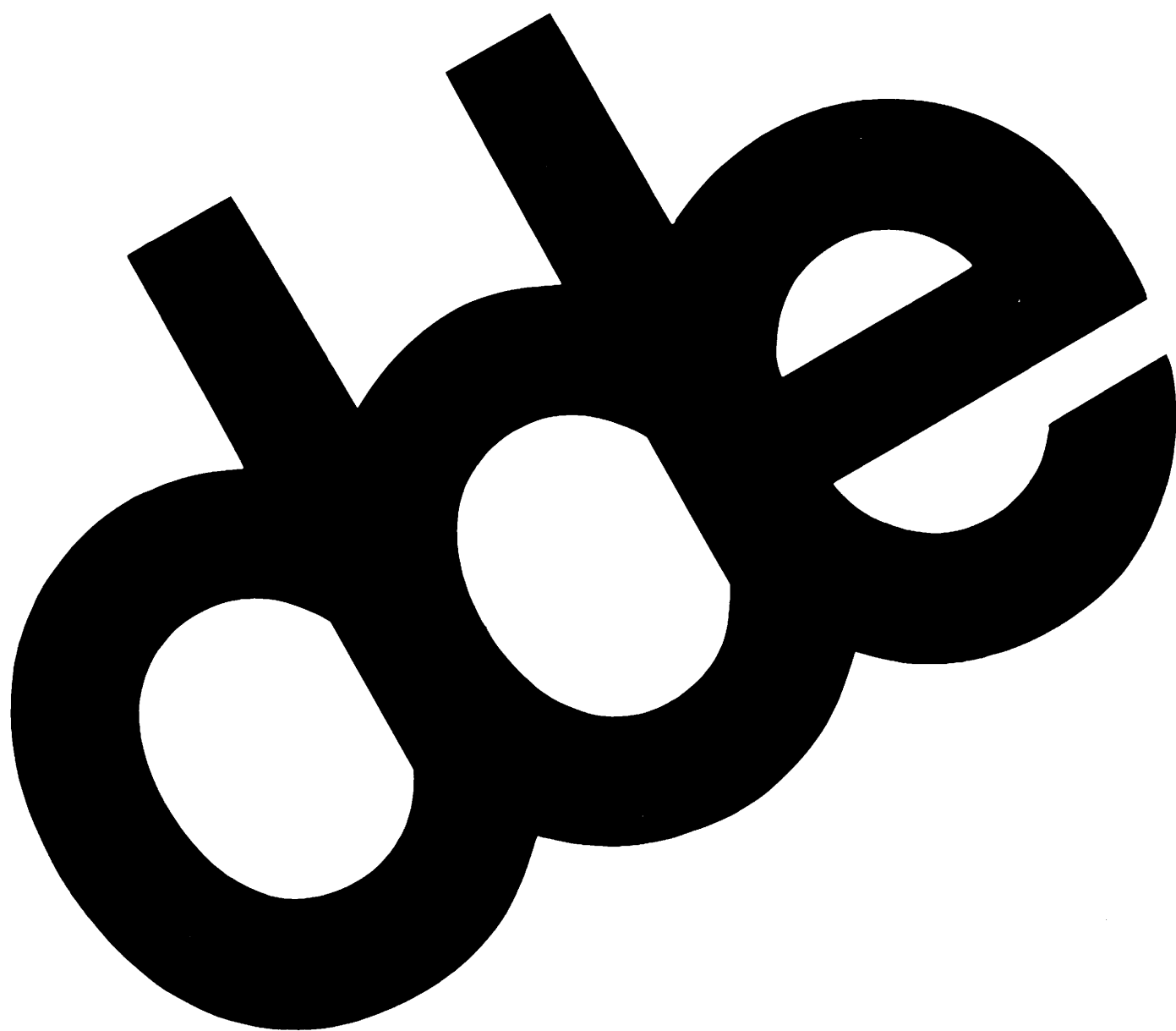


# Access Applikationsudvikling



(styre)formulare :

Lister skal starte med at være styrelse (derved trækkes ikke på databasen)

Boj: "Inside ODBC" Microsoft Press

---

EGU's sikre applikation har admin bruger :  
"jltadmin" pwd: "1234"

# Deltagerliste :

udskrevet: 24-JAN-97

---



## Access Applikationsudvikling

Den 27. til 29. januar 1997

John Flarup, afd. XN  
Scandinavian Airlines Data Denmark A/S

Marianne Elmlund  
Teledanmark Kursus Center Fyn (6916)

Thomas Buis-Larsen  
Teledanmark Kursus Center Fyn (6916)

Dorte Holm  
Geodatacentret

Gitte Engelsholm (ge)  
DDE INTERN ORDRE 131 Kursusafd.



<b>1. Makroer .....</b>	<b>1-1</b>
Reference til kontrolelement i en formular .....	1-1
Reference til et kontrolelements egenskaber.....	1-2
Reference til kontrolelement i en underformular .....	1-3
Ubundne felter til styring af parametre .....	1-4
Øvelse 1.1.....	1-5
Øvelse 1.2.....	1-6
<b>2. Avanceret formularhåndtering.....</b>	<b>2-1</b>
Egenskaben Rammetype .....	2-2
Placering af popupformular .....	2-3
Opbygning af avanceret søgeformular .....	2-4
Funktionen IIF .....	2-5
Opdatering af listen.....	2-5
Antal fundne rækker.....	2-5
Søgninger på mere end ét felt .....	2-6
Flere funktioner.....	2-6
Øvelse 2.1.....	2-9
Øvelse 2.2.....	2-10
Øvelse 2.3.....	2-11
<b>3. Menuredigering.....</b>	<b>3-1</b>
Menugenerator .....	3-1
Opret menulinie.....	3-2
Handlinger.....	3-3
Opret undermenu.....	3-4
Menuliniemakroer.....	3-5
Global menulinie.....	3-5
Øvelse 3.1.....	3-7
Øvelse 3.2.....	3-8
<b>4. Værktøjslinier.....</b>	<b>4-1</b>
Opret værktøjslinie.....	4-1
Indsæt værktøjslinie .....	4-2
Placering af værktøjslinie.....	4-3
Brugerdefinerede knapper .....	4-4
Øvelse 4.1.....	4-5

Arbejdsgrupper.....	5-1
Opret arbejdsgruppe .....	5-1
Opret brugerkonto .....	5-3
Tilføj brugerkonto til gruppekonto .....	5-4
Aktivering af proceduren logon .....	5-4
Skift adgangskode .....	5-4
Ejerskab af databasen .....	5-5
Nye grupper.....	5-5
Tilladelser.....	5-6
Import af database .....	5-7
Oprettelse af egen INI-fil .....	5-7
Andre kommandolinje-parametre .....	5-9
Andre former for sikkerhed.....	5-9
Makroen Autotaster.....	5-10
Tegnkoder .....	5-10
Checkliste.....	5-11
Gruppe- og brugertabeller .....	5-12
Forespørgsel, der viser brugere og grupper.....	5-12
Øvelse 5.1.....	5-13
Øvelse 5.2.....	5-14
<b>6. Access Basic .....</b>	<b>6-1</b>
Hvad kan Access Basic ? .....	6-1
Hvad indbærer arbejdet med Access Basic .....	6-1
Funktioner og subprocedurer .....	6-2
Første subprocedure .....	6-2
Subprocedure aktiveret fra timer .....	6-4
Automatisk udlogning v.h.a. subprocedure .....	6-5
Access Basic udviklingsmiljøet .....	6-6
Objekters metoder og egenskaber .....	6-8
Globale moduler.....	6-9
Oprettelse af en funktion i Access Basic.....	6-10
Opbyg funktion .....	6-11
Eksempler på funktioner .....	6-11
Brug af funktioner i forespørgsler.....	6-12
Datatyper i Access Basic.....	6-13
Erklæringer af variable.....	6-14
Access Basic og formularer .....	6-15
Hvad med fejl ? .....	6-16

Debug og Trace .....	6-19
API-kard .....	6-20
Udveksling af data med DDE .....	6-22
Udveksling af data med OLE .....	6-24
Øvelse 6.1 .....	6-25
Øvelse 6.2 .....	6-26
Øvelse 6.3 .....	6-27
Øvelse 6.4 .....	6-28
Øvelse 6.5 .....	6-29
Øvelse 6.6 .....	6-30
<b>7. Biblioteksdatabase .....</b>	<b>7-1</b>
Værktøjslinier i biblioteksdatabase .....	7-1
Vis/skjul-funktioner .....	7-1
Indlæsning af biblioteksdatabase .....	7-2
Kald af bibliotekets værktøjslinier .....	7-2
Skjul bibliotekets værktøjslinier .....	7-3
Øvelse 7.1 .....	7-5
<b>8. Dokumentation .....</b>	<b>8-1</b>
Udskriv definition .....	8-1
Databasestrukturvisning .....	8-2
Øvelse 8.1 .....	8-3
<b>9. Optimering og tuning .....</b>	<b>9-1</b>
Hvad er optimering ? .....	9-1
Tuning af MSACC20.INI .....	9-1
Andre metoder til optimering .....	9-2
ISAM-JET parametre .....	9-2
Øvelse 9.1 .....	9-5

— 3 dage

F:\P72\Windows\

Appl. MDB





# 1. Makroer

Makroer kan blandt meget andet bruges til at ændre kontrolelementernes egenskaber i en formular. Dette kræver, at der kan refereres til kontrolelementer og deres egenskaber.

## Reference til kontrolelement i en formular

En reference til et kontrolelement i en formular kan angives med følgende syntaks:

**Formularer![[formularnavn]]![[kontrolelementnavn]]**

## Eksempel

En makro skal åbne formularen Ordre synkroniseret på kontrolelementet Kundenr.

The screenshot shows the Access Macro Editor for a macro named "Makro: Eksempel1". The macro table contains one action:

Handling	Kommentar
ÅbnFormular	Åbner formularen Ordre

Below the macro table are the "Handlingsargumenter" (Action Arguments) for the "ÅbnFormular" action:

Formularnavn	Ordre
Visning	Formular
Filternavn	
Where-betingelse	[Kundenr]=[Formularer]![Ordre]![[K...]]
Datatilstand	Rediger
Vinduestilstand	Normal

On the right side of the dialog, there is a text box with the following text: "Indtast en SQL WHERE-delsætning eller udtryk, som vælger poster til formularen fra dens underliggende tabel eller forespørgsel. Klik på knappen Generer for at bruge..."

At the bottom of the dialog is a "Zoom" dialog box with the following text: "[Kundenr]=[Formularer]![Ordre]![Kundenr]". It has "OK" and "Annuller" buttons.



## Reference til et kontrolelements egenskaber

Egenskab er eksempelvis Titeltekt, Viser, Kursiv, VedKlik, Bredde og Skrifttype. En reference til et kontrolelements egenskaber kan angives med følgende syntaks:

**Formularer!***[formularnavn]*!*[kontrolelementnavn]*.egenskab

### Eksempel

En makro skal ændre teksten på en trykknop. Teksten skal skifte mellem 'Lukket' og 'Åben' for hvert tryk på knappen. Knappens navn er 'knap100'.

Betingelse	Handling	Kommentar
[knap100].[Titeltekt]='Åben'	AngivVærdi	Ændrer knaptekst til 'Lukket'
...	StopMakro	
[knap100].[Titeltekt]='Lukket'	AngivVærdi	Ændrer knaptekst til 'Åben'

Handlingsargumenter		
Element	[Knap100].[Titeltekt]	Indtast et
Udtryk	"Lukket"	
Element	[Knap100].[Titeltekt]	Indtast et
Udtryk	"Åben"	



## Reference til kontrolelement i en underformular

En reference til et kontrolelement i en underformular kan angives med følgende syntaks:

**Formularer!**[*hovedformularnavn*!] [*underformularkontrolelementnavn*].Formular!*[kontrolelementnavn]*

Bemærk, at en underformular betragtes som et kontrolelement af hovedformen.

### Eksempel

En makro skal som en del af sin funktion flytte markøren til kontrolelementet 'Produktnr' i underformularen 'sub'.

Makro: Makro1	
Handling	Kommentar
▶ Gå til kontrolelement	Flytter markøren til underformular

Handlingsargumenter	
Kontrolelementnavn	[Formularer!][Ordrer!][Sub].[Formu

[Formularer!][Ordrer!][Sub].[Formular!][Produktnr]	<input type="button" value="OK"/> <input type="button" value="Annuller"/>
--	--



## Ubundne felter til styring af parametre

Ubundne kontrolelementer indeholder værdier, der ikke gemmes i databasen. Disse værdier kan bruges som parametre i forbindelse med makroer, forespørgsler mm. De ubundne kontrolelementer kan anbringes i selve formularen eller i popup-formularer (omtales efterfølgende).

### Eksempel

Et ubundet kontrolfelt 'Knr' skal styre udskrivning af en rapport 'Ordrerapport'. Hvis feltet 'Knr' er tomt, skal rapporten udskrives for alle kunder. Hvis feltet 'Knr' indeholder et kundernr, skal der kun udskrives en rapport for denne kunde.

Makro: månedsoversigt	
Handling	Kommentar
ÅbnRapport	Udskriver Ordrerapport for kunden angivet i 'Knr'

Handlingsargumenter

Rapportnavn	Ordrerapport	Indtast en SQL WHERE-delsætning eller udtryk, som vælger poster til rapporten fra dens underliggende tabel eller
Visning	Vis udskrift	
Filternavn		
Where-betingelse	[Kundenr] Like '' & [Formularer]	

[Kundenr] Like '' & [Formularer]![Ordrer]![Knr]

OK  
Annuller





## Øvelse 1.2

*Fomål: at oprette en trykknappmakro der udskriver en rapport på baggrund af værdien i et ubundet kontrolfelt*

1. Åbn formularen 'hform' i designvisning
2. Opret en trykknop med teksten 'Udskriv månedsrapport'
3. Opret et ubundet kontrolfelt 'År' i formularen
4. Opret en makro tilknyttet trykknappen. Makroen skal udskrive månedsrapporten for det årstal, der er angivet i feltet 'År'. Hvis intet år er angivet, udskrives rapporten for alle årene.

Ordrer		Det er i dag fredag d.14.oktober 1994					
<b>Ordrenr</b>	10015	<b>Kunde-id</b>	PICCO	<b>Medarb.</b>	6	<b>Modtager</b>	Piccolo und mehr
<b>Pris</b>		<b>Spedition</b>		<b>Datoer</b>			
<b>Fragtkost.:</b>	221,00 kr	<input checked="" type="checkbox"/>	<b>Ekspres</b>	<b>Ordredato:</b>	05.06.1991		
<b>Moms af fragt:</b>	55,25 kr	<input type="checkbox"/>	<b>Eurofragt</b>	<b>Forsendelsesdato:</b>	20.06.1991		
<b>Ordretotal:</b>	12.455,63 kr	<input type="checkbox"/>	<b>Turbo</b>	<b>Behandlingstid:</b>	15 dage		
<b>Nedslag/tid:</b>	500,00 kr						
<b>At betale:</b>	12.231,88 kr	<b>Angiv årstal:</b>	93	<b>Udskriv månedsoversigt</b>			
Produktnr	Stykkpris	Styk	Rabat	I alt	Moms	Total	
▶ 11	102,90 kr	60	0%	6.174,00 kr	1.543,50 kr	7.717,50 kr	+
24	21,00 kr	25	0%	525,00 kr	131,25 kr	656,25 kr	
41	46,90 kr	20	0%	938,00 kr	234,50 kr	1.172,50 kr	
73	73,50 kr	15	0%	1.102,50 kr	275,63 kr	1.378,13 kr	
74	49,00 kr	25	0%	1.225,00 kr	306,25 kr	1.531,25 kr	↓
<b>Ordrespecifikationer</b>							<b>Skjul</b>



## 2. Avanceret formularhåndtering

Kald af en formular fra den aktuelle formular kan dels bruges til at videregive oplysninger til brugeren og dels til at gå i dialog med brugeren. Til disse formål anvendes typisk formularer af typen popup. Popupformularer forbliver oven på andre åbne formularer, selv når de ikke er aktive. Et eksempel på en popupformular er egenskabsarket i en formulars designvisning.

Der er to typer af popupformularer:

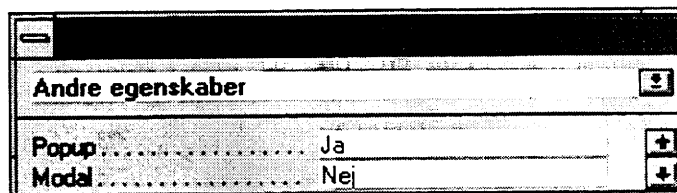
### 1. Ikke modal

Betyder, at man kan få adgang til andre objekter og menukommandoer i Access, mens formularen er åben.

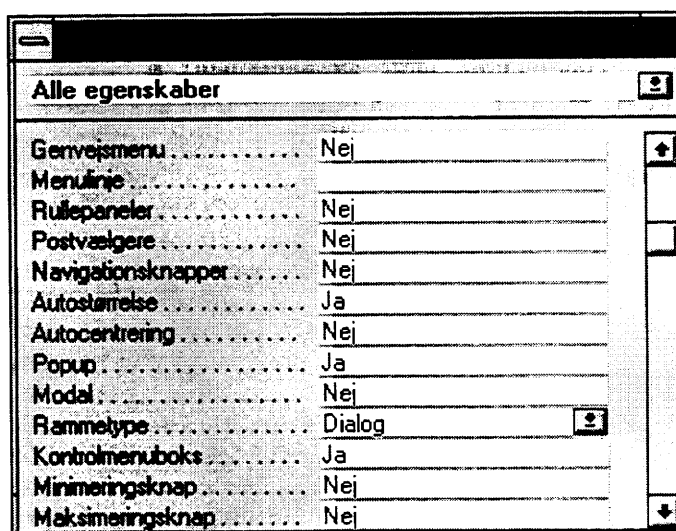
### 2. Modal

Betyder, at man er forhindret i at få adgang til andre objekter og menukommandoer i Access, før formularen er lukket eller skjult. En modal popupformular kaldes også en dialogboks.

En formular ændres til en popupformular ved at tilrette formularegenskaben Popup. Egenskaben Modal er ligeledes en formularegenskab.



Andre formularegenskaber, der er relevante i forhold til popupformularer.





Tip:

Angiv først egenskaben Kontrolmenuboks til 'Nej', når du har sikret dig, at popupformularen ikke skal redigeres mere, eller at den kan lukkes ned på anden vis.

## Egenskaben Rammetype

Indstillingerne i forbindelse med formularer er følgende:

- Ingen                      Formularen har ingen ramme eller relaterede rammeelementer
- Tynd                        Formularen har en tynd ramme og kan indeholde et hvilket som helst af rammeelementerne. Formularens størrelse kan ikke tilpasses
- Kan tilpasses            Standard. Formularen har standardrammen for Accessformularer og kan indeholde alle rammeelementer. Formularens størrelse kan tilpasses.
- Dialog                     Formularen har en tyk dobbeltramme og kan kun indeholde en titellinie og en kontrolmenu. Formularen kan ikke maksimeres, minimeres eller tilpasses i størrelse.

Bemærk, at hvis egenskaben Rammetype i forbindelse med formularer ændres, får det først virkning, når formularen har været lukket og åbnes igen i formularvisning.





## Placering af popupformular

Fra en makro, der åbner en popupformular, kan det bestemmes, hvor i skærbilledet popupformularen skal placeres. Det gøres ved hjælp af handlingen FlytEllerTilpasStørrelse.

### Eksempel

Handling	Kommentar
ÅbnFormular	Åbner formularen XX
FlytEllerTilpasStørrelse	Flytter XX til den ønskede placering

Handlingsargumenter	
Højre	6,046 cm
Ned	1,25 cm
Bredde	
Højde	

Beskrivelse af handlingsargumenter:

- Højre Den nye vandrette position af vinduets øverste venstre hjørne. Måles i cm fra venstre kant af klientvinduet (hele skærmen).
- Ned Den nye lodrette position af vinduets øverste venstre hjørne. Måles i cm fra den øverste kant af klientvinduet
- Bredde Vinduets nye bredde.
- Højde Vinduets nye højde.

## Opbygning af avanceret søgeformular

Access' søgemuligheder i formularer er relativt primitive. Man kan enten anvende menupunktet søgning, eller anvende et filter. Ofte er begge muligheder utilstrækkelige, og derfor vil man ret hurtigt begynde at overveje, hvordan et mere avanceret søgeskærbillede kan opbygges. I det følgende gennemgås en formularstruktur, der kan anvendes i en lang række sammenhænge.

### Eksempel

db

a\*

Sang	Kunstner	Titel	År	Type
About a girl	Nirvana	Bleach	1989	Album
About a girl	Nirvana	Unplugged in New York	1994	Album
All apologies	Nirvana	In utero	1993	Album
All I wanna do	Crow, Sheryl	Tuesday night music club	1993	Album
All I've got to do	Beatles, The	With The Beatles	1963	Album
All my loving	Beatles, The	With The Beatles	1963	Album

6 fund

Vis Album    Vis sang    Vis kunstner

**Søg**   **Nulstil**   **Link**

Ovenstående figur viser en søgeformular. Ideen er, at man indtaster et kriterium i det øverste felt, og derefter klikker på knappen SØG. Herefter vises information om de fundne poster, i dette tilfælde cd'ere, og i venstre hjørne vises antallet af fundne rækker. I det følgende gennemgås opbygningen af formularen.

Ubundet

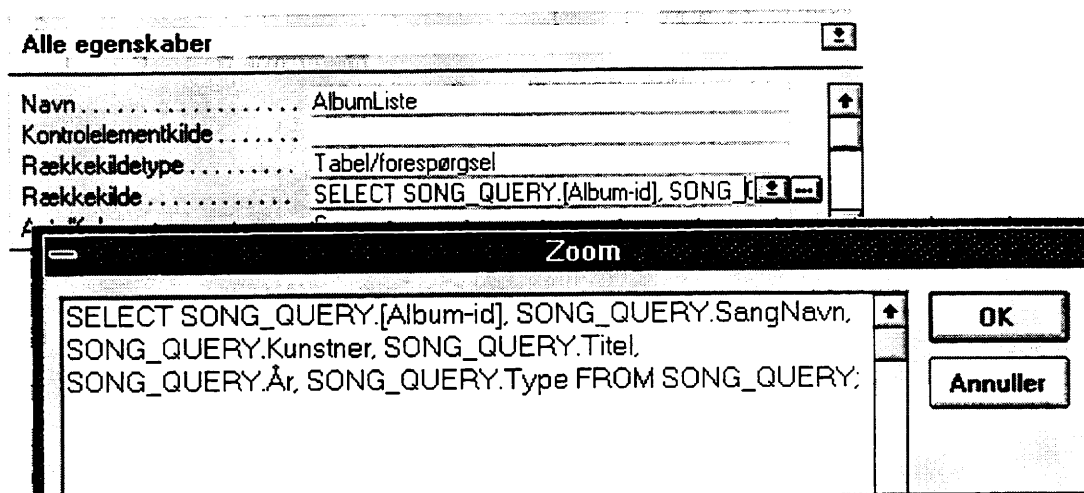
Sang	Kunstner	Titel	År	Type
Ubundet				

Ubundet

Vis Album    Vis sang    Vis kunstner



Selve formularen er ikke baseret på et datasæt. Derimod er der øverst på formularen et ubundet felt, som anvendes til indtastning af kriterier. Dette felt er i det konkrete eksempel en kombinationsboks, som henter data fra den datakilde, der søges på - i dette tilfælde en liste over sange. Dette gøres udelukkende for at lette søgningen. Midt på formularen er der en liste, og det er denne listes datasæt, der skal ændres når et nyt kriterium indtastes. Hvis intet kriterie indtastes skal alle rækker søges frem. I nederste venstre hjørne findes et felt, der skal fortælle antallet af fund. Hemmeligheden består i at lade listens datakilde bestå af en forespørgsel, som anvender det ubundne felt på formularen som datakilde. Listens dataegenskaber ser ud som følger:



Som det ses, er listen baseret på forespørgslen SONG\_QUERY. I denne forespørgsel er det feltet SangNavn, som der skal søges på. Derfor indsættes der i forespørgslen følgende kriterium for netop dette felt:

Like

```
Iif([Formularer]![FIND_SONGS]![SANG]="";"*";[Formularer]![FIND_SONGS]![SANG] &
"*)
```

## Funktionen IIF

Returnerer et af to argumenter, afhængig af udtrykkets udregning. Syntaksen er Iif(udtr, sanddel, falskdel).

## Opdatering af listen

Listen opdateres enten med F9 eller med en makro, der anvender handlingen Genforespørg.

## Antal fundne rækker

Antallet af fundne rækker kan beregnes på mange måder, men det nemmeste er at anvende en tekstboks, der som datakilde har listens egenskab LISTCOUNT, f.eks. =[MinListe].Listcount.



## Søgninger på mere end ét felt

Hvis man ønsker at lave formularer, der er baseret på mere end ét felt kan dette gøre ganske enkelt. Man opretter simpelthen flere ubundne felter på formularen og indsætter passende IIF's i listens tilhørende forespørgsel.

## Flere funktioner

Man kan ganske nemt udbygge skærbilledet således at den aktuelle post findes frem ved dobbeltklik på listen. Dette gøres ved at tilknytte en stump Access Basic kode eller en makro til hændelse på VedDobbeltKlik på listen.





