Dokumentation av databasen HarysHedligaBeliar

Översikt

Detta databassystem är skapat med SSMS för att hantera användarregistrering, inloggning och säkerhetsfunktioner. Det inkluderar tabeller för roller, användare, inloggningsförsök och lösenordshantering. Systemet har stored procedures för inloggning samt lösenordshantering och rapporter för inloggningsförsök.

Tabeller och Kolumner

Schema: **hhb**

hhb.Roles - Roller för användare

RoleID - *Primary Key, INT, Identity* - Unikt ID för rollen

RoleName - NVARCHAR(50), UNIQUE, NOT NULL - Namn på rollen (Customer och Admin)

hhb.Users - Användare i sytemet

UserID - *INT, PK, Identity* - Unikt ID för användaren.

UserName - *NVARCHAR(50), UNIQUE, NOT NULL* - Användarnamn.

Email - *NVARCHAR(50), UNIQUE, NOT NULL* - Användarens e-postadress.

PasswordHash - *VARBINARY(64), NOT NULL* - Hashad version av lösenordet.

PasswordSalt - *NVARCHAR(50), NOT NULL* - Salt för lösenordshashning.

FirstName - *NVARCHAR(50), NOT NULL* - För- och efternamn.

LastName - *NVARCHAR(50), NOT NULL* - För- och efternamn.

Street - *NVARCHAR(100) , NOT NULL* - Adressuppgifter.

Zip - *NVARCHAR(20), NOT NULL* - Adressuppgifter.

City - *NVARCHAR(50), NOT NULL* - Adressuppgifter.

Country - *NVARCHAR(50), NOT NULL* - Adressuppgifter.

PhoneNumber - *NVARCHAR(50), NOT NULL* - Telefonnummer.

IPAdress - *NVARCHAR(50), NOT NULL* - Användarens senaste IP-adress.

IsVerified - *BIT, DEFAULT 0, NOT NULL* - Om kontot är verifierat

IsLocked - *BIT, DEFAULT 0, NOT NULL* - Om kontot är låst.

VerificationCode - *NVARCHAR(50)* Kod för e-post verifiering.

VerificationExpiry - *NVARCHAR(50), DATETIME* - utgångstid för koden

CreatedAt - *DATETIME, DEFAULT GETDATE(), NOT NULL* - Datum då användaren skapades.

RoleID - *INT, FK, DEFAULT 1, NOT NULL* - Roll kopplad till användaren.

hhb.LogginAttempts - Loggning av inloggningsförsök

AttemptID - *INT, PK, Identity* - Unikt ID för inloggningsförsök.

UserID - *INT, FK, NULL* - Användar-ID kopplat till försöket.

IpAdress - *NVARCHAR(50), NOT NULL* - IP-adressen vid försöket.

TimeAttempt - *DATETIME, DEFAULT GETDATE(), NOT NULL* - Tidsstämpel för försöket.

Success - *BIT, NOT NULL* - Indikerar om inloggningen lyckades.

hhb.PasswordReset - Lösenordsåterställning

PasswordResetID - *INT, PK, Identity* - Unikt ID för återställningen.

UserID - *INT, FK, NOT NULL* - Användarens ID.

ResetCode - *NVARCHAR(50), NOT NULL* - Engångskod för återställning.

CreatedAt - *DATETIME, DEFAULT GETDATE(), NOT NULL* - När återställningen skapades.

ExpiresAt - *DATETIME, NOT NULL* - När återställningen går ut.

IsUsed - *BIT, NOT NULL* - Indikerar om koden redan använts.

Stored Procedures

TryLogin - Hanterar Inloggning

Verifierar att e-post finns, att kontot är verifierat och inte låst när man loggar in

Hashar lösenordet med algoritmen SHA_512 och jämför med databasen

Låser kontot när man skriver fel lösenordet 3 gånger under 15 minuter

Loggar inloggningsförsök i LogginAttempts tabellen och i TempLoginLog temporära tabellen

ForgotPassword - Begär lösenordsåterställning

Skapar en återställningskod om en tidigare kod inte redan är aktiv och lagrar den i PasswordReset tabellen

Raderar även gamla koden innan en ny genereras

SetForgottenPassword - Begär lösenordsåterställning

Kontrollerar att återställningskoden är korrekt, giltig och oanvänd

Hashar och uppdaterar den nya lösenordet som man skriver in samt markerar att återställningskoden är använd

Rapporter

UserLoginReport

Hämtar senaste lyckade och misslyckade inloggningsförsök för varje användare

LoginAttemptsReport

Beräknar antal lyckade och misslyckade inloggningar per IP-adress och datum

Använder window functions för att optimera datainsamlingen

Optimering och prestanda

Indexring: Primary och foreign keys används för snabb sökning genom ett clusters index

Hashning och Saltning: Användning av algoritmen SHA2-512 med salt för säker lösenordshantering

Tillfällig tabell #TempLoginLog: Förbättrar prestanda vid inloggning genom att minska databas operationer

Window Functions i rapporter: Använder Window Functions för att undvika subfrågor och förbättrar prestanda för analyser

SQL-exekveringplaner: Använda sig av SQL-exekveringplaner för att testa prestanda och analysera hur databas hanteringen kommer att utföra en viss SQL-fråga

Förbättringsmöjligheter

Använda sig av rate limiting på inloggnings försöken för att skydda mot brute-force attacker.

Använda en krypterad kolumn för IP-adresser för att öka säkerheten

Sammanfattning

Detta system är designat med säkerhet och prestanda i åtanke. Genom indexering, hasning och optimerade SQL-frågor säkerställs effektiv hantering av användardata och inloggningsförsök. Vidare förbättringar kan göras med ytterligare indexering Prestanda kan testas genom analyser av SQL-ekeveringsplaner

Skriven av Danilo Jovanovic