

§2. Relation model.

relationship.
column/attribute.
relation/tuple.

• Division $R \div S$.
find maximal $Q, Q \times S \subseteq R$.

$R \div S = \Pi_{R-S}(R) - \Pi_{R-S}((\Pi_{R-S}(R) \times S) - R)$.

SELECT sid FROM student AS s
WHERE NOT EXISTS
SELECT sid FROM course AS c EXCEPT
SELECT sid FROM enrollment WHERE
enr.sid = sc.sid.

key.

super key. (unique identification).

candidate key. unminimized

primary key. (selected)

foreign key. referential integrity constraint.

SELECT FROM.

select σ rename ρ . AS.

project Π

union $\cup$ UNION.

intersection $\cap$. INTERSECT

diff - EXCEPT.

NATURAL JOIN

$\bowtie$ $\bowtie_p$ $\bowtie_{pk}$ $\bowtie_{sk}$ $\bowtie_{all}$
Join using θ .
ON 1.0 = 2.9.

agg.

§7. Relational Database Design.

Decomposition: $R \rightarrow \{R_1, \dots, R_n\}$ that good & preserves.

Lossless: $r = \Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) \Leftrightarrow (R_1 \cap R_2 \rightarrow R_1) \vee (R_1 \cap R_2 \rightarrow R_2)$.

Normal Forms: 1NF $\xrightarrow{\text{partial}}$ 2NF $\xrightarrow{\text{trans}}$ 3NF $\xrightarrow{\text{not trans}}$ 4NF $\xrightarrow{\text{multivalued}}$ 5NF.
(break dep.)

Functional Dependency

$R = \{A, B, \dots\}$. $F = \{A \rightarrow B, \dots\}$.

Dependency Preserving:

$(\cup F_i)^+ = F^+$.

Armstrong's Axioms (F⁺)

1. $\beta \subseteq \alpha \Rightarrow \alpha \rightarrow \beta$. 2. $\alpha \rightarrow \beta \wedge \gamma \rightarrow \delta \Rightarrow \alpha \gamma \rightarrow \beta \delta$. 3. $\alpha \rightarrow \beta, \beta \rightarrow \gamma \Rightarrow \alpha \rightarrow \gamma$.

Attribute Closure A^+ . UBO.

Canonical Cover (F_c) "minimized".

Repeat 1. $\alpha \rightarrow \beta, \alpha \rightarrow \gamma \Rightarrow \alpha \rightarrow \beta \gamma$.

2. $\alpha_1 \alpha_2 \rightarrow \beta_1 \beta_2 \Rightarrow \alpha_1 \rightarrow \beta_1$.

BCNF

$\forall \alpha \rightarrow \beta \in F^+$. at least one

1. $\alpha \rightarrow \beta$ is trivial.

§3. Intro to SQL.

DataDefLang. DataManipulationLang. DataRetLang.

DDL

CREATE TABLE branch (branch-name char(20) primary key,
foreign key (branch-city)).

DROP TABLE branch;

ALTER TABLE branch ADD/DROP branch-city;

DML

SELECT α FROM table WHERE $b > 60$. $\Pi_{\alpha} \sigma_{b > 60}(\text{table})$
AS t_1 .

INSERT INTO table VALUES (

DELETE FROM table WHERE ...

UPDATE table SET ... WHERE ...

SELECT branch-name, avg(balance) FROM branch.

GROUP BY branch-name. HAVING avg(balance) > 1500.

FROM $\Rightarrow$ WHERE $\rightarrow$ GROUP $\rightarrow$ HAVING $\rightarrow$ SELECT $\rightarrow$ DISTINCT $\rightarrow$ ORDER.

unknown.

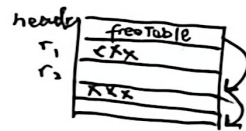
null. [VAR is null] VAR = null $\rightarrow$ unknown. ignored except COUNT(*).

IN. NOT IN. EXISTS.

JOIN.

§9. File Org.

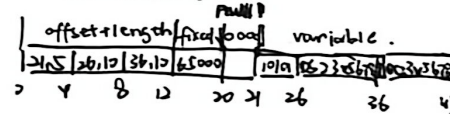
Fixed-Length Record



Sorted Page



Variable.



FileOrg.

Heap. free-space map.

Sequential: overflow, reorg.

Multiple Clustering.

B+.

Hash.

BufferManager:

row 1 2 3 4 5

col 1 2 3 4 5

pro: select β , update, zip rate, vector
con: update & delete. $\rightarrow$ zip time.

§8. Phys Storage.

cache
mainmem } primary
volatile.

FreshMem.
magneticDisk } secondary

OpticalDisk.
magneticTapes. } tertiary

capacity.
AccessTime = seekTime + $\frac{1}{2} \text{maxSeek}$

Rotational Latency: $\frac{1}{2} T$.
Data Transfer Rate: I/O Per Second. Mean Time To Fail.

RAID. R条. 并行.
Buffer-Replacement: LRU/MRU

RAID0: 2块分条.
RAID1: 镜像+冗余分条.
RAID5: 2块分条+冗余分条.
RAID6: 3块分条.