Ordrer							
10021		ERNSH					
<b>Pris</b>							
Frøtomkost:	751,70 kr						
Moms af frøgt:	187,93 kr						
Ordretotal:	14.660,62 kr						
Nedslag/tid:	500,00 kr						
<b>At betale:</b> 15.100,25 kr							
<b>Produkter:</b>				25			
<b>Leverandørar:</b>				11			
<b>Kategoriar:</b>				3			
<b>Produktneva:</b>				NuNuCa Nuff-Nougat-Creme			
<b>Dansk nava:</b>				NuNuCa chocoladecreme			
<b>Antal pr enhed:</b>				20 - 450 g glas			
<b>Stykpris:</b>				kr 98,00			
<b>Enheder på lager:</b>				76			
<b>Enheder i ordre:</b>				0			
<b>Genbestilling ved:</b>				30			
<b>Udgøst:</b>				<input type="checkbox"/>			
Produktnr	Stykpris	Styk	Råbat	I alt	Moms	Total	
1	84,00 kr	60	20%	4.032,00 kr	1.008,00 kr	5.040,00 kr	
10	151,90 kr	27	0%	4.101,30 kr	1.025,33 kr	5.126,63 kr	
▶ 25	68,60 kr	5	20%	274,40 kr	68,60 kr	343,00 kr	
33	7,00 kr	50	20%	280,00 kr	70,00 kr	350,00 kr	
36	91,00 kr	36	20%	2.620,80 kr	655,20 kr	3.276,00 kr	+
<b>Skjul produkt oplysninger</b>							
<b>Ordrespecifikationer</b>							<b>Skjul</b>

Placeringsforslag til popupformularen 'Prodopl'.

## Øvelse 2.1

Formål: at oprette en popupformular

1. Opret en formular ud fra tabellen 'Produkter'. Gem den under navnet 'Prodoplys'.
2. Tilret formularens egenskaber, således at følgende betingelser er opfyldt:
  - Formularen er af typen popup *Popup = Ja*
  - Rammestørrelsen må ikke kunne ændres *Rammetype = Tjnd*
  - Formularen skal indeholde en kontrolmenuboks *Kontrolmenu = Ja*
  - Overskriften skal være 'Produktoplysninger' *Titeltekst*
  - Formularen må ikke indeholde navigeringsknapper *Navigation = Nej*
  - Det skal være muligt at flytte fokus til andre vinduer

Formularen 'Prodoplys' kan eksempelvis have følgende layout:

Produktar:	1
Leverandrar:	1
Kategoriar:	1
Produktavnan:	Chai
Dansk navn:	Dharamsala-te
Antal pr enhed:	10 kasser á 20 poser
Styckpris:	kr 126,00
Enheder på lager:	39
Enheder i ordre:	0
Costestilling ved:	10
Udglst:	<input type="checkbox"/>

3. Opret en trykknapp med teksten 'Vis produktoplysninger'.
4. Opret en makro 'Prodopl', der åbner fomularen 'Prodoplys' og ændrer knapteksten til 'Skjul produktoplysninger'. Formularen skal vise oplysninger om det produkt, der er aktuelt i underformularen
5. Opret en makro 'Luk', der skjuler formularen 'Prodoplys' og ændrer knapteksten til 'Vis produktoplysninger'. Tilføj makroen 'Luk' til makroen 'Prodopl'.
6. Tilret makroen 'Prodopl' så formularen 'Prodoplys' placeres hensigtsmæssigt i billedet. Hensigtsmæssigt vil sige, at underformularen ikke dækkes af popupformularen.



## Øvelse 2.2

*Formål: at synkronisere popupformularen med markørbevægelserne i formularen 'hform'*

1. Tilret makroen 'Prodopl', så markøren efter opslag i formularen 'Prodoplys' vender tilbage til produktet i underformularen.  
Tip: det er ikke nødvendigt at angive en Where-betingelse.
2. Opret en makro 'Opdat'. Makroen skal skifte til formularen 'Prodoplys', finde oplysninger om det aktuelle produkt og derefter vende tilbage til underformularen. Makroen skal kun udføres når formularen 'Prodoplys' vises.  
Tip: Formularen 'Prodoplys' vises, når ingen af trykknapperne har teksten 'Vis ...'
3. Tilnyt makroen 'Opdat' til en passende hændelsesegenskab. Makroen skal køres, hver gang der skiftes til en ny post i underformularen.
4. Tilret makroen 'Hformåbn', så trykknappen 'Vis produktoplysninger' er skjult, når formularen åbnes.
5. Tilret makroen 'Vis\_skjul'. Hvis underformularen skjules, skal også 'Prodoplys' skjules.



Ikke tabel-bundet = ubundet



# Øvelse 2.3

Rekkefølge Nej  
Postfølger Nej

Formål: at oprette en søgeformular

- Opret en søgeformular, som kan bruges til at søge på kunder. Brug nedenstående støt-  
tepunkter og inspiration fra materialet:
  - Opret en ubundet formular, som indeholder et søgefelt.
  - Opret en forespørgsel, der v.h.a. af en IIF-funktion anvender formularens søgefelt som kriterie.
  - Opret en liste på formularen, der anvender forespørgslen som rækkekilde.
  - Opret en knap, der genforespørger på listen.
  - Opret et felt, der viser antal fund.
- Eksperimentér med søgeformularen, og undersøg om den fungerer korrekt. Overvej hvordan den kan udbygges.

⇒ listcomt bl. nr 1 større

Kriterie:

Kunde-ID	Firmanavn	Kontaktperson	Telefon	Fax
SUPRD	Suprêmes délices	Pascale Cartrain	(071) 23 67	(071) 23 67 22
SPLIR	Split Rail Beer & Al	Art Braunschweiger	(307) 555-46	(307) 555-6525
SPECD	Spécialités du mon	Dominique Perrier	(1) 47.55.60.	(1) 47.55.60.20
SIMOB	Simons Bistro	Jytte Petersen	31 12 34 56	31 13 35 57
SEVES	Seven Seas Import	Hari Kumar	(71) 555-171	(71) 555-5646
SAVEA	Save-a-lot Markets	Jose Pavarotti	(208) 555-80	
SANTG	Santé Gourmet	Jonas Bergulfsen	07-98 92 35	07-98 92 47

7 kunder fundet

Palet

Rekkefølge: ja

Postfølger: ja

aktiveret Nej

læst Ja

Zoom kun = Shift F2

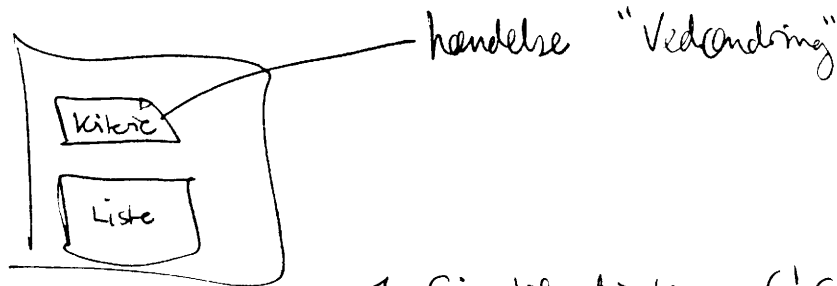
Requery = F9

kriterium

Like form [er]! [Find M. Med]! [Kriterium] & "\*" [Felt 3], [listcomt] & "kunder fundet"

Knap: handling = ~~genforespørg~~

Liste har en egenskab "listcomt" der kan benyttes til antal række-



1. Gå til listen ('Gå til KontrolElement')
2. Børjansøgning
3. Gå til kriterie
4. 'Rediger felt' (=F2) (ellers vil forrige tegn i buffer)
  - ↳ Send Taster

Tastetryk : {F2}

Vent : Nej

(Key buffer er blevet tømt)

## MENURÆDIGERING:

Fornbåret egenhedsliste 'menuligne' ...

tom ⇒ default

Menugenerator (gøres som et stort antal makroer. Vælg derfor navngivning der gør det muligt at skelne fra øvrige (ex '-' preped) )

Handlinger:

KørKode (es. stel Access Base)

KørMacro

UdførMenuKode

4 måder at udføre ting:

Menu pkt

Short-key

Verktøjslinje

Krupper

Handling: AngivVærdi Admin  
 Behøver: (currentUser()) <> "Admin"  
 Element: [Application].[menuligne]  
 Tilstyk: "=main\_menu"



## 3. Menuredigering

Menuer er en bekvem måde at stille et sæt kommandoer til rådighed for brugeren på. Menuer er ikke helt så 'gennemsigtige' som trykknapper, men optager til gengæld ingen plads i formularen eller rapporten. Menulinien vises lige under titellinien.

Menuer kan indeholde indbyggede Access-kommandoer, fx Afslut, kald af makroer samt kald af funktioner/procedurer.

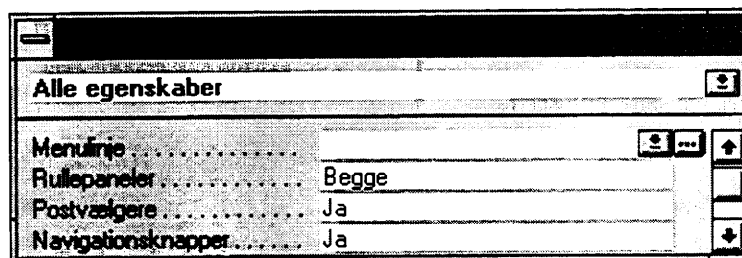
### Menugenerator

Brugerdefinerede menuliner oprettes ved hjælp af den indbyggede menugenerator. Menugeneratorens hjælp består dels i at stille skabeloner til rådighed og dels i at oprette/rette de nødvendige makroer.

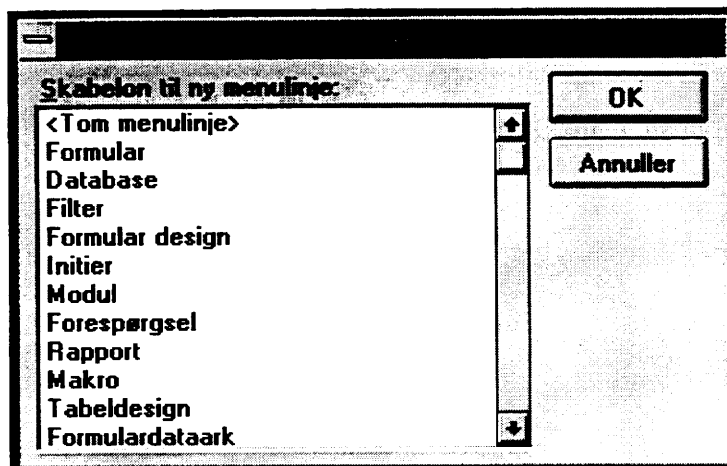
Menugeneratoren kan startes på følgende to måder:

1. Fra et vilkårligt vindue vælges: Filer, Tilføjelsesprogrammer, Menugenerator
2. Fra formular/rapport designvisning vælges: Formularegenskaber, Menulinie

Tryk på knappen Generer:



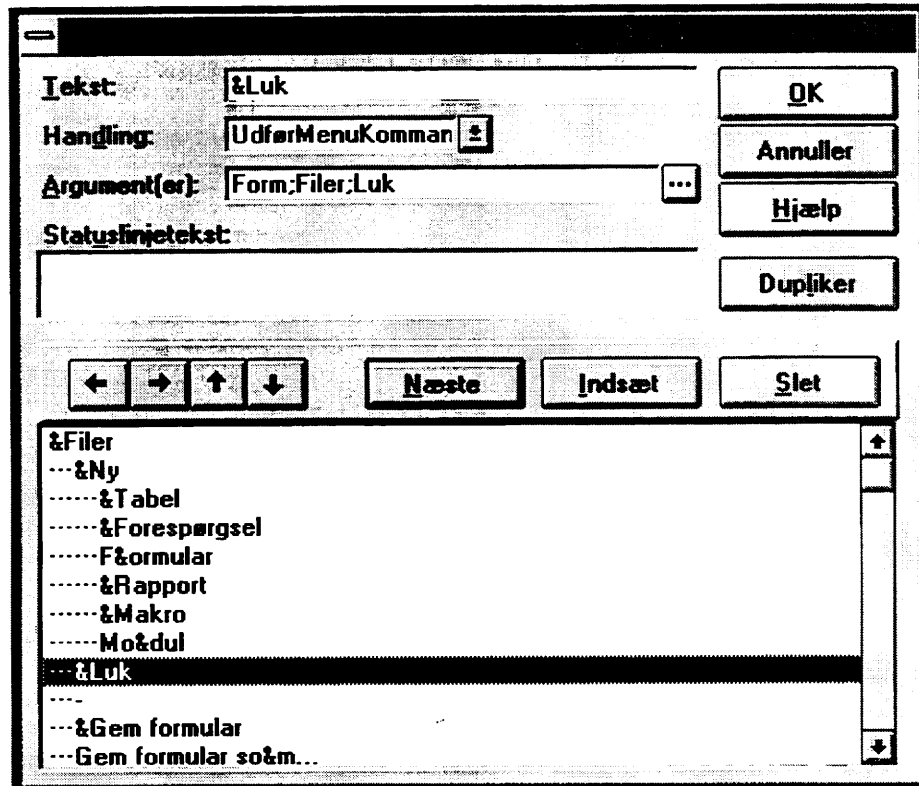
Efter opstart af menugeneratoren og valg af 'Ny menu' vises nedenstående vindue:





## Opret menulinie

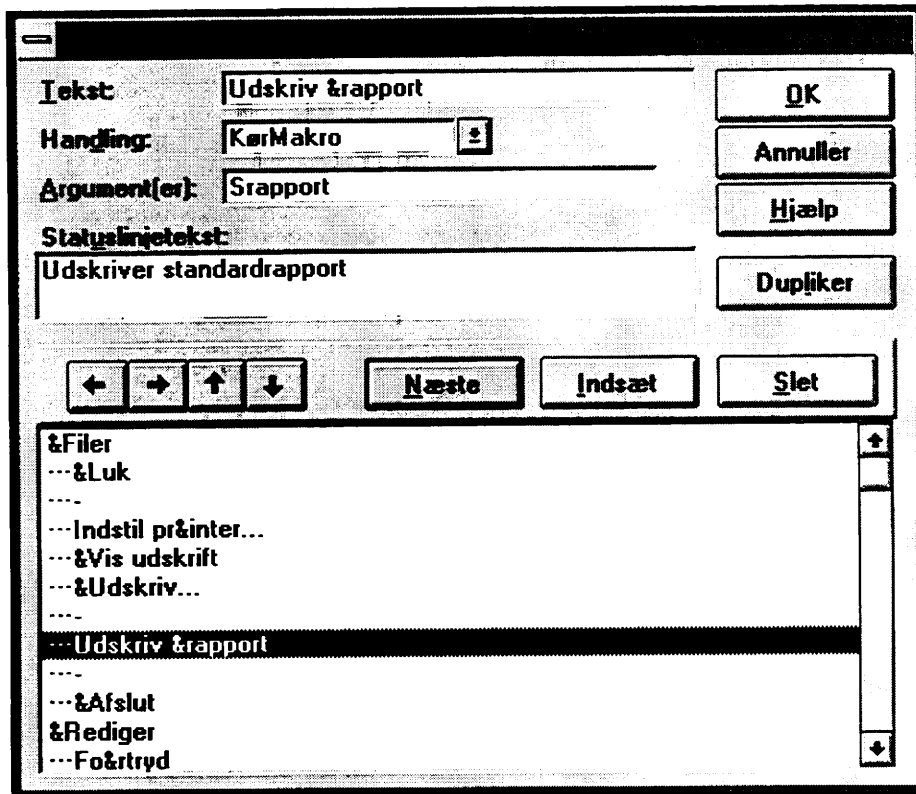
Det vil ofte være en fordel at tage udgangspunkt i en skabelon frem for at starte helt fra bunden. Fra skabelonen slettes de punkter, der ikke ønskes i den færdige menu, og derefter tilføjes egne menupunkter. Vælges fx skabelonen Formular, vil menugeneratoren give dette bud på en menu:



- Slet menupunkt  
Menupunkter slettes ved at markere punktet og klikke på Slet
- Indsæt menupunkt  
Placer markøren hvor menupunktet skal anbringes og klik på Indsæt. Vælg tekst, handling og argument
- Indryk menupunkt  
Marker menupunktet og klik på Pil-højre
  - Et punkt, der står venstrejusteret, vises som menunavn i menulinien
  - Et punkt, der er indrykket et niveau, vises under det pågældende menunavn
  - En enkelt bindestreg vises som separatorlinie
- Flyt menupunkt  
Marker menupunktet der skal flyttes og klik Pil-ned eller Pil-op indtil punktet har den ønskede placering



Menunavnet Filer kan fx tilrettes som vist nedenfor. Bemærk, at makroen Srapport skal være oprettet, inden menuen kan genereres.



Et menupunkt kan tildeles en 'hurtigtast', ved at medtage et &-tegn lige foran det tegn, der skal være hurtigtast. I eksemplet ovenfor er F hurtigtast for Filer og R hurtigtast for Udskriv rapport. En rapport kan udskrives ved at trykke ALT+F og dernæst trykke R.

## Handlinger

Der findes tre typer menuhandling:



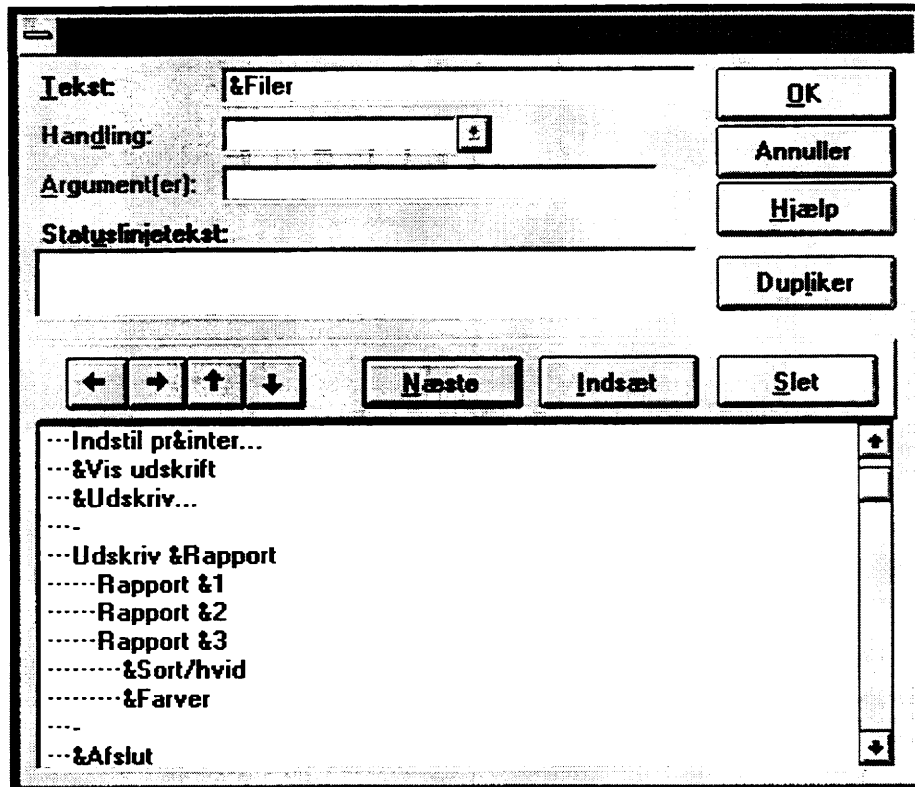
1. UdførMenuKommando  
Udfører en Access-menukommando
2. KørMakro  
Afvikler den angivne makro
3. KørKode  
Afvikler function/procedure skrevet i Access Basic



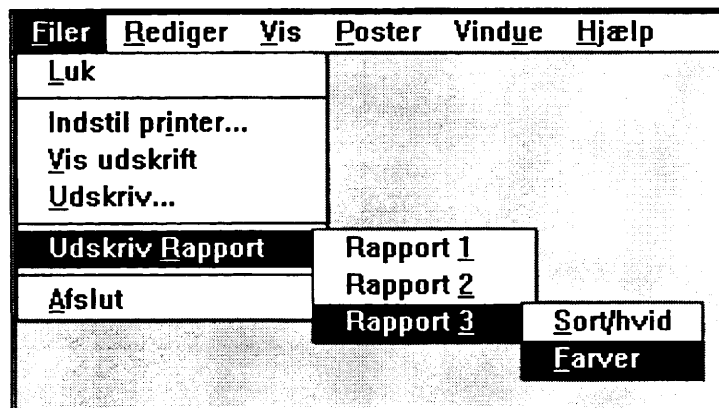
## Opret undermenu

Hver oprettet menu kan indeholde undermenuer i flere niveauer. Undermenuer bruges typisk, når man vil understrege et menupunkts relation til et andet.

Undermenuer oprettes ved at tilføje menupunkter, der ligger et niveau højere end den menu, der kalder dem. I eksemplet fra tidligere kan undermenuer tilføjes således:



Menuerne vil se således ud, når de kaldes:





## Menuliniemakroer

Når der fra menugeneratoren klikkes på Ok, genererer Access en række makroer, der definerer menulinien.

- En menuliniemakro, der indeholder handlingen TilføjMenu for hver menu på menulinien. Denne makro har det samme navn som menulinien.
- En makrogruppe for hver menu og undermenu på menulinien. Hver gruppe indeholder makroer for hvert menupunkt.

Makroerne kan ses i oversigtbilledet 'Makro', og de kan redigeres manuelt ligesom alle andre makroer. Bemærk dog, at hvis sammenhængen i menulien ødelægges, kan menugeneratoren ikke læse menulinien længere.

## Global menulinie

Hvis der er angivet en global menulinie i en applikation, vil den blive brugt i alle vinduer undtagen i formularer og rapporter, der har en menulinie tilknyttet direkte.

Globale menuliner tildes typisk ved opstart af applikationen, dvs. i AutoExec-makroen. Menuliner kan dog tildeles ved en vilkårlig hændelse.

## Eksempel

Applikationen skal bruge menulinien 'Ordreadm' som global menulinie. Indholdet af AutoExec kan da se således ud:

Handling	Kommentar
UdførMenumkommando	Skjul databasevinduet
ÅbnFormular	Åbner formularen Ordreadadministration
▶ AngivVærdi	Angiv global menulinie

Handlingsargumenter	
Element	[Application].[Menulinie]
Udtryk	'Ordreadm'

Den globale menulinie kan nulstilles, så der vendes tilbage til Access' indbyggede menuer. Det gøres ved at angive egenskaben Menulinie for objektet Application til en streng af længden nul ("").

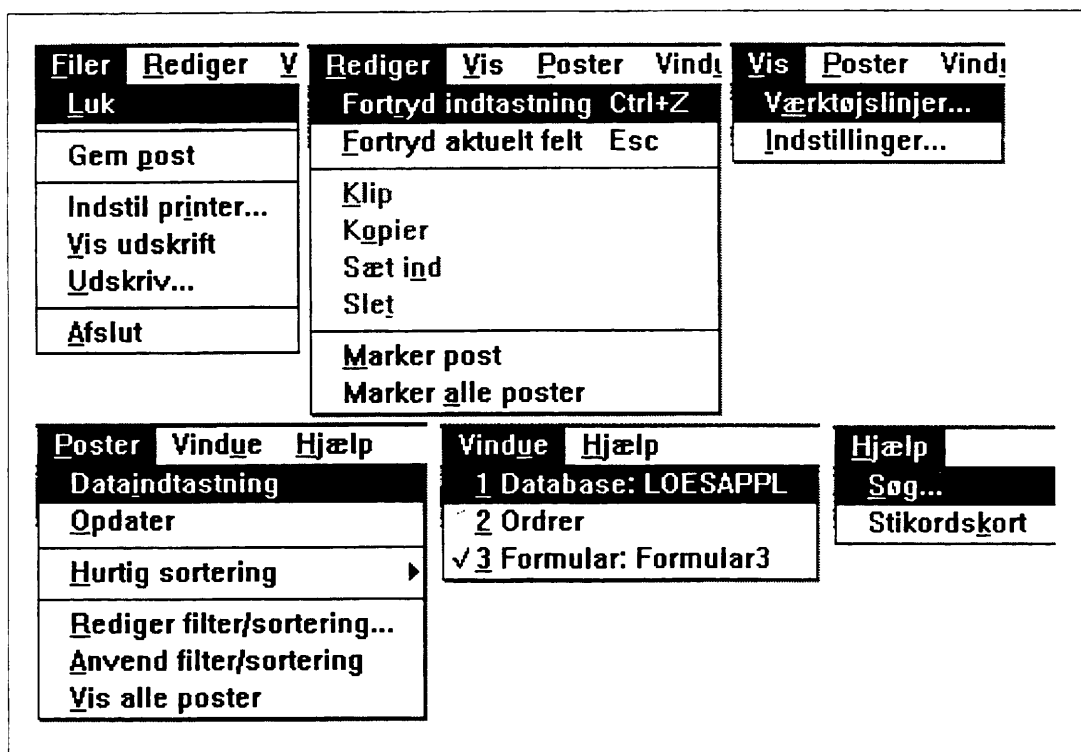




## Opgave 3.1

Formål: At oprette en brugerdefineret menulinie

1. Åbn formularen 'hform' , *Vis Egenskaber, Menulinje, ... , Ok* , *Gen søn Ny menulinje*
2. Opret en ny menulinie 'hform\_menu' ved hjælp af menugeneratoren.  
Menuen skal bestå af følgende menupunkter:



3. Opret en makro 'Autoexec' der skjuler Databasevinduet og åbner formularen 'hform'
4. Luk databasen og start den igen.

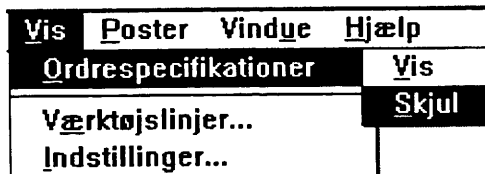
Handling: *Udfør Menükommando*  
 Menuen: *Database*  
 Menuenavn: *Vindue*  
 Kommando: *Skjul*



## Opgave 3.2

Formål: at tilføje nye kommandoer til en eksisterende menu

1. Menupunktet Vis i menulinien 'hform\_menu' skal udvides med følgende:



Tip: De nødvendige makroer er allerede oprettet

2. Kommandoen Afslut under menupunktet Filer skal tilrettes, så afslutningen sker gennem en modal popupformular, hvori afslutkommandoen kan fortrydes.
  - a) Opret en modal popupformular i stil med figuren nedenfor
  - b) Opret de nødvendige makroer
  - c) Tilret kommandoen Afslut i menuen 'hform\_menu'



Alt F4

Initials	Page
Date	Project

Access må ikke benyttes til fortrolig data (registrerer)  
 Kan ikke sikres med adgang hvis man først har en kopi af DB.

