

L1. Tabeller og enkle spørringer

- Her er noen oppgaver som dere skal løse i smågrupper i løpet av forelesningen.
- Ta vare på svarene og ta dem gjerne med på eksamen.

a) Hva slags datatype passer til kolonnen Tittel? INTEGER – DOUBLE – CHAR(10) – VARCHAR(100) – DATE – BOOLEAN	b) Hvor mange ulike verdier kan lagres i én byte? 2 – 8 – 80 – 100 – 256 – 512 – 1000
c) Vis FNr, Tittel og Sjanger fra tabell Film. SELECT FNr,, Sjanger FROM Film;	d) Vis alle opplysninger fra tabell Film. SELECT FROM
e) Vis tittel på filmer med aldersgrense lik 7 år. SELECT FROM Film WHERE =	f) Vis alle actionfilmer som har aldersgrense under 18 år. SELECT FROM WHERE=..... AND<.....;
g) Vis en sortert liste med produksjonsland fra tabellen Film. Hvert landnavn skal kun bli vist én gang og kun med store bokstaver. SELECT (.....) FROM Film ORDER BY	h) Vis titler på filmer produsert før 1990 i USA og Norge. Sortér med hensyn på produksjonsår. SELECT FROM Film WHERE ((..... = 'USA') (..... = 'Norge')) < ORDER BY

Tabellen Film

FNr	Tittel	Ar	Land	Sjanger	Alder	Tid	Pris
1	Casablanca	1942	USA	Drama	15	102	149.00
2	Fort Apache	1948	USA	Western	15	127	NULL
3	Apocalypse Now	1979	USA	Action	18	155	123.00
4	Streets of Fire	1984	USA	Action	15	93	NULL
5	High Noon	1952	USA	Western	15	85	123.00
6	Cinema Paradiso	1988	Italia	Komedie	11	123	NULL
7	Asterix hos britene	1988	Frankrike	Tegnefilm	7	78	149.00
8	Veiviseren	1987	Norge	Action	15	96	87.00
9	Salmer fra kjøkkenet	2002	Norge	Komedie	7	80	149.00
10	Anastasia	1997	USA	Tegnefilm	7	94	123.00
11	La Grande bouffe	1973	Frankrike	Drama	15	129	87.00
12	The Blues Brothers	1980	USA	Komedie	11	133	135.00
13	Beatles: Help	1965	England	Musikk	11	144	NULL

L2. Lage databasetabeller

- Her er noen oppgaver som dere skal løse i smågrupper i løpet av forelesningen.
- Ta vare på svarene og ta dem gjerne med på eksamen.

a) Tegn linjer fra rader i tabellen Country til «tilhørende» rader i tabellen City. Ser du et «mønster»? En-til-en? Mange-til-en? En-til-mange? Mange-til-mange?	b) Skriv på hva som bør være primærnøkler (PK) og fremmednøkler (FK) for de to tabellene. En av tabellene må få en ny kolonne. Hvilken? Fyll inn eksempeldata i denne kolonnen.
c) Skriv SQL-kode for å lage tabellen Country. CREATE TABLE Country (CountryCode, Name, Area, PRIMARY KEY (.....));	d) Skriv SQL-kode for å lage tabellen City. CREATE TABLE City (CityNo INTEGER, Name, PRIMARY KEY (.....));
e) Hva skjer hvis vi prøver å sette inn en ny rad i tabellen Country med disse verdiene: (CountryCode = 'NO', Name = 'Danmark', Area=42925)	f) Og hva skjer hvis vi prøver å sette inn en ny rad i tabellen City med disse verdiene: (CityNo = 4, Name = 'New York', CountryCode = 'US')
g) De to tabellene er «koblet» ved at primærnøkkelen i tabell A er «kopiert» og blitt til fremmednøkkel i tabell B. Kunne vi «snudd» på det, altså primærnøkkelen i tabell B hadde blitt primærnøkkel i tabell A?	h) Gjør oppgave c på nytt, men sørg for at brukeren ikke kan legge inn to like landnavn og at han/hun alltid må legge inn areal. CREATE TABLE Country (CountryCode, Name, Area, PRIMARY KEY (.....));

