

1. ອະທິບາຍວິວັດທະນາການຂອງລະບົບການຈັດການຖານຂໍ້ມູນທີ່ແບບກະຈາຍ.

ໃນຊຸມປີ 1970, ອົງການຈັດຕັ້ງ / ບໍລິສັດໄດ້ສ້າງລະບົບຖານຂໍ້ມູນໃນຮູບແບບສູນກາງ.(centralized database) ເຊິ່ງຈະເປັນລະບົບຖານຂໍ້ມູນທີ່ເກັບໄວ້ໃນຄອມພິວເຕີ/ເຊີບເວີສໍາລັບການນໍາໃຊ້ໃນການຈັດເກັບຂໍ້ມູນທີ່ກ່ຽວຂ້ອງກັບການດໍາເນີນທຸລະກິດ. ເຫດຜົນຕົ້ນຕໍສໍາລັບການເກີດຂຶ້ນຂອງລະບົບການຄຸ້ມຄອງຖານຂໍ້ມູນທີ່ແຈກຢາຍແມ່ນຄວາມຕ້ອງການທີ່ຈະປັບຕົວເຂົ້າກັບປະລິມານຂໍ້ມູນທີ່ເພີ່ມຂຶ້ນ.ແລະຄວາມຕ້ອງການຄວາມຍືດຫຍຸ່ນໃນການເຊື່ອມຕໍ່ແລະການເຂົ້າເຖິງຂໍ້ມູນຈາກທຸກບ່ອນ, ໂດຍບໍ່ຄໍານຶງເຖິງສະຖານທີ່ຂອງຂໍ້ມູນ

2. ອະທິບາຍຂໍ້ດີ ແລະ ຂໍ້ເສຍຂອງລະບົບການຈັດການຖານຂໍ້ມູນແບບກະຈາຍ.

ຂໍ້ດີ: ປະສິດທິພາບສູງ, ມີຄວາມຍືດຍຸ່ນ, ຄວາມທົນທານ ແລະ ການຮັບມືກັບຂໍ້ມູນທີ່ຫຼາຍໄດ້
ຂໍ້ເສຍ: ມີຄວາມຊັບຊ້ອນໃນການຈັດການ, ຄ່າໃຊ້ຈ່າຍສູງ, ມີຄວາມສ່ຽງສູງ, ແລະ ມີຄວາມຊັບຊ້ອນໃນການຮັກສາ.

3. ຈົ່ງອະທິບາຍເຖິງຄວາມແຕກຕ່າງລະຫວ່າງຖານຂໍ້ມູນແບບກະຈາຍແລະການປະມວນຜົນແບບກະຈາຍ

ຖານຂໍ້ມູນທີ່ແບບກະຈາຍ: ເປັນການຈັດເກັບຂໍ້ມູນໃນຮູບແບບທີ່ແຕກຕ່າງກັນບາງເທື່ອໃນຫຼາຍເຄື່ອງໂມງພິວເຕີທີ່ຕໍ່ກັນຜ່ານເຄື່ອງໂມງ ໂດຍຂໍ້ມູນອາດຈະຖືກແບ່ງແຍກເປັນສ່ວນຍ່ອຍ (fragments) ແລະກະຈາຍໄປຍັງເຄື່ອງໂມງພິວເຕີຕ່າງ ໆ ທີ່ເຊື່ອມຕໍ່ກັນເພື່ອເພີ່ມປະສິດທິພາບແລະຄວາມເຂົ້າເຖິງໃນການເກັບຂໍ້ມູນ
ການປະມວນຜົນແບບກະຈາຍ: ເປັນການກະຈາຍການປະມວນຜົນຂອງໂປຣແກຣມຫຼືວຽກງານຄໍານວນໃນຮູບແບບຂອງການແບ່ງງານ (task) ໄປຍັງຫຼາຍເຄື່ອງໂມງພິວເຕີທີ່ເຊື່ອມຕໍ່ກັນຜ່ານເຄື່ອງໂມງ ແຕ່ລະເຄື່ອງໂມງພິວເຕີຈະເຮັດວຽກຮ່ວມ ໆ ກັນເພື່ອປະມວນຜົນຂໍ້ມູນ

4. ຍົກຕົວຢ່າງ ແລະ ອະທິບາຍອົງປະກອບຂອງລະບົບການຈັດການຖານຂໍ້ມູນທີ່ແບບກະຈາຍ.

