

Vitamin Quiz 모범 답안

3장

P85

```
SELECT prodName, company, amount FROM productTbl WHERE cost >= 8 ;
```

P91

```
CREATE VIEW uv_over10
AS
    SELECT * FROM productTbl WHERE cost >= 10;
```

P92

```
CREATE PROC over10Proc
AS
    SELECT * FROM productTbl WHERE cost >= 10;
```

P103

데이터파일의 최대 크기 → 16 TB(테라바이트)
로그파일의 최대 크기 → 2 TB(테라바이트)

6장

P184

```
SELECT Name, height FROM userTbl
WHERE height = ANY
    (SELECT height FROM userTbl WHERE addr = '서울' AND mobile1 = '011')
```

P187

```
SELECT DISTINCT mobile1 FROM userTbl WHERE mobile1 IS NOT NULL ORDER BY mobile1
```

P195

```
SELECT userid AS [사용자아이디], SUM(price*amount*1.0) / COUNT(userid) AS [평균구매액]
FROM buyTbl GROUP BY userid
```

P201

```
SELECT AVG(price*amount*1.0) from #tmpTbl
```

DECLARE @total bigint -- @total은 총구매액을 누적시킬변수

DECLARE @priceXamount int -- @priceXamount는 각행의 가격과 수량을 곱한값을 저장할 변수

DECLARE @num int -- @num 은구매테이블의 순번열의 증가

DECLARE @count int -- @count는 전체행 숫자

SET @total=0

SET @num=1

SELECT @count=COUNT(*) FROM #tmpTbl

SET @count = @count + 1

WHILE @count <> @num

BEGIN

SELECT @priceXamount = price * amount FROM #tmpTbl

WHERE num = @num

SET @num = @num + 1

SET @total = @total + @priceXamount

END

PRINT @total / (@count - 1.0)

제목 없음 - 1 (GIS)				
EventClass	Duration	TextData	SPID	BinaryData
SQL:BatchCompleted	6	SELECT AVG(price*amount*1.0) from #...	53	
SQL:BatchCompleted	67	DECLARE @total bigint -- @total은 ...	53	

P212

WITH cte_buyTbl(userid, maxPurchase)

AS

(SELECT userid, MAX(price*amount) FROM buyTbl GROUP BY userid)

SELECT AVG(maxPurchase*1.0) AS [각 사용자별 최고구매액의 평균] FROM cte_buyTbl

P220

교재의 P266과 온라인도움말의 "AdventureWorks [SQL Server]" → "테이블"을 참조할 것!

(실습4와 동일하게 하기 위해서는 좀 까다로우므로 참조만 할 것)

P235

```
SELECT price, amount, CONVERT(DECIMAL(10,2), CONVERT(FLOAT, price)/amount)
      AS [단가/수량] FROM buyTbl
```

P243

```
UPDATE maxTbl SET
  col1 = STUFF( (SELECT col1 FROM maxTbl), 1, 1000000,
                (SELECT REVERSE((SELECT col1 FROM maxTbl)))),
  col2 = STUFF( (SELECT col2 FROM maxTbl), 1, 1000000,
                (SELECT REVERSE((SELECT col2 FROM maxTbl))))
```

P248

```
USE AdventureWorks
SELECT ROW_NUMBER( ) OVER(ORDER BY HireDate ASC) [입사 순위],
      LoginID, Title, HireDate
FROM HumanResources.Employee
ORDER BY HireDate ASC
```

```
USE AdventureWorks
SELECT ROW_NUMBER( ) OVER(ORDER BY HireDate ASC, LoginID ASC)[입사 순위],
      LoginID, Title, HireDate
FROM HumanResources.Employee
ORDER BY HireDate ASC
```

```
USE AdventureWorks
SELECT Title,
      ROW_NUMBER( ) OVER( PARTITION BY Title
                          ORDER BY HireDate ASC, LoginID ASC)[직종별 입사 순위],
      LoginID, HireDate
FROM HumanResources.Employee
ORDER BY Title, HireDate ASC
```

```
USE AdventureWorks
SELECT DENSE_RANK( ) OVER(ORDER BY HireDate ASC)[입사 순위],
      LoginID, Title, HireDate
FROM HumanResources.Employee
ORDER BY HireDate ASC
```

```
USE AdventureWorks
SELECT RANK( ) OVER(ORDER BY HireDate ASC)[입사 순위],
      LoginID, Title, HireDate
```

```
FROM HumanResources.Employee
ORDER BY HireDate ASC
```

```
USE AdventureWorks
SELECT NTILE(2) OVER(ORDER BY HireDate ASC) [반번호],
    LoginID, Title, HireDate
FROM HumanResources.Employee
ORDER BY HireDate ASC
```

```
USE AdventureWorks
SELECT NTILE(4) OVER(ORDER BY HireDate ASC) [반번호],
    LoginID, Title, HireDate
FROM HumanResources.Employee
ORDER BY HireDate ASC
```

P254

```
SELECT E.EmployeeID, C.FirstName, C.MiddleName, C.LastName, C.EmailAddress
FROM HumanResources.Employee E
    INNER JOIN Person.Contact C
        ON E.EmployeeID = C.ContactID
```

P259

```
SELECT S.stdName, S.addr, C.clubName, C.roomNo
FROM stdTbl S, stdclubTbl SC, clubTbl C
    WHERE S.stdName = SC.stdName AND SC.clubName = C.clubName
ORDER BY S.stdName
```

```
SELECT C.clubName, C.roomNo, S.stdName, S.addr
FROM stdTbl S, stdclubTbl SC, clubTbl C
    WHERE SC.stdName = S.stdName AND SC.clubName = C.clubName
ORDER BY C.clubName
```

7장

P298

동일하게 수행됨.

