

Database-baseret Web-publicering, forår 2001

Forelæsning 9, tirsdag den 27. marts 2001.

Teknikker for database-baserede web-sites

- Eksamen
- Form-variable
- Sekvenser
- Anvendelse af references og rækkefølgen af drop table
- Webservere med faste databaseforbindelser og scripting
- Java Applets
- Sikkerhed
- Opsætning og start af AOL-server på hug.it.edu
- Resultat af skemaundersøgelse
- WimpyQuest
- Lotdrækning

Sekvenser

Med en sekvens i Oracle (*sequence*) kan vi generere en række af unikke løbenumre (tal).

```
SQL> create sequence id_seq start with 5;
Sequence created.
```

```
SQL> select id_seq.nextval from dual;
NEXTVAL
-----
5
```

```
SQL> select id_seq.nextval from dual;
```

```
NEXTVAL
-----
6
```

```
SQL> select id_seq.currval from dual;
```

```
CURRVAL
-----
6
```

Tabellen `dual` er en tom tabel som vi f.eks. kan anvende i vores SQL-kommando når syntaksen dikterer, at vi skal angive en tabel uden egentligt at have brug for tabellen.

Form-variable

Post

- overfører skjulte form-variable med `hidden`
 - anvend *aldrig* ? og & direkte i urlen angivet i `action`
- ```
<form method=Post action=fil.tcl>
<input type=hidden name=v value="42">
 ...
</form>
```

#### Get

Man kan overføre form-variable direkte i urlen.

- ? angiver den første form-variabel.
- *form.var=va*lue
- & adskiller form-variable fra hinanden.
- anvend `urlencode` for at kode specialtegn i ens url.

```

```

```
I Tcl er der måske koder: <a href="vis_plade.tcl?plade_id=42&titel=[ns:urlencode
"Hans Og Grethe"]>
```

### Sekvenser fortsat

```
SQL> create table id_tab (id integer primary key);
Table created.
```

```
SQL> insert into id_tab values(id_seq.nextval);
1 row created.
```

```
SQL> insert into id_tab values(id_seq.nextval);
1 row created.
```

```
SQL> insert into id_tab values(id_seq.currval);
ORA-00001: unique constraint (NH.SYS_C0025624) violated
```

```
SQL> select * from id_tab;
ID

7
8
```

```
SQL> drop sequence id_seq;
Sequence dropped.
```

## Anvendelse af references og rækkefølgen af drop tab1e

```
create table tab1 (
 id integer primary key
);
create table tab2 (
 id references tab1
);
insert into tab1 values (1);
insert into tab1 values (2);
insert into tab2 values (1);
insert into tab2 values (2);
```

Nu kan jeg ikke skrive drop table tab1.:

```
SQL> drop table tab1;
drop table tab1
```

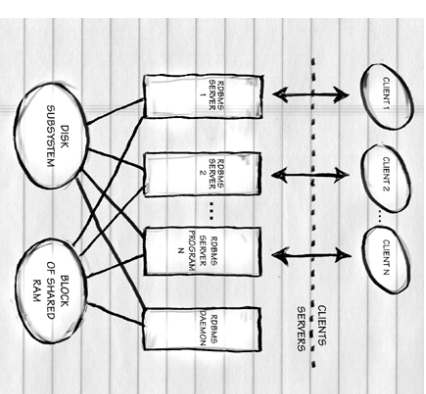
ORA-02449: unique/primary keys in table referenced by foreign keys

Jeg skal droppe tabel tab2 først:

```
SQL> drop table tab2;
Table dropped.
SQL> drop table tab1;
Table dropped.
```

## Generel Klient-Server Databasearkitektur

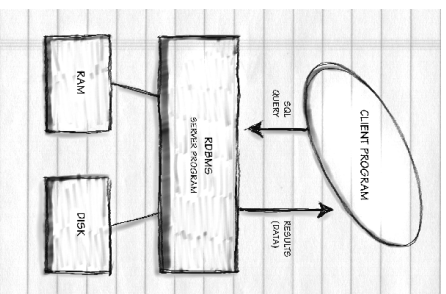
- Klienter kører potentielt på forskellige fysiske maskiner
- Databaseprocesser kører på samme fysiske maskine
- En databaseproces "forker" for hver klientforbindelse



Billede: <http://www.arsdigita.com/books/panda/napkin/12.2.gif>

## Teknikker for databasebaserede web-sites

- CGI-scripts
- Webservere med fæste databaseforbindelser og scripning (f.eks. AOLserver med Tcl)
- Java-applets



Billede: <http://www.arsdigita.com/books/panda/napkin/12.1.gif>

## CGI-scripts

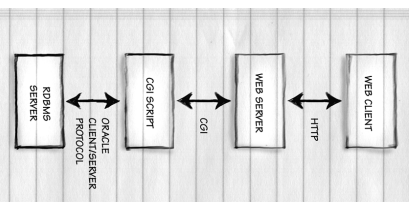
CGI står for Common Gateway Interface

**Fordele:**

- alle betydelige web-servere understøtter CGI-scripts
- programmeringssprogs-uafhængig

