Oppgave 1A:

Tog har normalformen BCNF, fordi tognr er en supernøkkel. Her har vi en relasjon på formen X->Y, hvor x er tognr og y er resten av attributtene i Tog.

TogTabell har normalformen BCNF, fordi (tognr, avgangstid) er en supernøkkel. Dette ser vi da vi har en relasjon på formen X->Y, hvor x er (tognr, avgangstid) og y er resten av attributtene i TogTabell.

Siden vi har flere FD'er i Plass må vi følge reglen om å ta den laveste normalformen som FD'ene tilfredsstiller for å finne normalformen. I Plass-relasjonen har vi to FD'er:

- (dato, togNr, vognNr, plassNr) -> vindu, ledig tilfredstiller BCNF, da (dato, togNr, vognNr, plassNr) er en kandidatnøkkel.
- FD'en (TogNr, vognNr, plassNr) -> vindu derimot tilfredstiller 1NF, fordi (TogNr, vognNr, plassNr) er en delmengde i en kandidatnøkkel. (TogNr, vognNr, plassNr) er ikke en kandidatnøkkel i seg selv, hvilket gjør at relasjonen ikke tilfredstiller BCNF. Vindu er heller ikke med i en kandidatnøkkel.

Vi ser dermed at den høyeste normalformen som Plass tilfredstiller er 1NF.

Oppgave 1B:

Før vi utfører FD-operasjonen:

	TogNr	**VognNr**	**PlassNr**	**Vindu**	**Ledig**	**Dato**
TogNr, vognNr, plassNr, vindu	TogNr	VognNr	PlassNr	Vindu	Ledig 1	Dato 1
Dato, togNr, vognNr, plassNr, ledig	TogNr	VognNr	PlassNr	Vindu 2	Ledig	Dato

	TogNr	**VognNr**	**PlassNr**	**Vindu**	**Ledig**	**Dato**
TogNr, vognNr, plassNr, vindu	TogNr	VognNr	PlassNr	Vindu	Ledig	Dato
Dato, togNr, vognNr, plassNr, ledig	TogNr	VognNr	PlassNr	Vindu	Ledig	Dato

<u>Vi har en rad som er uten signifikator, ergo er dekomposisjonen tapsfri.</u>

SETE er på BCNF. Den har en relasjon på formen X-> Y, hvor X er primærnøkkel. Her er X (togNr, vognNr, plassNr), mens Y er vindu. Vi ser at (togNr, vognNr, plassNr) identifiserer enhver rad i SETE.

Plass er på BCNF. Den har en relasjon på formen X-> Y, hvor X er primærnøkkel. Her er X (dato, togNr, vognNr, plassNr), mens Y er ledig. Vi ser at (togNr, vognNr, plassNr) identifiserer enhver rad i SETE.

I denne relasjonen bevares ikke primærnøkkelen til den opprinnelige tabellen(Plass). Dette gjør at primærnøkkelen må korrigeres i enhver rad til Sete, hvilket også impliserer mot at alle rader i tabellen må oppdateres.

Oppgave 1C:

∏(vognnr σ(tognr = 401 and dato like '20080610' (Sete ►◄ Plass)))
∏(vognnr σ(tognr = 401 and dato like '20080610' and vindu = 'true' and ledig = 'false' (Sete ►◄ Plass)))

Oppgave 1D:

∏(tognr σ (tognr = tognr and stasjon = stasjon and avgangsTid != avgangstid(TogTabell)))) = Ø;

Oppgave 2A:

Relasjonen har kandidatnøkkelene BCF, ACF og CDEF.

Oppgave 2B:

Den høyeste normalformen som R tilfredsstiller er 2NF.

Oppgave 2C:

Før chasealgoritmen er utført:

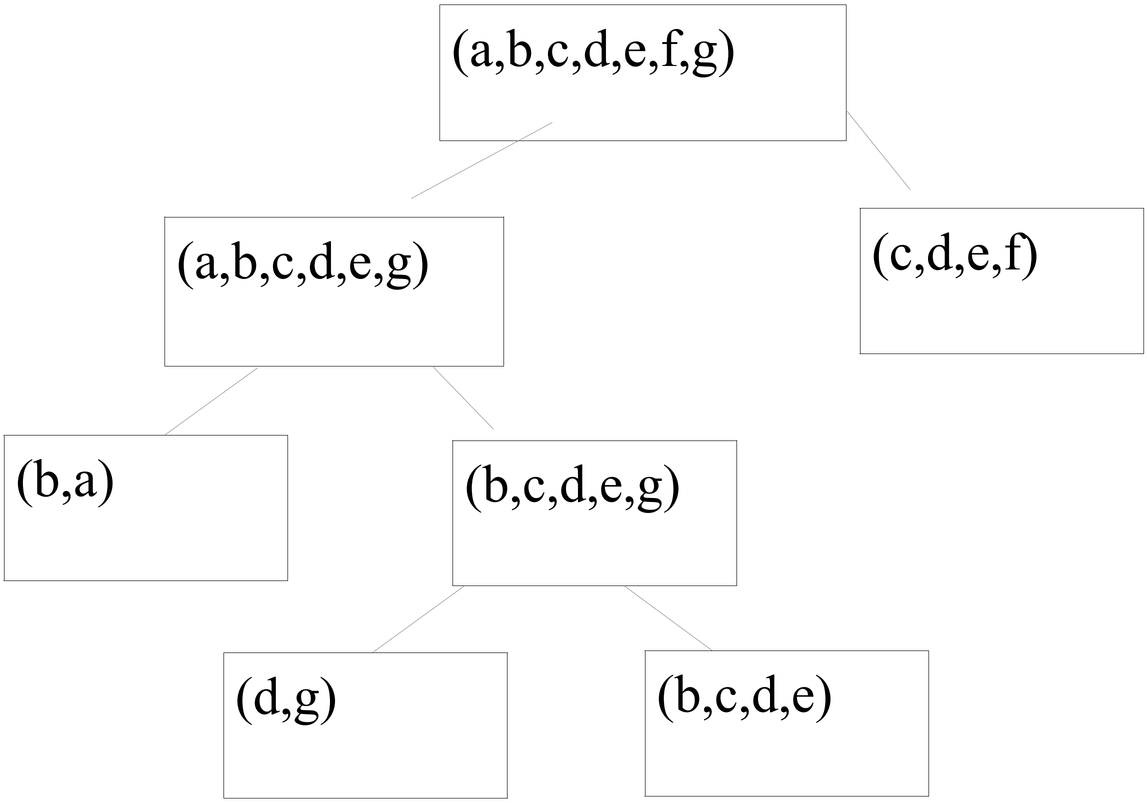
	A	B	C	D	E	F	G
ABF	A	B	C1	D1	E1	F	G1
ACF	A	B2	C	D2	E2	F	G2
BCDE	A3	B	C	D	E	F3	G3
DG	A4	B4	C4	D	E4	F4	G