P305

<실습2>의 4의 ①번과 같이 PUBS.MDF 파일을 이동한 후에,

USE master

CREATE DATABASE Pubs

ON (FILENAME= N'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\pubs.mdf')

FOR ATTACH.

P318

CREATE DATABASE [brain2DB]

ON PRIMARY

(NAME = N'brain2DB',

FILENAME = N'C:\temp\brain2DB.mdf' ,

SIZE = 10MB ,

MAXSIZE = 100MB ,

FILEGROWTH = 5MB)

LOG ON

(NAME = N'brain2DB_log',

FILENAME = N'C:\temp\brain2DB_log.LDF' ,

SIZE = 3MB ,

MAXSIZE = 1GB)

P323

교재 <실습5> 참조

8장

P366

먼저 현재 구매 테이블의 ID값을 확인

USE tableDB;

DBCC CHECKIDENT('buyTbl');

순번을 원하는 번호로 입력 (예로 다음에 10번부터 들어가게 하려면)

DBCC CHECKIDENT('buyTbl', RESEED, 9);

P371

[그림 8-1]을 보고 먼저 회원테이블(userTbl)의 데이터를 모두 입력한 후에, 구매테이블(buyTbl)을 입력

P375

개체탐색기에서 해당 테이블을 선택한 후, "오른쪽 마우스 버튼" >> "디자인"을 선택 →

Ctrl 키를 누른 상태에서 여러 개의 열을 선택한 후에, "오른쪽 마우스 버튼" >> "기본키 설정"을 선택하면 한번에

여러 개의 열이 기본키로 설정됨.

P380

제약조건의 내용이 충돌되지 않는다면 여러 개의 체크제약조건이 설정 가능.

예로 키가 0~200으로 설정하기 위해서 두 제약조건을 사용할 수 있음.

```
ALTER TABLE userTbl
    ADD CONSTRAINT CK_height
    CHECK
    (height >= 0)
```

```
ALTER TABLE userTbl
    ADD CONSTRAINT CK_height2
    CHECK
    (height <= 200)
```

위처럼 두개의 체크 제약조건은 가능하지만, 추가로 아래의 제약조건은 오류가 발생됨.

```
ALTER TABLE userTbl
    ADD CONSTRAINT CK_height3
    CHECK
    (height <= 100)
```

이미 제약조건1,2번에 의해서 0~200의 조건이지만, 추가로 100이하를 지정하는 것이므로 충돌됨.

P385

다른 세션을 새로 열고 아래와 같이 수행하면

```
USE tempDB;
```

```
DROP TABLE ##tempTbl; ➔ 성공 (전역 테이블이므로 테이블을 삭제 가능)
```

```
DROP TABLE #tempTbl; ➔ 실패 (지역 테이블이므로 테이블 존재를 다른 세션에서는 알수 없음)
```

P388

오류가 발생됨.

예제)

```
USE tempDB;
```

```
CREATE TABLE test1 ( a NCHAR(10));
```

```
INSERT INTO test1 VALUES ('뇌를자극하는시리즈');
```

```
ALTER TABLE test1
```

```
    ALTER COLUMN a NCHAR(5);
```

----- 오류 메시지 -----

메시지 8152, 수준 16, 상태 13, 줄 1

문자열이나 이진 데이터는 잘립니다.

문이 종료되었습니다.

P401

새로운 로그인과 사용자를 생성한 후에, 그 사용자의 디폴트 스키마를 HumanResources 로 지정하면 됨. 해당 하는 상세한 내용은 16장에서 설명됨.

P402

THIRD 인스턴스를 먼저 실행한 후에,

```
EXEC sp_addlinkedserver
```

```
    @server='THIRD',
```

```
    @srvproduct='SQLServer',
```

```
    @provider='SQLNCLI',
```

```
    @datasrc='BRAIN\THIRD'
```

P403

- 일반 쿼리문보다 속도가 약간 느려질 수 있다.
- 별도로 뷰를 관리해야 하므로, 관리차원에서도 약간 부담이 늘어난다.