255 samtidige brugere

Access er stadig mht. at stille søgning til rådighed for slutbrugere

- Fjernelse af menuer
- ————— værktøjslinier

MS egen eksempel DB 'Wives'

Fordsetter erfaring med 'Formulærer'

Macro

→ Vælg godt navn på knap, da det efterfølgende er svært at <sup>Andre</sup>  
 Knap: Tænd/sluk

Handling:            Angiv Værdi  
                           Element : [reservatoner] [synlig]  
                           Udtryk : Nej

Vis betingelse

Betingelse: [reservatoner].[synlig]= ja  
                           ...  
                           [reservatoner].[synlig]=Nej

Handl  
 Angiv  
 Skop  
 Angiv

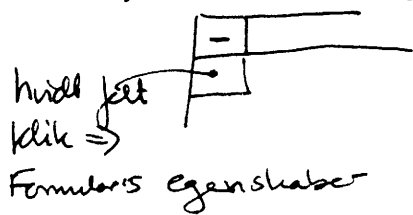
Andre knap's tekst så der skiftes mellem Sluk//Tænd

'Alle egenskaber', 'Titel tekst'

Element : [knap 32].Titeltekst  
 Udtryk : "Tænd"

→ Macro er svært at fejlsøge i !

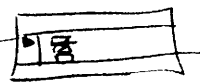
# Vinduetts/formulærens egenskaber



Rullepaneler (scrollbars) Nej



Postvælger (Record selectors) Nej



Navigations knapper (Navig. Buttons) Nej udvalgt række nr!

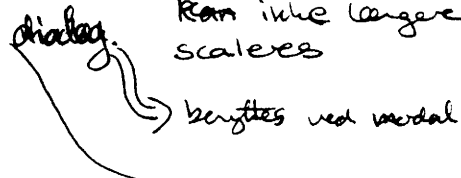
Autostørrelse

Rametype

tynd => vindues størrelse  
Kan ikke længere  
skaleres

Minimer-knap Nej

Maxim-knap Nej



Maksimeret vindue vil altid starte et andet vindue som maksimeret

Modal-vindue (vindue kan ikke lægges i baggrunden og andre må ikke få fokus; End ikke ~~maxim~~ligner)

Popup-vindue (ex værktøjsvindue. Kan ikke lægges i baggrunden men andre må gerne få fokus).



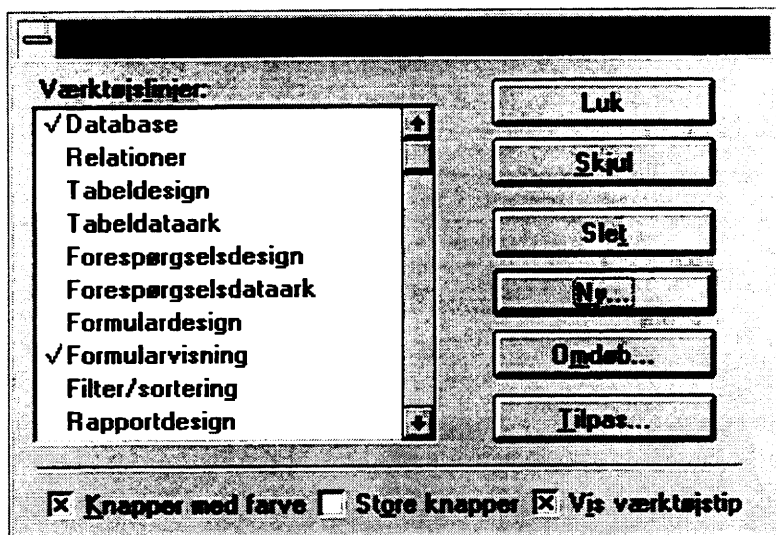
## 4. Værktøjslinier

Brugerdefinerede værktøjslinier kan bruges til, på en hensigtsmæssig og synlig måde, at give brugeren adgang til kommandoer i applikationen. Der kan bruges flere brugerdefinerede værktøjslinier i samme applikation. Handlingen VisVærktøjslinje i makroer bruges til at vise/skjule værktøjslinier.

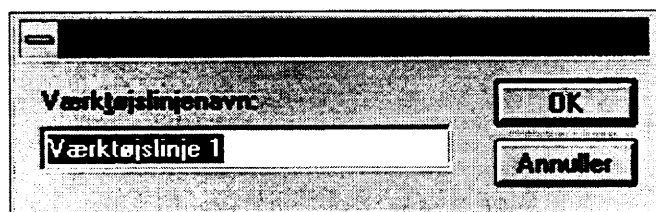
### Opret værktøjslinie

En brugerdefineret værktøjslinie oprettes således:

- Vælg kommandoen Værktøjslinjer... under menupunktet Vis

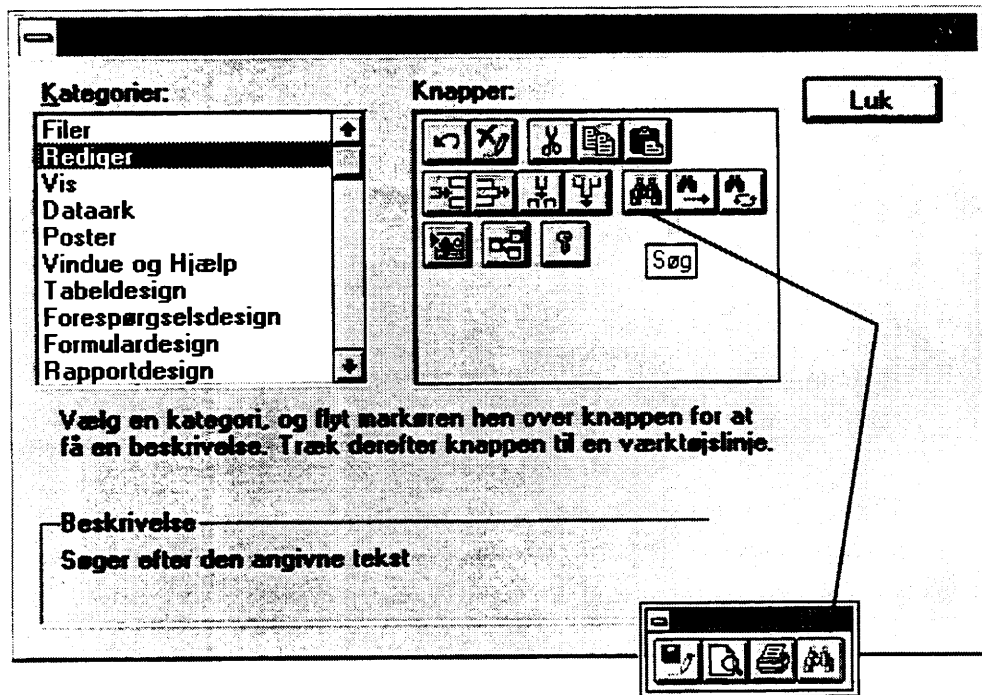


- Vælg Ny



- Giv værktøjslinien et navn
- Vælg Tilpas

'Vis', 'Indstillinger', "Innebyggede værktøjslinjer tilgængelige": Nej



- Træk ikoner, makroer, formularer mm. og placer dem i værktøjslinien

## Indsæt værktøjslinie

Værktøjslinier indsættes ved hjælp af makroer. Skal der kun bruges én værktøjslinie, placeres kaldet i makroen Autoexec. Bruges forskellige værktøjslinier, placeres kaldet i makroer knyttet til formularhændelserne VedAktivering og VedDeaktivering.

Handling	Kommentar
▶ VisVærktøjslinie	Kalder brugerdefineret værktøjslinie
VisVærktøjslinie	Skjuler indbygget værktøjslinie

Handlingsargumenter	
Værktøjslinie	Værktøjslinie 3
Vis	Ja
Værktøjslinie	Formularvisning
Vis	Nej

Bemærk, at der kan vises flere værktøjslinier på en gang. Det er derfor nødvendigt at skjule den eller de værktøjslinier, der ikke ønskes vist. I eksemplet ovenfor skjules den indbyggede værktøjslinie.



## Placering af værktøjslinie

Værktøjslinier kan trækkes med musen og placeres øverst, nederst, til venstre, til højre eller valgfrit i vinduet. Placeres værktøjslinien frit i vinduet, vises den som en popupformular. Den sidste placering af værktøjslinien huskes.

**Ordre**
Det er i dag tirsdag d.18.oktober 1994

10015
PICCO
6
Piccolo und mehr

**Pris**

Fragnetkost.: 221,00 kr

Moms af fragt: 55,25 kr

Ordretotal: 12.455,63 kr

Nedslag/tid: 500,00 kr

**Spedition**

Ekspres

Eurofragt

Turbo

**Datoer**

Ordredato: 05.06.1991

Forsendelsesdato: 20.06.1991

Behandlingstid: 15 dage

At betale: 12.231,88 kr

Angiv årstal:

**Udskriv månedsoversigt**

Produktnr	Stykpri	Styk	Rabat	I alt	Moms	Total
11				0 kr	1.543,50 kr	7.717,50 kr
24				0 kr	131,25 kr	656,25 kr
41				0 kr	234,50 kr	1.172,50 kr
73				0 kr	275,63 kr	1.378,13 kr
74				0 kr	306,25 kr	1.531,25 kr


Vis produkt

Ordrespecifikationer **Skjul**

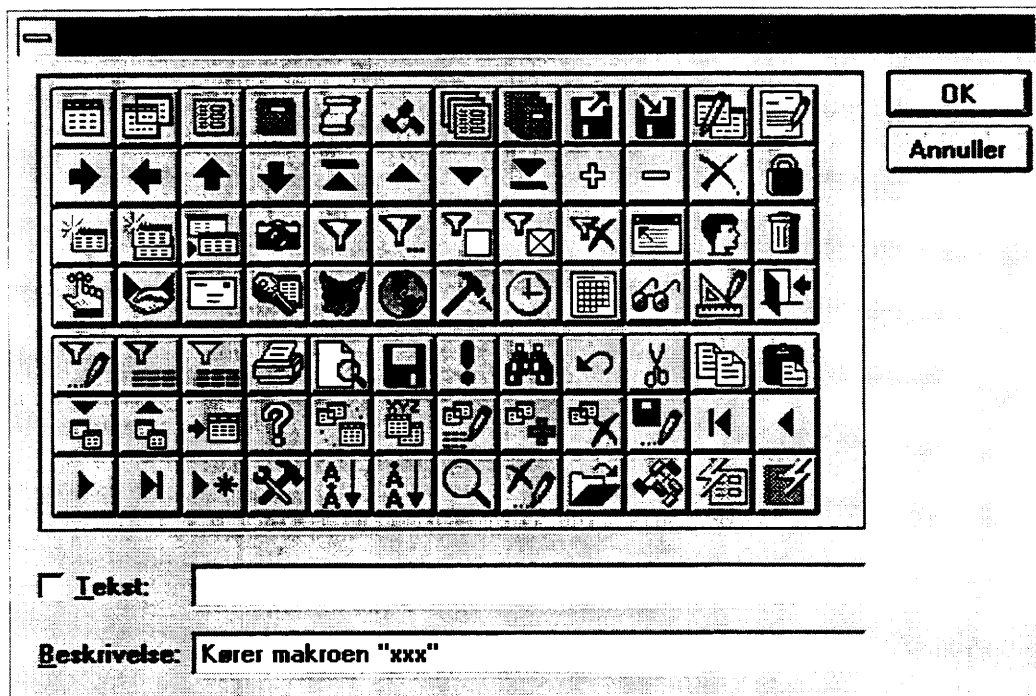


## Brugerdefinerede knapper

Knapper i værktøjslinien kan knyttes til formularer, makroer etc. Knytted knappen eksempelvis til en makro, vil makroen blive kørt, når der klikkes på knappen.

Når en makro trækkes op på værktøjslinien tildeles knappen symbolet: 

Ved at højreklikke på knappen kan billede eller tekst ændres.



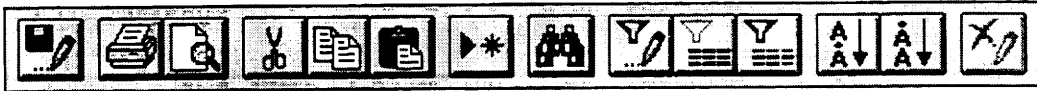
I feltet Tekst skrives en eventuel knaptekst. I feltet Beskrivelse skrives den tekst, der vises, når der peges på knappen.



## Øvelse 4.1

*Formål:* - at oprette en brugerdefineret værktøjslinie  
- at knytte værktøjslinie til formular

1. Opret en værktøjslinie 'hform\_værktøj' med følgende kommandoer:



2. Placér værktøjslinien øverst i vinduet.
3. Opret en makro der viser værktøjslinien 'hform\_værktøj', når formularen 'hform' aktiveres.
4. Opret en makro der skjuler værktøjslinien 'hform\_værktøj' og i stedet viser den indbyggede værktøjslinie, når en anden formular aktiveres.
5. Opret ved hjælp af guiden Autoformular en formular på tabellen 'Medlemmer'. Kontrollér at værktøjslinien skifter, når der skiftes mellem formularerne.
6. Tilføj en knap til værktøjslinien 'hform\_værktøj'. Knappen skal køre makroen 'Vis\_skjul'. Knappen skal være i stil med nedenstående eksempel.



7. Afprøv knappen. Tilret evt. makroen.

MSACC 20. INI

Arbgruppen

- brugere
- pasord
- grupper

SYSTEM.MDA

Databasen

- Tilladelser
- ejerskab

.MDB



## 5. Sikkerhed i flerbrugersystemer

Eftersom databaseprogrammer ofte gemmer og styrer vitale oplysninger, kan en sikring af disse oplysninger være en vigtig del af applikationsudviklingen. Sikkerhedsindstillinger i Access består af flere dele, som tilsammen sikrer en applikation.

- Arbejdsgruppe gemmes i systemdatabasen (standard SYSTEM.MDA)
- Konti gemmes i systemdatabasen (standard SYSTEM.MDA)
- Ejerskab gemmes i databasefilen (<dbnavn>.MDB)
- Tilladelser gemmes i databasefilen (<dbnavn>.MDB)

Hver del skal være på plads, inden sikkerheden er fuldstændig.

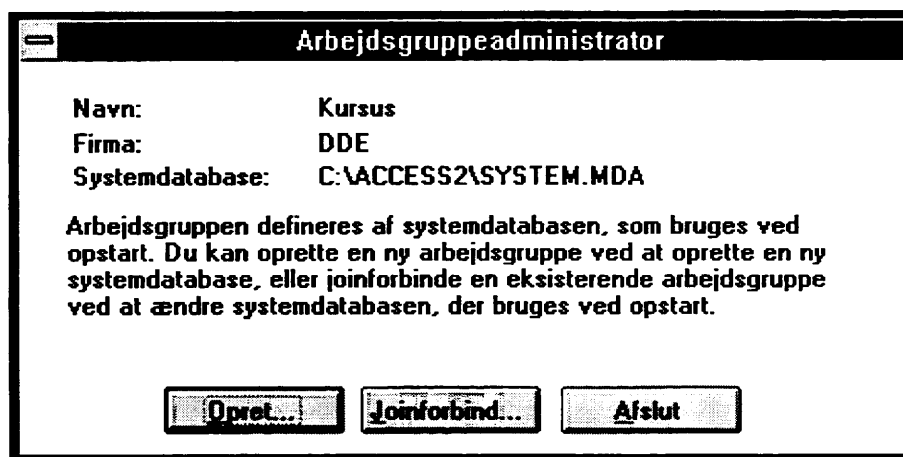
Hvis du vil oprette et sikkert system fra bunden, er den bedste fremgangsmåde at logge dig på med den konto, der skal eje databasen, og dernæst oprette databasen og dens objekter. Ofte ønsker man at sikre en allerede oprettet database. Det vil blive gennemgået nedenfor.

### Arbejdsgrupper

Inden de bruger- og gruppekonti, der skal bruges ved tildelingen af tilladelser kan oprettes, skal der vælges en arbejdsgruppe, hvor disse konti kan gemmes. Der kan vælges en eksisterende arbejdsgruppe, eller der kan oprettes en ny.

### Opret arbejdsgruppe

- Dobbeltklik på ikonet 'MS Access Arbejdsgruppeadministrator'



- Klik på 'Opret...'



**Oplysninger om arbejdsgruppeejer**

Systemdatabasen identificeres af det navn, den organisation og det arbejdsgruppe-ID, du angiver nedenfor. Der skelnes mellem store og små bogstaver.

Brug navnet og organisationen nedenfor, eller angiv noget andet. Hvis du vil sikre, at arbejdsgruppen er unik, skal du indtaste et unikt arbejdsgruppe-ID på højst 20 tal eller bogstaver.

**N**avn:

**O**rganisation:

**A**rbejdsgruppe-ID:

Gemmes  
må ej glemmes!  
Case sensitive

### VIGTIGT!

Hvis du får brug for at genoprette systemdatabasen, skal Navn, Organisation og Arbejdsgruppe-ID angives helt præcis - inkl. store/små bogstaver. Notér dine oplysninger og gem dem et sikkert sted

- Udfyld dialogboksen og klik OK

**Arbejdsgruppesystemdatabase**

Brug stien og navnet nedenfor, eller ret stien og navnet til den nye system- database.

**D**atabase:

- Ret filnavnet, så originalen ikke overskrives, og klik OK
- Verificér oplysningene ved at klikke OK
- Klik OK til arbejdsgruppen
- Klik Afbryd for at afslutte Arbejdsgruppeadministrator



## Opret brugerkonto

- Start Access med den arbejdsgruppe, hvor kontoen skal benyttes
- Åbn en database
- Vælg Brugere i menuen Sikkerhed fra databasevinduet

- Vælg Ny

- Udfyld dialogboksen og klik OK  
Navn/id skal være Logon/Adgangskode på den nye administrator/ejer af databasen

Man kan ikke slette tidligere oprettede <sup>default</sup> brugere/grupper  
'Administratører', 'Brugere', 'Gæster'

Alle brugere er automatisk medlem af gruppen 'brugere' kan ikke fjernes  
Den bruger der oprettes DB (v. input) bliver 'ejer', kan ikke ændres!



## Tilføj brugerkonto til gruppekonto

Den 'nye administrator' skal nu tilføjes til gruppen Administrator.

- Vælg Brugere i menuen Sikkerhed fra databasevinduet

- Vælg den ønskede bruger
- Tilføj grupper til eller fjern grupper fra brugeren og klik Luk

## Aktivering af proceduren logon

Indtil logonproceduren aktiveres, vil alle brugere automatisk blive logget på under brugerkontoen Administrator. Standardadgangskoden til denne konto er en streng af længden nul. For at aktivere logonproceduren skal adgangskoden for Administrator skiftes.

## Skift adgangskode

- Vælg Skift adgangskode i menuen Sikkerhed

Feltet 'Tidligere adgangskode' efterlades tomt.

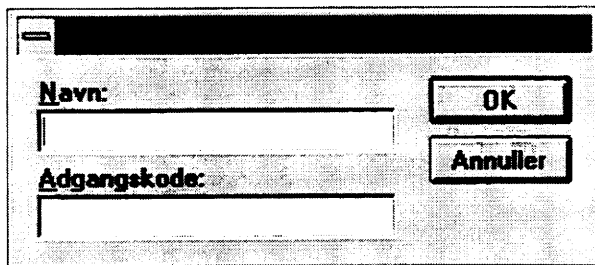


- Angiv ny kode, bekræft koden og klik OK  
Næste gang Access startes op vil dialogboksen Logon blive vist

## Ejerskab af databasen

Ejerskab af en database kan kun ændres ved at oprette en ny database under den brugerkonto, der skal eje databasen. Det er vigtigt, hvem der ejer databasen og dens objekter, fordi ejeren altid har adgang til sine egne objekter. Hvis ejeren fx er Administrator, er databasen i princippet åben for enhver bruger, der kan starte Access.

- Genstart Access



- Angiv Navn og Adgangskode for den 'ny administrator' og klik OK
- Opret en ny database

## Nye grupper

Da grupperne Administratorer og Brugere ikke kan slettes, og i øvrigt findes som standard i alle arbejdsgrupper, er det nødvendigt at oprette en ny adm-gruppe og en ny brugergruppe. Dermed kan alle uønskede rettigheder fjernes for standardgrupperne.

- Vælg Grupper fra menuen Sikkerhed fra databasevinduet



- Vælg Ny  
Opret en adm-gruppe og en bruger-gruppe. Tildel adm-gruppen til den 'ny administrator' og bruger-gruppen til de brugere af systemet



## Tilladelser

Nu da databasens ejerskab og grupper er i orden, mangler der i princippet kun at blive indlæst data. Det er dog en kæmpe fordel at bestemme sig for, hvilke tilladelser de enkelte grupper skal have, inden der indlæses data.

- Åbn den nye database
- Vælg Tilladelser fra menuen Sikkerhed fra databasevinduet

Brugernavn/gruppenavn: Administratorer  
Gæster  
**hformadmin**  
hformbruger  
Brugere

Objektnavn: <Aktuel database>

Vis:  Brugere  Grupper    Objekttype: Database

**Tilladelser**

<input checked="" type="checkbox"/> Åbne/køre	<input type="checkbox"/> Læse data	Tildel
<input checked="" type="checkbox"/> Åbne eksklusivt	<input type="checkbox"/> Opdatere data	
<input type="checkbox"/> Redigere design	<input type="checkbox"/> Indsætte data	Luk
<input type="checkbox"/> Administrere	<input type="checkbox"/> Slette data	

Aktuel bruger: nyadm

- Den nye gruppe 'hformadmin' skal have alle tilladelser på alle objekttyper inkl. databasen.
- Den nye gruppe 'hformbruger' skal have udvalgte tilladelser, der på den ene side sikrer, at gruppens brugere kan udføre deres arbejde og på den anden side sikrer, at de ikke kan 'lave ulykker'.
- Standardgrupperne skal typisk have frataget alle deres tilladelser. Ellers kan man ikke styre, hvem der kan tilgå databasen.





## Import af database

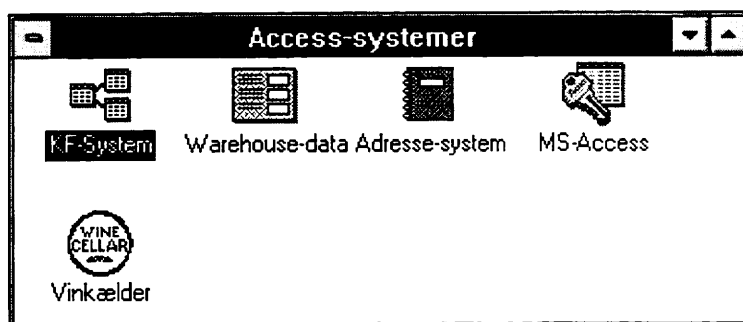
Nu da arbejdsgruppe, konti, ejerskab og tilladelser er på, plads mangler der kun at blive indlæst data, før basen er færdigsikret.

- Åbn databasen, der skal indlæses i
- Vælg Tilføjelsesprogrammer i menuen Filer fra databasevinduet
- Vælg Importer database...
- Vælg databasen, der skal indlæses fra
- Slet den oprindelige databasefil og omdøb evt. den nye databasefil

Bemærk, at fordi det kun er objekter, der importeres, vil eventuelle brugerdefinerede værktøjslinier gå tabt under importen.

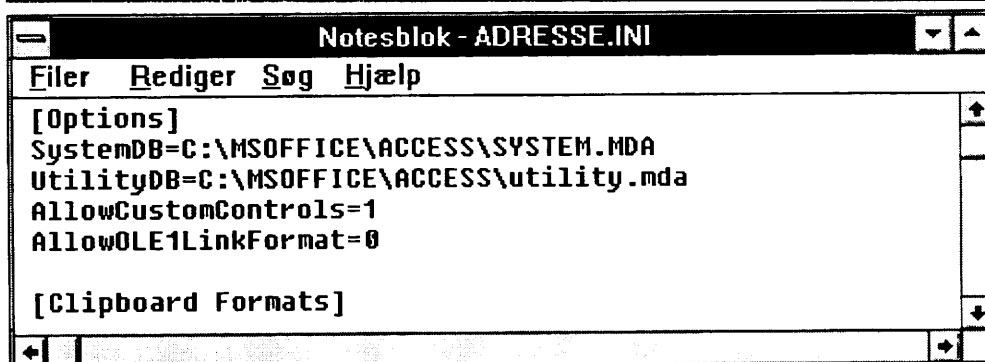
## Oprettelse af egen INI-fil

Ofte ønsker man at køre med flere systemer på en gang, og dermed ofte med flere forskellige sikkerhedsmodeller. Man ønsker måske at køre med et antal færdigsyede systemer samt den almindelige Access til brug for nyudvikling. I denne situation er der to muligheder. Man kan enten skifte arbejdsgruppe når man skifter system, v.h.a. arbejdsgruppeadministratoren, eller man kan oprette en speciel INI-fil for hvert enkelt system. Denne løsning er langt at foretrække, fordi den dels fjerner problemer i.f.m. arbejdsgruppeskift, og dels giver en række andre muligheder for at optimere det enkelte system INI-fils parametre.

