Desain database Proyek Akhir

"Toko TCG"

Team Member:

Surya Fadli Alamsyah	5025221059
Mohmmad Hanif Furqan Aufa Putra	5025221161
Muhammad Alif Satriadhi	5025221188

Lecturer:
Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D

INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA

2023/2024

A. Skenario Awal

Dengan boomingnya Permainan Kartu Perdagangan (TCG) saat ini, seorang pengusaha yang juga penggemar TCG telah memutuskan untuk membuka Pusat Permainan bertema TCG. Menyadari kebutuhan akan proses transaksi yang efisien, pengusaha tersebut telah meminta bantuan seorang kenalan untuk merancang sistem database. Pusat Permainan ini akan menawarkan berbagai paket kartu dari berbagai seri (Pokemon, Yu-Gi-Oh, Magic The Gathering, dll.) dan menampilkan paket booster terbatas, dengan beberapa paket tersedia untuk dijual.

Selain itu, pusat ini juga menawarkan Papan Permainan. Pemiliknya adalah seorang geek, sehingga dia adalah penggemar besar yang suka mengoleksi kartu-kartu. Oleh karena itu, pemilik juga menjual berbagai macam barang dagangan seperti topi, gelang, dan barang lainnya agar tidak hanya pemain yang bisa menikmati toko tersebut tetapi juga masyarakat biasa.

Pusat permainan ini bisa dinikmati tidak hanya oleh para pemain; pemilik juga membuat area lounge di mana masyarakat biasa dapat berkumpul. Mereka dapat memesan makanan ringan dan minuman di sana. Pusat permainan ini memiliki keanggotaan di mana anggota dapat mendapatkan diskon untuk pembelian dengan jumlah tertentu berdasarkan peringkat keanggotaan mereka. Ada 3 peringkat keanggotaan: Emas, Perak, dan Perunggu. Keanggotaan peringkat Emas memerlukan pengeluaran minimal 1,5 juta rupiah dengan diskon 20%. Peringkat Perak memerlukan pengeluaran minimal 1,4 juta rupiah dengan diskon 15%, dan peringkat Perunggu memerlukan pengeluaran minimal 900 ribu rupiah dengan diskon 10%. Anggota harus menyediakan data pribadi seperti nama dan nomor telepon untuk mendapatkan kartu keanggotaan.

Pengusaha menginginkan sistem yang mencatat data karyawan, termasuk ID Karyawan, nama, jenis kelamin, nomor telepon, email, dan usia. Setiap karyawan dapat menangani beberapa transaksi oleh pelanggan. Karyawan akan mencatat apakah pelanggan adalah anggota, jumlah total item yang dibeli, total jumlah yang harus dibayar, dan metode pembayaran yang diinginkan.

Dengan sistem database yang kokoh, Pusat Permainan bertema TCG ini dilengkapi dengan baik untuk menyempurnakan operasinya dan meningkatkan pengalaman pelanggan. Karyawan dapat menangani transaksi dengan efisien dan mengelola data pelanggan, memastikan interaksi yang lancar di setiap titik sentuhan. Dengan memanfaatkan prosedur penyimpanan dan fungsi, tugas seperti menghitung diskon keanggotaan, mencatat transaksi, dan menghasilkan laporan menjadi otomatis, membebaskan waktu berharga bagi staf untuk fokus pada memberikan layanan yang luar biasa.

Pelanggan mendapatkan pengalaman yang personal, karena status keanggotaan dan riwayat transaksi mereka mudah diakses, memungkinkan untuk promosi dan penghargaan yang ditargetkan. Baik membeli paket kartu, barang dagangan, atau menikmati area lounge, pelanggan dapat mengharapkan transaksi yang lancar dan layanan yang efisien. Selain itu, kemampuan untuk memeriksa ketersediaan item dan memperbarui informasi pelanggan memastikan bahwa inventaris tetap akurat, dan catatan pelanggan tetap terbaru.

