



Høgskolen i Telemark

EKSAMEN

5603 DATABASER II

11.05.2009

Tid:	9-13
Målform:	Bokmål/nynorsk
Sidetall:	8 med forside
Hjelpemidler:	Ingen
Merknader:	Ingen
Vedlegg:	Syntaks og klassebibliotek

Eksamensresultatene blir offentliggjort på nettet, via Arena høgskole. I tillegg finner du eksamensresultatslister på utsiden av eksamenskontoret, men da trenger du kandidatnummeret ditt, så du bør notere dette på en lapp og legge den i lommeboken.



Avdeling for allmennvitenskapelige fag



Prosentatsen ut for hver oppgave antyder vekt ved sensur.
Prøv å svare på så mange spørsmål som mulig. Lykke til!

Oppgave 1 (50%)

Følgende Oracle-database brukes av en bedrift som leier ut hytter på ukebasis:

- Hytte(hyttenr, beskrivelse, sengeplasser, ukepris)
- Reservasjon(resnr, tlfnr, hyttenr*, fra_uke, til_uke)

Databasen inneholder informasjon om hytter og reservasjoner. Kolonnen tlfnr knytter en reservasjon til en kunde. Vi tenker oss at reservasjoner gjelder ett bestemt år. Primærnøkler er understreket og fremmednøkler er merket med en stjerne. Kolonnen Reservasjon.hyttenr er fremmednøkkel mot Hytte.hyttenr. Kolonnene har følgende datatyper:

- INTEGER: Hytte.hyttenr, sengeplasser, resnr, Reservasjon.hyttenr, fra_uke, til_uke
- NUMBER(8,2): ukepris
- VARCHAR(n): beskrivelse, tlfnr

Du skal løse følgende programmeringsoppgaver:

- Skriv en PL/SQL lagret prosedyre for å registrere en ny hytte. Beskrivelsen, antall sengeplasser og ukeprisen skal være parametre til prosedyren. Løpenummeret hyttenr skal genereres automatisk. Hvis antall sengeplasser ikke er mellom 1 og 20 skal prosedyren gi en passende feilkode (et tall) via en ut-parameter. Vis også hvordan prosedyren kan testes (kalles) fra SQL*Plus.
- Skriv en PL/SQL trigger som forhindrer innsetting av en reservasjon som overlapper i tid med en eksisterende reservasjon for den samme hytta.
- Skriv en PL/SQL lagret prosedyre som bruker dbms_output.put_line() for å skrive ut hyttenr, antall sengeplasser og ukepris for samtlige hytter, og under hver hytte skriver ut resnr, fra-uke og til-uke for alle reservasjoner som gjelder denne hytta.
- Skriv en PL/SQL pakkespesifikasjon for prosedyrene du laget i oppgavene 1a og 1c.
- Skriv et C# konsollprogram som bruker klassebiblioteket ADO.NET for å søke etter hytter. Programmet skal lese inn minimum antall sengeplasser og maksimal ukepris fra bruker, og deretter skrive ut hyttenr og beskrivelse for alle hytter som tilfredsstiller søkekriteriene. Gjør nødvendige forutsetninger.

Oppgave 2 (10%)

Tabell T1 inneholder 40.000 rader fordelt på 1.000 blokker, og tabell T2 inneholder 120.000 rader fordelt på 3.000 blokker. Forklar virkemåten til algoritmen *Merge Join*, og anslå antall diskaksesser for å koble (join) T1 og T2 med denne algoritmen, når vi har tilgang på 10 blokker i minnet (RAM). Ingen av tabellene er sortert med hensyn på koblingskolonnen.



Oppgave 3 (20%)

Svar kort på følgende deloppgaver, typisk 3-5 setninger på hver.

- a. Hva inneholder System Global Area (SGA) i en Oracle-instans?
- b. Hva er forskjellen på en lagret funksjon og en lagret prosedyre?
- c. Hvorfor kan ikke :OLD og :NEW brukes i alle slags triggere?
- d. Hva er en markør (cursor)?
- e. Hva er forskjellen på isolasjonsnivåene READ COMMITTED og SERIALIZABLE?
- f. Hva brukes klassen DataSet til i ADO.NET "disconnected"?
- g. Hva er fordelene med LINQ i forhold til "tradisjonell databaseprogrammering"?
- h. Hva gjør kommandoen SHUTDOWN NORMAL?
- i. Hva slags data inneholder loggfilene (redo log files) i en Oracle-database?
- j. Hvordan håndteres mediefeil (diskkrasj) ved gjenoppbygging (recovery)?

Oppgave 4 (20%)

Gi utfyllende svar på følgende deloppgaver, 1-2 sider på hver deloppgave.

- a. Forklar hva ORM (Object-Relational Mapping) er. Bruk gjerne konkrete eksempler.
- b. Vis hvordan datasettet i oppgave 1 kan representeres ved hjelp av XML, og beskrives med henholdsvis DTD og XML Schema. Forklar også kort forskjellen på begrepene velformet og gyldig.