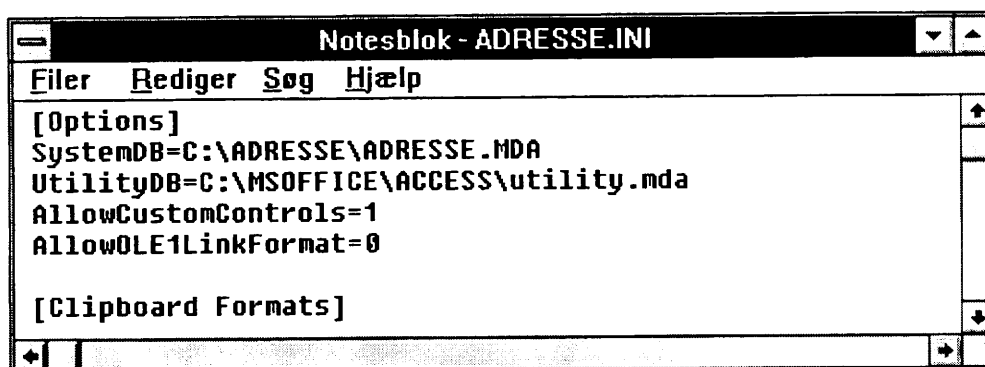


Følgende arbejdsgang anvendes:

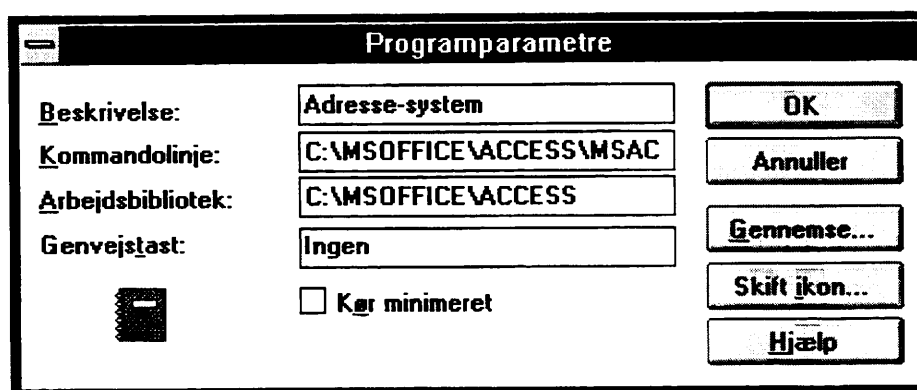
- Kopier MSACC20.INI, som er placeret i WINDOWS-kataloget. Kald den evt. det samme som MDB'en - f.eks. ADRESSE.INI. ADRESSE.INI kan placeres i WINDOWS-kataloget eller i samme katalog som MDB'en.
- Åbn ADRESSE.INI med en passende editor, f.eks. NOTEPAD.



- I sektionen [OPTIONS] tilrettes parametren SystemDB så den peger på den MDA, der ønskes anvendt i det aktuelle konfiguration.



- Når ADRESSE.INI aktiveres vil den samtidigt anvende ADRESSE.MDA
- Herefter mangler vi blot at oprette en ikon til systemet. Dette ikon skal dels starte Access med den korrekte database, og dels aktivere ADRESSE.INI i stedet for MSACC20.INI. Det nemmeste er at kopiere den almindelige Access-ikon.
- Aktivér ikonens programparametre



- I feltet kommandolinje skrives <ACCESS.EXE> /INI <INI-FIL> <MDB-FIL>, f.eks. C:\MSOFFICE\ACCESS\MSACCESS.EXE /INI C:\ADRESSE\ADRESSE.INI C:\ADRESSE\ADRESSE.MDB.



- Når ikonen aktiveres vil Access starte med at læse den angivne INI-fil, og dermed anvende den korrekte MDA-fil, og endelig indlæse den aktuelle MDB-fil.

## Andre kommandolinje-parametre

<database>	Åbner den angivne database. Medtag et stinavn, hvis det er nødvendigt.
/Excl	Åbner den angivne database med eksklusiv adgang. Hvis du vil åbne en database med delt adgang i et flerbrugermiljø, skal du skrive kommandolinjen uden denne indstilling.
/Ro	Åbner den angivne database med skrivebeskyttet adgang.
/User <brugernavn>	Starter Microsoft Access med det angivne brugernavn.
/Pwd <adgangskode>	Starter Microsoft Access med den angivne adgangskode.
/Ini <initialiseringsfil>	Starter Microsoft Access med indstillingerne i den angivne initialiseringsfil i stedet for standardfilen MSACC20.INI. Initialiseringsfilen skal have samme struktur som standardfilen MSACC20.INI.
/Compact <destinationsdatabase>	Komprimerer den angivne database foran indstillingen /Compact og lukker derefter Microsoft Access. Hvis du vil komprimere til et andet navn, skal du angive en destinationsdatabase.
/Repair	Reparerer den angivne database, og lukker derefter Microsoft Access.
/X <makro>	Starter Microsoft Access og kører den angivne makro. En anden måde at starte en makro, når du åbner en database, er ved at oprette en AutoExec-makro.
/Cmd	Angiver, at det, der kommer efter på kommandolinjen, er værdien, der vil blive returneret af funktionen Command i Access Basic. Denne indstilling skal være den sidste indstilling på kommandolinjen. Du kan benytte et semikolon (;) som et alternativ til /Cmd.

## Andre former for sikkerhed

I det allerede gennemgåede stof er der sikkerhedsfaktorer som fx at starte op gennem makroen Autoexec, at begrænse menukommandoerne til et minimum samt at skræddersy de værktøjslinier som brugeren ser. Endelig kan man vælge at kryptere databasen.



## Makroen Autotaster

Et sidste sikkerhedsaspekt omhandler tastetryk. F.eks. kan brugeren en række forskellige steder i Access taste F11, og dermed aktivere databasevinduet. Normalt ønsker man at fjerne denne mulighed, og dette gøres ved at "fange tastetryk" med en makro. Hvis der findes en makro kaldet Autotaster, vil Access udføre handlinger, som svarer til de tastetryk, der er angivet i makroen.

Følgende fremgangsmåde kan anvendes:

1. Opret en ny makro
2. Aktivér makronavnskolonnen ved at vælge Vis, Makronavne
3. Skriv tegnkode i kolonnen makronavn, og vælg tilhørende handlinger. Hvis ingen handlinger skal udføres vælges AnnullerHændelse.
4. Gem makroen under navnet Autotaster. (VK: Autokey)

Makronavn	Handling
^P	AnnullerHændelse
^A	AnnullerHændelse
{F11}	AnnullerHændelse

Handling : BeskedBoks  
 Meddelelse : Det nå du ikke  
 Bsp : Ja

## Tegnkoder

### Tegnkode

### Taster der fanges

^A or ^4	CTRL+en bogstav- eller taltast
{F1}..{F12}	En hvilken som helst funktionstast
^{F1}..^{F12}	CTRL+en hvilken som helst funktionstast
+{F1}..+{F12}	SKIFT+en hvilken som helst funktionstast
{INSERT}	INS
^{INSERT}	CTRL+INS
+{INSERT}	SKIFT+INS
{DELETE} eller {DEL}	DEL
^{DELETE} eller ^{DEL}	CTRL+DEL
+{DELETE} eller +{DEL}	SKIFT+DEL

Kritiske tastetryk :

F11 : DB vindue kommer op



## Checkliste

1. Opret ny arbejdsgruppe. Husk at gemme arbejdsgruppe id.
2. Åbn en eksisterende database
3. Opret ny adm-gruppe
4. Opret ny adm-bruger
5. Opret evt. ny brugergruppe og bruger
6. Tilknyt ny adm-bruger til ny adm-gruppe og til gammel adm-gruppe
7. Opret password på gammel administrator
8. Luk Access
9. Åbn Access - brug den nye adm-bruger
10. Opret en ny database
11. Check at ny adm-bruger er ejer af denne base
12. Fratag gammel adm-gruppe og gammel bruger-gruppe passende rettigheder
13. Fratag gammel adm-bruger passende rettigheder
14. Fjern gammel adm-bruger fra gammel adm-gruppe
15. Importer database, som skal sikres
16. Kryptér databasen
17. Opret INI-fil ved at kopiere MSACC20.INI
18. Tilpas ny INI-fil således at SystemDB i [Options] peger på den nye arbejdsgruppe.
19. Opret et passende ikon hvor kommandolinjepar metre /INI peger på den ny INI-fil.
20. Fjern menulinjer og værktøjslinjer, hvor det er passende.
21. Opret makroen Autotaster, for at fange passende tastetryk.

> 14. Fjern gammel adm-bruger fra gammel adm-gruppe < 14B. tag kopi af tomme DB

Kryptering → 10% perf. nedgang

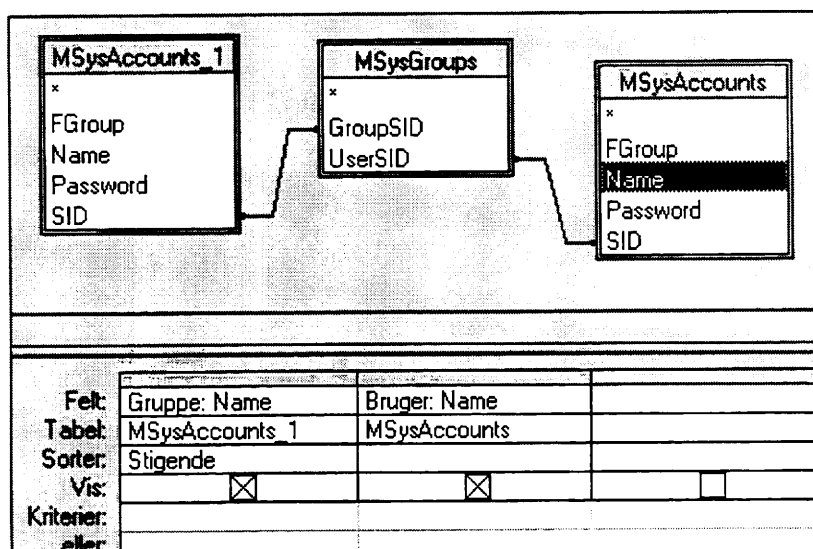


## Gruppe- og brugertabeller

Det er muligt at undersøge brugere og grupper på tabelniveau. Det betyder at man kan studere tabeller, der indeholder information om brugere, deres gruppemedlemskab, samt deres ejerforhold på databaseobjekter. Hele fidusen er at anvende to tabeller fra arbejdsgruppedatabasen (SYSTEM.MDA) samt et antal systemtabeller fra den aktuelle MDB-fil.

### Forespørgsel, der viser brugere og grupper

1. Indstil Access således at systemobjekter vises (Indstillinger, Generelt, Vis systemobjekter).
2. Tilknyt tabellerne MSysAccounts og MSysGroups fra arbejdsgruppen
3. Opret en forespørgsel, der anvender de to tabeller - se illustration.



	Gruppe	Bruger
▶	Admins	adradmin
	Admins	admin
	adradmins	adradmin
	adbrugere	adbrugere
	adbrugere	adradmin
	Guests	adradmin
	Guests	guest
	Users	adbrugere
	Users	adradmin
	Users	admin



# Øvelse 5.1

Formål: at sikre den database, applikationen ligger i

1. Gennemfør en sikring af databasen 'appl.mdb'

Støttepunkter:

- 1 • opret en ny arbejdsgruppe (arb.grp.adm)
- 2 • opret en ny adm-bruger
- 3 • tilføj gruppen Administratorer til adm-brugeren
- 4 • aktiver logonproceduren admin /pwd:admin
- 5 • start Access som den nye adm-bruger og opret en ny database
- 6 • tilføj en adm- og en bruger-gruppe til arbejdsgruppen
- 7 • rediger gruppernes tilladelser for alle objekttyper
- 8 • importer den oprindelige database

Nam : P72  
 Org : NULL  
 ID : P72  
 C:\MSOFFICE\ACCESS\GE-MDA

Num:geadm, id: 1234  
 L ikke password  
 geadm/1234  
 C:\ge\appl.mdb  
 Gruppe, Ny : nyadm/nyadm  
 nybrug/nybrug  
 Brugere : geadm tilføjet medlem af nyadm  
 nybrug tilføjet medlem af nybrug  
 ge: DB: åbn/kør

2. Undersøg sammen med din sidemand, om din database opfylder de tilladelser, du mener at have givet de forskellige brugere.
3. Undersøg sammen med din sidemand, om standardbrugere kan komme på basen og i givet fald hvilke operationer, de kan udføre.  
 Tip: Vælg Joinforbind i Arbejdsgruppeadministrator og sæt arbejdsgruppen tilbage til system.mda. Genstart Access.

4. Opret INI-fil til den sikrede applikation. C:\ge\appl-acc.ini
5. Opret ikon, der automatisk starter Access med den nye INI-fil og den sikrede MDB-fil.  
 C:\TUI C:\ge\appl-acc.ini C:\ge\appl.mdb
6. Opret makroen Autotaster, således at F11 og CTRL-F4 fanges.

1.7 : 'nyadm' gruppe gives tilladelse til alt  
 'nybrug' gruppe : DB: åbn/kør



## Øvelse 5.2

*Formål: at undersøge bruger- og gruppeta-beller.*

1. Opret en forespørgsel, der udskriver gruppe-og brugerinformation fra din arbejdsgruppe. Eksperimentér med at undersøge andre felter i de to tabeller.

Ekstra:

Opret en forespørgsel, der joiner

`MSysObjects.Id <=> MsysACEs.objectid`

`MsysACEs.SID <=> MSysAccounts.SID`

Forespørgselen skal vise felterne Name fra MSysObjects og MsysAccounts.

Hvad viser denne forespørgsel ?





## 6. Access Basic

Access Basic er et programmeringssprog, der er indbygget i Access. Sproget giver muligheder for at løse opgaver, der ikke kan klares med f.eks. makroer eller forespørgsler. Formålet med dette kapitel er at give et overblik over hvordan Access Basic kan anvendes til udvikling af Access applikationer. Access Basic kode placeres i moduler. Sådanne moduler kan enten være generelle for hele databasen eller knyttet til en formular eller rapport.

Globale moduler indeholder primært funktioner, der således kan anvendes overalt i Access.

Tilknyttede moduler indeholder subprocedurer der aktiveres når der på formularen/rapporten indtræffer hændelser - altså præcist som makroer.

### Hvad kan Access Basic ikke ?

Access Basic kan ikke erstatte makroerne AUTOEXEC og AUTOTASTER. Endvidere kan Access Basic ikke erstatte menumakroerne, som er gennemgået tidligere.

### Hvad kan Access Basic ?

Efterhånden som man arbejder med Access løber man før eller siden ind i problemer, der kun kan løses v.h.a. Access Basic. Hovedlinjerne i sprogets styrke er følgende:

- Udvikling af funktioner, der ikke er tilgængelige i Access.
- Enkel og nem tildeling af værdier til objekter - på samme måde som makrohandlingen Angivværdi.
- Generel procedural løsning af problemstillinger.
- Effektiv fejlhåndtering.
- Mulighed for kommunikation med Windows API'et.
- Mulighed for OLE- og DDE-baseret integration.
- Mulighed for opbygning af logisk og overskuelig programstruktur, der lettere kan dokumenteres end f.eks. makroer.
- Mulighed for at skrive generel kode.

### Hvad indbærer arbejdet med Access Basic

Access Basic er et programmeringssprog, der på mange måder ligner Visual Basic for Applications. Sproget er et såkaldt 3.generationsværktøj, og det rummer en lang række sætninger, kommandoer o.s.v., der naturligvis skal læres. Hvis man har erfaring fra lign. værktøjer, f.eks. Pascal, Visual Basic eller lign., vil overgangen til Access Basic være relativt enkel. Under alle omstændigheder kræves der et solidt kendskab til selve Access.



## Funktioner og subprocedurer

**Funktioner** returnerer normalt en værdi, som kan anvendes i udtryk og som datakilde til kontrolelementer. Værdien er normalt et resultat af en beregning. I Access findes der en lang række funktioner, som kan anvendes uden brug af moduler, fx Date(), som returnerer den aktuelle dato. En funktion kan aktiveres fra en makro ved hjælp af handlingen KørKode(), og kan generelt anvendes alle steder, hvor man i øvrigt anvender funktioner, f.eks. i formularer eller forespørgsler.

**Subprocedurer** returnerer ikke værdier. I stedet udføres en række handlinger. Oftest aktiveres en subprocedure af en hændelse - fx åbning eller lukning af en formular. En subprocedure kan ikke køres fra en makro.

### Første subprocedure

Access C, Pascal  
Sub ≈ procedure

I det følgende gennemgås hvordan man opretter en simpel subprocedure, der tænder og slukker for et felt på en form.

### Eksempel

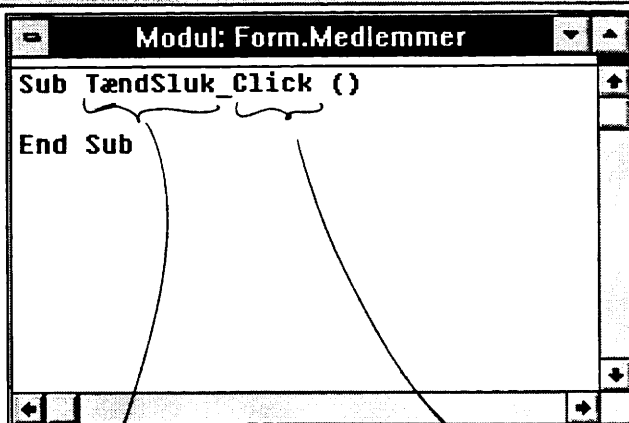
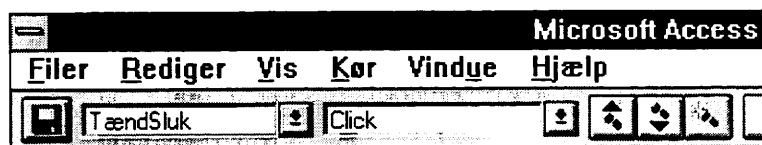
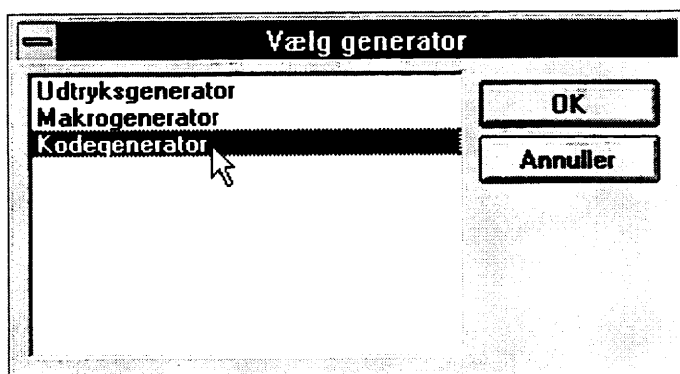
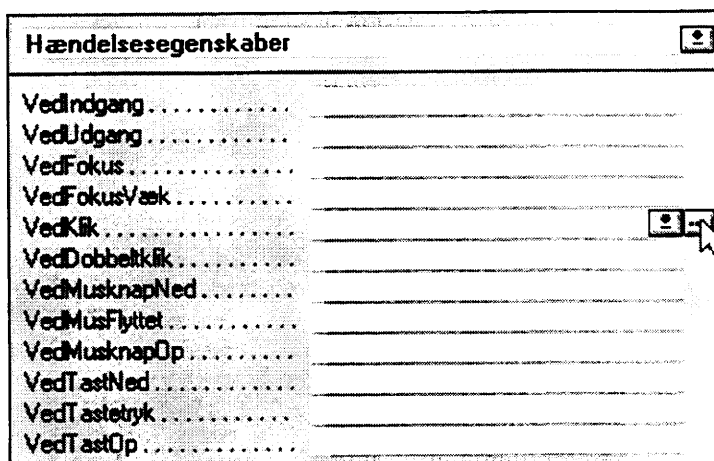
På nedenstående form ønsker vi at oprette en knap, der tænder og slukker for underformen.

Medlemmer							
Medlemsnr:	1	Navn:	Ole Olsen	I-dato:	01-08-91		
Adresse:	Overgade 57						
Postnummer:	4600	By:	Roskilde				
Reservationer							
Banenr	Tidspunkt	Spilledato	Res-dato	Baneleje	Leje af udstyr		
2	13	12-02-93	19-01-93	kr 101,00	kr 45,00		
2	15	08-05-92	15-04-92	kr 110,00	kr 42,00		
2	15	01-07-93	07-06-93	kr 30,00	kr 37,00		
2	16	19-10-92	28-09-92	kr 30,00	kr 49,00		
2	16	17-01-94	27-12-93	kr 30,00	kr 64,00		
3	14	04-03-94	10-02-94	kr 30,00	kr 68,00		
4	14	10-02-93	02-02-93	kr 0,00	kr 0,00		
Post: 1		af 38					



Følgende fremgangsmåde kan anvendes:

1. Åbn egenskabsvinduet for det objekt, der skal aktivere subproceduren.
2. Find den passende egenskab.
3. Aktiver kodegeneratoren.
4. Skriv Access Basic koden - herom senere



Kontrol element

hændelse



Herefter vil vi kigge nærmere på den kode, der skal skrives. Ved tryk på knappen TÆNDSLUK skal underformularen RESERVATIONER tændes hvis den er slukket og slukkes hvis den tændt. Dette gøres ved at tildele en passende værdi til underformularens egenskab SYNLIIG. En procedure der slukker for underformularen ser ud som følger:

```
Sub TændSluk_Click()
  [Reservationer].Visible = False
End Sub
```

En procedure der tænder for underformularen ser ud som følger:

```
Sub TændSluk_Click()
  [Reservationer].Visible = True
End Sub
```

*fuld adresse : Forms![Min Formular]![reservation].visible*

En procedure der tænder og slukker for underformularen ser ud som følger:

```
Sub TændSluk_Click()
  If [Reservationer].Visible = True Then
    [Reservationer].Visible = False
  Else
    [Reservationer].Visible = True
  End If
End Sub
```

Alternativt kan den skrives ved at anvende den logiske operator NOT:

```
Sub TændSluk_Click()
  [Reservationer].Visible = Not [Reservationer].Visible
End Sub
```

Ønsker man samtidigt at ændre på knappens titeltekst kan dette gøres på flg. måde:

```
Sub TændSluk_Click()
  If [Reservationer].Visible = True Then
    [Reservationer].Visible = False
    [TændSluk].Caption = "Tænd"
  Else
    [Reservationer].Visible = True
    [TændSluk].Caption = "Sluk"
  End If
End Sub
```

*|*  
*Titeltekst*

## Subprocedure aktiveret fra timer

Det er muligt at tilknytte en makro eller subprocedure til en formulars timer. Timeren kan bedst sammenlignes med et ur, hvor den aktuelle kode afvikles én gang for hvert timerudløb.



Hændelsesegenskaber	
VedFokusVæk .....	
VedKlik .....	
VedDobbeltklik .....	
VedMusknopNed .....	
VedMusFlyttet .....	
VedMusknopOp .....	
VedTastNed .....	
VedTasttryk .....	
VedTastOp .....	
VedFej .....	
VedTimerudløb .....	[Hændelsesprocedure]
Timerinterval .....	1000

På ovenstående figur ses hændelsesegenskaberne for en given formular. Hændelsen VedTimerUdløb angiver hvilken kode, der skal køre når hændelsen indtræffer. Timerinterval angiver antal millisekunder mellem timerens udløb.

Timeren kan f.eks. bruges til at vise et ur i formularens titeltekst. Dette kan gøres på følgende måde:

1. Aktivér formularens hændelsesegenskaber. *• hvidt felt øverst TV*
2. Sæt Timerinterval til 1000.
3. Aktivér kodegeneratoren for hændelsen VedTimerudløb.
4. Skriv koden, der indstiller formularens titeltekst.

Kodesætningen kan skrives på flere måder. Det vigtigste at forstå er, at objektet formulars egenskab titeltekst skal forandres af koden. Koden kan f.eks. se sådan ud:

```
Me.Caption = Time ' Objektet Me refererer altid til den
                  ' aktuelle formular. Funktionen Time
                  ' returnerer klokkeslæt.
```

## Automatisk udlogging v.h.a. subprocedure

Timeren kan f.eks. bruges til at skrive en automatisk udlogningsprocedure. Ideen er, at Access afsluttes, hvis musen ikke har været rørt i en time.

Følgende struktur anvendes:

*Max én timer pr. formular*



1. På en formular placeres en skjult tekstboks. Når formularen aktiveres skrives værdien 3600 i boksen.
2. Ved timerudløb tælles tekstboksens værdi én ned. Hvis værdien er nul afsluttes Access.
3. Når musen bevæges hen over passende områder af formularen sættes værdien af tekstboksen atter til 3600.

