

EKSAMENSFORSIDE

Hjemmeeksamen

Emnekode: DAT1000/DAT1000N	Emnenavn: Databaser 1	
Emneansvarlig: Bjørn Kristoffersen	Campus: Bø/Bakkenteigen/Nett	Fakultet: Handelshøyskolen
Utlev. dato og tidspunkt i WISEflow: 25.05.2021 kl. 09:00		Innlev. dato og tidspunkt i WISEflow: 25.05.2021 kl. 13:00
Antall oppgaver: 3	Antall vedlegg: Ingen	Ant. sider inkl. forside og vedlegg: 4
Hjelpemidler og samarbeid: Tillatte hjelpemidler: Alle hjelpemidler er tillatt.		
	Ja	Nei
Er det individuell eksamen?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Er det tillatt å samarbeide med andre personer?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Beskrivelse av individuell eksamen og samarbeid finner du på min.usn.no/eksamen		
Kriterier for besvarelsen:		
Skrifttype:	Skriftstørrelse:	Linjeavstand:
Antall ord (min/maks):	Maks antall sider ekskl. forside og vedlegg:	
Kildehenvisning: Bruk en av de vanlige referansestilene, f.eks. APA, beskrevet her: https://bibliotek.usn.no/oppgaveskriving/litteraturliste-og-kildehenvisninger/		
Vekting av oppgavene ved sensur: <ul style="list-style-type: none"> • Oppgave 1: 35 % • Oppgave 2: 35 % • Oppgave 3: 30 % 		
Du bør løse oppgave 1 før oppgave 2, fordi SQL-spørsmålene i oppgave 2 bygger på datamodellen du skal utvikle i oppgave 1. Vektingen gir også en god pekepinn til hvor mye tid du bør sette av til hver oppgave. Legg ved dine egne forutsetninger / tolkinger hvis oppgaveteksten er uklar eller tvetydig. Prøv å svare på så mange spørsmål som mulig. Lykke til!		

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Oppgave 1 (35 %)

I denne oppgaven skal du tegne en logisk datamodell (ER-diagram). Diagrammet skal vise entiteter og attributter med datatyper, forhold med kardinaliteter samt primærnøkler og fremmednøkler. Du skal tegne identifiserende forhold med heltrukne linjer og ikke-identifiserende forhold med stiplede linjer. Du kan tegne diagrammene med MySQL Workbench, eller du kan tegne for hånd (bruk da notasjonen i læreboken) og deretter ta et bilde av modellen, som du limer inn i besvarelsen.

Datamodellen skal lages på en slik måte at SQL-spørringene beskrevet i oppgave 2 lar seg løse.

Beskrivelse av systemet

Padel er en slags blanding av tennis og squash og er en sport i sterk vekst. PadelSprett er en (tenkt) kjede som drifter padelhaller over hele landet. De trenger et nettbasert registreringssystem og du skal designe databasen til dette systemet.

En padelmatch spilles som double, altså to spillere på hvert lag. Du trenger ikke å kjenne til alle reglene. Det holder å vite at en match spilles best av 3 sett (eller omganger), som betyr at man kan vinne enten med 2-0 eller 2-1.

For å spille på en av banene til PadelSprett må man registrere seg som spiller. Man oppgir telefonnummer, navn og et selvvalgt passord. Passord blir kryptert før de lagres i databasen. Man velger også et abonnement. Et abonnement har både en grunnpris pr. måned og en timepris, og det finnes noen få ulike typer, tilpasset de som spiller ofte og de som kun spiller en gang i mellom.

En padelhall kan ha flere baner. Disse er alltid nummererte fra 1 og oppover i hver hall. For hver hall skal man lagre navnet på hallen og telefonnummeret til en ansvarlig kontaktperson. Registrerte spillere kan reservere tid på ledige baner. En reservasjon gjøres alltid i hele timer, f.eks. fra kl. 18:00 til kl. 20:00, på en bestemt dato.

Spillerne legger selv inn resultatet av kamper de spiller. Man legger da inn dato, hvilke spillere som var med på de to lagene og hva som ble resultatet. Det er mulig å legge inn kampresultater selv om de ikke blir spilt på en av banene til PadelSprett, f.eks. kamper spilt på feriereiser.

Hvis man ikke har noen å spille med, så er det mulig å invitere andre til å bli med på en reservasjon. Spillere kan søke etter reservasjoner med ledige plasser og melde sin interesse. Den som opprettet reservasjonen velger hvem som får bli med og kan også lukke reservasjonen når tilstrekkelig antall spillere har meldt seg. Det kan skje at flere enn fire spillere deltar på en reservasjon.

For å gjøre det lettere å finne noen å spille med, skal det være mulig å følge andre spillere. Man blir varslet om reservasjoner med ledige plasser fra alle man følger.