ຕົວຢ່າງ: ລະບົບຖານຂໍ້ມູນສໍາລັບທຸລະກິດຫຼາຍສາຂາ

ອົງປະກອບ: ຄອມພິວເຕີທີ່ມີລັກສະນະເປັນແບບ workstation ຫຼາຍ ໆ ເຄື່ອງທີ່ເຊື່ອມຕໍ່ກັນເປັນເຄື່ອງໂມງ, ສອຟຕ໌ແວຣ໌ແລະຮາດແວຣ໌ເຄື່ອງໂມງທີ່ຖືກຕິດຕັ້ງຢູ່ໃນຄອມພິວເຕີເຄື່ອງຕ່າງ ໆ, Communication media, Transaction processor (TP), Data processor (DP)

5. จังหวะขยายเท็จเจ้าลักสละ transparency ต่างๆของระบบจัดการฐานข้อมูลแบบกระจาย

- Distribution transparency จะเป็นคุณลักษณะที่จะอนุญาตให้ฐานข้อมูลแบบกระจายจะถูกระบุหรือพิจารณาว่าเป็นฐานข้อมูลภายในคอมพิวเตอร์เครื่อง หนึ่ง ๆ เท่านั้น
- Transaction transparency จะเป็นคุณลักษณะที่ทำให้การอัปเดตข้อมูลจะทำการอัปเดตข้อมูลมากกว่าหนึ่ง site
- Failure Transparency: ระบบยังสามารถรับรู้ได้ว่ามีบางส่วนเสียหาย.
- Performance transparency จะเป็นคุณลักษณะที่ทำให้ระบบฐานข้อมูลแบบกระจายสามารถทำงานได้อย่างมีประสิทธิภาพเหมือนกับหรือใกล้เคียงกับระบบฐานข้อมูลแบบศูนย์กลาง ที่ซึ่งถึงแม้ว่าระบบฐานข้อมูลแบบกระจายจะมีการเชื่อมต่อ/เชื่อมโยงกันระหว่างเครือข่ายก็ไม่
- Heterogeneity transparency จะเป็นคุณลักษณะที่จะอนุญาตให้ระบบฐานข้อมูลสามารถรวบรวมข้อมูลที่ได้จากหลายฐานข้อมูลย่อยเข้าด้วยกัน

6. ทำหน้าที่และอธิบายชนิดของ Distribution transparency?

- **Location Transparency:** ให้ความโปร่งใสในเรื่องของตำแหน่งที่เก็บข้อมูล ผู้ใช้สามารถเข้าถึงข้อมูลได้โดยไม่ต้องรู้ว่าข้อมูลนั้นอยู่บนเครื่องใด.
- **Fragmentation Transparency:** ให้ความโปร่งใสในการแบ่งแยกข้อมูลเป็นส่วนย่อย (fragmentation) ผู้ใช้สามารถเข้าถึงข้อมูลได้โดยไม่ต้องรู้ว่าข้อมูลถูกแบ่งออกเป็นส่วนย่อยอย่างไร
- **Local mapping transparency** จะปรากฏก็ต่อเมื่อผู้ใช้หรือผู้เขียนโปรแกรมต้องการที่จะกำหนดชื่อและสถานที่ที่ใช้ในการจัดเก็บฐานข้อมูลย่อย

7. อธิบายคำร้องขอในรูปแบบต่างๆ.