P409

6장을 참조해서 직접 실습

P412

뷰와 관련있는 카탈로그 뷰 : sys.objects, sys.tables, sys.views

sys.sql_dependencies 는 온라인 도움말 참조

9장

P438

```
CREATE TABLE tbl6
(
    a INT UNIQUE ,
    b INT ,
    c INT ,
    d INT
)

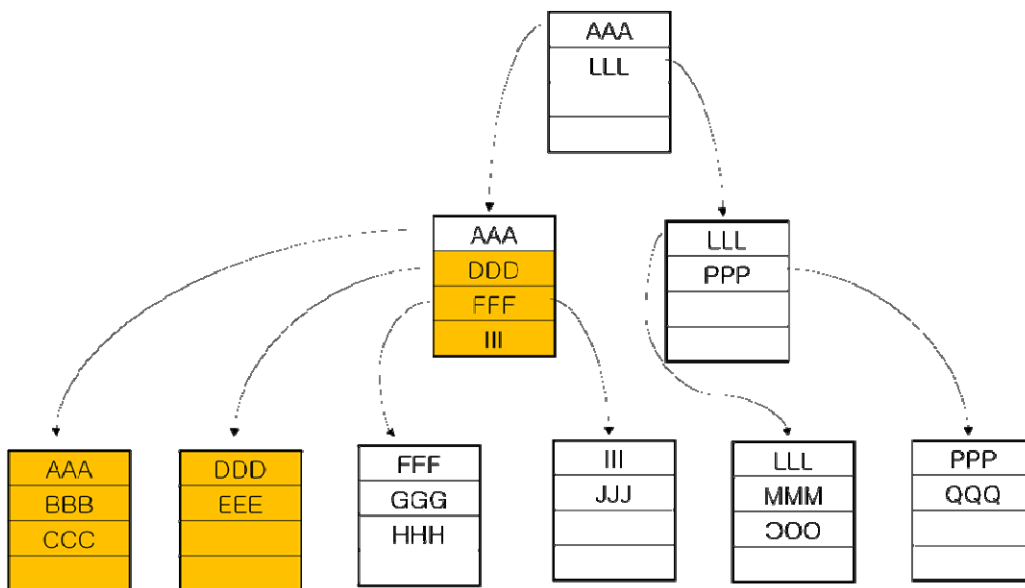
GO

EXEC sp_helpindex tbl6;
```

	index_name	index_description	index_keys
1	UQ__tbl6__308E3499	nonclustered, unique, unique key located on PRIMARY	a

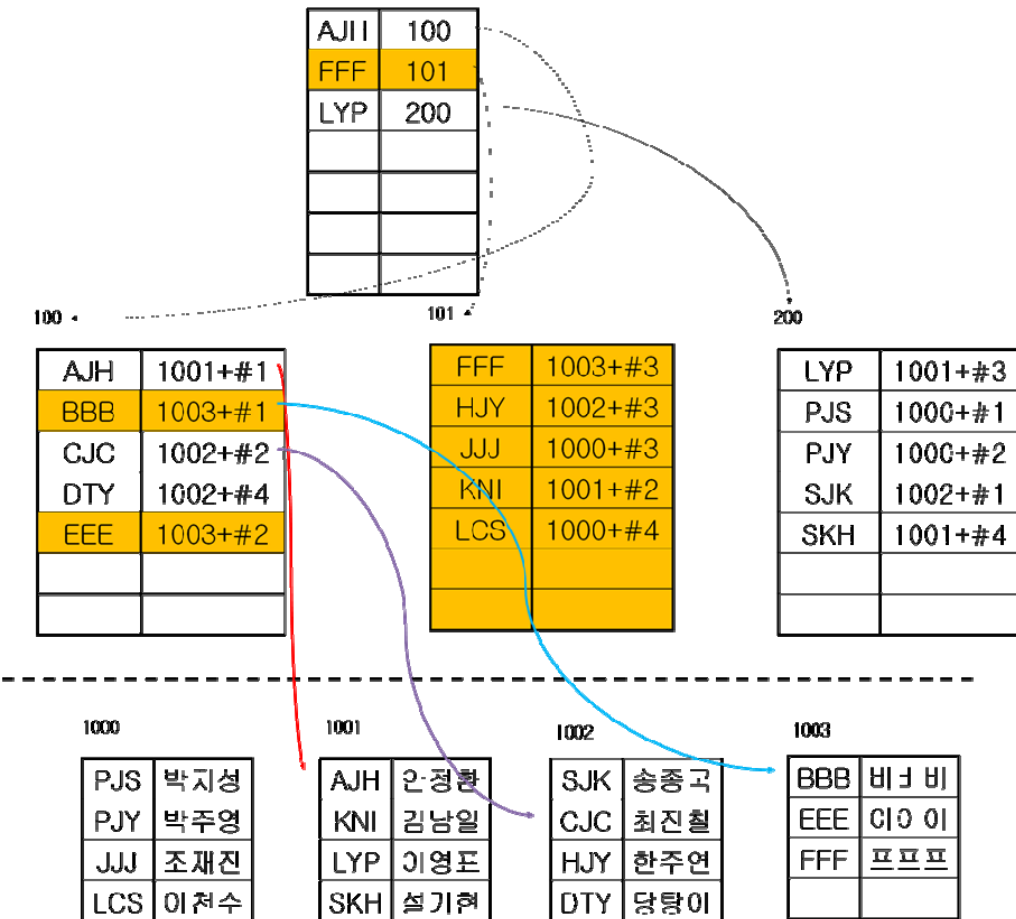
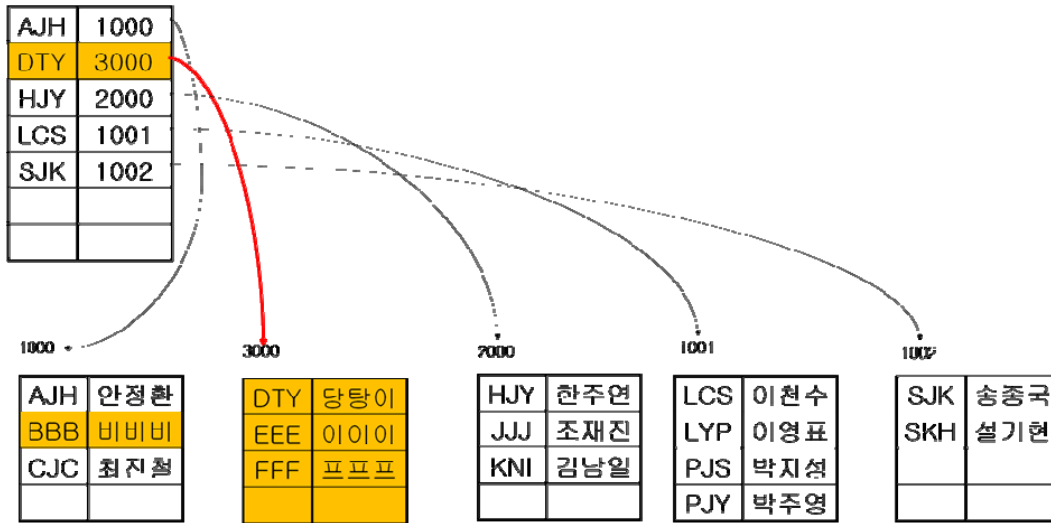
그외의 제약조건은 Clustered, Nonclustered 를 붙일 수 없음.

