|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6084 | ISYS6123 | ISYS6123003 | ISYS6169 | ISYS6169001  Database | Introduction to Database Systems | Database Systems |
| **Information Systems** | **E222-ISYS6123003-CA06-00** |
| ***Valid on*** *Even Semester Year 2021/2022* | **Revision 00** |

1. Seluruh kelompok tidak diperkenankan untuk:

*The whole group is not allowed to:*

* + 1. Melihat sebagian atau seluruh proyek kelompok lain,

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

* + 1. Menyadur sebagian maupun seluruh proyek dari buku,

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

* + 1. Mendownload sebagian maupun seluruh proyek dari internet,

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

* + 1. 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,*

* + 1. Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + 1. 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* |
| 40% | 60% |

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

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

|  |
| --- |
| **Software**  *Software* |
| Microsoft Office 365  SQL Server Developer 2019  SQL Server Management Studio 18.9.1  Visual Paradigm Community Edition 16.3 |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri dan proyek 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* |
| SQL | SQL, VPP, Image Files (JPG / PNG) |

## Soal

*Case*

**RoCALink**

**RoCALink** is a bike shop that sells many types of bikes, from various brands. The staff manages all activites in **RoCALink** related **to transactions and all the bikes available at the shop**.

Every staff that hired by **RoCALink** have a task to **serve a customer who wants to buy a bike from the shop**. Every staff must be following the procedures to become a staff, which are:

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

“STXXX”

X => number between 0 – 9

* Every **transaction** made with the customer have all the information about **staff, customer, transaction date, bikes purchased, and the quantity of each bike**. Every **transaction** has an identification number with the following format:

“TRXXX”

X => number between 0 – 9

* Every **bike** sold by **RoCALink** must have its **own name and price**. Every **bike** has an identification number with the following format:

“BKXXX”

X => number between 0 – 9

* Every **bike** has its **type** **data** that store information about the **type name**. Every **bike type** has an identification number with the following format:

“BTXXX”

X => number between 0 – 9

* Every **bike** has its **groupset** **data** that is installed on a bike and must have its own **name, number of gears, and has info on its wireless capability**. Every **groupse**t has an identification number with the following format:

“GRXXX”

X => number between 0 – 9

* Every **bike** has its **brand data** and must have its own **brand name, description, website, and nationality**. Every brand has an identification number with the following format:

“BRXXX”

X => number between 0 – 9

Every customer that wants to buy bikes at **RoCALink** must be following the **transaction procedures**, those are:

* Every customer that wants to purchase a bike must already completed personal information like **name, phone number, gender**, **and** **email**. Every customer has an identification number with the following format:

“CUXXX”

X => number between 0 – 9

* Customer can purchase **more than one bike** in every transaction.

**Constraints:**

* Staff name must be more than 4 characters.
* Staff email must end with “@rocalink.com”
* Staff gender must be either “Male” or “Female” (without quote).
* Staff Phone number must start with “08”
* Customer name must be more than 4 characters.
* Customer gender must be either “Male” or “Female” (without quote).
* Customer email must end with “@gmail.com”
* Customer Phone number must start with “08”
* Brand website must start with “www.”
* Groupset gear number must be between 4 and 12.
* Groupset wireless capability must either be “True” or “False” (without quote).
* Transaction date must not be more than the current date.
* Transaction quantity must be more than 0.
* Bike price must be more than 0.

Now **RoCALink** still using manual management system to maintain the **sales transactions** and **bike**. You as a programmer is hired to help **RoCALink** to create a database system that can store data and maintain the **sales** **transactions** and **bike stock**. The tasks that you must do are:

1. Create Entity Relationship Diagram to maintain **sales** **transactions**.
2. Create a database system using DDL syntax that is relevant with **sales** **transactions**.
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.

1. Create query using DML syntax to simulate the transactions process for **sales** **transactions**.

**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 **RoCALink**,the staff has asked you to provide some query that resulting important data. The requirements are:
2. Display CustomerID, CustomerName, Total Item Variety (obtained from total varieties of bike in transaction and ends with 'Types') for every male customer whose name starts with 'A'.
3. Display BikeTypeName, BikeTypeID, And Bike Count (obtained from total bikes that has that groupset), for every bike that has a groupset which name starts with 'Shimano' and has number of gears between 7 and 12.
4. Display StaffID, StaffName, Number of Transactions (obtained from total number of different bike sold in one transaction) and Number of Bikes Sold (obtained from total number of bikes sold by staff, and end with ' Bikes') for every staff whose gender is Female and has a name between 5 and 10 characters long.
5. Display GroupsetID, GroupsetName, Bike Count (obtained from total number of bikes with that groupset), and Average Price (obtained from average price of bikes with that groupset, using rupiah money format) for every bike that has a brand that starts with 'C', and has an average price of more than 150000000.
6. Display all TransactionID, CustomerName, StaffName, and Transaction Day (obtained from the day name of the transaction date) for every transaction made by staff that has above average salary and was done in February.

**(alias subquery)**

1. Display StaffName, BikeName, TransactionID, Transaction Month (obtained from the month name of the transaction date) for every transaction that has a transaction quantity more than the maximum of all the transaction quantity from all transactions made at the 12th day of the month.

**(alias subquery)**

1. Display Average Bikes Sold (obtained from average of total number of bikes sold and ends with ' Bikes'), for every bike that is priced between 100000000 and 150000000 which is not sold over a year ago.

**(alias subquery)**

1. Display Max Bikes Purchased (obtained from max of total quantity of bikes and ends with ' Bikes') for every bike that has name starts with 'S' and are bought between July and December.

**(alias subquery)**

1. Create a view named CustomerView that display CustomerName, Total Transactions (obtained from total number of different bike bought by customer), Total Bikes Bought (obtained from total quantity of bikes bought by customer from all transactions), and Customer Phone (obtained by replacing 0 in front of the phone number with '+62'), for every customer that has made between 2 and 5 transactions and has bought more than 5 bikes.
2. Create a view named TransactionView that display TransactionID, Max Quantity (obtained from the maximum quantity of bikes bought in that transaction), Min Quantity (obtained from the minimum quantity of bikes bought in that transaction), and Days Elapsed (obtained from the number of days that has passed since the transaction was made), for every transaction where Max Quantity is not equals Min Quantity and is made by male staff.

**File that must be collected**:

1. Entity Relationship Diagram (.vpp, .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