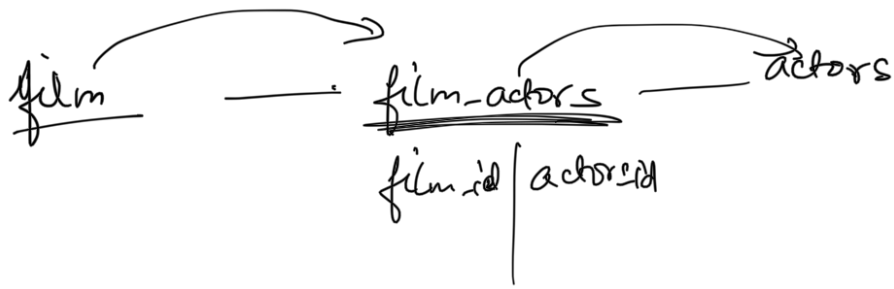


VIEWS & WINDOW FUNCTIONS



Film → all the actors
 Actor → all the films

VIEW

- ① Select statement only.
- ② No update in the view query.

SELECT *

FROM

view

SELECT as ...
 F ---
 --- AS
view1

Query

SELECT *

FROM

view1

⇒

Step 1

SELECT *

FROM

(SELECT actor, film ...
) AS view1

↙ slow ↘ fast

↓

Execute the query

VIEWS

- Multiple teams, multiple members.

↓
Views

WINDOW FUNCTIONS

employees		
emp-no	department	salary

department,		AVG(salary)	
[HR		50,000	
Tech		20,000	

empNo	Dept	salary	avg_salary
1	Tech	60,000	55,000 -
2	Tech	50,000	55,000 -
3	HR	50,000	60,000 -
4	HR	30,000	40,000 -

Window Functions

emp-no	dept	salary	dept_avg	dept_max

10 mins break

10:40 pm → SQL Query to solve problem given

users

id	email	first_name	last_name	...
----	-------	------------	-----------	-----

NULL
"V"
"V"

Hey Madhav,
|||
Hey there

email	salutation
	first_name → "there"

SELECT email,
CASE
WHEN (first_name IS NULL
OR LENGTH(first_name) <= 1)
THEN "there"
ELSE first_name
END AS salutation
FROM users

Switch, case

CASE
→ WHEN (condition 1) THEN output 1
→ WHEN (condition 2) THEN output 2
→ WHEN (condition 3) THEN ...
→
→ ELSE output
END

condition
↓ OR, AND

STORED PROCEDURES

output type \leftarrow Int function (int ^{Input} arg1, str arg2, ...)
=
=
=
=
return ...
- }
 \rightarrow function name

CREATE PROCEDURE film-in-stock (
IN p-film-id INT, p-store-id INT,
OUT p-film-count INT)

SELECT count(*) INTO p-film-count

FROM inventory

WHERE film-id = p-film-id

AND store-id = p-store-id ;

END;;

\downarrow
film-in-stock(12, 20, var-name)

Database

\downarrow
log of all changes (File)
- [] ts
- [] ts
=

Tables \rightarrow Trees

In-memory

flushed
HDD

SQL Query Optimisations

-
- (1) Build the right index.
- (2) Subquery → Join
 ↓ ↓
 result from Index
Subquery
 of
 Index.

3



view-name \longrightarrow $\left(\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \right)$ AS view-name