

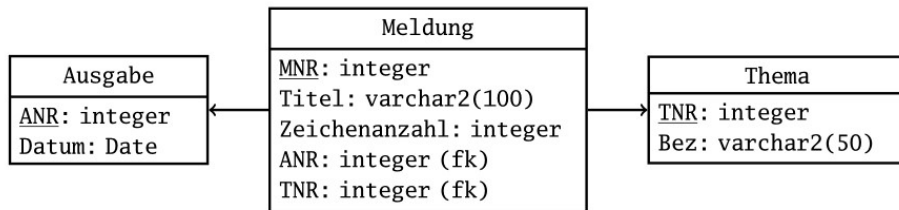
Aufbau Klausur

Donnerstag, 31. Dezember 2020 10:49

1	Gespeicherte Prozedur mit PL/SQL implementieren	20 Punkte
2	DAO mittels JDBC implementieren	20 Punkte
3	Indizes	10 Punkte
4	Transaktionen	10 Punkte

Gespeicherte Prozedur - Lösung

Dienstag, 7. Januar 2020 12:06



In dieser Aufgabe sollen Sie die gespeicherte Prozedur `erstelle_neue_ausgabe` entwickeln, die eine neue Zeitungsausgabe in der Tabelle `Ausgabe` anlegen soll. Beachten Sie dabei folgendes:

- Gehen Sie davon aus, dass die Primärschlüssel automatisch erzeugt werden und daher beim Eintragen der neuen Ausgabe kein Schlüssel mit angegeben werden darf.
- Ihre Prozedur muss überprüfen, dass die neue Ausgabe aktueller als alle vorangehenden ist. Ansonsten muss die Ausnahme `datumsfehler` ausgelöst werden.
- Die Zeitung erscheint sonntags nicht. Sollte das Datum ein Sonntag sein, so muss ebenfalls die Ausnahme `datumsfehler` ausgelöst werden. Für einen Sonntag gilt in der Datenbank: `to_char(datum, 'DY') = 'SO'`.

```
create or replace erstelle_neue_ausgabe(p_datum date) as  
datumsfehler exception;
```

```
begin
```

```
end;
```

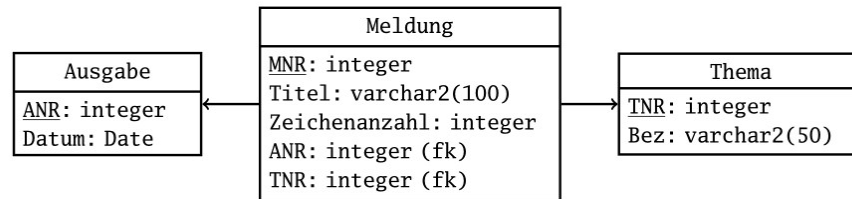
```
create or replace procedure erstelle_neue_ausgabe(p_datum date) as  
datumsfehler exception;  
v_max_datum date;  
begin  
if to_char(p_datum, 'DY') = 'SO' then  
raise datumsfehler;  
end if;  
Select max(datum) into v_max_datum from ausgabe;
```

```
If p_datum <= v_max_datum then  
raise datumsfehler;  
end if;
```

```
Insert into ausgabe(datum) values(p_datum);  
end
```

JDBC / DAO - Lösung

Dienstag, 14. Januar 2020 12:56



In dieser Aufgabe sollen Sie eine Klasse `ZeitungService` entwickeln, die die Schnittstelle `IZeitungService` implementiert.

```
public interface IZeitungService {
    /**
     * Erstellt eine neue Zeitungsmeldung.
     * @param titel der Titel der Meldung
     * @param anzZeichen die Anzahl der Zeichen des Meldungstextes
     * @param tnr der Fremdschlüssel auf das Thema
     * @throws ServiceException falls die Anzahl der Zeichen
     * für das Thema überschritten wird
     * @throws DataStoreException bei allen Datenbankfehlern
     */
    void erstelleMeldung(String titel, int anzZeichen, int tnr);

    /**
     * Versorgt den Service mit einer Datenbankverbindung.
     */
    void setConnection(Connection connection);
}
```

Beachten Sie folgende Punkte bei der Methode `erstelleMeldung`:

- Sie erzeugt eine Meldung für die Ausgabe des folgenden Tages, wobei Sie davon ausgehen können, dass der Eintrag für die Ausgabe bereits in der Tabelle `Ausgabe` vorhanden ist. Das aktuelle Datum kann in der Datenbank mit der Funktion `sysdate` ermittelt werden.
- Prüfen Sie, ob die Obergrenze für Zeichen eines Themas von 20.000 Zeichen eingehalten wird, wofür die Zeichenanzahlen der Meldungen eines Themas addiert werden. Liegt die Summe über dem Limit, trägt die Prozedur die Meldung nicht in die Tabelle ein, sondern löst eine `ServiceException` aus.
- Tabelle `Meldung` beinhaltet einen automatisch erzeugten Primärschlüssel.

```
void erstelleMeldung(String titel, int anzZeichen, int tnr) {

    String sql1 = "select anr from ausgabe where datum = sysdate +1"
    int anr;
    try (PreparedStatement ps1 = c.prepareStatement(sql1) {
        try (ResultSet rs1 = ps1.executeQuery()) {
            rs1.next() // Ausgabe ist ja da
            anr = rs1.getInt("anr");
        }
    } catch (SQLException e) {
        throw new DataStoreException(e);
    }
}
```

```
String sql2 = "select sum(zeichenanzahl) as anz from meldung where anr=? And tnr=?";
int zeichenanzahl;
try (PreparedStatement ps2 = c.prepareStatement(sql2) {
    ps2.setInt(1, anr);
    ps2.setInt(2, tnr);
    try (ResultSet rs2 = ps2.executeQuery()) {
        rs2.next();
        zeichenanzahl = rs2.getInt("anz");
        if (zeichenanzahl + anzZeichen > 20000) {
            throw new ServiceException();
        }
    }
} catch (SQLException e) {
    throw new DataStoreException(e);
}
```

```
String sql3 = "insert into Meldung(Titel, Zeichenanzahl, ANR, TNR) values(?,?,?,?);
try (PreparedStatement ps3 = c.prepareStatement(sql3) {
    ps3.setString(1, titel);
    ps3.setInt(2, anzZeichen);
    ps3.setInt(3, anr);
    ps3.setInt(4, tnr);
    ps3.executeUpdate();
} catch (SQLException e) {
    throw new DataStoreException(e);
}
}
```

in die Tabelle ein, sondern löst eine **ServiceException** aus.

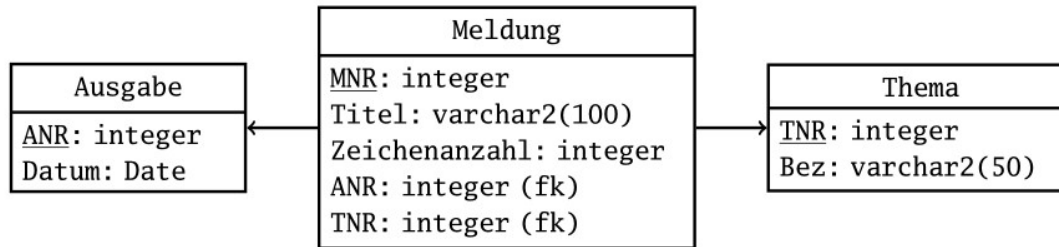
- Tabelle **Meldung** beinhaltet einen automatisch erzeugten Primärschlüssel.

}

}

Indizes - Lösung

Dienstag, 7. Januar 2020 12:06



Gehen Sie davon aus, dass für jede Tabelle des Klausurdatenmodells ein Index auf dem Primärschlüssel erstellt wurde. Ist es unter folgenden Voraussetzungen sinnvoll, weitere Indexe zu erstellen?

- Es gibt ca. 10 Themen
- Es gibt tägliche Ausgaben (bis auf sonntags) für ca. 50 Jahre
- Es wird per Abfrage auf alle Meldungen einer Ausgabe bzw. auf alle Meldungen zu einem Thema zugegriffen.

Wenn ja, welche Spalte bzw. Spalten sollten einen Index bekommen? Begründen Sie Ihre Antwort.

Da nur Abfragen mit Bedingung (where) auf ANR und TNR stattfinden, kommen nur diese beiden Spalten in Frage.

Meldung: ANR: $1 / (50 * 300)$ ist $1/15000 < 5\%$ -> Index setzen

Meldung: TNR: $1/10 = 10\%$ -> kein Index

Index ab $\leq 5\%$

Transaktionen - Lösung

Dienstag, 7. Januar 2020 12:06

Nehmen Sie an, in der Tabelle **Meldung** befinden sich die folgenden Daten.

MNR	Titel	Zeichenanzahl
42	Deutschland gewinnt gegen Australien	523
43	Regierung weiterhin unter Druck	459

In zwei parallel laufenden Sitzungen auf der Datenbank werden die beiden unten aufgeführten Transaktionen **T1** und **T2** ausgeführt. Beide Transaktionen sind auf die Konsistenzstufe **Repeatable Read** eingestellt.

Was ist das Ergebnis der Ausführung der Transaktionen **T1** und **T2**? Begründen Sie Ihre Antwort.

Repeatable Read: lange slock, lange xlock

Zeitpunkt	Sitzung 1	Sitzung 2
1		begin T2;
2		select Zeichenanzahl into v_az from Meldung where MNR=42
3	begin T1;	
4	select Zeichenanzahl into v_az from Meldung where MNR=43	
5	update Meldung set Zeichenanzahl=519 where MNR=42	
6	commit T1	
7		update Meldung set Zeichenanzahl=499 where MNR=43
8		commit T2

Slock lang 42

Slock lang 43

Versuch xlock 42
Slock bereits voranden
Bleibt stehen

Versuch xlock 43
Slock bereits voranden
Bleibt stehen

Deadlock