Tabellene City og Country

CityNo	Name	?
1	Oslo	
2	Bergen	
3	Stockholm	

CountryCode	Name	Area	...?
NO	Norge	385207	
SE	Sverige	449964	

L3. Likekoblinger og gruppering

- Her er noen oppgaver som dere skal løse i smågrupper i løpet av forelesningen.
- Ta vare på svarene og ta dem gjerne med på eksamen.

a) Hvor mange rader inneholder kryssproduktet av tabellene City og Country?	b) For hvor mange av «radkombinasjonene» i kryssproduktet fra oppgave a) er hver by «koblet» til landet de tilhører?
c) Hvorfor er vi nesten aldri interessert i følgende spørring: SELECT * FROM City, Country;	d) Utvid SQL-spørringen fra oppgave c) med en WHERE-betingelse slik at vi får en naturlig likekobling. SELECT FROM City, Country WHERE =;
e) Skriv likekoblingen fra oppgave d) ved hjelp av INNER JOIN notasjon. SELECT FROM INNER JOIN ON =;	f) Skriv en SQL-spørring som viser antall byer i hvert land. Det holder å vise landkoden og antallet. SELECT,(....) AS FROM City GROUP BY
g) Prøv å beskrive en spørring der du både må koble tabeller og bruke gruppering.	h) Hvordan skriver man likekoblinger mellom tre tabeller ved hjelp av INNER JOIN notasjon?

Tabellene City og Country

CityNo	Name	CountryCode
1	Oslo	NO
2	Bergen	NO
3	Stockholm	SE
4	Sandefjord	NO
5	Uddevalla	SE

CountryCode	Name	Area
NO	Norge	385207
SE	Sverige	449964
DK	Danmark	42925

L4. Innsetting, oppdatering og sletting

- Her er noen oppgaver som dere skal løse i smågrupper i løpet av forelesningen.
- Ta vare på svarene og ta dem gjerne med på eksamen.

a) Skriv SQL-kode for å sette inn ny by Paris i nytt land Frankrike.	b) Hva må endres i oppgave a) hvis CityNo er autonummerert?
c) Det har skjedd en systematisk målefeil i kolonnen Country.Area; alle verdier må økes med 500 kvadratkilometer.	d) Skriv SQL-kode som sletter alle byene i Norge.
e) Skriv SQL-kode som legger inn et nytt land Tyskland, og lag deretter en spørring som viser alle land med alle sine byer. Alle land skal med i utskriften, også land som det ennå ikke er registrert byer til. Tips: Bruk en ytre likekobling.	f) Skriv en SQL-spørring som finner antall byer i hvert land. Alle land skal med i utskriften, også de som det ennå ikke er registrert byer til.
g) Tenk at du skal lage et SQL-skript som oppretter en database med tabellene City og Country – med eksempeldata. Hvilke kommandoer vil du bruke og i hvilken rekkefølge?	

Tabellene City og Country

CityNo	Name	CountryCode
1	Oslo	NO
2	Bergen	NO
3	Stockholm	SE
4	Sandefjord	NO
5	Uddevalla	SE

CountryCode	Name	Area
NO	Norge	385207
SE	Sverige	449964
DK	Danmark	42925

L5. Datamodellering med ER

- Her er noen oppgaver som dere skal løse i smågrupper i løpet av forelesningen.
- Ta vare på svarene og ta dem gjerne med på eksamen.

