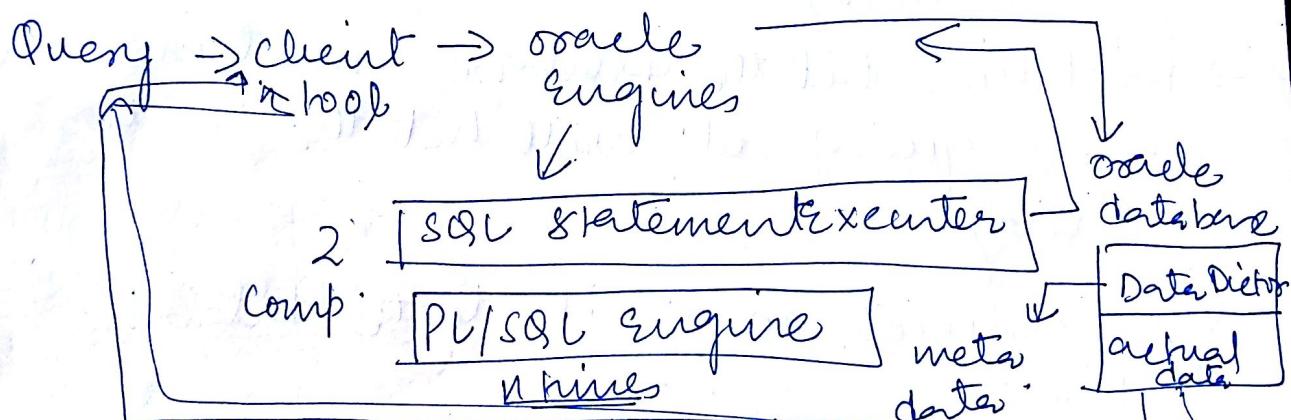


(1)

## View

- View is not maintaining physical table.
  - while as ~~View~~<sup>table</sup> is maintaining physical table
  - View is a database object
  - Contains logical copy of data.
- DB-object → logical copy of data from table.
- View provides security.
  - View increases database performance.

## Query execution process :-

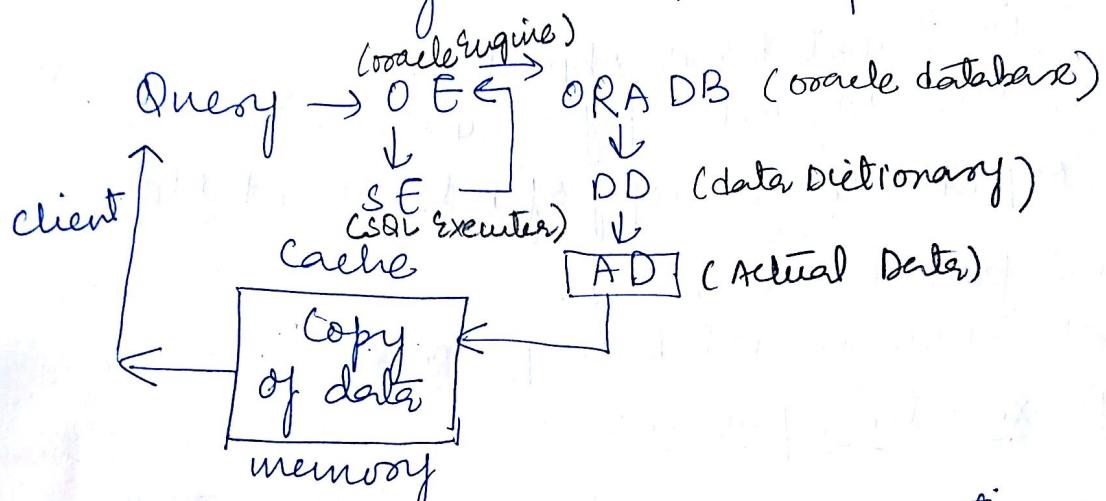


PL/SQL → Group of statements  
 ↳ SQL statements to SQL Statement  
 ↳ PL/SQL executor to the PL/SQL.

Actual data in the form of data blocks.

→ no. of hits increases. → performance decreases.