### Eksempel:

På fomularen Medlemmer ønsker vi at implementere udlogningsfunktionen. Først placeres en tekstboks på formularen kaldet TidTilbage. Dernæst instilles timeren til 1000. Når formularen åbnes skrives værdien 3600 i denne boks. Dette gøres med flg. kodestump:

```
Sub Form_Open()
    [TidTilbage] = 3600
End Sub
```

Ved hvert timerudløb tælles tekstboksens værdi ned:

```
Sub Form_Timer ()
    [TidTilbage] = [TidTilbage] - 1
    If [TidTilbage] = 0 Then
        DoCmd Quit A_SAVE
    End If
End Sub
```

Endelig sættes værdien af tekstboksen til 3600 når musen flyttes ind over f.eks. formularens detaljesektion. Dette gøres med flg. kode på hændelsen VedMusFlyttet:

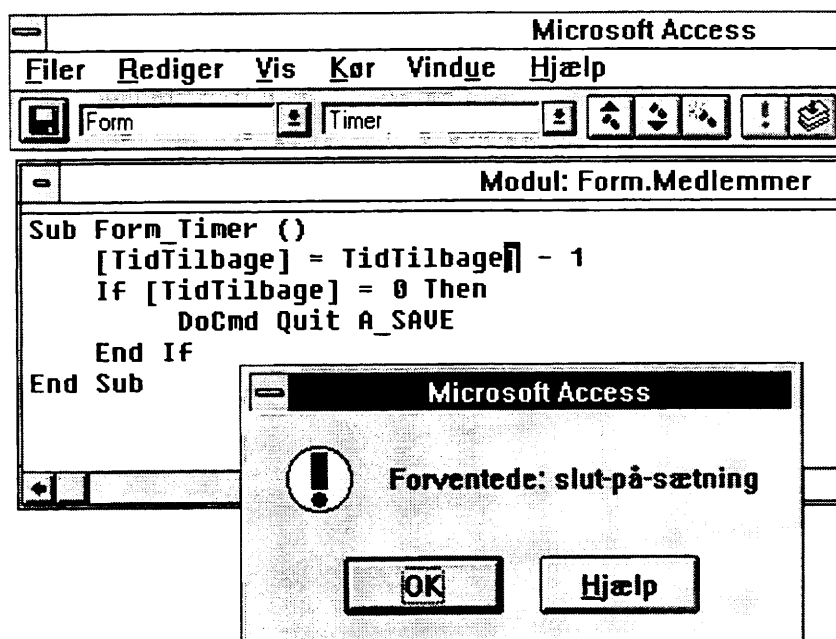
```
Sub Detalje0_MouseMove ()
    [TidTilbage] = 3600
End Sub
```

### **Access Basic udviklingsmiljøet**

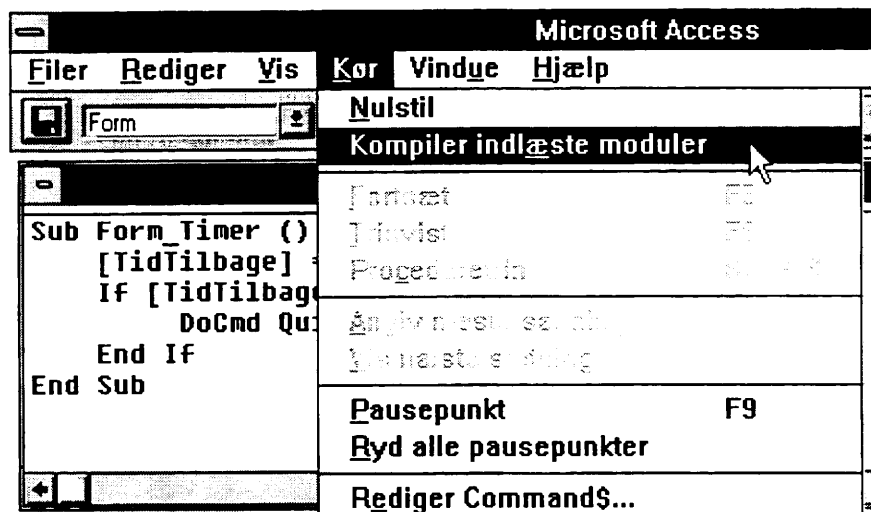
Efter at have prøvet at programmere i Access Basic, vil vi nu kigge et øjeblik på udviklingsmiljøet. Access Basic koden syntakscheckes hver gang en linje forlades. Hvis der er fejl i linjen fremkommer en meddelelse:

*Functions Iudsæt Tegn*

```
Dim CT As Control
Set CT = Forms![formular 1]! UDDATA
CT.SetFocus
Husk_Aendring
SendKey Tegn, False
End Function
```



Når koden er skrevet færdig skal den kompileres. Det betyder at den dels syntakscheckes, og dernæst oversættes til et format, der øger afviklingstiden.



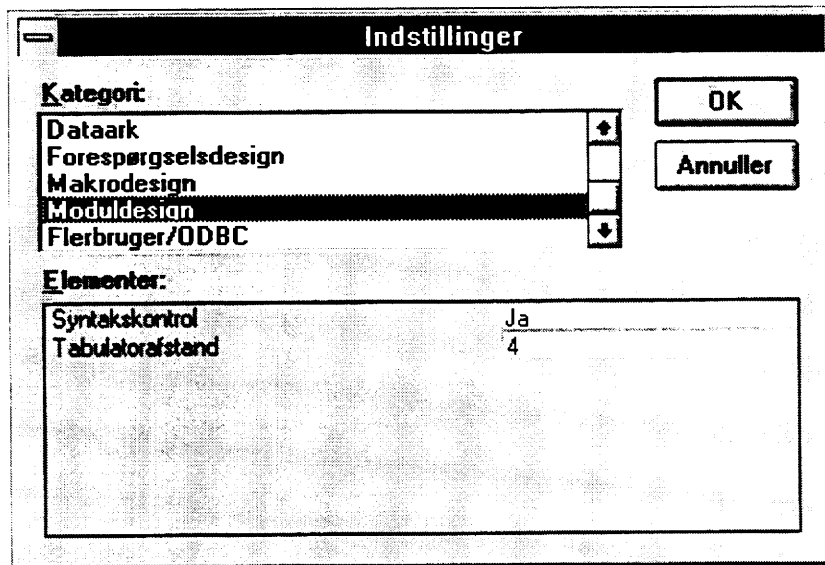
Det er muligt at fjerne den automatiske syntaks kontrol, som udføres ved linjeskift. Dette gøres ved at vælge menupunkt Vis, Indstillinger, og derefter vælge kategorien Moduldesign:

Udvælgelses forespørgsel

Felt: Frugt: Besogn.Frugt  
 = Udvælg

My procedure: Type 'Sub' eller 'Function'

{  
 Kan kun kaldes fra Function.  
 Kan ikke returnere værdi
 }



## Objekters metoder og egenskaber

En stor del af programmeringen i Access Basic består i at manipulere objekters egenskaber. Den generelle syntaks for dette er:

```
Objekt.Egenskab = værdi
```

Hvis der refereres til kontrolelementer anvendes knækparenteser, f.eks.:

```
[AntalFundet].Visible = False
```

```
[AntalFundet] = 17
```

```
[AntalFundet] = [AntalFundet] + 1
```

Hvis man ønsker at anvende en metode gøres dette ganske enkelt ved at skrive:

```
Objekt.Metode
```

## Eksempel

I nedenstående kodeeksempel, der stammer fra søgeformularen, der blev omtalt i kapitel 2, anvendes en metode på et objekt af typen Liste.

Sub Opdater_Click ()	
[AntalFundet] = "Søger ..."	Skriv til kontrolelement
[AlbumListe].Requery	<b>Anvend metoden Requery</b>
AntalHits = [AlbumListe].listcount	Aflæs egenskaben Listcount
If AntalHits > 0 Then	
[AntalFundet] = AntalHits & " fund"	Skriv til kontrolelement
[AlbumListe].visible = True	Indstil egenskaben Visible
[DetailType].visible = True	Indstil egenskaben Visible
Else	

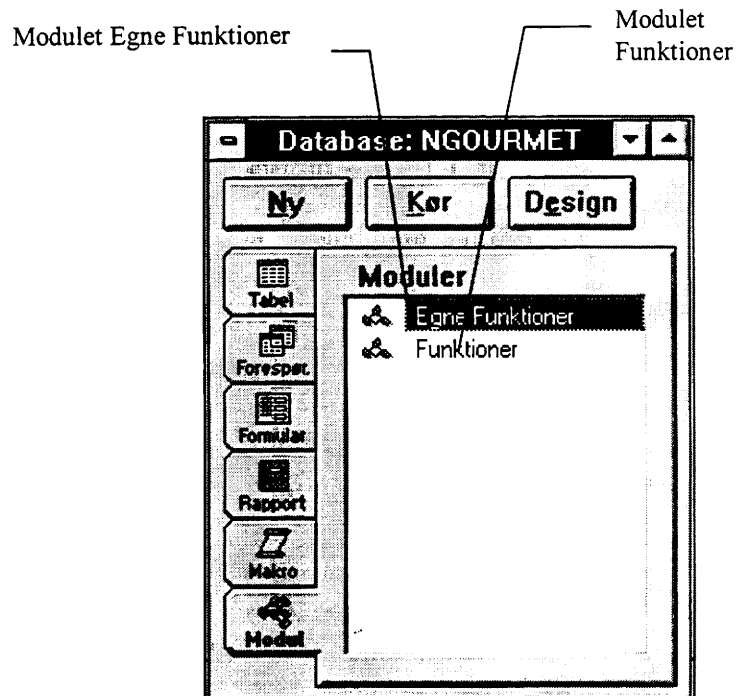




```
[AntalFundet] = "Ingen sange fundet"      Skriv til kontrolelement
End If
End Sub
```

## Globale moduler

Hvis man ønsker at skrive funktioner, der kan anvendes af alle dele af applikationen, skal disse placeres i et globalt modul. Modulerne oprettes og vedligeholdes fra databasevinduet fane MODUL.



Globale moduler anvendes som oftest til at rumme funktioner - altså beregninger, der returnerer en værdi baseret på et antal argumenter. Funktionerne kan anvendes overalt i Access.

### Eksempel:

Der skal konstrueres en funktion 'BeregnFragt' som beregner fragtomkostningerne for en given faktura. Reglen er, at fragten er gratis, hvis den samlede fakturatotal er større end 3500 kroner og ellers beregnes som 10 % af fakturatotalen.

Formularen indeholder allerede feltet FakturaTotal. Funktionen skal anvendes til at udfylde et felt på en formular, og skal som argument have fakturatotalen

1. Opret et modul, hvori den nye funktion kan placeres.
2. Skriv funktionen - herom senere.
3. Skift til den formular, hvor fragtberegningen skal anvendes.



4. Tilknyt funktionen til et nyt kontrolelement.

<b>Fakturatotal:</b>	kr 4.000,00	<b>Fakturatotal:</b>	kr 1.200,00
<b>Fragt:</b>	kr 0,00	<b>Fragt:</b>	kr 120,00
<b>Ialt:</b>	kr 4.000,00	<b>Ialt:</b>	kr 1.320,00

<b>Fakturatotal:</b>	FakturaTotal
<b>Fragt:</b>	=BeregnFragt([FakturaTotal])
<b>Ialt:</b>	=[FakturaTotal]+[Fragt]

Kald af funktion  
BeregnFragt, som  
er skrevet i Access  
Basic.

## Oprettelse af en funktion i Access Basic

1. Opret et nyt modul ved at vælge Modul, Ny.
2. Vælg Rediger, Ny Procedure.

3. Vælg Function og indtast Navn
4. Vælg OK
5. Indtast Access Basic kode for funktionen.
6. Vælg Filer, Gem.
7. Vælg Filer, Afslut.
8. Åbn den relevante formular i designvisning.
9. Opret en ubunden tekstboks.
10. Skriv =BeregnFragt([FakturaTotal]) i feltets kontrolelementkilde.
11. Kør formularen.

typecheck på et tidligere tidspunkt  
datatype

as currency

as double  
as currency



## Opbyg funktion

Funktionens start- og slutpunkt indsættes altid automatisk

Her skiftes til andre funktioner i modulet.

Her skrives eventuelle argumenter adskilt af et semikolon



```
Function BeregnFragt (FakturaTotal)
If FakturaTotal > 3500 Then
    BeregnFragt = 0
Else
    BeregnFragt = FakturaTotal * .1
End If
End Function
```

- Funktionsnavn og argument
- Hvis Faktura er større end 3500
- så skal returneres 0
- ellers
- returner 10 % af faktura
- slut if-then-else sætning
- slut funktion

Resultatet af beregningen skal altid lægges over i funktionens navn.

msgbox "Frugt er gratis"

### TIP:

Det er muligt at "snyde" en funktion, så den ikke returnerer en værdi. Dette gøres ved simpelt hen at udelade den kodesætning, hvor resultatet af beregningen lægges over i funktionens navn. Fordelen herved er, at man på denne måde kan aktivere en funktion fra en makro uden nødvendigvis at skulle returnere en værdi.

Vis, Brugervalgbar : ? BeregnFragt(1000)

## Eksempler på funktioner

Funktionen beregner alder i år mellem to datoer, nemlig fødselsdatoen og dags dato. Funktionen fanger ikke negativ alder. Funktionen anvender de indbyggede funktioner Day(), Year og Month.

```
Function Alder (Fdato, DatoIdag)
    If Month(DatoIdag) < Month(Fdato) Or (Month(DatoIdag) = _
    Month(Fdato) And Day(DatoIdag) < Day(Fdato)) Then
        Alder = Year(DatoIdag) - Year(Fdato) - 1
    Else
        Alder = Year(DatoIdag) - Year(Fdato)
    End If
End Function
```

Funktionen Initcap returnerer et navn, hvor første bogstav er stort og øvrige bogstaver små. Funktionen anvender de indbyggede funktioner UCase(), LCase(), Left(), Right() og Len().

```
Function Initcap (Inavn)
    Initcap = UCase(Left(Inavn, 1)) & LCase(Right(Inavn, Len(Inavn) - 1))
End Function
```



Funktionen Fac beregner fakultet af et givent tal.

```
Function Fac (X As Integer) As Double
  If X < 0 Or X > 170 Then
    Fac = 0
  ElseIf X = 0 Then
    Fac = 1
  Else
    Fac = X * Fac(X - 1)
  End If
End Function
```

Funktionen SlutMåned beregner sidste dato i måneden, hvor inddata er en given dato. Funktionen anvender de indbyggede funktioner DateSerial(), Year() og Month().

```
Function SlutMåned (InputDato As Variant) As Variant
  SlutMåned = DateSerial(Year(InputDato), Month(InputDato) + 1, 0)
End Function
```

## Brug af funktioner i forespørgsler

En funktion, der er skrevet i Access Basic kan som hovedregel anvendes alle steder, hvor almindelige funktioner bruges. F.eks. i forespørgsler:

Udvælgelsesforespørgsel: Medarbejderes alder			
Fornavn	Efternavn	Alder/dag	
Anders	Finnsen	54	
Johanne	Larsen	32	
Mette	Pedersen	58	
Søren	Buck	32	
Michael	Sørensen	33	
Robert	Kjeldsen	36	
Lise	Christensen	38	
Anne	Drewesen	30	
Anton	Hellum	46	
Torben	Svensen	23	
Caroline	Pallesen	23	
Jens	Børgesen	33	
John	Møller	35	
Lars	Poulsen	30	

Post: 2 af 14



Medarbejdere			
* Medarbejdere			
Medarbejdernr			
Efternavn			
Fornavn			
Stilling			
Fødselsdato			
Ansættelsesdato			
Felt:	Fornavn	Efternavn	Alder(dag: Alder([Fødselsdato].Date!))
Sorter:			
Vis:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kriterier:			
eller:			

## Datatyper i Access Basic

Datatype	Lager	Interval
Integer	2 byte	32.768 til 32.767.
Long(langt heltal)	4 byte	2.147.483.648 til 2.147.483.647.
Single(reelt tal med enkeltpræcision)	4 byte	3,402823E38 til 1,401298E-45 for negative værdier; 1,401298E-45 til 3,402823E38 for positive værdier og 0.
Double (reelt tal med dobbeltpræcision)	8 byte	1,79769313486232E308 til 4,94065645841247E-324 for negative værdier; 4,94065645841247E-324 til 1,79769313486232E308 for positive værdier og 0.
Currency (skaleret heltal)	8 byte	-922.337.203.685.477,5808 til 922.337.203.685.477,5807.
String	1 byte pr. tegn	0 til ca. 65.535 byte.
Variant	Efter behov	Enhver talværdi indtil intervallet for Double eller en vilkårlig tegntekst.



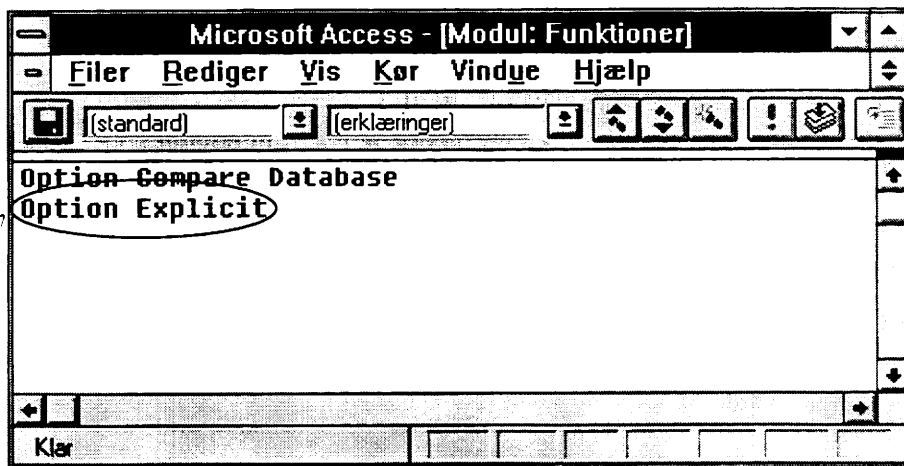
## Erklæringer af variable

En variable kan enten anvendes uden forudgående erklæring, eller kan erklæres før den anvendes. Under normale omstændigheder ønsker man altid at erklære variable. Dette gøres med kommandoen DIM.

```
Function Alder (Fdato As Variant, DatoIdag as Variant) as Integer
    Dim BeregnetAlder as Integer

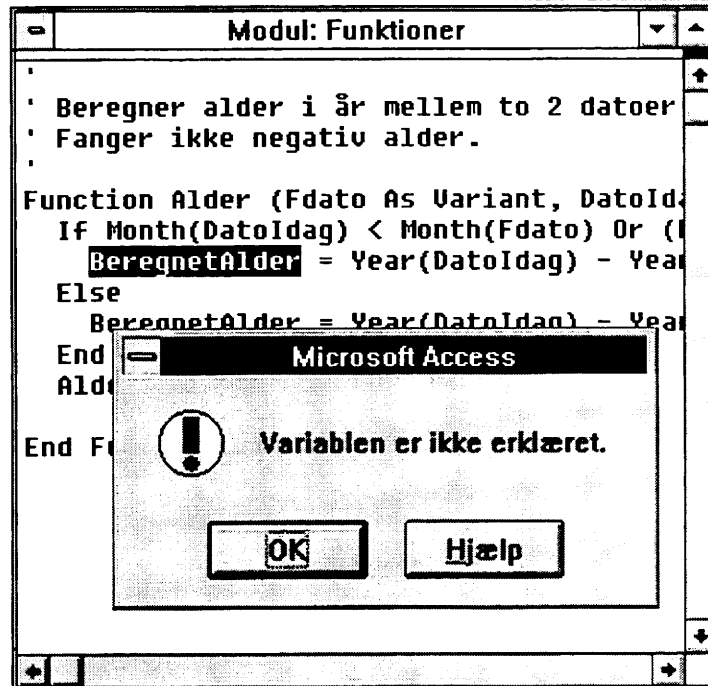
    If Month(DatoIdag) < Month(Fdato) Or (Month(DatoIdag) = _
    Month(Fdato) And Day(DatoIdag) < Day(Fdato)) Then
        BeregnetAlder = Year(DatoIdag) - Year(Fdato) - 1
    Else
        BeregnetAlder = Year(DatoIdag) - Year(Fdato)
    End If
    Alder = BeregnetAlder
End Function
```

Hvis man ønsker at gennemtvunge erklæring af alle variable - hvilket generelt er en meget god ide, gøres dette ved at skrive **OPTION EXPLICIT** i modulets erklæringsdel. Herved bliver fejlfinding meget lettere samtidigt med at lagerforbrug reduceres. Hvis en variabel anvendes uden erklæring fremkommer flg. meddelelse:



*Kræver eksplisit variable erklæring*

*I Access 7: Vis, indstillinger, Moduldesign: her kan sættes check på variable erklæring*



## Access Basic og formularer

Ved at anvende Access Basic er det muligt at opbygge avancerede formularer. I følgende eksempel opbygges en formular til styring af udskrivning af en rapport.



Ideen er, at man vælger et udskrivningsmedie fra gruppeboksen MedieType og derefter trykker på knappen Udskriv. Herefter udskrives rapporten Baneoversigt til den valgte medietype.

Den til hørende Access Basic kode er placeret på knappen hændelse VedUdskrivning:



```

Modul: Form.Printform

Sub Udskriv_Click ()

  If IsNull([MedieType]) Then
    MsgBox "Medie skal vælges"
    Exit Sub
  End If

  If [MedieType] = 1 Then
    DoCmd OpenReport "Banetotal", A_PREVIEW
  ElseIf [MedieType] = 2 Then
    DoCmd OutputTo A_REPORT, "Banetotal", A_FORMATTXT, "C:\BANER.TXT"
  ElseIf [MedieType] = 3 Then
    DoCmd OutputTo A_REPORT, "Banetotal", A_FORMATXLS, "C:\BANER.XLS"
  End If

End Sub

```

I koden anvendes følgende funktioner og kommandoer:

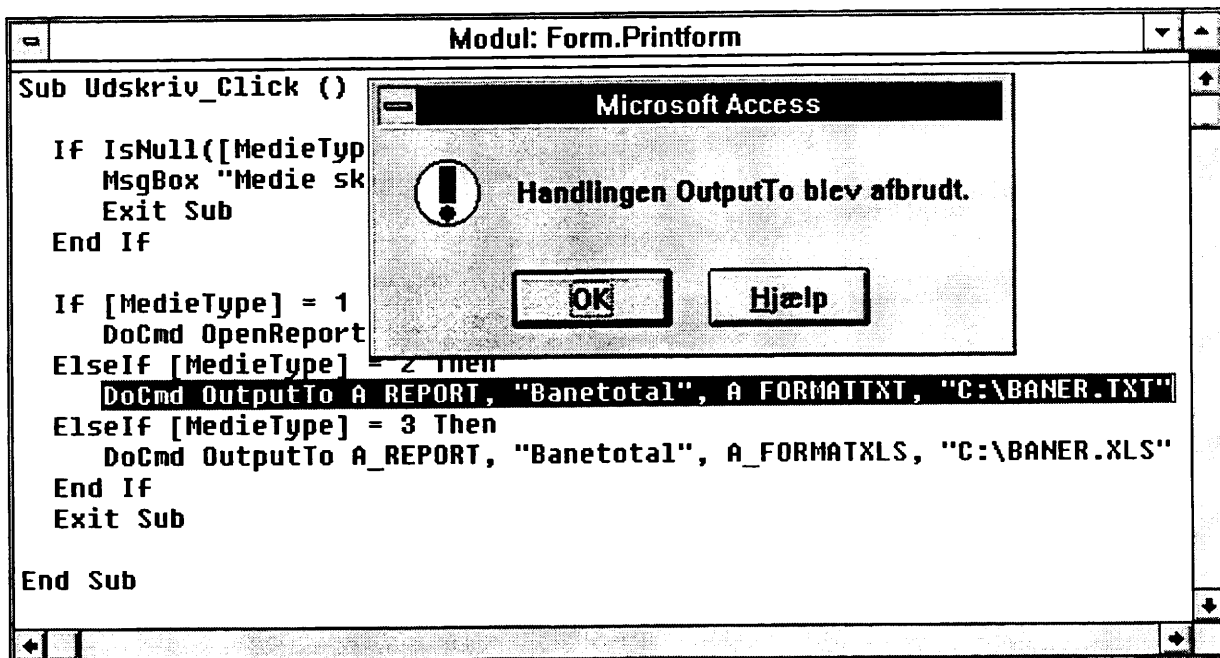
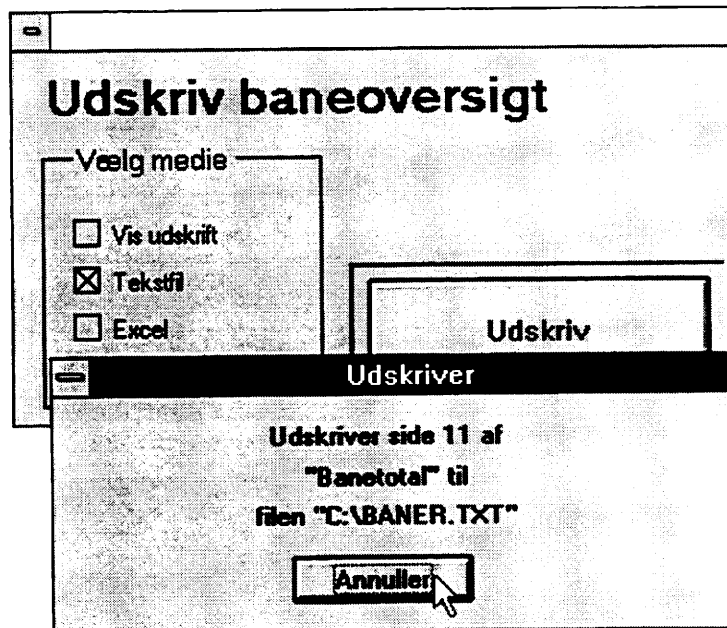
Funktionen ISNULL()	Undersøger om en medietype er valgt. Hvis intet er valgt i gruppeboksen returnerer funktionen SAND.
Kommandoen MSGBOX	Giver en besked til brugeren.
DoCmd OpenReport	Skriver en rapport til skærm eller printer
DoCmd OutputTo	Skriver et databaseobjekt til ekstern fil.