§10. Indexing

Primary Index. (clustering) / secondary.

Sparse/Dense. multilevel.

B+ Tree Index.

internal: $\lceil \frac{n}{2} \rceil$, n children.

leaf: $\lceil \frac{n-1}{2} \rceil$, n-1 vals.

insert: split. first $\lceil \frac{n}{2} \rceil$.

delete: merge, or distribute.

estimate:

$h = \lceil \log_{\lceil \frac{n}{2} \rceil} \lceil \frac{k}{\lceil \frac{n}{2} \rceil} \rceil \rceil + 1$.

$\lceil \log_{\lceil \frac{n}{2} \rceil} \lceil \frac{k}{\lceil \frac{n}{2} \rceil} \rceil \rceil + 1$.

LSM Tree. (Log-Structured Merge).

Bitmap.

§11. Query Processing.

parse, trans, optimize, evaluate. Disk I/O.
block seek (random).
cost: $b \cdot t_f + S \cdot t_s$

1. Selection σ .

- A1. Linear Search. $s=1, b=b_r, \sigma_{a=1} \text{ equality } a_{ij}$
 $t_s + b_r \cdot t_b / t_f + \lfloor \frac{b_r}{b_b} \rfloor t_b$
- A2. B⁺ Eq On Key $s=b+h+1, (h+1) \cdot (t_r+t_s)$
- A3. B⁺ Eq On Nonkey $s=h+1, b=h+t_b$
- A4. 2nd B⁺ Eq Key $s=b+h+1$
- A4. 2nd B⁺ Eq Nonkey $s=b+h+n$
- A5. B⁺ Comp $s=h+1, b=h+t_b$
- A6. 2nd B⁺ Comp $s=b+h+n$

2. Sorting

External Sort. $m-1$ input, $m-1$ output.
M 块内可用 buf 数. b_r 块大小. b_b 块数.
Q 生成 sorted runs $N_0 = \lfloor \frac{b_r}{m} \rfloor, b=2b_r$
③ Merge Pass. 每 $m-1$ runs.

通过数 $p = \lceil \log_{m-1} N_0 \rceil, N=108, B=5$
 $p = \lceil \log_{4} \frac{108}{5} \rceil = 3$
 $b = b_r (2^{p+1})$
#0: $\lceil \frac{108}{5} \rceil = 22$ (x5)
#1: $\lceil \frac{22}{5} \rceil = 5$ (x5)
#2: $\lceil \frac{5}{5} \rceil = 1$ (x5)
#3: 1. (x5)
passes = 4.

§12. Query Opt.

Equivalence Rules.

σ 拆开. 交换. $\sigma_{\theta_1 \wedge \theta_2}(E) = \sigma_{\theta_1}(\sigma_{\theta_2}(E)) = \sigma_{\theta_2}(\sigma_{\theta_1}(E))$.
 π 合并. $\pi_{L_1} \pi_{L_2} \pi_{L_3}(E) = \pi_{L_1}(E)$
 σ 下推. $\sigma_{\theta}(E_1 \bowtie E_2) = \sigma_{\theta}(E_1) \bowtie E_2$. (θ 只涉及 E_1 的属性)
 $\bowtie$ 的定义. $\sigma_{\theta}(E_1 \bowtie E_2) = E_1 \bowtie_{\theta} E_2, \sigma_{\theta_1 \wedge \theta_2}(E_1 \bowtie E_2) = E_1 \bowtie_{\theta_1} E_2$
 $\bowtie$ 交换结合. $E_1 \bowtie E_2 = E_2 \bowtie E_1; (E_1 \bowtie E_2) \bowtie E_3 = E_1 \bowtie (E_2 \bowtie E_3)$
 $(E_1 \bowtie_{\theta_1} E_2) \bowtie_{\theta_2 \wedge \theta_3} E_3 = E_1 \bowtie_{\theta_1 \wedge \theta_2} (E_2 \bowtie_{\theta_3} E_3)$

Size Estimation

直方图法. 属性 A 在 r 中有多少个值.
 $n_r, b_r, l_r, f_r, V(A, r)$. histogram
blocking factor. equi-width/depth .

Selection.

$$|\sigma_{A=a}(r)| = \frac{1}{V(A, r)} \cdot n_r$$
$$|\sigma_{A \leq b}(r)| = \frac{b-a}{\max(V(A, r)) - \min(V(A, r))} \cdot n_r$$

Complex:

$$\text{conj } |\sigma_{\theta_1 \wedge \theta_2 \wedge \theta_3}(r)| = \frac{s_1}{n_r} \cdot \frac{s_2}{n_r} \cdot \frac{s_3}{n_r} \cdot n_r$$

$$\text{disj } |\sigma_{\theta_1 \vee \theta_2 \vee \theta_3}(r)| = [1 - (1 - \frac{s_1}{n_r})(1 - \frac{s_2}{n_r})(1 - \frac{s_3}{n_r})] \cdot n_r$$