Oppgave 2 (35 %)

I denne oppgaven skal du skrive SQL-spørringer mot databasen du laget datamodell for i oppgave 1. Hvis du har laget modellen med MySQL Workbench, så bør du vurdere å lage en database med «forward engineer», slik at du kan teste ut SQL-spørringene. Men hvis du ikke er trygg på hvordan man gjør dette, så er det kanskje bedre bare å skrive SQL-spørringer uten å teste. Pass også på at du ikke bruker for lang tid på å teste en enkelt spørring, kanskje det bare er en liten detalj som er feil.

Hvis du oppdager at datamodellen du laget i oppgave 1 ikke inneholder nok opplysninger, eller er uegnet for å svare på en av SQL-oppgavene under, og du ikke har tid til å rette opp datamodellen, så kan du på et slikt spørsmål beskrive et tillegg eller en endring i datamodellen, f.eks. ved å skrive: «Jeg antar i dette spørsmålet at databasen inneholder følgende tabeller...». For full uttelling må slike tillegg gi en hensiktsmessig og normalisert database.

Tips: MySQL-dokumentasjonen har forklaring med eksempler til ulike funksjoner. På denne samlesiden er alle funksjonene ordnet etter datatype: <https://dev.mysql.com/doc/refman/8.0/en/functions.html>

2-a (5 %)

Skriv en SQL-spørring som viser spillere med etternavn som starter med «M» og inneholder flere enn 3 bokstaver. Sørg for at fornavn og etternavn blir vist i én kolonne, adskilt med et blankt tegn. Sorter utskriften først med hensyn på etternavn og deretter på fornavn (hvis etternavnene er like).

2-b (5 %)

Skriv en SQL-spørring som viser antall kamper som er spilt i hver padelhall siste 30 dager. Ta med telefonnummeret til hallen i utskriften.

2-c (5 %)

Øk grunnpris pr. måned på abonnementet «Hobby» med 5 %.

2-d (5 %)

Skriv en SQL-spørring som viser beløpet hver spiller skal betale en bestemt måned, som er lik grunnpris pr. måned pluss summert pris for reserverte timer, basert på det abonnementet spilleren har. Velg måned selv.

2-e (5 %)

Skriv en SQL-oppgave som tester evne til å bruke *delspørringer* med **NOT EXISTS**, og løs deretter denne oppgaven. Prøv å lage en oppgave som er nyttig. Legg vekt på at du formulerer spørsmålet så presist som mulig. Krevende SQL-oppgaver vil gi mer uttelling enn enkle.

2-f (5 %)

Skriv en SQL-oppgave som tester evne til å kombinere *gruppering* og *ytre koblinger*, og løs deretter denne oppgaven. Prøv å lage en oppgave som er nyttig. Legg vekt på at du formulerer spørsmålet så presist som mulig. Krevende SQL-oppgaver vil gi mer uttelling enn enkle.

2-g (5 %)

Lag et view som viser alle padelmatchene dette året fram til dagens dato. Ta med etternavnet på alle spillerne i utskriften.

Oppgave 3 (30 %)

3-a (10 %)

Databasen fra oppgave 1 skal utvides med mulighet for å bestille noen få produkter, som f.eks. padelracketer og padelballer. For å få til dette så blir det foreslått å legge til følgende tabell:

- Bestilling(PNr, Navn, Pris, SNr, Fornavn, Etternavn, Tidspunkt, Antall)

Eksempelrad:

- (15, 'Padelracket Pro X', 949.00, 104, 'Peder', 'Aas', '2021-05-16 15:39', 2)

Her er 15 produktnummeret til padelracketen med navn Padelracket Pro X. Denne racketen koster kr. 949.00. Spiller 104, med navn Peder Aas, kjøpte 2 stykker av denne rackettypen 16. mai 2021 kl. 15:39.

Forklar kort med bruk av konkrete eksempler hva som er uheldig med den nye tabellen. Skriv deretter ned funksjonelle avhengigheter, bestem kandidatnøkkel og utfør normalisering til BCNF. Få fram hvert normaliseringssteg. Du skal ikke legge til flere kolonner. Vis primærnøkler med understreking og merk fremmednøkler med en stjerne i resultatet.

NB! Vis alle tabeller og kolonner i resultatet uavhengig av hvilke tabeller du valgte i oppgave 1.

3-b (10 %)

Det er ofte hensiktsmessig å gi ulike brukere forskjellige rettigheter i en database. Bruk databasen du laget i oppgave 1 som eksempel og foreslå aktuelle rettigheter for ulike brukere. Begrunn valgene. Vis også eksempler på SQL-kommandoer som oppretter brukere og tildeler disse rettigheter.

3-c (10 %)

Det finnes ulike måter å begrense hva som skal være lovlige verdier i en databasetabell. Forklar så mange som mulig av slike teknikker med konkrete og relevante eksempler fra databasen du laget i oppgave 1.