## Hvad med fejl ?

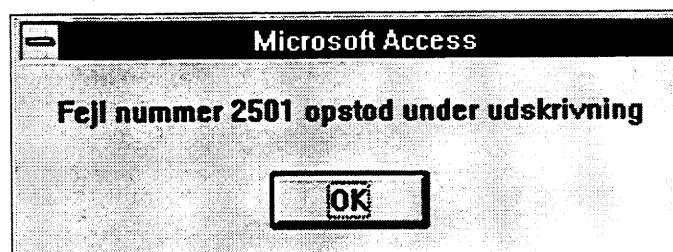
I foregående eksempel "går programmet ned", når brugeren trykker på knappen ANNULLER under eksporten til Excel eller tekstfil. Dette skyldes at der, set fra Access Basic, indtræffer en fejl når en handling annulleres eller afsluttes for tidligt. Derfor bør alle Access Basic funktioner og subs indeholde en fejlhåndtering. Fejlhåndteringen sikrer at programmet afsluttes på en fornuftig måde selvom noget går galt.

Hvis brugeren uden fejlhåndtering afbryder eksporten af data sker følgende:





Hvis man implementerer fejlhåndteringen vil koden f.eks. terminere med følgende besked:





Selve fejlhåndteringen er vist på nedenstående figur:

```

Modul: Form.Printform

Sub Udskriv_Click ()
  On Error GoTo Udskrivningsfejl

  IF IsNull([MedieType]) Then
    MsgBox "Medie skal vælges"
    Exit Sub
  End IF

  IF [MedieType] = 1 Then
    DoCmd OpenReport "Banetotal", A_PREVIEW
  ElseIf [MedieType] = 2 Then
    DoCmd OutputTo A_REPORT, "Banetotal", A_FORMATTXT, "C:\BANER.TXT"
  ElseIf [MedieType] = 3 Then
    DoCmd OutputTo A_REPORT, "Banetotal", A_FORMATXLS, "C:\BANER.XLS"
  End IF
  Exit Sub

Udskrivningsfejl:
  MsgBox "Fejl nummer " & Err & " opstod under udskrivning"
  Exit Sub

End Sub

```

- |                       |   |
|-----------------------|---|
| On Error Goto <Mærke> | Erklæring af fejlhåndteringen. <Mærke> er en henvisning til det sted i koden, hvorfra programafviklingen skal genoptages efter fejl.  |
| Funktionen Err()      | Returnerer nummeret på den aktuelle fejl. I eksemplet bruges Err() til at vise nummeret på den aktuelle fejl. Funktionen kan også anvendes i en IF-THEN-ELSE konstruktion til at udføre bestemte handlinger på bestemte fejl. |
| Funktionen Err()      | Returnerer en kort fejltekst.   |
| <Mærke>               | En tekst efterfulgt af kolon.   |

## Debug og Trace

I.f.m. programudvikling er der ofte behov at undersøge koden mens den afvikles. Man ønsker f.e.ks at stoppe programafviklingen på et bestemt tidspunkt for studere programmets tilstand.

Pausepunkt	Placeres i koden, og bevirker at programafviklingen afbrydes når Pausepunktet passerer.
Trinvis kørsel	Afviklingsform, hvor koden køres en linje af gangen.
Brugerudtryksvindue	Et vindue, der gør det muligt at overvåge variable og programobjekter i øvrigt.
Debug.print	Metode, der gør det muligt at udskrive variable til brugerudtryksvinduet.

Indsættelse af pausepunkt:



```

Modul: Form.Printform
Sub Udskriv_Click ()
  On Error GoTo Udskrivningsfejl

  IF IsNull([MedieType]) Then
    MsgBox "Medie skal vælges"
    Exit Sub
  End IF

```

```

Brugerudtryksvindue [Form.Printform:Udskriv_Click]
?[MedieType]
2
?[Medietype].backcolor
16777215
?Me.name
Printform

```



## API-kald

Access Basic er udstyret med mange indbyggede funktioner og faciliteter. Alligevel findes der en mængde ting, som ikke kan udføres med dette værktøj, fordi disse simpelthen ikke er indbygget i Access Basic. Man kunne f.eks. ønske at skrive et lille program, der sikrer at brugeren ikke kan skifte fra Access til andre applikationer, v.h.a. ALT-TAB. Dette kan ikke umiddelbart lade sig gøre, og det er her API-programmering kommer ind i billedet.

Application Programming Interface (API) er betegnelsen for en mængde funktioner, der er indbygget i Windows. Der findes en lang lang række funktioner, som kan udføre en masse forskellige ting. Ideen med API'et er at Windows tilbyder applikationerne, f.eks. Access, en række funktioner.

Hele pointen er at vi kan kalde API-funktioner fra Access Basic. For at kunne gøre dette skal vi bruge API-dokumentationen, som ikke følger med Access. Denne dokumentation følger med Windows SDK (Software Developers Kit), Visual Basic og Visual C++.

API'et er fællesbetegnelsen for en række filer, hvor de vigtigste er:

KERNEL.EXE	Indeholder funktioner til at indlæse og eksekvere applikationer, lagerhåndtering m.v.
USER.EXE	Funktioner til grænsefladeoperationer, f.eks. tastaturinput, vinduesoperationer m.v.
GDI.EXE	Grafiske funktioner.

Før en API-funktion kan anvendes af Access skal den erklæres, og dermed gøres kendt for Access. I erklæringen skal funktionens navn og placering i API-filerne angives.

### API-eksempel 1:

Vi ønsker at oprette en funktion, der gør hele Access modal. Det betyder at brugeren ikke kan sætte Access i baggrunden.

- Først oprettes et nyt modul, som skal indeholde funktionen.
- Erklær API-funktionerne i modulets erklæringsdel
- Skriv funktion, der skal anvende API-funktion

For at kunne skrive vores funktion, skal vi anvende to API-funktioner nemlig FindWindow(), som kan finde det vindue, der skal gøres modalt, og SetSysModalWindow(), som kan omdanne et vindue til at være modalt. Disse to funktioner er fundet ved at studere før omtalte API-dokumentation.



```

Filer Rediger Vis Kør Vindue Hjælp
[standard] [erklæringer]
Modul: API
Option Explicit
Declare Function SetSysModalWindow% Lib "User" (ByVal hWnd%)
Declare Function FindWindow% Lib "User" (ByVal lpzClassName As Any,
    ByVal lpzWindow As Any)
  
```

```

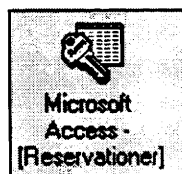
Modul: API
Function ModalAccess ()
    Const lpClassName = "OMain"
    Dim AccessHandle
    Dim x As Integer
    AccessHandle = FindWindow(lpClassName, 0&)
    x = SetSysModalWindow(AccessHandle)
End Function
  
```

Når denne funktion aktiveres, fra en knap eller lign., fastlåses Access som modal.

### API-eksempel 2:

I dette eksempel ønsker vi at ændre Access-vinduets titellinje, således at der ikke står Microsoft Access - [Formular-navn], men noget andet.

**Microsoft Access - [Reservationer]**



For at gøre dette skal vi atter anvende funktionen FindWindow() og sub'en SetWindowText, der ændrer et vindues titeltekst.



```

Modul: API

Option Explicit

Declare Function FindWindow% Lib "User" (ByVal lpzClassName As Any, _
    ByVal lpzWindow As Any)
Declare Sub SetWindowText Lib "User" (ByVal hWnd%, ByVal lpString$)
  
```

```

Modul: API

Function SetCaption (DenneCaption As String)
    Dim hWnd%
    hWnd% = FindWindow%("Omain", 0&)
    Call SetWindowText(hWnd%, DenneCaption)
End Function
  
```

## Udveksling af data med DDE

Det er muligt at anvende moduler til at håndtere udveksling af data mellem Access og andre programmer, fx MS-Word. Dataudvekslingen kan f.eks. udføres ved hjælp af DDE-kommandoer.

DDE (Dynamic Data Exchange) er en Windows-standard, som gør det muligt - fra det ene program - at udføre handlinger i et andet program. Fx er det muligt fra Access at udskrive et dokument i MS-Word.

### Eksempel:

I en Access-applikation findes en tabel, der indeholder kunders navne og adresser. Disse navne og adresser anvendes ofte til udskrivning af standardbreve. Disse breve udformes i MS-Word og flettes ved hjælp af dette programs flettefunktion. Derfor ønsker man at lave et modul i Access Basic, der udfører følgende handlinger:

1. Starter MS-Word (hvis det ikke allerede er startet), og indlæser flettedokumentet K\_BREVE.DOC, der anvender Access databasen som datakilde.
2. Udfører fletningen, og gemmer resultatet i et MS-Word dokumentet NYEBREVE.DOC
3. Afslutter MS-Word, og returnerer til Access.



Fordelen ved denne fremgangsmåde er at dataudvekslingen foregår automatisk, og at brugerne af systemet ikke behøver at tænke på om MS-Word er startet eller ej.

Fletningen startes af funktionen 'UdskrivIWord()'.

Access Basic kode	Beskrivelse
Function UdskrivIWord ()	Funktionens navn
On Error Resume Next	Ved fejl fortsæt da til næste linie
Dim Kanal, Dummy, StartWord	Erklær variable
Kanal = DDEInitiate("Winword", "system")	Etablér DDE-samtale med Winword
If Err Then	Hvis samtalen ikke kunne etableres...
Dummy = Shell("c:\msoffice\winword\winword", 2)	..skal programmet Winword startes...
Kanal = DDEInitiate("Winword", "system")	...og samtalen derefter etableres
StartWord = 1	Husk at programmet Winword blev startet
End If	Slut hvis-sætning
DDEExecute Kanal, "[SkStandardbibl ""F:\P66\WINDOWS"",0]"	Skift standardbibliotek i WinWord
DDEExecute Kanal, "[FilerÅbn .Navn = ""K_BREVE.DOC""]"	Åbn dokumentet K_BREVE.DOC
DDEExecute Kanal, "[BrevfletningTilDok]"	Udfør brev fletning
DDEExecute Kanal, "[FilerGemSom .Navn = ""NYEBREVE.DOC""]"	Gem de flettede breve som NYEBREVE.DOC
If StartWord = 1 Then	Hvis programmet WinWord blev startet...
DDEExecute Kanal, "[FilerAfslut 2]"	...så luk det.
Else	ellers
DDEExecute Kanal, "[FilerLuk 2]"	luk de flettede breve og flettedokumentet
DDEExecute Kanal, "[FilerLuk 2]"	
End If	Slut hvis-sætning
DDETerminate Kanal	Afslut samtalen med Winword
End Function	Slut funktion

### Funktionen DDEInitiate

Indleder en DDE-konversation (Dynamic Data Exchange) med et andet program.

#### Syntaks

Variable = DDEInitiate(program, emne)



### Sætningen DDEExecute

Benytter en åben DDE-kanal (Dynamic Data Exchange) til at sende en kommando til et andet program.

#### Syntaks

DDEExecute kanalnr, kommando

### Sætningen DDETerminate

Lukker den angivne DDE-kanal (Dynamic Data Exchange).

#### Syntaks

DDETerminate kanalnr

### Funktionen Shell

Kører et eksekverbart program.

#### Syntaks

Dummy = Shell(kommandostreng [, vinduestypografi])

Nedenstående tabel viser tilladte værdier for argumentet vinduestypografi.

Værdi	Vinduestypografi
1, 5, 9	Normal med fokus
2	Minimeret med fokus
3	Maksimeret med fokus
4, 8	Normal uden fokus
6, 7	Minimeret uden fokus

## Udveksling af data med OLE

Der findes en anden teknologi, som kan anvendes når man vil implementere dataudveksling, nemlig OLE (Object Linking and Embedding). OLE er uden tvivl den teknologi, der vil blive set anvendt fremover, og den er derfor vigtig at studere nærmere. Ovenstående eksempel vil i en OLE implementation se ud som følger:

```
Function UdskrivIWordOle ()
    Dim Msw As Object
    Set Msw = CreateObject("word.basic")
    Msw.FilerÅbn "F:\P66\WINDOWS\K_BREVE.DOC"
    Msw.Brevfletningtildok
    Msw.FilerGemSom " F:\P66\WINDOWS\NYEBREVE.DOC"
End Function
```

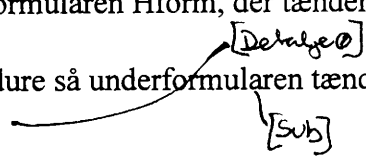
Den afgørende forskel mellem de to teknologier er, at OLE direkte "snakker" med Words objekter, og ikke behøver at vide, hvor Word-programmet er placeret.





## Øvelse 6.1

*Formål: at oprette Access Basic subprocedurer, der indstiller egenskaber.*

1. Opret en knap på formularen Hform, der tænder og slukker for feltet Behandlingstid.
2. Opret en subprocedure så underformularen tændes og slukkes ved dobbeltklik på selve overformularen. 
3. Opret en ur, på et passende sted på formularen - f.eks. på titellinjen eller i en tekstboks.

Ekstra:

3. Opret en tekstboks, hvori der skrives hjælpetekster, når musen bevæger sig ind over forskellige felter på formularen.



## Øvelse 6.2

*Formål: at oprette Access Basic subs.*

1. Opret et nyt modul i databasen APPL.MDB.
2. Opret en sub TitelLinie, som tilføjer brugernavn og tidspunkt til titelteksen i formularen Hform, hver gang formularen aktiveres.

Ekstra:

3. Opret en automatisk udlogging på formularen hform, således at Access afsluttes, hvis maskinen ikke har været rørt i 30 minutter.



## Øvelse 6.3

Formål: at oprette Access Basic funktioner.

1. Opret et nyt modul i databasen APPL.MDB.
2. Opret en funktion BeregnMoms, som kan returnere momsen af et givent beløb.
3. Gem modulet under navnet Egne Funktioner.
4. Test den nye funktion ved at kalde den fra en forespørgsel. - *Ordre\_fragt\_moms*
5. Test funktionen ved at udføre momsregningen på formularen Hform v.h.a. den nye funktion. *L jekket 'Moms' 'Moms af fragt'*

Ekstra 1:

1. Opret en funktion Rabat, som returnerer rabatten af et beløb, således at der gives 10 %, hvis man køber for over 1000 kroner og 20 %, hvis man køber for over 4000 kroner.
2. Test funktionen på passende måde. *Vis, Brugerdefinerede*
3. Eksperimentér med forskellige datatyper for funktionens returnværdi.

Ekstra 2

1. Skriv en funktion ArbejdsDag(), der tager en dato som argument, og undersøger om denne dag er en arbejdsdag (mandag til fredag) eller er en dag i weekenden. Hvis dagen er en arbejdsdag returneres værdien ARBEJDSDAG ellers returneres værdien WEEKEND. Funktionen kan f.eks. implementeres v.h.a. den indbyggede funktion Weekday().

*Functi Rabat (Beløb as Currency) As double  
if beløb > 1000 Then  
Rabat = beløb \* .2*

*? Rabat (1000.3333)  
100.0333*



## Øvelse 6.4

*Formål: at arbejde med Access Basic i formularer.*

1. Opret en forespørgsel baseret på tabellen ordrer. Forespørgelsen skal summere fragtomkostningerne grupperet på kunde-id. Opret endvidere en simpel rapport, der viser data fra denne forespørgsel.
2. Opret en ubundet formular til styring af udskrivningen af den nye rapport. Eksperimentér med at sende rapporten til forskellige uddatamedier. Brug eksemplet i materialet som inspiration.
3. Opret en passende fejlhåndtering, således at brugeren får en fornuftig besked, hvis noget går galt.
4. Eksperimentér med at indsætte stoppunkter i koden - og med at trace koden fra et stoppunkt.



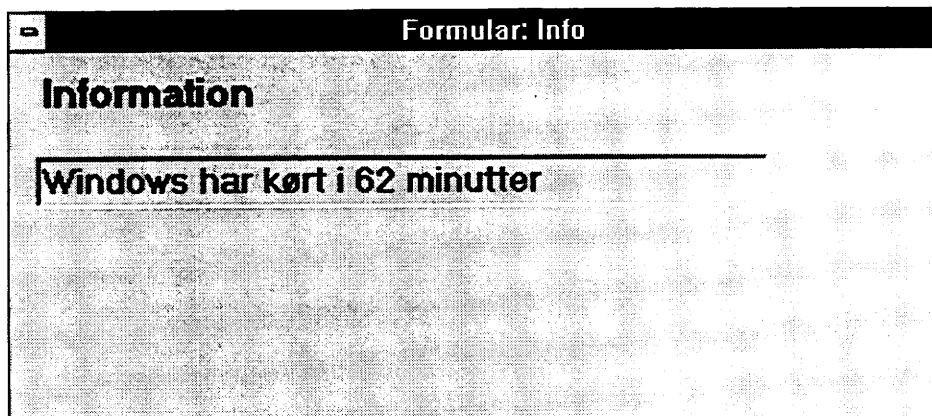
## Øvelse 6.5

*Formål: at arbejde Windows API'et*

1. API-funktionen `GetTickCount()` returnerer antallet af millisekunder, der er gået siden Windows blev startet. Ifølge API-referencen kaldes funktionen på følgende måde:

```
Declare Function GetTickCount Lib "User" () As Long
```

2. Opret erklæringen af API-funktionen i et nyt modul, og skriv derefter en funktion, der returnerer antallet af millisekunder, der er gået siden Windows blev startet.
3. Udbyg funktionen således at den formaterer returværdien til en mere læsebar enhed. f.eks. minutter.





## Øvelse 6.6

*Formål: at anvende Access Basic til udveksling af data*

1. Åbn databasen APPL.MDB.
2. Tilret makroen FletTilWord, så handlingen OverførTekst eksporterer tabellen Kunder til F:\<PC-NUMMER>\WINDOWS\KUNDER.DOC.
3. Tilret funktionen UdskrivIWord i modulet Fletning, så sætningen  
DDEExecute Kanal, "[SkStandardbibl ""C:\", 0]"  
skifter til biblioteket F:\<PC-NUMMER>\WINDOWS\.
4. Tilknyt makroen FletTilWord til en trykknop i værktøjslinien.
5. Test flettefunktionen ved at køre makroen. Start herefter MS-Word, og åbn dokumentet F:\<PC-NUMMER>\WINDOWS\NYEBREVE.DOC, for at undersøge resultatet. **Husk at lukke dokumentet inden makroen eventuelt aktiveres igen.**
6. Åbn MS-Word dokumentet F:\<PC-NUMMER>\WINDOWS\K\_BREVE.DOC og tilret det. Husk at gemme og lukke dokumentet efter tilretningen.
7. Test flettefunktionen på ny.
8. I funktionen UdskrivIWord findes kommandoen

```
Dummy = Shell("c:\msoffice\winword\winword", 2)
```

Ret argumentet vinduestypografi fra 2 til 3, og prøv at køre funktionen igen. MS-Word skal være lukket, for at ændringen er synlig.

9. Implementér en OLE-baseret version af overstående funktion - kig i materialet for gode ideer.

*Gem ikke 'makroer' i biblioteker*

## 7. Biblioteksdatabase

Når du i en database opretter formularer, moduler mm., er disse kun tilgængelige i den database, de blev oprettet i. En biblioteks database er en samling af procedurer og databaseobjekter, der kan kaldes fra enhver database. Biblioteksdata-baser kan også bruges til at gemme værktøjslinier, der skal bruges i flere databaser.

### Værktøjslinier i biblioteksdatabase

Kræver følgende punkter:

1. Opret en ny database
2. Opret de ønskede værktøjslinier
3. Opret funktioner til at vise/skjule værktøjslinierne
4. Skjul alle værktøjslinier i basen
5. Omdøb databasefilen til <databasenavn>.mda
6. Tilføj biblioteksbasen under overskriften [Libraries] i filen MSACC20.INI
7. Genstart Access
8. Opret makroer i lokalbasen, der kalder vis/skjul-funktioner fra biblioteksbasen

### Vis/skjul-funktioner

I biblioteksdatabase skal oprettes et modul, der indeholder funktioner til at vise/skjule de lagrede værktøjslinier. Koden til de to funktioner vises nedenfor.

```
Function tænd (v1)
DoCmd ShowToolbar v1, A_TOOLBAR_YES
End Function
```

```
Function sluk (v1)
DoCmd ShowToolbar v1, A_TOOLBAR_NO
End Function
```

Bemærk at funktionen som argument tager navnet på værktøjslinien. Dermed behøves kun én funktion til hhv. at vise en værktøjslinie og skjule en værktøjslinie.



## Indlæsning af biblioteksdatabase

Der kan indlæses én eller flere biblioteksdata-baser, når Access startes op. Skriv en tilføjelse for hvert bibliotek i sektionen [Libraries] i filen c:\windows\msacc20.ini. Sektionen [Libraries] indeholder alle indlæste biblioteksdata-baser.

Tilføjelsen skrives efter følgende syntaks: *filnavn = [ro/rw]*

Hvis biblioteksdatabase-n ligger Access-biblioteket, er det tilstrækkeligt at angive filnavnet. Hvis ikke angives filnavnet med fuld sti.

### Eksempel

```
[Libraries]
d:\access.app\bib.mda = ro
```

## Kald af bibliotekets værktøjslinier

Efter opstart med den nye biblioteksdatabase er de oprettede værktøjslinier i princippet tilgængelige. De kan dog kun vises ved at kalde en funktion i biblioteksdatabase-n. Det kan gøres ved hjælp af en makro i den lokale database. Det vil i praksis sige at de 'styres' på samme måde som lokale værktøjslinier.

### Eksempel

For at vise værktøjslinien 'bruger1' fra biblioteksdatabase-n, kan følgende makro køres:

Handling		Kommentar
▶	KørKode	
Handlingsargumenter		
Funktionsnavn	tænd("bruger1")	

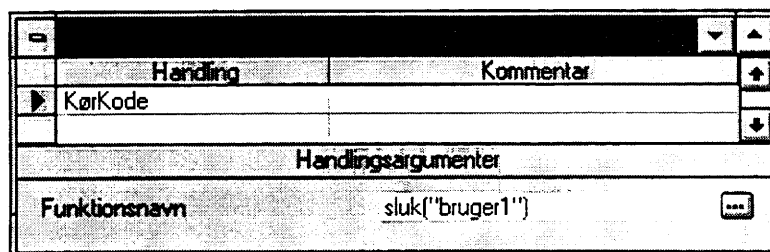
Når funktionen er udført, kan brugerne se værktøjslinien ved at vælge 'Værktøjslinier...' i menuen vis. Brugeren kan dermed selv vise/skjule værktøjslinien.





## Skjul bibliotekets værktøjslinier

Hvis en given værktøjslinie kun skal være tilgængelig i udvalgte formularer, skal det også være muligt at skjule dem igen. For at skjule værktøjslinien 'bruger1' fra biblioteksdatabase, kan følgende makro køres:



Når funktionen er udført, kan brugerne ikke længere se værktøjslinien ved at vælge 'Værktøjslinier...' i menuen vis. Brugeren kan dermed ikke selv vise/skjule værktøjslinien.

I .INI set

[options]

Debuglibraries = True

⇒ i Access: Vis, procedurer, Databaser: utility  
 Her kan ses på alle MS funktioner ex BuilderZoom()  
 .. weekday()  
 ↓  
 velg .MDA.lib





## Øvelse 7.1

Formål: at oprette en biblioteksdatabase indeholdende værktøjslinier

1. Opret en biblioteksdatabase med mindst én brugerdefineret værktøjslinie
2. Opret funktioner i biblioteksdatabase til at vise/skjule værktøjslinier
3. Indlæs biblioteksdatabase
4. Kald en eller flere værktøjslinier fra biblioteksdatabase frem i den lokale database





## 8. Dokumentation

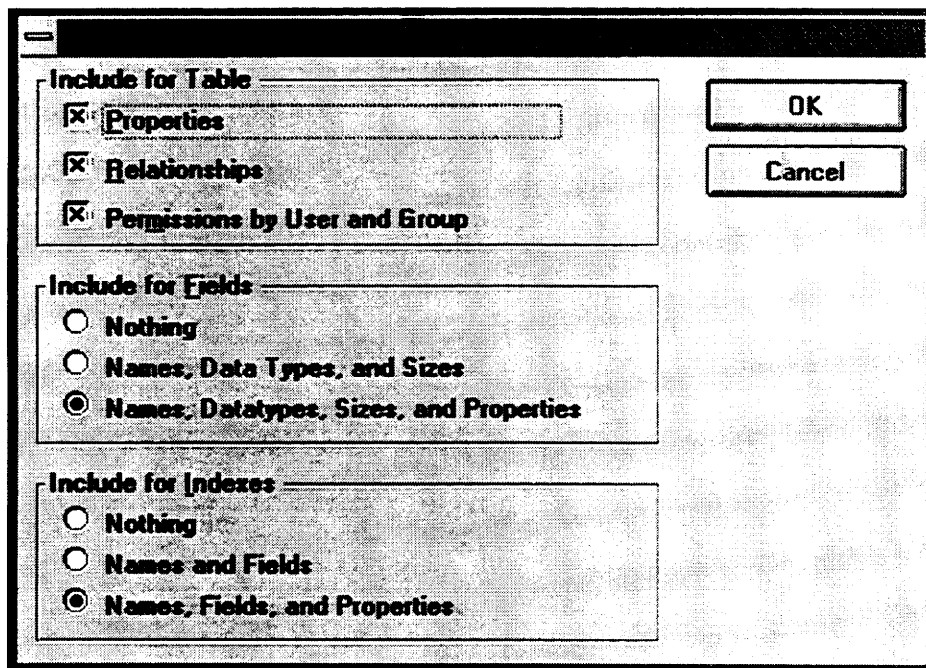
En af de mere tidskrævende opgaver i forbindelse med udvikling, er at skrive den nødvendige dokumentation. Den del af dokumentationen, der drejer sig om at beskrive de enkelte objekter, kan i Access udskrives af programmet.