P441



P449

BBB(비비비), EEE (이이이), FFF(프프프) 3개가 입력된다고 가정하면



P472

[그림 9-37]을 보면

우선, 100을 찾기 위해 루트페이지, 001페이지를 읽고

또, 200을 찾기 위해 루트페이지, 001페이지를 읽고

또, 300을 찾기 위해 루트페이지, 001페이지를 읽었으므로

검색은 3회가 되고, 각 검색당 2페이지씩 읽었으므로 총 6페이지를 읽은 것이다.

P485

* 포괄열이 있는 인덱스 생성 이전

```
Select * INTO Cust_Include from AdventureWorks.Sales.Customer ORDER BY rowguid;
EXEC sp_IndexInfo Cust_Include
```

	테이블이름	인덱스이름	인덱스타입	페이지개...	크기(KB)	행개수
1	Cust_Include	NULL	HEAP	137	1096	19185

* 포괄열이 있는 인덱스 생성 이후

```
CREATE NONCLUSTERED INDEX idx_cust_include
    ON Cust_Include (CustomerID)
    INCLUDE (AccountNumber, CustomerType);
EXEC sp_IndexInfo Cust_Include
```

	테이블이름	인덱스이름	인덱스타입	페이지개...	크기(KB)	행개수
1	Cust_Include	NULL	HEAP	137	1096	19185
2	Cust_Include	idx_cust_include	NONCLUSTERED	74	592	19185

P503

```
CREATE VIEW vProfit WITH SCHEMABINDING AS
    SELECT ProductID, Name , (ListPrice - StandardCost) AS Profit FROM Production.Product
    WHERE (ListPrice - StandardCost) >= 1000;
```

GO

```
CREATE UNIQUE CLUSTERED INDEX vIdx_vProfit ON vProfit (ProductID);
```

```
SELECT ProductID, Name FROM Production.Product WHERE (ListPrice - StandardCost) >= 500
```

⇒ 인덱싱된 뷰 사용하지 않음

Clustered Index Scan	
클러스터형 인덱스에서 전체 또는 특정 범위를 검색합니다.	
물리적 연산	Clustered Index Scan
Logical Operation	Clustered Index Scan
실제 행 수	63
예상 I/O 비용	0.0120139
예상 CPU 비용	0.0007114
예상 연산자 비용	0.0127253(100%)
예상 하위 트리 비용	0.0127253
예상 행 수	151.2
예상 행 크기	81바이트
실제 다시 바인딩 횟수	0
실제 되감기 횟수	0
정렬됨	False
노드 ID	1
조건자	
([AdventureWorks].[Production].[Product].[ListPrice]-[AdventureWorks].[Production].[Product].[StandardCost])>=(\$500.0000)	
개체	
[AdventureWorks].[Production].[Product].[PK_Product_ProductID]	
출력 목록	
[AdventureWorks].[Production].	

SELECT ProductID, Name FROM Production.Product WHERE (ListPrice - StandardCost) >= 1000

⇒ 15페이지 읽음

{19개 행 적용됨}
테이블 'Product'. 검색 수 1, 논리적 읽기 수 15, 물리적 읽기 수 0, 미리 읽기 수 0, LOB 논리적 읽기 수 0,

10장

P538

P534의 잠금힌트를 사용해서 직접 실습

P544

<실습8>의 1번의 쿼리창1에서

SET DEADLOCK_PRIORITY HIGH

를 설정하고 다시 실습하면 쿼리창2의 쿼리문이 희생됨.

P549

SECONDBANK에 '당탕이'의 계좌 자체가 없는 상태에서 아래를 실행하면,

```
BEGIN DISTRIBUTED TRANSACTION
```

```
    UPDATE bankBook SET money = money - 200 WHERE NAME = '우재남';
```

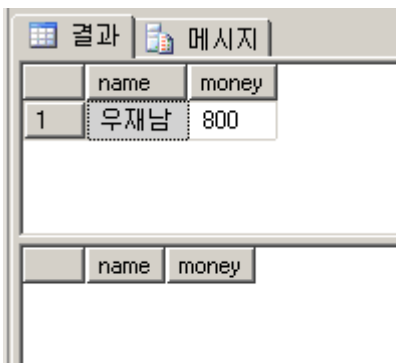
```
    UPDATE SECONDBANK.tempDB.dbo.bankBook SET money = money + 200
```

```
        WHERE NAME = '당탕이';
```

```
COMMIT TRANSACTION
```

```
SELECT * FROM bankBook;
```

```
SELECT * FROM SECONDBANK.tempDB.dbo.bankBook;
```



	name	money
1	우재남	800

‘우재남’의 계좌에서는 돈이 빠지고 ‘당탕이’는 아예 계좌가 없으므로 당연히 돈이 아예 안들어 간다. 결국 200원이 공중에서 사라졌다.

