1. Good Evening
2. Lecture begins at 9:05
3. Topic — Backlog of DBMS

AGENDA

1. CTE ✓
2. Recursive CTE ✓
3. Deadlocks ✓
4. Compound Indexes.
5. Builtin Fns
6. Order of execution of clause in ✓ SELECT
7. Doubts on Views

1. Stored Proc
2. UDFs
3. Full-text Search
4. DataTypes
5. Compound Indexes
6. Builtin Fns.
7. Doubt on Views

CTE : Common Table Expression.

1. Procedural stuff
2. Step by Step.

Products

id	name	qis
1	A	20
2	B	30
3	C	10
4	D	50
5	E	60

Order_Items

order_id	pid	qty
1	2	2
1	3	5
1	4	10
2	1	5
2	4	15
3	3	2
3	1	4

pid	name	aqty
1	A	11
2	B	28
3	C	3
4	D	25
5	E	60

1. ✓ Find products that are ordered (JOIN)
2. ✓ GROUP BY → to find for every product the qty that has been ordered.
3. qis − qordered.

cte1 =

```
SELECT p.product_id, IFNULL(oi.qty, 0) AS qty
FROM Products p
LEFT JOIN order_items oi
     ON p.product_id = oi.product_id
```

pid	qty.o
1	5
1	4
2	2
3	5
3	2
4	10
4	15
5	0

pid	oqty
1	9
2	2
3	7
4	25
5	0

2. cte2 =

```
SELECT pid, SUM(qty) AS oqty
FROM cte1
GROUP BY pid;
```

3. cte3 =

```
SELECT p.id, (p.qis - cte2.oqty) AS rqty
FROM Products p
JOIN cte2 ON p.id = cte2.pid
```

pid	rqty
1	11
2	28
⋮	
5	60
⋮	

SQ in FROM clause is called a derived table.

CTE compares with derived table, but can be prepared sequentially.

Recursive CTE

1. Fibonacci
2. Employees.

$$\begin{tikzcd}
 & & a \arrow[dl, dash] \arrow[dr, dash] & & \\
 & b \arrow[dl, dash] \arrow[d, dash] & & c \arrow[d, dash] \arrow[dr, dash] & \\
d & e & & f & g
\end{tikzcd}$$

Employees (e)

id	name	mto
1	a	null
2	b	1
3	c	1
4	d	2
5	e	2
6	f	3
7	g	3

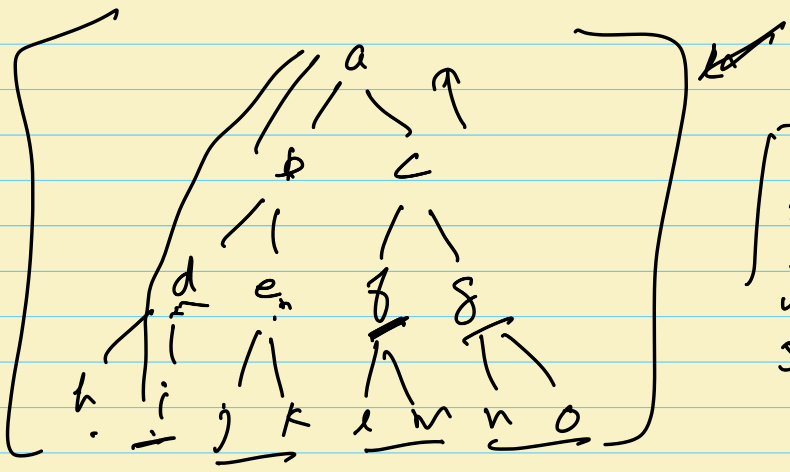
m

1	a	null
2	b	1
3	c	1
4	d	2
5	e	2
6	f	3
7	g	3

emp_name	m_name
a	null
b	a
c	a
d	b
e	b
f	c
g	c

e	p_t_top
a	a
b	b→a
c	c→a
d	d→b→a
e	e-b-a
f	fca
g	gca

```
SELECT   e.n, m.n
    FROM   employees e
LEFT JOIN    employees m
          ON  e.rto = m.id
```



