Bildungszentrum Uster
Höhere Berufsbildung Uster
HBU

Prüfung „T-SQL - Transact SQL“ (Lösung)

Fach: Datenbanken 2

Dozent: Lukas Müller

Unterlagen: Keine

Hilfsmittel: Keine

Dauer: 60 Minuten

Bemerkungen: Alle Lösungen sind auf die vorgegebenen Blätter zu erstellen. Falls der Platz nicht reicht, ist die Rückseite des vorhergehenden Blattes zu verwenden.

Name: _____

Klasse: _____

Schulort: _____

Datum: _____

Punkte: _____ / 28

Note: _____

Korrektor: Lukas Müller _____

Notenskala: $\frac{\textit{Erreichte Punktzahl} \times 5}{\textit{Max. Punktzahl}} + 1 =$ Note (auf 1/10 Noten gerundet)

Aufgaben

A1 2

a) Nennen Sie mir min. 2 Vorteile die sich bei der Verwendung von DB-Prozeduren ergeben.
b) Gibt es auch Nachteile, falls ja welche?

Kapselung von SQL-Befehlen (Logik), keine redundante Programmierung, Reduktion des Datenaustauschs zwischen Server und Client.

Ja, DB-Prozeduren sind Datenbankspezifisch → hoher Aufwand bei Portierung

A2 2

Nennen Sie mir min. 4 Kontrollstrukturen von T-SQL.

SET, IF – THEN – ELSE, LOOP, WHILE, FOR, RETURN

A3 2

a) Erklären Sie die grundlegenden Unterschiede zwischen Prozeduren und Funktionen einer Programmiersprache.
b) Erklären Sie den zentralen Unterschied (auf die Anwendung bezogen) zwischen Prozeduren und Funktionen in der Datenbankprogrammierung.

Prozeduren haben im Unterschied zu Funktionen keinen Rückgabewert (Return Code).

Funktionen können anstelle von Tabellen (retruns as Table) oder Spaltennamen in Abfragen verwendet werden.

A4 2

a) Ist es möglich beim Aufruf einer DB-Prozedur auch Input / Output Parameter angeben zu können?
b) Erläutern Sie den Unterschied zwischen "***call by reference***" bzw. "***call by value***" in Bezug auf die Parameterübergabe beim Aufruf einer Prozedur.

Ja

Bei call by reference wird ein Zeiger (Speicheradresse) einer Variable, bei call by value wird eine Kopie der Variable an die Prozedur übergeben.

A5 2

Erklären Sie den Unterschied zwischen einer Stored Procedures und einem DB-Trigger?

Die Trigger-Prozeduren gehörten immer zu einer Tabelle. Trigger-Prozeduren besitzen keine Input-/ Ouput-Parameter und können nicht aufgerufen werden.

A6 2

Zu welchen SQL-Befehlen können DB-Trigger einer Tabelle hinterlegt werden?

DELETE, INSERT, UPDATE

A7 2

Wozu dienen die Tabellen ***deleted*** und ***inserted*** in einer DB-Triggerprozedur?

Logische Tabellen welche nur in einer Trigger-Prozedur sichtbar sind. Sie enthalten die Datensätze welche eingefügt bzw. gelöscht wurden.

A8 2

Auf welche zwei Arten kann die referenzielle Integrität in einer Datenbank implementiert werden?

Deklarativ mit Primary Key, Foreign Key (ALTER TABLE ADD CONSTRAINT ...)

Mit Hilfe von Trigger - Prozeduren

A9 2

Was verstehen Sie unter dem Begriff ***Exception-Handling*** und wo findet es seine Anwendung?

Ausnahmebehandlung. Auftretende Fehler werden durch die Definition einer Fehlerbehandlungsroutine abgefangen und können individuell behandelt werden.

A10 2

Schreiben Sie eine Stored Procedure "***uspErhoehePreis***" welche in der Tabelle Product (ID, Name, Preis) eine Preiserhöhung von 5% durchführt.

```
CREATE PROCEDURE uspErhoehePreis
AS
UPDATE TAB_PRODUCT
       SET PREIS = PREIS * 1.05
```

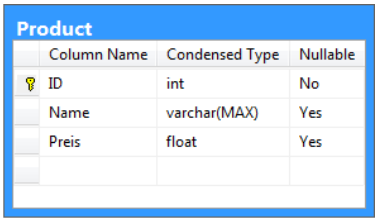
Product

Column Name	Condensed Type	Nullable
ID	int	No
Name	varchar(MAX)	Yes
Preis	float	Yes

A11 2

Schreiben Sie eine Stored Procedure "***uspInsertProdukt***" welche in der Tabelle Product (ID , Name, Preis) ein neues Produkt hinzufügt.

```
CREATE PROCEDURE upsInsertProdukt
       @Id    int,
       @Name  nvarchar(max),
       @preis float
AS
INSERT INTO TAB_PRODUCT (ID, Name, Preis)
       Values(@Id, @Name, @preis)
```



A12 2

Schreiben Sie den Prozeduraufruf um nachfolgendes Produkt mit der oben erstellten Stored Procedure „***uspInsertProdukt***“ einzufügen.

Produktdaten:

ID = 2211
Name = Scott
Preis = 3500.00

```
EXEC uspInsertProdukt 2211, ‚Scott‘, 3500.00
```

A13 2

Sie möchten in einem Trigger den Wert eines Feldes in einer Tabelle setzen. Wenn der Anwender einen Datensatz ändert, soll automatisch das Feld ***letzte_bearbeitung (DateTime)*** gesetzt werden. Wie komme ich am schnellsten zu einem Ergebnis (der Trigger soll für die vollständige Datenmenge zuständig sein)?

```
CREATE TRIGGER trTestTblUpdate ON dbo.TestTbl FOR UPDATE
AS
IF (@@ROWCOUNT = 0)
    RETURN

UPDATE dbo.TestTbl
SET datum = CURRENT_TIMESTAMP
WHERE recid IN (SELECT recid FROM INSERTED)

GO
```

A14 2

In einer Lagerdatenbank wird die Regel festgelegt, dass gleichzeitig beim Löschen eines Produktes aus Tabelle ***TAB_PRODUCT (PROD_ID, PROD_NAME, PROD_PREIS)*** auch die Lagerplätze Tabelle ***TAB_LAGERORT (LAG_ID, LAG_BEZ, PROD_ID)*** des jeweiligen Produktes mitgelöscht werden.

a) Wie kann diese Anforderung in einer Datenbank gelöst werden?
b) Erstellen Sie hierzu ein Lösungsbeispiel.

a) Durch die Definition eines Delete-Triggers bei der Tabelle TAB_PRODUCT.

b)

```
create trigger td_tab_product on TAB_PRODUCT
for delete as
begin

    delete TAB_LAGERORT
    from   TAB_LAGERORT lg, deleted d
    where  lg.PROD_ID = d.PROD_ID
    return
end
```