|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6169 | ISYS6169001  Database Systems |
| **Information Systems** | **O222-ISYS6169-LL05-00** |
| ***Valid on*** *Odd Semester Year 2021/2022* | **Revision 01** |

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 other 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. Jangan lupa untuk melihat kriteria penilaian proyek yang ditempel di papan pengumuman, atau tanya asisten anda.

*Don’t forget to look at the project assessment criteria that posted on the announcement board, or ask your teaching assistant.*

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* |
| Microsoft SQL Server 2019  Microsoft Word 2010  Visual Paradigm 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* |
| - | VPP, PNG, SQL |

## Soal

*Case*

**HaLLo Pharmacy**

**HaLLo Pharmacy** is a famous pharmacy located in Jakarta. **HaLLo Pharmacy** manage transactions like **selling a medicine to customer** and **purchasing medicines from vendor.**

In **HaLLo Pharmacy,** every employee have a task to **serve a customer who wants to buy a medicine** and **purchase medicines from vendor**. Every employee must be following the procedures to become a employee, which are:

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

“EMXXX”

X => number between 0 – 9

* Employee can purchase medicines from a vendor.
* Every **purchase transaction** made with the vendor must have all the information about employee, vendor, transaction date, medicines purchased, and the quantity of each medicine. Every **purchase transaction** has an identification number with the following format:

“PCXXX”

X => number between 0 – 9

* Every medicine purchased from vendor must have some information about the medicine which are name, price, description, and stock. Every **medicine** has an identification number with the following format:

“MDXXX”

X => number between 0 – 9

* Every medicine has its own category data, and the category data has name and identification id with the following format:

“CTXXX”

X => number between 0 – 9

* Employee can also **serve a customer** who wants to buy a medicine.
* Every **sales transaction** made by the customer have all the information about employee, customer, transaction date, medicines sold, and the quantity of each medicine. Every **sales transaction** has an identification number with the following format:

“SLXXX”

X => number between 0 – 9

* Everymedicine sold by **HaLLo Pharmarcy** must have some information about the medicine which are name, price, description, and stock. Every **medicine** has an identification number with the following format:

“MDXXX”

X => number between 0 – 9

Every customer that wants to buy medicine at **HaLLo Pharmacy** must be following the **sales transaction procedures**, those are:

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

“CUXXX”

X => number between 0 – 9

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

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

* Every vendor that wants to sell their ingredient must already completed personal information like name, phone, address, email, and established year. Every vendor has an identification number with the following format:

“VNXXX”

X => number between 0 – 9

* vendor can sell **more than one medicine** in every transaction.

**Notes:**

* Vendor name must be more than 3 characters.
* Customer email must ends with “@hallo.com” (without quote).
* Employee email must ends with “@hallo.com” (without quote).
* Vendor email must ends with “.com” (without quote).
* Customer gender must be either “Male” or “Female” (without quote).
* Employee gender must be either “Male” or “Female” (without quote).
* Medicine price must be between 5000 and 100000.
* Category name must **not** be either “Amidopyrine”, “Phenacetin”, or “Methaqualone” (without quote).

Now **HaLLo Pharmacy** still using manual management system to maintain the **sales** and **purchase transactions**. You as a professional database administrator are asked by the owner of **HaLLo Pharmacy** to create a database system that can store data and maintain the **sales** and **purchase transactions**. The tasks that you must do are:

1. Create 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 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 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 database management process in **HaLLo Pharmacy**,the owner asked you to provide some query that resulting important data. The requirements that asked from the owner are:
2. Display Customer ID (obtained from CustomerID), Customer Name (obtained from Customer Name), Transaction Date (obtained from TransactionDate in 'Mon dd, yyyy' format), Medicine Bought (obtained from the sum of quantity) for every sales transaction that occured between 20th and 30th day of the month, and the customer gender is Female.
3. Display Employee Number (Obtained from the last 3 characters of EmployeeID), Employee Name (obtained from EmployeeName), and Customer Served (obtained from count of customers served by the employee ended with ' customer(s)’) for each employee whose name contains 'b' and the gender is 'Female'. **note**: the employee may serve the same customer more than once.
4. Display Customer ID (obtained from CustomerID), Customer Name (obtained from CustomerName), Date of Birth (obtained from CustomerDOB in 'dd Mon yyyy' format), Transaction Count (obtained from count of transaction made by the customer), Total Purchase (obtained from the sum of quantity times the medicine price) for each sales transaction which occured between 1st and 6th month and served by employee whose id is either 'EM004', 'EM006', or 'EM008'.
5. Display Employee ID (obtained from EmployeeID), Employee Name (obtained from EmployeeName), Local Phone Number (obtained by replacing the first 2 number of EmployeePhone with '62'), Transaction Done (obtained from total transaction done by the employee), and Total Medicine Bought (obtained from sum of quantity ended with ' item(s)'), for each purchase transaction that happened between 10th and 20th day of the month and the vendor's EstablishedYear is after 2000.
6. Display Numeric Medicine ID (obtained from the last 3 characters of MedicineID), Medicine Name (obtained from MedicineName in uppercase format), Category Name (obtained from CategoryName), Price (obtained by adding 'Rp. ' in front of MedicinePrice), Medicine Stock (obtained from MedicineStock) for each medicine which price is greater than 50000 and the stock is less than the average quantity of medicine sold in sales transaction.

(**alias subquery**)

1. Display Employee Code (obtained by replacing 'EM' with 'EMPLOYEE' in EmployeeId), Employee Name (obtained from EmployeeName), Transaction Date (obtained from TransactionDate in 'mm/dd/yyyy' format), Medicine Name (obtained from MedicineName), Medicine Price (obtained from MedicinePrice), and Quantity for every sales transaction which happened on 2nd day of the month and served by employee whose salary is less than the average salary of all staff.

(**alias subquery**)

1. Display Customer ID (obtained from CustomerID), Customer Name (obtained from Customer Name), Local Customer Phone (obtained by replacing the two first number of CustomerPhone to '62'), Date of Birth (obtained by CustomerDOB in 'Mon dd, yyyy' format), Medicine Bought (obtained from sum of quantity and ends with ' item(s)') for each customer whose gender is not Male and medicine bought is greater than average of quantity sold in sales transaction.

(**alias subquery**)

1. Display Employee ID (obtained from EmployeeID), Employee Name (obtained from the first character of EmployeeName until the character before the first space), Salary (obtained by adding 'Rp. ' in front of EmployeeSalary), Date of Birth (obtained from EmployeeDOB in 'Mon dd, yyyy' format), and Transaction Served (obtained from the count of sales transaction served by the employee) for each employee whose name contains space and the total transaction served is greater than the average of total transaction served by all employees.

(**alias subquery**)

1. Create a view named 'VendorMaximumAverageQuantityViewer' to display Vendor ID (obtained from VendorID), Vendor Name (obtained from VendorName), Average Supplied Quantity (obtained from average quantity of the supplied medicine ended with ' item(s)'), Maximum Supplied Quantity ( obtained from maximum quantity of the supplied medicine ended with ' item(s)') for each vendor whose name contains 'a' and the maximum quantity of the supplied medicine is greater than 5.
2. Create a view named 'VendorSupplyViewer' to display Vendor Number (obtained from the last 3 character of VendorID), Vendor Name (obtained from VendorName), Vendor Address (obtained from VendorAddress), Total Supplied Value (obtained by adding 'Rp. ' in front of the sum of medicine price times quantity), Medicine Type Supplied (obtained from count of medicine ended with ' medicine(s)') for each vendor which the total supplied value is greater than 150000 and total medicine supplied is greater than 2.

**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