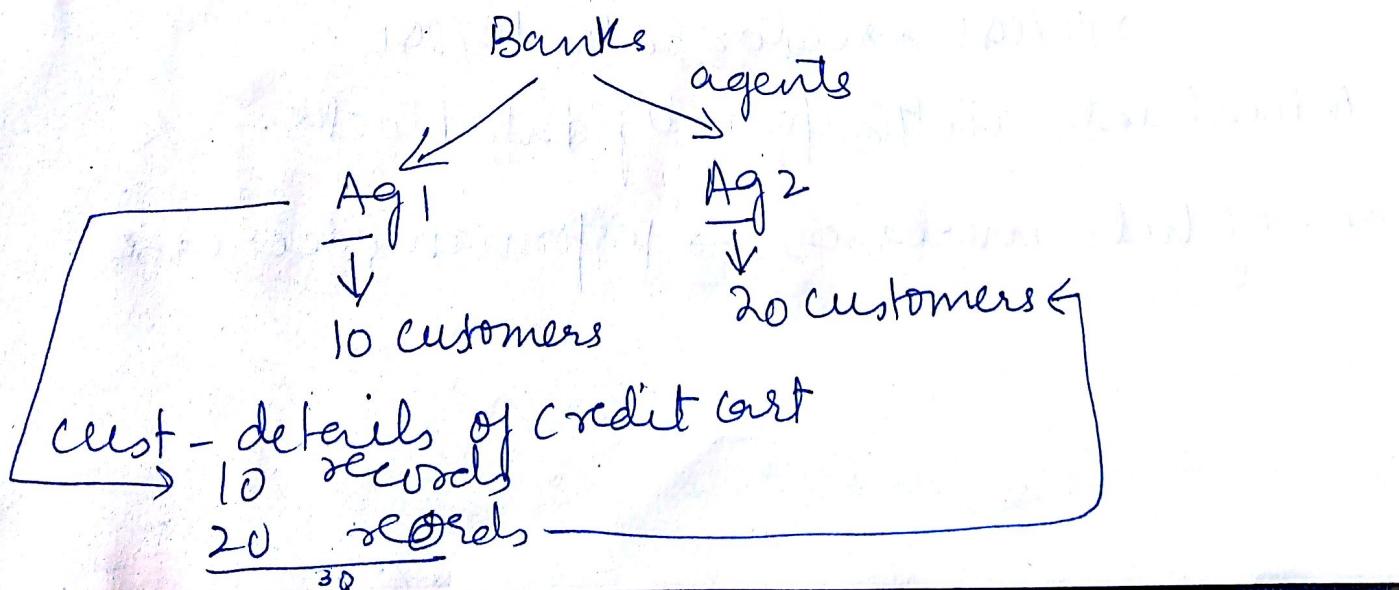
→ If we maintain a view then every time it will not hit the cache memory instead it will fetch the frequent data from Cache memory in the form of view.



→ first time hit the database , next time same query , it will hit the cache memory.

-View reduces no. of hits to the database engine.

### Security provided by View:-



VW - AG1

(2)

↳ Copy of Agent 1 customer data.

VW - AG2

↳ Copy of Agent 2 customer data.

No permission on original data.

### Types of Views:-

- 1) Simple View / Updatable View
- 2) Composite View / Read only View.  
    ↳ Join View.
- c) Foreign View.
- d) Materialized View.
- e) Inline View.

### SIMPLE VIEW:-

- a view that is created by a single table

data

create view <name> as select ... from  
--- where <conditions>  
order by <--->;

SQL > Grant create view to dinesh;

SQL > connect dinesh  
pwd melon

SQL >

Create View VW-Salesman.  
as Select \* from Emp  
where Job = 'Salesman' order by Salary.

- View maintains data dynamically.
- View should not occupy all physical disk space.

Materialized View maintains data physically.

Ex: select ename, sal, Job from VW-Salesman;

View with different column names.

Create View VW-name as Select empno  
"EID", ENAME as "EMPLOYEE-NAME", SAL "SALARY"  
JOB "Description" from Emp where Job = 'MGR';

update → in View actually effects the actual table

accept DML operations called - updatable views.

Composite View or read only.

→ based on Multiple table data.

→ also called Joins View.

Create View VW-Cust-Acc-Info AS

Select C.cid "CustID", } Cust-details  
C.Cname "CustName",  
C.adr.add.no "Acc no", } Cust-all  
C.adr.acc-bal "Balance", } details

(3)

at.act-type "ActType", } act-type.  
 at.act-Name "Act Name":

From cust-details c, cust-act ct, cad,  
 Act-types at where  
 $c.cId = Cad.cId \text{ AND}$

Cad.act-type=at.act-type;

- update, delete - on composite view
- no insert - on composite view
- Maintained under User-Views

### Force View:-

Force View is used basically for the situations when we create a view using a table but the table is not created at the time we use force command.

Syntax :- Create Force View <view-name>  
 as <select Statement>

### Materialized View:-

materialized view is a database object that contains the results of the query. It may be a local copy of data located remotely or may be a subset the rows and/or columns of a table or join result or may be a summary using an aggregate function.

### Inline View:-

An inline view is a Select Statement in the From Clause of another Statement.

These views are commonly used to simplify complex queries by removing joins operations.

```
Select * From( Select dept_no,  
count(*), emp_count From emp  
Group By dept_no ) emp,  
dept where dept_no = emp.dept_no
```