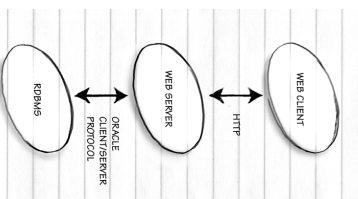
**Ulemper:**

- CGI-programmet startes som en ny operativsystemprocess for hver forespørgsel
- En ny databaseserver-process startes for hver forespørgsel
- CGI-programmet skal autoriseres af den nystartede databaseserver
- Det tager tid at lukke CGI-programmet



Langsom database-forbindelse med CGI.

Billede: `http://www.photo.net/wtr/thebook/mapkin/12.4.gif`



Hurtig database-forbindelse med f.eks. Tcl

og AOLserver der anvender faste database-forbindelser.

Billede: `http://www.photo.net/wtr/thebook/mapkin/12.3.gif`

## Java Applets

### Fordele:

- Hurtig eksekvering på Klienten (browseren), på trods af høj downloadtid
- Potentiale for god usability

### Ulemper:

- Problemer med sikkerhed - ekstra åben port til databasen
- Problemer med licensen - da en port til databaseadgang skal være åben skal der typisk betales licens for antallet af samtidige brugere
- Potentielt mange databaseforbindelser og mange kørende databaseservere - hvornår kan man antage at en databaseforbindelse ikke bliver anvendt?
- Understøttes kun af få browsere - dette bliver bedre med tiden

## Webservere med faste databaseforbindelser og scripting

### Fordele:

- Hurtig opstart af fortløbet program
- Hurtig tilgang til eksisterende databaseforbindelser

### Ulemper:

- Stærkt knyttet til en bestemt web-server - tab af portabilitet

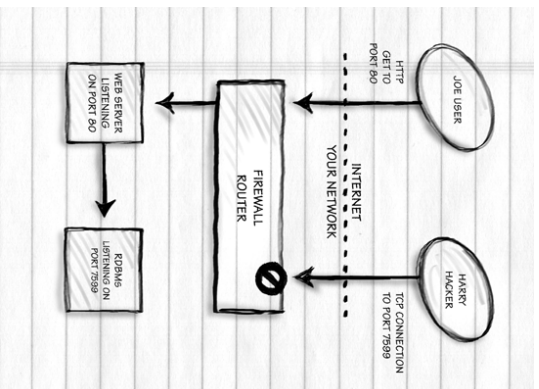
## Sikkerhed

### For at en hacker kan angribe data kræves:

1. Forbindelse til databaseservens IP-adresse
2. Kodeord til databaseserver eller
1. Forbindelse til serverens IP-adresse
2. Kodeord til root, ved at hacke maskinen eller ...

### For at undgå angreb:

- Flyt databaseserver bag firewall og tillad kun forbindelse for webserver
- Flyt webserver bag firewall og tillad kun adgang udefra via port 80



Billede: <http://www.arsdigita.com/books/panda/napkin/12.5.gif>

**Opsætning og start af AOLserver på hug.it.edu**  
**Opstart af AOLserver fra UNIX-prompt:**

```
cd /usr/local/aolserver/
bin/nsd-oracle -t nsd.mael.tcl
```

**Konfigurationsfilen /usr/local/aolserver/nsd.mael.tcl:**

Jeg har klippet en del i filen!

```
#-----
mael's server
#-----
set home /usr/local/aolserver
set host [ns_info hostname]

ns_section "ns/parameters"
ns_param Home /usr/local/aolserver
ns_param serverlog /web/mael/log/server.log
ns_param pidfile /web/mael/log/nspid.mael
ns_param user mael
```

**Konfigurationsfilen /usr/local/aolserver/nsd.mael.tcl, fortsat**

```
ns_section "ns/server/mael"
ns_param directoryfile index.html
ns_param pageroot /web/mael/www
ns_param enabletcpipages on

ns_section "ns/server/mael/tcl"
ns_param library /web/mael/tcl

ns_section "ns/server/mael/module/nslog"
ns_param file /web/mael/log/access.log

ns_section "ns/server/mael/module/nssock"
ns_param port 8002
ns_param hostname $host
```

**Konfigurationsfilen /usr/local/aolserver/nsd.mael.tcl, fortsat**

Her følger konfiguration af Oracle! AOLserver.

```
ns_section "ns/db/drivers"
ns_param ora8 ora8.so

ns_section "ns/db/pool/main"
ns_param Driver ora8
ns_param Connections 2
ns_param User mael
ns_param Password #####
ns_param Verbose On

ns_section "ns/db/pools"
ns_param main main

ns_section "ns/server/mael/db"
ns_param Pools *
```

