

Project Case
ISYS6169 | ISYS6279 | ISYS6280 | T0206
Database Systems

Software Laboratory Center: Binus University

O212-ISYS6169-JK02-00

Valid on *Odd Semester Year 2020/2021*

Revision:
with minor changes from Algorithm Bootcamp

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.

2. 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"

3. 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

4. Persentase penilaian untuk matakuliah ini adalah sebagai berikut:

Marking percentage for this subject is described as follows:

Tugas Mandiri <i>Assignment</i>	Proyek <i>Project</i>	UAP <i>Final Exam</i>
30%	30%	40%

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

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

Software <i>Software</i>
SQL Server Management Studio 18.5.1 SQL Server Developer 2019 Microsoft Office 365 Visual Paradigm Community Edition 16.1

6. 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 <i>Assignment</i>	Proyek <i>Project</i>	UAP <i>Final Exam</i>
SQL	SQL, VPP, Image Files (JPG / PNG)	SQL

BakMovie

BakMovie is the best website for selling movies managed by your friend, Baki. Baki and his friends manage all of activities that belongs to **BakMovie**. Users who are registered to the site can purchase movies online through the site. To improve **BakMovie's** management, Baki hired you as a database administrator are required to analyze and design a database for Bob's shop using **SQL Server Management Studio**.

Every user in **BakMovie** must follow the **sales transaction procedures**, which are:

- **User** that wants to buy a movie must have a personal information like **nickname, full name, email, city, and description**. Every **user** has an identification number with the following format:

<p style="text-align: center;">“USRXXX” X => number between 0 – 9</p>
--

- Every **sales transaction** made have all the information about **the user, the transaction date, the movie(s) purchased, and the quantity of each movie**. Every **sales transaction** has an identification number with the following format:

<p style="text-align: center;">“SALXXX” X => number between 0 – 9</p>
--

- User can purchase **more than one movie** in a sales transaction

Every movie in **BakMovie** must be registered the with the following information, which are:

- **Movie** to be sold must have a complete information like **the director, the publisher, title, description, release date, and price**. Every **movie** has an identification number with the following format:

<p style="text-align: center;">“MOVXXX” X => number between 0 – 9</p>
--

- Every movie **director** has **name, email, city, address, phone**, and an identification number with the following format:

<p style="text-align: center;">“DIRXXX” X => number between 0 – 9</p>
--

- Every movie **publisher** has **name, email, city, address, phone**, and an identification number with the following format:

<p>“PUBXXX”</p> <p>X => number between 0 – 9</p>

- Every **genre** has a **name** and an identification number with the following format:

<p>“GENXXX”</p> <p>X => number between 0 – 9</p>

- Every movie may have **more than one** genre.

Every user that is already registered in **BakMovie** can also give a **review** to a movie by following these procedures:

- User can only leave **one** review on every movie.
- Every movie **review** contains all information about **the user, the movie, recommendation status, review content**, and **review date**.

Notes:

- User nickname must be more than 5 characters.
- User email must have ‘@’ character (without quote).
- Publisher address must be more than 15 characters.
- Publisher phone must be numeric.
- Director email must have ‘@’ character (without quote).
- Director phone must be numeric.
- Movie description must be more than 20 characters.
- Movie release year must be between 2000 and 2016.
- Review content must be more than 20 characters.

Now **BakMovie** is still using manual management system to maintain the **sales transactions**. You as his precious friend wants to help **BakMovie** to create a database system that can store data and maintain the **sales transactions** and **review transactions**. The tasks that you must do are:

- a. Create **Entity Relationship Diagram** to maintain **sales transactions** and **review transactions**.
- b. Create a database system using **DDL syntax** that relevant with **sales transactions**. The database system must include database and tables with the required procedures.

- c. 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 to 10 data.
 - **Transaction** table must be filled with more than or equals to 15 data.
 - **Transaction detail / Mapping table** must be filled with more than or equals 25 data.
 - For the **Recommendation Status** in **Review** table, must be filled with: **Recommended** or **Not Recommended**.
- d. Rika asked you to provide some query that resulting important data.
1. Display MovieID, MovieTitle, MovieDescription and Reviews Movie (obtained from the total review of that movie and ended with ' review(s)') for each movie that is reviewed by an user who lived in 'Bandung' or has a recommendation status of 'Not Recommended'.
 2. Display GenreID, GenreName, and Total Movie (obtained from the count of movie) for every movie which is developed by a director whose ID is between 'DIR004' and 'DIR008' and the movie is released in February.
 3. Display DirectorID, DirectorName, Local Phone (obtained by replacing DirectorPhone first character into '+62'), Movie Sold (obtained from the sum of quantity), and Total Transaction (obtained from the count of transaction) for every director whose ID between 'DIR003' and 'DIR009' and the Movie Sold is more than 20.
 4. Display UserNickname, User City (obtained from UserCity in uppercase format), Total Movie Purchased (obtained from the sum of quantity), and Movie Owned (obtained from the count of movie) for every user which ID is 'USR002' or 'USR003' and purchased a movie that is released on odd month.
 5. Display Numeric User Id (obtained from the last 3 characters of UserId), Nickname (obtained from UserNickname in uppercase format), and UserCity for every user who purchased a movie with a quantity higher than average of all movie quantity sold and the user nickname contains 'l' character.

6. Display MovieID, MovieTitle, PublisherName, and Publisher Email (obtained from the characters before '@' in PublisherEmail) for every movie which is sold with quantity more than the maximum quantity of all transaction that occurred on the 22nd day of that month.
7. Display DirectorID, DirectorName, Movie Title (obtained from MovieTitle in lowercase format), and Total Genre (obtained from the count of genre and ended with ' genre(s)') for every director who created a movie that release on a day before the maximum release day of every movie that is released on September.
8. Display UserNickname, User First Name (obtained from the first word of UserFullName), and Total Quantity (obtained from the sum of quantity) for every user who purchased a movie with Total Quantity greater or equal than the maximum quantity of all transaction that occurred on 20th day of the month.
9. Create a view named '**CustomUserViewer**' to display UserId, UserNickname, Maximum Quantity (obtained from maximum of quantity sold), and Minimum Quantity (obtained from minimum of quantity sold), for every transaction that occurred on the 19th day of the month and the user's nickname contains 'h' character.
10. Create a view named '**CustomPublisherViewer**' to display PublisherName, MovieTitle, Release Date (obtained from MovieReleaseDate in 'dd Mon yyyy' format), Total Purchase (obtained from the sum of quantity), and Minimum Purchase (obtained from the minimum of quantity) for every publisher from Jakarta and the movie was released in July.

Files 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 answer the 10 cases. (.sql)
5. Documentation of your project that contains the explanation for each query (.pdf)