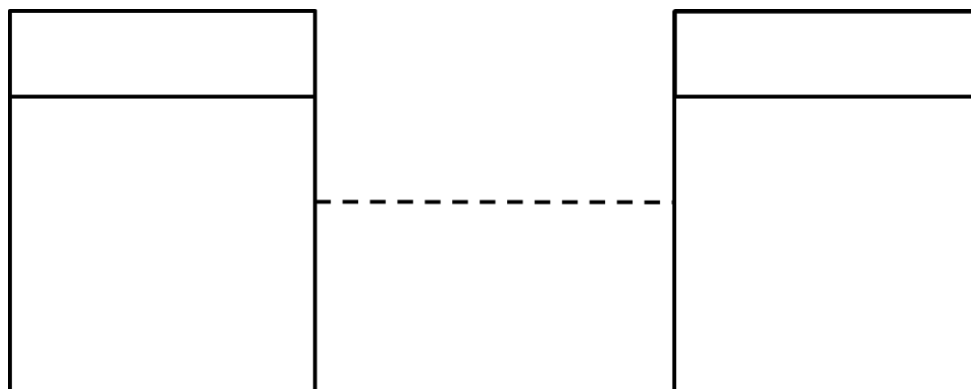
a) Bruk City og Country som eksempel. Skriv navn på entitetene (tabellene) som overskrifter i det tomme ER-diagrammet.	b) Skriv deretter inn alle attributtene (kolonnene) i nedre del av entitetsboksene.
c) Skriv (PK) etter attributtene (kolonnene) som er primærnøkler.	d) Skriv (FK) etter attributtene (kolonnene) som er fremmednøkler.
e) Sett på «kråkeføtter» og «streker» for å få fram om forholdet mellom City og Country er en-til-en, en-til-mange, mange-til-en eller mange-til-mange (maksimumskardinalitet).	f) Utvid diagrammet med minimumskardinalitet (0 eller på innsiden av maksimum).
g) Hva slags forhold vil du si det er mellom studenter og emner? En-til-en, en-til-mange, mange-til-en eller mange-til-mange? Hva med forholdet mellom emner og studier?	h) «Forward engineering» innebærer å gå fra ER-diagram til SQL-skript med CREATE TABLE (enkelt sagt). «Reverse engineering» er det motsatte. Hva kan disse to teknikkene brukes til?

Tabellene City og Country

CityNo	Name	CountryCode
1	Oslo	NO
2	Bergen	NO
3	Stockholm	SE
4	Sandefjord	NO
5	Uddevalla	SE

CountryCode	Name	Area
NO	Norge	385207
SE	Sverige	449964
DK	Danmark	42925

Logisk ER-diagram

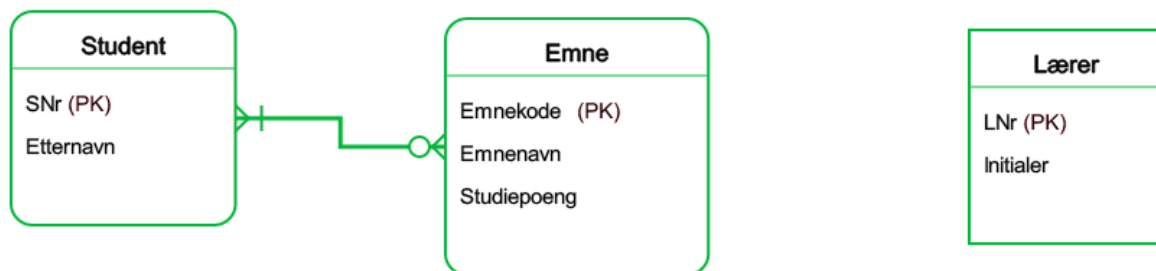


L6. Fra modell til database

- Her er noen oppgaver som dere skal løse i smågrupper i løpet av forelesningen.
- Ta vare på svarene og ta dem gjerne med på eksamen.

a) Utvid den konseptuelle datamodellen med et forhold mellom Emne og Lærer. Et emne kan (på denne skolen) kun ha én lærer, men en lærer kan undervise flere emner.	b) Du skal nå oversette til logisk datamodell. Start med å tegne opp entitetene på nytt.
c) Sett også på forholdet mellom Emne og Lærer.	d) Løs opp mange-til-mange forholdet mellom Student og Emne med en koblingsentitet og to nye forhold.
e) Utvid datamodellen med fremmednøkler.	f) For å forstå koblingstabeller og identifiserende forhold: Tegn opp tabeller med eksempeldata og trekk linjer mellom rader som «hører sammen».

