Adam into an problem?

Click to Continue

Adam was working for a leading IT organizations. The application he maintained had a huge database with 350+ tables. Joe, Adam's colleague was developing a report which pulled data from 20+ tables close to around 25 columns. To simplify the report design he wanted to reduce the number of tables he queried. So he requested Adam to create a table with the 25 column from which he expected data.

Adam created database views to solve this problem. Let us see how to create Views.

Database View

Click to Continue

A ***view*** is a logical table built from one or more tables or view(s).

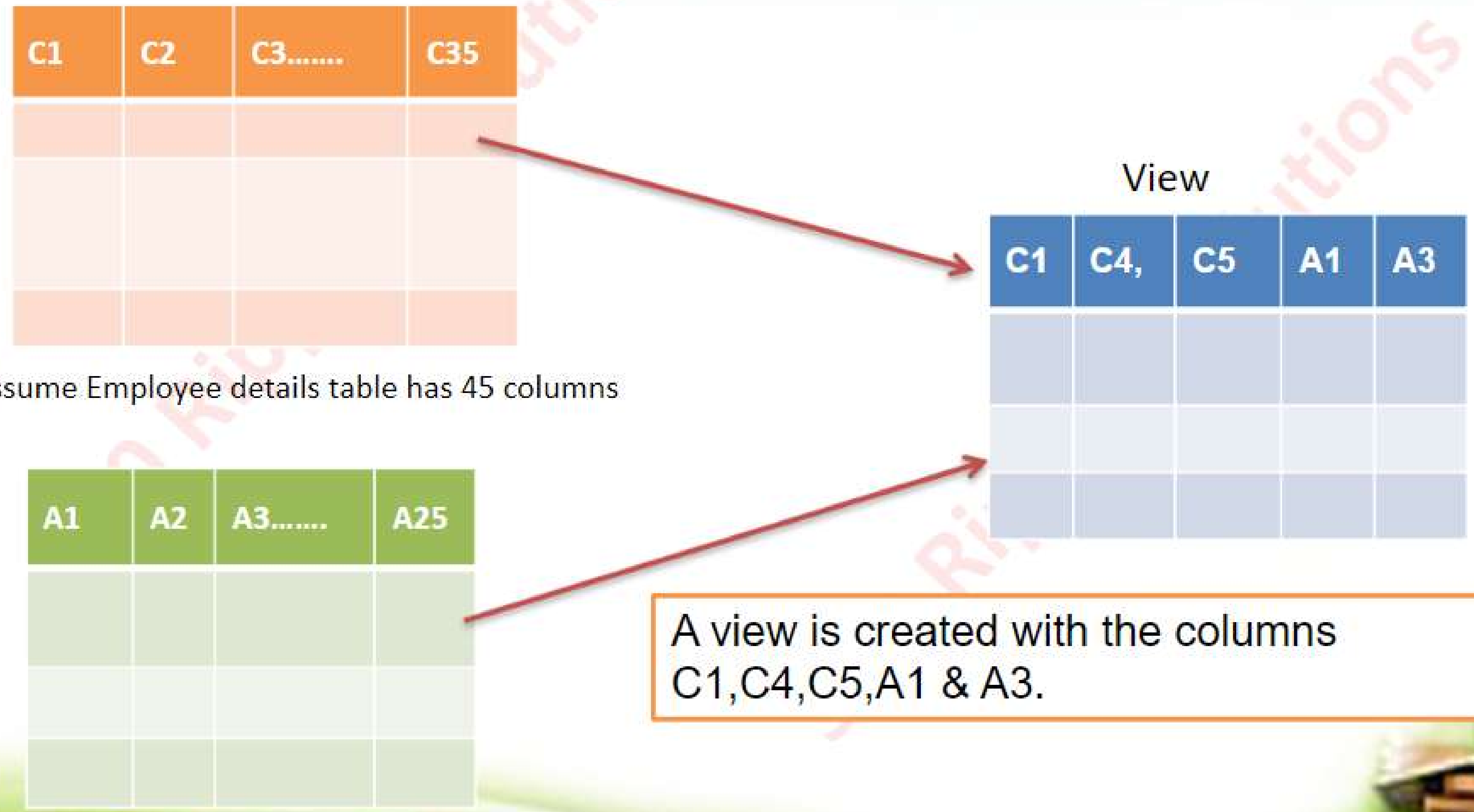
- This holds data's from multiple columns of the selected tables (or) views.
- The tables from which the view is built is referred to as ***"base tables"***.
- The view being a logical table is physically stored as a "***select***" statement in the data dictionary.

A peek into a sample View.

Click to Continue

For developers to create a report to extract data from columns C1,C4,C5 and A1,A3.

Assume an employee table has 35 columns



Assume Employee details table has 45 columns

A view is created with the columns C1,C4,C5,A1 & A3.