이를 해결하기 위해서는 아래와 같이 수정할 수 있다.

```
BEGIN TRY
```

```
    BEGIN DISTRIBUTED TRANSACTION
```

```
        UPDATE bankBook SET money = money - 200 WHERE NAME = '우재남';
```

```
        IF ((SELECT COUNT(*) FROM SECONDBANK.tempDB.dbo.bankBook
            WHERE NAME = '당탕이') >= 1)
```

```
            UPDATE SECONDBANK.tempDB.dbo.bankBook
```

```
                SET money = money + 200 WHERE NAME = '당탕이';
```

```
        ELSE
```

```
            RAISERROR(' ',16,1)
```

```
    COMMIT TRANSACTION
```

```
END TRY
```

```
BEGIN CATCH
```

```
    ROLLBACK TRANSACTION
```

```
END CATCH;
```

11장

P568

```
CREATE PROC usp_Encrypt2 WITH ENCRYPTION
AS
    SELECT 'X' + SUBSTRING(name,2,2) as [이름],
           YEAR(GETDATE()) - birthYear as '나이', height as '신장' FROM userTbl;

GO

EXECUTE usp_Encrypt2;
```

P571

(1) sp_pkeys는 PRIMARY KEY 제약 조건을 사용하여 명시적으로 정의된 열에 대한 정보를 반환합니다

```
EXEC sp_pkeys @table_name = N'buyTbl', @table_owner = N'dbo'
```

(2) 기본 키 테이블 이름은 제공되고 외래 키 테이블 이름이 NULL인 경우 sp_fkeys는 외래 키를 포함한 모든 테이블을 지정된 테이블로 반환합니다.

```
EXEC sp_fkeys @pktable_name = N'userTbl', @pktable_owner = N'dbo'
```

sp_tables 및 sp_databases 는 Books Online을 참조

P582

클러스터형 인덱스를 생성시에는 작동이 조금 다르다.

인덱스 생성시에 "CREATE CLUSTERED INDEX idx_spTbl_id on spTbl (CustomerID);" 로 변경한 후 동일하게 직접 실행해 볼 것.

P595

모두 가능

```
EXEC sp_rename 'buyTbl.groupName', 'gname', 'COLUMN'; ➔ 성공
```

```
ALTER TABLE buyTbl ADD etc NCHAR(10); ➔ 성공
```

12장

P613

(1) KEYSET

```
USE master ;
```

```
RESTORE DATABASE sqlDB FROM DISK ='c:\sqlldb.bak' WITH REPLACE ;
```

```
USE sqlDB;
```

```

DECLARE userTbl_cursor1 CURSOR GLOBAL KEYSET
    FOR SELECT * FROM userTbl;

OPEN userTbl_cursor1;

FETCH NEXT FROM userTbl_cursor1;

UPDATE userTbl SET addr = N'미국';
FETCH NEXT FROM userTbl_cursor1; ➔ 변경된 '미국'으로 주소가 보임

INSERT INTO userTbl VALUES('ZZZ',N'마징가',1960,N'일본','000','0000000','999');

FETCH NEXT FROM userTbl_cursor1; ➔ 계속 반복해도 '마징가'는 안보임

CLOSE userTbl_cursor1;
DEALLOCATE userTbl_cursor1;
    
```

(2) DYNAMIC

```

USE master ;
RESTORE DATABASE sqlDB FROM DISK ='c:\sqlldb.bak' WITH REPLACE ;
USE sqlDB;

DECLARE userTbl_cursor2 CURSOR GLOBAL DYNAMIC
    FOR SELECT * FROM userTbl;

OPEN userTbl_cursor2;

FETCH NEXT FROM userTbl_cursor2;

UPDATE userTbl SET addr = N'미국'; ➔ 변경된 '미국'으로 주소가 보임
FETCH NEXT FROM userTbl_cursor2;

INSERT INTO userTbl VALUES('ZZZ',N'마징가',1960,N'일본','000','0000000','999');

FETCH NEXT FROM userTbl_cursor2; ➔ 계속 반복하면 '마징가'가 마지막에 보임

CLOSE userTbl_cursor2;
DEALLOCATE userTbl_cursor2;
    
```

P619

확인이 됨.

```
DEALLOCATE keysetTbl_cursor;
DECLARE keysetTbl_cursor CURSOR GLOBAL FORWARD_ONLY DYNAMIC TYPE_WARNING
FOR SELECT id, txt FROM keysetTbl;
```

```
DECLARE @result CURSOR
EXEC sp_describe_cursor @cursor_return = @result OUTPUT,
    @cursor_source = N'GLOBAL', @cursor_identity = N'keysetTbl_cursor'
FETCH NEXT from @result
WHILE (@@FETCH_STATUS <> -1)
    FETCH NEXT FROM @result
```

```
OPEN keysetTbl_cursor;
FETCH NEXT FROM keysetTbl_cursor;
```

INSERT INTO keysetTbl VALUES(4,'XXX');

FETCH NEXT FROM keysetTbl_cursor; ➔ 여러 번 실행하면 'XXX'가 보임.