Konseptuell/begrepsmessig datamodell



Logisk datamodell

Tabeller med eksempeldata

L7. Normalisering

- Her er noen oppgaver som dere skal løse i smågrupper i løpet av forelesningen.
- Ta vare på svarene og ta dem gjerne med på eksamen.

a) Kan du si noe om hvilke funksjonelle avhengigheter som gjelder i eksempeltabellen? Er det mulig at $A \rightarrow B$ holder? Hva med $A \rightarrow C$? Eller $B \rightarrow C$? $C \rightarrow B$? $E \rightarrow C$?	b) La oss gi kolonnene mer meningsfulle navn. $A=ByNr$, $B=PersonNr$, $C=ByNavn$, $D=Etternavn$, $E=Fødselsdato$. Tabellen heter Bosted. Hvilke funksjonelle avhengigheter gjelder?
c) Bestem kandidatnøkkel for tabellen.	d) Utfør normalisering til BCNF. Merk primærnøkler og fremmednøkler i sluttresultatet.
e) Hvilken normalform er eksempeltabellen på?	f) Hvordan kan man bruke normaliseringsteori når man designer en ny database?

Eksempeltabell

A	B	C	D	E
10	75	Oslo	Hansen	23.05.1974
14	98	Bergen	Nilsen	16.07.1992
11	50	Stockholm	Nilsen	02.12.2001
10	32	Oslo	Andersen	30.10.1999
14	77	Bergen	Hansen	23.05.1974

L8. Subtyper og modelleringsmønstre

- Her er noen oppgaver som dere skal løse i smågrupper i løpet av forelesningen.
- Ta vare på svarene og ta dem gjerne med på eksamen.

a) Et universitet vil lagre opplysninger om ansatte og studenter. Om alle personer vil man lagre personnummer og navn. Om ansatte vil man også lagre stilling og lønn. Om studenter vil man også lagre antall studiepoeng. Tegn ER-diagram med subtyper.	b) Vis hvordan eksempeldataene under kan representeres på tabellform, når man bruker de tre ulike måtene å implementere subtyper på i en relasjonsdatabase.
c) Ordre og Ordelinje er et eksempel på modelleringsmønsteret master/detail – eller hode/linje. Gi noen eksempler på dette mønsteret fra andre systemer/databaser.	d) Et innlegg på et diskusjonsforum inneholder et nr, en overskrift og selve teksten. Dessuten er et innlegg svar på et annet innlegg. Vis ER-modellen for Innlegg og vis hvordan innlegg lar seg representere i en tabell.
e) Både diskusjonsforum og slektstrær er eksempler på hierarkier. Hvordan kan slektstrær representeres på tabellform – inspirert av diskusjonsforumet?	f) Hva er fordelen med å kjenne til vanlige modelleringsmønstre?

Eksempeldata

- Ansatt Per Hansen har personnummer 01019012345 og lønn 750.000.
- Student Mari Mo har personnummer 10100154321 og 65 studiepoeng.
- Student Ove Li har personnummer 03030333221 og 25 studiepoeng.

L9. Views og indekser

- Her er noen oppgaver som dere skal løse i smågrupper i løpet av forelesningen.
- Ta vare på svarene og ta dem gjerne med på eksamen.

a) Skriv SQL-kode for å lage et view CityWithCountryName som viser kolonnene City.CityNo, City.Name, Country.Name og Country.Area, for alle byer i Norge og Sverige. Kolonnen City.Name skal omnavnes til CityName og Country.Name omnavnes til CountryName.	b) Skriv SQL-kode for å lage et view CityCount som viser antall byer i hvert land. View'et skal ha tre kolonner: CountryCode, CountryName og NrOfCities.
c) Hva skjer hvis vi ber databasesystemet om å utføre følgende spørring: UPDATE CityWithCountryName SET Area = Area + 50;	d) Og hva skjer hvis vi ber databasesystemet om å utføre denne spørringen: UPDATE CityCount SET NrOfCities = NrOfCities – 1 WHERE CountryCode = 'NO';
e) Lag en SQL-spørring som viser antall land som det er registrert flere enn 2 byer i. <i>Tips:</i> Spørring mot view.	f) Bruk SQL for å lage en indeks som effektiviserer søk på bynavn.