Manajemen Pusat Permainan dapat memperoleh wawasan berharga tentang kinerja bisnis melalui fungsi-fungsi seperti menghitung total penjualan dan mengevaluasi kinerja karyawan. Metrik-metrik ini memungkinkan pengambilan keputusan yang berinformasi dan memfasilitasi perencanaan strategis untuk mendorong pertumbuhan dan profitabilitas. Secara keseluruhan, integrasi fungsionalitas database canggih meningkatkan operasi Pusat Permainan, memupuk komunitas yang bersemangat dari para penggemar TCG dan pengunjung casual sama.

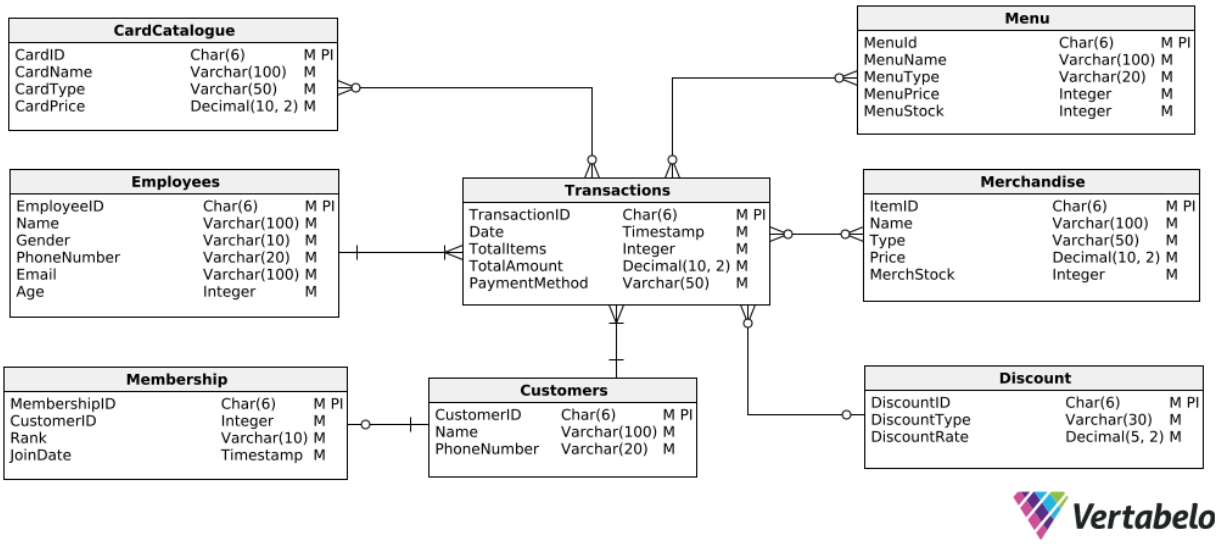
Asumsi:

- Diasumsikan bahwa pelanggan yang mengunjungi Pusat Permainan bertema TCG diharapkan untuk melakukan setidaknya satu transaksi yang melibatkan pembelian barang dari setidaknya tiga jenis produk yang tersedia di toko. Harapan ini mendorong para pengunjung untuk menjelajahi berbagai penawaran, termasuk paket kartu, barang dagangan, dan makanan/minuman dari area lounge.

- Selain diskon keanggotaan berdasarkan ambang batas pengeluaran, ada jenis diskon lain yang ditawarkan di Pusat Permainan. Diskon ini mungkin termasuk penawaran promosi, penjualan musiman, atau diskon acara khusus. Untuk menampung variasi ini, tabel terpisah untuk diskon, yaitu "Diskon," telah dibuat dalam skema database. Tabel ini memungkinkan manajemen yang fleksibel dan aplikasi berbagai jenis diskon di seluruh transaksi.

