

Udlånsregistrering af skolens bærbare computere (del 1 af 3) - databasediagram

Fokus:

Design af en relationel database med korrekt navngivning og normalisering.

Opgavebeskrivelse:

Du skal udvikle et **database-diagram** til et system, der registrerer udlån af skolens bærbare computere. Diagrammet skal implementeres i en **Oracle-database**, og designet skal overholde principperne for korrekt navngivning og normalisering.

Krav til systemet:

1. **Om computere og mus:**
 - Fabrikat (f.eks. DELL)
 - Model (f.eks. Inspiron 9300)
 - Computernummer (f.eks. PC-0722)
 - Mus (optisk/alm/nej)
2. **Om brugerne:**
 - Elevnummer
 - Navn
 - Adresse
 - Postnr. og by
 - CPR-nummer
 - Email
 - Stamklasse
3. **Ved udlån:**
 - Elevnummer
 - Computernummer
 - Udlånsdato
 - Udløbsdato

Funktionaliteter:

- Registrering af udlån.
- Oversigt over, hvem der har computere udlånt lige nu, og deres afleveringsdato.
- Oversigt over lånere, der har overskredet afleveringsfristen.
- Mulighed for at sende mails til lånere, der ikke har afleveret til tiden.
- Oversigt over tilgængelige computere.
- Historisk oversigt over tidligere udlån.

Udvidelsesmuligheder:

- Registrering af undervisere.
- Reservering af computere (en underviser kan reservere et antal computere i en bestemt periode).
- Grafiske udlånsstatistikker.
- "Blacklisting" af lånere, der ikke afleverer computere til tiden.

Aflevering:

- ZIP-fil, der indeholder:
 - Et **database-diagram**, som opfylder ovenstående krav og viser relationerne mellem tabellerne i systemet.

OBS:

Systemet skal bruge en relationel database, Oracle. Spørg din instruktør, hvis du er i tvivl om noget.

Frist:

Opgaven skal forevises **efter 5 arbejdsdage**.

Forklaringer til opgaven:

Når opgavestiller ønsker at bruge Oracle som databasesystem, betyder det, at det **database-diagram**, du udvikler, ikke kun skal være kompatibelt med den relationelle struktur, men også med de **datatyper**, som Oracle understøtter. Dette omfatter blandt andet datatyper som:

- **VARCHAR2** til tekststrengene (f.eks. navne, adresser).
- **NUMBER** til tal (f.eks. priser, antal).
- **DATE** til datoer (f.eks. udlånsdatoer).
- **CLOB** (Character Large Object) til store tekstdata, hvis der er behov for lange beskrivelser.

Det er vigtigt at vælge de rigtige datatyper til hver kolonne for at sikre optimal ydeevne og kompatibilitet med Oracle. Dette vil sikre, at din database fungerer korrekt inden for Oracles system.

Med hensyn til korrekt **navngivning og normalisering** betyder det følgende:

1. **Korrekt navngivning:**

- **Tabeller og kolonner** skal have logiske og sigende navne, der præcist beskriver deres indhold eller funktion. Eksempelvis skal en tabel, der indeholder oplysninger om bøger, hedde "Books" eller "Computer", og kolonnerne skal hedde noget som "Fabricator", "Model", osv.
- Undgå uforståelige eller forkortede navne. Brug hele ord som "StudentNumber" i stedet for "SN" eller "CompNum".

2. Normalisering:

Normalisering er en proces, hvor du organiserer data i tabeller for at minimere data duplikering og sikre dataintegritet. Et normaliseret database-design følger disse regler:

- **1. normalform:** Fjern gentagende grupper af data og sørg for, at hver kolonne indeholder kun én type værdi.
- **2. normalform:** Sikrer, at hver tabel kun indeholder data, der er direkte relateret til den primære nøgle.
- **3. normalform:** Eliminere afhængighed af ikke-nøgle attributter, så data kun afhænger af primærnøglen.

Kort sagt, din opgave er at designe en database, der undgår unødvendig data duplikering og sikrer, at dataene er lette at opdatere og vedligeholde.