### Udskriv definition

Dokumentation for et enkelt objekt udskrives således:

- Marker et objekt i databasevinduet
- Vælg 'Udskriv definition...' i menuen Filer

Er der eksempelvis valgt en tabel, vil følgende indstillinger være mulige:



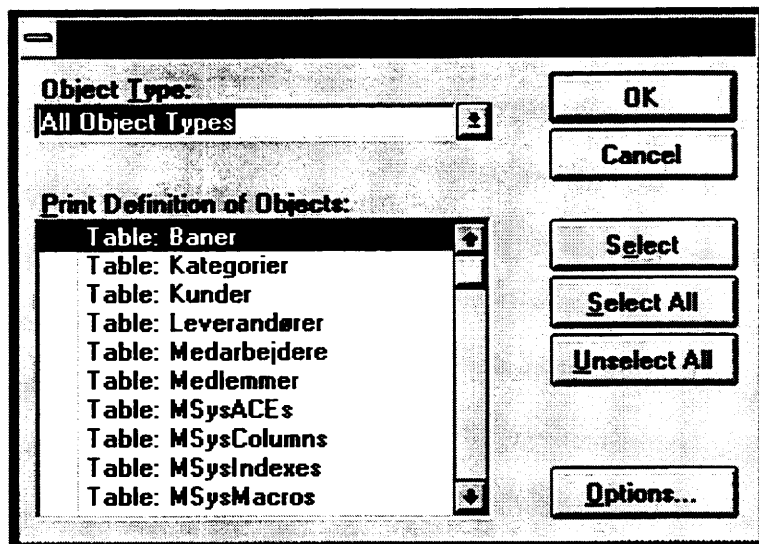
- Vælg OK for at se dokumentationen i Vis udskrift
- Vælg 'Udskriv...' i menuen filer for at udskrive til printer



## Databasestrukturvisning

Dokumentation for flere objekter eller for alle objekter i databasen inkl. databasen udskrives således:

- Vælg Tilføjesprogrammer i menuen Filer fra databasevinduet
- Vælg Databasestrukturvisning i undermenuen



- Vælg objekttype, objekter og Options
- Vælg OK for at se dokumentationen i Vis udskrift
- Vælg 'Udskriv...' i menuen filer for at udskrive til printer

En fuldstændig dokumentation af hele basen kan nemt tage en time og fylde flere hundrede sider.



## Øvelse 8.1

Formål: at udskrive dokumentation på en tabel

1. Udskriv en fuldstændig dokumentation af tabellen ordrer
2. Udskriv dokumentation om selve basen - ikke dens objekter
3. Udskriv dokumentation om en makro







## 9. Optimering og tuning

### Hvad er optimering ?

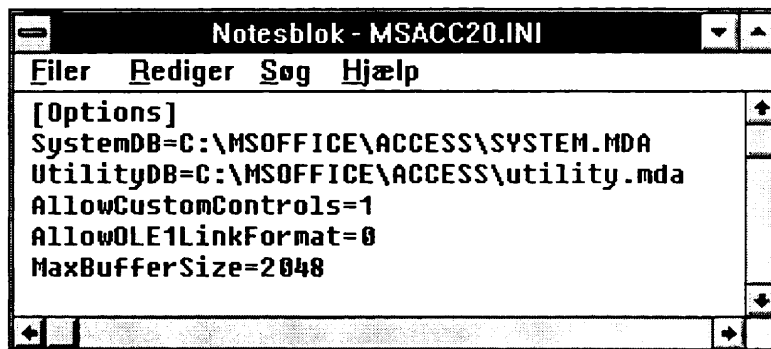
Det handler ganske enkelt om at få Access til at køre så hurtigt som muligt. Dette mål kan dels nås ved at opbygge forespørgsler, formularer o.s.v. på optimal vis, og dels ved at optimere en række parametre i MS-Access initialiseringsfilen MSACC20.INI.

### Tuning af MSACC20.INI

Der findes i INI-filen en række parametre som kan få Access til at køre hurtigere. De vigtigste gennemgås i det følgende.

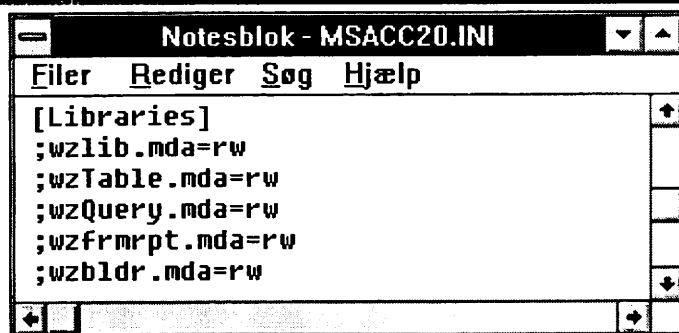
#### MaxBufferSize

Parameteren MaxBufferSize refererer til mængden af arbejdslager, som Access reserverer til intern brug. Standardværdien er 512 kb og der kan afsættes op til 4096 kb. Hvis man har mindst 8 mb arbejdslager, kan det være en god idé at eksperimentere med denne indstilling. MaxBufferSize skrives i ini-filens options sektion. Nedenstående figur viser en konfiguration med 2 mb (2048 kb).



#### Fjern guider

Når et system er færdigt er der ingen grund til at indlæse guider. Disse er kun relevante i udviklingsmiljøet. Derfor bør man fjerne guiderne i den færdige applikation. Dette gør dels opstarten af Access hurtigere og frigiver op til 315 kb arbejdslager. Guiderne kan dels fjernes fra et menu-punkt i Access, men ofte vælger man at redigere direkte i den aktuelle ini-fil. Guiderne er nemlig registreret i ini-filen under sektionen Libraries. Man kan enten slette linjerne eller blot placere et semikolon i starten af linjen.



## Andre metoder til optimering

### Komprimér databasen

Komprimér så ofte som muligt. Database-filen vokser løbende, og bliver mindre effektiv efterhånden som databaseobjekter slettes. I praksis falder filens størrelse nemlig ikke - hvilket betyder at Access kører 30-50 % langsommere i visse forespørgsler, i forhold til en komprimeret database. Hvis basen er meget stor - ellers harddisken hvorpå basen er placeret er meget lille, bør defragmenteringsværktøjer anvendes ofte. Man kan evt. tilbyde sine (super)brugere en ikon, som gør det muligt at defragmentere uden brug af menuerne i Access.

### Access Basic

Helt generelt giver brugen af Access Basic i applikationer mere fart end f.eks. makroer. Endvidere vil API-kald altid køre langt hurtigere end noget som helst andet, man selv kunne tænkes at implementere. Endelig bør variable altid erklæres med passende datatyper - idet dette vil mindske lagerforbruger.

## ISAM-JET parametre

Nedenstående tabel viser ISAM-JET parametre, som relateres til Access databasemotorens indstillinger. Justering af disse bør foretages med forsigtighed og altid med en passende sikkerhedskopi af MSACC20.INI. Disse parametre findes i sektionen ISAM.

Indstilling	Beskrivelse	Eksempel
PageTimeout	Angiver tidsrummet mellem placeringen af data, der ikke benyttes, i en intern cache, og når den markeres til at blive opdateret. Indstillingerne udtrykkes i enheder på 100 millisekunder. Standard er 5 (500 millisekunder eller 0,5 sekunder).	PageTimeout=5
LockedPageTimeout	Angiver tidsrummet mellem placeringen af data, der benyttes, i en intern cache, og når de markeres til at blive opdateret. Indstillingerne udtrykkes i enheder på 100 millisekunder. Standard er 5 (500 millisekunder eller 0,5 sekunder).	LockedPageTimeout=5



CursorTimeout	Angiver timeout for markører i intervaller på tiendedelesekunder. Standard er 5 (eller 0,5 sekunder).	CursorTimeout=5
LockRetry	Angiver antallet af gange et forsøg på at få adgang til en låst side skal gentages. Standard er 20.	LockRetry=20
CommitLockRetry	Angiver antallet af gange Microsoft Jet-databasemotoren forsøger at sætte en lås på data for at gemme ændringer i data. Hvis det ikke lykkes at sætte en permanent lås, lykkes opdateringer ikke. Antallet af forsøg, Jet-databasemotoren gør for at sætte en permanent lås, er direkte relateret til LockRetry-indtastningen. For hvert forsøg på at sætte en permanent lås, forsøger den at sætte en lås det antal gange, der er angivet i LockRetry-indstillingen. Hvis CommitLockRetry f.eks. er 20 og LockRetry er 20, forsøger Jet-databasemotoren at sætte en permanent lås indtil 20 gange, og hver gang forsøger den at sætte en lås indtil 20 gange og hermed forsøge låsning 400 gange ialt. Standard er 20.	CommitLockRetry=20
ReadAheadPages	Angiver antallet af sider, der skal læses forud, når sekventielle skanninger foretages. Standard er 16.	ReadAheadPages=16

Åben alle formularer, hvor der er mange rækker, allerede ved opstart af applikation. Men sæt dem til visible.

Veel lukken af synlig formular :

Me-visible = False

Who-Called-From-Sqlquery-visible = True





## Øvelse 9.1

*Formål: at eksperimentere med optimering af Access*

1. Optimér den sikrede applikation, som blev oprettet tidligere. Prøv at indstille MaxBufferSize samt at fjerne guiderne. Undersøg hvordan dette påvirker performance og evt. lagerforbruger.
2. Prøv endvidere at arbejde med ISAM-indstillingerne i INI-filen - kan performanceændring mærkes ?
3. Prøv - på en kopi af en database - at slette store datamængder, f.eks. flere store tabeller, og undersøg hvordan dette påvirker database-filens størrelse.
4. Komprimér herefter basen, og undersøg hvordan dette påvirker dens størrelse.

'Fraseradm' (tabel)

Frasernummer	Tekstdata	Levingsdato
1		01-01-1997
2		01-01-1997
3		29-01-1997
4		29-01-1997

JET : Data Access Objects

Functie BeregnAntalIdag

```

IF i > 0 then
    Docnd OpenForm "MinForm"
else
    Docnd OpenForm "Ere anden form"
  
```

OpenQuery

.INI : (på W95/NT findes ingen .ini-fil, i stedet ligger det i 'registre

[Options]

[ISAM]

[ODBC]

[Run time options]

[Libraries]

; w2lib.mda=rw

;  
;

↑ udkommenter til spare lager

[... Wizards] ← kan slettes

## [dBASE IV]

Filter=dBASE IV (\*.dbf)|\*.dbf|Alle filer (\*.\*)|\*.\*|  
Extension=dbf  
OneTablePerFile=Yes  
IndexDialog=Yes  
IndexFilter=dBASE-indeks (\*.ndx;\*.mdx)|\*.ndx;\*.mdx|Alle filer (  
\*.\*)|\*.\*|  
CreateDbOnExport=No

## [dBase ISAM]

CollatingSequence=International  
Century=Off  
Date=American  
Mark=47  
Deleted=On

## [ODBC]

QueryTimeout=60  
LoginTimeout=20

## [Btrieve]

Filter=Btrieve (file.ddf)|file.ddf|  
Extension=ddf  
OneTablePerFile=No  
IndexDialog=No  
CreateDbOnExport=Yes

## [Btrieve ISAM]

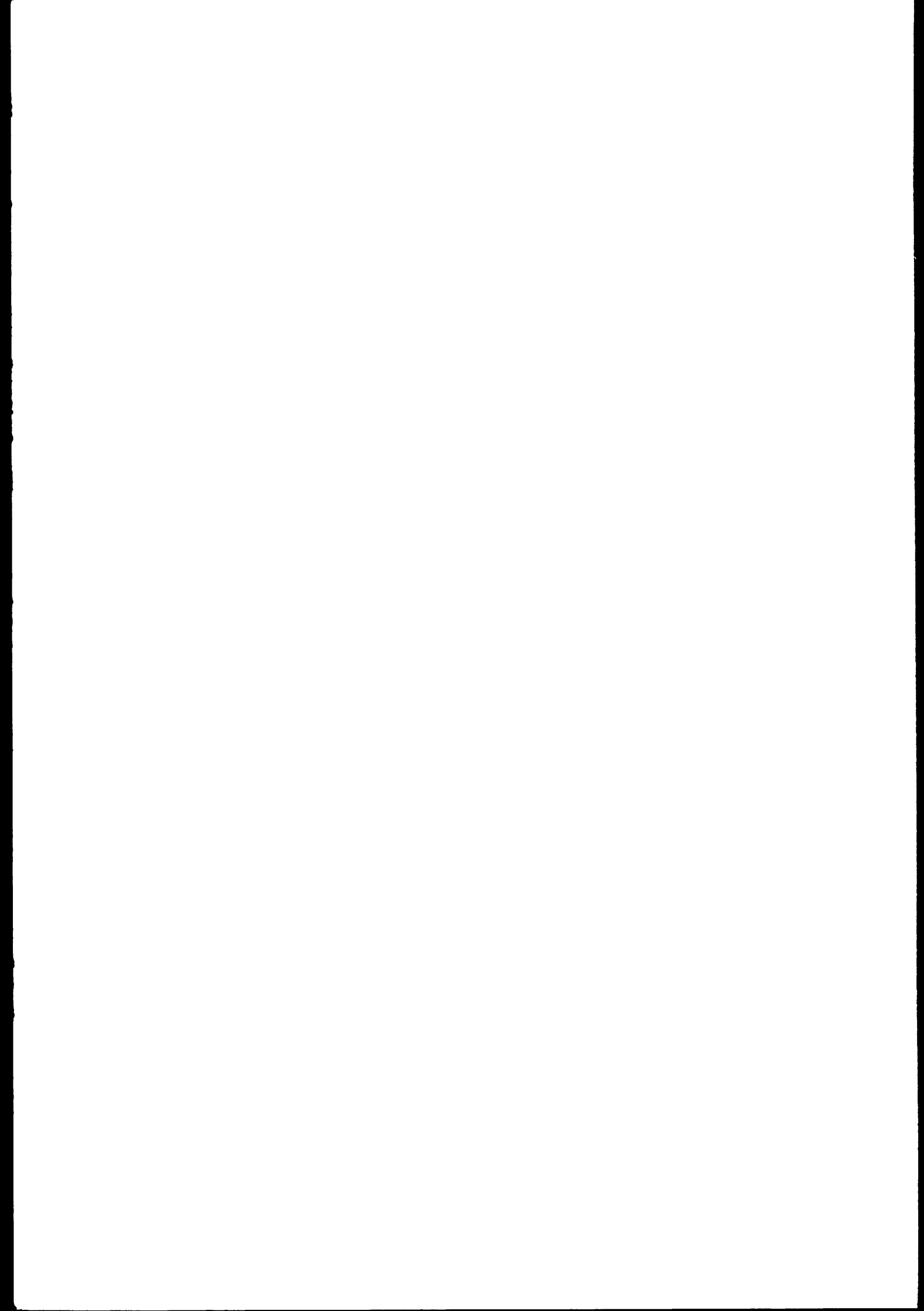
DataCodePage=OEM

## [OLE CONTROL WIZARDS]

OutlineWizardEntry=Outl1016Ctrl.COutlCtrl, Data Outline Control  
Wizard, wz\_OutlStartup,w

## [Run-Time Options]

TitleBar=Det store pladesystem  
StartupScreen=arkade.bmp





Option Compare Database 'Anvend databasens rækkefølge til strengsammenligninger.

Function CheckPrevious (NumberToCheck)

Dim db As Database

Dim ThisTabel As Recordset

Dim Hit

Set db = CurrentDB()

Set ThisTabel = db.OpenRecordset("GP1")

Hit = False

ThisTabel.MoveFirst

Do Until Hit Or ThisTabel.eof

    If ThisTabel![Com from letter] = NumberToCheck Then

        Hit = True

    End If

    ThisTabel.MoveNext

Loop

ThisTabel.Close

If Hit Then

    MsgBox "Posten er gemt..."

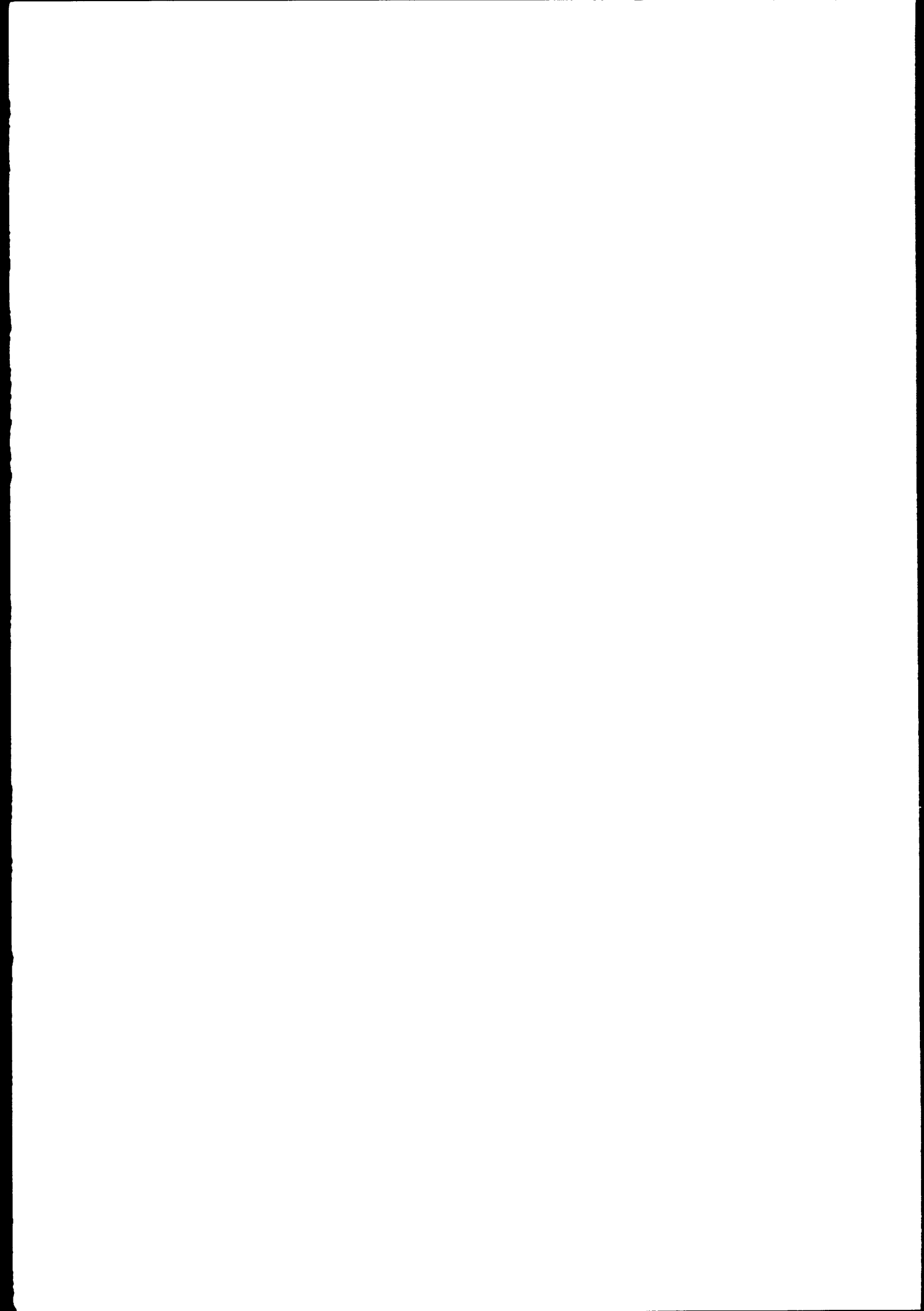
Else

    MsgBox "Nummeret findes ej i forvejen..."

    DoCmd CancelEvent

End If

End Function



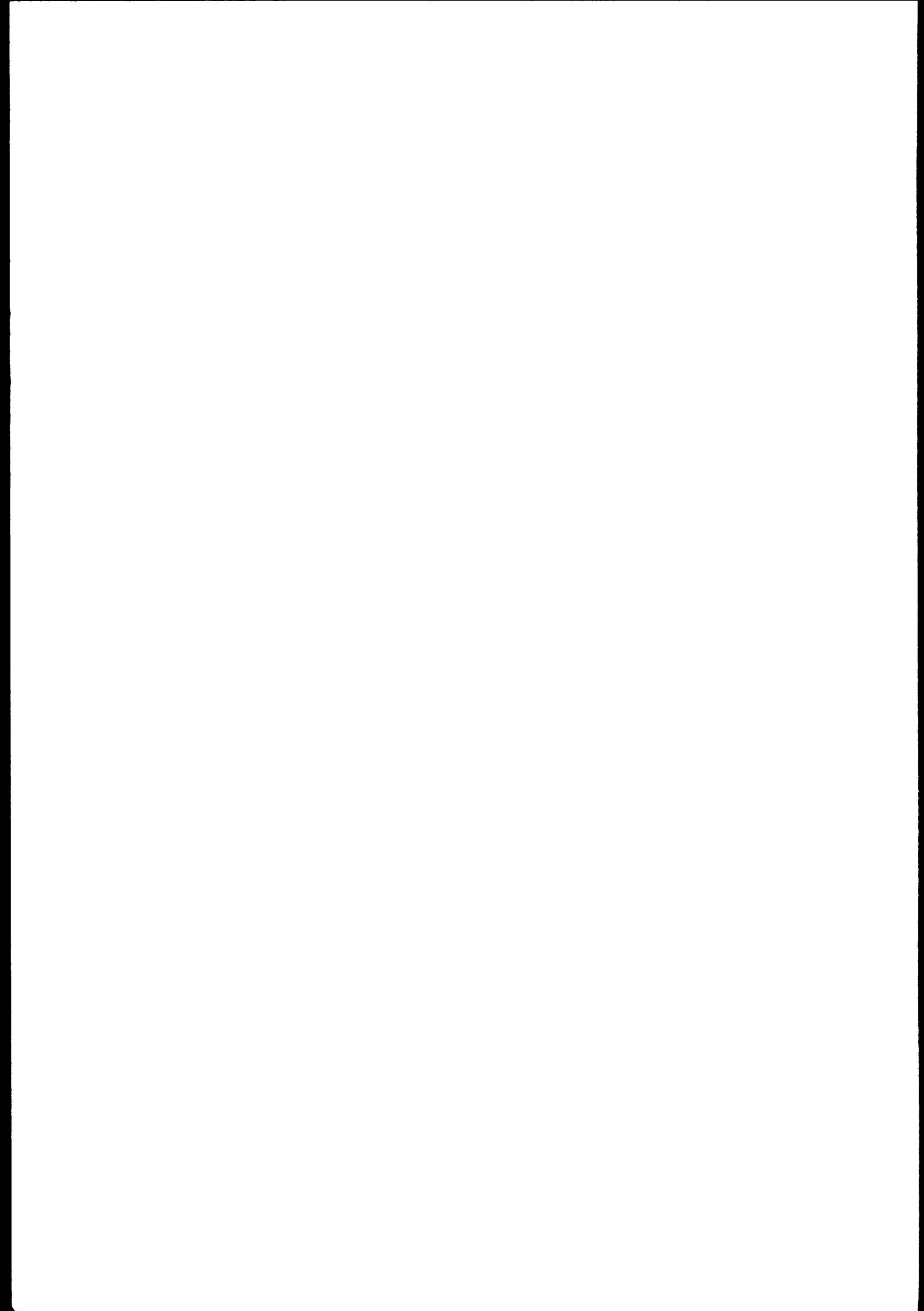
Option Compare Database 'Anvend databasens rækkefølge til strengsammenligninger.

```
Function BeregnAntalIdag ()
  Dim db As Database
  Dim MineData As Recordset
  Dim i As Integer

  Set db = CurrentDB()
  Set MineData = db.OpenRecordset("FraseAdm")
  i = 0

  MineData.MoveFirst
  Do Until MineData.eof
    If MineData![Leveringsdato] = Date Then
      i = i + 1
    End If
    MineData.MoveNext
  Loop
  MineData.Close
  MsgBox "Der var " & i & " rækker fra i dag."

End Function
```



## Funktionstaster

### Se også

Funktionstasterne er vist nedenfor i fire grupper: globale funktionstaster, som bruges overalt i Microsoft Access, funktionstaster til brug i designvisning, funktionstaster til brug i dataark- og formularvisning og endelig funktionstaster, der bruges i modulvinduet.