13장

P634

UPDATE uv_deliver SET userid = 'DDY' WHERE userid='DTI'; ➔ 작동 실패(제약조건으로 인함)

```
CREATE TRIGGER trg_update
ON uv_deliver
INSTEAD OF UPDATE
AS
BEGIN
```

ALTER TABLE buyTbl NOCHECK CONSTRAINT FK_buyTbl_userid_7F60ED59

➔ 제약조건을 잠깐 작동중지시킴

UPDATE userTbl SET userid = (SELECT userid FROM inserted)
WHERE userid= (SELECT userid FROM deleted)

UPDATE buyTbl SET userid = (SELECT userid FROM inserted)
WHERE userid= (SELECT userid FROM deleted)

ALTER TABLE buyTbl CHECK CONSTRAINT FK_buyTbl_userid_7F60ED59

➔ 제약조건을 다시 활성화시킴

END;

UPDATE uv_deliver SET userid = 'DDY' WHERE userid='DTI'; ➔ 작동 성공

P651

USE triggerDB;

SELECT name, is_recursive_triggers_on FROM sys.databases
WHERE name = 'triggerDB';

ALTER DATABASE triggerDB
SET RECURSIVE_TRIGGERS ON;

CREATE TRIGGER **trg_recuAA**

ON **recuAA**

AFTER INSERT

AS

IF ((SELECT trigger_nestlevel()) >= 32)

RETURN

DECLARE @id INT

SELECT @id = trigger_nestlevel() -- 현재트리거레벨값

PRINT '트리거레벨==> ' + CAST(@id AS CHAR(5))

INSERT INTO **recuAA** VALUES ('직접재귀트리거')

GO

INSERT INTO recuAA VALUES ('처음입력값');

select * from recuAA;

14장

P661

도움말의 색인에서 "contains"를 검색한 후에 아래 부분을 참조한다.

CONTAINS(Transact-SQL) 부분 도움말의 “예” 6번.

6. <generation_term>에 CONTAINS 사용

다음 예에서는 riding, ridden 등 ride에서 파생된 단어가 있는 모든 제품을 검색합니다.

코드 복사

```
USE AdventureWorks;
```

```
GO
```

```
SELECT Description
```

```
FROM Production.ProductDescription
```

```
WHERE CONTAINS(Description, ' FORMSOF (INFLECTIONAL, ride) ');
```

```
GO
```

P673

직접 noiseKOR.txt 파일에 단어를 추가해서 실습

16장

P708

Books Online 참조

▣ 고정 서버 역할의 사용 권한 매핑

고정 서버 역할	서버 수준 사용 권한
bulkadmin	허가된 사용 권한: ADMINISTER BULK OPERATIONS
dbcreator	허가된 사용 권한: CREATE DATABASE
diskadmin	허가된 사용 권한: ALTER RESOURCES
processadmin	허가된 사용 권한: ALTER ANY CONNECTION, ALTER SERVER STATE
securityadmin	허가된 사용 권한: ALTER ANY LOGIN
serveradmin	허가된 사용 권한: ALTER ANY ENDPOINT, ALTER RESOURCES, ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN, VIEW SERVER STATE
setupadmin	허가된 사용 권한: ALTER ANY LINKED SERVER
sysadmin	GRANT 옵션을 사용하여 허가된 사용 권한: CONTROL SERVER

▣ 사용 권한에 고정 데이터베이스 역할 매핑

고정 데이터베이스 역할	데이터베이스 수준 사용 권한	서버 수준 사용 권한
db_accessadmin	허가된 사용 권한: ALTER ANY USER, CREATE SCHEMA	허가된 사용 권한: VIEW ANY DATABASE
db_accessadmin	GRANT 옵션을 사용하여 허가된 사용 권한: CONNECT	
db_backupoperator	허가된 사용 권한: BACKUP DATABASE, BACKUP LOG, CHECKPOINT	허가된 사용 권한: VIEW ANY DATABASE
db_datareader	허가된 사용 권한: SELECT	허가된 사용 권한: VIEW ANY DATABASE
db_datawriter	허가된 사용 권한: DELETE, INSERT, UPDATE	허가된 사용 권한: VIEW ANY DATABASE
db_ddladmin	허가된 사용 권한: ALTER ANY ASSEMBLY, ALTER ANY ASYMMETRIC KEY, ALTER ANY CERTIFICATE, ALTER ANY CONTRACT, ALTER ANY DATABASE DDL TRIGGER, ALTER ANY DATABASE EVENT, NOTIFICATION, ALTER ANY DATASPACE, ALTER ANY FULLTEXT CATALOG, ALTER ANY MESSAGE TYPE, ALTER ANY REMOTE SERVICE BINDING, ALTER ANY ROUTE, ALTER ANY SCHEMA, ALTER ANY SERVICE, ALTER ANY SYMMETRIC KEY, CHECKPOINT, CREATE AGGREGATE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE QUEUE, CREATE RULE, CREATE SYNONYM, CREATE TABLE, CREATE TYPE, CREATE VIEW, CREATE XML SCHEMA COLLECTION, REFERENCES	허가된 사용 권한: VIEW ANY DATABASE
db_denydatareader	거부된 사용 권한: SELECT	허가된 사용 권한: VIEW ANY DATABASE
db_denydatawriter	거부된 사용 권한: DELETE, INSERT, UPDATE	
db_owner	GRANT 옵션을 사용하여 허가된 사용 권한: CONTROL	허가된 사용 권한: VIEW ANY DATABASE
db_securityadmin	허가된 사용 권한: ALTER ANY APPLICATION ROLE, ALTER ANY ROLE, CREATE SCHEMA, VIEW DEFINITION	허가된 사용 권한: VIEW ANY DATABASE