- **Data Access Request** เป็นการร้องขอเพื่อเข้าถึงข้อมูลในฐานข้อมูลแบบกระจาย เช่น การเรียกดูข้อมูล, การค้นหา, หรือการอ่านข้อมูล
- **Processing Request:** เป็นการร้องขอเพื่อประมวลผลข้อมูลหรือทำงานที่เกี่ยวข้องกับข้อมูลในระบบฐานข้อมูลแบบกระจาย
- **Management Request** เป็นการร้องขอเพื่อการจัดการและควบคุมระบบฐานข้อมูลแบบกระจาย
- **Control Request** เป็นการร้องขอที่เกี่ยวข้องกับการควบคุมการทำงานของระบบฐานข้อมูลแบบกระจาย

8. การดำเนินการ query optimization ในระบบฐานข้อมูลมีวัตถุประสงค์หลักๆ อยู่เพื่อการปรับปรุงประสิทธิภาพของคำสั่ง query ในการเข้าถึงและดึงข้อมูลจากฐานข้อมูล โดยทำให้คำสั่ง query ทำงานได้เร็วขึ้น ลดการใช้ทรัพยากรของระบบชนิดของการดำเนินการ query optimization

1. การแบ่งแยกข้อมูล (Data Partitioning)
2. การเลือกแม่แบบการกระจายข้อมูล (Choosing Data Distribution Patterns)
3. การเลือกตัวแปรและดัชนี (Variable and Index Selection)
4. การแบ่งการจัดการคำสั่ง query (Query Decomposition)
5. การแบ่งการจัดการแบบขั้นสูง (Advanced Query Decomposition)

9. อธิบายวิธีการแบ่งข้อมูลออกเป็นส่วนน้อย (Data Fragmentation) อย่างไร
ประกอบมี 3 วิธี, มีตัวอย่าง.

-Horizontal Fragmentation: แบ่งข้อมูลโดยแถว, เช่น: ฐานข้อมูลของพนักงานโดยอาชีพ.

-Vertical Fragmentation: แบ่งข้อมูลตามต้น เช่น: ฐานข้อมูลพนักงานเงินเดือน และข้อมูลพนักงาน.

-Mixed Fragmentation: ปะสมแบบวนรอบและแนวตั้ง เช่น: ฐานข้อมูลพนักงานเงินเดือน และต้น.

10. การซ้ำซ้อนของข้อมูล (data replication) หมายถึงการทำซ้ำข้อมูลภายในระบบฐานข้อมูล โดยการเก็บข้อมูลเดียวกันในหลายๆ ที่ เป้าหมายหลักของการทำซ้ำซ้อนข้อมูลคือเพื่อเพิ่มความถูกต้อง ความเชื่อถือได้ และประสิทธิภาพในการเข้าถึงข้อมูล

วิธีการที่ใช้ในการทำให้ข้อมูลมีการซ้ำซ้อนมีดังนี้:

1. การทำซ้ำแบบแผนที่ (Replication by Mapping)
2. การทำซ้ำแบบเวียนเรื่อง (Replication by Propagation)
3. การทำซ้ำแบบเลือก (Selective Replication)

11. Normalization คือการปรับปรุงข้อมูลที่อยู่ใน Database ทำเพื่อตรวจสอบ แก้ไขการซ้ำกันของข้อมูล และลดความ

ซ้ำซ้อนของข้อมูลใน Database อีกด้วย อีกทั้งช่วยรักษาความถูกต้องให้ข้อมูลโดยให้อยู่ในรูปแบบ Normal Form (NF).