### Globale funktionstaster

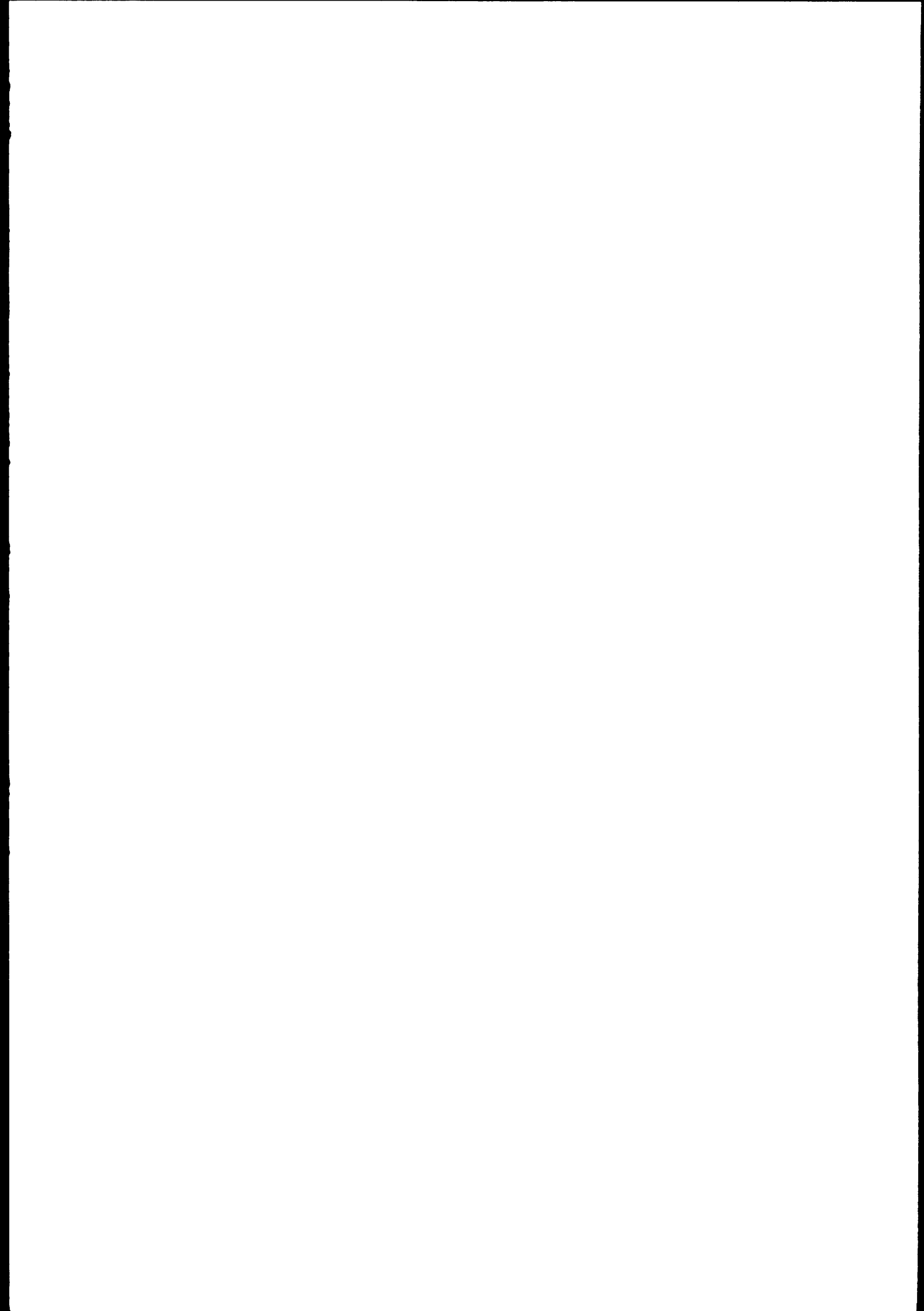
Tryk på	For at
F1	Vise kontekstbetinget Hjælp, der er relateret til den markerede kommando, dialogboks, egenskab, kontrolelement, handling, Access Basic-nøgleord eller vindue.
SKIFT+F1	Vise <u>spørgsmålstegnsmarkøren</u> . Flyt markøren hen til det element, du ønsker hjælp til, såsom værktøjslinjen, og klik for at få kontekstbetinget Hjælp.
CTRL+F4	Lukke det aktive vindue.
ALT+F4	Afslutte Microsoft Access eller lukke en dialogboks.
CTRL+F6	Skifte mellem åbne vinduer.
F11 eller ALT+F1	Anbringe databasevinduet øverst.
F12 eller ALT+F2	Åbne dialogboksen <b>Gem som</b> .
SKIFT+F12	
eller ALT+SKIFT+F2	Gemme et databaseobjekt.
ALT+MELLEMRUM	Vise <u>Kontrolmenu</u> .

### Funktionstaster, som bruges i designvisning

Tryk på	For at
F2	Skifte mellem redigeringstilstand (med indsætningspunktet vist) og flyttetilstand. (tabeller, forespørgsler og makroer i designvisning).
SKIFT+F2	Åbne <u>zoomboksen</u> til en mere hensigtsmæssig indtastning af udtryk og anden tekst i små inputområder.
F6	Skifte mellem vinduets øverste og nederste del (tabeller og forespørgsler i designvisning og filtervinduet).

### Funktionstaster, som bruges i dataark- og formularvisning

Tryk på	For at
F2	Skifte mellem redigeringstilstand (med indsætningspunktet vist) og flyttetilstand.
F4	Åbne en kombinationsboks eller liste.
SKIFT+F4	Søge efter den næste forekomst af den tekst, der er angivet i en dialogboksene <b>Søg</b> eller <b>Erstat</b> , når dialogboksen er lukket.
F5	Flytte til <u>boksen "Postnummer"</u> . Skriv i boksen nummeret på den post, du vil gå til, og tryk på ENTER.
F6	Bladre fremad gennem sidehovedet, detaljesektionen og sidefoden i en formular i formularvisning.
SKIFT+F6	Bladre baglæns gennem sidefoden, detaljesektionen og sidehovedet i en formular i formularvisning.
F7	Åbne dialogboksen <b>Søg</b> .
SKIFT+F7	Åbne dialogboksen <b>Erstat</b> .
F8	Slå udvidtilstanden til. Gentagne tryk på F8 <b>udvider</b> det markerede med ordet, feltet, posten og alle poster.
SKIFT+F8	Vende F8-markeringen om. Tryk på ESC for at annullere udvidtilstanden.
F9	Beregne felterne i vinduet på ny.
SKIFT+F9	<u>Genforespørg</u> i de underliggende tabeller. Når du trykker på

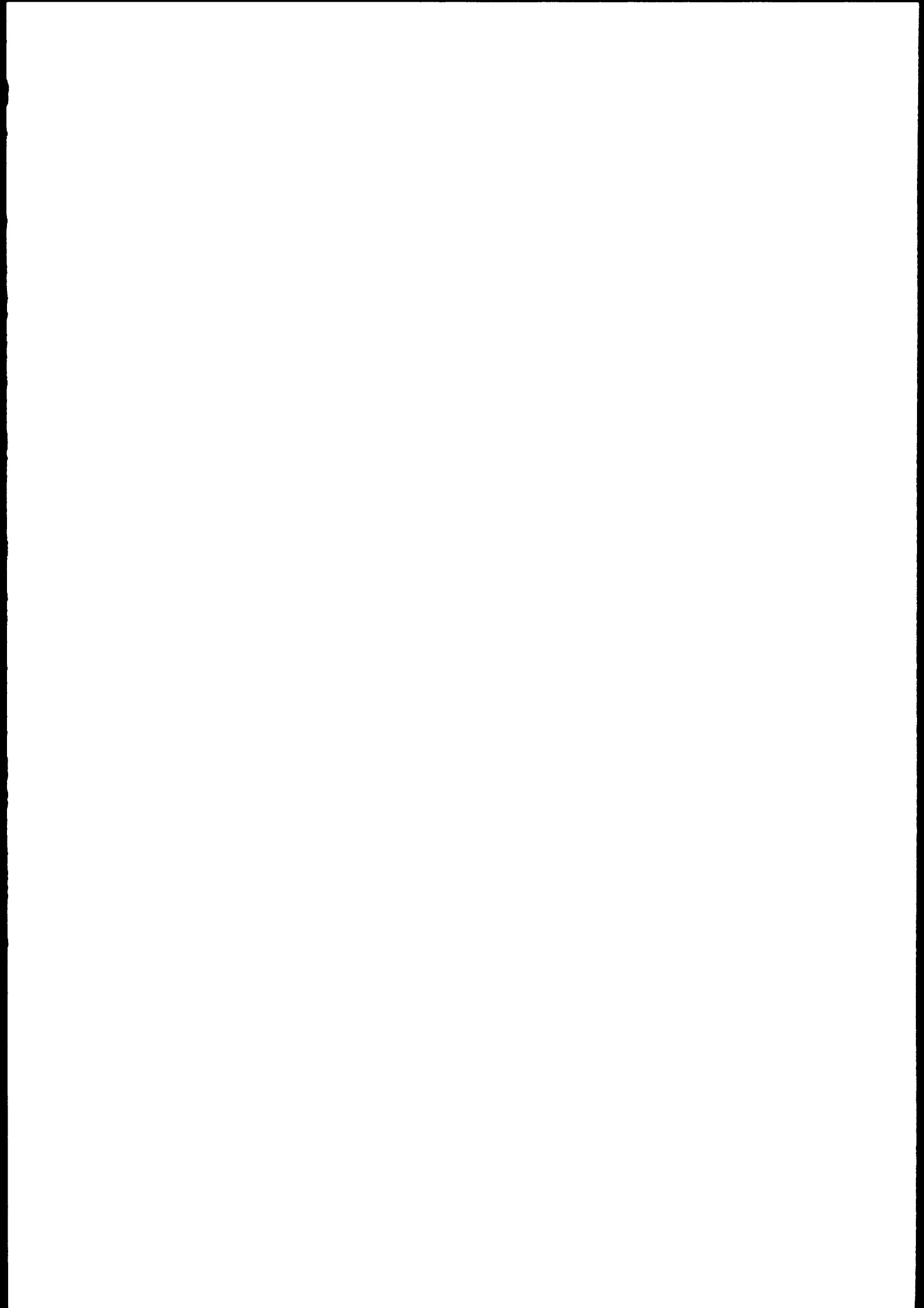


SKIFT+F9

Genforespørg i de underliggende tabeller. Når du trykker på SKIFT+F9 i en underformular, forespørges der kun i den underliggende tabel for denne underformular.

#### Funktionstaster, som bruges i modulvinduet

<b>Tryk på</b>	<b>For at</b>
F2	Vise procedurer.
SKIFT+F2	Gå til den procedure, der er markeret i modulvinduet.
F3	Søge efter den næste forekomst af den tekst, der er angivet i en af dialogboksene <b>Søg</b> eller <b>Erstat</b> .
SKIFT+F3	Søge efter den forrige forekomst af den tekst, der er angivet i en af dialogboksene <b>Søg</b> eller <b>Erstat</b> .
F5	Fortsætte udførelsen.
F6	Skifte mellem den øverste og nederste rude, hvis du har opdelt vinduet.
F7	Åbne dialogboksen <b>Søg</b> .
SKIFT+F7	Åbne dialogboksen <b>Erstat</b> .
F8	Køre trinvist (enkelt trin).
SKIFT+F8	Køre trinvist (proceduretrin).
F9	Slå et <u>pausepunkt</u> til og fra på den markerede linje.



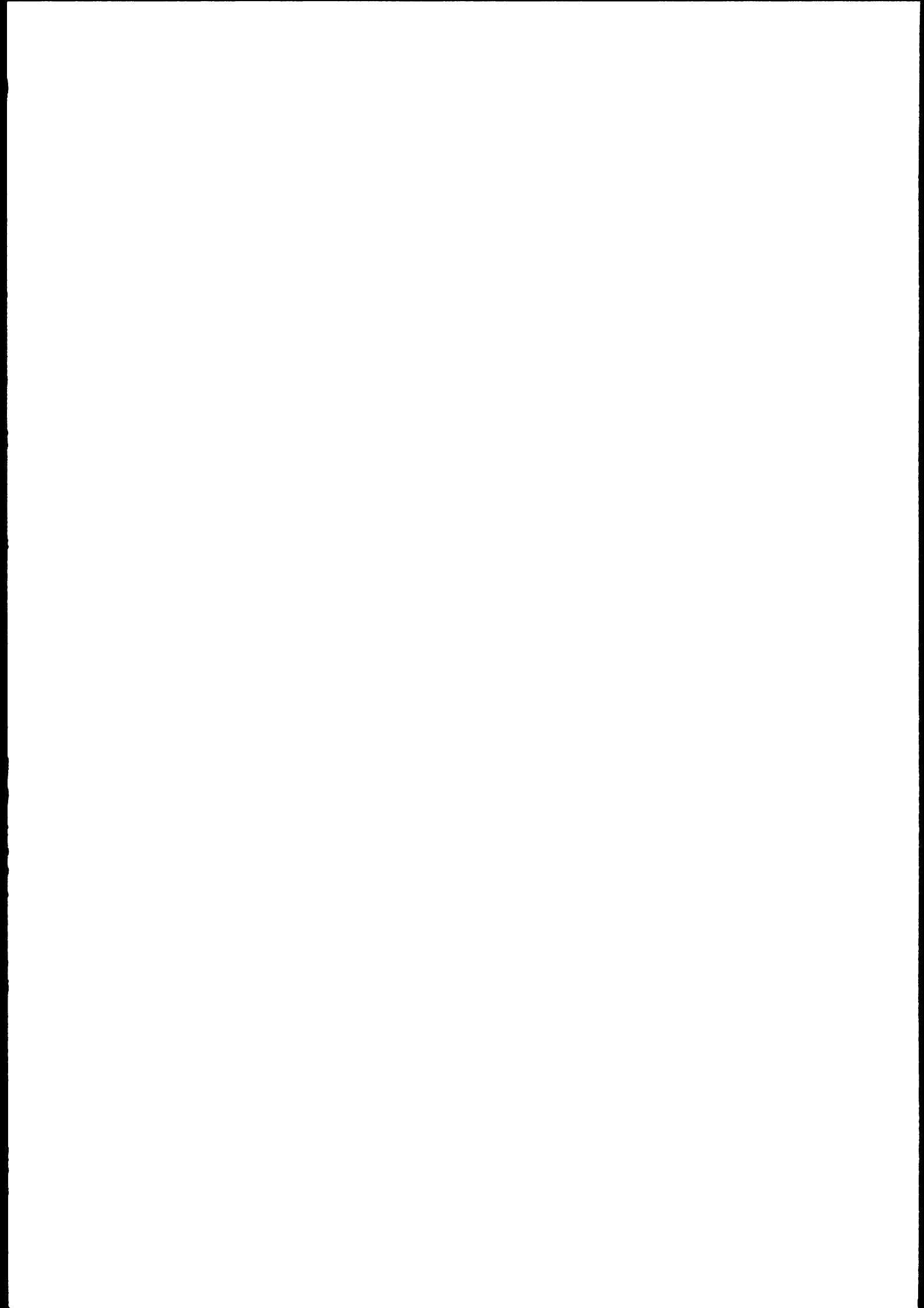


Brugere

<u>Brugernavn</u>	<u>Grupper som brugeren tilhører</u>
admin	Admins, Users
ge	nybrug, Users
geadm	Admins, nyadm, nybrug, Users
guest	Guests

Grupper

<u>Gruppenavn</u>	<u>Grupper som brugeren tilhører</u>
Admins	admin, geadm
Guests	guest
nyadm	geadm
nybrug	ge, geadm
Users	admin, ge, geadm

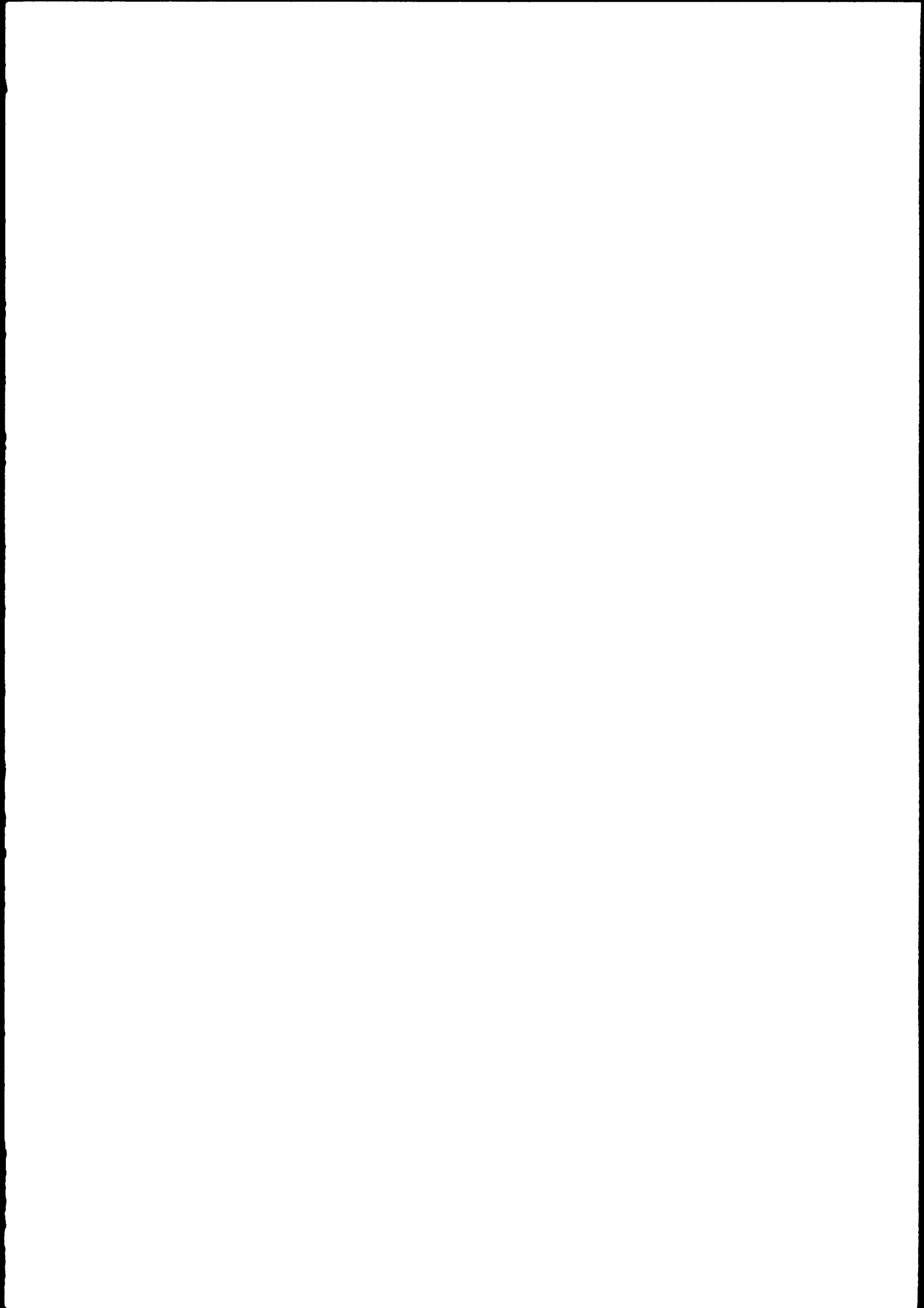


**Egenskaber**

DatoOprettet: 27-01-1997 14.49.26 Ejer: admin  
SenestOpdateret: 27-01-1997 14.52.28

**Handlinger**

Navn	Betingelse	Handling	Argument	Værdi
	CurrentUser(<>"Admin"	AngivVærdi	Element:	[Application].[MenuBar]
			Udtryk:	"_main_menu"

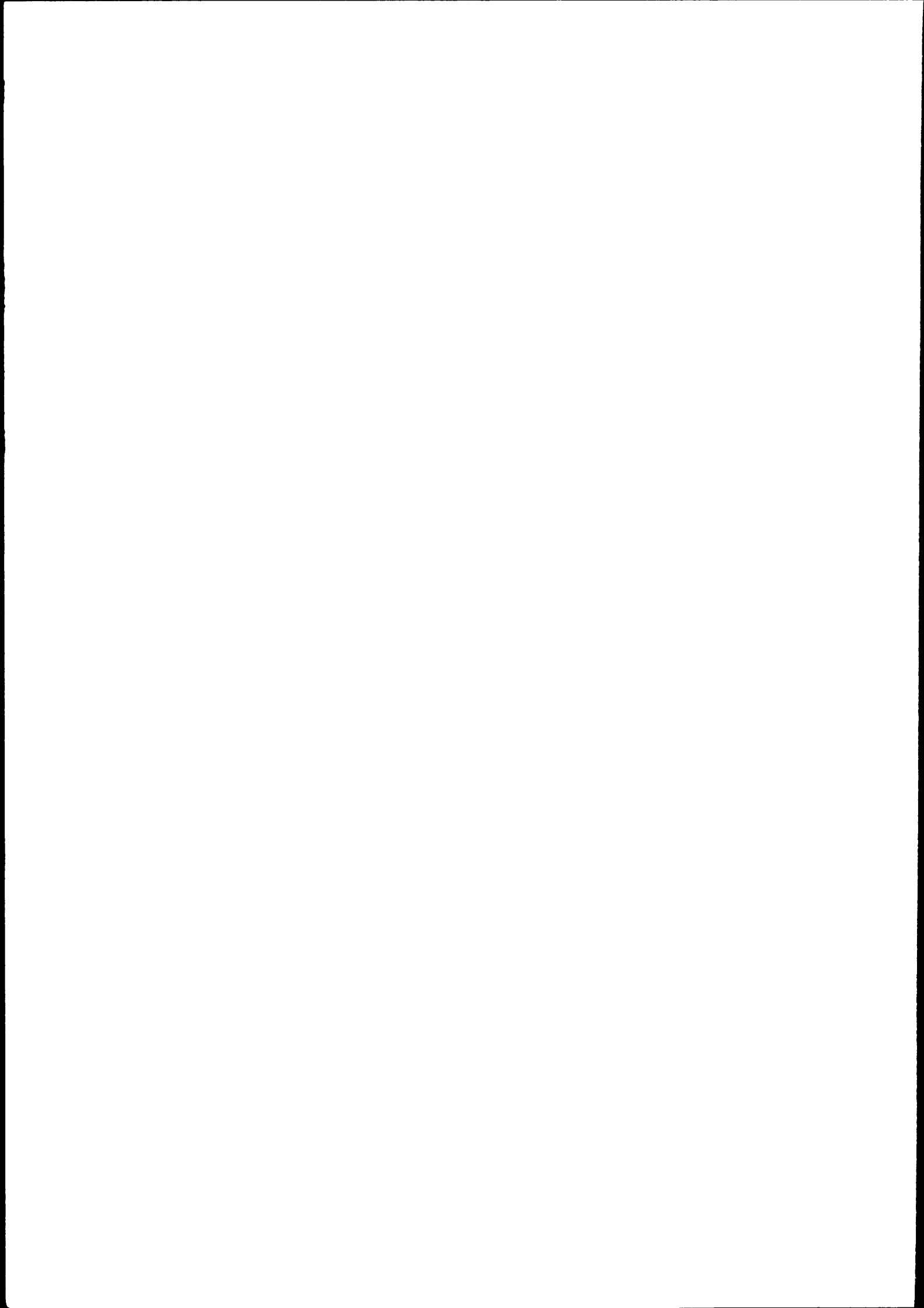


**Egenskaber**

DatoOprettet: 27-01-1997 13.08.52 Ejer: admin  
SenestOpdateret: 27-01-1997 13.43.50

**Handlinger**

Navn	Betingelse	Handling	Argument	Værdi
		GåTilKontrollement	Kontrollementnavn:	MedlemsListe
<i>Hop til listen</i>				
		Genforespørg	Kontrollementnavn:	MedlemsListe
<i>Gentegn listen</i>				
		SendTaster	Tastetryk: Vent:	{F2} 0
<i>Send et F2</i>				
		GåTilKontrollement	Kontrollementnavn:	Kriterium
<i>Hop tilbage</i>				





# DELTAGERBEVIS

Det bekræftes herved at:

**Gitte Engelsholm**

Har deltaget i kurset:

**Access Applikationsudvikling**

I perioden

**27. til 29. januar 1997**

Med venlig hilsen

Dansk Data Elektronik A/S

**Erik Gudmundsson**  
Kursuslærer







Common Object Model

A model for binary component interoperability  
Inside OLE 2 / Kang

MS transaction Serve File-  
2 free commit

Internet Explorer porters til Solaris og HP AIX  
4 (Kashville) 30  
5 (Memphis) Kan redogør HTML

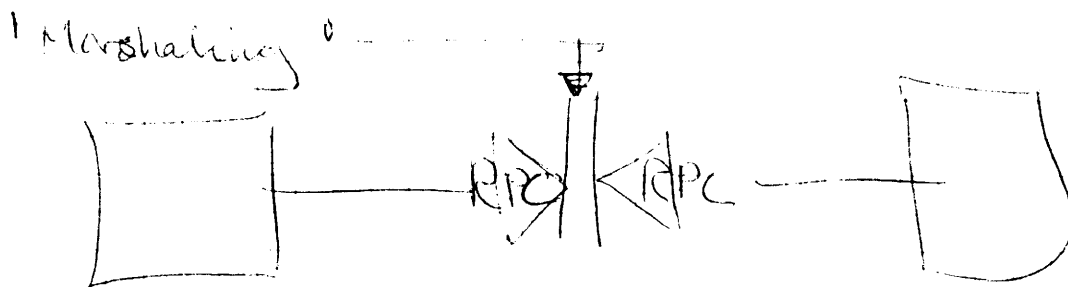
1990 is ASCII = HTML

\* Truly DLL arch

ODBC	MAPI	TAPI
DB	mail	telephony

Remote Procedure Call (DCE)

C++ ves i embeddet HTML-lifsystem



WMF (Windows Meta File Format)



## Spørg EGU

0. Jeg vil gerne have kopi af staveplade appl.

1. Data læser fra Access i ODBC til SQL Plus

2. På formular ~~Indsættelse~~ Indsættelse-behandling der holder <sup>generel</sup> junk som løber alle jeller på ~~for~~ i gennem og hvis type text så InitCap

3. ~~Sæt alle~~ <sup>ganske vist</sup> tabel.kolonner af type text som er optional, til 'Ja'

4. Kan man i DAO åbne en vilkårlig ODBC-database: Nej?  
Hvordan løses så opgave åla 'cursor'/recordset uden anvendelse af DAO

5. Hvordan refereres til brugerid/gruppe (uden DAO)?

MS Access Developer's Toolkit (2000-4000 kr excl. moms)

API-dokumentation: Visual C++ DTK eller <sup>eller</sup> Windows 3.11 SDK

Access 7 har replikering, men er ægte 32 bits og kan derfor ikke afvikles på Windows 3.11

Access 8 har integration med internet applikationer → HTML

MS Information Server ≠ NT ≠ Access 8

Runtime version generering ved ADT:

- opret bib til installation
- kopier ~~mda~~ mda til bibliotek
- Opret egen .ini-fil
- Start Wizard (ADT, Setup Wizard)

EDIT Access 7: SYSTEM.MDA omdøbt til SYSTEM.MDW  
Kortgrup

Lav ikon for DB-komprimering og distribuer med!



Dansk Data Elektronik A/S  
Herlev Hovedgade 199  
DK 2730 Herlev  
Tel. (+ 45) 42 84 50 11  
Fax (+ 45) 42 84 52 20