- Karyawan diharapkan untuk aktif berinteraksi dengan pelanggan selama kunjungan mereka ke Pusat Permainan. Interaksi ini melibatkan memberikan bantuan, menjawab pertanyaan, memproses transaksi, dan memastikan pengalaman keseluruhan yang positif bagi para pelanggan. Oleh karena itu, terdapat hubungan banyak-ke-banyak antara karyawan dan transaksi, yang menunjukkan bahwa beberapa karyawan dapat melayani beberapa pelanggan di berbagai transaksi. Penyusunan ini memastikan penyampaian layanan yang efisien dan kepuasan pelanggan sepanjang kunjungan mereka ke Pusat Permainan.

B. CDM



Database Detail:

Table:

1. CardCatalogue:

   Attributes: CardID (PK), CardName, CardType, CardPrice, CardStock
2. CardCatalogue_Transactions:

   Composite Primary Key: CardCatalogue_CardID, Transactions_TransactionID
3. Customers:

   Attributes: CustomerID (PK), Name, PhoneNumber
4. Discount:

   Attributes: DiscountID (PK), DiscountType, DiscountRate, Transactions_TransactionID (FK)
5. Employees:

   Attributes: EmployeeID (PK), Name, Gender, PhoneNumber, Email, Age
6. Employees_Transactions:

   Composite Primary Key: Employees_EmployeeID, Transactions_TransactionID
7. Membership:

   Attributes: MembershipID (PK), CustomerID (FK), Rank, JoinDate
8. Menu:

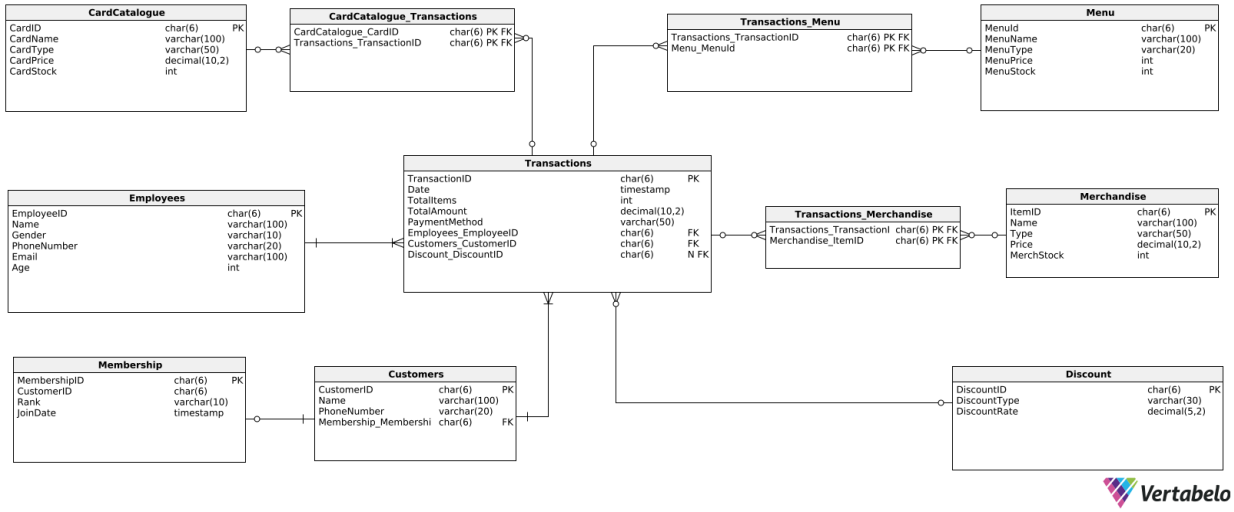
   Attributes: MenuId (PK), MenuName, MenuType, MenuPrice, MenuStock

9. Menu_Transactions:
   Composite Primary Key: Menu_MenuId, Transactions_TransactionID
10. Merchandise:
    Attributes: ItemID (PK), Name, Type, Price, MerchStock
11. Transactions:
    Attributes: TransactionID (PK), Date, TotalItems, TotalAmount, PaymentMethod, Customers_CustomerID (FK)
12. Transactions_Merchandise:
    Composite Primary Key: Transactions_TransactionID, Merchandise_ItemID

Relationships:

-CardCatalogue_Transactions(Many-to-Many):
->Cardinality:ManyCardCataloguetoManyTransactions
-Discount_Transactions(One-to-Many):
->Cardinality:OneDiscounttoManyTransactions
-Employees_Transactions(Many-to-Many):
->Cardinality:ManyEmployeestoManyTransactions
-Membership_Customers(One-to-Many):
->Cardinality:OneMembershiptoManyCustomers
-Menu_Transactions(Many-to-Many):
->Cardinality:ManyMenutoManyTransactions
-Transactions_Customers(One-to-Many):
->Cardinality:OneTransactiontoManyCustomers
-Transactions_Merchandise(Many-to-Many):
->Cardinality:ManyTransactionstoManyMerchandise

C. PDM



Database Detail:

Tables:

1. CardCatalogue
   ->Columns: CardID (PK), CardName, CardType, CardPrice, CardStock
2. CardCatalogue_Transactions
   ->Columns: CardCatalogue_CardID (FK), Transactions_TransactionID (FK)
3. Customers
   ->Columns: CustomerID (PK), Name, PhoneNumber
4. Discount
   ->Columns: DiscountID (PK), DiscountType, DiscountRate, Transactions_TransactionID (FK)
5. Employees
   ->Columns: EmployeeID (PK), Name, Gender, PhoneNumber, Email, Age
6. Employees_Transactions
   ->Columns: Employees_EmployeeID (FK), Transactions_TransactionID (FK)
7. Membership
   ->Columns: MembershipID (PK), CustomerID (FK), Rank, JoinDate
8. Menu
   ->Columns: MenuId (PK), MenuName, MenuType, MenuPrice, MenuStock
9. Menu_Transactions
   ->Columns: Menu_MenuId (FK), Transactions_TransactionID (FK)
10. Merchandise
    ->Columns: ItemID (PK), Name, Type, Price, MerchStock

11. Transactions
    ->Columns: TransactionID (PK), Date, TotalItems, TotalAmount, PaymentMethod, Customers_CustomerID (FK)
12. Transactions_Merchandise
    ->Columns: Transactions_TransactionID (FK), Merchandise_ItemID (FK)

Index:

- Index on CardCatalogue_Transactions for (CardCatalogue_CardID, Transactions_TransactionID)
- Index on Discount for Transactions_TransactionID
- Index on Employees_Transactions for (Employees_EmployeeID, Transactions_TransactionID)
- Index on Membership for Customers_CustomerID
- Index on Menu_Transactions for (Menu_MenuId, Transactions_TransactionID)
- Index on Transactions for Customers_CustomerID
- Index on Transactions_Merchandise for (Transactions_TransactionID, Merchandise_ItemID)

Foreign Keys:

- Foreign key constraint on CardCatalogue_Transactions referencing CardCatalogue
- Foreign key constraint on CardCatalogue_Transactions referencing Transactions
- Foreign key constraint on Discount referencing Transactions
- Foreign key constraint on Employees_Transactions referencing Employees
- Foreign key constraint on Employees_Transactions referencing Transactions
- Foreign key constraint on Membership referencing Customers
- Foreign key constraint on Menu_Transactions referencing Menu
- Foreign key constraint on Menu_Transactions referencing Transactions
- Foreign key constraint on Transactions referencing Customers
- Foreign key constraint on Transactions_Merchandise referencing Transactions
- Foreign key constraint on Transactions_Merchandise referencing Merchandise