P714

```
CREATE LOGIN [quizUser1]
WITH PASSWORD='1234',
    DEFAULT_DATABASE = master,
    CHECK_EXPIRATION = OFF,
    CHECK_POLICY = ON ;

GO

EXEC sp_addsrvrolemember 'quizUser1', 'sysadmin';

GO
```

P720

<실습3>을 참조해서 직접 실습

P728

```
-- 로그인 및 사용자 3명 생성

USE sqlDB;
```

```
CREATE LOGIN quizUser3
WITH    PASSWORD = 'password',
        DEFAULT_DATABASE = [sqlDB],
        CHECK_POLICY = OFF,
        CHECK_EXPIRATION = OFF;

GO

CREATE USER quizUser3 FOR LOGIN quizUser3;
GO
```

```
CREATE LOGIN quizUser4
WITH    PASSWORD = 'password',
        DEFAULT_DATABASE = [sqlDB],
        CHECK_POLICY = OFF,
        CHECK_EXPIRATION = OFF;

GO

CREATE USER quizUser4 FOR LOGIN quizUser4;
GO
```

```
CREATE LOGIN quizUser5
WITH    PASSWORD = 'password',
        DEFAULT_DATABASE = [sqlDB],
        CHECK_POLICY = OFF,
        CHECK_EXPIRATION = OFF;

GO

CREATE USER quizUser5 FOR LOGIN quizUser5;
GO
```

-- 역할 생성

```
USE sqlDB;
CREATE ROLE quizROLE;
```

-- 역할에 권한 부여

```
GRANT SELECT,INSERT,DELETE,UPDATE ON OBJECT::dbo.userTbl TO quizROLE;
```

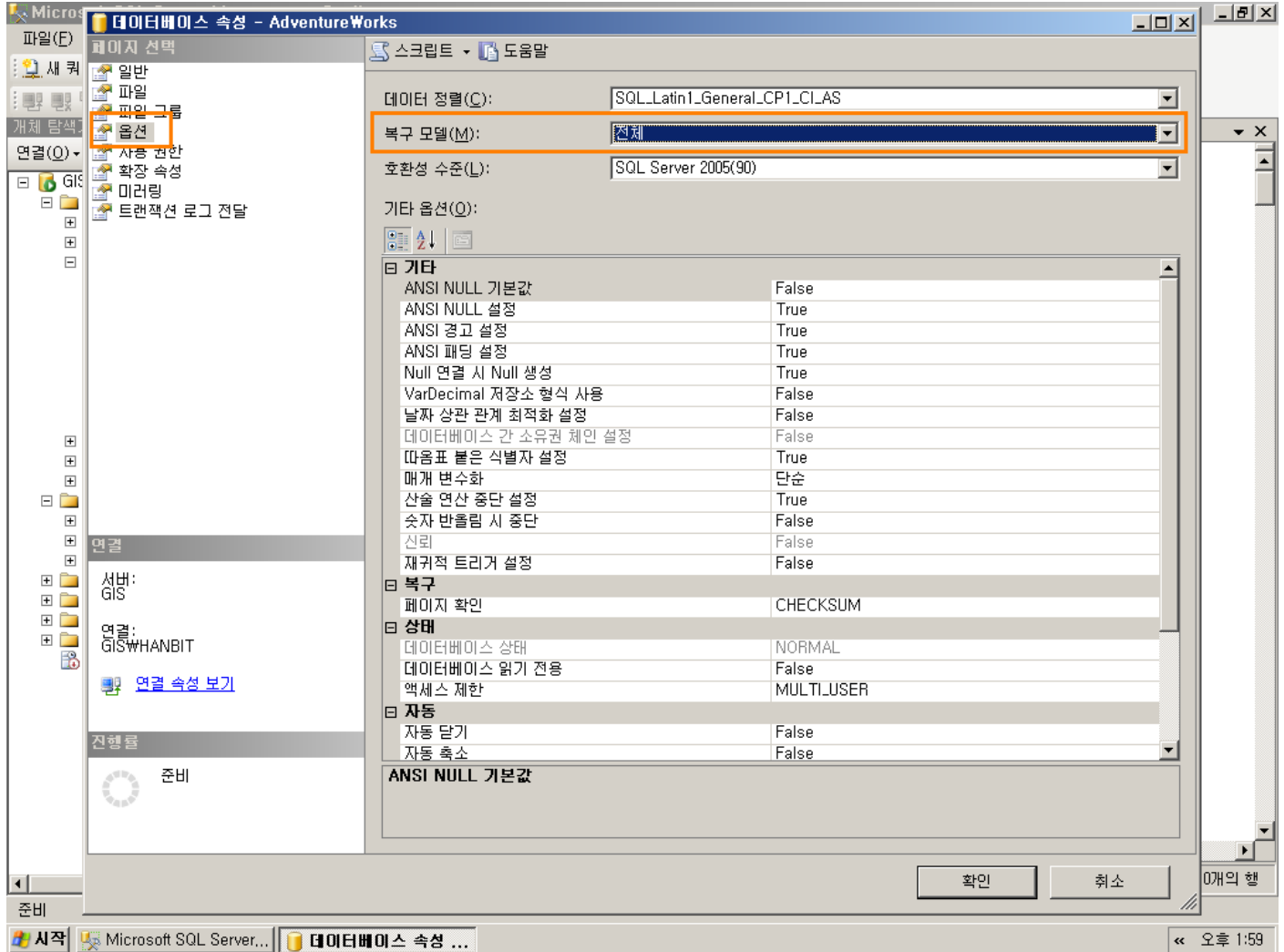
-- 역할에 사용자 추가

```
EXEC sp_addrolemember 'quizROLE', 'quizUser3'
EXEC sp_addrolemember 'quizROLE', 'quizUser4'
EXEC sp_addrolemember 'quizROLE', 'quizUser5'
```