1	a	null
2	b	1
3	c	1
4	d	2
5	e	2
⋮		

en	path_to_top
a	a
b	b-a
c	c-a
d	d-b-a
e	e-b-a
f	f-c-a
g	g-c-a
h	h-d-b-a
i	i-d-b-a
j	j-e-b-a

k	k-c-b-a
l	l f c a
m	m f c a
n	n g c a
o	o g c a

Break

10:48 — 10:53

DEADLOCKS

1. ✓ What? : DBMS is a situation in which DBMS can't process next instruction.
2. ✓ How are they created?
3. [How are they solved?

SERIALIZABLE

T1 = currentTran

→ SELECT *
FROM Accounts ①
WHERE name = 'xxx'
FOR UPDATE;

→ Select * ③
FROM Accounts
WHERE name = 'yyy'
FOR UPDATE;

SERIALIZABLE

T2 = myTran

→ ②

'yyy'

→ ④

'xxx'

xxx ^{T1}

yyy ^{T2}

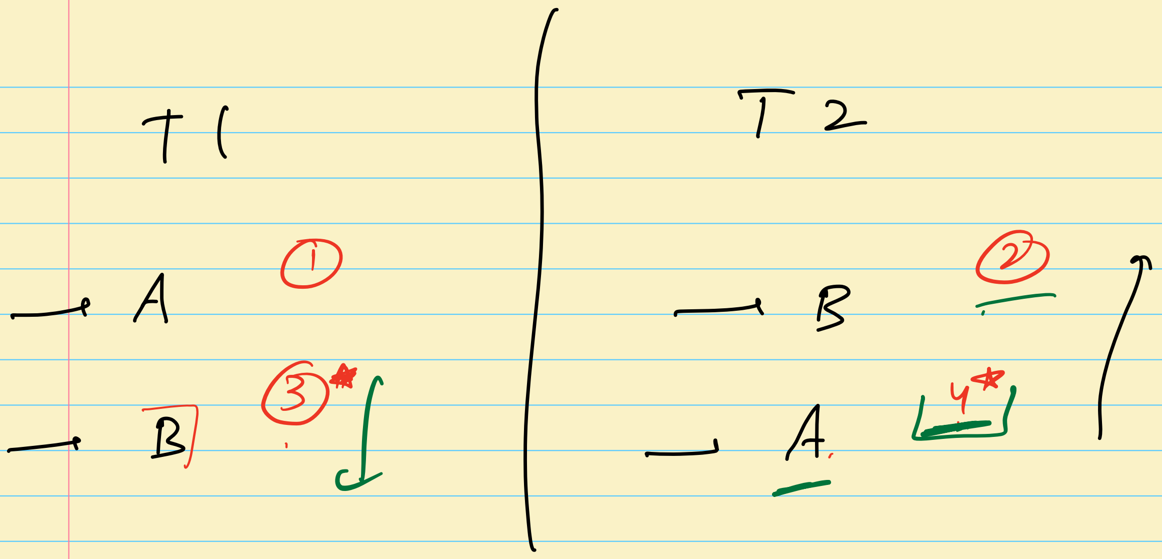
T1 will block at 3 [T1 wants yyy but T2 has it]

T2 will block at 4 [T2 wants xxx but T1 has it]

A B

T1 (A) → wants B

T2 (B) → wants A



A^{T1} B^{T2}

How is it resolved?

T2 will rollback, T1 will proceed.

ORDER OF EXECUTION OF VARIOUS CLAUSE IN SELECT

Appearance in query

SELECT

FROM

WHERE

GROUP BY

HAVING

ORDER BY

LIMIT

Order of Execution

FROM & JOINS

WHERE

GROUP BY

HAVING

SELECT.

ORDER BY.

LIMIT