D. Function and Procedure

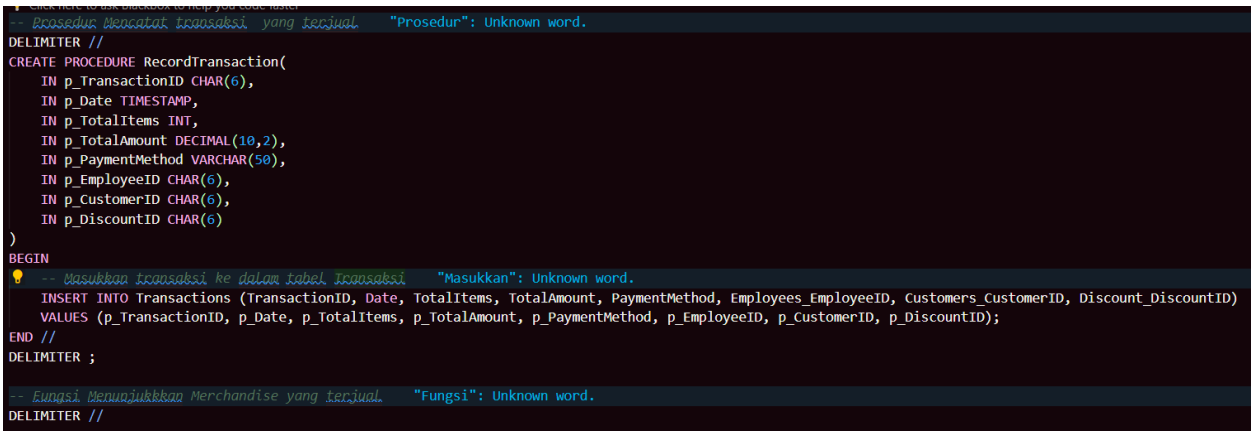
Dalam rancangan database ini, kami memperkenalkan beberapa fungsi dan prosedur untuk memfasilitasi operasional dan pelaporan yang efisien:

1. Prosedur Rekam Transaksi (RecordTransaction):
   Deskripsi: Prosedur ini bertujuan untuk merekam detail transaksi ke dalam database setelah setiap transaksi baru.
   Parameter: ID Transaksi (transaction_id), Tanggal (date), Total Barang (total_items), Total Jumlah (total_amount), Metode Pembayaran (payment_method), ID Pegawai (employee_id), ID Pelanggan (customer_id), ID Diskon (discount_id).
   Langkah-langkah:

   a. Masukkan data transaksi baru ke dalam tabel Transaksi dengan menggunakan nilai parameter yang diberikan.
   b. Rekam semua informasi terkait transaksi seperti tanggal, total barang, total jumlah, metode pembayaran, ID pegawai, ID pelanggan, dan ID diskon.
   c. Pastikan bahwa data transaksi telah direkam dengan sukses di dalam tabel Transaksi.
   d. Jika proses rekaman berhasil, transaksi dianggap berhasil direkam dalam database.

   Dengan menggunakan prosedur ini, setiap kali terjadi transaksi baru, data transaksi akan secara otomatis direkam dalam database dengan mengikuti langkah-langkah yang telah ditentukan. Hal ini memastikan bahwa informasi transaksi tersimpan dengan benar untuk keperluan pencatatan dan analisis lebih lanjut.

SQL Code:



```
-- Prosedur Mencatat transaksi yang terjual    "Prosedur": Unknown word.
DELIMITER //
CREATE PROCEDURE RecordTransaction(
    IN p_TransactionID CHAR(6),
    IN p_Date TIMESTAMP,
    IN p_TotalItems INT,
    IN p_TotalAmount DECIMAL(10,2),
    IN p_PaymentMethod VARCHAR(50),
    IN p_EmployeeID CHAR(6),
    IN p_CustomerID CHAR(6),
    IN p_DiscountID CHAR(6)
)
BEGIN
    -- Masukkan transaksi ke dalam tabel Transaksi    "Masukkan": Unknown word.
    INSERT INTO Transactions (TransactionID, Date, TotalItems, TotalAmount, PaymentMethod, Employees_EmployeeID, Customers_CustomerID, Discount_DiscountID)
    VALUES (p_TransactionID, p_Date, p_TotalItems, p_TotalAmount, p_PaymentMethod, p_EmployeeID, p_CustomerID, p_DiscountID);
END //
DELIMITER ;

-- Fungsi Menunjukkan Merchandise yang terjual.    "Fungsi": Unknown word.
DELIMITER //
```

2. Prosedur Hitung Produk Terjual (CountSoldProducts):
   Deskripsi: Prosedur ini bertujuan untuk menghitung jumlah produk yang terjual dalam database setelah setiap transaksi produk baru dimasukkan.
   Parameter: Tidak ada.
   Langkah-langkah:
   a. Menyiapkan variabel untuk menyimpan jumlah produk yang terjual.
   b. Menghitung jumlah produk yang terjual dengan menghitung jumlah entri dalam tabel Transaksi_Merchandise.
   c. Mengembalikan jumlah produk yang terjual.

