

Tilbakemelding gruppe: 2

Oppgave 1

a)

SQL-koden for å opprette tabellen passering er ikke levert, men dere har skrevet begrunnelse for valg av datatyper i tabellen.

Det er fornuftig å skille dato og klokkeslett i to kolonner, men begrunnelsen for det er noe uklar. Dere bør referere til konkret plassbehov. Det er riktig at DATE og TIME hver for seg tar mindre plass (3 bytes) enn DATETIME (5 bytes), men DATE og TIME **til sammen** bruker 6 bytes. For å lagre både dato og klokkeslett må dere jo ha både en DATE kolonne og en TIME-kolonne. DATETIME kan lagre dette i én og samme kolonne. Derfor må plassbehovet for disse to til sammen som må sammenliknes med én DATETIME-kolonne.

StNr: Bra begrunnelse for å endre datatypen i oppgaveteksten. (Den skulle egentlig ha vært INT, men feil versjon av skriptet ble lagt ut.) Det er riktig at TINYINTs maksimale verdi er 127 hvis man skal lagre både positive og negative verdier (-128 til +127). I dette tilfellet bør alle stasjonsnummer være positive slik dere skriver. Det finnes egne varianter av datatypene for å lagre bare positive verdier. Ved å bruke datatypen TINYINT **UNSIGNED** kan kolonnen lagre verdier mellom 0 og 255 (1 byte) som er nok i dette tilfellet.

Dere skriver ikke noe om bakgrunnen for valg av datatype for Belop. Her **kan** TINYINT UNSIGNED brukes hvis man forutsetter at prisene for én passering ikke vil øke til mer enn 255 kr, men det kan være en urealistisk antagelse, så SMALLINT som dere har valgt er nok bedre.

b)

Utgangen er korrekt, men dere bør forklare litt mer. Hvorfor beregner dere plassbehovet for RegNr til 10 bytes + 1-2 bytes? Med UTF-8 tegnsatt vil mange tegn kreve 2 bytes pr tegn. Vil det være naturlig at $10/2=5$ er gjennomsnittlig lengde på registreringsnummer?

c)

INSERTene er korrekt, men GRANT-setningene hører ikke hjemme her. Det er ikke slik at eiere av kjøretøy også skal være brukere av databasen og ha rettigheter til denne.

Oppgave 2

a) Korrekt.

b) Korrekt syntaks i SQL setningene, men GRANT ALL gir alle rettigheter til tabellene i databasen. Dette omfatter mye mer enn å registrere og vedlikeholde data i alle tabeller. GRANT ALL gir altså for mange rettigheter i forhold til oppgaven. SELECT, INSERT, UPDATE og DELETE er det som er nødvendig for å kunne registrere og vedlikeholde data i tabeller i databasen.

c) Korrekt

Oppgave 3

a) Funksjonen gir riktig resultat og hensyntar rushtid korrekt. Når dere har brukt SMALLINT som datatype for beløp i tabellen passering, bør denne funksjonen også returnere SMALLINT slik at det er samsvar mellom datatypene. Hva skjer hvis RegNr eller StNr ikke finnes i databasen? Da vil SELECT-setningen med JOIN ikke gi noen rader som svar, og metoden returnere NULL. Det er for så vidt ok og logisk, men det kan også være lurt å vise dette tilfellet eksplisitt i koden, f.eks. først sjekke om disse verdiene finnes og evt. eksplisitt returnere NULL da.

b) Korrekt - bra.

c) Prosedyren setter inn en ny passering korrekt, men har likevel noen mangler i forhold til oppgavens punkt om «så god feilhåndtering som mulig»: Den sjekker om RegNr og StNr finnes i databasen, men gir ikke spesifikke feilmeldinger i hver av disse av situasjonene. Den sjekker heller ikke om ny_dato og ny_Tidspunkt inneholder lovlige verdier.

Prosedyren mangler også transaksjonshåndtering med START TRANSACTION, COMMIT og ROLLBACK. Siden prosedyren gjør endringer i databasen, bør den ha det.

Dessuten behøver ikke kolonnen beløp å være med i INSERT-setningen. Dette settes jo automatisk av triggeren i 3b. (Mulig at dere har definert beløp som NOT NULL og dere får en feilmelding om det hvis dere fjerner den?)

d) Morsom og kreativ måte å sette inn testdata på!

Oppgave 4

a) Korrekt

b) Korrekt

c) Spørringen er korrekt, men jeg er litt overrasket over at dere fikk type=index i kjøreplanen på denne. (Hadde vært fint om dere viste kjøreplanen i besvarelsen.) Hvis dere ikke allerede har opprettet en indeks på kolonnen Dato burde planen vist type=ALL. Siden dere ikke har levert koden for tabellen passering ser jeg heller ikke hvordan primærnøkkelen er definert. Dersom Dato er første kolonne i primærnøkkelen vil indeksen kunne brukes på spørringen, ellers ikke.

Dere skriver også "*Index er typen som blir brukt hvis selve indeksen inneholder alle kolonnene som skal vises.*". Det gjør den jo ikke i dette tilfellet, siden dere SELECTer alle kolonner (*) i tabellen. Aksessmetode *index* kan også brukes dersom indeksen **ikke** inneholder alle kolonner som skal vises, men i så fall er det for å unngå sortering av data, dvs. hvis spørringen bruker ORDER BY på en indeksert kolonne.

Til slutt skriver dere: "*Optimalisatoren bruker index fordi det er en sammenligning (Altså WHERE) i selve spørringen.*" Det er heller ikke riktig. Se f.eks. spørringene i a) og b). Begge disse har WHERE i spørringen, men bruker ikke aksessmetoden *index*, men *ref* og *range*. Hvis spørringen har en WHERE som sammenlikner med en kolonne som **ikke** er indeksert, kan ikke noen indeks brukes, og optimalisatoren vil bruke type=ALL (Full Table Scan).

d) Denne oppgaven ble litt vanskelig for dere hvis dere fikk en kjøreplan med type=index i c). Spørringen burde ha gitt type=ALL fordi det ikke er noen indeks på kolonnen Dato. Måten å redusere tidsforbruket på er altså å opprette en slik indeks:
CREATE INDEX PasseringDatoIDX ON passering(Dato) ;

Når dere blir spurt om hva som kan gjøres for å redusere tidsforbruket for en SQL-spørring er svaret som regel **aldri** å endre selve spørringen. Det er jo optimalisatorens jobb å finne raskeste måte å utføre spørringen på, og selve SQL-spørringen kan sjelden påvirke dette uten at resultatet av spørringen endres. Spørringen dere skriver i 4d bruker altså like lang tid som den i 4c. Hvilke kolonner som skal "vises" påvirker ikke tidsbruken.

e) SQL-spørringen er korrekt og gir riktig resultat.

Oppgaven ber dere vise kjøreplanen på **tabellform**, mens dere har vist den **grafisk** form (visual). Det er det en del informasjon fra tabellformen som ikke tas med og vises grafisk, bl.a. hvilke verdier som brukes for oppslag i indeksene. Dette får dere da heller ikke forklart.

Dere forklarer greit betydningen av de enkelte opplysningene i den visuelle planen. Det gir ekstra bonus om dere også klarer å få fram sammenhengene mellom de ulike delene. Hvilken rekkefølge leses tabellene i? Hvorfor brukes de ulike aksessmetodene?

Bø 28.10.2022

Jon Kvisli