Views Advantages

Click to Continue

- Restricted access to data by creating views with the required set of columns from a data table.

  For example: An user has a access only to few columns of a table.

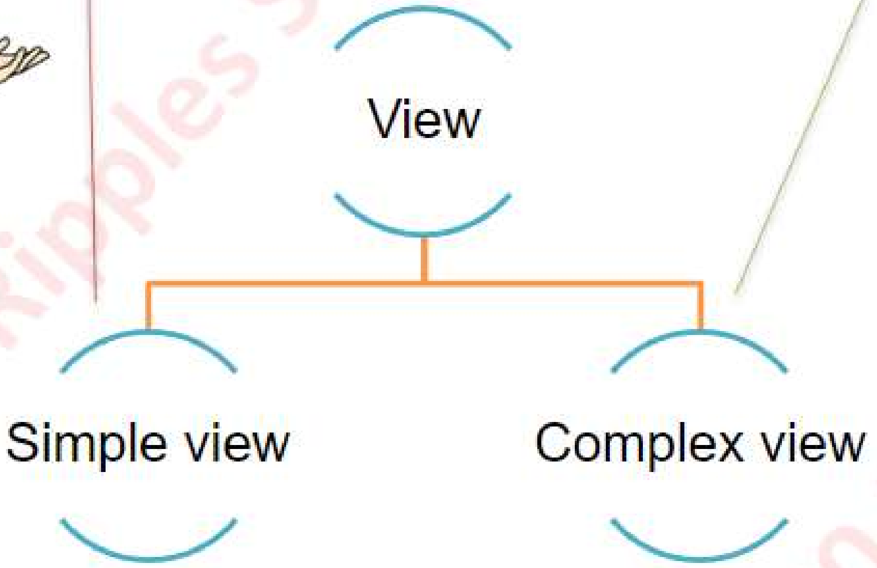
- Simplifies DQL queries to fetch results from a single view rather than multiple table.
- This increases the performance of the data retrieval process.
- Views provide access to data to group of users according to their particular criteria.

Types of Views

Click to Continue

Simple View:
- View created from only one table.
- DML operations can be performed in this view.

Complex view :
- View created from many tables.
- DML operations cannot be performed directly in this view.



How to create a View?

Click to Continue

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW
view_name
[  (alias)  [,alias]…)   ]
AS select statement
[WITH CHECK OPTION [CONSTRAINT constraint_name]];
```

Check option ensures that rows accessible to the view can only be inserted or updated.

Illustration of a simple View

Click to Continue

Scenario: Assume an application has a credit_card_info table with the following columns.

Id	CC_Number	Customer_Name	Credit_Limit	Card Type
1	1234	Jack	42000	Visa
2	4321	Tim	35000	Amex
……	…….	…….	…..	
99	2367	Steve	75000	Visa
100	9876	Johnson	12300	Master

Please create this table in MySQL using your work bench.

Credit Card Table View.

Click to Continue

Create a simple view for the credit_card table in MySQL using your work bench.

```
CREATE VIEW Credit_Card_View AS
   SELECT CC_Number,Customer_Name, Card_Type
   FROM Credit_Card_Info;
```

Here,

- Credit_Card_View – Represents the view name
- Credit_Card_Info – Represents the base table
- CC_Number,Customer_Name, Card_Type – Represents the columns from which data needs to be retrieved to build the view.

How to retrieve from a View?

Click to Continue

Syntax:

```
select * from view_name;
```

To retrieve from the credit card view,

```
select * from Credit_Card_View;
```

CC_Number	Customer_ Name	Card_Type
1234	Jack	Visa
4321	Tim	Amex
…….	…….	
2367	Steve	Visa
9876	Johnson	Master

Please try retrieving from the view in your MySQL work bench.

How to delete a view?

Click to Continue

```
DROP VIEW view_name;
```

Illustration:

```
DROP VIEW Credit_Card_View;
```

Please try deleting the credit card view in MySQL work bench.

View with Check Option's

Click to Continue

"WITH CHECK OPTION" prevents the user from inserting a data in a view violating the constraint.

If mentioned it ensures that every row that is inserted/Updated/Deleted in the view must adhere to the definition of the view.

Check Option Illustration

Click to Continue

Now the credit card view is created for only cards of type Visa,

```
CREATE VIEW Credit_Card_View AS
    SELECT CC_Number,Customer_Name, Card_Type
    FROM Credit_Card_Info where card_type='Visa'  WITH CHECK OPTION
```

If one tries to insert a record in

```
    INSERT INTO Credit_Card_View VALUES(2,'Ram','Master')
```

The above DML will throw an error cause the view was created for credit cards of type Visa and we are trying to insert a Non-Visa card.

Please recreate the view and try this in your MySQL work bench.