Etter at chasealgoritmen er utført:

	A	B	C	D	E	F	G
ABF	A	B	C1	D1	E1	F	G1
ACF	A	B	C	D2	E2	F	G2
BCDE	A	B	C	D	E	F3	G
DG	A4	B4	C4	D	E4	F4	G

Etter at man har utført Chase-algoritmen ferdig er det ingen rad med signifikatorer.

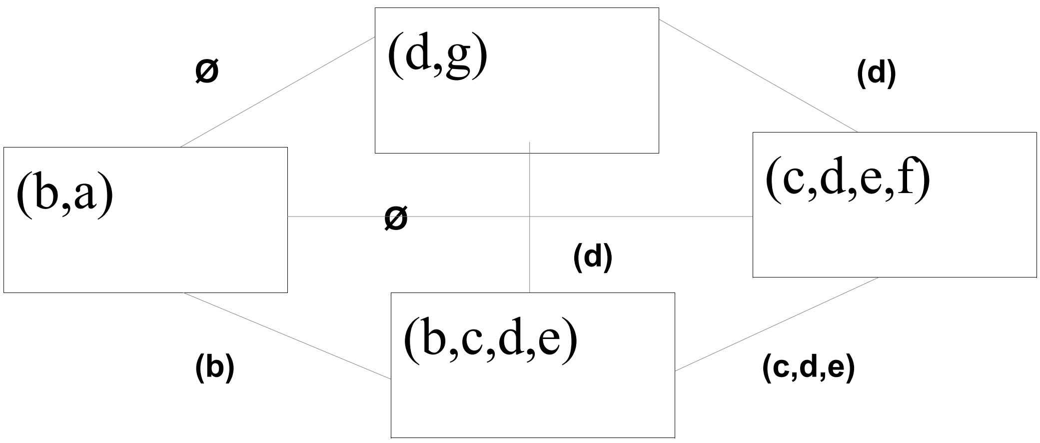
Oppgave 2D:



Relasjonene vi har etter dekomposisjonen er:
R1(b,a)
R2(d,g)
R3(b,c,d,e)
R4(c,d,e,f)

Dekomposisjonen er ikke FD-bevarende, da alle Fder etter dekomposisjonen kan ikke sjekkes lokalt i en av de nye relasjonene.

Snittgrafen er syklisk. Det finnes en global instans der de lokale instansene er projeksjonen av den globale instansen. Dette ser vi at vi ikke kan ta snittet av relasjoner R2 med R3 eller ta snittet av R1 med R3. Relasjonene kan dermed inneholde noen støyinstanser.



Oppgave 2E:

	A	B	C	D	E	F	G
R(a,b,c,d,e,f,g)	A	B	C	D	E	F	G
R(c,d,f)	A2	B2	C	D	E2	F	G

Vi har ingen rader uten signifikator etter å ha utført chasealgoritmen, ergo holder ikke cdf->b.

Oppgave 2F:

Utvid Q med MVDen DG ->->AC. Vis at CDF → B nå følger fra Q.

Vi har en MVD dersom hvis y er delmengde i X, XY er samtlige attributter i R, så X eller hvis X-> Y. I denne relasjonen er X er plassholder til DG og Y for AC. Vi ser at Y verken er en delmengde i X, eller utgjør samtlige attributter i relasjonen når det kryssmultipliseres med Y. Vi må dermed ha at DF->AC for at DG->->AC.

CDF -> B
CDF-> G
CDFG -> A
ACDFG -> B

Vi antar at CDF -> B er sann. Fra Q har vi at d -> g, ergo har vi at cdf -> g. Om vi har at cdf -> g har vi fra Q at cdfg -> a. Om vi har at cdfg-> g har vi fra Q at acdfg -> b. Vi kan fra dette se at tillukningen fra cdf gir acdfeg, hvilket gir oss den ikke-trivielle fd'en cdf -> B. Vi kan dermed konkludere med at CDF -> B er sann om vi utvider Q med MVDen DG ->->AC.