Database administration & sikkerhed

Denne uges opgave handler om at administrere forskellige grader af adgang til en database i konteksten af en organisation, f.eks. en webstore. I skal opdele dataene så forskellige medarbejdere og kunder kan tilgå relevant information om køb. til det formål skal i også immpplementere en form for access control, som i som udgangspunkt selv må bestemme hvordan i løser. Det kan være et login, access tokens eller andet. Jeres kode skal udvikles på baggrund af jeres Bikestore datasæt fra uge 6 & 7 og præsenteres som en præsentation på fredag hvor i lægger vægt på hvordan i har håndteret adgangs rettigheder samt hvilke overvejelser i har haft i forbindelse med implementeringen af jeres DB administration.

Overvej også:

- Hvordan sikrer man at potentielle fjendtlige aktører ikke kan lave ændringer i databasen?
- Hvordan mindsker man risikoen for at medarbejdere eller systemer laver utilsigtede ændringer i databasen?
- Hvis man opbevarer sensitivt data, f.eks. i henhold til GDPR, hvordan sikrer man at kun relevante medarbejdere og systemer har adgang til det data?
- Brug gerne følgende diagram som inspiration: database_sikkerhed_conspiracy.excalidraw

Overvejelser

- Bemærkning om administration af sensitivt data?

  > For det andet, hvordan administrerer man at forskellige medarbejdere kun har adgang til at se og ændre data, der er relevant for deres arbejde.

- Hvordan kan det implementeres
  - I SQL med *grant permission...*
  - I et python lag rundt om
  - Er der en en ide i begge dele?
- Hvordan kan stored procedures anvendes til at undgå eks. fejlhåndtering af data eller SQL injections.
- Secrets management?
  - Hvordan undgår man at lække fx admin password mm.
- Giv dem nogen eksemper på lgrupper, der skal have forskellige access rights.
  - admin
  - customer

    - warehouse
    - analytics
    - ...
- Evt. skal de lave undergrupper.
- Hvilke access rights skal de have og hvorfor?
    - forklar/dokumentér

Læringsmål

Formålet med opgaven er at give en praktisk og reflekterende forståelse af:

- Hvorfor og hvordan man administrerer adgang til forskellige typer data.
- Separation mellem kundedata og virksomheds-/organisationsdata.
- Forskelle i adgangsbehov mellem brugertyper og organisationsniveauer.
- Hvordan sikkerhed kan tænkes ind fra starten i databasedesign og vedligeholdelse.

Implementering af rettigheder og håndtering af risiko

Overvej hvordan adgangskontrol kan implementeres:

- Direkte i SQL ved hjælp af `GRANT`, `REVOKE`, `ROLES`
- I applikationen der interagerer med databasen (f.eks. et Python-lag med validering)
- Hvilken metode giver bedst mening i jeres kontekst?
- Hvordan beskytter du systemet mod SQL-injections?
    - Brug af stored procedures til at styre adgang til bestemte funktioner og undgå direkte manipulation
- Hvordan undgår du utilsigtede datalæk og adgang til administratoroplysninger?
- Overvej om der er elementer af **separation of duties** og **principle of least privilege** principperne du kan anvende.

Forskellige grader af adgang til en database i konteksten af en organisation, f.eks. en webstore eller et Human-Resources system. Dette aspekt af data management er kritisk, fordi en database både kan indeholder data, der skal være offentligt tilgængeligt og sensitivt data, der ikke ikke skal være offentligt eller deles på tværs af alle organisationsniveauer. I eksempelvis en webstore, skal produktinformation være offentligt tilgængeligt, mens man kunne have sensitivt persondata på sine medarbejdere eller kunder. Det kan lyde selvindlysende den dag i dag, men det var det bestemt ikke inden man begyndte at regulere området, eksempelvis som i EU's GDPR (General Data Protection Regulation) fra 2018.

Mulige problemstillinger:

- Hvordan sikrer man at potentielle personer med dårlige intentioner ikke kan lave ændringer i databasen?
- Hvordan mindsker man risikoen for at medarbejdere eller systemer laver utilsigtede ændringer i databasen?
- Hvis man opbevarer sensitivt data, f.eks. i henhold til GDPR, hvordan sikrer man at kun relevante medarbejdere og systemer har adgang til det data?

Separation af datatyper

Når man arbejder med databaser bør et centralt aspekt af ens arbejdsgang være kontinuerlige overvejelser omkring hvillke datatyper der indgår i en given database samt hvem der kan tilgå den. GDPR vedrører især det der hedder persomfølsomme data, men noget kundedata kan også være omfattet det, samtidig kan forretningsdata være beskyttet gennem virksomheders fortrolighedsaftaler som Intellectual property. Typisk arnbejder virksomheder med 3 hovedkategorier af datatyper, der skal behandles ud fra forskellige forudsætninger.

- **Kundedata**: navn, e-mail, købshistorik
- **Forretningsdata**: lagerstatus, ordrebehandling, medarbejderdata
- **Personfølsomme data**: helbredsoplysninger, CPR-nummer, adgangslogs

Overvej hvordan disse datatyper bør opdeles og "silogøres" for at undgå god data sikkerhed. Et godt eksempel kan være Novo nordisk opgaven vi havde i uge 5. I kan også orientere jer i ugen pensum dokument for at indtænke dem i dataen.
