

Eksamensopgave i Databaser for udviklere maj 2020

*Opgaven skal løses og afleveres **individuel**. I må snakke sammen, men ikke lave parprogrammering, fælles formuleringer eller udveksle kode- eller rapportdele uanset hvordan dette sker.*

Bookingsystem til en hotelkæde.

Der skal konstrueres et bookingsystem til en hotelkæde, der har specialiseret sig i virksomhedsarrangementer og kurser – herunder generelt kaldt arrangementer

Arrangementerne varer mellem to og fem dage og afvikles altid som et sammenhængende antal dage og kun hele dage.

Hotellerne har alle et antal mødelokaler af varierende kapacitet (dvs hvor mange personer, der kan være i lokalet) og de råder også over et antal enkeltværelser.

Når en virksomhedskunde ønsker at afvikle et arrangement på et af hotellerne, henvender han sig og oplyser startdato, varighed i dage og antal deltagere. Hotellet accepterer forespørgslen hvis der i hele perioden er et tilstrækkeligt antal værelser ledige og man har et tilstrækkeligt stort mødelokale, der er ledigt i hele perioden. Hvis forespørgslen kan accepteres, bookes det ønskede antal værelser og det mindste ledige mødelokale, der er stort nok til deltagerantallet, bookes også.

Alle arrangementer er inklusive forplejning (morgenmad, frokost og aftensmad). Forplejningen starter med morgenmad første dag og slutter med frokost sidste dag. Som eksempel vil et 3 dages arrangement indbefatte 3 morgenmad, 3 frokost, 2 aftensmad og 2 overnatninger.

Der arbejdes med en prismodel for arrangementer, hvor man betaler en pris per dag per deltager. Denne pris kan være forskellig fra hotel til hotel og fra periode til periode (der er med andre ord historik på prisen). Der afregnes med den pris der er gældende for arrangementets første dag.

Da hotelkæden har nogle meget store kunder, har nogle af disse kunder rabat. Rabatten gemmes som et procenttal og fratrækkes den pris kunden ellers skulle betale.

Øvrige krav:

Grundlaget for databasen i dette system er de allerede nævnte krav udvidet med nedenstående krav

Krav D1

For at forenkle booking er det besluttet, at der skal være en redundant tabel indeholdende hotelid, dato samt det aktuelt ledige antal værelser på det pågældende hotel den pågældende dato.

Krav D2

Oprettelse af en bookning sker efter følgende use-case

- Bruger indtaster kundeid, navn på det ønskede hotel, antal deltagere, dato for arrangementets start og antal dage arrangementet varer
- Det kontrolleres om der et ledigt lokale med plads til alle deltagere og om der er ledige værelser til alle deltagere
- Hvis der er plads, udskrives hvilket lokale bookingen vil blive tildelt og brugeren vil blive bedt om at bekræfte, at han ønsker at gennemføre bookningen. Hvis der ikke er plads, udskrives en beklagelse til brugeren
- Når brugeren har bekræftet bookningen, gennemføres bookningen.

Spørgsmål til opgaven

Til besvarelsen ønskes følgende:

1. Et domæneklassediagram i UML og et tabeldesign svarende til systemets data. Der lægges vægt på, at mulighederne for at lade DBMS foretage validering af data udnyttes. Det er ikke tilladt at lave redundans ud over den redundans, der kræves i krav D1. Tabeldesignet kommenteres kort, attributter, hvis anvendelse ikke umiddelbart kan ses ud fra attributtens navn, kommenteres også. Det færdige script inklusive create tables, constraints og insert af testdata medtages i bilag. Der skal være tale om et script du selv har lavet – det må ikke være et script lavet af SQL-server.
2. Der er følgende regler om data, der ønskes implementeret i databasen
 - Hotellerne holder lukket i week-ender, det ønskes derfor, at der ikke kan oprettes bookinger, der indeholder en lørdag eller en søndag.
 - Der må ikke kunne oprettes priser med tilbagevirkende kraftDiskuter hvorledes de to ting kan kontrolleres i databasen og implementer det
3. Der ønskes valgt de tre indeks i denne database, du vil anse for at være de mest relevante. Der argumenteres for de valgte index'er og der laves creates til de pågældende index. Du skal finde index for det fulde, tænkte system i en produktionssituation, dvs med produktionsdata og alle funktionaliteter i det fulde system. Forklaringer og creates af de valgte index'er medtages i rapporten.
4. I forbindelse med den redundans, der omtales i krav D1, ønskes der en trigger, der ved indsættelse af en ny booking opdaterer det aktuelt antal ledige værelser korrekt. Du kan antage, at bookinger oprettes en efter en.
5. Der ønskes en stored procedure, der tager et bookingid som parameter og udskriver den samlede, endelige pris, som kunden skal betale for den pågældende booking
6. Lav en User defined function (UDF), der som parametre tager et hotelid, en startdato, en varighed og et antal personer og returnerer et id på det mindste lokale, der har plads til antallet af personer i hele perioden, såfremt der ikke er ledige lokaler, returneres null.
7. Denne delopgave har fokus på transaktioner og samtidighedskontrol. Oprettelse af en booking som beskrevet i krav D2 ønskes implementeret i en konsolapplikation i ADO og C# med transaktionslogik (herunder samtidighedskontrol). Der forklares i rapporten hvordan din løsning omkring samtidighedskontrol er lavet og en alternativ løsning forklares ligeledes. Den valgte løsning og alternativet sammenlignes. Der argumenteres for valg af isolation-level. Konsolapplikationen vedlægges i bilag.
8. Der ønskes konstrueret en konsol-applikation i C# som ved hjælp af Linq To Entities laver følgende:
 - Der ønskes en funktionalitet, der indlæser et navn på et lokale og udskriver virksomhedens navn, fradato, varighed og antal deltagere på alle fremtidige bookinger. Udskriften skal være sorteret på fradato. Funktionaliteten skal laves i to udgaver. En udgave, der anvender deferred execution og lazy loading og en udgave, der anvender immediate execution og eager loading
 - Der ønskes en funktionalitet, der kan oprette et nyt lokale. De nødvendige oplysninger indlæses fra tastatur
 - Der ønskes en funktionalitet, der indlæser en dato og udskriver hvor mange mennesker, der skal have henholdsvis morgenmad, frokost, aftensmad og overnatning den pågældende dag

Ved alle tre funktionaliteter skal der anvendes link-attributter (navigation properties) og der må ikke anvendes stored procedures, udf'er eller views.

Konsolapplikationen vedlægges udskrevet i bilag.

Vurdering af databasedelen

Der anvendes ved bedømmelsen nedenstående vægte af de enkelte opgaver. Det er dog væsentligt at fastslå, at der også indgår en helhedsvurdering.

De ovennævnte databasespørgsmål indgår med følgende vægt

- Spørgsmål 2, 3, 4, 5 og 6 udgør hver 10%
- Spørgsmål 1 og 8 udgør hver 15 %
- Spørgsmål 7 udgør 20 %

Krav til aflevering af rapport i databaser for udviklere

Rapporten

Rapportens udformning og indhold skal være dækkende for det udførte arbejde. Rapporten skal således omfatte:

- Kortfattet indledning, hvor evt. uklarheder præciseres, og overordnede beslutninger forklares og begrundes.
- Et domæneklassediagram i UML.
- Besvarelse af de 8 spørgsmål.
- En konklusion.

En del af konklusionen skal være en opsummering af, hvad der er nået og hvad der ikke er nået.

Rapportens udformning og indhold skal være velstruktureret. Rapporten, uden bilag, må maksimalt fylde 8 sider.

Bilag til rapporten skal som minimum indeholde

- samtlige databasescripts (alle creates, insert af testdata, stored procedures, UDF og triggere)
- de to konsolapplikationer svarende til opgave 7 og 8 – (afleveres i pdf-format).

Aflevering

Der skal afleveres på WISEFLOW, se separat mail herfra.

Afleveringen indeholder

- rapporten incl. bilag i én pdf-fil

Opgaven udleveres mandag den 25. maj 2020, kl. 08.30, men offentliggøres på CANVAS søndag den 24. maj om eftermiddagen

Besvarelsen skal uploades til WISEFLOW senest fredag den 29. maj 2020 kl. 16.00.

NB. Hvis det i forløbet bliver nødvendigt at udsende yderligere information, vil dette ske til jeres skolemail. Check derfor denne jævnligt.