|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6280  Database Systems |
| **Information Systems** | **E192-ISYS6280-DN01-00** |
| ***Valid on*** *Even Semester Year 2018/2019* | **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 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 Enterprise 2016  Microsoft Office Word 365  Microsoft Office Visio 2013 |

## 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* |
| - | VSDX, PNG, SQL |

## Soal

*Case*

**Dagnarok**

**Dagnarok** is a new RPG game that wants to be developed by a famous game company in Jakarta. In **Dagnarok**, every **character** can purchase **equipments** from **NPC** (Non Player Character). The **character** also can sell their **equipment** to the **NPC**. Besides do the selling and purchasing transaction in **NPC**. The game also will **keep information** of **used equipment history** of every **character**. The game developer who’s in charge of developing **Dagnarok** in the company make a regulation for the game, which are:

* Every **character** must have a personal information like name, gender, email, level, and money. Every **character** has an identification number with the following format:

“CHXXX”

X => number between 0 – 9

* Every **equipment** has its own **type** data contains of its **type** **name** and the **type** data has an identification number with the following format:

“ETXXX”

X => number between 0 – 9

* Every **equipment** has its own name, type, price, STR (Strength), INT (Intelligence), VIT (Vitality), AGI (Agility), DEX (Dexterity), LUK (Luck). Every **equipment** has an identification number with the following format:

“EQXXX”

X => number between 0 – 9

* Every character can **use** and **unuse** the equipment that they already bought from **NPC**.
* Every time the character start to **use** or **stop using** an equipment, the game will record the **character** **equipment**’**s** data in the database. The **character** **equipment**’**s** data will have all the information about character, equipment, used datetime and unused datetime. Every **character** **equipment**’**s** has an identification number with the following format:

“CQXXX”

X => number between 0 – 9

* If the character is **still using** the equipment, then the **unused datetime** will be specified by **NULL** in the database. And if the user **stop using** the equipment, the **unused datetime** will be **updated** with the **current date** and **time**.
* Every **NPC** has its own **location** data consist of its **location name** and the location data has an identification number with the following format:

“NLXXX”

X => number between 0 – 9

* Every **NPC** has its own name, location, and gender. Every **NPC** has an identification number with the following format:

“NPXXX”

X => number between 0 – 9

Every character that wants to purchase equipments from NPC must be following the **purchase transaction procedures**, those are:

* Every **purchase transaction** made with the **NPC** has all the information about character, NPC, transaction date, equipment(s) purchased, and the quantity of each equipment. Every **purchase transaction** has an identification number with the following format:

“PHXXX”

X => number between 0 – 9

* Character can **purchase one or more equipment** from the NPC.

Every character that wants to sell equipments to NPC must be following the **sales transaction procedures**, those are:

* Every **sales transaction** made by the **character** has all the information about character, NPC, transaction date, equipment(s) sold, and the quantity of each equipment. Every **sales transaction** has an identification number with the following format:

“SHXXX”

X => number between 0 – 9

* Character can also **sell** **one or more equipment** to the NPC.

**Notes:**

* Character name must be at least 2 words.
* Character name must more than 5 characters and less than 30 characters.
* Character gender must be either “Male” or “Female” (without quote).
* Character email must follow the below requirements:
  + Contain ‘@’ character
  + Ends with ‘.com’
  + There must be some character (at least one) before ‘@’
  + There must be some character (at least one) between ‘@’ & ‘.com’.

i.e. (a@.com is **not** **valid**).

* Character level must be between 1 and 100
* NPC gender must be either “Male” or “Female” (without quote).
* Unused datetime must be greater than used datetime.

Now **Dagnarok** intended to upgrade their data management system to maintain the **sales transactions** and **purchase transactions**. You as a **database administrator** in **Dagnarok** development team are asked to design and create a **database system** that can store data and maintain their **sales** and **purchase transactions**. The tasks that you must do are:

1. Create Entity Relationship Diagram to maintain **sales transactions**, **purchase transactions,** andall the game regulation that described above.
2. Create a database system using DDL syntax that relevant with the designed **Entity Relationship Diagram**.
3. Create query using DML syntax to fill the tables in database systems with data based on the following requirements:

* **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 **NPC** **Location** table, the table must be filled with the following data:

|  |
| --- |
| NPC Location Names |
| Frontera |
| Marrac |
| Iclude |
| Payun |
| Gingerbreat |

* For **Equipment** **Type** table, the table must be filled with the following data:

|  |
| --- |
| Equipment Type Names |
| Head |
| Body |
| Main-hand |
| Off-hand |
| Foot |

1. Create **insert** queries using DML syntax to simulate how the data inserted to the database if there is **a new transaction** with **more than one equipment** for **sales** and **purchase transactions**.

