|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6084 | ISYS6123  Database | Introduction to Database Systems |
| **Information Systems** | **E212-ISYS6123-LO05-00** |
| ***Valid on*** *Even Semester Year 2020/2021* | **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* |
| • 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 dan proyek untuk matakuliah ini adalah sebagai berikut:

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

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

## Soal

*Case*

**LOmian**

**LOmian** is a noodle shop that is currently popular among the public, because of its good quality and hygienic food. **LOmian** is famous for its Asian noodles, not only that they also sell other noodles with a wide selection of noodle types. **LOmian** is also known for his professional employee, the following are the attributes owned by the employee:

* Every employee must have personal information such as name, address, gender. Every employee has an identification number with the following format:

“EMXXX”

X => number between 0 – 9

* Staff can purchase an ingredient from a supplier.
* Every **purchase transaction** made with the supplier has all the information about THE employee, supplier, transaction date, ingredients purchased, and the quantity of each ingredient. Every **purchase transaction** has an identification number with the following format:

“PUXXX”

X => number between 0 – 9

* Every ingredient purchased from a supplier has its own name and price. Every **ingredient** has an identification number with the following format:

“INXXX”

X => number between 0 – 9

* An employee can serve a customer who wants to buy noodles.
* Every **sales** **transaction** made by the customer has all the information about the employee, customer, transaction date, noodle sold, and the quantity of each noodle purchased. Every **sales** **transaction** has an identification number with the following format:

“TRXXX”

X => number between 0 – 9

* Everynoodle has its own name, noodle type, and price. Every **noodle** has an identification number with the following format:

“NOXXX”

X => number between 0 – 9

* Every **noodle type** has its own identification number with the following format:

“NTXXX”

X => number between 0 – 9

Every customer that wants to buy noodle at **LOmian** must be following the **sales transaction procedures**, those are:

* Every customer that wants to purchase a product must complete personal information such as name, gender, address. Every customer has an identification number with the following format:

“CUXXX”

X => number between 0 – 9

Every supplier that wants to sell their ingredient must be following the **purchase transaction procedures**, those are:

* Every supplier that wants to sell their ingredient must complete personal information such as name, gender, address. Every supplier has an identification number with the following format:

“SUXXX”

X => number between 0 – 9

**Notes:**

* Employee names must between 15 and 30.
* Employee gender must be male or female.
* Customer gender must be male or female.
* Supplier gender must be male or female.
* Noodle name must end with noodle.
* Noodle price must be over or equals to 15000.
* Ingredient price must be over 0.

**LOmian** is currently having difficulty handling every noodle **sales transaction** and ingredient **purchase transaction** because they use a **manual management system**. As an expert in database systems, you are asked to create a database system that can help **LOmian** goes well in business. The tasks that you must do are:

1. Create an Entity Relationship Diagram to maintain **sales** and **purchase transactions**.
2. Create a database system using DDL syntax that relevant with **sales** and **purchase transactions**.
3. Create queries 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 15 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** and **purchase transactions**.

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

1. To support the database management process in **LOmian**,the owner asked you to provide some query that resulting in important data. The requirements that asked from him are:
2. Display EmployeeName, CustomerName, TransactionDate (obtained from TransactionDate in ‘Mon dd, yyyy’ format), TransactionCount (obtained from the total transaction) that handled by an employee whose name is 2 words names and the gender is female.
3. Display IngredientName, IngredientPrice (obtained by adding ‘Rp. ‘ without quotation mark and followed by the IngredientPrice), TotalQuantity(obtained from total quantity followed by adding ‘ Items(s)’ without quotation mark), SupplierName (obtained from SupplierName in uppercase format) for every purchase that has total quantity less than 5 and the total transaction more than 5.
4. Display NoodleName, NoodlePrice (obtained by adding ‘Rp. ’ without quotation mark and followed the NoodlePrice), TransactionCount (obtained from total transaction followed by adding ‘ Transaction(s)’ without quotation mark), TotalQuantity (obtained from total quantity followed by adding ‘ Qty(s)’ without quotation mark), TotalPrice (obtained from adding ‘Rp. ’ without quotation mark and followed by total quantity multiplied by NoodlePrice) for every transaction that occurs after the 20th and the total transaction more than or equals to 5.
5. Display IngredientName, IngredientPrice (obtained from adding ‘Rp. ’ without quotation mark followed by IngredientPrice), PurchaseCount (obtained from total purchase followed by adding ‘ Purchase(s)’ without quotation mark), TotalQuantity (obtained from total quantity followed by adding ‘ Qty(s)’ without quotation mark), TotalPrice (obtained by adding ‘Rp. ’ without quotation mark followed by total quantity multiplied by IngredientPrice) for every purchase that occurs on November and the IngredientPrice between 20000 and 30000.
6. Display NoodleID (obtained from NoodleID by replacing ‘NO’ with ‘Noodle ’ without quotation mark), NoodleName, NoodlePrice (obtained by adding ‘Rp. ’ without quotation mark followed by NoodlePrice) for every noodle that has price over the average noodle price every Wednesday. **(alias subquery)**
7. Display TransactionDate (obtained from TransactionDate in ‘Mon dd, yyyy’ format),

EmployeeName, Quantity for every transaction which the quantity less than the average quantity that occurs in 5 days before 25th November 2020. **(alias subquery)**

1. Display IngredientID (obtained from IngredientID by replacing ‘ID’ with ‘Ingredient ’ without quotation mark), IngredientName, IngredientPrice for every Ingredient which price over the minimum ingredient price and less than average ingredient price that occurs in 10th month and the IngredientName has at least 2 words.

(**alias subquery**)

1. Display TransactionDate (obtained from TransactionDate in ‘Mon dd, yyyy’ format), NoodleTypeName, Total Transaction (obtained from total transaction followed by adding ‘ Transaction(s)’ without quotation mark) for every transaction which the quantity less than the maximum quantity every Wednesday and Friday.

(**alias subquery**)

1. Create View named ‘CustomerRecord’ to display CustomerID (obtained from CustomerID by replacing ‘CU’ with ‘Customer ’ without quotation mark), CustomerName, TransactionCount (obtained from total transaction), QuantityBought (obtained from total quantity) for every transaction which the first character of the second name of the CustomerName start with ‘H’ without quotation mark and the total quantity more than 10.
2. Create View named ‘NoodleRecord’ to display TransactionDate (obtained from TransactionDate in ‘Mon dd, yyyy’ format), NoodleName (obtained from NoodleName by replacing ‘Noodle’ with ‘Mian’ without quotation mark and in lowercase format), NoodlePrice (obtained from adding ‘Rp. ’ without quotation mark followed by NoodlePrice after 20% discount), QuantityBought (obtained from total quantity followed by adding ‘ Qty(s)’ without quotation mark) for every transaction that has total transaction less than 5 and the total quantity over 10.

**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