Tabellene City og Country

CityNo	Name	CountryCode
1	Oslo	NO
2	Bergen	NO
3	Stockholm	SE
4	Sandefjord	NO
5	Uddevalla	SE

CountryCode	Name	Area
NO	Norge	385207
SE	Sverige	449964
DK	Danmark	42925

L10. Databaseadministrasjon

- Her er noen oppgaver som dere skal løse i smågrupper i løpet av forelesningen.
- Ta vare på svarene og ta dem gjerne med på eksamen.

a) Lag to brukere per og kari, samt to tilhørende databaser perdb og karidb.	b) Gi per og kari alle rettigheter på hver sin database.
c) Gi dessuten kari leserettigheter på tabellen Kunde i databasen hobbyhuset.	d) Lag en rolle student.
e) Gi rollen student leserettigheter på alle tabellene i databasen hobbyhuset.	f) Gi rollen student til per.
g) Trekk tilbake leserettigheten på tabellen Kunde i databasen hobbyhuset fra bruker kari.	h) Hva skal til for at bruker per skal kunne gi sine rettigheter videre til andre brukere?
i) Hva er de viktigste oppgavene til et databasehåndteringssystem?	j) Nevn noen viktige oppgaver for en databaseadministrator.

Eksempelkommandoer

DROP DATABASE IF EXISTS db;

CREATE DATABASE db;

USE db;

DROP USER 'ola'@'%';

CREATE USER 'ola'@'%' IDENTIFIED BY 'hemmelig';

CREATE USER 'ola'@'localhost' IDENTIFIED BY 'hemmelig';

GRANT SELECT ON db.* TO 'ola'@'%';

GRANT SELECT ON db.tab1 TO 'ola'@'%' WITH GRANT OPTION;

GRANT ALL PRIVILEGES ON db.* TO 'ola'@'%';

REVOKE DELETE ON db.* FROM 'ola'@'%';

CREATE ROLE 'innsyn'@'%';

GRANT SELECT ON db.* TO 'innsyn'@'%';

GRANT 'innsyn'@'%' TO 'ola'@'%';

L11. Relasjonsmodellen og avansert SQL

- Her er noen oppgaver som dere skal løse i smågrupper i løpet av forelesningen.
- Ta vare på svarene og ta dem gjerne med på eksamen.

a) Lag en SQL-spørring som for alle filmer viser tittel og anbefalt målgruppe («barn», «ungdom» eller «voksen») basert på aldersgrense.	b) Lag en SQL-spørring som finner alle filmer som koster mer enn gjennomsnittet.
c) Lag en SQL-spørring som finner de dyreste filmene innen hver sjanger.	d) Lag en SQL-spørring som bruker operatoren IN for å vise alle filmer produsert i USA, Italia og Frankrike.
e) I hvilke situasjoner er det aktuelt å bruke NOT EXISTS?	f) Bruk relasjonsalgebra for å vise FNr og Tittel på filmer produsert før 1960.

Tabellen Film

FNr	Tittel	Ar	Land	Sjanger	Alder	Tid	Pris
1	Casablanca	1942	USA	Drama	15	102	149.00
2	Fort Apache	1948	USA	Western	15	127	NULL
3	Apocalypse Now	1979	USA	Action	18	155	123.00
4	Streets of Fire	1984	USA	Action	15	93	NULL
5	High Noon	1952	USA	Western	15	85	123.00
6	Cinema Paradiso	1988	Italia	Komedie	11	123	NULL
7	Asterix hos britene	1988	Frankrike	Tegnefilm	7	78	149.00
8	Veiviseren	1987	Norge	Action	15	96	87.00
9	Salmer fra kjøkkenet	2002	Norge	Komedie	7	80	149.00
10	Anastasia	1997	USA	Tegnefilm	7	94	123.00
11	La Grande bouffe	1973	Frankrike	Drama	15	129	87.00
12	The Blues Brothers	1980	USA	Komedie	11	133	135.00
13	Beatles: Help	1965	England	Musikk	11	144	NULL