[Table 1](#_Toc333824024)

[Create 1](#_Toc333824025)

[Primary key/index 1](#_Toc333824026)

[Select 1](#_Toc333824027)

[Select with join 1](#_Toc333824028)

[Insert 2](#_Toc333824029)

[Insert values 2](#_Toc333824030)

[Insert from table 2](#_Toc333824031)

[Update 2](#_Toc333824032)

[Simple 2](#_Toc333824033)

[Update select 3](#_Toc333824034)

[View 4](#_Toc333824035)

[Stored procedure 4](#_Toc333824036)

[Simple 4](#_Toc333824037)

[With cursor 4](#_Toc333824038)

[Function 5](#_Toc333824039)

[Trigger 6](#_Toc333824040)

[insert 6](#_Toc333824041)

[Update 7](#_Toc333824042)

[Recovery models 8](#_Toc333824043)

[Backups 8](#_Toc333824044)

[Full Backup 8](#_Toc333824045)

[Transaction Log 8](#_Toc333824046)

[Differential Backup 8](#_Toc333824047)

[Restores 9](#_Toc333824048)

[With recovery 9](#_Toc333824049)

[No Replace 9](#_Toc333824050)

[Replace 9](#_Toc333824051)

[With no recovery 9](#_Toc333824052)

[Database Mirroring 10](#_Toc333824053)

[Datatypes 10](#_Toc333824054)

[Clusters 10](#_Toc333824055)

[Indexes 12](#_Toc333824056)

[Fragmentation 12](#_Toc333824057)

[Database Design 12](#_Toc333824058)

[Normalization 12](#_Toc333824059)

# Table

## Create

SET ANSI\_NULLS OFF

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[dba\_options](

[timeout\_amount] [smallint] NOT NULL

) ON [PRIMARY]

### Primary key/index

ALTER TABLE [dbo].[dba\_options] ADD CONSTRAINT [PK\_dba\_options] PRIMARY KEY CLUSTERED

(

[timeout\_amount] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, IGNORE\_DUP\_KEY = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

## Select

**Simple select**

SELECT \*

FROM tablea a

WHERE a.id > 10

ORDER BY a.id ASC

## Select with join

SELECT a.name, b.address

FROM tablea a

LEFT JOIN tableb b ON (b.id = a.id)

WHERE a.id > 10

ORDER BY a.name ASC

## Insert

### Insert values

INSERT INTO tablea

(a.id,

a.name)

VALUES (

1,

‘jim’)

### Insert from table

INSERT tablea

(id,

Name)

SELECT id, name

FROM tableb

## Update

### Simple

UPDATE tablea

SET name = ‘testing’

### Update select

UPDATE tablea

SET name = (

SELECT name

FROM tablec)

# View

CREATE VIEW [dbo].[v\_dba\_last\_database\_backup] AS

SELECT

s.database\_name

,max(s.backup\_start\_date) AS max\_backup\_start\_date

,s.TYPE

FROM msdb.dbo.backupset s

JOIN dbo.dba\_database\_expiration e ON (s.database\_name = e.name)

GROUP BY s.database\_name, s.TYPE

# Stored procedure

## Simple

create Procedure p\_x

@parameter varchar(50)

As

BEGIN

SET NOCOUNT ON;

DECLARE @dbname varchar(30)

END

## With cursor

ALTER PROCEDURE [dbo].[p\_dba\_copybackups\_mirrored]

AS

DECLARE @local\_variable int

DECLARE back\_cursor CURSOR FOR

SELECT a.media\_set\_id,

a.family\_sequence\_number,

a.media\_family\_id,

a.physical\_device\_name,

b.backup\_start\_date,

b.backup\_finish\_date,

b.database\_name,

d.source\_server,

d.source\_share,

d.source\_path

FROM msdb.dbo.backupset b

JOIN dba\_db.dbo.dba\_disaster\_dblist d ON (b.database\_name = d.name)

INNER JOIN msdb.dbo.backupmediafamily a ON (b.media\_set\_id = a.media\_set\_id)

JOIN master.sys.databases s on (d.name = s.name)