Prosentatsen ut for kvar oppgåve antyder vekt ved sensur.
Freist å svare på så mange spørsmål som mogleg. Lykke til!

Oppgåve 1 (50%)

Følgjande Oracle-database nyttast av ein bedrift som leiger ut hytter på vekebasis:

- Hytte(hyttenr, omtale, sengeplassar, vekepris)
- Reservasjon(resnr, tlfnr, hyttenr*, fraa_veke, til_veke)

Databasen inneheld informasjon om hytter og reservasjonar. Kolonna tlfnr knyter ein reservasjon til ein kunde. Vi tenker oss at reservasjonar gjeld eitt bestemt år. Primærnøkklar er understreka og framandnøkklar er merkte med ei stjerne. Kolonna Reservasjon.hyttenr er framandnøkkel mot Hytte.hyttenr. Kolonnene har følgjande datatyper:

- INTEGER: Hytte.hyttenr, sengeplassar, resnr, Reservasjon.hyttenr, fraa_veke, til_veke
- NUMBER(8,2): vekepris
- VARCHAR(n): omtale, tlfnr

Du skal løyse følgjande programmeringsoppgåver:

- Skriv ei PL/SQL lagra prosedyre for å registrere ei ny hytte. Omtala, talet på sengeplassar og vekeprisen skal være parametrar til prosedyra. Løpenummeret hyttenr skal bli generert automatisk. Viss talet på sengeplassar ikkje er mellom 1 og 20 skal prosedyra gje ein passende feilkode (eit tal) via ein ut-parameter. Syn óg korleis ein kan teste (kalle) prosedyra frå SQL*Plus.
- Skriv ein PL/SQL trigger som stansar innsetting av ein reservasjon som overlappar i tid med ein eksisterande reservasjon for den same hytta.
- Skriv ein PL/SQL lagra prosedyre som nyttar dbms_output.put_line() for å skrive ut hyttenr, talet på sengeplassar og vekepris for samtlege hytter, og under kvar hytte skriver ut resnr, frå-veke og til-veke for alle reservasjonar som gjeld denne hytta.
- Skriv ein PL/SQL pakkespesifikasjon for prosedyrene du laga i oppgåvene 1a og 1c.
- Skriv eit C# konsollprogram som nyttar klassebiblioteket ADO.NET for å søke etter hytter. Programmet skal lese inn minimum tal på sengeplassar og maksimal vekepris frå brukar, og deretter skrive ut hyttenr og omtale for alle hytter som tilfredsstillir søkekriteria. Gjer naudsynte føresetnader.

Oppgåve 2 (10%)

Tabell T1 inneheld 40.000 rader fordelt på 1.000 blokker, og tabell T2 inneheld 120.000 rader fordelt på 3.000 blokker. Forklar virkemåten til algoritma *Merge Join*, og anslå talet på diskaksessar for å koble (join) T1 og T2 med denne algoritma, når vi har tilgang på 10 blokker i minnet (RAM). Ingen av tabellene er sortert med omsyn på koblingskolonna.



Oppgave 3 (20%)

Svar kort på følgjande deloppgåver, typisk 3-5 setninger på kvar.

- k. Kva inneheld System Global Area (SGA) i ein Oracle-instans?
- l. Kva er forskjellen på ein lagra funksjon og ei lagra prosedyre?
- m. Kvifor kan ikkje :OLD og :NEW nyttast i alle slags triggere?
- n. Kva er ein markør (cursor)?
- o. Kva er forskjellen på isolasjonsnivåene READ COMMITTED og SERIALIZABLE?
- p. Kva blir klassa DataSet nytta til i ADO.NET "disconnected"?
- q. Kva for føremoner har LINQ i forhold til "tradisjonell databaseprogrammering"?
- r. Kva gjer kommandoen SHUTDOWN NORMAL?
- s. Kva slags data inneheld loggfilene (redo log files) i ein Oracle-database?
- t. Korleis blir mediefeil (diskkrasj) handtera ved gjenoppbygging (recovery)?

Oppgave 4 (20%)

Gje utfyllande svar på følgjande deloppgåver, 1-2 sider på kvar deloppgåve.

- c. Forklar kva ORM (Object-Relational Mapping) er. Nytt gjerne konkrete døme.
- d. Syn korleis ein kan representere datasettet i oppgave 1 ved hjelp av XML, og definere det med henholdsvis DTD og XML Schema. Forklar óg kort forskjellen på omgrepa velforma og gyldig.

Vedlegg: Syntaks og klassebibliotek

PL/SQL

[] angir valgfrie elementer, | angir "eller", og ... antyder elementer som kan gjentas.
"Grammatikken" er uformell.

