2401960435 – Reynaldy Sentosa – UAS DBS.

1.

SELECT

\*

FROM

Participant p

WHERE

EXISTS(

SELECT

\*

FROM

OnGoingCourse ogc JOIN

Course c ON

ogc.Course\_Id = c.Course\_Id

WHERE

Course\_Name = 'SQL Server' AND

[Level] LIKE 'Advanced' AND

ogc.Participant\_Id = p.Participant\_Id

)

2.

SELECT

\*

FROM

Teacher

WHERE

Teacher\_Id IN(

SELECT

Teacher\_Id

FROM

Course

WHERE

Course\_Name LIKE 'Project Management dan Leadership'

)

3.

CREATE VIEW

ShowActive

AS

SELECT

PFirst\_Name,

PLast\_Name,

Course\_Name

FROM

OnGoingCourse ogc JOIN

Course c ON

ogc.Course\_Id = c.Course\_Id JOIN

Participant p ON

ogc.Participant\_Id = p.Participant\_Id

WHERE

[Status] LIKE 'Active' AND

YEAR([Start\_date]) <= 2022 AND

YEAR(End\_date) >= 2022

EXCEPT

SELECT

PFirst\_Name,

PLast\_Name,

Course\_Name

FROM

OnGoingCourse ogc JOIN

Course c ON

ogc.Course\_Id = c.Course\_Id JOIN

Participant p ON

ogc.Participant\_Id = p.Participant\_Id

WHERE

[Status] LIKE 'Active' AND

YEAR([Start\_date]) <= 2022 AND

DATE(End\_date) BETWEEN 1 AND 31 AND

MONTH(End\_date) = 1 AND

YEAR(End\_date) = 2022

4.

CREATE TRIGGER update\_participant

AFTER UPDATE

ON Participant

AS

PRINT 'Data has been recorded'

5. Inconsistent Analysis Problem.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time | T1 | T2 | T3 | Balx | Baly | Sum |
| 1 |  | BEGIN TRAN |  | 150 | 50 | 0 |
| 2 | Sum = 0 | BEGIN TRAN |
| 3 | BEGIN TRAN | Read\_lock(Balx) | Write\_lock(Balx) |
| 4 | Write\_lock(Baly) | WAIT | Read(Balx) |
| 5 | Read\_lock(Baly) | Balx += 50 |
| 6 | WAIT | Write(Balx) | 200 |
| 7 | COMMIT or Unlock(Balx) |
| 8 | Read(Balx) |  |
| 9 | Sum += Balx | 200 |
| 10 | Read\_lock(Baly) |
| 11 | Read(Baly) | WAIT |
| 12 | Baly -= 20 |
| 13 | Write(Baly) | 30 |
| 14 | COMMIT or Unlock(Baly) |
| 15 |  | Read(Baly) |
| 16 | Sum += Baly | 170 |
| 17 | COMMIT or  Unlock(Balx, Baly) |

6. Stripping (Level 0) merupakan konsep RAID yang memecah atau membagi data ke beberapa disk, sedangkan Mirroring (Level 1) merupakan konsep RAID yang menyalin atau membuat tiruan data untuk dimasukkan ke beberapa (atau lebih dari satu) disk. Setelah data dimasukkan ke dalam disk, Stripping bekerja dengan membagi setiap disk dalam beberapa strip (sebaris blok data pada setiap disk yang saling berbeda). Dengan begitu, konsep ini akan memiliki **performa** yang baik karena kita dapat mengakses lebih dari satu (blok data berbeda pada setiap) disk sekaligus dalam satu strip sebelum beralih ke strip berikutnya. Meskipun demikian, namun konsep ini juga disebut Nonredundant karena tidak memberikan proteksi tambahan terhadap data-data yang ternyata dimasukkan apa adanya ke dalam disk dan tidak bertanggung jawab terhadap jika terjadi kehilangan data. Hal ini berbeda dengan Mirroring yang dapat menyimpan data yang sama dalam disk berbeda, sehingga jika terjadi kehilangan data yang semisal diakibatkan adanya (sebuah) disk yang rusak, maka kita tetap dapat mengambil data cadangan pada disk lainnya (yang menyimpan data yang sama) atau memasang disk baru yang dapat rebuilding atau bersinkronisasi dengan disk lainnya tersebut. Dari semuanya, Stepping dapat menampung lebih banyak data berbeda (**berkapasitas** lebih besar), sedangkan Mirroring lebih dapat **diandalkan** atau bisa dibilang lebih aman. Di sisi lain, saya berpendapat bahwa Nested RAID 1+0 sangatlah cocok untuk GoTraining Database. Hal ini karena tipe RAID tersebut sesungguhnya merupakan RAID Level 1 dengan tingkat keandalan yang baik dan dikombinasikan dengan RAID Level 0 yang berkinerja cukup baik. Selain itu, baik Stripping maupun Mirroring merupakan kedua konsep dasar yang mudah dipahami, sehingga dapat lebih mudah baik untuk diimplementasikan maupun dikombinasikan.

7. Vertical Fragmentation yang tercocok untuk mendekomposisi tabel Participant jika dibandingkan baik dengan Horizontal maupun Mixed Fragmentation. Hal ini bertujuan agar pembagian tabel (berdasarkan ID, nama lengkap, dan kontak) untuk penyimpanan antarsistem tampak lebih tertata rapi.

Tabel\_1:

|  |  |  |
| --- | --- | --- |
| Client\_Id | Participant\_Id | Tuple\_Id |
| C12 | PP152 | 1 |
| C12 | PP514 | 2 |
| C11 | PP432 | 3 |
| C10 | PP651 | 4 |

Tabel\_2:

|  |  |  |
| --- | --- | --- |
| PFirst\_Name | PLast\_Name | Tuple\_Id |
| David | Ford | 1 |
| Matthew | Lee | 2 |
| Maria | White | 3 |
| Susan | Susan | 4 |

Tabel\_3:

|  |  |  |
| --- | --- | --- |
| Phone\_Number | Participant\_email | Tuple\_Id |
| ++6286161717 | [David.Ford@mail.com](mailto:David.Ford@mail.com) | 1 |
| +62856171611 | [Matthew.Lee@mail.com](mailto:Matthew.Lee@mail.com) | 2 |
| +62815625416 | [Maria.White@mail.com](mailto:Maria.White@mail.com) | 3 |
| +62816161719 | [Susan.susan@mail.com](mailto:Susan.susan@mail.com) | 4 |

Proyeksi untuk kembali ke tabel awal (Tabel Participant): πParticipant(Tabel\_1**⋈Tabel\_2⋈Tabel\_3**).

8. Video: <https://www.loom.com/share/4a9e6e54aa2d45d1b1a9daacc1b911f9>.