   Dengan menggunakan prosedur ini, kita dapat dengan mudah mengetahui jumlah total produk yang telah terjual setiap kali terjadi transaksi baru di pusat permainan bertema TCG. Hal ini membantu dalam melacak kinerja penjualan dan menghasilkan laporan yang akurat untuk analisis lebih lanjut.

SQL Code:

```
-- Fungsi Menunjukkkan Merchandise yang terjual
DELIMITER //

CREATE FUNCTION CountSoldProducts()
RETURNS INT
BEGIN
    DECLARE sold_products INT;
    -- Hitung jumlah total produk yang terjual
    SELECT COUNT(*) INTO sold_products
    FROM Transactions_Merchandise;
    -- Kembalikan jumlah total produk yang terjual
    RETURN sold_products;
END;
//
DELIMITER ;
```

   Trigger Setelah Penyisipan Transaksi (After_Insert_Transaction):

   Deskripsi: Trigger ini diaktifkan setelah setiap penyisipan data baru ke dalam tabel Transaksi.

Tindakan: Memanggil prosedur Rekam Transaksi untuk merekam detail transaksi ke dalam database setelah setiap transaksi baru dimasukkan.

Trigger ini memastikan bahwa setiap kali ada transaksi baru yang dimasukkan ke dalam database, prosedur Rekam Transaksi akan otomatis dipanggil untuk merekam detail transaksi tersebut. Hal ini membantu dalam pemeliharaan data transaksi yang akurat dan tepat waktu di pusat permainan bertema TCG.

Trigger Setelah Penyisipan Transaksi Barang Dagangan (After_Insert_Transaction_Merchandise):

Deskripsi: Trigger ini diaktifkan setelah setiap penyisipan data baru ke dalam tabel Transaksi_Merchandise.

Tindakan: Memanggil fungsi Hitung Produk Terjual untuk menghitung jumlah produk yang terjual setiap kali transaksi produk baru dimasukkan.

Trigger ini memastikan bahwa setiap kali ada transaksi produk baru yang dimasukkan ke dalam database, fungsi Hitung Produk Terjual akan otomatis dipanggil untuk menghitung jumlah produk yang terjual. Hal ini membantu dalam melacak kinerja penjualan produk dan menyediakan data yang relevan untuk analisis dan pelaporan di pusat permainan bertema TCG.

SQL Code:

a.

```
-- Obat pemicu untuk fungsi Count Sold Products
DELIMITER //
CREATE TRIGGER After_Insert_Transaction_Merchandise
AFTER INSERT ON Transactions_Merchandise
FOR EACH ROW
BEGIN
    -- Panggil fungsi CountSoldProducts setelah seti
    DECLARE total_sold INT;
    SET total_sold = CountSoldProducts();
    -- Secara opsional, Anda dapat melakukan tindako
END //
DELIMITER ;
```

```
CREATE TRIGGER After_Insert_Transaction
AFTER INSERT ON Transactions
FOR EACH ROW
BEGIN
    -- Panggil prosedur RecordTransaction setelah setiap transaksi baru dimasukkan    "Panggil": Unknown word.
    CALL RecordTransaction(NEW.TransactionID, NEW.Date, NEW.TotalItems, NEW.TotalAmount, NEW.PaymentMethod, NEW.Employees_EmployeeID, NEW.Customers_CustomerID, NEW.Discou
END;
DELIMITER ;
```

b.