RIGHT OUTER JOIN master.sys.database\_mirroring p on (s.database\_id = p.database\_id)

where (datediff(day,b.backup\_start\_date,getdate()) = 0 ) AND

charindex('hourly',a.physical\_device\_name) = 0 AND

(p.mirroring\_role = 1 OR

p.mirroring\_role IS null)

order by d.group\_id asc

OPEN back\_cursor

FETCH back\_cursor INTO @media\_set\_id, @family\_sequence\_number, @media\_family\_id,

@physical\_device\_name, @backup\_start\_date, @backup\_finish\_date, @database\_name,

@source\_server, @source\_share, @source\_path

WHILE (@@fetch\_status = 0)

BEGIN

FETCH back\_cursor INTO @media\_set\_id, @family\_sequence\_number, @media\_family\_id,

@physical\_device\_name, @backup\_start\_date, @backup\_finish\_date, @database\_name,

@source\_server, @source\_share, @source\_path

END

DEALLOCATE back\_cursor

RETURN 0

# Function

CREATE FUNCTION [dbo].[f\_dba\_get\_sqlversion]

(

-- Add the parameters for the function here

)

RETURNS nvarchar(4)

AS

BEGIN

-- Declare the return variable here

DECLARE @Result nvarchar(4)

-- Add the T-SQL statements to compute the return value here

SET @Result = substring(@@VERSION,patindex('%-%',@@VERSION)-5, 4 )

-- Return the result of the function

RETURN @Result

END

# Trigger

After or instead of

## insert

CREATE TRIGGER [dbo].[ti\_dba\_database\_expiration]

ON [dbo].[dba\_database\_expiration]

AFTER INSERT

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

--UPDATE dbo.dba\_database\_expiration

-- set last\_modified\_date = getdate()

-- FROM inserted i,

-- dba\_database\_expiration f

-- WHERE i.name = f.name

INSERT dbo.dba\_database\_expiration\_history

(name

,db\_owner

,db\_dbid

,db\_cr\_date

,db\_expiration\_date

,db\_restore\_date

,db\_purpose

,db\_application\_name

,db\_location

,db\_url

,db\_release\_number

,db\_status

,db\_backup\_db)

SELECT name

,db\_owner

,db\_dbid

,db\_cr\_date

,db\_expiration\_date

,db\_restore\_date

,db\_purpose

,db\_application\_name

,db\_location

,db\_url

,db\_release\_number

,db\_status

,1

FROM inserted

END

## Update

CREATE TRIGGER [dbo].[tu\_dba\_database\_expiration]

ON [dbo].[dba\_database\_expiration]

AFTER UPDATE

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

UPDATE dbo.dba\_database\_expiration

set last\_modified\_date = getdate()

FROM inserted i,

dba\_database\_expiration f

WHERE i.name = f.name

INSERT dbo.dba\_database\_expiration\_history

(name

,db\_owner

,db\_dbid

,db\_cr\_date

,db\_expiration\_date

,db\_restore\_date

,db\_purpose

,db\_application\_name

,db\_location

,db\_url

,db\_release\_number

,db\_status,

last\_modified\_date

,db\_backup\_db)

SELECT name

,db\_owner

,db\_dbid

,db\_cr\_date

,db\_expiration\_date

,db\_restore\_date

,db\_purpose

,db\_application\_name

,db\_location

,db\_url

,db\_release\_number

,db\_status,

getdate()

,db\_backup\_db

FROM inserted

END

# Recovery models

Full

Bulk log

Simple

# Backups

## Full Backup

BACKUP DATABASE [test] TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Backup\test\_backup\_20120604120003.bak' WITH NOFORMAT, NOINIT, NAME = N'test-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10

GO

Transaction Log – this one truncates

BACKUP LOG [test] TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Backup\test\_backup\_20120604120003.bak' WITH NOFORMAT, NOINIT, NAME = N'test-Transaction Log Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10

GO

Can also backup tail of log – this one does not truncate.

