Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + - Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai kelompok** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the group is proved to the actions described in point 1 above, the score of the group which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan proyek, segala jenis pengumpulan proyek di luar jadwal tidak dilayani.

*Pay attention to the submission schedule for the project, all kinds of submission outside the project schedule will not be accepted*

1. Bila Anda tidak membaca peraturan ini, maka Anda dianggap telah membaca dan menyetujuinya

*If you have missed to read these regulations, so you are considered to have read and agreed on it*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| 30% | 30% | 40% |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| SQL Server Management Studio 18.5.1  SQL Server Developer 2019  Microsoft Office 365  Visual Paradigm Community Edition 16.1 |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri, proyek dan uap untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment, project, and final exam collection for this subject are described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| SQL | SQL, VPP, Image Files (JPG / PNG) | SQL |

## Soal

*Case*

**Bluejack Library**

**Bluejack Library** is an official library of Bluejack University which provides a lot of book choices for its student to borrow.

Every **staff** registered in Bluejack Library can **handle borrow transaction** from **student** andalso **receive book(s) donation** from **donator**. Every staff must have the following data to register:

* Every staff registered must have a personal information like name, gender, address, phone and salary. Every staff has an identification number with the following format:

“SFXXX”

X => number between 0 – 9

* Student can borrow book(s) from staff.
* Every **borrow transaction** made with the staff have all the information about student, staff, borrow date, return date, and information about borrowed book(s). To increase flexibility, each borrow transaction might consists of **different type of books** with **different return date**. However, student can only borrow **one** book of **each kind** in **a single transaction**. Every **borrow transaction** has an identification number with the following format:

“BTXXX”

X => number between 0 – 9

* Staff can also receive book donation from donator who wants to donate book(s) to Bluejack Library.
* Every **donation transaction** made by the donator have all the information about staff, donator, transaction date, donated book, and its quantity. Every **donation transaction** has an identification number with the following format:

“DTXXX”

X => number between 0 – 9

* Every **book** stored in Bluejack Library have its own title, publish date, stock, rating, and its category. Every **book** has an identification number with the following format:

“BKXXX”

X => number between 0 – 9

* Each book available in Bluejack Library can be classified into several different **categories**. Every **book category** has an identification number with the following format:

“BCXXX”

X => number between 0 – 9

Every **student** in Bluejack Library who wants to **borrow** book(s) has to follow the **borrow transaction** requirements that includes:

* Provides necessary personal information such as name, gender, address, and email. Every **student** has an identification number with the following format:

“STXXX”

X => number between 0 – 9

Every **donator** who wants to **donate** books to Bluejack Library has to follow the **donation transaction** requirements that includes:

* Provides necessary personal information such as name, gender, address, and phone number. Every **donator** has an identification number with the following format:

“DRXXX”

X => number between 0 – 9

* Donator can donate a **minimum** quantity of 10 books and **maximum** quantity of 500 books for each book.

**Notes:**

* Student and staff gender must be either “**Male**” or “**Female**” (without quote).
* Student email must contain ‘**@**’ (without quote)
* Staff phone must start with ‘**+62’** (without quote)
* Donator name must **consist of more than 1 character**.
* Book publish year must be **more than 2011**.
* Borrow transaction’s return date must be **more than today’s date**.
* Quantity must be **between 10 and 500 (inclusively).**

Until now, Bluejack Library still relies on manual method to store and manage its data. However, its new management decided to change it into a database system which offers more efficiency and flexibility for storing data. Therefore, you as a database engineer are asked to complete the following requirements:

1. Create Entity Relationship Diagram to maintain **borrow and donation transactions**.
2. Create a database system using DDL syntax that relevant with **borrow and donation transaction**.
3. Create query using DML syntax to fill the tables in database systems with data based on the following conditions:

* **Master** table must be filled with more than or equals 10 data.
* **Transaction** table must be filled with more than or equals 15 data.
* **Transaction detail** table must be filled with more than or equals 25 data.
* For the **Book Category** table, the table must be filled with the following data:

|  |
| --- |
| **Book Category** |
| Fantasy |
| Mystery |
| Education |
| Romance |
| Sci-fi |

1. Create query using DML syntax to simulate the transactions process for **borrow and donation transaction**.

**Note**: DML syntax to **fill database** and DML syntax to **simulate** the **transactions process** should be a **different query**.