17장

P748

AdventureWorks 의 데이터베이스 속성에서 변경



USE AdventureWorks;

-- 전체백업

BACKUP DATABASE AdventureWorks TO DISK='C:\Wadv.bak';

-- 차등백업

BACKUP DATABASE AdventureWorks TO DISK='C:\Wadv.bak' WITH DIFFERENTIAL;

-- 로그백업

BACKUP LOG AdventureWorks TO DISK='C:\Wadv.bak';

P753

```
sp_addumpdevice [ @devtype = ] 'device_type'
    , [ @logicalname = ] 'logical_name'
    , [ @physicalname = ] 'physical_name'
    [ , { [ @cntrltype = ] controller_type |
        [ @devstatus = ] 'device_status' }
    ]
```

P769

마지막 차등백업의 이전의 차등백업은 필요 없다. 그러므로, 복원순서는
1회 (전체) → 3회(차등) → 4회 (로그) → 임시로그백업 복원

P781

로그파일 삭제시의 오류 메시지

파일 활성화 오류입니다. 물리적 파일 이름 "c:\로그파일\testDB2_log.ldf"이(가) 잘못된 것 같습니다.
메시지 945, 수준 14, 상태 2, 줄 1
파일에 액세스할 수 없거나 메모리 또는 디스크 공간이 부족하여 데이터베이스 'testDB2'을(를) 열 수 없습니다.
자세한 내용은 SQL Server 오류 로그를 참조하십시오.
메시지 5069, 수준 16, 상태 1, 줄 1
ALTER DATABASE 문이 실패했습니다.

P788

```
CREATE DATABASE [testDB3] ON PRIMARY
( NAME = 'testDB3'
, FILENAME = 'c:\데이터파일\testDB3.mdf' )
FOR ATTACH ATTACH_REBUILD_LOG
```

P806

<실습10>을 참조해서 직접 실습

P813

백업은 현재의 데이터베이스를 별도의 장치에 보관하는 것을 말하며, 보관된 데이터는 복원시키기 전에는 사용할 수가 없는 상태가 된다.

스냅샷은 현재 데이터베이스의 상태를 보관하는 용도로 사용하지만, 보관된 데이터도 실시간으로 읽을 수가 있다. 하지만, 스냅샷은 백업의 대신으로 사용하면 오히려 큰 문제를 일으킬 소지가 있다.

18장

P820

- (1) 비용이 가장 많이 들더라도 서비스의 중단 시간이 있어서는 안된다.
➔ 서버 클러스터링 + 데이터베이스 미러링 + 디스크 미러링
- (2) 서비스가 잠시 중단되어도 큰 문제가 없고 되도록 최소의 비용으로 구성해야 한다.
➔ 로그 전달 또는 데이터베이스 미러링
- (3) 성능을 최대로 유지할 수 있어야 한다.
➔ 로그 전달 + 디스크 미러링(RAID 0)
- (4) 가장 짧은 서비스 중단을 보장해야 한다.
➔ 데이터베이스 미러링

P835

로그전달의 사용법은 교재를 참조.

장점은 다른 것에 비해서 프로세스를 적게 차지하며, SQL Server 2005 이전 버전에서도 제공됨.

P858

Books Online 의 색인에서 "데이터베이스 미러링 [SQL Server]"부분 참조

P903

Books Online 의 색인에서 "복제 [SQL Server]"부분 참조

19장

P929

- (1) <실습3>의 2-⑤ 이후에 오라클에서 아래 코드 입력

```
CREATE TABLE oraTbl (id NUMBER(3), NAME VARCHAR(10));
INSERT INTO oraTbl VALUES (1, '나일번');
INSERT INTO oraTbl VALUES (2, '나이번');
INSERT INTO oraTbl VALUES (3, '나삼번');
```

- (2) SQL Server에 sqlDB4 데이터베이스 생성

- (3) <실습4>의 1-① 에서 [작업] ➔ [데이터베이스 가져오기]를 클릭

(4) 그 이후는 내보내기와 비슷하므로 직접 실습

P930

MS Access 도 다른 DB와 동일한 개념으로 사용할 수 있음.

20장

P946

<실습2>의 2-④ 쿼리문을 아래처럼 수정

```
DECLARE @backupDate DATETIME
DECLARE @fileName NVARCHAR(50)
SELECT @backupDate = GETDATE();
SET @fileName = 'c:\W백업폴더\sqlDB-LOG-'
    + CAST(YEAR(@backupDate) AS CHAR(4)) + '-'
    + CAST(MONTH(@backupDate) AS VARCHAR(2)) + '-'
    + CAST(DAY(@backupDate) AS VARCHAR(2))
    + '.bak'
BACKUP LOG sqlDB TO disk = @fileName;
```

그 외는 필요에 따라 적절히 수정하면 됨.