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

1. To support database management process in **Dagnarok**,your company asked you to provide some query that resulting important data. The requirements that asked from your company are:
2. Display Character Name, Character Level (obtained by adding ‘Lv ’ in front of the character level), Total STR (obtained from the total of STR for current used equipment), Total VIT (obtained from the total of VIT for current used equipment), Total AGI (obtained from the total of AGI for current used equipment), Total DEX (obtained from the total of DEX for current used equipment), and Total LUK (obtained from the total of LUK for current used equipment) for every character whose gender is male and the equipment is still used.
3. Display EquipmentID, EquipmentName, EquipmentPrice and Total Sales Income (obtained from total of equipment sold multiplied by equipment price) for every equipment which equipment type is either ‘Body’ or ‘Head’ and the Total Sales Income is lower than 1.000.000.
4. Display Day (obtained from the day of equipment used datetime, i.e. if the equipment used datetime is at ‘2019-01-02 12:50:24.000’ then the day is 2), Month (obtained from the month name of the equipment used datetime), CharacterName, Average STR (obtained from the average of STR for used equipment in decimal format), Average VIT (obtained from the average of VIT for used equipment in decimal format), Average INT (obtained from the average of INT for used equipment in decimal format), Average AGI (obtained from the average of AGI for used equipment in decimal format), Average DEX (obtained from the average of DEX for used equipment in decimal format), and Average LUK (obtained from the average of LUK for used equipment in decimal format) for every character that has used more than 1 equipment and the Average STR is more than 6.
5. Display NpcLocationName and Number Purchase Transaction (obtained from the number of purchase transaction ended with ‘ Transaction(s)’) for every NPC location that the NPC ever do purchase transaction on the first quarter and do purchase transaction with character that minimum level is more than 60.
6. Display CharacterId (obtained by replacing ‘CH’ with ‘Customer ’ from the character id), CharacterName, PurchaseDate (obtained from the purchase date in ‘dd mon yyyy’ format), Quantity (obtained from the quantity bought ended with ‘ item(s)’), EquipmentName, and EquipmentPrice for every sales transaction that the equipment is the most expensive that ever purchased by the character.

(**alias subquery**)

1. Display CharacterName, EquipmentName, STR, VIT, and UsedAt (obtained from the equipment used datetime in ‘Mon dd, yyyy’ format) for every character that’s currently using equipment that has the highest STR or highest VIT of all equipment and sort the data in descending order based on the UsedAt.

(**alias subquery**)

1. Display CharacterName, EquipmentName, and EquipmentPrice (obtained by adding ‘Rp. ’ in front of the price that in proper format. i.e. if the price is 150000, then the EquipmentPrice will be ‘Rp. 150,000,-’) for every character that’s currently using either the most expensive equipment of all equipment or the cheapest equipment of all equipment and the equipment is still used.

(**alias subquery**)

1. Display CharacterName, CharacterEmail, EquipmentName, EquipmentPrice (obtained by adding ‘Rp. ’ in front of equipment price), Quantity for sales transaction that has the most Total Income (obtained by the total of multiplication of equipment price and quantity sold) with at least 2 kinds of equipment sold.

(**alias subquery**)

1. Create a view named ‘**Profit per NPC**’ to display NpcName, Total Sales Income (obtained from the total of equipment price multiplied by quantity), and Average Sales Income (obtained from the average of equipment price multiplied by quantity) for every NPC that ever do sales transaction that occurred in weekdays (except Saturday and Sunday) and the Total Sales Income is more than the Average Sales Income.
2. Create view named ‘**Profit per Date**’ to display Transaction Date (obtained from SalesDate in ‘dd Mon yyyy’ format) and Total Profit (obtained by adding ‘Rp. ’ in front of the total of equipment price multiplied by quantity sold) for each sales transaction that has at least 2 equipments sold (not its quantity) and the total profit is more than 3.000.000.

**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