Databaseobjekter

```
CREATE [OR REPLACE] PROCEDURE prosedyrenavn  
  (argument [IN|OUT|IN OUT] datatype , ...)  
IS | AS  
blokk
```

```
CREATE [OR REPLACE] FUNCTION funksjonsnavn  
  (argument [IN] datatype , ...) RETURN datatype  
IS | AS  
blokk
```

```
CREATE [OR REPLACE] PACKAGE pakkenavn  
IS | AS  
  <spesifikasjon av funksjoner og prosedyrer>  
END pakkenavn;
```

```
CREATE [OR REPLACE] PACKAGE BODY pakkenavn  
IS | AS  
  <definisjon av funksjoner og prosedyrer>  
END pakkenavn;
```

```
CREATE [OR REPLACE] TRIGGER triggernavn  
  [BEFORE | AFTER] hendelse [OR hendelse [OR hendelse]]  
ON tabellnavn  
  [FOR EACH ROW]  
blokk;
```

Hendelser

```
INSERT | UPDATE [OF kolonnenavn] | DELETE
```

Blokker

```
[DECLARE  
  deklarasjon ; ...]  
BEGIN  
  setning ; ...  
[EXCEPTION  
  setning ; ...]  
END;
```

Unntak

```
RAISE unntak;
```

```
EXCEPTION  
  [WHEN unntak [OR unntak...] THEN setning; ...]  
  [WHEN OTHERS THEN setning; ...]
```

Noen forhåndsdefinerte Oracle-unntak

NO_DATA_FOUND | TOO_MANY_ROWS | INVALID_CURSOR
| ZERO_DIVIDE | DUP_VAL_ON_INDEX

Markører

CURSOR markoernavn [(parameternavn datatype , ...)]
IS utvalg_spoerring ;

OPEN markoernavn [(uttrykk , ...)] ;

FETCH markoernavn INTO variabelnavn;

CLOSE markoernavn;

Markør-attributter

%ROWCOUNT | %FOUND | %NOTFOUND | %ISOPEN

Valg og løkker

IF betingelse THEN
 setning ; ...
[ELSIF betingelse THEN
 setning ; ...]
[ELSE
 setning ; ...]
END IF ;

LOOP
 [setning ; ...]
 EXIT [WHEN betingelse] ;
 [setning ; ...]
END LOOP ;

FOR teller IN [REVERSE] nedre_grense .. oevre_grense LOOP
 setning ; ...
END LOOP ;

WHILE betingelse LOOP
 setning ; ...
END LOOP ;

Variabler og datatyper

Deklarasjon

variabelnavn datatype ;

Noen datatyper

CHAR(n) | VARCHAR2(n) | INTEGER | NUMBER(p,s)
| DATE | tabellnavn.kolonnenavn%TYPE | tabellnavn%ROWTYPE

Select i PL/SQL

SELECT uttrykk , ... INTO variabelnavn , ...
FROM tabellnavn
WHERE betingelse;

ADO.NET

Eksempel på connection-string

```
string connStr =  
    "DATA SOURCE=db;USER ID=scott;PASSWORD=tiger";
```

Utvalgte klasser, egenskaper og metoder fra ADO.NET "connected"

- OracleConnection
 - OracleConnection(string connStr)
 - OracleTransaction BeginTransaction(IsolationLevel l)
 - void Close()
 - void Open()
- OracleCommand
 - OracleCommand(string sql, OracleConnection conn)
 - int ExecuteNonQuery()
 - OracleDataReader ExecuteReader()
 - object ExecuteScalar()
- OracleDataReader
 - int RecordsAffected { get; }
 - void Close()
 - bool Read()
- OracleParameter
 - ParameterDirection Direction { get; set; }
 - bool IsNullable { get; }
 - string ParameterName { get; set; }
 - byte Precision { get; set; }
 - byte Scale { get; set; }
 - int Size { get; set; }
 - string SourceColumn { get; set; }
 - object Value { get; set; }
- OracleTransaction
 - OracleConnection Connection { get; }
 - IsolationLevel IsolationLevel { get; }
 - void Commit()
 - void Rollback()

Utvalgte klasser fra ADO.NET "disconnected"

- Constraint, DataColumn, DataRelation, DataRow, DataSet, DataTable, DataTableReader, DataView, OracleDataAdapter

Spørreprosessering

"Worst case" formler for ulike algoritmer, der $\lceil x \rceil$ er definert som det minste heltallet $n \geq x$, og der n_r og b_r angir henholdsvis antall poster (rader) og antall blokker i relasjonen (tabellen) r .

- Sekvensielt søk: b_r
- Binærsøk: $\lceil \log_2(b_r) \rceil$
- Sort-merge: $b_r(2 \times \lceil \log_{M-1}(b_r/M) \rceil + 1)$, der M er antall blokker tilgjengelig i RAM
- Nested-Loop Join: $n_r \times b_s + b_r$
- Block Nested-Loop Join: $b_r \times b_s + b_r$
- Merge Join: $b_r + b_s$, forutsatt at r og s er sorterte