BACKUP LOG [test] TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Backup\test\_backup\_20120604120003.bak' WITH NO\_TRUNCATE , NOFORMAT, NOINIT, NAME = N'test-Transaction Log Backup', SKIP, NOREWIND, NOUNLOAD, NORECOVERY , STATS = 10

GO

## Differential Backup

BACKUP DATABASE [test] TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Backup\test\_backup\_20120604120003.bak' WITH DIFFERENTIAL , NOFORMAT, NOINIT, NAME = N'test-Differential Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10

GO

# Restores

## With recovery

### No Replace

RESTORE DATABASE [dba\_db] FROM DISK = N'D:\xpraid\_data\projects\dba\_db\db\2005\dba\_db\_backup\_201011100059.bak' WITH FILE = 1,

MOVE N'dba\_db' TO N'C:\Program Files\Microsoft SQL Server\MSSQL10\_50.DBS2008R2\MSSQL\DATA\dba\_db.mdf',

MOVE N'dba\_db\_log' TO N'C:\Program Files\Microsoft SQL Server\MSSQL10\_50.DBS2008R2\MSSQL\DATA\dba\_db\_1.ldf', NOUNLOAD, STATS = 5

RESTORE LOG [test1] FROM DISK = N'D:\xpraid\_data\projects\dba\_db\db\2005\dba\_db\_backup\_20110812010958.trn' WITH FILE = 1, NOUNLOAD, STATS = 10

### Replace

RESTORE DATABASE [dba\_db] FROM DISK = N'D:\xpraid\_data\projects\dba\_db\db\2005\dba\_db\_backup\_20110812005930.bak' WITH FILE = 1,

MOVE N'dba\_db' TO N'C:\Program Files\Microsoft SQL Server\MSSQL10\_50.DBS2008R2\MSSQL\DATA\dba\_db.mdf',

MOVE N'dba\_db\_log' TO N'C:\Program Files\Microsoft SQL Server\MSSQL10\_50.DBS2008R2\MSSQL\DATA\dba\_db\_1.ldf', NOUNLOAD, REPLACE, STATS = 5

## With no recovery

RESTORE DATABASE [dba\_db] FROM DISK = N'D:\xpraid\_data\projects\dba\_db\db\2005\dba\_db\_backup\_20110812005930.bak' WITH FILE = 1,

MOVE N'dba\_db' TO N'C:\Program Files\Microsoft SQL Server\MSSQL10\_50.DBS2008R2\MSSQL\DATA\dba\_db.mdf',

MOVE N'dba\_db\_log' TO N'C:\Program Files\Microsoft SQL Server\MSSQL10\_50.DBS2008R2\MSSQL\DATA\dba\_db\_1.ldf', NORECOVERY, NOUNLOAD, STATS = 5

RESTORE LOG [test1] FROM DISK = N'D:\xpraid\_data\projects\dba\_db\db\2005\dba\_db\_backup\_20110812010958.trn' WITH FILE = 1, NORECOVERY, NOUNLOAD, STATS = 10

# Database Mirroring

Principal

Mirror

Witness

High Performance (asynchronous)

Commit changes at principal then transmit changes to mirror.

High Safety w/o automatic failover (synchronous)

Always commit changes at both principal and mirror

High Safety with automatic failover (synchronous)

Requires a witness. Always commit changes at both principal and mirror if available. Witness contols automatic failover.

# Datatypes

Varchar(max)

# Clusters

At least 2 nodes

4 nic cards

6 ip address

Cluster ip, sql ip, 2 machine ip for heartbeat, 2 ip for regular

The heartbeat chcks 2/min.

Resource groups

There should be at least 2 resource groups. 1 for the sql and 1 for the o/s.

There is a quorum disk.

# Indexes

## Fragmentation

<5% do nothing

5 - <30% reorganize

30%+ rebuild.

# Database Design

## Normalization

|  |  |  |
| --- | --- | --- |
| Level |  |  |
| 1 | First normal form | No repeating groups – all repeating groups have been moved into separate tables. |
| 2 | 2nd normal form | Non key fields depend on the entire primary key(automatic if no compound keys) |
| 3 | 3rd normal form | Non key fields must not depend on other non-key fields. |
| 4 | 4th normal form |  |
| 5 | 5th normal form |  |