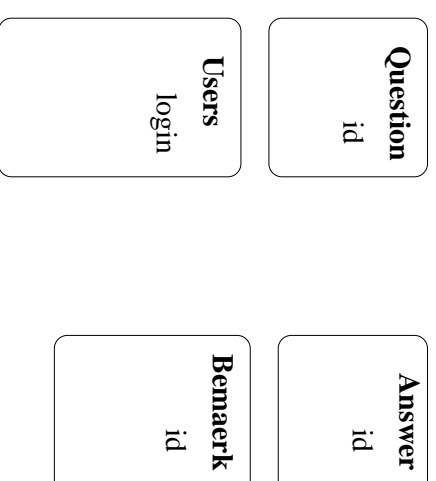
## Resultat af skemaundersøgelse

Resultatet kan ses på URL: `http://hug.itu.dk:8077/WimpyQuest/stat.tcl`

### Bemærkninger

- Jeg skal blive bedre til at anvende tavlen.
  - Stor bredde ved spg: "Støftet præsenteres for hurtig"
  - Bredden ses også ved spg: "Øvelsesopgaverne er svære"
  - Der skal være mere tid ved øvelserne.
  - Godt, at spørgelisterne anvendes.
  - Plancherne skal på til tiden!!!
  - Stor bredde ved spg: "Kursat tager for meget tid"
  - Godt, at man mener at kurset er relevant for ens uddannelse.
  - Ønsker om andet sprog end Tcl.
  - Opgavelister er krydsfiske
- ### Opfølgning
- Vi har talt om eksamen idag.
  - Jeg vil i højere grad hjælpe til ved øvelserne.
  - Studnævnet har allerede set resultatet

## Datamodel for WimpyQuest



## WimpyQuest

URL: `http://hug.itu.dk:8077/WimpyQuest/answer_form.tcl`

### Krav til Datamodel

- Spørgsmål skal komme fra en tabel
  - Kun registrerede brugere skal kunne afgive svar
  - Svar skal gemmes anonymt. Hvorledes forhindrer vi at en bruger afgiver svar to gange.
  - Der skal være mulighed for at skrive en enkelt bemærkning.
- Hvilke tabeller har vi brug for?
- Hvilke søkvenser har vi brug for?
- Hvilke felter skal der være i tabellene?
- Hvilke tabeller refererer til andre tabeller?

## Uddrag af wimpyquest.sql

```
drop sequence question_seq;
drop sequence bemaerk_seq;
drop table bemaerk;
drop table answer;
drop table question;
drop table users;

create sequence question_seq start with 1;
create table question (
 id integer primary key,
 question varchar(4000) not null
);

create table users (
 login varchar(100) primary key,
 name varchar(100) not null,
 password varchar(100) not null,
 response varchar(1) check(response in ('t','f'))
);

create table answer (
 id integer references question,
 answer integer not null
);
```

#### Uddrag af wimpyquest.t.sql

```
create sequence bemaerk_seq start with 1;
create table bemaerk (
 id integer primary key,
 bemaerk varchar(4000)
);

insert into question (id, question)
values (question_seq.nextval, 'Forlæseren forklarer stoffet godt');
...
commit;
```

WimpyQuest består af følgende filer:

- wimpyquest.sql
- answer\_form.tcl
- add\_answer.tcl
- stat.tcl

#### Opbygning af add\_answer.tcl

Form-variabel: login, password, submit og svar formateret som svar ID, hvor ID er spørgsmålets nummer

Filen add\_answer.tcl gør følgende i den nævnte rækkefølge:

1. Checker at login er tilstede; hvis ikke returner fejl.
2. Henter password fra databasen til det givne login
3. Hvis submit er lig "Email Mig Mit Password", så sendes password med email
4. Checker at password er korrekt
5. Checker at der ikke allerede er afgivet svar på det givne login.
6. Henter spørgsmål fra databasen
7. Løber gennem alle spørgsmål og samler svar op i en liste
8. Når alle svar er tilstede, så indsættes de i databasen; ellers returneres fejl
9. Hvis der er en bemærkning indsættes denne i databasen
10. Tabel users opdateres med at login har afgivet svar.

#### Lodrækning (vins.tcl)

**Opgave:** Lav et program vins.tcl, som finder den person der skal have vinen.

**Krav:** Personen skal have afgivet svar.

Opbygning af vins.tcl:

1. Find antal personer som har afgivet et svar, *n*
2. Hent personer fra databasen sorteret efter navn
3. Find et tilfældigt *r* tal mellem 0 og *n*-1.
4. Lav en løkke, som returnerer den *r*'te person fra listen af navne.

```
set query "select name from users where response = 't' order by name"
set selection [ns_db select $db $query]
set r [randomRange $n]
set ____ [____ getrow $db $selection] {
 while { [____ == $r] {
 if { $____ == $r } {
 set variables_after_query
 break;
 }
 incr ____
 }
}
```

#### Introduktion til Øvelse 8

#### Konstruktion af et web-baseret pladekartotek