3. Join $\bowtie$.

Nested-Loop Join: $\text{worst } b_r \cdot b_r + n_r \cdot b_s$
 $s = n_r + b_r$
 $b = b_r + b_s$
 $s = 2$

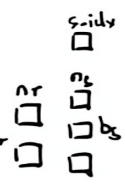
Block Nested-loop Join $b = b_r + b_r \cdot b_s$
 $s = b_r + b_r$

Index nested-loop join $T = b_r(t_t + t_s) + n_r \cdot c$

Merge Join. r, s sorted by θ . 设每块 b_b 个块中只.
 $b = b_r + b_s, s = \lceil \frac{b_r}{b_b} \rceil + \lceil \frac{b_s}{b_b} \rceil$ hybrid (s 块内有 idx).

Hash Join

$$b = 3(b_r + b_s) + 4n_r$$
$$\text{partition 块 } b_r + b_s, \text{ 总 } b_r + b_s + 2n_r$$
$$\text{join: } (b_r + b_s) + 2n_r$$
$$s = 2(\lceil \frac{b_r}{b_b} \rceil + \lceil \frac{b_s}{b_b} \rceil)$$
$$\text{partition:}$$
$$\text{worst } 2(b_r + b_s \lceil \log_{m-1} b_s - 1 \rceil) + b_r + b_s$$



§13. Transaction L.

ALID. Atomic. (consistency. Isolation. Durability).
Concurrency. (性能/IO).
Serializability (可串行化). seemingly fair.
Conflict. $R-W, W-W$.
通过给 T_i, 不同事务, 相同数据项, 至少一个 write.
Anomalies: Dirty Read, Unrepeatable Read, Lost Update, Phantom.
无 conflict 则可交换, 否则 serializability 冲突等价.
Precedent graph.
 $T_i \rightarrow T_j$ if conflict & T_i accessed data earlier.
conflict serializable $\Rightarrow$ acyclic.
order: topological sorting.

Locking.

Shared/Exclusive lock (R/W) - shrinking.
Two-Phase Protocol (2PL) > $\phi \Rightarrow S \Rightarrow X$
Growing/Shrinking Phase. growing
conflict serializable. 11 为理由 lock point 决定 (最后获取锁时).
Strict 2PL: X 持有到 commit/abort
Rigorous 2PL: SX 持有到.

§14. Concurrency Control.

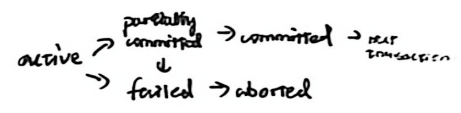
Lock-based Protocols.

Shared/Exclusive lock (R/W) - shrinking.
Two-Phase Protocol (2PL) > $\phi \Rightarrow S \Rightarrow X$
Growing/Shrinking Phase. growing
conflict serializable. 11 为理由 lock point 决定 (最后获取锁时).
Strict 2PL: X 持有到 commit/abort
Rigorous 2PL: SX 持有到.

§15. Recovery Sys.

Transaction failure, System Crash, Disk Failure.
Logical system. 事务 T 修改 $X, V_1 \rightarrow V_2$.
Log: $\langle T_i, X, V_1, V_2 \rangle$.
Undo(T_i), Redo(T_i).
没提交. 提交, 没落盘.

Deferred/Immediate Modification.
WALL Write-Ahead Logging.
commit: $\langle T_i \text{ commit} \rangle$ 写到日志并落盘.
undo. redo. fuzzy: ATT, DPT.
Checkpoint. 日志刷进主存, 脏页刷盘. <checkpoint L>.



Isolation Level:
Read Uncommitted.
Read Committed. 不读 Dirty Read.
Repeatable Read.
Serializable.
Recoverable Schedule.
 T_j 读 T_i 写的. 则 T_i 比 T_j 先提交.
Cascadeless Schedule
 T_j 读 T_i 写的. 则 T_i 已提交.
Timestamp; Snapshot Isolation.

Deadlock Handling.
Prevention: 开始前预防; 按顺序; timeouts
wait-die: old wait, young die
round-wait: old wounds, young waits.
Detection:
wait-for graph. $T_i \rightarrow T_j$: T_i wait for T_j lock
cyclic $\Rightarrow$ deadlock. victim (starvation)

Multiple Granularity

	IS	IS	IX	S	SIX	X
Intention						
holds	IS	IS	IX	S	SIX	X
	IX	IX	IX	S	SIX	X
	SX	SX	IX	S	SIX	X
	X	X	IX	S	SIX	X

Recovery. redo E commit 的. undo 未 commit 的.
Compensation Log Record. (undo 中 crash)
ARIES. (Analysis. Redo. Undo).
Analysis: 从 checkpoint 起更新 ATT (UndoList) 和 DPT.
Redo: 从 DPT 中未落的 RedoLSN 起. Redo 在 DPT. [RedoLSN, PageLSN].
Undo: 选择 UndoList 中最大的 LSN 为 undo. 与 CLR.
Result: committed 的得存. 反之元.