1. To support database management process in **Bluejack Library**,you are asked to provide several queries to acquire some essential data. The required query include:
2. Display StudentName, StudentAddress, BorrowTransactionID, BorrowTransactionDate, and number of books borrowed (obtained from the total number of books borrowed) for every borrow transaction happened in 2020 by student whose address ends with ‘ Street’.
3. Display BookTitle, Publish Month (obtained from the month of the book publish date), BookCategoryName, and Total Sum Donation (obtained from the total donation quantity) for each book whose category name contains ‘y’ and published in an odd month.
4. Display BorrowTransactionID, Borrow Transaction Date (obtained from BorrowTransactionDate with ‘dd mon yyyy’ format), StudentName, Books Borrowed (obtained from the total number of borrowed books), and Average Book Rating (obtained from the average rating of borrowed books) for every borrow transaction whose student has ‘gmail’ domain email and Average Book Rating more than 4.0.
5. Display DonatorName (obtained from DonatorName and started with ‘Ms.’), DonationDate (obtained from DonationDate with ‘Mon dd, yyyy’ format), Books Donated (obtained from the number of donated books), and Average Rating (obtained from the average rating of the donated books) for each donation happened in the first two weeks (inclusively between the 1st and the 14th day) from a female donator.
6. Display DonatorName, DonationDate, StaffName, StaffGender, and StaffSalary (obtained from StaffSalary and started with ‘Rp.’) for every donation completed by staff whose salary is more than the average staff salary and its donator name consists of minimum two words. Sort the result by DonationDate in descending order.

**(alias subquery)**

1. Display DonationID, BookTitle (obtained from removing all white spaces from BookTitle), Rating Percentage (obtained from multiplying the BookRating with 20 and added with ‘%’ at the end), Quantity, and DonatorPhone for each donation with book rating more than the average rating and DonatorAddress consists of more than 15 characters.   
   **(alias subquery)**
2. Display BorrowTransactionID, Borrow Date (obtained from BorrowTransactionDate in ‘mm-dd-yyyy’ format), Return Day (obtained from the day name of the return date), BookTitle, BookRating (obtained from BookRating followed by ‘ star(s)’), and BookCategoryName for each borrow transaction which contains book(s) whose rating is equal to the minimum rating or the maximum rating of all available books while also have more than 10 stocks. Sort the result based on the BorrowTransactionID in descending order.

**(alias subquery)**

1. Display StudentName (obtained from StudentName added with ‘Mr. ’ at the beginning), StudentEmail (obtained by removing ‘.com’ from StudentEmail), Books Borrowed (obtained from the total number of books borrowed) for each borrow transaction done by male student and served by staff whose salary is more than the average staff salary. Then, combine it with StudentName (obtained from StudentName added with ‘Ms. ’ at the beginning), StudentEmail (obtained by removing ‘.com’ from StudentEmail), Books Borrowed (obtained from the total number of books borrowed) for each borrow transaction done by female student and served by staff whose salary is less than the average staff salary.

