

Gruppeopgave del 3

Afleveres senest ved øvelserne d. 21. april.

I denne opgave skal I arbejde videre med jeres "case" fra første og anden gruppeopgave, og programmere en række SQL forespørgsler, views, etc. I skal skrive en SQL sætning for hvert af nedenstående punkter, som hver specificerer et bestemt krav til sætningen. I skal for hvert punkt aflevere en kort beskrivelse, en udskrift af SQL sætningen, og tilhørende output fra databasen.

I skal i gruppen diskutere, hvilke forespørgsler der kan stilles til jeres database, som vil kunne opfylde kravene. Vælg et eksempel, **formuler spørgsmålet i naturligt sprog, og formuler derefter SQL til det**. For punkt 18-21 skal der ikke formuleres spørgsmål, men i stedet kort gøres rede for, hvilket formål denne funktionalitet kunne tjene i jeres case. Alle løsninger skal anvende tabeller fra jeres database. Hvis det er nødvendigt, må I gerne tilføje nye tabeller til databasen. Hvis I ændrer på tabellerne (entiteterne/attributterne), skal jeres datamodel fra delopgave 1 og 2 revideres.

1. Anvend data fra mindst to forskellige tabeller og som minimum SELECT og ORDER BY.
2. Join en tabel med sig selv og anvend GROUP BY og HAVING.
3. Anvend data fra en eller flere tabeller, DISTINCT og WHERE ... IN.
4. Anvend EXISTS og minimum to tabeller
5. Anvend 3 subqueries inde i hinanden
6. Anvend en korelleret subquery.
7. Sammenlign værdierne i en tabel T1 med værdierne i en anden tabel T2 og find alle værdier i T1, som er større end gennemsnittet i T2.
8. Anvend "COUNT(DISTINCT x) > y", hver x og y er attributter.
9. Anvend mindst 3 tabeller og en UNION
10. Anvend mindst 3 tabeller og en INTERSECTION.
11. Anvend mindst 3 tabeller og en UNION og en INTERSECTION.
12. Anvend en join, en betingelse og en GROUP BY.
13. Brug COUNT, MAX og AVG i en forespørgsel med minimum 3 tabeller
14. En forespørgsel der tæller antal og og gennemsnit for hver gruppe i en tabel.
15. Anvend to tabeller og en LEFT OUTER JOIN.
16. Lav en "derived table" (subquery) i FROM listen og anvend den sammen med en anden tabel til at undersøge om noget er i en mængde.
17. Anvend to NOT EXISTS inde i hinanden.
18. Lav et view, hvor der er beregnet et eller flere felter.
19. Lav et view på en eller flere af jeres tabeller, hvor de data man få adgang til at se afhænger af, hvem man er.
20. Giv en bruger adgang til en af tabellerne i jeres database.
21. Fjern brugerens adgang til tabellen fra nr. 21.

Besvarelsen skal have en forside, hvor der skal stå alle gruppemedlemmers navne, gruppens nr., hjælpelærens navn og navnet på jeres case. **Vedlæg desuden de rettede besvarelser af delopgave 1 og 2, samt et Oracle brugernavn og password, så hjælpelæreren kan afprøve forespørgsler på jeres data.**