การ Normalization มีได้ถึง 5 ระดับ (บางตำราอาจบอกว่ามีถึง 6 ระดับ) แต่ในระดับที่ 3 ก็ถือว่าเพียงพอสำหรับการ

ออกแบบ Database ในปัจจุบันแล้วซึ่งประกอบไปด้วย

✓ 1NF: กำจัด Repeating Group

✓ 2NF: กำจัด Partial Dependency

✓ 3NF: กำจัด Transitive Dependency

1. ขั้นตอนการท 1 First Normal Form(1NF)

2. ขั้นตอนการท 2 Second Normal Form(2NF)

3. ขั้นตอนการท 3 Third Normal Form(3NF)

4. ขั้นตอนการท 4 Boyce-Codd Normal Form(BCNF) \

5. ขั้นตอนการท 5 Fourth Normal Form(4NF)

6. ขั้นตอนการท 6 Fifth Normal Form(5NF)

12. ภาษา SQL แหม่นຫຍັງ? SQL ຫຼື Structured Query Language ແມ່ນພາສາທີ່ໃຊ້ໃນການ

ເຂົ້າເຖິງຖານຂໍ້ມູນຈາກການດຶງຂໍ້ມູນໄປຫາການຈັດການຂໍ້ມູນ SQL ເປັນພາສາການຂຽນໂປລແກລມທີ່ໄດ້

ຮັບການຍອມຮັບຢ່າງກວ້າງຂວາງທີ່ສຸດ. ພວກເຮົາມັກຈະໃຊ້ພາສາ SQL ຜ່ານໂປຼແກຼມຕ່າງໆ. ທີ່ຕ້ອງເຮັດ

ວຽກກັບລະບົບຖານຂໍ້ມູນ

13. Horizontal fragmentation

Empid	Name	State	Age	Salary
43125	Nancy	New York	28	4500
43126	Andrew	Texas	26	4250
43129	Janet	New York	32	5000
43135	Margaret	New York	47	8500
43136	Steven	Texas	36	6500
43139	Michael	Texas	24	4000
43140	Robert	Texas	29	4500

- CREATE TABLE Employee (Empid INT, Name VARCHAR(50), State VARCHAR(50), Age INT, Salary INT);
- INSERT INTO Employee (Empid, Name, State, Age, Salary)
VALUES (43125, 'Nancy', 'New York', 28, 4500), (43126, 'Andrew', 'Texas', 26, 4250), (43129, 'Janet', 'New York', 32, 5000), (43135, 'Margaret', 'New York', 47, 8500), (43136, 'Steven', 'Texas', 36, 6500), (43139, 'Michael', 'Texas', 24, 4000), (43140, 'Robert', 'Texas', 29, 4500);
- CREATE TABLE Employee_NewYork AS SELECT * FROM Employee WHERE State = 'New York';
- CREATE TABLE Employee_Texas AS SELECT * FROM Employee WHERE State = 'Texas';

15. ການດຶງຂໍ້ມູນ ຈາກແຖວມາເປັນຖັນ(Row to Column)

```
CREATE TABLE Sales (  
    Product VARCHAR(50),  
    Month VARCHAR(3),  
    Sales INT  
);
```

```
INSERT INTO Sales (Product, Month, Sales)  
VALUES  
    ('A', 'Jan', 100),  
    ('A', 'Feb', 200),  
    ('A', 'Mar', 300),  
    ('B', 'Jan', 150),  
    ('B', 'Feb', 250),  
    ('B', 'Mar', 350);
```

Sales table:

Product	Month	Sales
A	Jan	100
A	Feb	200
A	Mar	300
B	Jan	150
B	Feb	250
B	Mar	350

```
SELECT Product, Jan, Feb, Mar  
FROM sales  
PIVOT  
(  
    SUM(Sales)  
    FOR Month IN (Jan, Feb, Mar)  
) AS SalesByMonth;
```

Product	Jan	Feb	Mar
A	100	200	300
B	150	250	350

14. Vertical fragmentation

```
CREATE TABLE Employee_NewYork AS SELECT * FROM Employee  
WHERE State = 'New York';
```

```
CREATE TABLE Employee_Texas AS SELECT * FROM Employee  
WHERE State = 'Texas';
```