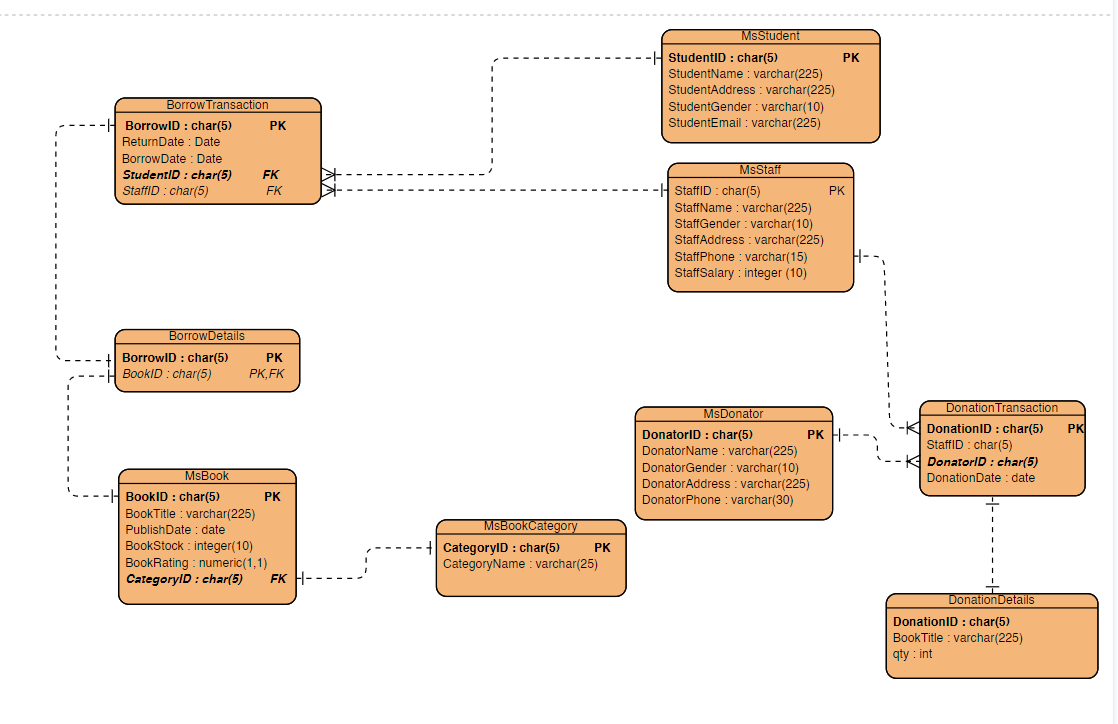
**(alias subquery)**

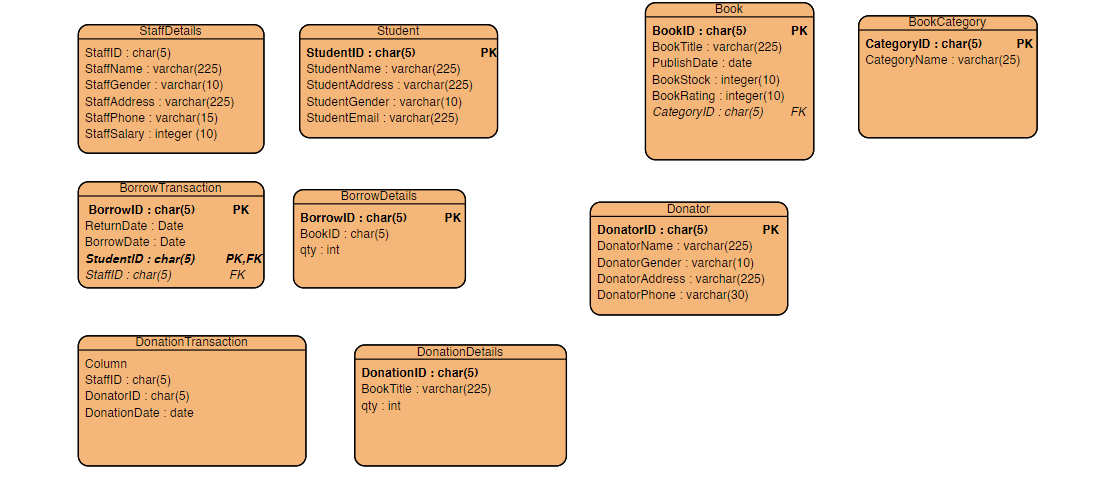
1. Create a view named ‘ViewDonationDetail’ to display DonatorName, Donation Transaction (obtained from the number of donation transaction), Average Quantity (obtained from the average donation quantity) for each donation done by donator whose address ended with ‘ Street’ or ‘ Avenue’ and Donation Transaction more than 1.
2. Create a view named ‘ViewStudentBorrowingData’ to display StudentName, Borrow Transaction (obtained from the total number of transaction), and Average Duration (obtained from the average different days between the borrow date and return date) for each borrow transaction done by male student whose email contains ‘yahoo’.

**File that must be collected**:

1. Entity Relationship Diagram (.vsdx, .png)
2. Query to create the database system. (.sql)
3. Query to insert data into tables. (.sql)
4. Query to simulate the transactions processes. (.sql)
5. Query to answer the 10 cases. (.sql)

Here are the rules that you must follow to create your project:

1. Use appropriate software for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya
2. Use the techniques taught during practicum
3. Collect appropriate files for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya
4. Include the other files that can support your project, such as:
   * All files in your project
   * Other files (image, audio, video, etc.) used in your project
   * \*.DOC file (documentation of your project) that contains the reference links of additional files (image, audio, video, etc.) used in your project
   * 



--create database SuperMarket

go

use SuperMarket

create table MsCategory(

CategoryID char(10) primary key check(CategoryID like 'CT[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

CategoryName varchar(255) not null

)

create table MsProduct(

ProductID char(10) primary key check(ProductID like 'PD[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

CategoryID char(10) foreign key references MsCategory(CategoryID)on update cascade on delete cascade not null,

ProductName varchar(255) not null,

ProductPrice int not null,

Stock int not null

)

create table MsStaffPosition(

StaffPositionID char(10) primary key check(StaffPositionID like 'SP[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

StaffPosition varchar(255) not null

)

create table MsStaff(

StaffID char(10) primary key check(StaffID like 'ST[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

StaffPositionID char(10) foreign key references MsStaffPosition(StaffPositionID) on update cascade on delete cascade not null,

StaffName varchar(255) not null,

StaffGender varchar(8) check(StaffGender in ('Male','Female')),

StaffEmail varchar(255) check(StaffEmail like '%@%'),

StaffDOB date not null

)

create table TransactionHeader(

TransactionID char(15) primary key check(TransactionID like 'TS[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

StaffID char(10) foreign key references MsStaff(StaffID) on update cascade on delete cascade not null,

TransactionDate date not null,

PaymentMethod varchar(50) check(PaymentMethod in ('Card', 'Cash'))

)

create table TransactionDetail(

TransactionID char(15) foreign key references TransactionHeader(TransactionID) on update cascade on delete cascade not null,

ProductID char(10) foreign key references MsProduct(ProductID) on update cascade on delete cascade not null,

Quantity int not null,

primary key(TransactionID, ProductID)

)

Select \*From TransactionDetail

